UNIVERSIDAD LA SALLE

CARRERA DE INGENIERÍA DE SISTEMAS



"SISTEMA INTELIGENTE DE RECONOCIMIENTO ÓPTICO DE CARACTERES BASADO EN REDES NEURONALES PARA LA DETECCIÓN Y EXTRACCIÓN DE INFORMACIÓN EN DOCUMENTOS DE IDENTIDAD CASO: PARROQUIA SAN MIGUEL ARCÁNGEL"

MODALIDAD DE TITULACIÓN: PROYECTO DE GRADO

POSTULANTE: MAURICIO LEANDRO RAMÍREZ DÁVALOS

TUTOR: ING. ROBERTO CARLOS MAMANI MAYTA

LA PAZ - BOLIVIA

2026

DEDICATORIA

Este trabajo está dedicado a:

A mi madre, Marcela Dávalos, por su amor, sacrificio y apoyo incondicional, que me han permitido seguir adelante y crecer cada día.

A mi hermana, Lucía Ramírez, por su constante apoyo y compañía, que me han motivado a superar los retos de mi formación.

A la memoria de mi abuelo, Silverio Dávalos, quien me acompaña y guía desde el cielo.

ÍNDICE

ÍNDICE DE CONTENIDO

CAPÍTULO 1. GENERALIDADES

1.1. INTRODUCCIÓN 1

1.2. ANTECEDENTES 1

1.2.1. Antecedentes institucionales 2
1.2.1.1. Visión de la institución 2
1.2.1.2. Misión de la institución 3
1.2.2. Antecedentes académicos 3

1.3. PLANTEAMIENTO DEL PROBLEMA 4

1.3.1. Identificación del problema 5
1.3.2. Formulación del problema 5

1.4. OBJETIVOS 6

1.4.1. Objetivo general 6
1.4.2. Objetivos específicos 6

1.5. JUSTIFICACIONES 6

1.5.1. Justificación técnica 6
1.5.2. Justificación económica 7
1.5.3. Justificación social 7

1.6. ALCANCES Y LÍMITES 7

1.6.1. Alcances 8
1.6.1.1. Alcance temático 8
1.6.1.2. Alcance geográfico 8
1.6.1.3. Alcance temporal 8
1.6.2. Límites 8

CAPÍTULO 2. MARCO TEÓRICO

2.1. SISTEMAS DE INFORMACIÓN 10

2.1.1. Propiedades de los sistemas de información 10
2.1.2. Clasificación de los sistemas de información 11

2.2. INGENIERÍA DE SOFTWARE 13

2.2.1. Gestión de proyectos de software 13
2.2.2. Diseño y desarrollo de software 14
2.2.3. Validación de software 16
2.2.3.1. Pruebas de componentes 16

2.2.3.2. Pruebas del sistema 17
2.2.3.3. Pruebas de aceptación 17
2.2.4. Metodologías 17
2.2.4.1. Programación extrema (XP) 17
2.2.4.2. CRISP DM 20

2.3. INTELIGENCIA ARTIFICIAL 22

2.3.1. Machine Learning 23
2.3.1.1. Aprendizaje supervisado 23
2.3.1.2. Aprendizaje no supervisado 24
2.3.1.3. Aprendizaje por refuerzo 25
2.3.2. Visión Artificial 26
2.3.2.1. Proceso de análisis a través de Visión Artificial 26
2.3.2.2. Reconocimiento de patrones 27
2.3.3. Redes Neuronales Artificiales (ANN) 28
2.3.3.1. Modelo de neurona biológica 29
2.3.3.2. Modelo de neurona artificial 30
2.3.3.3. Función de activación 32
2.3.3.4. Arquitectura de las redes neuronal artificiales 33
2.3.4. Redes neuronales profundas (DNNs) 35
2.3.5. Redes neuronales convolucionales (CNNs) 36
2.3.5.1. Capas de convolución 37
2.3.5.2. Capas de pooling 38
2.3.5.3. Yolo 38
2.3.5.4. Faster RCNN 40
2.3.6. Redes neuronales recurrentes (RNNs) 41
2.3.6.1. Celdas Long Short-Term Memory (LSTM) 43
2.3.6.2. Celdas Gated Recurrent Unit (GRU) 43
2.3.7. Metodología de desarrollo de una red neuronal 44

2.4. Herramientas 46

2.4.1. Python 46
2.4.2. Tensorflow 46
2.4.3. Keras 46
2.4.4. Laravel 46
2.4.5. Flask 47
2.4.6. FastAPI 47

CAPÍTULO 3. MARCO APLICATIVO

3.1. PLANIFICACIÓN 48

3.2. ANÁLISIS DE LOS PROCESOS ACTUALES 51

3.2.1. Recopilación de Información Mediante Entrevistas 51
3.2.1.1. Funciones del personal 51

3.2.1.2. Procesos administrativos 52
3.2.2. Observación de las Actividades 55
3.2.2.1. Limitaciones del Sistema Actual 56
3.2.3. Identificación de la Documentación Recepcionada 59
3.2.4. Identificación de Información Extraída 60
3.2.5. Requerimientos de Usuario 60
3.2.6. Historias de Usuario 61
3.2.7. Planificación de Iteraciones 68

3.3. CONSTRUCCIÓN DE DATASETS 70

3.3.1. Dataset de Documentos para la Detección y Clasificación de Texto 70
3.3.1.1. Recopilación y escaneo de documentación 70
3.3.1.2. Generación de documentación sintética 72
3.3.1.3. Etiquetado de información de interés 76
3.3.1.4. Distribución del dataset de documentos 79
3.3.2. Dataset de Fragmentos de Texto para Extracción de Información 80
3.3.2.1. Extracción de fragmentos de texto 80
3.3.2.2. Preprocesamiento de los fragmentos 82
3.3.2.3. Etiquetado del texto contenido 83
3.3.2.4. Distribución del dataset de fragmentos de texto 84

3.4. EVALUACIÓN DE ARQUITECTURAS DE REDES NEURONALES 85

3.4.1. Comparativa de Modelos de Detección y Clasificación de Regiones 85
3.4.1.1. Selección de Modelos 85
3.4.1.2. Métricas de Evaluación 86
3.4.1.3. Condiciones de Entrenamiento 86
3.4.1.4. YOLO 87
3.4.1.5. FASTER RCNN 94
3.4.1.6. Análisis Comparativo de Resultados 101
3.4.2. Comparativa de Modelos para la Extracción de Texto 102
3.4.2.1. Selección de Modelos 102
3.4.2.2. Arquitecturas 103
3.4.2.3. Implementación de la Capa CTC 105
3.4.2.4. Condiciones de Entrenamiento 105
3.4.2.5. Evolución del Entrenamiento 106
3.4.2.6. Métricas de Evaluación 107
3.4.2.7. Análisis Comparativo de Resultados 108

3.5. DESARROLLO DEL SISTEMA 108

3.5.1. Arquitectura del Sistema 108
3.5.2. Diseño de la Base de Datos 109
3.5.3. Seguridad del Sistema 110
3.5.3.1. Autenticación y control de acceso 111
3.5.3.2. Protección de la base de datos 111
3.5.4. Iteración 1 111

3.5.4.1. Planificación 111
3.5.4.2. Diseño 112
3.5.4.3. Desarrollo 115
3.5.4.4. Pruebas 117
3.5.5. Iteración 2 121
3.5.5.1. Planificación 121
3.5.5.2. Diseño 122
3.5.5.3. Desarrollo 127
3.5.5.4. Pruebas 129
3.5.6. Iteración 3 133
3.5.6.1. Planificación 133
3.5.6.2. Diseño 134
3.5.6.3. Desarrollo 136
3.5.6.4. Pruebas 137
3.5.7. Iteración 4 139
3.5.7.1. Planificación 139
3.5.7.2. Diseño 140
3.5.7.3. Desarrollo 143
3.5.7.4. Pruebas 145
3.5.8. Iteración 5 147
3.5.8.1. Planificación 147
3.5.8.2. Diseño 148
3.5.8.3. Desarrollo 150
3.5.8.4. Pruebas 152
3.5.9. Iteración 6 154
3.5.9.1. Planificación 154
3.5.9.2. Diseño 155
3.5.9.3. Desarrollo 157
3.5.9.4. Pruebas 158

3.6. EVALUACIÓN DEL SISTEMA EN LA DOCUMENTACIÓN PERSONAL ... 160

3.6.1. Reducción de Demoras en el Ingreso de Datos 161
3.6.1.1. Propósito 161
3.6.1.2. Metodología 161
3.6.1.3. Resultados 161
3.6.2. Minimización de Inconsistencias en los Registros 164
3.6.2.1. Propósito 164
3.6.2.2. Metodología 164
3.6.2.3. Resultados 166

3.7. EVALUACIÓN TÉCNICA 168

3.7.1. Usabilidad 168
3.7.2. Funcionalidad 170
3.7.3. Mantenibilidad 172
3.7.4. Portabilidad 173

3.7.5. Resultados 174

3.8. EVALUACIÓN ECONÓMICA 174

3.8.1. Costos del Desarrollo de Software 174
3.8.1.1. Líneas de Código 175
3.8.1.2. Tipo de Proyecto 176
3.8.1.3. Factor de Ajuste de Esfuerzo 177
3.8.1.4. Esfuerzo 178
3.8.1.5. Tiempo 178
3.8.1.6. Costo de Software 178
3.8.2. Costos de Operación del Software 179
3.8.3. Costo Total del Proyecto 180

CAPÍTULO 4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES 181

4.2. RECOMENDACIONES 182

BIBLIOGRAFÍA

GLOSARIO DE TÉRMINOS

ANEXOS

ÍNDICE DE FIGURAS

Figura 1 *Sistema de información de una organización* 10
Figura 2 *Etapas de diseño* 15
Figura 3 *Etapas de pruebas* 16
Figura 4 *Las doce prácticas básicas de XP* 18
Figura 5 *Ciclo de vida de un proyecto según CRISP-DM* 21
Figura 6 *Funcionamiento del aprendizaje reforzado* 25
Figura 7 *Modelo de una neurona biológica* 30
Figura 8 *Comparación entre neuronas biológicas y artificiales* 31
Figura 9 *Red neuronal simple* 34
Figura 10 *Red neuronal profunda* 36
Figura 11 *Arquitectura de una red neuronal convolucional* 37
Figura 12 *Operación de convolución* 37
Figura 13 *Operaciones de pooling* 38
Figura 14 *Arquitectura YOLOv8* 39
Figura 15 *Arquitectura Faster-RCNN* 41
Figura 16 *Diagrama cíclico de una RNN* 42
Figura 17 *Ciclo de vida de una red neuronal* 44
Figura 18 *Diagrama de actividades del proceso de inscripción* 52
Figura 19 *Diagrama de actividades del proceso de registro de sacramentos* 53
Figura 20 *Diagrama de actividades del proceso de solicitud de certificados* 54
Figura 21 *Diagrama de actividades del proceso de registro de intenciones* 55
Figura 22 *Diagrama de actividades del flujo de gestión de información sacramental* 56
Figura 23 *Estructura fragmentada de tablas en archivos Access por sacramento* 57
Figura 24 *Formulario de registro de Confirmación en Access* 58
Figura 25 *Diagrama de Casos de Uso de Alto Nivel* 69
Figura 26 *Recolección de documentación* 71
Figura 27 *Flujo de generación de documentos sintéticos* 72
Figura 28 *Datos ficticios de cédulas de identidad* 73
Figura 29 *Función de Python para el llenado de plantillas* 74

Figura 30 *Función de Python para aplicar variaciones a documentos sintéticos*.....76
Figura 31 *Proceso de etiquetado de campos clave en certificados* 78
Figura 32 *Histograma de distribución de documentos reales y sintéticos* 79
Figura 33 *Estructura del dataset de documentos* 80
Figura 34 *Función en Python para recorte de fragmentos de texto* 81
Figura 35 *Fragmentos de texto extraídos de documentos* 81
Figura 36 *Función en Python para el ajuste de dimensiones*.....82
Figura 37 *Recortes de texto con padding y etiquetas asociadas* 83
Figura 38 *Archivo CSV con recortes de texto y "ground truth"*.....83
Figura 39 *Estructura del dataset para la extracción de texto* 84
Figura 40 *Salida de los bloques de la arquitectura YOLO*.....88
Figura 41 *Esquema del proceso de detección en la arquitectura YOLO* 89
Figura 42 *Evolución de la pérdida - YOLO certificados de nacimiento* 90
Figura 43 *Evolución del mAP- YOLO certificados de nacimiento* 90
Figura 44 *Evolución de la pérdida - YOLO cédulas de identidad*.....91
Figura 45 *Evolución de la pérdida - YOLO cédulas de identidad*.....91
Figura 46 *Matriz de confusión - YOLO certificados de nacimiento* 92
Figura 47 *Matriz de confusión - YOLO cédulas de identidad* 93
Figura 48 *Adaptación de arquitectura Faster R-CNN*.....94
Figura 49 *Salida de los bloques de la arquitectura Faster R-CNN*.....95
Figura 50 *Proceso de detección de la arquitectura Faster R-CNN* 96
Figura 51 *Evolución de la pérdida - Faster R-CNN certificados de nacimiento* 97
Figura 52 *Evolución del mAP - Faster R-CNN certificados de nacimiento*.....97
Figura 53 *Evolución de la pérdida - Faster R-CNN cédulas de identidad*.....98
Figura 54 *Evolución del mAP - Faster R-CNN cédulas de identidad* 98
Figura 55 *Matriz de confusión - Faster R-CNN certificados de nacimiento* 99
Figura 56 *Matriz de confusión - Faster R-CNN cedulas de identidad* 100
Figura 57 *Diagrama de red neuronal híbrida CNN-BiLSTM*.....104
Figura 58 *Diagrama de red neuronal híbrida CNN-BiGRU*.....104
Figura 59 *Implementación de la capa CTC para reconocimiento de texto*.....105
Figura 60 *Evolución de la pérdida - Modelo LSTM bidireccional* 106

Figura 61 *Evolución de la pérdida - Modelo GRU bidireccional* 107
Figura 62 *Arquitectura del sistema* 109
Figura 63 *Estructura conceptual de la base de datos del sistema* 110
Figura 64 *Diagrama de casos de uso - Gestión de usuarios* 113
Figura 65 *Diagrama relacional - Gestión de usuarios* 114
Figura 66 *Wireframe - Gestión de usuarios, roles y permisos* 115
Figura 67 *Pantalla de inicio de sesión del sistema* 116
Figura 68 *Pantalla de gestión de usuarios, roles y permisos* 117
Figura 69 *Diagrama de casos de uso - Gestión de feligreses* 123
Figura 70 *Diagrama relacional - Gestión de feligreses* 124
Figura 71 *Diagrama de secuencia - Procesamiento automático de documentos* 125
Figura 72 *Wireframe – Registro de feligreses* 126
Figura 73 *Función de procesamiento de documentos en FastAPI* 127
Figura 74 *Servicio de escaneo Flask-Laravel* 128
Figura 75 *Formulario de registro con datos extraídos automáticamente* 129
Figura 76 *Diagrama de casos de uso - Gestión de cursos sacramentales* 134
Figura 77 *Diagrama relacional- Gestión de cursos sacramentales* 135
Figura 78 *Wireframe – Cursos y sesiones* 136
Figura 79 *Pantalla de cursos y sesiones* 137
Figura 80 *Diagrama de casos de uso – Gestión de inscripciones* 141
Figura 81 *Diagrama relacional - Gestión de inscripciones* 142
Figura 82 *Wireframe – Inscripciones de un curso* 143
Figura 83 *Pantalla de formulario inscripciones* 144
Figura 84 *Pantalla de inscripciones de un curso* 145
Figura 85 *Diagrama de casos de uso – Gestión de celebraciones sacramentales* 149
Figura 86 *Diagrama relacional – Gestión de celebraciones sacramentales* 150
Figura 87 *Pantallas para la programación de celebraciones* 151
Figura 88 *Pantalla de gestión de celebraciones* 152
Figura 89 *Diagrama relacional – Gestión de registros sacramentales* 156
Figura 90 *Diagrama de casos de uso – Gestión de registros sacramentales* 157
Figura 91 *Pantallas para la impresión de certificados* 158

Figura 92 *Registros oficiales con inconsistencias* 165

ÍNDICE DE TABLAS

Tabla 1 *Planificación del proyecto según la metodología* 48
Tabla 2 *Documentación requerida por sacramento* 59
Tabla 3 *Campos de información relevantes por sacramento* 60
Tabla 4 *Requerimientos del usuario* 61
Tabla 5 *Historia de usuario - Acceso al sistema* 62
Tabla 6 *Historia de usuario - Gestión de roles y permisos* 62
Tabla 7 *Historia de usuario - Registro de feligreses* 63
Tabla 8 *Historia de usuario - Integración de redes neuronales* 63
Tabla 9 *Historia de usuario - Integración de servicio de escaneo* 64
Tabla 10 *Historia de usuario - Gestión de cursos sacramentales* 64
Tabla 11 *Historia de usuario - Programación de sesiones de cursos* 65
Tabla 12 *Historia de usuario - Inscripción a cursos* 65
Tabla 13 *Historia de usuario – Monitoreo de feligreses inscritos* 66
Tabla 14 *Historia de usuario - Creación de registros sacramentales* 66
Tabla 15 *Historia de usuario - Programación de celebraciones sacramentales* 67
Tabla 16 *Historia de usuario - Emisión de certificados sacramentales* 67
Tabla 17 *Historia de usuario - Monitoreo de celebraciones sacramentales* 68
Tabla 18 *Plan de Iteraciones* 68
Tabla 19 *Documentación escaneada* 71
Tabla 20 *Transformaciones aplicadas* 75
Tabla 21 *Etiquetas en cedulas de identidad* 76
Tabla 22 *Etiquetas en certificados de nacimiento* 77
Tabla 23 *Arquitecturas CNN seleccionadas para evaluación* 86
Tabla 24 *Métricas de evaluación en la detección de regiones y clases* 86
Tabla 25 *Métricas de evaluación en detección de texto con YOLOv8s* 94
Tabla 26 *Métricas de evaluación en detección de texto con Faster R-CNN* 101
Tabla 27 *Comparación de métricas YOLOv8s y Faster R-CNN* 102
Tabla 28 *Arquitecturas preseleccionadas para evaluación* 103

Tabla 29 *Métricas de evaluación para reconocimiento de texto* 107
Tabla 30 *Evaluación de modelos en reconocimiento de texto* 108
Tabla 31 *Plan de tareas - Iteración 1* 112
Tabla 32 *Caso de prueba – Inicio de sesión* 118
Tabla 33 *Caso de prueba – Crear usuario* 118
Tabla 34 *Caso de prueba – Inyección SQL en login* 119
Tabla 35 *Caso de prueba – Acceso sin autenticación* 120
Tabla 36 *Caso de prueba – Restricción de permisos por rol* 121
Tabla 37 *Plan de tareas - Iteración 2* 122
Tabla 38 *Caso de prueba – Registro manual de feligreses* 130
Tabla 39 *Caso de prueba – Escaneo de documentos* 131
Tabla 40 *Caso de prueba – Procesamiento de imagen con redes neuronales* 132
Tabla 41 *Plan de tareas - Iteración 3* 133
Tabla 42 *Caso de prueba – Creación de cursos sacramentales* 138
Tabla 43 *Caso de prueba – Creación de sesiones* 138
Tabla 44 *Plan de tareas - Iteración 4* 140
Tabla 45 *Caso de prueba – Inscripción de feligreses a un curso* 146
Tabla 46 *Caso de prueba – Registro de documentación* 146
Tabla 47 *Plan de tareas - Iteración 5* 148
Tabla 48 *Caso de prueba – Programación de celebraciones sacramentales* 153
Tabla 49 *Caso de prueba – Asociación de celebración a curso* 153
Tabla 50 *Plan de tareas - Iteración 6* 155
Tabla 51 *Caso de prueba – Generación de registro sacramental* 159
Tabla 52 *Caso de prueba – Impresión de certificado* 159
Tabla 53 *Tiempos de registro – Sacramento de bautismo* 162
Tabla 54 *Tiempos de registro – Sacramento de primera comunión* 162
Tabla 55 *Tiempos de registro – Sacramento de confirmación* 163
Tabla 56 *Tiempos de registro – Sacramento de matrimonio* 163
Tabla 57 *Reducción porcentual de demoras en el ingreso de datos* 164
Tabla 58 *Probabilidad de inconsistencias por registro* 166
Tabla 59 *Caso de prueba – Validación de campos obligatorios* 166

Tabla 60 *Caso de prueba – Modificación de datos de feligreses* 167
Tabla 61 *Encuesta de Usabilidad del Sistema* 169
Tabla 62 *Parámetros Punto Función* 170
Tabla 63 *Valores de Ajuste de Complejidad* 170
Tabla 64 *Ajuste de Complejidad del Sistema* 171
Tabla 65 *Portabilidad del Sistema* 173
Tabla 66 *Resultados de la Evaluación Técnica* 174
Tabla 67 *Líneas de Código por Punto Función* 175
Tabla 68 *Tipos de Proyectos en el Modelo COCOMO* 176
Tabla 69 *Coeficientes COCOMO* 176
Tabla 70 *Conductores de Costo en el Modelo COCOMO* 177
Tabla 71 *Costos de Operación del Software* 179
Tabla 72 *Costos Total del Proyecto* 180

ÍNDICE DE ECUACIONES

Ecuación 1: Ecuación del estímulo 31
Ecuación 2: Ecuación de la función de activación 32
Ecuación 3: Función de activación lineal 32
Ecuación 4: Función de activación sigmoide logística 33
Ecuación 5: Función de activación tangente hiperbólica 33
Ecuación 6: Punto de Función 171
Ecuación 7: Índice de Mantenibilidad 172
Ecuación 8: Ecuación KLDC por PF 175
Ecuación 9: Ecuación de Esfuerzo 178
Ecuación 10: Ecuación de Tiempo 178
Ecuación 11: Ecuación de Costo de Software 178

ABSTRACT

ABSTRACT

Information management requires attention and care, especially when dealing with physical and personal documents, as tasks like manually transcribing and reviewing data often lead to errors. When information handling is constant, relying on manual processes increases the workload and slows everything down. Therefore, it is necessary to have systems that can automatically extract information, reducing the time spent capturing and reviewing data, and allowing staff to focus on verifying its accuracy. Although human intervention is still required, automation helps speed up the work and makes records more reliable.

In the case of the San Miguel Arcángel Parish, manually transcribing data from identity cards and birth certificates was time-consuming and had led to inconsistencies in the records. To solve this problem, automatic information extraction was implemented using a sequential approach, in which areas of interest are first detected in each document, and then the text contained in those areas is extracted.

A YOLOv8s convolutional neural network was used to detect relevant areas. This network was retrained with a custom dataset adapted to classes representing the content within each area, composed of both real and synthetic documents. A hybrid network with LSTM cells was used for text extraction. This network was trained with a custom dataset created from text images of personal documents. This architecture allowed the network to learn character characteristics and predict the most probable sequence for accurate text retrieval.

The system was evaluated under real-world conditions, using authentic documentation and the hardware available at the parish. The results showed a 17.32% reduction in average administrative processing times. Furthermore, the workflow was simplified by eliminating repetitive transcriptions, which decreased the likelihood of inconsistencies in the records, making them more reliable and consistent.

RESUMEN

La gestión de información requiere atención y cuidado, especialmente cuando se trata de documentos físicos y personales, ya que tareas como transcribir y revisar datos de manera manual suelen generar errores. Cuando el manejo de información es constante, depender de procesos manuales aumenta la carga de trabajo y hace que todo sea más lento. Por eso, es necesario contar con sistemas que puedan extraer la información automáticamente, reduciendo el tiempo que se dedica a capturar y revisar los datos, y permitiendo que el personal se concentre en verificar que la información sea correcta. Aunque todavía se requiere la intervención de personas, la automatización ayuda a que el trabajo sea más rápido y los registros más confiables.

En el caso de la Parroquia San Miguel Arcángel, la transcripción manual de datos provenientes de cédulas de identidad y certificados de nacimiento requería mucho tiempo y había provocado inconsistencias en los registros. Para solucionar este problema, se implementó la extracción automática de información mediante un enfoque secuencial, en el que primero se detectan las áreas de interés en cada documento y, posteriormente, se extrae el texto contenido en esas zonas.

En la detección de áreas relevantes se utilizó una red neuronal convolucional YOLOv8s, reentrenada con un dataset propio adaptado a clases que representan el contenido dentro de cada área, compuesto por documentos reales y sintéticos. Para la extracción del texto se empleó una red híbrida con celdas LSTM, entrenada con un dataset propio creado a partir de imágenes de texto de los documentos personales. Esta arquitectura permitió que la red aprendiera las características de los caracteres y predijera la secuencia más probable para recuperar el texto de manera precisa.

El sistema se evaluó en condiciones reales, utilizando documentación auténtica y el hardware disponible en la parroquia. Los resultados mostraron una reducción del 17,32 % en los tiempos promedio de los procesos administrativos, además de que el flujo de trabajo se simplificó al eliminar transcripciones repetitivas, lo que disminuyó la probabilidad de inconsistencias en los registros, haciéndolos más confiables y consistentes.

CAPÍTULO 1.
GENERALIDADES

1.1. INTRODUCCIÓN

En la era digital, la eficiencia en la gestión de información es un factor crucial para el buen funcionamiento de las organizaciones, especialmente en aquellas que manejan grandes volúmenes de datos. Sin embargo, muchas instituciones continúan utilizando métodos manuales de transcripción y registro, lo que genera diversos inconvenientes como errores humanos, retrasos en el procesamiento de la información y una mayor carga administrativa. Este problema es particularmente evidente en entornos donde los registros y documentos son esenciales para la operación diaria, como en el caso de la Parroquia San Miguel Arcángel, donde el seguimiento de los participantes en los cursos sacramentales depende de la transcripción manual de datos provenientes de documentos físicos.

La digitalización de la información en estos procesos es una necesidad imperiosa para optimizar el manejo de registros y garantizar la precisión de los datos. Sin embargo, la automatización de este proceso requiere el uso de tecnologías avanzadas que no solo permitan extraer la información de manera precisa, sino que también ofrezcan una solución adecuada a las necesidades específicas de la parroquia.

Con la implementación de sistemas de reconocimiento de caracteres basados en redes neuronales, es posible transformar la manera en que se gestionan los datos de los cursos sacramentales. Estas tecnologías de inteligencia artificial ofrecen la capacidad de reconocer y extraer texto de documentos escaneados, reduciendo significativamente el riesgo de errores y mejorando la eficiencia en la actualización y acceso a la información.

La solución propuesta busca modernizar el proceso de transcripción de documentos, permitiendo que la información se procese de forma automática, rápida y precisa. De esta forma, se optimiza el uso de los recursos humanos y se facilita la actualización de registros sin depender de un proceso manual lento y propenso a errores. La implementación de esta tecnología en la Parroquia San Miguel Arcángel representa una oportunidad para mejorar la gestión de los datos relacionados con los participantes en los cursos sacramentales, brindando una herramienta confiable que permite a la parroquia ofrecer un mejor servicio a sus feligreses.

1.2. ANTECEDENTES

A continuación, se presentan los antecedentes institucionales y académicos que contextualizan este proyecto, destacando las bases teóricas y experiencias previas relevantes.

1.2.1. Antecedentes institucionales

La Parroquia San Miguel Arcángel, ubicada en la zona de Calacoto, tiene sus orígenes en la labor pastoral de la Congregación de la Resurrección. En 1959, los sacerdotes resurreccionistas, liderados por el padre Bill Burns, llegaron a la zona con el objetivo de evangelizar y servir a la comunidad. A través de su vocación, asumieron la responsabilidad de fundar la Parroquia de San Miguel Arcángel, consolidando un espacio de fe y encuentro para todos los feligreses.

Desde sus inicios, la parroquia no sólo se ha dedicado a la administración de los sacramentos, sino también a la educación espiritual de la comunidad. Los padres resurreccionistas participaron activamente en la dirección de instituciones educativas como el colegio Calvert y el colegio Saint Andrews, además de contribuir en la construcción y formación en la escuela Julio Patiño, el establecimiento fiscal más grande de la región. En el ámbito religioso, la preparación para los sacramentos de iniciación en los colegios fiscales comenzó en 1980 con la labor de la Hna. Ivonne Nosal, quien promovió la catequesis familiar.

Hoy en día, la parroquia sigue desempeñando un papel fundamental en la formación espiritual de la comunidad, organizando cursos sacramentales periódicos para quienes buscan recibir el bautismo, la primera comunión, la confirmación y el matrimonio. Además, existen diversos grupos formativos dentro de la parroquia, como Éxodo y Tras sus Huellas, que se encargan de la formación de jóvenes, niños y adultos en los distintos sacramentos. Otro grupo importante está compuesto por ministros, quienes apoyan activamente en todas las celebraciones litúrgicas.

Actualmente, la parroquia está bajo la dirección del Rvdo. P. Slawomir Siemieniec, CR., quien continúa liderando y guiando la comunidad en su labor pastoral y formativa.

1.2.1.1. Visión de la institución

Ser una comunidad evangelizadora y educativa que, inspirada en el Misterio Pascual, transforme vidas y sociedades a través de la fe, la esperanza y el amor. Aspiramos a fortalecer la formación espiritual de los fieles, especialmente jóvenes y familias, fomentando comunidades unidas en Cristo Resucitado. Nos comprometemos a ser un modelo de fraternidad cristiana que promueva la justicia, la verdad y la paz, extendiendo nuestra labor pastoral y educativa a todos, con especial énfasis en quienes enfrentan condiciones de vulnerabilidad.

1.2.1.2. Misión de la institución

La Congregación de la Resurrección se dedica a proclamar el Misterio Pascual mediante la evangelización y la educación, promoviendo la fe, la esperanza y el amor en la sociedad. Acompaña especialmente a jóvenes y familias, formando comunidades cristianas vivas y solidarias. A través de su labor pastoral y educativa, trabaja para superar la injusticia y fomentar la transformación espiritual de las personas y su entorno.

1.2.2. Antecedentes académicos

“RECONOCIMIENTO DE TEXTO MANUSCRITO CONTINUO EN LENGUA AYMARA” es una tesis de grado desarrollada por Ramiro Balmaceda Chuquimia en la Universidad Mayor de San Andrés en el año 2007. En este trabajo se implementó un sistema de reconocimiento automático de texto manuscrito en lengua aymara utilizando Modelos de Markov de Capa Oculta (HMMs). El estudio abordó la segmentación y análisis de características de escritura manuscrita, considerando atributos específicos del idioma aymara, como los caracteres h’ o q’, y la integración del reconocimiento óptico de caracteres (OCR) en el procesamiento de imágenes off-line.

En contraste con el proyecto mencionado, que se enfoca en el reconocimiento de texto manuscrito continuo en lengua aymara, considerando sus características lingüísticas mediante modelos estadísticos como los HMMs, el presente proyecto emplea redes neuronales para el reconocimiento inteligente de caracteres, permitiendo la extracción estructurada y automatizada de información relevante a partir de documentos impresos.

“USO DE DEEP LEARNING PARA LA DIGITALIZACION DE DOCUMENTOS HISTÓRICOS” trabajo de fin de grado desarrollado por Luis Enrique Güendián García en el año 2019 en la Universidad Carlos III de Madrid, donde se implementó la herramienta Ocropus, un OCR basado en redes neuronales LSTM, para la transcripción de documentos antiguos. El estudio abordó la optimización de hiperparámetros, la determinación del tamaño mínimo de datos de entrenamiento y la aplicación de Transfer Learning para mejorar la precisión en la digitalización de textos históricos impresos.

En contraste con el proyecto mencionado, el presente trabajo se enfoca en la extracción de información específica de certificados y documentos sacramentales en la Parroquia San Miguel Arcángel para automatizar el registro de cada sacramento. A diferencia de la digitalización

general de documentos históricos, este proyecto emplea técnicas avanzadas de procesamiento de imágenes y reconocimiento óptico de caracteres (OCR) para extraer solo los datos relevantes con mayor precisión y eficiencia.

“DISEÑO DE UNA APLICACIÓN DE RECONOCIMIENTO ÓPTICO DE CARACTERES MEDIANTE DEEP LEARNING” tesis desarrollada por Paula Miles Uribe, en el año 2020, en la Escuela Técnica Superior de Ingeniería y Sistemas de Telecomunicación, en la que se propone una herramienta personalizada de reconocimiento óptico de caracteres (OCR) que utiliza una red neuronal convolucional para mejorar el proceso de digitalización de documentos mecanografiados. El proyecto se enfoca en la segmentación de imágenes y la predicción de caracteres, con el fin de generar un texto editable con mínima corrección manual. Se utilizó MATLAB como plataforma de desarrollo, empleando herramientas como Image Processing Toolbox para el tratamiento de imágenes, Deep Learning Toolbox para el entrenamiento de la red neuronal y Report Generator Toolbox para la obtención del texto final en formato Word.

A diferencia del trabajo de Paula Miles Uribe, que utiliza MATLAB para la digitalización de textos mecanografiados, este proyecto empleará Python y técnicas avanzadas de procesamiento de imágenes e inteligencia artificial para identificar y extraer datos clave de documentos con estructuras variadas.

1.3. PLANTEAMIENTO DEL PROBLEMA

En la actualidad, la información necesita ser procesada de manera eficiente para garantizar su precisión y disponibilidad. Sin embargo, muchas organizaciones aún dependen de procesos manuales para la gestión de datos, lo que puede generar errores, retrasos y dificultades en la recuperación de la información. La digitalización y automatización de estos procesos se han convertido en una necesidad para optimizar el manejo de registros y mejorar la eficiencia en la administración de datos.

En el ámbito de la gestión de inscripciones y seguimiento de participantes en actividades formativas, la mayoría de las instituciones educativas, como colegios, universidades y centros de capacitación, han adoptado sistemas digitales que automatizan estos procesos. Sin embargo, aún existen organizaciones que continúan utilizando métodos manuales o herramientas básicas, lo que no sólo incrementa la carga administrativa, sino que también aumenta el riesgo de inconsistencias y pérdida de información, afectando la confiabilidad en la gestión de información.

1.3.1. Identificación del problema

Los cursos sacramentales en la Parroquia San Miguel Arcángel requieren un seguimiento continuo de los participantes, desde su inscripción hasta la emisión de certificados. La información necesaria se obtiene a partir de documentos físicos o copias, principalmente certificados de nacimiento y cédulas de identidad. Actualmente, esta información se registra mediante la transcripción manual en hojas de cálculo de Excel y bases de datos en Access.

Este proceso es realizado por una sola persona, lo que implica una carga administrativa considerable y aumenta el riesgo de errores en el proceso de transcripción. Además, la gestión y actualización de la información de los participantes requiere un esfuerzo adicional, ya que cualquier cambio en los datos, como nuevas inscripciones o modificaciones, debe reflejarse constantemente en los registros. Esto hace que el proceso sea más lento y susceptible a inconsistencias.

Si bien existen programas de digitalización en el mercado que podrían facilitar la extracción de información de los certificados o documentos, muchos de ellos no se ajustan a las necesidades específicas de la parroquia. Algunos requieren configuraciones complejas, mientras que otros dependen de procesos semiautomatizados, como copiar y pegar datos, lo que también es susceptible a errores en la selección de información relevante para el seguimiento de los participantes en los cursos sacramentales. En otras palabras, no se integran completamente con los requerimientos del proceso.

Todo este proceso manual para el seguimiento a los cursos sacramentales no sólo ralentiza las operaciones, sino que también dificulta el acceso rápido y preciso a la información de los participantes. La constante actualización de registros y la transcripción manual de datos son propensas a errores, lo que puede afectar directamente la planificación, coordinación y eficiencia de las actividades formativas.

1.3.2. Formulación del problema

¿Cómo mejorar la eficiencia del proceso de registro y reducir las inconsistencias en la información de los feligreses, considerando que la información se obtiene de documentos de identidad y actualmente se ingresa mediante procedimientos manuales en la Parroquia San Miguel Arcángel?

1.4. OBJETIVOS

1.4.1. Objetivo general

Desarrollar un sistema inteligente de reconocimiento óptico de caracteres basado en redes neuronales para la detección y extracción de información en documentos de identidad, a fin de mejorar la eficiencia del proceso de registro y reducir inconsistencias en la información de los feligreses de la Parroquia San Miguel Arcángel.

1.4.2. Objetivos específicos

- Analizar los procesos actuales de registro de información personal a partir de documentos de identidad, con la finalidad de identificar los requerimientos del sistema.

- Construir datasets a partir de imágenes digitalizadas de documentos de identidad, con la finalidad de entrenar redes neuronales orientadas a la detección y extracción de información.

- Evaluar diferentes arquitecturas de redes neuronales en función de su desempeño durante el entrenamiento, para seleccionar las más adecuadas en la detección y extracción de información en documentos de identidad.

- Desarrollar el sistema integrando los modelos de redes neuronales, para permitir la detección y extracción automática de información en documentos de identidad.

- Validar el sistema de detección y extracción de información en documentos de identidad, para garantizar su correcto desempeño en escenarios reales.

1.5. JUSTIFICACIONES

1.5.1. Justificación técnica

El proyecto integra tecnologías avanzadas como redes neuronales y Reconocimiento Óptico de Caracteres, con el propósito de automatizar el registro y la extracción de información contenida en documentos. Se emplearán herramientas como GIMP para la creación de documentos base, la anotación de datos será con la herramienta LabelImg y el procesamiento de imágenes será a través de OpenCV.

En cuanto al desarrollo se utilizará el lenguaje de programación Python, dentro de entornos seguros y controlados mediante Anaconda, utilizando Jupyter Notebook como plataforma interactiva para codificación y experimentación. Así mismo, se emplearán bibliotecas como TensorFlow y Keras para el diseño y entrenamiento de modelos, así como el Framework Ultralytics para implementar modelos de aprendizaje profundo. Además, se utilizará TensorBoard para monitorear el comportamiento y las métricas de los entrenamientos.

1.5.2. Justificación económica

La automatización de procesos reducirá la dependencia de métodos manuales, optimizando los recursos humanos y materiales. Esto disminuirá los costos relacionados con errores y correcciones, ahorrando tiempo en tareas repetitivas. Además, permitirá una mejor distribución de recursos, ya que el personal administrativo podrá enfocarse en actividades de mayor valor. Esto incrementará la eficiencia operativa, mejorando el funcionamiento general de la parroquia. A largo plazo, facilitará una gestión más efectiva y sostenible.

1.5.3. Justificación social

La propuesta se justifica socialmente al ofrecer un servicio más confiable y ágil para los feligreses de la Parroquia San Miguel Arcángel. La automatización del proceso mejorará la fiabilidad de los registros, asegurando que la información personal sea precisa y libre de errores. Esto facilitará la emisión de documentos válidos, como certificados de los sacramentos recibidos, cuando los feligreses los necesiten como respaldo, reduciendo así la probabilidad de errores y proporcionando un respaldo efectivo.

1.6. ALCANCES Y LÍMITES

En todo proyecto, es fundamental definir claramente sus límites y alcances, ya que estos determinan el enfoque, los recursos necesarios y los resultados esperados. El establecimiento de límites permite enfocar los esfuerzos en áreas específicas, mientras que los alcances definen lo que se logrará dentro del marco del proyecto. Por lo tanto, el presente proyecto está sujeto a los siguientes límites y alcances, los cuales guiarán su desarrollo y asegurarán el cumplimiento de los objetivos establecidos.

1.6.1. Alcances

1.6.1.1. Alcance temático

El alcance temático de este proyecto abarcará los siguientes temas.

Las líneas de investigación incluyen inteligencia artificial, redes neuronales, procesamiento de imágenes, reconocimiento óptico de caracteres (OCR), digitalización y automatización de datos.

En cuanto a las áreas temáticas, se abordará el desarrollo y entrenamiento de modelos de redes neuronales para el reconocimiento de caracteres, la comparación de diferentes arquitecturas para determinar las más precisas y el desarrollo de un sistema de gestión de registros parroquiales.

1.6.1.2. Alcance geográfico

El desarrollo del proyecto se llevará a cabo en la Parroquia San Miguel Arcángel, ubicada en la zona sur de la ciudad de La Paz, específicamente en Calacoto, Avenida Ballivián, Plaza San Miguel, calle 21. Allí se trabajará con documentos físicos utilizados en el seguimiento de los cursos sacramentales. Para ello, se establecerá un ambiente de pruebas en el cual se digitalizarán los registros y se evaluará la eficacia del sistema propuesto. Se empleará un equipo anfitrión con sistema operativo Windows, en el cual se ejecutará el software desarrollado. La prueba y validación del sistema se realizarán con documentos reales proporcionados por la parroquia, asegurando que el modelo sea aplicable a las necesidades específicas del entorno.

1.6.1.3. Alcance temporal

El proyecto se desarrollará en un máximo de ocho meses, abarcando análisis, desarrollo y pruebas. Se respetarán los plazos y normativas universitarias para la entrega y defensa del proyecto.

1.6.2. Límites

- La red neuronal será entrenada para reconocer únicamente documentos en idioma español, sin considerar otros idiomas.

- Las redes neuronales requerirán imágenes claras y de alta resolución para su procesamiento.

- El sistema estará orientado exclusivamente al reconocimiento de caracteres impresos, excluyendo la escritura manuscrita.
- El sistema considerará exclusivamente documentación personal emitida en Bolivia, concretamente certificados de nacimiento y cédulas de identidad.

CAPÍTULO 2.
MARCO TEÓRICO

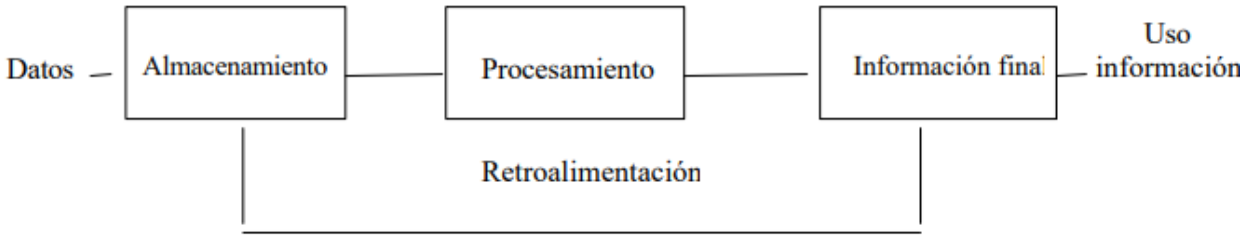
2.1. SISTEMAS DE INFORMACIÓN

Andreu, Ricart, & Valor (1991) citado por Hernández Trasobares (2003, p. 1), define a un sistema de infromación como un conjunto estructurado de procesos que operan sobre datos organizados según las necesidades de una empresa. Su función principal es recopilar, procesar y distribuir información de manera selectiva, con el propósito de facilitar la operación empresarial, el control organizacional y la toma de decisiones estratégicas.

Desde este punto de vista, los sistemas de información no son sólo herramientas que ayudan en la gestión empresarial; también son fundamentales para optimizar procesos y mejorar la eficiencia dentro de las organizaciones. Su habilidad para integrar datos relevantes y presentarlos de forma clara permite a las empresas adaptarse a entornos cambiantes y tomar decisiones informadas que fomenten su crecimiento.

Figura 1

Sistema de información de una organización



Nota: El gráfico muestra los componentes de un sistema de información, donde los datos son procesados y transformados en información útil, con un mecanismo de retroalimentación para evaluar su efectividad. Tomado de "Los sistemas de información: evolución y desarrollo" (p. 1), por A. Hernández Trasobares, 2003, Proyecto Social.

2.1.1. Propiedades de los sistemas de información

Según Beynon-Davies (2018), el diseño de los sistemas de información debe considerar características clave antes de su implementación, ya que estas determinan su éxito. Las tres características esenciales que un sistema de información debe tener son:

- Funcionalidad: Se refiere a lo que el sistema es capaz de hacer, según los requisitos de la organización.

- Usabilidad: Es la facilidad con la que los usuarios pueden interactuar con el sistema para lograr el propósito para el cual fue creado. Esto se refleja principalmente en la interfaz hombre-máquina.

- Utilidad: Mide el valor del sistema, evaluando si cumple efectivamente con lo que es necesario para la organización y la actividad humana que apoya.

El equilibrio entre estas tres características es fundamental para asegurar que un sistema de información tenga éxito. Mientras que la funcionalidad establece las tareas que el sistema debe llevar a cabo, la usabilidad se encarga de que sea fácil de usar, y la utilidad verifica su efectividad en el apoyo a las actividades de la organización. Por eso, es vital considerar estos aspectos desde la fase de diseño hasta la implementación, para crear un sistema que no sólo funcione bien, sino que también sea accesible y ofrezca un valor real a la organización.

2.1.2. Clasificación de los sistemas de información

Laudon & Laudon (1996) citado por Hernández Trasobares (2003, p. 3), la clasificación más útil de los sistemas de información se basa en su utilidad dentro de los diferentes niveles organizacionales. La estructura empresarial se divide en cuatro niveles: el nivel operativo, enfocado en las operaciones diarias de la organización; el nivel del conocimiento, relacionado con los empleados que manejan la información, generalmente el departamento de informática; el nivel administrativo, que abarca a los gerentes intermedios encargados de la gestión y control de la organización; y el nivel estratégico, correspondiente a la alta dirección, responsable de la toma de decisiones estratégicas.

- Sistema de Procesamiento de Operaciones (SPO): Se trata de sistemas informáticos encargados de gestionar las operaciones diarias y rutinarias de la empresa, tales como aplicaciones de nóminas, seguimiento de pedidos, auditoría, registro y administración de datos de empleados. Su función principal es generar información esencial que será utilizada por otros sistemas de información dentro de la organización. Estos sistemas son empleados principalmente por el personal del nivel operativo, ya que facilitan el control y ejecución de tareas repetitivas necesarias para el funcionamiento cotidiano de la empresa.

- Sistemas de Trabajo del Conocimiento (STC): Son aquellos sistemas diseñados para apoyar a los empleados que manejan información en la creación e integración de nuevos conocimientos dentro de la empresa. Suelen incluir estaciones de trabajo especializadas que

facilitan el procesamiento y análisis de información crítica para la toma de decisiones estratégicas. Estos sistemas forman parte del nivel de conocimiento y son fundamentales para optimizar los flujos de información en la organización.

- Sistemas de Automatización en la Oficina (SAO): Comprenden herramientas informáticas destinadas a mejorar la productividad de los empleados que trabajan con información en los niveles inferiores de la empresa. Algunos ejemplos de estos sistemas incluyen procesadores de texto, hojas de cálculo, correo y agendas electrónicas. Al igual que los Sistemas de Trabajo del Conocimiento, los SAO pertenecen al nivel de conocimiento, ya que facilitan la gestión y transmisión de información dentro de la organización.

- Sistemas de Información para la Administración (SIA): Son sistemas diseñados para asistir a la gerencia y a los directivos en los niveles intermedios de la organización. Su propósito es apoyar procesos de planificación, control y toma de decisiones mediante la generación de informes sobre las actividades ordinarias del negocio. Estos sistemas pueden ser utilizados en áreas como control de inventarios, presupuestación anual y análisis de decisiones de inversión y financiación.

- Sistemas para el Soporte de Decisiones (SSD): Son herramientas interactivas que ayudan a los usuarios en el proceso de toma de decisiones al permitir el análisis de datos y la utilización de modelos matemáticos para resolver problemas no estructurados. Estos sistemas son empleados principalmente por la gerencia intermedia y permiten realizar análisis como estudios de costos, evaluación de precios y beneficios, así como segmentación de ventas por ubicación geográfica.

- Sistemas de Soporte Gerencial (SSG): Se encuentran en el nivel estratégico de la organización y están diseñados para facilitar la toma de decisiones de alto nivel mediante el uso de gráficos avanzados y sistemas de comunicación. Son empleados por la alta dirección para definir estrategias generales de la empresa, tales como la planificación de ventas a largo plazo, la gestión de operaciones y la planificación de recursos humanos.

Esta clasificación nos ayuda a entender cómo los sistemas de información se integran en la estructura de una organización y facilitan la gestión empresarial en diferentes niveles. Los sistemas operativos garantizan que las actividades diarias se desarrollen sin problemas, mientras que los sistemas enfocados en el conocimiento mejoran la forma en que se maneja y procesa la

información dentro de la empresa. Además, los sistemas administrativos ofrecen apoyo en la planificación y el control de operaciones, lo que permite una gestión más eficiente. En los niveles más altos, los sistemas de soporte para la toma de decisiones y gerenciales proporcionan herramientas analíticas y estratégicas que refuerzan la toma de decisiones y la planificación a largo plazo.

2.2. INGENIERÍA DE SOFTWARE

La ingeniería de software se ocupa de manejar de manera organizada todo el proceso de desarrollo y gestión del software, comenzando desde la planificación inicial hasta el mantenimiento una vez que ya está en funcionamiento. (Sommerville, 2011)

Por lo tanto, la ingeniería de software no se limita solo a escribir código. Es un enfoque metódico que abarca todas las etapas que atraviesa un sistema, desde la idea inicial hasta su mantenimiento. Su objetivo es asegurarse de que cada una de estas fases se realice de manera planificada y evaluable, para garantizar que el producto final sea de alta calidad y funcione de manera confiable.

2.2.1. Gestión de proyectos de software

La gestión de proyectos es una disciplina que se extiende a lo largo de diferentes tipos de proyectos, ya sea en el mundo de las tecnologías de la información y la comunicación (TIC) o en campos como la construcción, la ingeniería y la gestión empresarial, por mencionar algunos. Esta área abarca todos los procesos necesarios para dirigir, administrar y gestionar cualquier proyecto, sin importar el tipo de producto que se esté creando. Su principal objetivo es asegurar que el proyecto tenga éxito. (Pradel & Raya, 2013)

En el mundo del desarrollo de software, la gestión de proyectos necesita ir de la mano con métodos, técnicas y herramientas que son específicas para este tipo de proyecto. Aunque hay quienes piensan que la gestión de proyectos no está directamente relacionada con la ingeniería del software, la realidad es que es una parte fundamental de cualquier proyecto de desarrollo. Algunas de las tareas clave que se asocian con la gestión de proyectos en este campo son:

- Estudio de viabilidad: antes de empezar, es fundamental identificar las necesidades de la organización y evaluar si este proyecto es la mejor alternativa, teniendo en cuenta los costos y el tiempo que se requerirá.

- Estimación: es importante calcular tanto el costo como el tiempo del proyecto, y comprender cómo los recursos impactan en el tiempo de desarrollo, ya que esta relación no siempre es lineal.

- Definir objetivos claros: los objetivos del proyecto deben estar bien definidos, ya que su cumplimiento o incumplimiento será lo que determine si el proyecto tiene éxito o fracasa.

- Formar el equipo de desarrollo: el equipo debe ser constituido en función de los recursos estimados, incluyendo no solo a los desarrolladores, sino también a representantes de los interesados.

- Establecer hitos: es importante definir puntos clave para hacer un seguimiento y garantizar que el proyecto avance adecuadamente.

- Identificar riesgos: es esencial reconocer posibles problemas que puedan poner en peligro el éxito del proyecto, ya sean de naturaleza tecnológica, organizativa o legal.

2.2.2. Diseño y desarrollo de software

El diseño de software se puede entender como una representación de cómo se estructurará el sistema que se va a implementar. Esto incluye aspectos como los modelos, las estructuras de datos, las interfaces entre los diferentes componentes y, en ocasiones, los algoritmos que se utilizarán. En lugar de saltar directamente a una versión final, los diseñadores suelen trabajar de manera iterativa. Esto significa que van añadiendo más formalidad y detalles a medida que avanzan, haciendo ajustes y correcciones a los diseños anteriores en el camino. (Sommerville, 2011)

Bajo la anterior definición, podemos dividir el diseño de software en cuatro etapas:

- Diseño arquitectónico: en esta fase, se establece la estructura general del sistema, identificando los componentes clave (subsistemas o módulos), sus interrelaciones y cómo se organizan entre sí.

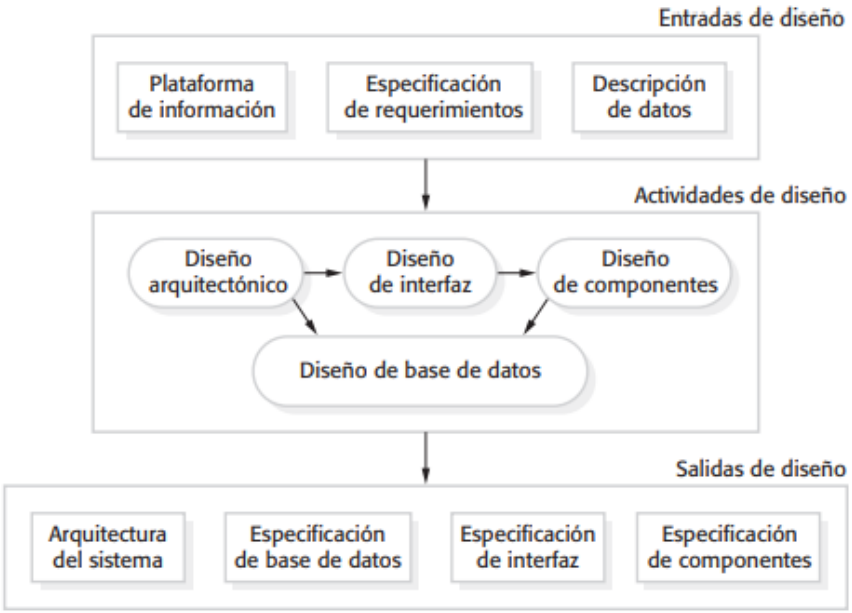
- Diseño de interfaz: aquí se detallan las interfaces que conectan los diferentes componentes del sistema. Una vez que se acuerdan las especificaciones de la interfaz, los componentes se diseñan y desarrollan de manera simultánea.

- Diseño de componentes: durante esta etapa, se diseña cada componente del sistema para definir su funcionamiento. Esto puede incluir una descripción general de la funcionalidad esperada, dejando los detalles del diseño en manos del programador.

- Diseño de base de datos: En esta fase, se crean las estructuras de datos y se determina cómo se representarán las mismas en una base de datos. El trabajo en esta etapa depende de si se va a reutilizar una base de datos existente o si será necesario desarrollar una nueva.

Figura 2

Etapas de diseño



Nota: El diagrama ilustra que las etapas del diseño siguen un orden secuencial, pero siempre hay un flujo de retroalimentación entre ellas, lo que resulta en una continua revisión y mejora del diseño. Tomado de “Ingeniería de Software” (p. 39), por I. Sommerville, 2011.

El desarrollo de software y el diseño están íntimamente relacionados, ya que el desarrollo depende de las decisiones que se toman durante la fase de diseño. La arquitectura del diseño establece la estructura del sistema, lo que facilita su desarrollo posterior. En el diseño de interfaces, se especifican las conexiones entre los diferentes componentes, que luego se convierten en código durante el desarrollo. De esta forma, el diseño actúa como un marco fundamental para el desarrollo del sistema.

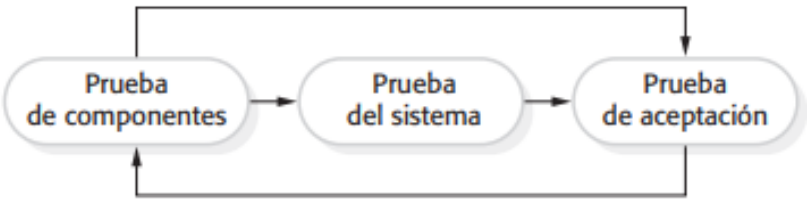
2.2.3. Validación de software

Según Sommerville (2011), la verificación y validación del software tienen como objetivo garantizar que el sistema que se desarrolla no sólo cumpla con los requisitos establecidos, sino que también satisfaga las necesidades del cliente. Este proceso, que abarca pruebas con datos simulados, inspecciones y revisiones en diferentes etapas del ciclo de vida, es iterativo y puede requerir regresar a fases anteriores si se encuentran errores en los componentes o interfaces del sistema.

Por lo que, la validación del software no se trata de un proceso aislado, sino que se realiza en cada fase del desarrollo, desde la recolección de requisitos hasta las pruebas finales con datos reales de los usuarios. Como se observa en la Figura 3, este enfoque gradual ayuda a identificar errores de forma progresiva: primero en los componentes individuales, luego en el sistema integrado, y finalmente en la solución completa.

Figura 3

Etapas de pruebas



Nota: El gráfico muestra el flujo de pruebas en el desarrollo de software: prueba de componentes, prueba del sistema y prueba de aceptación. El proceso es iterativo, volviendo a la prueba de componentes si se encuentran errores en etapas posteriores. Tomado de “Ingeniería de Software” (p. 41), por I. Sommerville, 2011.

2.2.3.1. Pruebas de componentes

Son un proceso clave donde los desarrolladores evalúan los distintos componentes que forman el sistema. Cada uno de estos componentes se prueba de forma independiente, lo que significa que se analizan sin la influencia de otros elementos del sistema. Estos componentes pueden ser entidades simples, como funciones o clases de objetos, o incluso grupos coherentes de estas entidades. Normalmente, se utilizan herramientas de automatización de pruebas que permiten repetir las pruebas de los componentes cada vez que se lanzan nuevas versiones.

2.2.3.2. Pruebas del sistema

Los componentes del sistema se unen para formar un todo. Durante este proceso, el objetivo es identificar posibles fallos que puedan surgir de interacciones inesperadas entre los diferentes elementos, así como cualquier problema relacionado con las interfaces. También se asegura que el sistema cumpla con todos los requisitos establecidos, tanto los funcionales como los no funcionales, y se examinan las propiedades emergentes que surgen de su conjunto. En sistemas de gran escala, esta integración generalmente se lleva a cabo en etapas, agrupando componentes en subsistemas que se evalúan por separado antes de integrarse en el sistema final.

2.2.3.3. Pruebas de aceptación

Las pruebas de aceptación son la etapa final en el proceso de evaluación de un sistema antes de que se implemente oficialmente. Durante esta fase, el software se pone a prueba con datos reales que proporciona el cliente, en lugar de usar datos simulados. Este enfoque ayuda a detectar errores o deficiencias en la definición de los requisitos, ya que los datos reales exponen al sistema a condiciones diferentes. Además, estas pruebas pueden revelar que algunas funciones no cumplen con las expectativas del usuario o que el rendimiento del sistema no es el que se esperaba.

2.2.4. Metodologías

2.2.4.1. Programación extrema (XP)

La Programación Extrema (XP) es una metodología ágil centrada en la entrega rápida y continua de software funcional, mediante iteraciones cortas y colaborativas. Los métodos ágiles se basan en un enfoque progresivo para la especificación, desarrollo y entrega de software. Son ideales para proyectos donde los requisitos pueden cambiar rápidamente a medida que avanza el desarrollo. Su propósito es entregar software funcional a los clientes de manera rápida, permitiéndoles sugerir nuevos requerimientos para las siguientes iteraciones. (Sommerville, 2011)

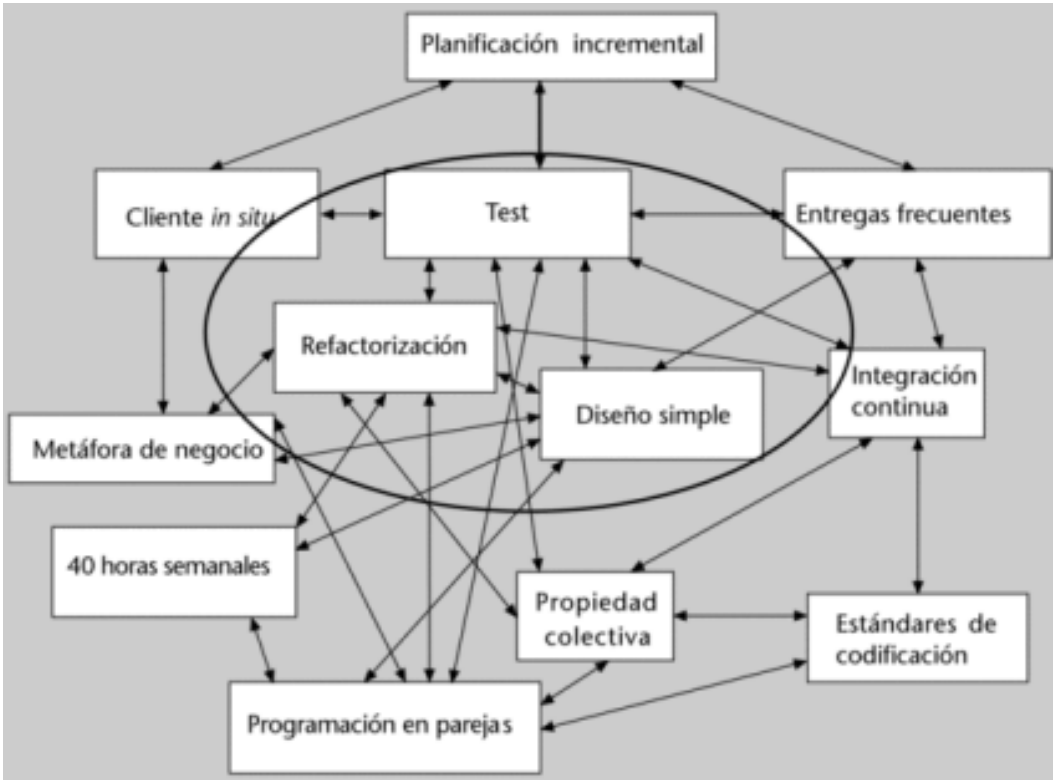
En XP, los requisitos se presentan a través de historias de usuario, las cuales se transforman en tareas concretas a desarrollar. Los programadores suelen trabajar en parejas y, antes de lanzarse a escribir el código, se dedican a crear pruebas para cada tarea. Es fundamental que

estas pruebas sean satisfactorias antes de que el nuevo código se integre al sistema. (Sommerville, 2011)

Kent Beck, Ward Cunningham y Ron Jeffries, creadores de la Programación Extrema eligieron las prácticas más efectivas de sus proyectos para sentar las bases de XP. Así, surgieron doce prácticas que se complementan entre sí, con el objetivo de mejorar la calidad del código, fomentar la colaboración en equipo y aumentar la satisfacción del cliente.

Figura 4

Las doce prácticas básicas de XP



Nota: El gráfico muestra las prácticas interconectadas de la Programación Extrema (XP). Tomado de "Introducción a las metodologías ágiles" (p. 24), por Fernández González.

a) Diseño simple

El diseño debería ser lo más simple posible para que el código sea fácil de entender. La idea principal es optar por el diseño más sencillo que funcione, deshaciéndose del código innecesario y asegurándose de que sea claro y fácil de mantener.

b) Refactorización

Refactorizar es un proceso que implica mejorar el código de manera continua, manteniéndolo limpio y sencillo, incluso tras hacer cambios. Esto ayuda a evitar que el código se convierta en un caos con el tiempo, asegurando que siga siendo funcional y eficiente.

c) Test

Las pruebas son esenciales en XP y deben redactarse antes de escribir el código. Esto permite prever posibles inconvenientes y garantiza que el código funcione de manera adecuada.

d) Estándares de codificación

El equipo necesita adherirse a normas de codificación comunes para que el código sea más fácil de entender y modificar. Esto abarca convenciones de nombres y estilos que hacen que el código sea más claro y consistente.

e) Propiedad colectiva del código

En XP, el código no pertenece a un solo programador. Todo el equipo comparte la responsabilidad de modificar y mejorar el código, asegurándose de que se mantenga claro, eficiente y fácil de entender, sin importar quién lo escribió al principio.

f) Programación por parejas

En vez de que cada programador trabaje por su cuenta, dos de ellos se unen en una sola estación de trabajo. Esto no solo eleva la calidad del código, sino que también minimiza los errores, ya que uno se encarga de la estrategia mientras el otro se enfoca en los detalles más finos.

g) Integración continua

Las funcionalidades se integran al sistema en cuanto se desarrollan, lo que permite hacer pruebas de integración con frecuencia. Esto ayuda a identificar errores rápidamente y asegura que todos estén trabajando con la versión más actual del código.

h) 40 horas semanales

XP fomenta que los equipos trabajen de forma eficiente sin sobrepasar las 40 horas a la semana. El exceso de trabajo puede afectar tanto la productividad como la moral del equipo, por lo que se busca lograr un equilibrio que evite el agotamiento.

i) Metáfora del negocio

El equipo de desarrollo y el cliente final necesitan tener una "metáfora" compartida que capture la esencia del negocio y cómo debería funcionar el sistema. Esto no sólo ayuda a que ambos se entiendan mejor, sino que también garantiza que tengan una visión clara de los objetivos del proyecto.

j) Cliente in situ

El cliente necesita ser parte del equipo de desarrollo y estar presente en persona durante todo el proceso. Esto hace que la comunicación sea más fluida y permite que el equipo obtenga retroalimentación al instante sobre el sistema, lo que a su vez mejora la calidad del producto final.

k) Entregas frecuentes

Es fundamental lanzar versiones pequeñas del sistema lo más pronto posible. Aunque no estén completamente terminadas, deben ofrecer una idea clara de cómo será el producto final y ayudar al usuario a familiarizarse con él. Cada nueva versión debe aportar un valor real al negocio, garantizando que el cliente obtenga lo que realmente necesita.

l) Planificación incremental

La planificación en XP es bastante adaptable y se ajusta a medida que las necesidades del negocio evolucionan. En cada ciclo de replanificación, se revisan las prioridades para asegurarse de que se esté enfocando en lo que realmente aporta más valor.

2.2.4.2. CRISP DM

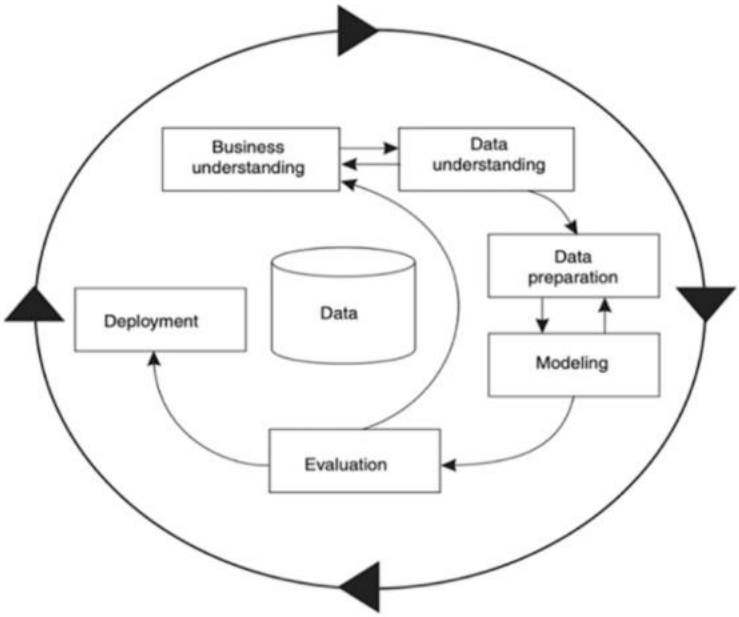
En respuesta a las necesidades comunes en los proyectos de minería de datos durante los años noventa, un grupo de organizaciones como Teradata, SPSS-ISL, Daimler-Chrysler y OHRA propuso una metodología estandarizada llamada CRISP-DM (Cross Industry Standard Process for Data Mining). Esta guía, considerada hoy en día como el estándar de facto en la minería de

datos, ofrece una visión general del ciclo de vida de estos proyectos, incluyendo las fases principales, sus tareas asociadas y las relaciones entre ellas. (Mora, Wang, Marx Gomez, & Duran-Limon, 2023)

El ciclo de vida de CRISP-DM, mostrado en la Figura 5, se compone de seis fases que se recorren de forma iterativa, permitiendo hacer ajustes y mejoras según se avanza. Esta estructura flexible ayuda a adaptarse a los cambios y a seguir aprendiendo durante el desarrollo.

Figura 5

Ciclo de vida de un proyecto según CRISP-DM



Nota: El gráfico muestra las fases iterativas del modelo para los proyectos de ciencia de datos, CRISP-DM, que abarca desde la comprensión del negocio hasta el despliegue. Tomado de "Development Methodologies for Big Data Analytics Systems: Plan-driven, Agile, Hybrid, Lightweight Approaches " (p. 138), por Mora, Wang, Marx Gomez, & Duran-Limon, 2023.

A continuación, se describen brevemente las seis fases que conforman el ciclo de vida de CRISP-DM, siguiendo la explicación presentada por [Apellido(s) del autor(es)] (Año) en su obra Título del libro.

a) Comprensión del negocio

Se identifican los objetivos principales del proyecto desde el punto de vista organizacional, lo que permite definir con claridad el problema a resolver y establecer un plan inicial de trabajo.

b) Comprensión de los datos

Se recopila la información disponible y se realiza un análisis exploratorio para conocer su estructura, detectar posibles errores y empezar a formular hipótesis útiles para el desarrollo.

c) Preparación de los datos

Consiste en transformar los datos crudos en un conjunto adecuado para el modelado, lo que implica limpiar, seleccionar, codificar y reorganizar la información según las necesidades del proyecto.

d) Modelado

Se aplican técnicas analíticas o de aprendizaje automático para construir un modelo que aborde el problema planteado. En esta fase también se ajustan parámetros para mejorar el rendimiento del modelo.

e) Evaluación

Se analiza si el modelo cumple con los objetivos definidos, verificando su efectividad y revisando todo el proceso previo para identificar posibles mejoras o aspectos no contemplados.

f) Implementación

Finalmente, los resultados se integran en un entorno real. Esta fase puede implicar desde la entrega de informes hasta la puesta en marcha de sistemas automatizados, asegurando que el resultado sea útil y aplicable.

2.3. INTELIGENCIA ARTIFICIAL

Según Pérez Orozco (2018), podemos definir a la inteligencia artificial (IA) como la rama de las ciencias computacionales que se encarga del diseño y construcción de sistemas capaces de realizar tareas asociadas con la inteligencia humana. Es decir, la inteligencia artificial no sólo

replica habilidades humanas, sino que también mejora y automatiza tareas antes exclusivas de los humanos.

La inteligencia artificial (IA) está teniendo un rápido crecimiento en diversas áreas de aplicación, incluyendo el reconocimiento de imágenes, la mejora de estrategias comerciales en finanzas, el procesamiento de datos médicos, el mantenimiento predictivo en la industria, la detección de objetos en vehículos autónomos, la distribución de contenido en redes sociales y la protección contra amenazas de ciberseguridad en sistemas bancarios. (Rouhiainen, 2018)

Las aplicaciones de la inteligencia artificial no solo están revolucionando sectores industriales, sino que también están generando cambios profundos en la forma en que operan diversas industrias. Desde el reconocimiento de imágenes hasta la protección contra amenazas cibernéticas, la IA está permitiendo un nivel de automatización y precisión que antes no era posible. Su implementación en áreas como la salud y las finanzas muestra cómo la IA puede mejorar la eficiencia operativa, permitiendo decisiones más rápidas y precisas, mientras optimiza recursos y reduce costos.

2.3.1. Machine Learning

El aprendizaje automático (machine learning) es un campo que permite a las computadoras "aprender" a partir de los datos. A diferencia de la programación convencional, que requiere la codificación paso a paso de soluciones específicas, el aprendizaje automático se enfoca en crear algoritmos generales capaces de identificar patrones en diversos tipos de datos. (Bobadilla, 2020)

El enfoque del aprendizaje automático representa una evolución significativa en la programación, ya que permite a las máquinas adaptarse y mejorar de manera autónoma a medida que procesan más información. Esto elimina la necesidad de programación manual para cada situación, lo que lo hace especialmente útil en campos donde los datos son grandes y complejos, como en la predicción de tendencias de mercado o la clasificación de imágenes.

2.3.1.1. Aprendizaje supervisado

El aprendizaje supervisado en machine learning se utiliza cuando cada dato o conjunto de datos de entrada está asociado a una etiqueta específica. A partir de estos datos etiquetados, se aplican algoritmos de clasificación para entrenar un modelo que, una vez completado el proceso

de entrenamiento, pueda predecir la etiqueta de nuevos datos que no estaban presentes en el conjunto de datos original. (Bobadilla, 2020)

En otras palabras, el aprendizaje supervisado no solo permite la predicción de etiquetas de nuevos datos, sino que también ofrece la posibilidad de generalizar patrones aprendidos a partir de ejemplos previos.

Un ejemplo práctico de aprendizaje supervisado se puede observar en el reconocimiento de frutas mediante imágenes. Supongamos que se entrena un modelo de aprendizaje supervisado con un conjunto de datos que contiene imágenes etiquetadas de diferentes frutas, como manzanas, plátanos y naranjas. Tras el proceso de entrenamiento, el modelo aprende a identificar características clave de cada fruta, como su forma, color y textura. Posteriormente, cuando se le presenta una nueva imagen de una fruta que no ha visto antes, el modelo es capaz de predecir correctamente su etiqueta, como "manzana", basándose en los patrones que ha aprendido de los datos previos.

2.3.1.2. Aprendizaje no supervisado

En el aprendizaje no supervisado, se trabaja con datos que no están etiquetados o cuya estructura es desconocida. Este enfoque permite explorar la estructura interna de los datos para descubrir patrones o relaciones significativas, sin necesidad de contar con una variable de resultado predefinida o una función de recompensa. (Raschka & Mirjalili, 2019)

El aprendizaje no supervisado es particularmente valioso cuando se dispone de grandes cantidades de datos no etiquetados. A través de este enfoque, los algoritmos pueden agrupar o clasificar los datos de manera autónoma, descubriendo patrones que no eran evidentes a simple vista.

El agrupamiento (clustering) es una técnica de análisis de datos que organiza grandes volúmenes de información en subgrupos significativos, conocidos como clústers, sin necesidad de contar con conocimiento previo sobre los miembros de los grupos. Cada clúster representa un conjunto de objetos que comparten similitudes entre sí, pero que se diferencian de los objetos de otros clústers. Esta técnica resulta eficaz para estructurar datos y descubrir relaciones relevantes dentro de ellos. (Raschka & Mirjalili, 2019)

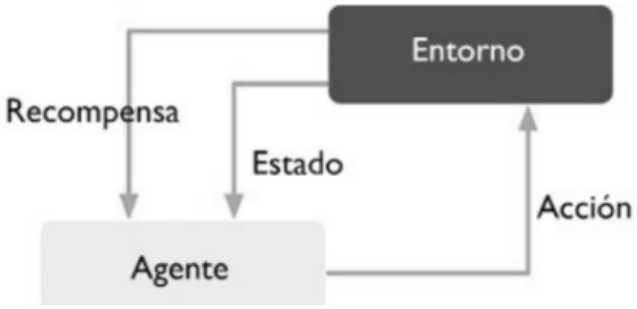
En el mundo del análisis de datos, el agrupamiento se convierte en una herramienta valiosa para entender el comportamiento de los clientes, especialmente cuando se aplica el modelo RFM (Recencia, Frecuencia, Valor monetario). Este modelo ayuda a clasificar a los clientes en diferentes grupos según tres criterios fundamentales: cuán reciente fue su última compra, con qué frecuencia compran y cuánto dinero han gastado en total. Gracias al agrupamiento, podemos detectar patrones en el comportamiento de los clientes y, por ejemplo, identificar segmentos de alto valor que compran a menudo y gastan mucho, así como grupos de clientes que podrían beneficiarse de una estrategia de retención.

2.3.1.3. Aprendizaje por refuerzo

El aprendizaje reforzado es un tipo de aprendizaje automático en el que un agente mejora su desempeño a partir de la interacción con su entorno. A diferencia del aprendizaje supervisado, donde se proporciona una etiqueta o valor correcto, en el aprendizaje reforzado el agente recibe una señal de recompensa que indica el impacto de sus acciones según una función de recompensa. A través de un proceso de ensayo y error, el agente ajusta su comportamiento para maximizar esta recompensa y optimizar su rendimiento en la tarea asignada. (Raschka & Mirjalili, 2019)

Figura 6

Funcionamiento del aprendizaje reforzado



Nota: El gráfico ilustra el proceso fundamental del aprendizaje reforzado, donde un agente se relaciona con su entorno, recibe recompensas o castigos según sus acciones, y adapta su comportamiento para maximizar las recompensas a lo largo del tiempo. Tomado de “Python Machine Learning”, por Raschka & Mirjalili, 2019.

Un ejemplo claro de aprendizaje reforzado se puede ver en cómo un niño aprende a atarse los zapatos. Al principio, el niño no tiene idea de cómo hacerlo bien, así que empieza a probar diferentes formas. Cada vez que se acerca a hacerlo correctamente, recibe una recompensa,

como un elogio. Si comete un error, simplemente intenta otra estrategia hasta que finalmente logra atarse los zapatos de manera adecuada. Con el tiempo, se vuelve más eficiente en esta tarea, ya que ha aprendido qué acciones le llevan a obtener esa recompensa. Este tipo de aprendizaje se basa en la retroalimentación que recibe a lo largo del proceso y en la mejora continua a través de la experiencia.

2.3.2. Visión Artificial

La visión artificial es un área de estudio dedicada a la captura, el análisis y la interpretación de imágenes y videos. Está vinculada a varias disciplinas como la inteligencia artificial, el aprendizaje automático, el procesamiento de señales, la computación gráfica y la robótica. (Duque Domingo, Gómez García-Bermejo, & Zalama Casanova, 2024)

Esta disciplina ha ganado mucha importancia en los últimos años gracias a su uso en diferentes áreas, como la automatización industrial, la seguridad, la medicina y los vehículos autónomos. Mediante algoritmos avanzados y técnicas de aprendizaje profundo, la visión artificial permite que las máquinas reconozcan patrones, identifiquen objetos y tomen decisiones basadas en información visual, lo que abre un mundo de nuevas oportunidades para la innovación tecnológica.

2.3.2.1. Proceso de análisis a través de Visión Artificial

- Captura: La información visual de una escena se obtiene a través de dispositivos electrónicos como cámaras, escáneres o satélites. Estos aparatos pueden ofrecer tanto imágenes individuales como secuencias de video.

- Preprocesamiento: En esta fase, se ajusta la imagen para mejorar su calidad y hacer más fácil su análisis posterior. Se utilizan técnicas para reducir el ruido, ajustar el contraste y resaltar características importantes, como bordes, líneas o áreas con diferentes morfologías.

- Segmentación: La imagen se divide en grupos de píxeles o segmentos que comparten características como niveles de gris, color o textura. Según lo que se quiera lograr con el análisis, la segmentación puede enfocarse en obtener áreas específicas o bordes, que luego pueden unirse para definir contornos.

- Morfología: Aquí se llevan a cabo operaciones que se basan en la teoría de conjuntos, aplicadas a regiones y contornos, con el fin de mejorar la segmentación. Algunas de las técnicas más comunes incluyen la erosión, dilatación, apertura y cierre.

- Extracción de características: Se calculan diferentes descriptores geométricos y topológicos de las regiones que se han identificado después de la segmentación, como el perímetro, el área, la cantidad de regiones conexas, entre otros.

- Reconocimiento de objetos: En esta etapa, se identifican los elementos que aparecen en la escena utilizando técnicas que van desde la umbralización hasta métodos más avanzados que se basan en Inteligencia Artificial.

- Interpretación de la escena. Aquí se analizan los objetos y sus relaciones para darles un significado dentro del contexto de la aplicación. Esta etapa determina las acciones a seguir según la información extraída de la imagen procesada.

2.3.2.2. Reconocimiento de patrones

El reconocimiento de patrones es una disciplina científica aplicada en computación que, mediante el uso de estadísticas, matemáticas y análisis vectorial multidimensional, permite a los algoritmos identificar características clave en datos e imágenes. Su objetivo es detectar patrones, tendencias y anomalías a través de múltiples comparaciones en cortos periodos de tiempo. Este proceso facilita la toma de decisiones al revelar estructuras ocultas en la información analizada. (Valbuena Castro, 2021)

Gracias a sus increíbles capacidades, el reconocimiento de patrones se utiliza en un montón de áreas, como la inteligencia artificial, la visión por computadora, entre otras. Hoy en día, lo vemos en acción en sistemas de reconocimiento facial, en diagnósticos médicos y en el análisis de datos financieros. Además, que, al combinarlo con modelos de aprendizaje automático, se logra mejorar tanto la precisión como la eficiencia de los procesos, lo que ayuda a optimizar la detección de información relevante en grandes volúmenes de datos.

Según Kazabov (1998) citado por Valbuena Castro (2021, p. 178), el reconocimiento de patrones es una de las herramientas más utilizadas en la ingeniería del conocimiento dentro de la inteligencia artificial. Y distingue dos tipos principales de reconocimiento de patrones:

- Reconocimiento de patrones espaciales: Este tipo de análisis se enfoca en identificar estructuras dentro de los datos. Se aplica en el procesamiento de imágenes, señales y mapas geográficos, donde es necesario segmentar regiones específicas para extraer información relevante. Un ejemplo de ello es la separación de píxeles en imágenes digitales para facilitar su análisis.

- Reconocimiento de patrones temporales: Se centra en la detección de tendencias y variaciones a lo largo del tiempo. Es utilizado en el reconocimiento de voz, el análisis de señales cardíacas para diagnosticar arritmias y el estudio de la actividad cerebral. En estos casos, se analizan frecuencias y secuencias temporales para establecer comparaciones con patrones previamente identificados.

Ambos enfoques del reconocimiento de patrones son fundamentales en diversas áreas, como la medicina, donde son esenciales para diagnósticos por imágenes y el análisis de señales médicas. En biología, facilitan los estudios genéticos; en la industria, ayudan a optimizar el control de calidad. En el ámbito financiero, se utilizan para predecir mercados; en seguridad, permiten la identificación biométrica; y en geología, son clave para el análisis sísmico y la clasificación de minerales.

2.3.3. Redes Neuronales Artificiales (ANN)

Una red neuronal artificial se puede entender de manera simplificada como un conjunto de operaciones matemáticas, en las que sumas y multiplicaciones se aplican a los datos de entrada mediante un algoritmo. A través de un proceso de ajuste continuo, la ANN es capaz de modificar sus parámetros para generar un valor de salida que se acerque al resultado esperado. Este comportamiento adaptativo le permite realizar tareas complejas, tales como la adecuación de curvas (fitting), el reconocimiento de patrones, el clustering o segmentación, y la realización de predicciones, entre otras aplicaciones en diversas áreas científicas. (Hagan, Demuth, Beale, & De Jesús, 2014)

Las redes neuronales artificiales han demostrado ser increíblemente versátiles, aplicándose a una amplia variedad de problemas computacionales. Algunos de sus usos más destacados incluyen el almacenamiento y la recuperación de datos, la clasificación de patrones, la agrupación de datos y la resolución de problemas de optimización con restricciones. Estos sistemas, que se inspiran en las redes neuronales biológicas, funcionan como modelos

estadísticos que pueden representar y procesar relaciones no lineales. Gracias a esta capacidad, pueden abordar tareas que son bastante complejas para los algoritmos tradicionales, convirtiéndose en herramientas esenciales en situaciones donde se requiere adaptabilidad, aprendizaje automático y generalización.

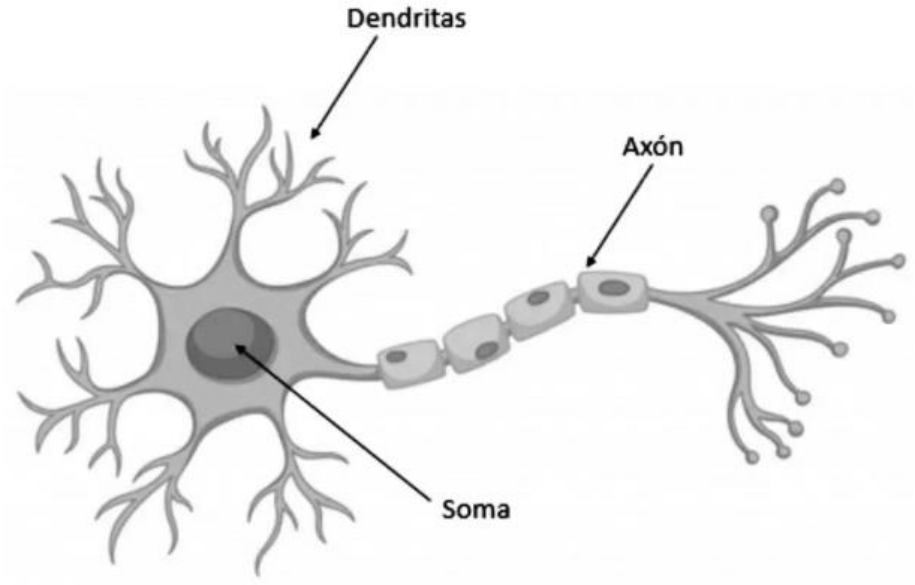
2.3.3.1. Modelo de neurona biológica

Según Restrepo, Viloria, & Robles (2021), las redes neuronales biológicas están compuestas por un intrincado conjunto de células nerviosas interconectadas que conforman el sistema neuronal del cerebro humano. Cada neurona cuenta con una estructura básica formada por dendritas, un cuerpo celular y un axón, cuya función principal es recibir, procesar y transmitir señales electroquímicas a través de un sistema de sinapsis mediado por neurotransmisores. Estas células se distinguen del resto del organismo por su capacidad de comunicación especializada, clave en los procesos de aprendizaje, especialmente durante las primeras etapas del desarrollo, cuando se forman nuevas conexiones como resultado de la experiencia.

Entender a fondo cómo funcionan las neuronas biológicas ha sido clave para crear modelos computacionales que se inspiran en ellas, como las redes neuronales artificiales. Estos modelos intentan replicar, de manera simplificada, cómo las neuronas reciben estímulos, los procesan y generan respuestas a través de sus conexiones con otras neuronas. Como se muestra en la Figura 7, una neurona está formada por dendritas, que recogen los impulsos nerviosos de otras neuronas; el soma, que se encarga de procesar esa información; y el axón, que envía un nuevo impulso a otras neuronas. Esta estructura básica es fundamental para el diseño de las unidades artificiales.

Figura 7

Modelo de una neurona biológica



Nota: El gráfico muestra la estructura fundamental de una neurona biológica, poniendo énfasis en las dendritas, el soma y el axón. Ilustra de manera sencilla cómo estas partes colaboran para permitir la transmisión de señales en el sistema nervioso, destacando su habilidad para recibir, procesar y enviar información entre las células nerviosas. Tomado de “Redes Neuronales artificiales: Qué son y cómo se entrenan”, por García & Olivera, 2019.

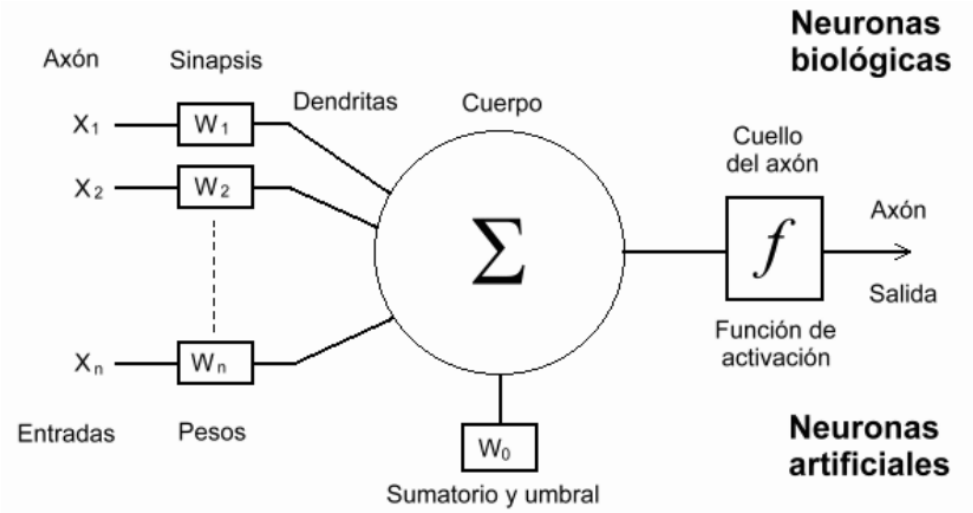
2.3.3.2. Modelo de neurona artificial

Las redes neuronales artificiales fueron desarrolladas tomando como base la estructura del cerebro humano, lo que explica por qué sus neuronas artificiales están conectadas de una manera similar a las biológicas. Estos modelos buscan imitar los mecanismos de aprendizaje y razonamiento del cerebro, replicando su capacidad de abstracción y generalización. (Restrepo, Viloria, & Robles, 2021)

Esta analogía estructural ayuda a que las redes neuronales artificiales realicen tareas complejas mediante el procesamiento paralelo de información. En la Figura 8, se puede ver una representación esquemática de una neurona artificial, que refleja cómo estas unidades están diseñadas para recibir varias entradas, procesarlas internamente y producir una salida, simulando así la forma en que se comunica el cerebro humano.

Figura 8

Comparación entre neuronas biológicas y artificiales



Nota: Esta imagen muestra una comparación entre la estructura de una neurona biológica y una neurona artificial, resaltando las similitudes entre sus partes: las entradas y los pesos son como las dendritas y sinapsis; la sumatoria y el umbral se asemejan al cuerpo celular; la función de activación es comparable al cuello del axón; y la salida, al axón mismo. Tomado de “Redes Neuronales: una visión superficial”, por Sancho Caparrini.

A partir de esta base conceptual, el modelo computacional de una red neuronal artificial se construye sobre un componente esencial: la neurona artificial. Esta neurona capta estímulos del exterior que se convierten en señales numéricas, y su funcionamiento se puede describir matemáticamente, tal como se ilustra en la Ecuación 1.

Ecuación 1: Ecuación del estímulo

$$z = \sum_{i=1}^{n} (x_i \cdot w_i) + w_0 \tag{1}$$

Donde:

- n: es el número total de entradas que recibe la neurona.
- x_i: representa el valor de la i-ésima entrada a la neurona.
- w_i: es peso asociado a la i-ésima entrada.
- w_0: es el sesgo (bias) o umbral.
- z: es la salida ponderada y ajustada de la neurona.

Luego, la salida obtenida a partir de los estímulos pasa por una función de ajuste o activación, que transforma el valor obtenido de los estímulos en una salida modificada, como se muestra en la Ecuación 2.

Ecuación 2: Ecuación de la función de activación

$$a = f_{act}(z) \tag{2}$$

Donde:

f_{act}: representa la función de activación.
a: corresponde a la salida de la neurona.

Esta función de activación es esencial para definir la salida de la neurona, ya que introduce la no linealidad en el modelo, lo que permite a la red neuronal captar relaciones complejas entre las entradas y salidas.

2.3.3.3. Función de activación

Según Restrepo, Viloria & Robles (2021), las funciones de activación juegan un papel esencial al ajustar la salida de una neurona frente a un estímulo, y además son determinantes en el proceso de entrenamiento de una red. En su análisis, identifican tres funciones como las más utilizadas por su efectividad: la activación lineal, la sigmoide logística y la tangente hiperbólica.

a) Función de activación lineal

Esta función establece una relación lineal entre el estímulo que recibe la neurona y su salida, tal como se detalla en la Ecuación 3. Cuando se aplica esta función de activación, los valores de z permanecen sin cambios, lo que significa que la salida a es igual al estímulo de entrada.

Ecuación 3: Función de activación lineal

$$f_{act}(z) = z \tag{3}$$

Donde:

f_{act}: representa la función de activación.
z: es la salida ponderada y ajustada de la neurona.

b) Función de activación sigmoide logística

Esta función de activación ajusta los valores de z dentro del rango de [0, 1]. Así, sin importar cuán alto sea el valor del estímulo, la función sigmoide logística nunca sobrepasará 1, y para valores de z que sean menores que 0, el resultado no caerá por debajo de 0. Este comportamiento se expresa matemáticamente en la Ecuación 4.

Ecuación 4: Función de activación sigmoide logística

$$f_{act}(z) = \frac{1}{1 + e^{-z}} \tag{4}$$

Donde:

f_{act}: representa la función de activación.
z: es la salida ponderada y ajustada de la neurona.
e: es la constante base del logaritmo natural.

c) Función de activación tangente hiperbólica

Esta función se describe en la Ecuación 5, y sus resultados se sitúan en el rango de [−1,1]. Por lo tanto, para cualquier valor de z, la función tangente hiperbólica no excederá 1, y para valores de z que sean menores que 0, no bajará de -1.

Ecuación 5: Función de activación tangente hiperbólica

$$f_{act}(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{5}$$

Donde:

f_{act}: representa la función de activación.
z: es la salida ponderada y ajustada de la neurona.
e: es la constante base del logaritmo natural.

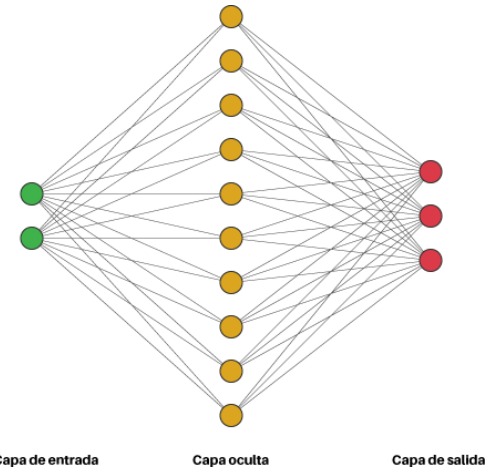
2.3.3.4. Arquitectura de las redes neuronal artificiales

Una red neuronal se forma con un conjunto de neuronas interconectadas, dispuestas en distintos niveles, creando lo que denominamos "capas de neuronas". La capa inicial, que llamamos capa de entrada, es la que recolecta los datos, mientras que la capa final, también llamada capa de

salida, genera el resultado que buscamos. Entre ambas, existen capas intermedias, y la cantidad de estas puede cambiar según la complejidad que tenga el modelo.

Figura 9

Red neuronal simple



Nota: Esta imagen se refleja la estructura básica de una red neuronal, que consta de una capa de entrada, una capa oculta y una capa de salida. Tomado de "¿Qué es una Red Neuronal?", por Sotaquirá, 2018.

Según Flórez & Fernández (2008), hay cuatro parámetros clave que definen la arquitectura de una red neuronal artificial: la cantidad de capas en el sistema, el número de neuronas en cada capa, el nivel de conectividad entre las neuronas y el tipo de conexiones que se establecen. Además, las arquitecturas neuronales se pueden clasificar según diferentes criterios.

a) Según su estructura en capas

- Redes monocapa: están formadas por una sola capa de neuronas. En estas redes, las neuronas se conectan entre sí de manera lateral. Este tipo de red es ideal para resolver problemas de autoasociación y clusterización.

- Redes multicapa: donde las neuronas se organizan en varias capas, que incluyen la capa de entrada, las capas ocultas y la capa de salida.

b) Según el flujo de datos en la red

- Redes unidireccionales: también conocidas como redes de propagación hacia adelante, donde ninguna salida neuronal se convierte en entrada para unidades de la misma capa o de

capas anteriores. La información se mueve en una sola dirección, desde las neuronas de entrada hasta las neuronas de salida de la red.

- Redes de propagación hacia atrás: en las que las salidas de las neuronas pueden actuar como entradas para unidades del mismo nivel (conexiones laterales) o de niveles anteriores.

c) Según el tipo de respuesta de la red

- Redes heteroasociativas: que están diseñadas para que, al presentar un patrón A, el sistema responda con otro patrón diferente B. Estas redes requieren al menos dos capas: una para captar y almacenar la información de entrada y otra para mantener la salida con la información asociada.

- Redes autoasociativas: están entrenadas para asociar un patrón consigo mismo. Su función principal es que, al presentar un patrón A' que ha sido alterado por ruido, la respuesta sea el patrón original A, actuando, así como un filtro.

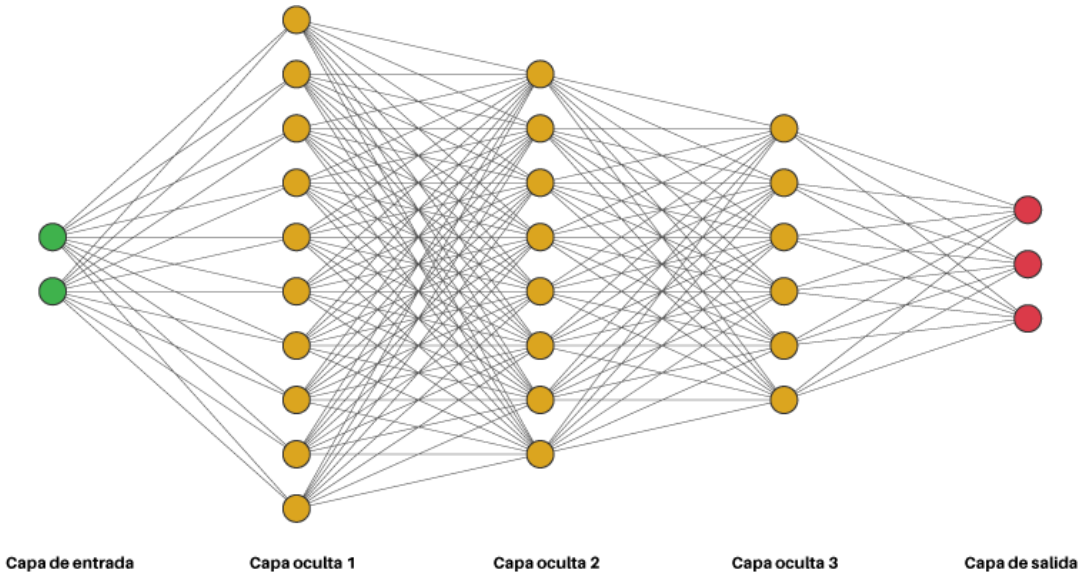
2.3.4. Redes neuronales profundas (DNNs)

Una red neuronal profunda es un modelo de aprendizaje automático que trata de replicar cómo el cerebro humano procesa la información. A diferencia de los algoritmos tradicionales, que se basan en reglas fijas, las DNN tienen la capacidad de aprender patrones a partir de los datos y hacer predicciones basadas en experiencias pasadas. (Mercier, 2025)

Las redes neuronales pueden tener un número variable de capas ocultas, y a medida que se añaden más capas, su complejidad aumenta. Las redes más tradicionales suelen contar con entre 2 y 3 capas ocultas, mientras que las redes de aprendizaje profundo pueden llegar a tener muchas más. Al incorporar más capas, la red se vuelve más "profunda", lo que da origen al término "Red Neuronal Profunda".

Figura 10

Red neuronal profunda



Nota: En esta imagen se muestra la estructura básica de una Red Neuronal Profunda, la cual se caracteriza por tener múltiples capas ocultas entre la capa de entrada y la capa de salida. Tomado de "¿Qué es una Red Neuronal?", por Sotaquirá, 2018.

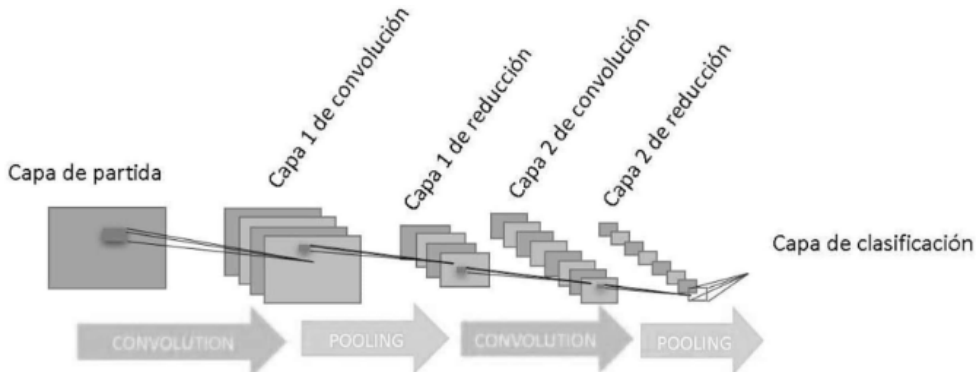
2.3.5. Redes neuronales convolucionales (CNNs)

Según Pérez & Gegúndez (2021), a diferencia de las redes neuronales tradicionales que solo utilizan capas de neuronas, las redes neuronales convolucionales añaden capas extra, como las de convolución y pooling. Estas capas juegan un papel crucial en las primeras etapas del procesamiento. Se encargan de extraer características clave de los datos en bruto, transformando el espacio de entrada en uno más compacto. Esto hace que las capas neuronales que vienen después tengan un trabajo más fácil, ya que reciben una representación más manejable y significativa de los datos.

Por lo que, se puede distinguir dos faces en el funcionamiento de las redes neuronales convolucionales: una fase de extracción, donde se extraen las características más relevantes de los datos utilizando capas como las de convolución y pooling. Y una fase de clasificación, en la que esas características se utilizan como entrada para una red neuronal convencional, que se encarga de realizar la clasificación final.

Figura 11

Arquitectura de una red neuronal convolucional



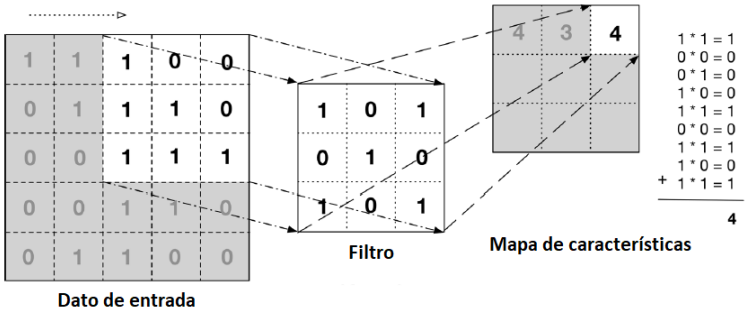
Nota: La imagen muestra de manera general cómo está estructurada una red neuronal convolucional. Comienza con una capa de entrada y luego avanza a través de las etapas de extracción de características y clasificación. Tomado de “Redes neuronales convolucionales: un modelo de Deep Learning en imágenes diagnósticas”, por Lubinus, Rueda, Marconi, & Arias, 2021.

2.3.5.1. Capas de convolución

Su nombre viene dado de la operación clave en su funcionamiento: la operación de convolución. Una convolución es simplemente una multiplicación elemento por elemento de dos matrices. Cuando realizamos una operación de convolución, lo que hacemos es aplicar los valores de la segunda matriz (que también se conoce como filtro o kernel) a la primera matriz.

Figura 12

Operación de convolución



Nota: El gráfico muestra el proceso de convolución utilizando matrices, donde se aplica un filtro (kernel) sobre un dato de entrada, generando un mapa de características. Tomado de “Aprendizaje Profundo: Redes Neuronales Convolucionales CNN”, por Sotil, 2022.

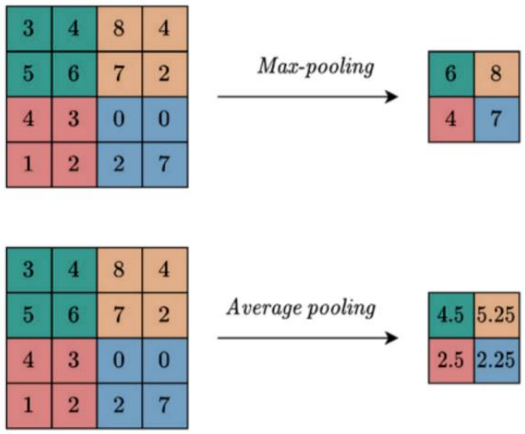
2.3.5.2. Capas de pooling

El cálculo de la salida de una capa de pooling se basa en el tipo de operación que elijas, que normalmente se centra en un valor representativo de la región que estás analizando. Las operaciones más comunes son:

- Max-pooling: asigna como salida en la posición actual el máximo valor de la región analizada.
- Average pooling: asigna como salida en la posición actual la media de la región analizada.

Figura 13

Operaciones de pooling



Nota: El gráfico ilustra las operaciones de max-pooling y average pooling sobre una región o mapa de características. Tomado de “Deep learning: fundamentos, teoría y aplicación”, por Pérez & Gegúndez, 2021.

2.3.5.3. Yolo

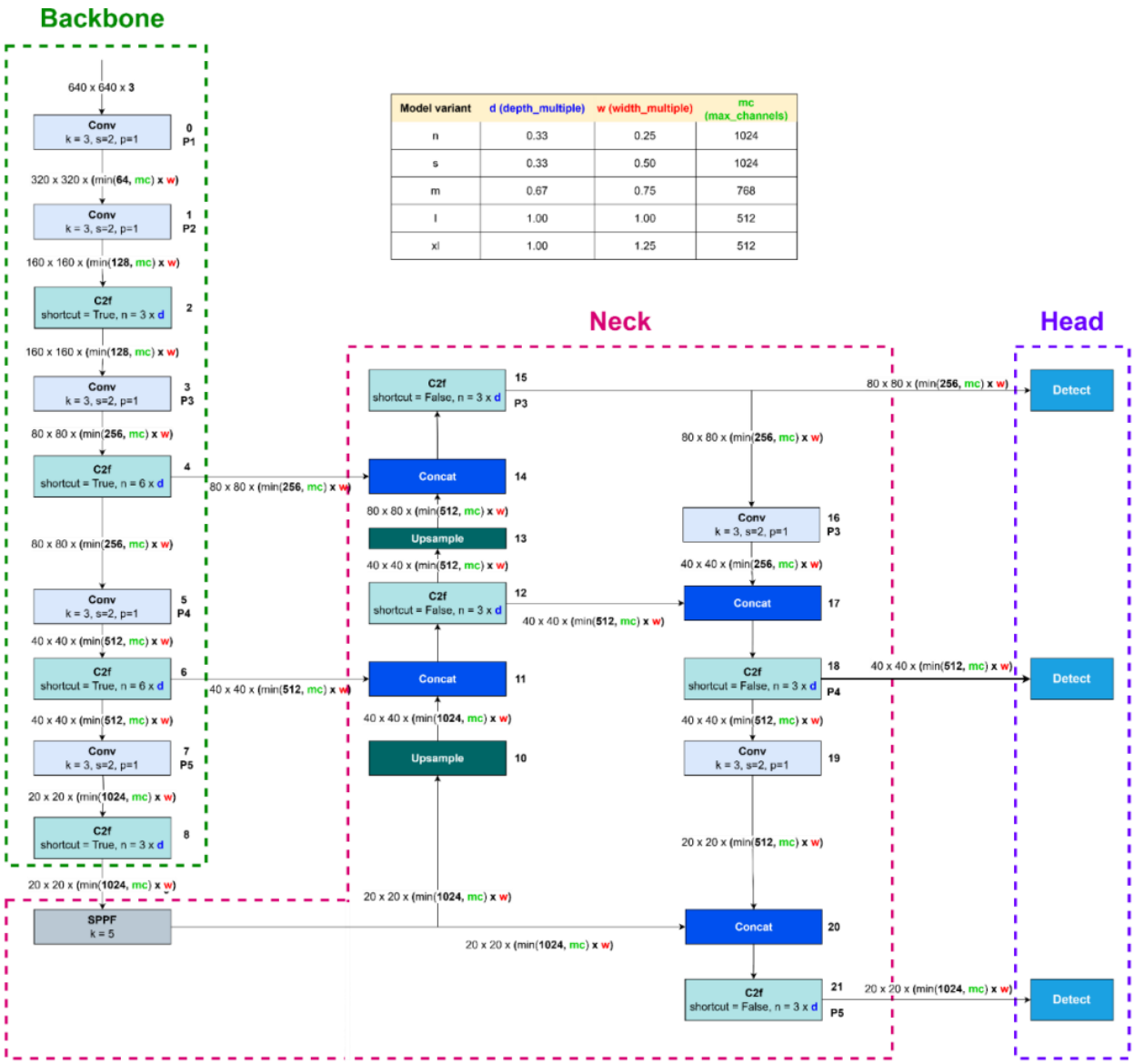
Según la documentación oficial de Ultralytics (s.f.), YOLO (You Only Look Once) es una familia de modelos de visión por computadora diseñada para detectar y segmentar objetos en imágenes de manera rápida, precisa y eficiente. Su característica principal es procesar la imagen completa en una sola pasada por la red neuronal, lo que lo hace mucho más rápido que los métodos tradicionales basados en regiones de interés o múltiples etapas.

YOLO ha evolucionado a lo largo de los años mediante múltiples versiones, desde YOLOv2 hasta YOLOv11, consolidándose como una familia de modelos líder en detección y segmentación de objetos. Cada versión ha aportado mejoras que han permitido ampliar sus capacidades y

aplicaciones en visión por computadora. En la Figura 14 se muestra la arquitectura correspondiente a la versión YOLOv8.

Figura 14

Arquitectura YOLOv8



Nota: En la figura se muestra la arquitectura de YOLOv8, así como las variantes del modelo disponibles: n, s, m, l, xl, que permiten ajustar el equilibrio entre precisión y eficiencia según el caso de uso.Tomado de “YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review” (pág. 4), por Hidayatullah, Syakrani, Sholahuddin, Gelar & Tubagus, 2025.

La arquitectura de los modelos YOLO se compone de tres componentes principales:

- Columna vertebral (Backbone): Es una red neuronal convolucional encargada de extraer las características más relevantes de la imagen de entrada. Su diseño optimiza el flujo de información entre capas, generando representaciones precisas de los objetos y facilitando que las etapas posteriores procesen datos más completos y útiles.

- Cuello (Neck): Funciona como un concentrador de información, fusionando los mapas de características generados por la columna vertebral a diferentes resoluciones. Esta combinación de información semántica y espacial permite al modelo detectar objetos de distintos tamaños con mayor precisión, especialmente aquellos pequeños o complejos.

- Cabeza de detección (Head): Es el módulo responsable de generar las predicciones finales, incluyendo cajas delimitadoras, probabilidades de clase y puntuaciones de confianza para cada celda de la imagen. Estas predicciones se integran posteriormente para producir las detecciones finales que identifican los objetos presentes.

2.3.5.4. Faster RCNN

Faster R-CNN es un sistema de detección de objetos avanzado que combina eficiencia y precisión mediante la integración de varios módulos. La arquitectura está diseñada como una red unificada, donde las características convolucionales se comparten entre la generación de propuestas de regiones y la detección de objetos (Ren, He, Girshick, & Sun, 2016).

Las partes principales de Faster R-CNN se pueden dividir en:

- Red troncal (Backbone): Extrae las características más relevantes de la imagen de entrada, proporcionando mapas de características que sirven como base para los módulos posteriores.

- Red de Propuestas de Región (RPN – Region Proposal Network): Genera automáticamente propuestas de regiones con puntajes de objetividad, utilizando los mapas de características de la red troncal y permitiendo la regresión de cajas delimitadoras y la clasificación de regiones.

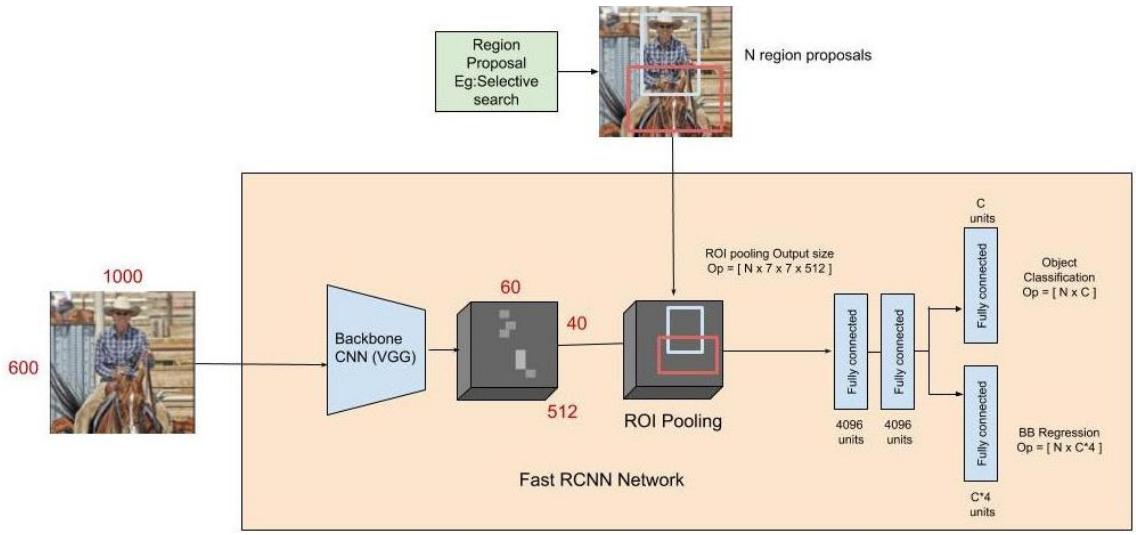
- ROI Pooling: Convierte las regiones de interés de tamaño variable en representaciones de tamaño fijo, asegurando que puedan ser procesadas correctamente por el detector.

- Detector (Head): Utiliza las regiones propuestas para generar las predicciones finales de los objetos, cajas delimitadoras, probabilidades de clase y puntuaciones de confianza.

Una característica clave de Faster R-CNN es la flexibilidad de su backbone, que permite modificada o reemplazada según los requerimientos del sistema, utilizando arquitecturas como VGG-16, ResNet o MobileNet. Como se muestra en la Figura 15, el funcionamiento del sistema pasando por los bloques opartes

Figura 15

Arquitectura Faster-RCNN



Nota: El diagrama muestra la arquitectura de Faster R-CNN, incluyendo la red troncal, la generación de propuestas de región, el ROI Pooling y la detección final de objetos. Tomado de “Fast RCNN”, por Jain, (2025).

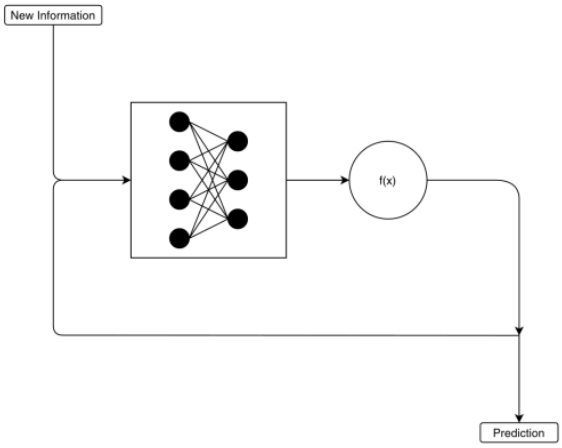
2.3.6. Redes neuronales recurrentes (RNNs)

Las redes neuronales recurrentes son un fascinante avance en el campo del aprendizaje profundo. Lo que realmente las hace destacar es su capacidad para manejar datos en secuencia. Gracias a esta habilidad, se utilizan en una variedad de tareas, como el análisis de video, la generación de subtítulos para imágenes, el procesamiento del lenguaje natural y el análisis musical. A diferencia de otras redes neuronales artificiales que tratan las entradas de manera independiente, las RNN pueden captar las relaciones temporales y secuenciales entre los datos, lo que las convierte en una herramienta poderosa en el mundo de la inteligencia artificial. (Arana, 2021)

Además de su habilidad para manejar datos secuenciales, las redes neuronales recurrentes son notables por su estructura interna, que les permite “recordar” información del pasado mientras procesan nuevos datos. Esto se logra gracias a la retroalimentación entre sus nodos, lo que ayuda a entender el contexto en tareas como la traducción automática o el reconocimiento de voz. Como se muestra en la Figura 16, esta arquitectura permite que la salida de una etapa se use como entrada en la siguiente, capturando así las relaciones temporales en la información.

Figura 16

Diagrama cíclico de una RNN



Nota: El diagrama ilustra una red neuronal recurrente, donde la retroalimentación permite que el modelo maneje secuencias de datos, recordando información del pasado para hacer predicciones. Tomado de “Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales” (p. 6), por Arana, 2021.

Tal y como se observa en la Figura 16, las redes neuronales recurrentes tienen la capacidad de procesar datos secuenciales, teniendo en cuenta el contexto previo en cada paso. Sin embargo, enfrentan una limitación significativa conocida como memoria a corto plazo. Este inconveniente está vinculado al problema del desvanecimiento del gradiente, lo que complica que la red retenga información relevante cuando las secuencias son muy extensas. En otras palabras, a medida que la red avanza en el procesamiento de nuevos datos, tiende a olvidar elementos clave del pasado, lo que impacta negativamente en la calidad de sus predicciones o análisis en situaciones donde hay dependencias a largo plazo.

2.3.6.1. Celdas Long Short-Term Memory (LSTM)

Las celdas Long Short-Term Memory son unidades especializadas dentro de las redes neuronales recurrentes, diseñadas para procesar secuencias de datos manteniendo información durante intervalos prolongados. Este tipo de celda soluciona el problema del desvanecimiento del gradiente presente en las RNN tradicionales, lo que permite aprender relaciones de dependencia a largo plazo dentro de una serie temporal o textual (Hochreiter & Schmidhuber, 1997). El funcionamiento interno de una LSTM se basa en un conjunto de compuertas que regulan el flujo de información dentro de la celda. Estas compuertas son:

- Compuerta de olvido: decide qué información del estado previo debe descartarse.
- Compuerta de entrada: controla qué nueva información se incorporará al estado interno.
- Compuerta de salida: determina qué parte del estado actualizado se utilizará para generar la salida del paso actual.

Una celda LSTM recibe como entrada el vector actual $x(t)$, así como la salida $h(t-1)$ y el estado interno $c(t-1)$ provenientes del paso de tiempo anterior. Con estos valores, la celda calcula una nueva salida $h(t)$ y un nuevo estado $c(t)$, manteniendo así una memoria acumulada que se actualiza en cada ciclo. Esta memoria permite que la salida incorpore información relevante de todos los pasos previos, lo que hace a las LSTM especialmente útiles para representar dependencias complejas en secuencias largas (McCaffrey, 2018).

2.3.6.2. Celdas Gated Recurrent Unit (GRU)

Las celdas Gated Recurrent Unit fueron propuestas como una alternativa más simple y computacionalmente eficiente que las LSTM, manteniendo una capacidad similar para capturar dependencias de largo plazo en secuencias. A diferencia de las celdas LSTM, las GRU no poseen un estado de celda independiente, en su lugar, gestionan toda la información únicamente mediante un estado oculto. Esto reduce el número de parámetros y facilita el entrenamiento, especialmente en conjuntos de datos más pequeños o arquitecturas donde se requiere mayor velocidad (Dobilas, 2022).

El funcionamiento interno de una GRU puede dividirse en tres componentes principales, los cuales controlan qué información recordar, olvidar y transferir a través del tiempo.

- Puerta de reinicio: combina la entrada actual $x(t)$ y el estado oculto previo $h(t-1)$ y pasa el resultado por una función sigmoide, que produce valores entre 0 y 1. Estos valores determinan qué parte de la memoria anterior se olvida o se conserva parcialmente.

- Puerta de actualización: aplica una función sigmoide sobre $x(t)$ y $h(t-1)$ con pesos distintos. Controla cuánto del estado previo se mantiene y cuánto se reemplaza con el candidato de estado oculto, actuando como un regulador de la memoria a largo plazo.

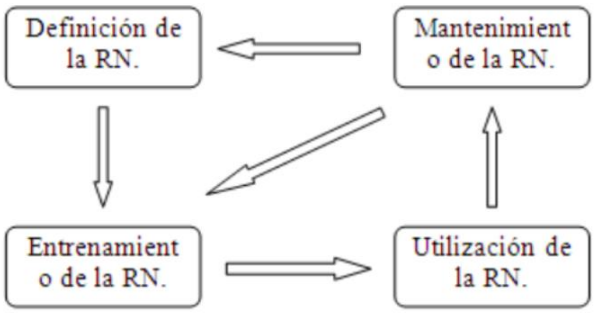
- Candidato de estado oculto: combina la entrada y el estado previo modulado por la puerta de reinicio, pasa el resultado por una función de activación $tanh$ y la puerta de actualización determina cuánto se incorpora al nuevo estado $h(t)$.

2.3.7. Metodología de desarrollo de una red neuronal

El desarrollo de una red neuronal artificial implica varias etapas clave que aseguran su correcto funcionamiento y su capacidad para adaptarse al problema que se quiere resolver. Según Quispe (2017), estas etapas incluyen la definición de la red, donde se determinan las capas y las funciones de activación; el entrenamiento, en el que se selecciona el tipo de aprendizaje y los algoritmos; la utilización de la red, que se refiere al procesamiento de entradas para generar salidas; y, por último, el mantenimiento, que permite ajustar la red ante nuevos datos y situaciones, garantizando su efectividad a lo largo del tiempo.

Figura 17

Ciclo de vida de una red neuronal



Nota: El diagrama ilustra las fases clave en el desarrollo de una red neuronal: definición, entrenamiento, utilización y mantenimiento continuo para asegurar su efectividad a lo largo del tiempo. Tomado de “Redes neuronales artificiales, metodología de desarrollo y aplicaciones” (p. 9), por Quispe, 2017.

a) Definición de la red neuronal

La fase de definición de la red neuronal es crucial, ya que implica elegir el modelo adecuado e identificar las variables clave que nos ayudarán a resolver el problema. En esta etapa, se determina el tamaño de la red, el tipo de problema que estamos abordando y el tipo de asociación que se va a utilizar. También es importante seleccionar las funciones de transferencia o umbral, que son las que activan o desactivan las neuronas en las capas ocultas y de salida, además de los algoritmos de entrenamiento que se emplearán.

b) Entrenamiento de la red neuronal

La fase de entrenamiento de una red neuronal consiste en elegir un método de entrenamiento, que se realiza a través de un algoritmo de aprendizaje. Durante este proceso, la red ajusta los pesos de las conexiones entre las diferentes capas para optimizar su rendimiento. Además, la red tiene la habilidad de adaptarse y cambiar en función de la información que ha recibido anteriormente, lo que le permite aprender y ajustarse a los datos de entrada.

c) Utilización de la red neuronal

La fase de utilización de la red neuronal es cuando la red reacciona a un estímulo de entrada. A diferencia del aprendizaje, en esta etapa no se modifican los pesos ni se calculan errores, ya que esos procesos solo tienen lugar durante el entrenamiento. En esta fase de utilización, la red simplemente produce una salida basada en las entradas que recibe.

d) Mantenimiento de la red neuronal

El mantenimiento de una red neuronal es la etapa final de su ciclo de vida. Dado que muchas de estas redes se entrenan para resolver problemas que pueden cambiar, es crucial realizar validaciones constantes a lo largo del tiempo para asegurar que sigan funcionando correctamente. A medida que aparecen nuevos datos que la red no conoce, es necesario hacer ajustes o reentrenamientos para garantizar que siga siendo efectiva en su tarea.

2.4. Herramientas

2.4.1. Python

Python es un lenguaje de programación de alto nivel ampliamente utilizado en el desarrollo de software, aplicaciones web, ciencia de datos y aprendizaje automático. Se caracteriza por su sintaxis sencilla y fácil de comprender, lo que permite a los desarrolladores escribir programas con menos líneas de código en comparación con otros lenguajes. Python cuenta con una amplia biblioteca estándar y numerosas bibliotecas adicionales que proporcionan funciones reutilizables para diversas tareas. (Amazon Web Services, s.f.)

2.4.2. Tensorflow

TensorFlow es una plataforma de código abierto diseñada para el desarrollo y la implementación de aplicaciones de aprendizaje automático. Proporciona un ecosistema flexible de herramientas, bibliotecas y recursos que permiten a investigadores y desarrolladores crear, entrenar e implementar modelos de aprendizaje automático de manera eficiente. Además, ofrece APIs de alto nivel, como Keras, que facilitan la construcción y el entrenamiento de modelos, permitiendo su implementación en diversas plataformas como servidores, la nube, dispositivos móviles, navegadores y otros entornos tecnológicos. (Google Open Source, s.f.)

2.4.3. Keras

Keras es una biblioteca de código abierto escrita en Python utilizada para la creación y desarrollo de redes neuronales. Funciona como una interfaz de programación (API) que permite construir modelos de aprendizaje profundo de forma sencilla mediante bloques modulares que representan las capas de una red neuronal. Keras no opera como un framework independiente, sino que utiliza motores de backend de frameworks de aprendizaje automático como TensorFlow, Theano o Microsoft Cognitive Toolkit. (IONOS, 2020)

2.4.4. Laravel

Laravel es un framework para el desarrollo de aplicaciones web que proporciona una estructura y un punto de partida para crear aplicaciones. Su objetivo es permitir que los desarrolladores se concentren en el desarrollo de la aplicación mientras el framework gestiona diversos aspectos del sistema. Laravel incluye características como inyección de dependencias, una capa de

abstracción de bases de datos, colas de trabajo, tareas programadas y herramientas para realizar pruebas unitarias y de integración. Además, ofrece una sintaxis expresiva y una estructura organizada que facilita el desarrollo de aplicaciones web. (Laravel, s.f.)

2.4.5. Flask

Flask es un framework web minimalista desarrollado en Python y publicado en 2010 por Armin Ronacher. A diferencia de los frameworks full-stack como Django, Flask adopta un enfoque más ligero, ya que incluye únicamente componentes básicos como el motor de plantillas Jinja y una biblioteca llamada Werkzeug. Este framework permite integrar funciones y herramientas de terceros según las necesidades del desarrollador. Además, Flask es un software gratuito y de código abierto distribuido bajo licencia BSD. (IONOS, 2023)

2.4.6. FastAPI

FastAPI es un framework web desarrollado en Python que permite crear interfaces de programación de aplicaciones (API) de forma rápida y sencilla. Proporciona herramientas y funcionalidades que facilitan la creación de aplicaciones web capaces de recibir y enviar datos a través de Internet, permitiendo la comunicación entre diferentes aplicaciones. Este framework utiliza programación asincrónica, lo que permite ejecutar múltiples tareas al mismo tiempo sin que se bloqueen entre sí, facilitando el manejo de múltiples solicitudes de usuarios. (Yadav, 2024)

CAPÍTULO 3.
MARCO APLICATIVO

3.1. PLANIFICACIÓN

Para el desarrollo del presente proyecto, ser realizó la planificación en base a las metodologías XP y CRISP-DM que nos permitieron organizar de forma clara las actividades que se deben realizar junto con sus fases y entregables. En la Tabla 1 se muestra la planificación para el desarrollo del presente proyecto.

Tabla 1

Planificación del proyecto según la metodología

Objetivo	Metodología	Actividades	Entregables
Análisis de los procesos actuales	CRISP DM - Comprensión del Negocio	Recopilación de información mediante entrevistas.	Diagramas de actividades de los procesos administrativos.
		Observación de las actividades.	Limitaciones detectadas.
	CRISP DM - Comprensión de los Datos	Identificación de la documentación recepcionada.	Tabla de documentación requerida.
		Identificación de la información extraída de la documentación.	Tabla de información relevante.
	XP - Planificación	Elaboración de historias de usuario con criterios de aceptación.	Historias de usuario.
		Planificación de iteraciones según esfuerzo y tiempo.	Plan de iteraciones.
Construcción de datasets	CRISP DM – Preparación de los Datos	Dataset de Documentos para la Detección y Clasificación de Texto	
		Recopilación y escaneo de documentos.	Tabla de la documentación recolectada.
		Generación de documentos sintéticos.	Diagrama de flujo de la generación de documentos. Tabla de transformaciones aplicadas.
		Etiquetado de información de interés.	Tablas de etiquetas definidas para documentos. Figura del proceso de etiquetado.
		Distribución del dataset en subconjuntos.	Histograma de la distribución de muestras de documentos. Diagrama de estructura del dataset.
		Dataset de Fragmentos de Texto para Extracción de Información	

		Extracción de fragmentos de texto de los documentos etiquetados	Figura de fragmentos de texto extraídos.
		Preprocesamiento de los recortes.	Figura de recortes de texto tras el preprocesamiento.
		Etiquetado del texto contenido en los recortes.	Figura de recortes de texto con sus etiquetas.
		División de datasets en conjuntos de entrenamiento y validación.	Diagrama de estructura del dataset.
Evaluación de arquitectura de redes neuronales	Evaluación de Arquitecturas en Detección y Clasificación de Regiones		
	CRISP DM - Modelado	Revisión de literatura sobre arquitecturas CNN.	Tabla de arquitecturas seleccionadas para evaluación.
	CRISP-DM – Evaluación	Definición de métricas de evaluación.	Tablas de métricas de evaluación seleccionadas.
		Definición de condiciones de entrenamiento.	Configuración inicial para la comparación.
		Entrenamiento de modelos	Gráficos y reportes de métricas finales de desempeño.
		Comparación de desempeño de los modelos entrenados.	Tabla comparativa con métricas.
	Evaluación de Arquitecturas Híbridas para Extracción de Texto		
	CRISP DM - Modelado	Revisión de literatura sobre arquitecturas híbridas CNN+RNN.	Tabla de arquitecturas seleccionadas para evaluación.
	CRISP-DM – Evaluación	Definición de condiciones de entrenamiento.	Configuración inicial para la comparación.
		Entrenamiento de modelos	Gráficos y reportes de métricas finales de desempeño.
		Definición de métricas de evaluación.	Tablas de métricas de evaluación seleccionadas.
		Comparación de desempeño de los modelos entrenados.	Tabla comparativa con métricas.
Desarrollo del sistema	XP - Iteración 1	Planificación	Tabla de tareas.
		Diseño	Diagramas de casos de uso y ER.
		Desarrollo	Capturas de pantalla del sistema.
		Pruebas	Casos de Prueba
	XP – Iteración 2	Planificación	Tabla de tareas.

		Diseño	Diagramas de secuencias y casos de uso.
		Desarrollo	Capturas de pantalla del sistema.
		Pruebas	Casos de Prueba
	XP – Iteración 3	Planificación	Tabla de tareas.
		Diseño	Diagramas de casos de uso y ER.
		Desarrollo	Capturas de pantalla del sistema.
		Pruebas	Casos de Prueba
	XP – Iteración 4	Planificación	Tabla de tareas.
		Diseño	Diagramas de casos de uso y ER.
		Desarrollo	Capturas de pantalla del sistema.
		Pruebas	Casos de Prueba
	XP – Iteración 5	Planificación	Tabla de tareas.
		Diseño	Diagramas de secuencias.
		Desarrollo	Capturas de pantalla del sistema.
		Pruebas	Casos de Prueba
	XP – Iteración 6	Planificación	Tabla de tareas.
		Diseño	Diagramas de secuencias.
		Desarrollo	Capturas de pantalla del sistema.
		Pruebas	Casos de Prueba
Validación del sistema	Evaluación en la Documentación Personal	Medición de tiempos de registro con el sistema y sin el sistema.	Tabla con reducción porcentual de tiempos por sacramento.
		Revisión de registros oficiales para detectar inconsistencias.	Tabla de probabilidad de inconsistencias.
		Ejecución de pruebas funcionales	Tablas de pruebas funcionales.
	Evaluación Técnica	Evaluación de la calidad del sistema (usabilidad, funcionalidad, mantenibilidad y portabilidad)	Tabla resumen con resultados de cada factor y puntaje global.
	Evaluación Económica	Cálculo de costos de desarrollo de software mediante COCOMO.	Tabla de Costos de Desarrollo. Tabla de Costos de Operación. Tabla de Costo Total.

Nota: La tabla muestra la planificación del presente proyecto para el cumplimiento de los objetivos propuestos. Elaboración propia.

3.2. ANÁLISIS DE LOS PROCESOS ACTUALES

Con el fin de comprender el funcionamiento del sistema administrativo y documental de la Parroquia San Miguel Arcángel, se llevó a cabo una recopilación de información a través de entrevistas al personal, observación directa de las actividades administrativas y revisión de la documentación que los feligreses presentan para recibir los diferentes sacramentos.

3.2.1. Recopilación de Información Mediante Entrevistas

Las entrevistas con el personal se realizaron con el fin de conocer sus funciones, los procesos administrativos habituales y las limitaciones del sistema actual. Los resultados obtenidos se exponen a continuación, lo que permite identificar de manera clara las necesidades de la Parroquia y sirve como base para definir los requerimientos adecuados del sistema a desarrollar, asegurando que el proyecto aborde de forma efectiva los problemas identificados.

3.2.1.1. Funciones del personal

La Parroquia San Miguel Arcángel cuenta con roles y responsabilidades para atender las necesidades de la comunidad en cuanto a la recepción de sacramentos y la gestión documental asociada. A continuación, se describen las funciones del personal.

- Párroco: Supervisa todas las actividades parroquiales, autoriza procesos, celebra los sacramentos y es responsable de la firma de todos los certificados emitidos. Además, tiene la responsabilidad última sobre la documentación administrativa.

- Sacerdotes: Encargados de las celebraciones litúrgicas, participan ocasionalmente en la emisión de certificados y recepción de la documentación necesaria para inscribirse en los cursos que permiten recibir los diferentes sacramentos.

- Secretaría: El eje central en la administración documental. Recepcionan toda la documentación entregada por los feligreses, verifican que se cumplan los requisitos para la inscripción a los cursos sacramentales y controlan el pago correspondiente.

- Catequistas: Responsables de la formación religiosa a los feligreses mediante cursos sacramentales. Además, gestionan el control de asistencia y validan la aprobación del curso, requisito indispensable para que los feligreses puedan recibir el sacramento.

3.2.1.2. Procesos administrativos

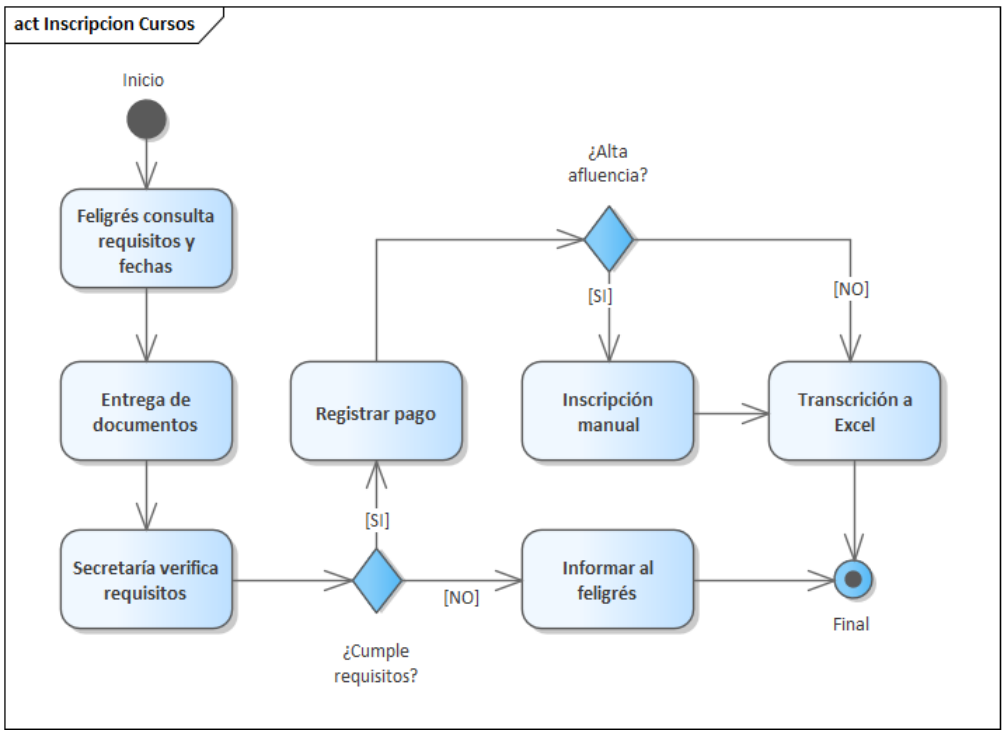
a) Inscripción a los cursos sacramentales

Los feligreses interesados en recibir algún sacramento se acercan a la parroquia para consultar los requisitos y las fechas de inicio. Luego, entregan la documentación requerida a secretaría, donde se verifica el cumplimiento de los requisitos, como el rango de edad correspondiente o la recepción previa de sacramentos necesarios. También se registra el pago, ya sea parcial o total.

Cuando hay una gran afluencia de personas, la inscripción se realiza manualmente en hojas físicas para agilizar la atención. Posteriormente, esta información se transfiere a una hoja de cálculo en Excel, donde se organizan las listas de inscritos que luego se entregan a los catequistas. En los casos en que el flujo de personas es bajo, la inscripción se realiza directamente en el archivo digital.

Figura 18

Diagrama de actividades del proceso de inscripción



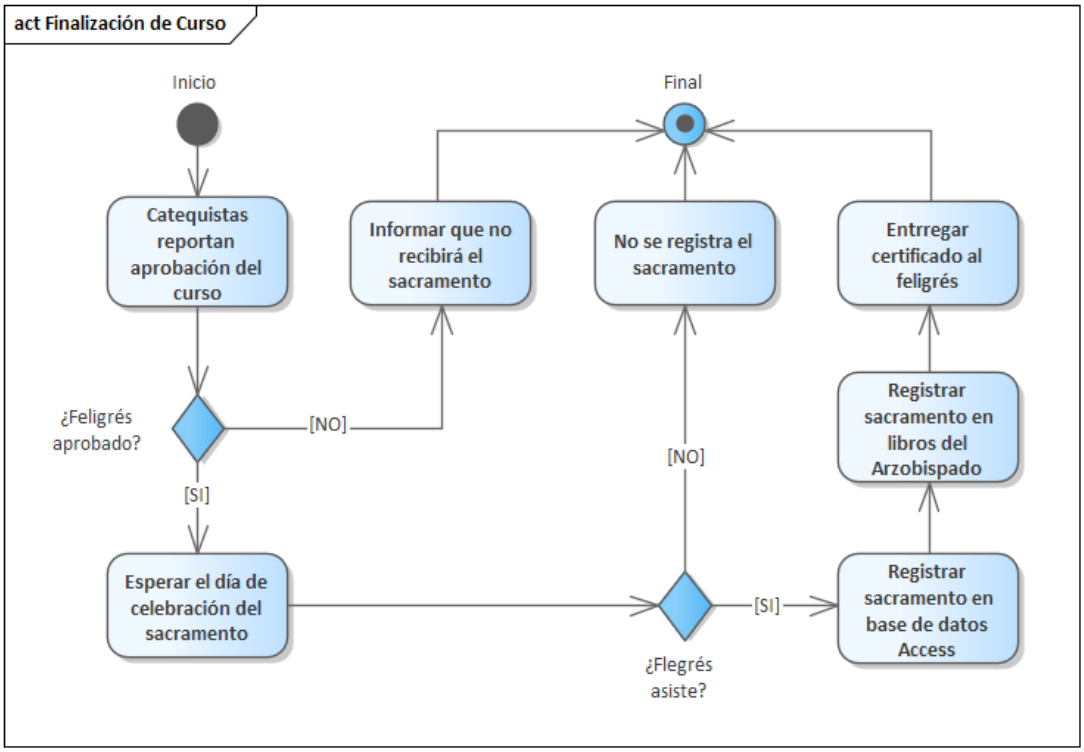
Nota: Representa el proceso mediante el cual el feligrés se inscribe para recibir un sacramento, entregando sus documentos y siendo registrado manualmente por secretaria. Elaboración propia mediante Enterprise Architect.

b) Finalización del curso y recibimiento del sacramento

Una vez finalizado el curso, los catequistas reportan a secretaría la asistencia y aprobación de los feligreses. Con esta información, se espera el día en que los participantes recibirán el sacramento. En caso de que algún feligrés no se presente, no se realiza el registro formal del sacramento. Si asisten con normalidad, se procede al registro del sacramento en la base de datos Access y en los libros oficiales del Arzobispado, lo que garantiza su validez oficial.

Figura 19

Diagrama de actividades del proceso de registro de sacramentos



Nota: La figura ilustra el proceso desde la finalización del curso hasta el registro del sacramento en la base de datos, incluyendo celebración y entrega del certificado. Elaboración propia mediante Enterprise Architect.

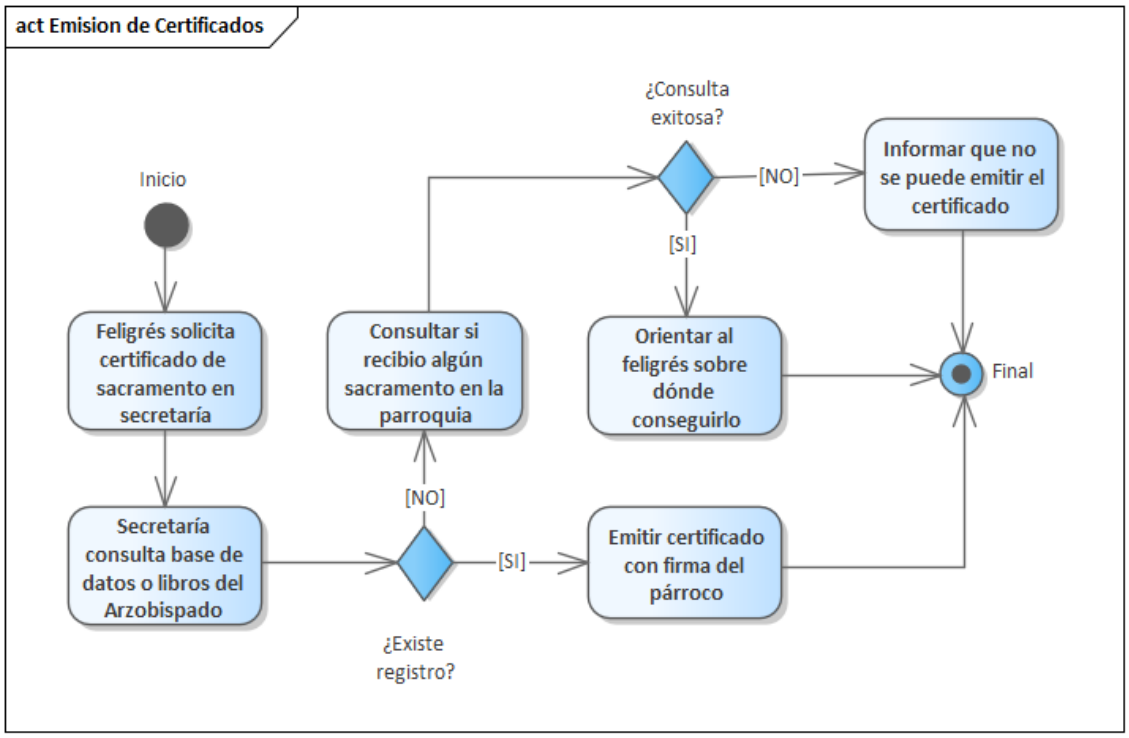
c) Emisión de certificados sacramentales

Los feligreses que hayan recibido sacramentos anteriormente y necesiten un certificado para trámites o para acceder a otro sacramento, pueden solicitarlo en la secretaría. Para ello, se consulta la base de datos o los libros oficiales, y se emite el certificado correspondiente con la firma del párroco.

En algunos casos, se presentan personas que no recibieron el sacramento en esta parroquia. Si se cuenta con información suficiente, secretaría podría orientar sobre la parroquia en la cual posiblemente recibieron el sacramento y dónde podrían solicitar el certificado correspondiente.

Figura 20

Diagrama de actividades del proceso de solicitud de certificados



Nota: Detalla el procedimiento que se sigue cuando un feligrés solicita un certificado sacramental. Elaboración propia mediante Enterprise Architect.

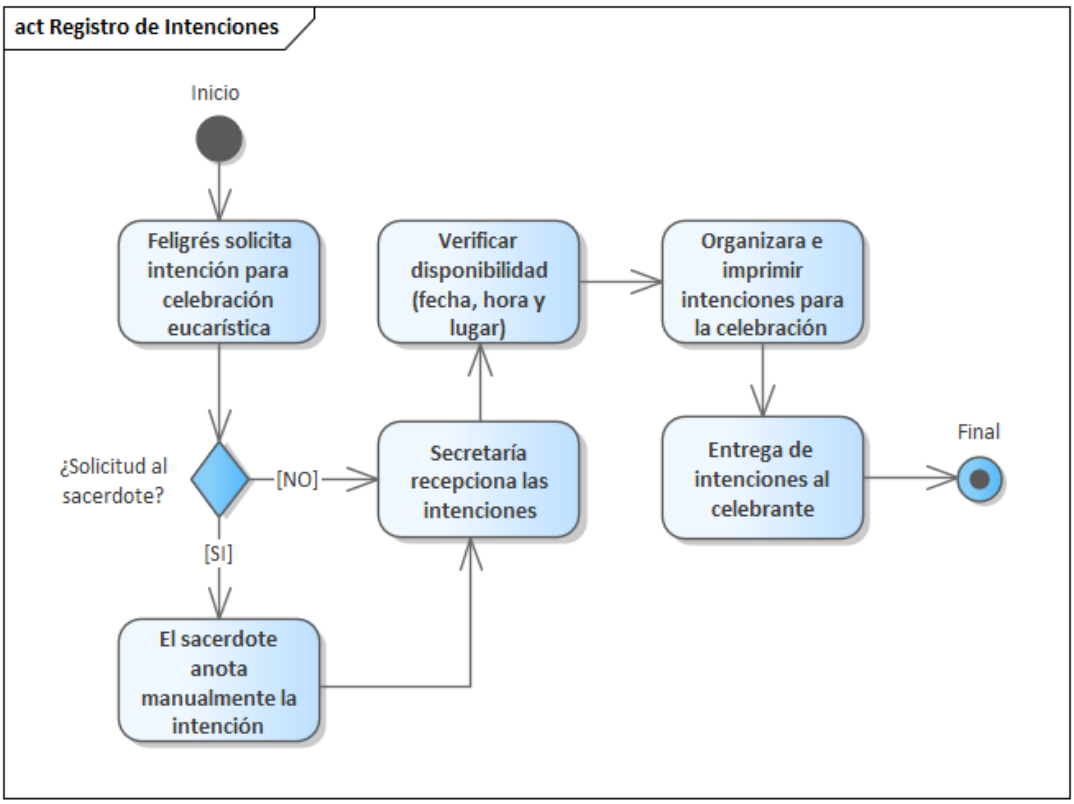
d) Registro de intenciones

La parroquia también recibe solicitudes de intenciones para misas, ya sea por salud, necesidades personales o en memoria de un difunto. Estas intenciones pueden ser solicitadas directamente en la secretaría o a los sacerdotes, quienes las registran manualmente durante la atención a los feligreses. Posteriormente, estas notas deben ser transcritas por secretaría, quien se encarga de organizarlas e imprimirlas para su adecuada mención durante las celebraciones litúrgicas.

Las intenciones se programan según disponibilidad de horarios y espacios, y pueden celebrarse tanto en la iglesia de San Miguel como en la capilla de Santa Rosa, atendiendo a la solicitud del feligrés y la programación parroquial.

Figura 21

Diagrama de actividades del proceso de registro de intenciones



Nota: Muestra el flujo mediante el cual un feligrés solicita registrar una intención, incluyendo lugar, fecha y motivo, siendo ingresada por secretaria. Elaboración propia mediante Enterprise Architect.

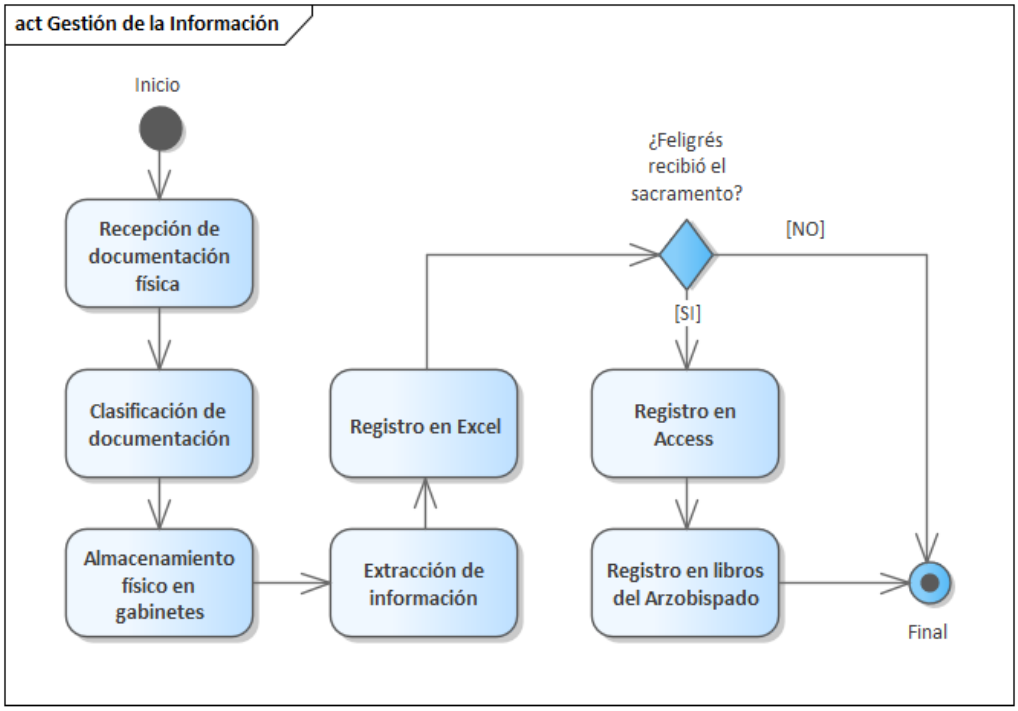
3.2.2. Observación de las Actividades

Para el manejo de registros sacramentales, se utiliza Microsoft Access como base de datos principal. Cada tipo de sacramento cuenta con su propio archivo de Access, y dentro de estos, los registros se encuentran divididos por rangos de años. De manera complementaria, se utilizan hojas de Excel para elaborar listas de inscritos, llevar controles temporales y generar informes requeridos por el Arzobispado, tanto sobre sacramentos celebrados como sobre el manejo contable.

Los documentos físicos entregados por los feligreses son almacenados en gabinetes, organizados en carpetas y sobres clasificados por sacramento y año. A partir de estos documentos, la secretaría extrae la información que se registra primero en Excel y una vez recibido el sacramento, en Access, como se representa en la Figura 22.

Figura 22

Diagrama de actividades del flujo de gestión de información sacramental



Nota: La figura muestra cómo se gestiona la información de los feligrese desde la recepción hasta el registro oficial en Access. Elaboración propia mediante Enterprise Architect.

Es importante destacar que una vez completado el registro en Access, la información se transcribe también en los libros físicos y oficiales proporcionados por el Arzobispado, los cuales garantizan la validez canónica de los sacramentos y actúan como un respaldo institucional.

3.2.2.1. Limitaciones del Sistema Actual

A partir de la observación del funcionamiento actual de la parroquia y de las herramientas utilizadas, se recopilaron evidencias visuales que se encuentran documentadas en los Anexos D y E, las cuales respaldan las limitaciones y los problemas detectados.

a) Fragmentación y dispersión de la información

Cada sacramento tiene su propio archivo en Access, lo que genera dispersión de la información y dificulta el seguimiento de un feligrés a través de los diferentes sacramentos. Como se muestra en la Figura 23, dentro de cada archivo los registros están organizados en múltiples tablas distribuidas por rangos de años, con numerosas copias y versiones de los mismos registros.

Figura 23

Estructura fragmentada de tablas en archivos Access por sacramento



Nota: La figura muestra múltiples tablas de Primera Comunión y Confirmación con nombres similares, versiones y copias acumuladas a lo largo del tiempo. Evidenciando la fragmentación y dispersión de la información en los registros parroquiales. Elaboración propia.

b) Localización de registros deficiente

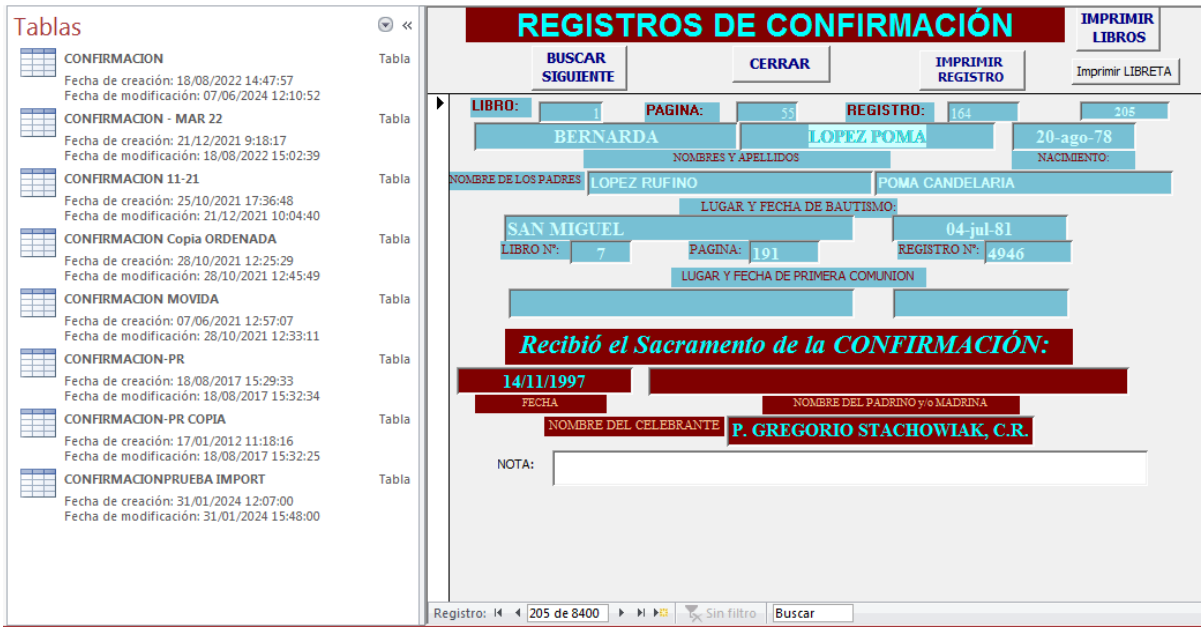
Actualmente, el único criterio de búsqueda para los registros de un feligrés es el apellido paterno. Y dado que la información está dividida en múltiples tablas según rangos de años, se debe seleccionar la tabla en la cual se realizará la búsqueda mediante botones específicos para cada rango. Además, si el apellido paterno ha sido registrado incorrectamente, como ya ha ocurrido, es necesario revisar varias tablas manualmente.

c) Interfaz poco intuitiva

Los formularios de Access utilizados no cuentan con mecanismos de validación, menús guiados ni ayudas visuales, como se observa en la Figura 24, lo que dificulta la operación para los usuarios y aumenta la probabilidad de errores al ingresar o actualizar registros. la interfaz poco intuitiva obliga a los usuarios a depender de su experiencia personal o de instrucciones externas, incrementando el tiempo necesario para completar las tareas y reduciendo la eficiencia en la gestión de los registros sacramentales.

Figura 24

Formulario de registro de Confirmación en Access



Nota: La figura muestra el formulario de registro sacramental de Confirmación utilizado actualmente en la parroquia. La interfaz presenta una estética y funcionalidad obsoletas, evidentes en la disposición de los elementos, los colores y el tamaño de la tipografía, lo que limita su usabilidad. Elaboración propia.

d) Acceso sin control de usuarios

El manejo de los registros no cuenta con niveles de acceso diferenciado, por lo que todos los usuarios tienen permisos completos sobre los archivos de Access. Esto ha derivado en errores modificaciones accidentales y perdida de información, afectando directamente a la integridad de los registros.

3.2.3. Identificación de la Documentación Recepcionada

La parroquia San Miguel Arcángel recibe y gestiona documentación destinada a garantizar la válida y lícita celebración de los sacramentos, especialmente los de iniciación cristiana, cuya interconexión está expresamente establecida por el Derecho Canónico (Holy See, 1983). La parroquia maneja diversos tipos de documentos relacionados con los procesos administrativos y litúrgicos, que permiten llevar un control adecuado sobre la formación sacramental, la emisión de certificados y la celebración de los sacramentos.

La documentación requerida varía según el sacramento y cumple funciones específicas, pudiendo clasificarse en dos categorías:

- Documentación solicitada por la parroquia, destinada al registro de los datos personales del feligrés, como el certificado de nacimiento o la cédula de identidad.

- Documentación exigida por la Iglesia Católica, son requisito previo para comprobar que el feligrés ha recibido los sacramentos anteriores correspondientes, tales como el certificado de bautismo, primera comunión, confirmación o matrimonio.

En este sentido, cada sacramento tiene requisitos específicos que deben ser presentados al momento de la inscripción. A continuación, se presenta la Tabla 2 que detalla la documentación necesaria para cada sacramento.

Tabla 2

Documentación requerida por sacramento

Sacramento	Documentación requerida
Bautismo	Certificado de nacimiento del feligrés. Certificado de confirmación o matrimonio de los padrinos
Primera Comunión	Certificado de bautismo del feligrés.
Confirmación	Certificado de bautismo y de primera comunión del feligrés. Certificado de confirmación o matrimonio de los padrinos.
Matrimonio	Certificado de bautismo, primera comunión, confirmación y fotocopias de cédula de identidad de los novios. Fotocopias de cedula de identidad de testigos. Certificado de matrimonio de los padrinos.

Nota: La tabla presenta la documentación que los feligreses deben entregar al inscribirse en los cursos sacramentales, organizada por sacramento y de acuerdo con lo establecido por el Derecho Canónico. Elaboración propia.

Así mismo, se debe tomar en cuenta que la documentación recibida presenta una alta variabilidad en cuanto a estructura, formato y calidad, sobre todo en la documentación exigida por la Iglesia Católica. Al no existir un modelo unificado para la emisión de certificados de bautismo, primera comunión, confirmación y matrimonio, y dado que estos son emitidos por diferentes parroquias alrededor de Bolivia, existen notables diferencias en su presentación. A tal punto, que en algunos casos los certificados combinan secciones impresas con anotaciones manuales.

3.2.4. Identificación de Información Extraída

Cada sacramento exige la recolección de ciertos información clave, que se extraen directamente de la documentación presentada al momento de la inscripción. La Tabla 3 muestra, por cada sacramento la información que se considera necesaria.

Tabla 3

Campos de información relevantes por sacramento

Sacramento	Información requerida
Bautismo	Nombres completos del feligrés, de los padres y de los padrinos; fecha de nacimiento; lugar de nacimiento (departamento, provincia, localidad); número de oficialía, número de libro y número de partida.
Primera Comunión	Nombres completos del feligrés y de los padres; lugar y fecha de bautismo; fecha de nacimiento.
Confirmación	Nombres completos del feligrés, de los padres y de los padrinos; fecha de nacimiento; lugar y fecha de bautismo; número de libro, página y partida del registro bautismal; lugar y fecha de primera comunión.
Matrimonio	Nombres completos de los novios, de los padres, de los padrinos y de los testigos; lugar y fecha de bautismo.

Nota: La tabla resume los campos de datos más relevantes que se extraen de los documentos presentados para cada sacramento, según los requisitos indicados previamente en la Tabla 2. Elaboración propia.

3.2.5. Requerimientos de Usuario

Los requerimientos del usuario son las necesidades y expectativas del personal que utilizará el sistema, fueron recabados a partir de entrevistas realizadas al personal de la parroquia San Miguel Arcángel (ver Anexo F). En la Tabla 4 se presentan los requerimientos identificados y agrupados por módulos.

Tabla 4

Requerimientos del usuario

Módulo	Id	Descripción
Gestión de Usuarios	UR-01	El sistema debe permitir el acceso únicamente al personal con credenciales válidas y permitir la gestión de usuarios.
	UR-02	El sistema debe permitir asignar y modificar roles a los usuarios. Cada rol tendrá permisos específicos que determinan el acceso a funcionalidades.
Gestión de Feligreses	UR-03	Permitir el registro manual de feligreses incluyendo datos personales y documentación adjunta. Debe incluir validaciones de campos obligatorios y permitir editar registros según permisos de rol.
	UR-04	Automatizar la identificación y extracción de datos de documentos escaneados (certificados de nacimiento y cédulas de identidad).
Gestión de Cursos	UR-06	Permitir crear, editar, eliminar y consultar cursos sacramentales, incluyendo descripción, horarios, costo y sacramento correspondiente, así como la consulta independiente de cada curso.
	UR-07	Permitir crear y modificar sesiones de cursos con tema, fecha, horario y lugar de cada sesión.
Gestión de Inscripciones	UR-08	Permitir inscribir feligreses a uno o varios cursos, consultar los cursos disponibles y los feligreses inscritos.
	UR-09	Registrar pagos, asistencia de los feligreses durante los cursos, y permitir la visualización de la información relevante de cada participante.
Gestión de Celebraciones Sacramentales	UR-10	Permitir establecer fecha, hora y lugar de celebraciones, diferenciar entre comunitarias y particulares, reprogramar según disponibilidad de espacios.
	UR-11	Registrar y mantener actualizado un calendario de celebraciones sacramentales incluyendo fecha, hora y lugar
Gestión de Registros Sacramentales	UR-12	Migrar información de inscripciones a registros oficiales solo para feligreses que culminaron el curso.
	UR-13	Generar certificados de sacramentos según formatos oficiales, previsualizar antes de imprimir, enviar a impresora seleccionada.

Nota: La tabla presenta los requerimientos del usuario identificados a partir de las entrevistas al personal de la parroquia San Miguel Arcángel, reflejando las necesidades y expectativas de los usuarios. Elaboración propia.

3.2.6. Historias de Usuario

Se elaboró un conjunto de historias de usuario a partir de los requerimientos identificados, con el objetivo de guiar el desarrollo del sistema. Estas historias fueron organizadas considerando su prioridad, el esfuerzo requerido y la iteración en la que se planea implementarlas.

- Iteración asignada: Indica en qué iteración se planea implementar cada historia.

- Prioridad: Define la relevancia funcional de cada historia. Se clasifican en tres niveles: alta, media y baja, en función del impacto que tendrían en el funcionamiento general del sistema.

- Puntos estimados: Estimación del esfuerzo necesario para desarrollar la historia, en escala de 0 a 5, donde cada punto equivale aproximadamente a 1 semana de trabajo.

Una vez definidos estos criterios, se presentan las historias de usuario del sistema, cada una acompañada de sus criterios de aceptación.

Tabla 5

Historia de usuario - Acceso al sistema

HU-01: Acceso al sistema	
Usuario: Párroco	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 1
Criterios de aceptación: • El párroco debe poder gestionar las credenciales de acceso al sistema, incluyendo la creación, edición y eliminación de usuarios. • El sistema permite acceder únicamente al personal con credenciales válidas.	

Nota: Esta historia de usuario describe la funcionalidad encargada de garantizar la seguridad y control de acceso al sistema. Elaboración propia.

Tabla 6

Historia de usuario - Gestión de roles y permisos

HU-02: Gestión de roles y permisos	
Usuario: Párroco	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 1
Criterios de aceptación: • El sistema debe permitir asignar y modificar roles a los usuarios • Cada rol tiene permisos específicos que determinan acceso a funcionalidades. • El sistema restringe acciones según el rol asignado.	

Nota: Esta historia de usuario implementa el control de acceso basado en roles. Elaboración propia.

Tabla 7

Historia de usuario - Registro de feligreses

HU-03: Registro de feligreses	
Usuario: Secretaría	**Prioridad:** Alta
Puntos estimados: 1.5	**Iteración asignada:** 2
Criterios de aceptación: • Permite ingresar manualmente los datos personales de los feligreses (nombres, fecha y lugar de nacimiento, libro, partida, oficialía). • Incluye validaciones de campos obligatorios y formatos en los formularios de registro. • Permite editar registros existentes según permisos de rol. • Permite asociar documentación adjunta	

Nota: Esta historia de usuario encargada de del registro y gestión de la información personal de los feligreses. Elaboración propia.

Tabla 8

Historia de usuario - Integración de redes neuronales

HU-04: Integración de redes neuronales	
Usuario: Secretaría / Párroco	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 2
Criterios de aceptación: • Identifica automáticamente el tipo de documento escaneado (certificados de nacimiento, cédulas de identidad u otros). • Extrae automáticamente la información relevante mediante los modelos de detección y extracción. • Los datos extraídos se muestran para previsualización y corrección manual antes del registro. • No realiza inscripción directa, solo sugiere los datos para registro.	

Nota: Esta historia de usuario se centra en automatizar la captura de datos de documentos escaneados, asegurando el control humano antes de registrar la información. Elaboración propia.

Tabla 9

Historia de usuario - Integración de servicio de escaneo

HU-05: Integración de servicio de escaneo	
Usuario: Secretaría / Párroco	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 2
Criterios de aceptación: • Permite que el sistema web solicite al servicio externo la operación de escaneo de documentos. • Permite recibir el archivo escaneado desde el servicio externo y almacenarlo en el sistema.	

Nota: Esta historia aborda la integración con hardware externo, necesario porque los navegadores web no pueden interactuar directamente con escáneres o impresoras, asegurando que el sistema pueda manejar documentos físicos. Elaboración propia.

Tabla 10

Historia de usuario - Gestión de cursos sacramentales

HU-06: Gestión de cursos sacramentales	
Usuario: Párroco	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 3
Criterios de aceptación: • Permite crear, editar y eliminar cursos sacramentales. • Cada curso incluye descripción, horarios, costo y sacramento correspondiente. • Permite consultar la lista completa de cursos y sus detalles. • Permite consultar de manera independiente la información de cada curso.	

Nota: Esta historia de usuario permite planificar y administrar la formación sacramental, hacia los feligreses. Elaboración propia.

Tabla 11

Historia de usuario - Programación de sesiones de cursos

HU-07: Programación de sesiones de cursos	
Usuario: Párroco / Secretaría	**Prioridad:** Media
Puntos estimados: 1	**Iteración asignada:** 3
Criterios de aceptación: • Permite crear sesiones junto con el tema a tratar, fecha, horarios y lugar de cada sesión de curso. • Permite modificar la programación de sesiones existentes.	

Nota: Esta historia de usuario garantiza la correcta planificación de los cursos, asegurando que los contenidos se impartan de manera completa y ordenada. Elaboración propia.

Tabla 12

Historia de usuario - Inscripción a cursos

HU-08: Inscripción a cursos	
Usuario: Párroco / Secretaría / Catequista	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 4
Criterios de aceptación: • Permite inscribir feligreses a uno o varios cursos de manera simultánea. • Proporciona una vista de los cursos disponibles a inscripción. • Permite consultar los feligreses inscritos en cada curso. • Permite cancelar inscripciones según los permisos de rol.	

Nota: Esta historia de usuario describe la funcionalidad encargada de administrar las inscripciones de los feligreses en los cursos sacramentales, así como registrar la asistencia a los mismos. Elaboración propia.

Tabla 13

Historia de usuario – Monitoreo de feligreses inscritos

HU-09: Monitoreo de feligreses inscritos	
Usuario: Secretaría / Párroco / Catequista	**Prioridad:** Media
Puntos estimados: 1	**Iteración asignada:** 4
Criterios de aceptación: • Permite registrar el monto pagado por los feligreses. • Permite registrar y actualizar la asistencia de los feligreses durante los cursos. • Permite visualizar la información más relevante de la documentación proporcionada por los feligreses.	

Nota: Esta historia de usuario permite supervisar la documentación, los pagos y la asistencia de los feligreses después de su inscripción, asegurando un seguimiento completo y actualizado de cada participante. Elaboración propia.

Tabla 14

Historia de usuario - Creación de registros sacramentales

HU-10: Creación de registros sacramentales	
Usuario: Secretaría / Párroco	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 6
Criterios de aceptación: • Permite migrar la información registrada en la inscripción a un registro oficial del sacramento únicamente a los feligreses que culminaron el curso satisfactoriamente. • Los registros sacramentales creados deberán mantener correlación con los libros físicos que exige el Arzobispado. • El sistema debe permitir agregar observaciones en caso de ser necesario para los sacramentos de Bautismo y Matrimonio. • El sistema debe permitir la búsqueda de registros en base a distintos criterios.	

Nota: Esta historia de usuario controla qué registros se vuelven oficiales, asegurando que solo feligreses que completaron la formación aparezcan en los libros sacramentales. Elaboración propia.

Tabla 15

Historia de usuario - Programación de celebraciones sacramentales

HU-11: - Programación de celebraciones sacramentales	
Usuario: Secretaría / Párroco / Sacerdote	**Prioridad:** Media
Puntos estimados: 1	**Iteración asignada:** 5
Criterios de aceptación: • Permite establecer fecha, hora y lugar de cada celebración sacramental. • Permite diferenciar celebraciones comunitarias (Primera Comunión, Confirmación) y particulares (Matrimonio, Bautizo). • Permite modificar y reprogramar celebraciones según disponibilidad de espacios. • Permite asociar celebraciones a cursos correspondientes.	

Nota: Esta historia de usuario organiza y planifica las celebraciones sacramentales para asegurar logística y cumplimiento litúrgico. Elaboración propia.

Tabla 16

Historia de usuario - Emisión de certificados sacramentales

HU-12: Emisión de certificados sacramentales	
Usuario: Secretaría / Párroco	**Prioridad:** Alta
Puntos estimados: 1	**Iteración asignada:** 6
Criterios de aceptación: • Permite generar certificados de sacramentos según formatos oficiales de la parroquia. • Permite previsualizar certificados antes de imprimir. • Permite enviar certificados a la impresora seleccionada. • Establece la fecha y responsable de la emisión.	

Nota: Esta historia de usuario asegura la correcta emisión de certificados oficiales de cada sacramento. Elaboración propia.

Tabla 17

Historia de usuario - Monitoreo de celebraciones sacramentales

HU-13: Monitoreo de celebraciones sacramentales	
Usuario: Secretaría / Párroco / Sacerdote	**Prioridad:** Baja
Puntos estimados: 0.5	**Iteración asignada:** 5
Criterios de aceptación: • Permite registrar las celebraciones sacramentales, incluyendo fecha, hora y lugar. • Permite visualizar de manera organizada el calendario de celebraciones próximas. • Permite actualizar la información de las celebraciones en caso de cambios.	

Nota: Esta historia de usuario permite mantener un control actualizado y centralizado de las celebraciones sacramentales, asegurando que la información sobre fechas, lugares y participantes esté siempre disponible para la planificación y coordinación de la parroquia. Elaboración propia.

3.2.7. Planificación de Iteraciones

Tabla 18

Plan de Iteraciones

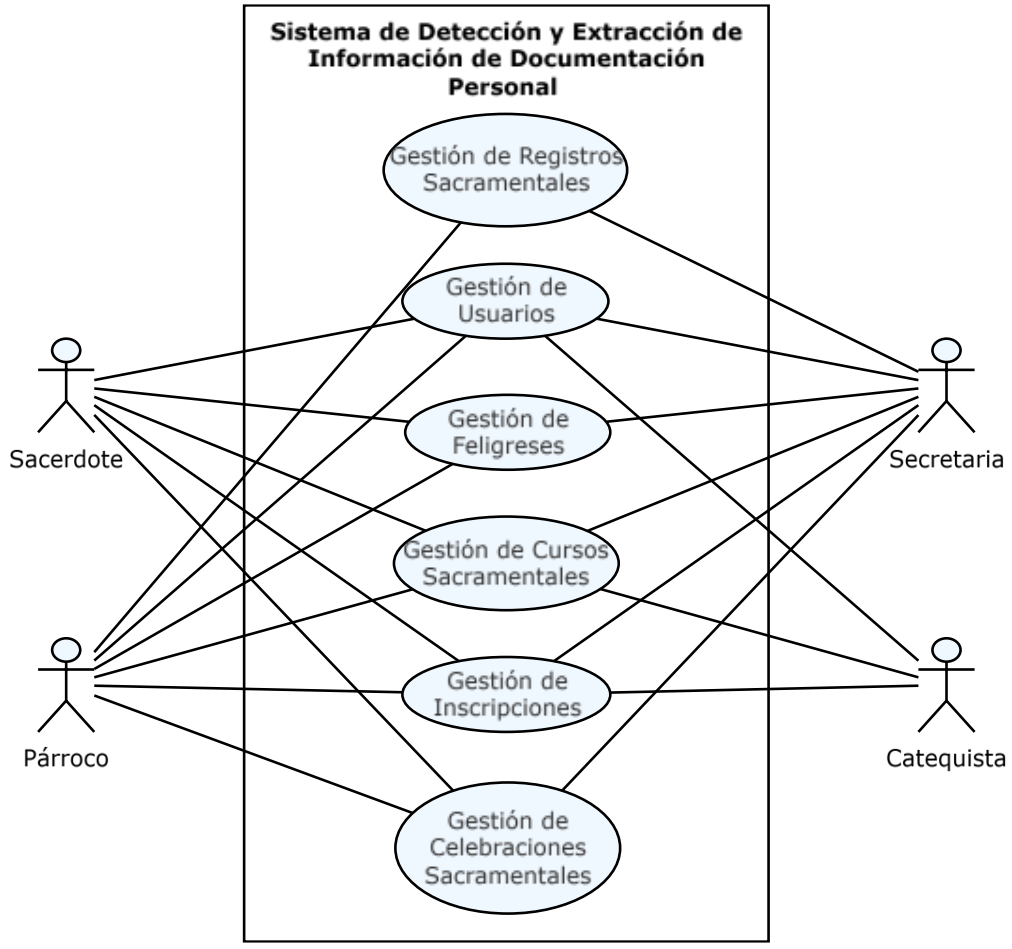
Iter.	Módulo	Historias	Duración aproximada	Fecha de inicio	Fecha de fin
1	Gestión de Usuarios	HU-01 HU-02	2 semanas	12/08/2025	23/08/2025
2	Gestión de Feligreses	HU-03 HU-04 HU-05	3 semanas	24/08/2025	13/09/2025
3	Gestión de Cursos	HU-06 HU-07	2 semanas	14/09/2025	27/09/2025
4	Gestión de Inscripciones	HU-08 HU-09	2 semanas	28/09/2025	11/10/2025
5	Gestión de Celebraciones Sacramentales	HU-11 HU-13	1 semana	12/10/2025	18/10/2025
6	Gestión de Registros Sacramentales	HU-10 HU-12	1.5 semanas	19/10/2025	01/12/2025

Nota: La tabla muestra cada iteración del proyecto junto a sus historias de usuario y las fechas de inicio y fin de su implementación. Elaboración propia.

Durante la etapa de planificación de la iteración se definieron los seis módulos funcionales principales del sistema a partir de los requerimientos y de las historias de usuario. Estos módulos corresponden a Gestión de Registros Sacramentales, Gestión de Usuarios, Gestión de Feligreses, Gestión de Cursos Sacramentales, Gestión de Inscripciones y Gestión de Celebraciones Sacramentales, y se presentan en la Figura 25 mediante el diagrama de casos de uso de alto nivel. A ambos lados del diagrama se ubican los cuatro actores del sistema, que son el Sacerdote, la Secretaria, el Párroco y el Catequista, cada uno con sus respectivas interacciones dentro del sistema.

Figura 25

Diagrama de Casos de Uso de Alto Nivel



Nota: La figura muestra los seis módulos funcionales del sistema y cómo interactúan con los distintos actores. Elaboración propia mediante draw.io.

3.3. CONSTRUCCIÓN DE DATASETS

Para el entrenamiento de una red neuronal es indispensable contar con un conjunto de datos (dataset) que permita al modelo aprender y especializarse en una tarea específica. En el presente proyecto de extracción automática de información de documentos personales fue necesaria la construcción de dos datasets, cada uno con un propósito particular, dado que la propuesta se divide en dos tareas o procesos principales.

El primer dataset está enfocado en la tarea de detección de la información de interés dentro de los documentos (como la oficialía, fecha de nacimiento, nombre del padre y nombre de la madre). Para ello, se utilizan imágenes completas de los documentos junto con anotaciones en forma de regiones delimitadas (bounding boxes) y su respectiva clase. De esta manera, la red neuronal puede aprender a identificar en qué lugar se encuentra cada campo relevante y qué tipo de información contiene.

El segundo dataset corresponde a la tarea de extracción mediante reconocimiento óptico de caracteres (OCR). Una vez localizados los campos de interés, es necesario transcribir su contenido textual. Para lograrlo, este dataset debe estar conformado por fragmentos de texto recortados a partir de los documentos, cada uno acompañado de su etiqueta textual (ground truth). Con esta información, una red neuronal puede aprender a reconocer y transcribir de manera precisa letras, números y otros caracteres.

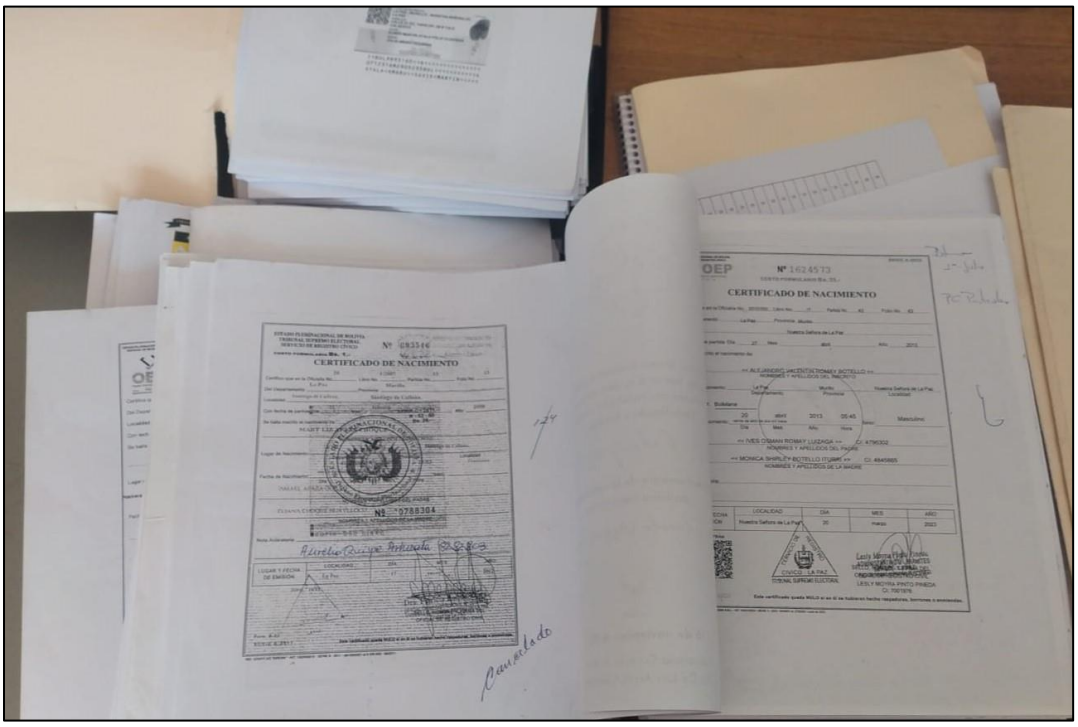
3.3.1. Dataset de Documentos para la Detección y Clasificación de Texto

3.3.1.1. Recopilación y escaneo de documentación

Para la construcción de ambos datasets, la Parroquia San Miguel Arcángel proporcionó acceso a la documentación física con la que contaba, entre ella fotocopias de certificados de nacimiento y cédulas de identidad. A partir de esta documentación fue necesario realizar un proceso de selección, ya que no todos los ejemplares eran los más ideales. Como se observa en la Figura 26, algunos documentos presentaban observaciones anotadas con marcadores y otros correspondían a versiones antiguas de certificados de nacimiento o cédulas de identidad que no coincidían con el formato actual en el que se emite esta documentación.

Figura 26

Recolección de documentación



Nota: La figura muestra la documentación recolectada y revisada para su selección. Se puede observar que algunos certificados de nacimiento presentan anotaciones relacionadas con inscripciones y montos cancelados para los cursos. Elaboración propia.

Una vez seleccionados los documentos, estos fueron escaneados empleando el escáner disponible en la misma parroquia. Esto con el fin de mantener la uniformidad en las características de digitalización, considerando que otros escáneres pueden producir variaciones con respecto al contraste. En la Tabla 19 se muestran los resultados de este proceso.

Tabla 19

Documentación escaneada

Tipo de documento	Cantidad	Formato	Dimensiones (px)
Certificado de Nacimiento	129	JPG	2550 × 3300
Cédula de Identidad	94	JPG	2550 × 3300

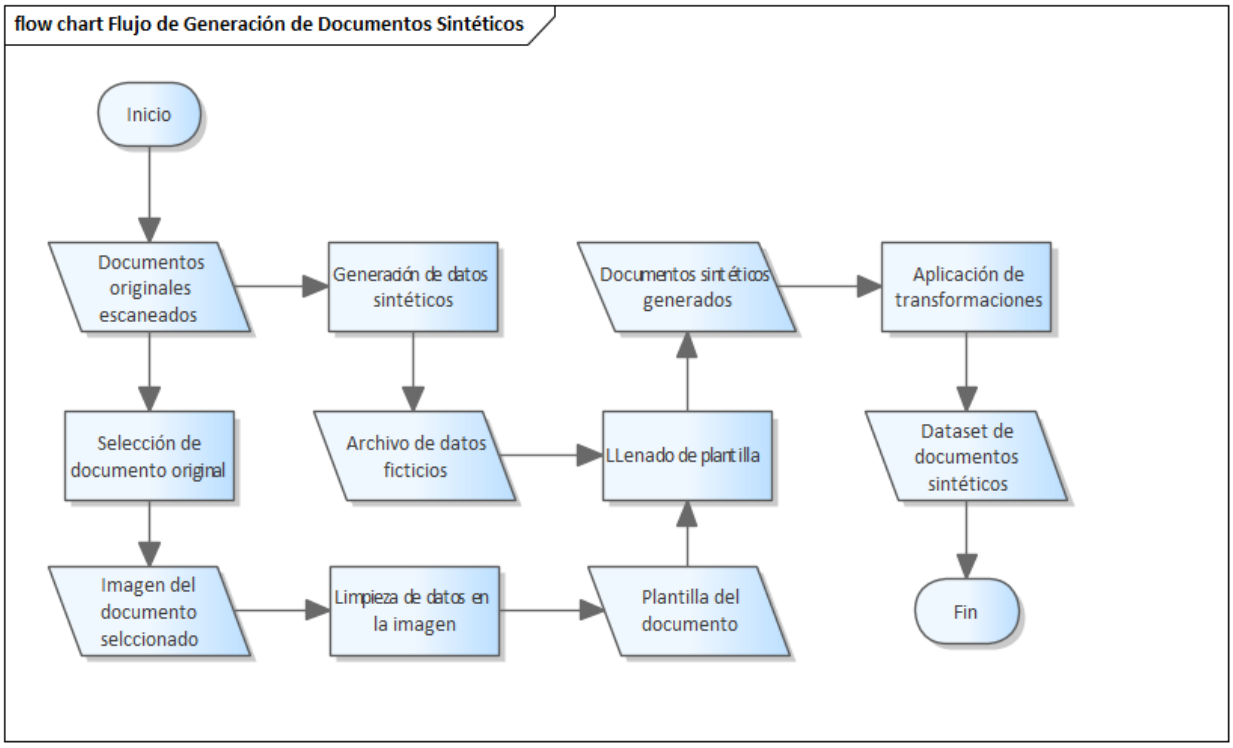
Nota: La tabla muestra la documentación escaneada que conforma el punto de partida para la construcción de los datasets. Elaboración propia.

3.3.1.2. Generación de documentación sintética

Tras realizar la recolección de documentos, se evidenció la poca disponibilidad de ejemplares adecuados para la construcción de los datasets. Ante este obstáculo, se decidió realizar la generación de documentación sintética tanto a certificados de nacimiento como a cédulas de identidad con el propósito de aumentar nuestros datos. En la Figura 27 se ilustra el flujo de trabajo para la generación de documentación sintética.

Figura 27

Flujo de generación de documentos sintéticos



Nota: La figura muestra las etapas del proceso de generación de documentos sintéticos, desde la recolección de documentos reales, hasta la creación del dataset. Elaboración propia mediante Enterprise Architect.

a) Selección de documentos originales

Para la generación de nuevos documentos fue necesario contar con ejemplares en buenas condiciones tanto de certificados de nacimiento como de cédulas de identidad, un ejemplar de cada tipo de documento. Esto se debe a que la creación de documentación sintética debía partir

de documentos reales que serían transformados en plantillas base. Los criterios de selección fueron los siguientes:

- Que el documento estuviera recto y sin inclinación, ya que una orientación incorrecta dificultaría el llenado posterior de información en la plantilla.

- Que el diseño y el fondo del documento fueran claramente visibles y no presentaran deterioro, pues estas características podrían afectar los procesos de reconocimiento de texto.

Si bien algunos defectos como desgaste o pérdida de color en los documentos son condiciones presentes en documentos reales, se decidió optar por el uso de ejemplares en buenas condiciones para facilitar la generación inicial de plantillas.

b) Generación de datos sintéticos

Dado que cada tipo de documento contiene información específica, fue necesario crear dos archivos separados que incluyeran los datos a insertar en las plantillas. Estos datos se generaron procurando que fueran lo más realistas y variados posible, con el fin de garantizar la diversidad necesaria para el entrenamiento de las redes neuronales. Como se muestra en la Figura 28, se optó por utilizar archivos Excel para almacenar esta información, debido a su fácil lectura y manipulación en el entorno de Jupyter mediante Python.

Figura 28

Datos ficticios de cédulas de identidad

```
datos_cedulas = pd.read_excel("../Generador cedulas/datos_ci.xlsx")
datos_cedulas.head()
```

	serie	seccion	numero	nombres	apellidos	fecha	emision	expiracion	firma	lugar	domicilio	ocupacion	estado
0	48723	16971	37004906	NILDA LEONEL	NAVARRO BONIFAZ	08/11/2004	04/01/1973	INDEFINIDO	CALATAYUD BONIFAZ	LA PAZ - MURILLO - PALCA	CALLE 8 # 33, MIRAFLORES	ING. CIVIL	SOLTERO
1	56969	31343	90559028	BRYAN FELIX	MAMANI CHOQUE	09/08/2010	29/05/1968	13/07/1975	MAMANI CHOQUE	LA PAZ - LOS ANDES - LURIBAY	CALLE 8 # 33, MIRAFLORES	CHOFER	CASADA
2	15892	48159	68101798	MARTINA	VARGAS CARI	16/07/1987	31/12/2008	25/10/1972	VARGAS CARI	LA PAZ - LOS ANDES - LURIBAY	AV. ARCE # 1234, SAN JORGE	JUBILADO	CASADO

Nota: La figura muestra la estructura y el contenido del archivo datos_ci.xlsx, el cual fue utilizado para generar documentación sintética de cédulas de identidad. Elaboración propia mediante Jupyter Notebook.

c) Limpieza de datos de las imagenes

Una vez seleccionados los ejemplares de cada documento, se procedió a su limpieza de sus imágenes con el fin de crear las plantillas base. Para ello se utilizó el software de edición GIMP, mediante el cual se eliminó el contenido textual y los datos sensibles de los documentos originales. Se prestó especial cuidado al remover el texto, con el objetivo de preservar la mayor cantidad de detalles posibles, como los patrones de fondo y otros elementos de diseño característicos del documento original. De esta manera se obtuvieron plantillas limpias que conservan la estructura y el diseño del documento.

d) Llenado de plantillas

Una vez obtenidas las plantillas, se desarrolló un código en Python para realizar el llenado masivo y automático de las mismas, utilizando los archivos de información sintética. Para este proceso fue necesario definir las coordenadas exactas donde aparecería cada dato, así como el tamaño y la tipografía del texto, respetando los márgenes, la alineación y la jerarquía visual del formato original. En la Figura 29 se muestra una de las funciones implementadas para el llenado de la información.

Figura 29

Función de Python para el llenado de plantillas

```python
def dibujar_texto(imagen, persona):
    draw = ImageDraw.Draw(imagen)
    for campo, cfg in coordenadas.items():
        x_base, y = cfg["pos"]
        tamano, alineacion = cfg["size"], cfg.get("align","left")
        fuente = ImageFont.truetype(ruta_fuente, size=tamano)
        texto = str(persona.get(campo, "")) or ""
        if campo in ["padre","madre"] and persona.get(f"ci{1 if campo=='padre' else 2}"):
            texto += f"    CI: {persona.get(f'ci{1 if campo=='padre' else 2}')}"
        if campo=="feligres" and texto: texto = f"<< {texto} >>"
        if texto:
            x = x_base - draw.textlength(texto,font=fuente)/2 if alineacion=="center" else x_base
            draw.text((x, y), texto, font=fuente, fill=color_texto)
    return imagen
```

Nota: La imagen presenta un fragmento de código Python de la función dibujar_texto, la cual se encarga de insertar de forma masiva y automática datos ficticios en plantillas de documentos, ajustando su posición, tamaño, fuente y alineación según las coordenadas predefinidas. Elaboración propia mediante Jupyter Notebook.

e) Aplicación de transformaciones

Dado que se utilizó una única plantilla por tipo de documento para generar todos los ejemplares sintéticos, fue necesario aplicar diversas transformaciones a las imágenes con el fin de simular condiciones reales de escaneo. Esto se debe a que, en los datos generados inicialmente, los documentos siempre aparecían en la misma posición y en condiciones demasiado ideales, lo cual no representa escenarios reales.

Por tal motivo, se consideraron aspectos como variaciones en la orientación del documento y la simulación de desgaste propio del papel, entre otros. En la Tabla 20 se detallan las

Tabla 20

Transformaciones aplicadas

Transformación	Descripción
Contraste	Se ajustó el contraste de la imagen dentro del rango 0.7 a 1.3 para simular variaciones de iluminación.
Brillo	Se redujo el brillo dentro del rango 0.7 a 1.0 para simular documentos escaneados con poca luz o sombra.
Rotación	Se realizaron rotaciones aleatorias entre -1° y 1° para simular ligeras inclinaciones durante el escaneo.
Desplazamiento	Se aplicaron desplazamientos en los ejes X e Y dentro del rango de -150 a 150 píxeles para reflejar variaciones de posición del documento.
Manchas	Se agregaron entre 1 y 4 manchas con variaciones de tamaño, opacidad y rotación para simular suciedad o defectos del documento.
Ruido	Se añadió ruido de tipo “luminancia” es decir, puntos blancos y negros granulados en la imagen, para simular suciedad o imperfecciones del escaneo.

Nota: La tabla muestra las transformaciones seleccionadas y aplicadas a los documentos sintéticos con el fin de simular condiciones de escaneo como variaciones de iluminación, inclinación, ruido o suciedad. Elaboración propia.

Para la aplicación de estas transformaciones se desarrolló un código en Python compuesto por varias funciones, cada una encargada de aplicar una transformación específica con parámetros definidos de manera aleatoria sobre las imágenes. Tal como se observa en la Figura 30, estas funciones son invocadas por una función principal, la cual se encarga de ejecutarlas en conjunto.

Figura 30

Función de Python para aplicar variaciones a documentos sintéticos

```python
def aplicar_variaciones(imagen, imagen_manchas):
    img = imagen.copy().convert('RGBA')
    img = img.rotate(random.uniform(*rango_rotacion), resample=Image.BICUBIC,
                     expand=False, fillcolor=(243,242,248,255))
    dx, dy = random.randint(*rango_desplazamiento_x), random.randint(*rango_desplazamiento_y)
    temp = Image.new(img.mode, img.size, (243,242,248,255))
    temp.paste(img, (dx, dy))
    img = temp
    img = ImageEnhance.Contrast(img).enhance(random.uniform(*rango_contraste))
    img = oscurecer_imagen(img)
    img = agregar_ruido_global(img)
    img = aplicar_manchas(img, imagen_manchas)
    return img.convert('RGB') if img.mode=='RGBA' else img
```

Nota: La imagen muestra la función aplicar_variaciones en Python, utilizada para simular condiciones de escaneo mediante transformaciones como rotación, desplazamiento, ajuste de contraste y brillo, además de añadir manchas y ruido. Elaboración propia mediante Jupyter Notebook.

3.3.1.3. Etiquetado de información de interés

Al manejar dos tipos de documentos había que considera que cada uno contiene información de interés que las redes neuronales deben aprender a reconocer y extraer de manera automática. Para ello, se definieron etiquetas que corresponden a los principales datos presentes en las cédulas de identidad y los certificados de nacimiento, como se observa en las Tablas 21 y 22.

Tabla 21

Etiquetas en cedulas de identidad

Etiqueta	Descripción	Ejemplo
feligresC	Nombre completo del feligrés	MAURICIO DAVALOS LOPEZ
nombresC	Nombres del feligrés	MAURICIO
apellidosC	Apellidos del feligrés	DAVALOS LOPEZ
ci	Número de cédula de identidad	6740048

Nota: La tabla muestra las etiquetas definidas en las cédulas de identidad, junto con su descripción y un ejemplo de cómo aparecen en los documentos. Elaboración propia.

Tabla 22

Etiquetas en certificados de nacimiento

Etiqueta	Descripción	Ejemplo
oficialía	Número de oficialía	2689
libro	Número de libro	LIBN-24
partida	Número de partida	30
departamento	Departamento de	La Paz
provincia	Provincia de	Murillo
localidad	Localidad de	Nuestra Señora de La Paz
día	Día de nacimiento	14
mes	Mes de nacimiento	Febrero
año	Año de nacimiento	2002
padre	Nombre completo del padre	SILVERIO DAVALOS SANCHEZ
madre	Nombre completo de la madre	MARCELA IRENE LOPEZ
feligrés	Nombre completo del feligrés	MAURICIO DAVALOS LOPEZ

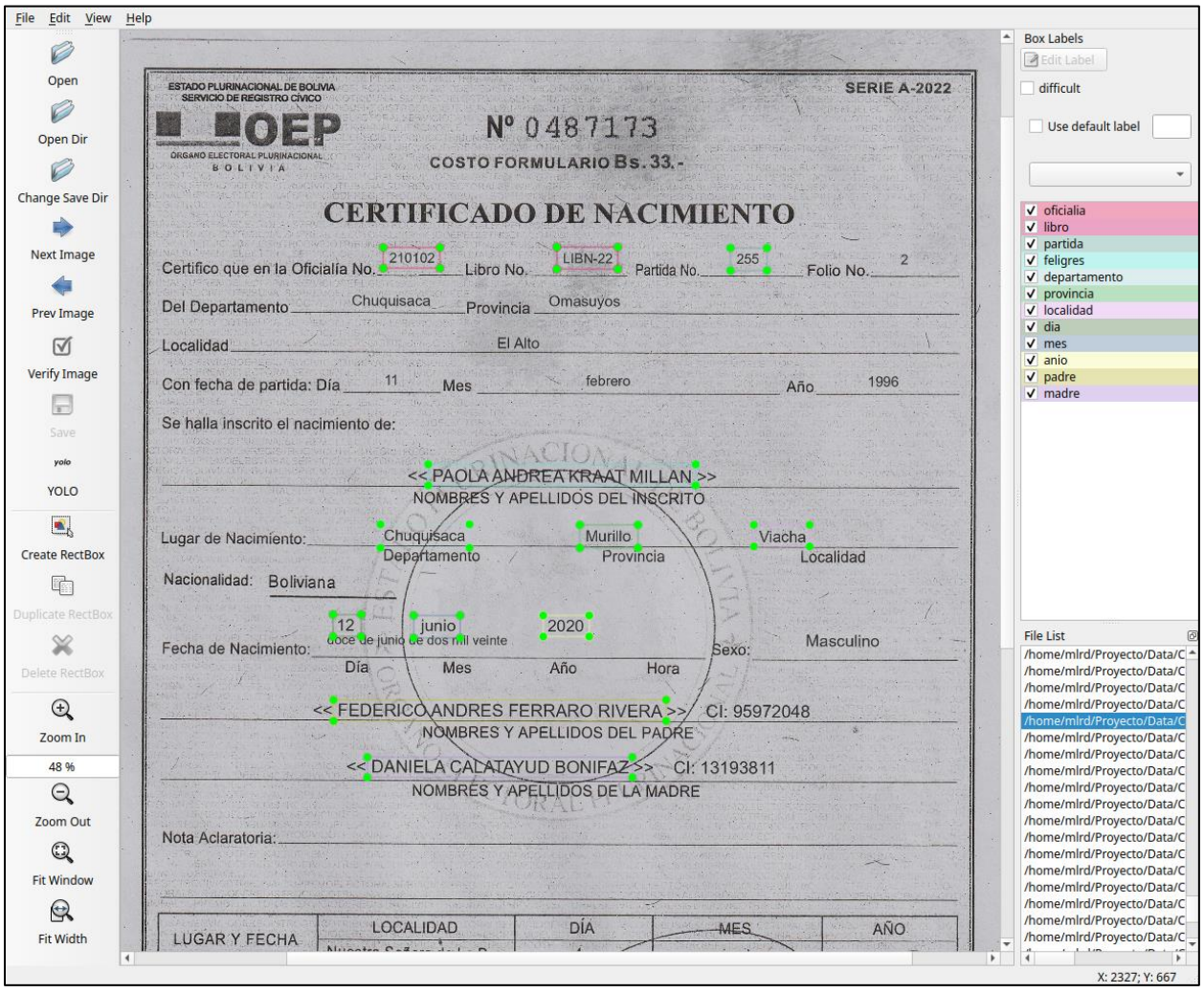
Nota: La tabla muestra las etiquetas definidas en los certificados de nacimiento, junto con su descripción y un ejemplo de cómo aparecen en los documentos. Elaboración propia.

Aunque ambos documentos comparten información, como el nombre del feligrés, se decidió etiquetar esa información de manera diferente en cada tipo de documento. Esto se debe a que la forma en que se presenta la información varía entre la cédula de identidad y el certificado de nacimiento, lo que podría complicar el aprendizaje de las redes neuronales si se usara la misma etiqueta. Con esta elección, se busca facilitar la identificación de patrones y mejorar la precisión en la etapa de reconocimiento automático.

Una vez definidas las etiquetas a utilizar, se utilizó la herramienta “LabelImg” para obtener las anotaciones de todas las imágenes de los distintos documentos que conforman el dataset. En la Figura 31 se muestra el entorno de trabajo durante el proceso de etiquetado, donde se delimitó manualmente la información mediante bounding boxes asociadas a su respectiva etiqueta.

Figura 31

Proceso de etiquetado de campos clave en certificados



Nota: La figura muestra el entorno de trabajo con la herramienta “LabelImg” durante el proceso de anotación en un certificado de nacimiento. Elaboración propia.

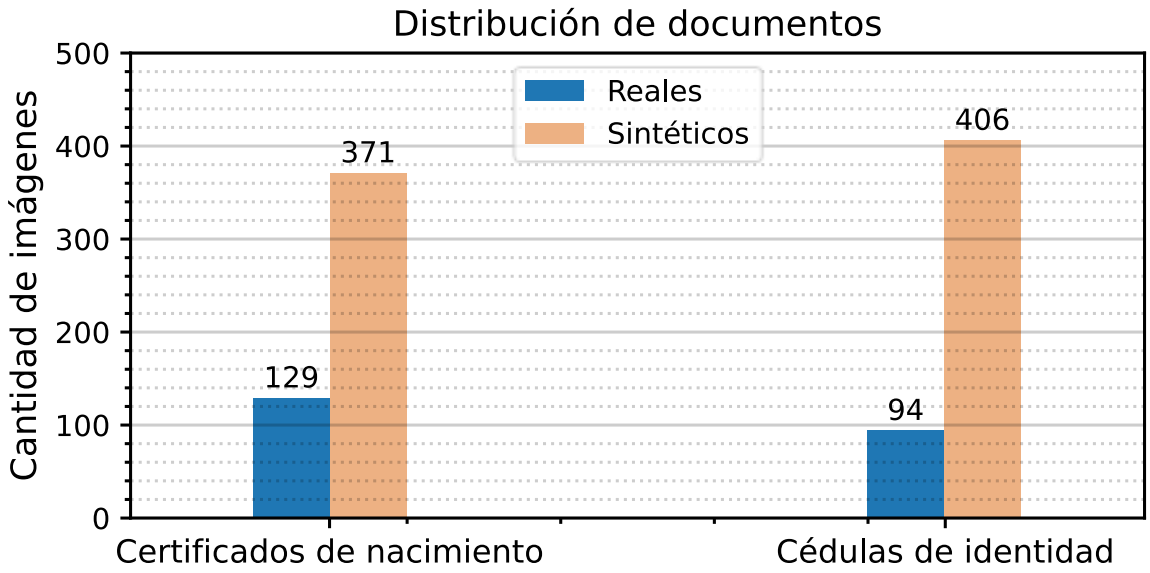
Al finalizar el proceso de etiquetado, las anotaciones de cada imagen de documento se almacenaron en distintos formatos soportados por la herramienta LabelImg, entre ellos XML (Pascal VOC) y YOLO. Adicionalmente, se desarrolló un script para convertir estas anotaciones al formato COCO, con el fin de contar con mayor flexibilidad y compatibilidad en el entrenamiento y evaluación de diferentes arquitecturas de redes neuronales.

3.3.1.4. Distribución del dataset de documentos

Tras la generación de documentación sintética, se logró obtener una cantidad de ejemplares suficiente con sus correspondientes anotaciones, tanto para certificados de nacimiento como para cédulas de identidad. Como se observa en la Figura 32, el dataset final quedó conformado por un conjunto de imágenes más amplio y equilibrado, compuesto por documentos reales y sintéticos.

Figura 32

Histograma de distribución de documentos reales y sintéticos

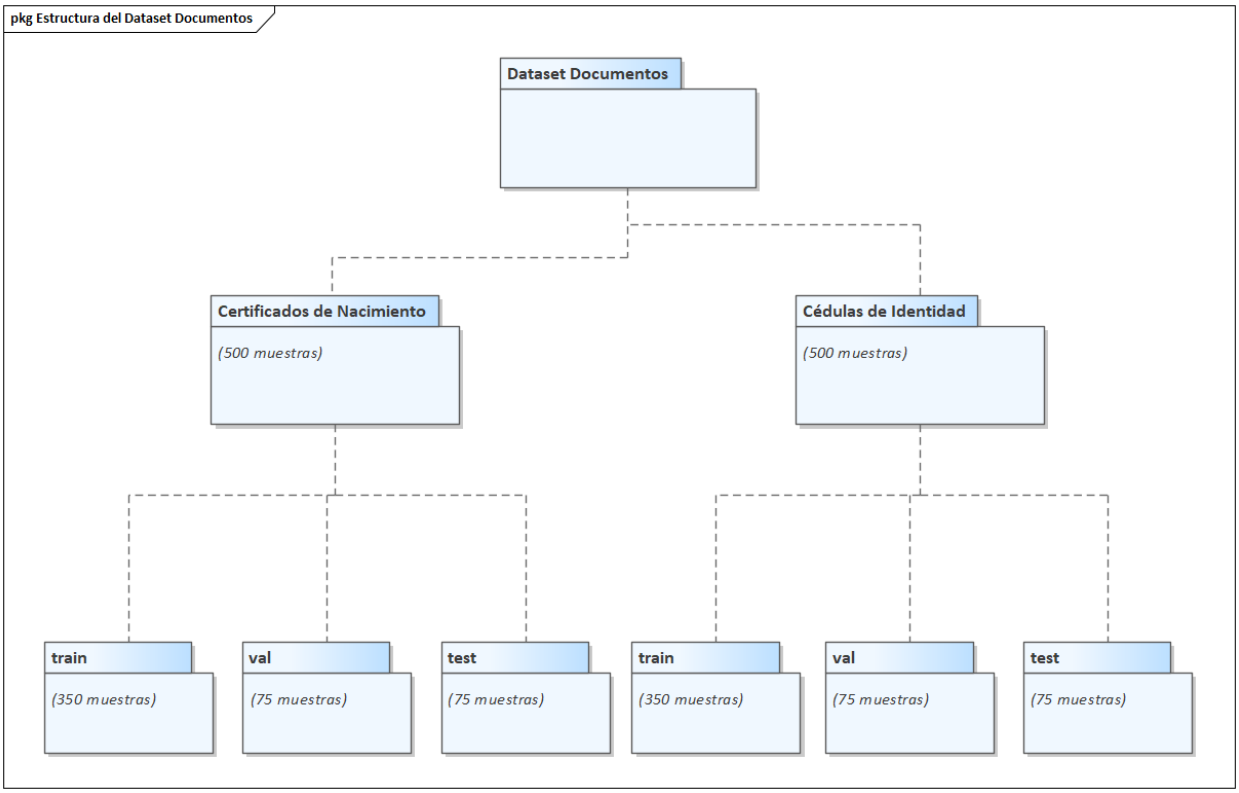


Nota: La figura muestra la proporción de documentos reales frente a documentos generados sintéticamente, incluyendo certificados de nacimiento y cédulas de identidad. Elaboración propia mediante Jupyter Notebook.

Posteriormente, los conjuntos de imágenes de cada tipo de documento fueron divididos en tres subconjuntos: train (entrenamiento), val (validación) y test (pruebas), como se muestra en la Figura 33. La asignación se realizó de la siguiente manera: 70% de las imágenes para entrenamiento, 15% para validación y 15% para prueba.

Figura 33

Estructura del dataset de documentos



Nota: La figura muestra la organización del dataset, donde cada carpeta (train, val, test) contiene tanto las imágenes como sus correspondientes anotaciones. Elaboración propia mediante Enterprise Architect.

3.3.2. Dataset de Fragmentos de Texto para Extracción de Información

3.3.2.1. Extracción de fragmentos de texto

Una vez construido el dataset de documentos, se procedió a construir el dataset para la extracción de información. Para ello, se realizó la extracción de fragmentos de texto a partir de los documentos y sus anotaciones previamente registradas, utilizando la función mostrada en la Figura 34. Esta estrategia permitió disponer de muestras directas y representativas del dominio de nuestro caso de estudio, en lugar de recurrir a datasets públicos con tipografías diversas que podrían dificultar el aprendizaje de la red neuronal y reducir la precisión.

Figura 34

Función en Python para recorte de fragmentos de texto

```python
def recortar_y_guardar(imagen, ruta_txt, carpeta_salida, contador):
    h, w = imagen.shape[:2]
    with open(ruta_txt, 'r') as f:
        for linea in f:
            partes = linea.strip().split()
            if len(partes) != 5:
                continue
            _, x_c, y_c, ancho, alto = map(float, partes)
            x1, y1 = max(int((x_c - ancho/2)*w),0), max(int((y_c - alto/2)*h),0)
            x2, y2 = min(int((x_c + ancho/2)*w),w), min(int((y_c + alto/2)*h),h)
            cv2.imwrite(os.path.join(carpeta_salida, f"IMG_{contador:04d}.jpg"), imagen[y1:y2, x1:x2])
            contador += 1
    return contador
```

Nota: La imagen muestra la función recortar_y_guardar que lee las coordenadas de las anotaciones de un archivo de texto para recortar y guardar fragmentos de texto de una imagen. Elaboración propia mediante Jupyter Notebook.

Tras ejecutar la función sobre la totalidad de las imágenes del dataset de documentos, se obtuvo un conjunto aproximado de 6000 recortes, cada uno correspondiente a fragmentos de texto asociados a la información de interés previamente anotada. Como se observa en la Figura 35, estos recortes presentan variaciones en sus dimensiones, determinadas por la extensión y la disposición del texto dentro de los documentos.

Figura 35

Fragmentos de texto extraídos de documentos



Nota: La figura muestra recortes de texto extraídos de diferentes documentos, junto con los nombres con los que fueron guardados y sus dimensiones en píxeles. Elaboración propia.

3.3.2.2. Preprocesamiento de los fragmentos

Para el entrenamiento de la red neuronal era necesario unificar el tamaño de las imágenes de entrada, ya que los modelos de aprendizaje profundo requieren dimensiones homogéneas. Sin embargo, los recortes de texto presentaban longitudes variables, lo que impedía un redimensionamiento directo sin distorsionar los caracteres. En el caso de una red neuronal para la extracción de texto, es decir un OCR, esta alteración afectaría la legibilidad y podría generar confusiones entre símbolos visualmente similares, como el número “0” y la letra “O”.

Para resolver este desafío, se desarrolló una un código en Python que preserva la proporción original de cada recorte y adapta su tamaño de manera uniforme, en la Figura 36 una de las funciones más importantes de este proceso.

Figura 36

Función en Python para el ajuste de dimensiones

```python
def agregar_padding(imagen_redimensionada):
    forma = tf.shape(imagen_redimensionada)
    pad_alto = IMG_HEIGHT - forma[0]
    pad_ancho = IMG_WIDTH - forma[1]

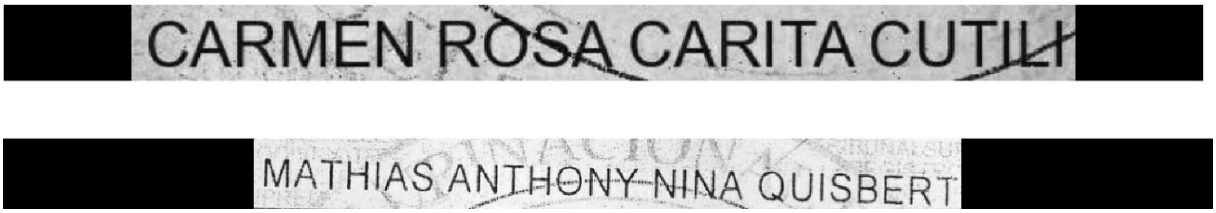
    paddings = [
        [pad_alto // 2, pad_alto - pad_alto // 2],
        [pad_ancho // 2, pad_ancho - pad_ancho // 2],
        [0, 0]
    ]
    imagen_padded = tf.pad(imagen_redimensionada, paddings, "CONSTANT")
    return tf.transpose(imagen_padded, perm=[1, 0, 2])
```

Nota: La imagen muestra la función agregar_padding que toma una imagen redimensionada y le añade un relleno (padding) simétrico. Asegurando que la imagen final tenga el mismo tamaño y forma. Elaboración propia mediante Jupyter Notebook.

Mediante la implementación de esta función, al agregar padding se unificaron los recortes a 1024 píxeles de ancho por 64 de alto, como se observa en la Figura 37. Estas dimensiones se definieron tras analizar las longitudes y alturas de los fragmentos: 1024 px permitía abarcar los anchos máximos sin recortar, y 64 px resultaba suficiente para preservar el detalle de los caracteres.

Figura 37

Recortes de texto con padding y etiquetas asociadas



Nota: La figura muestra dos recortes de texto preprocesados a través de la normalización de tamaño con padding, manteniendo la relación de aspecto. Elaboración propia.

3.3.2.3. Etiquetado del texto contenido

Una vez que se preprocesaron los recortes de texto, cada fragmento debía tener su etiqueta correcta o "ground truth", correspondiente al contenido textual real del recorte, como se observa en la Figura 38. Para ello, se utilizó un archivo .csv que contiene el nombre de cada imagen y su texto asociado, asegurando la correspondencia entre los recortes y su contenido durante el entrenamiento de la red neuronal.

Figura 38

Archivo CSV con recortes de texto y "ground truth"

	FILENAME	IDENTITY
0	IMG_0001.jpg	3
1	IMG_0002.jpg	3-48
2	IMG_0003.jpg	479
3	IMG_0004.jpg	La Paz
4	IMG_0005.jpg	Murillo
5	IMG_0006.jpg	Nuestra Señora de La Paz
6	IMG_0007.jpg	7
7	IMG_0008.jpg	octubre
8	IMG_0009.jpg	1947
9	IMG_0010.jpg	OSCAR GALLARDO BALLESTEROS

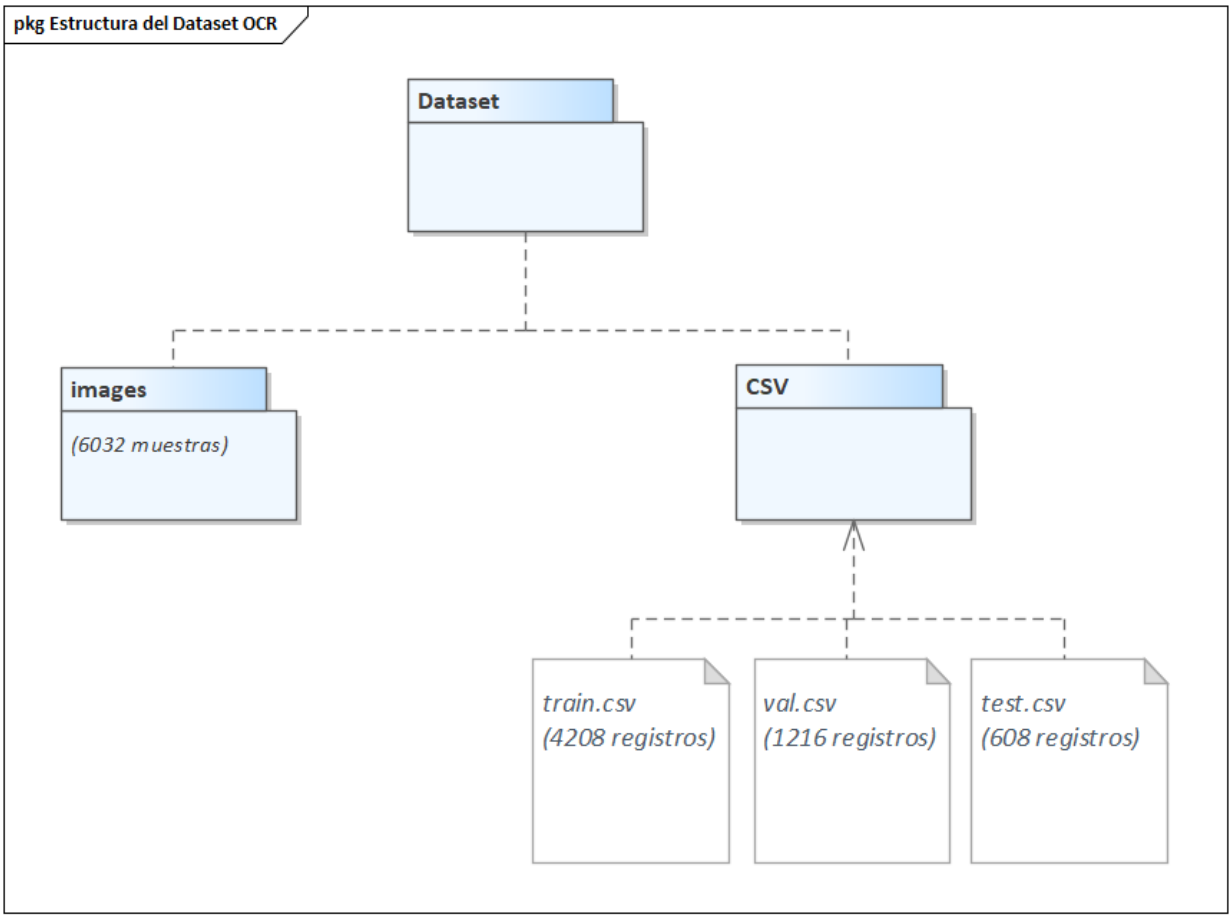
Nota: La figura muestra las columnas FILENAME e IDENTITY, donde se registra el nombre de cada recorte de texto y su contenido textual asociado.

3.3.2.4. Distribución del dataset de fragmentos de texto

Una vez se completaron los procesos de extracción, preprocesamiento y etiquetado de los fragmentos de texto, se organizó el conjunto de imágenes para el dataset en tres subconjuntos: entrenamiento (train), validación (val) y pruebas (test). Como se observa en la Figura 39, a diferencia del dataset de documentos, todas las imágenes se ubican en un único directorio, y la asignación a cada subconjunto se gestiona a través de archivos CSV, que especifican qué imágenes pertenecen a train, val y test.

Figura 39

Estructura del dataset para la extracción de texto



Nota: La figura muestra la organización del dataset de fragmentos de texto, mostrando cómo todas las imágenes se mantienen en un único directorio mientras que los archivos CSV separan las imágenes en subconjuntos de entrenamiento (train), validación (val) y prueba (test). Elaboración propia mediante Enterprise Architect.

En cuanto a la distribución, se asignó un 70 % de las imágenes al entrenamiento, un 20 % a la validación y un 10 % a la prueba. Así mismo, cada archivo CSV mantiene la asociación entre el nombre de cada recorte de texto y su correspondiente etiqueta (ground truth).

3.4. EVALUACIÓN DE ARQUITECTURAS DE REDES NEURONALES

La evaluación de arquitecturas se realizó mediante una comparación en paralelo para las dos tareas del proyecto: detección de regiones con clasificación y reconocimiento óptico de caracteres.

3.4.1. Comparativa de Modelos de Detección y Clasificación de Regiones

3.4.1.1. Selección de Modelos

Las redes neuronales convolucionales (CNN) son la arquitectura más utilizada para la detección de objetos en imágenes, debido a su capacidad para extraer características espaciales jerárquicas y reconocer patrones visuales complejos. Por esta razón, se decidió explorar diferentes arquitecturas dentro de este tipo de red, las cuales resultan especialmente adecuadas para nuestro caso, ya que buscamos detectar regiones en documentos escaneados, donde los campos de interés pueden variar en tamaño y forma.

Para la selección de los modelos a ser sometidos a evaluación, se realizó una revisión de distintas investigaciones sobre el desempeño de diferentes arquitecturas en el contexto del manejo de documentación.

Evaluaciones previas muestran que YOLO ofrece un equilibrio óptimo entre velocidad y precisión, permitiendo la detección de objeto de tamños medianos y grandes, como mencionan Redmon, Santosh, Girshick y Farhadi (2016). Sin embargo, Faster R-CNN ha mostrado un desempeño destacado al entrenarse con múltiples muestras de baja calidad, alcanzando una precisión del 72,42 %, lo que evidencia su eficacia en la detección de textos en condiciones complejas (Pradhan, Pradhan, & Pradhan, 2022).

Estas investigaciones nos permitieron seleccionar las arquitecturas que serían evaluadas en el proyecto, considerando su desempeño reportado para la tarea a realizar. La Tabla 23 presenta las arquitecturas seleccionadas para la detección de regiones y clases.

Tabla 23

Arquitecturas CNN seleccionadas para evaluación

Tarea	Arquitectura	Justificación
Detección de regiones y asignación de clases	Faster R-CNN	Alta precisión en objetos pequeños o superpuestos.
	YOLO	Rápido, con buena precisión en objetos grandes y medianos

Nota: La tabla presenta las arquitecturas seleccionadas para la detección de regiones y asignación de clases, indicando la justificación de cada una. Elaboración propia.

3.4.1.2. Métricas de Evaluación

Para la comparación de los modelos YOLO y Faster R-CNN se seleccionaron métricas relevantes que permiten evaluar tanto la exactitud de las predicciones como la capacidad de los modelos para identificar correctamente las instancias presentes en las imágenes. En la tabla 24 se presentan las métricas consideradas junto a una breve descripción.

Tabla 24

Métricas de evaluación en la detección de regiones y clases

Métrica	Descripción
mAP@0.5	Precisión promedio considerando IoU ≥ 0.5.
mAP@0.5:0.95	Precisión promedio considerando IoU desde 0.5 hasta 0.95.
Precisión	Porcentaje de predicciones correctas respecto al total de predicciones.
Recall	Porcentaje de objetos correctamente detectados sobre el total de objetos.
F1-Score	Métrica que combina Precisión y Recall, proporciona el desempeño general del modelo en términos de exactitud y cobertura.

Nota: La tabla presenta las métricas seleccionadas para medir el desempeño de los modelos YOLO y Faster R-CNN en la detección y clasificación de regiones de texto en certificados de nacimiento y cédulas de identidad. Elaboración propia

3.4.1.3. Condiciones de Entrenamiento

Para garantizar una comparación justa entre YOLO y Faster R-CNN, se definieron condiciones homogéneas con el objetivo de minimizar diferencias externas y permitir una comparación directa

del desempeño entre ambas arquitecturas. Los modelos se instanciaron en un entorno de trabajo de Jupyter Notebook, utilizando versiones preentrenadas y aplicando transfer learning, ajustando la capa de salida al número de clases definido para cada tipo de documento.

Se estableció SGD (Stochastic Gradient Descent) como optimizador, con una tasa de aprendizaje inicial de 0.001, momentum de 0.9 y weight decay de 1e-4, así como un batch de 4 imágenes por iteración y resoluciones de entrada de 960×960 píxeles, con el objetivo de preservar los detalles en regiones de texto pequeñas sin comprometer la eficiencia. Se definió un total de 50 epochs, suficientes para que ambos modelos pudieran alcanzaran convergencia.

3.4.1.4. YOLO

En cuanto a YOLO, se seleccionó el modelo YOLOv8s, una versión ligera diseñada para ofrecer alta velocidad de inferencia sin comprometer significativamente la precisión. Este modelo cuenta con aproximadamente 11 millones de parámetros para entrenamiento. Además, su arquitectura pertenece a la categoría de detectores one-stage, lo que significa que realiza de manera conjunta la detección de regiones de interés y la clasificación de objetos en una sola pasada por la red.

a) Arquitectura

El modelo se configuró modificando el archivo yolov8s.yaml según la documentación de la versión s. Se aplicó un factor de ancho (width=0.50), reduciendo a la mitad el número de canales de la arquitectura estándar para disminuir la complejidad y el consumo de memoria. Además, se ajustó el parámetro nc para definir el número de clases según el tipo de documento:

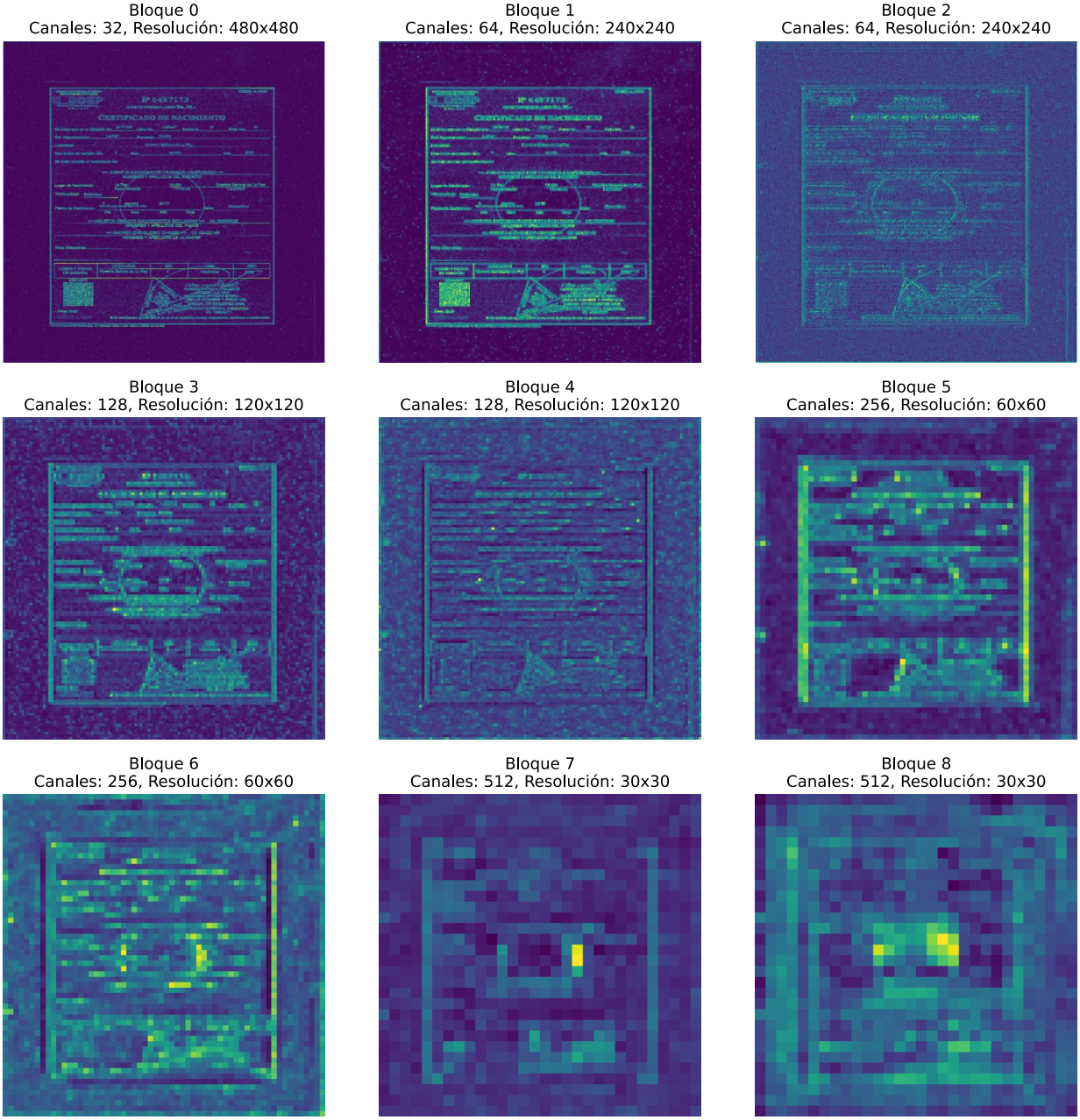
- $nc = 12$, para certificados de nacimiento.
- $nc = 4$, para cédulas de identidad.

Se trabajó con imágenes de 960×960 píxeles y 3 canales de color (RGB), por lo que la entrada al modelo corresponde a un tensor de 3×960×960, donde cada canal contiene 921,600 valores que representan la intensidad de color normalizada de cada píxel.

Para comprobar los cambios realizados en la arquitectura, se analizaron los feature maps generados por los primeros bloques del backbone, como se muestra en la Figura 40. Esto permitió observar cómo varían la resolución y el número de canales según los parámetros definidos, y cómo se extrae características progresivamente más complejas para la detección.

Figura 40

Salida de los bloques de la arquitectura YOLO

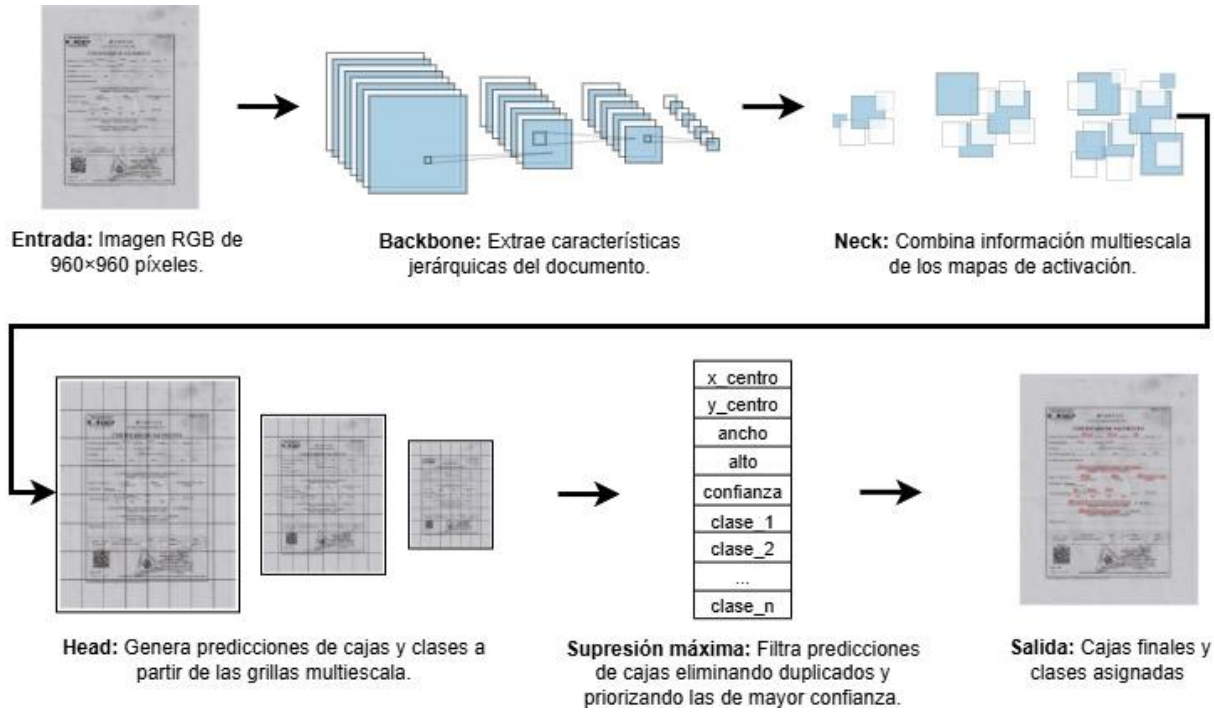


Nota: La figura muestra los primeros nueve bloques del backbone de YOLO y cómo cambian la resolución y el número de canales según los parámetros de la arquitectura. Los canales se representan de forma resumida mediante un promedio de los feature maps. Elaboración propia mediante Jupyter Notebook.

La salida del modelo es un vector que incluye las coordenadas y dimensiones de la caja delimitadora (x, y, w, h), la confianza de objeto (object_conf) y las probabilidades de clase. Para certificados de nacimiento (nc=12), las últimas doce posiciones indican la probabilidad de cada clase, mientras que para cédulas de identidad (nc=4) corresponden a las cuatro clases definidas. La Figura 41 muestra de manera resumida el flujo completo del proceso que sigue el modelo YOLO configurado.

Figura 41

Esquema del proceso de detección en la arquitectura YOLO



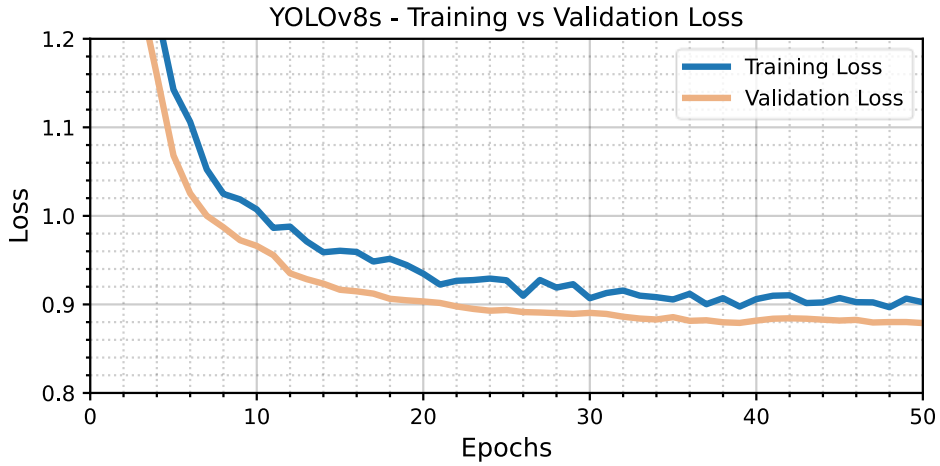
Nota: La figura muestra cómo YOLO extrae características, combina información multiescala, genera predicciones de cajas y clases, y aplica NMS para eliminar detecciones redundantes. Elaboración propia mediante draw.io.

b) Entrenamiento en Certificados de Nacimiento

Durante el entrenamiento del modelo YOLOv8s, se registraron la pérdida y el mAP. La Figura 42 muestra cómo la pérdida disminuye y se estabiliza, indicando buena generalización, mientras que la Figura 43 evidencia el aumento y estabilización del mAP, reflejando la mejora en la detección y clasificación de los campos textuales en certificados de nacimiento.

Figura 42

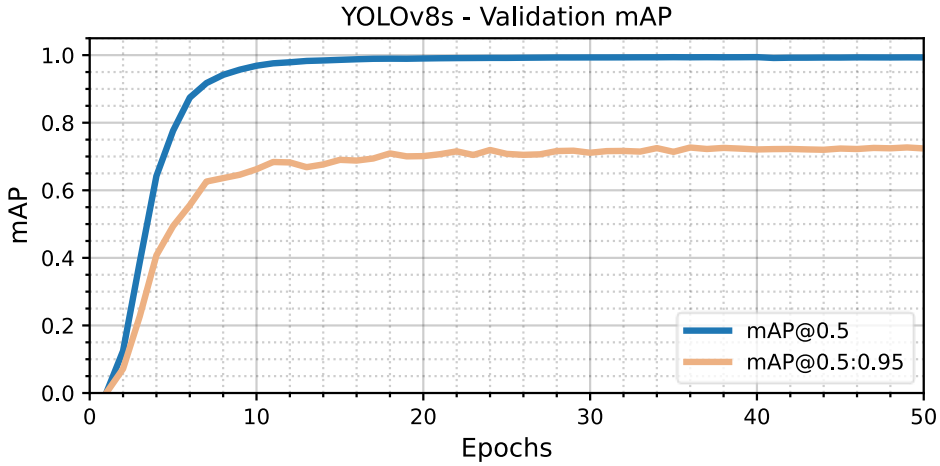
Evolución de la pérdida - YOLO certificados de nacimiento



Nota: La gráfica muestra la evolución de las pérdidas durante 50 épocas, estabilizándose ambas en un valor cercano a 0.9, con la pérdida de validación ligeramente inferior a la de entrenamiento, sin sobreajuste. Elaboración propia.

Figura 43

Evolución del mAP- YOLO certificados de nacimiento



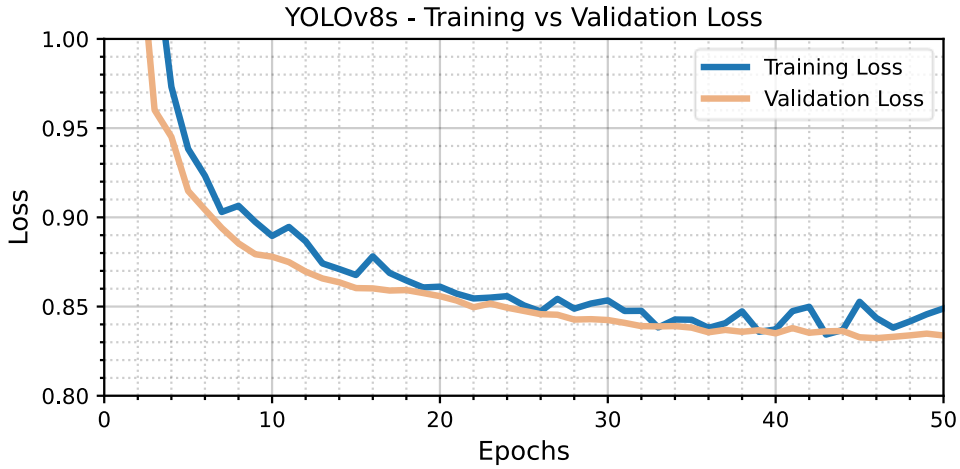
Nota: La gráfica muestra la evolución del mAP durante 50 épocas, con mAP@0.5 cercano a 1 y mAP@0.5:0.95 alrededor de 0.7, reflejando que el mAP a umbrales más estrictos es más exigente. Elaboración propia.

c) Entrenamiento en Cedulas de Identidad

Se registraron la pérdida y el mAP. La Figura 44 muestra cómo la pérdida disminuye y se estabiliza, indicando buena generalización, mientras que la Figura 45 evidencia el aumento y estabilización del mAP, reflejando la mejora en la detección y clasificación de los campos.

Figura 44

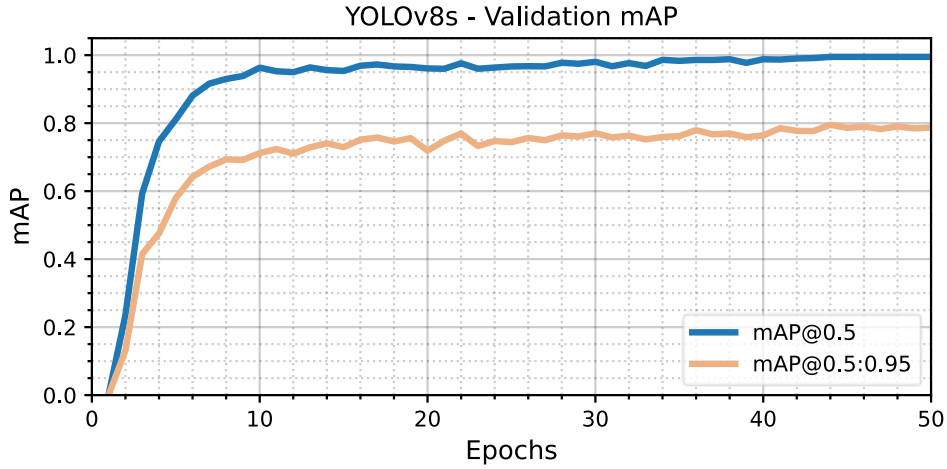
Evolución de la pérdida - YOLO cédulas de identidad



Nota: La gráfica muestra la evolución de las pérdidas durante 50 épocas, con algunas fluctuaciones, estabilizándose ambas en un valor cercano a 0.85, y con la pérdida de validación ligeramente inferior a la de entrenamiento, sin sobreajuste. Elaboración propia.

Figura 45

Evolución de la pérdida - YOLO cédulas de identidad



Nota: La gráfica muestra la evolución del mAP durante 50 épocas, con mAP@0.5 cercano a 1 y mAP@0.5:0.95 alrededor de 0.8, reflejando que el mAP a umbrales más estrictos es más exigente. Elaboración propia.

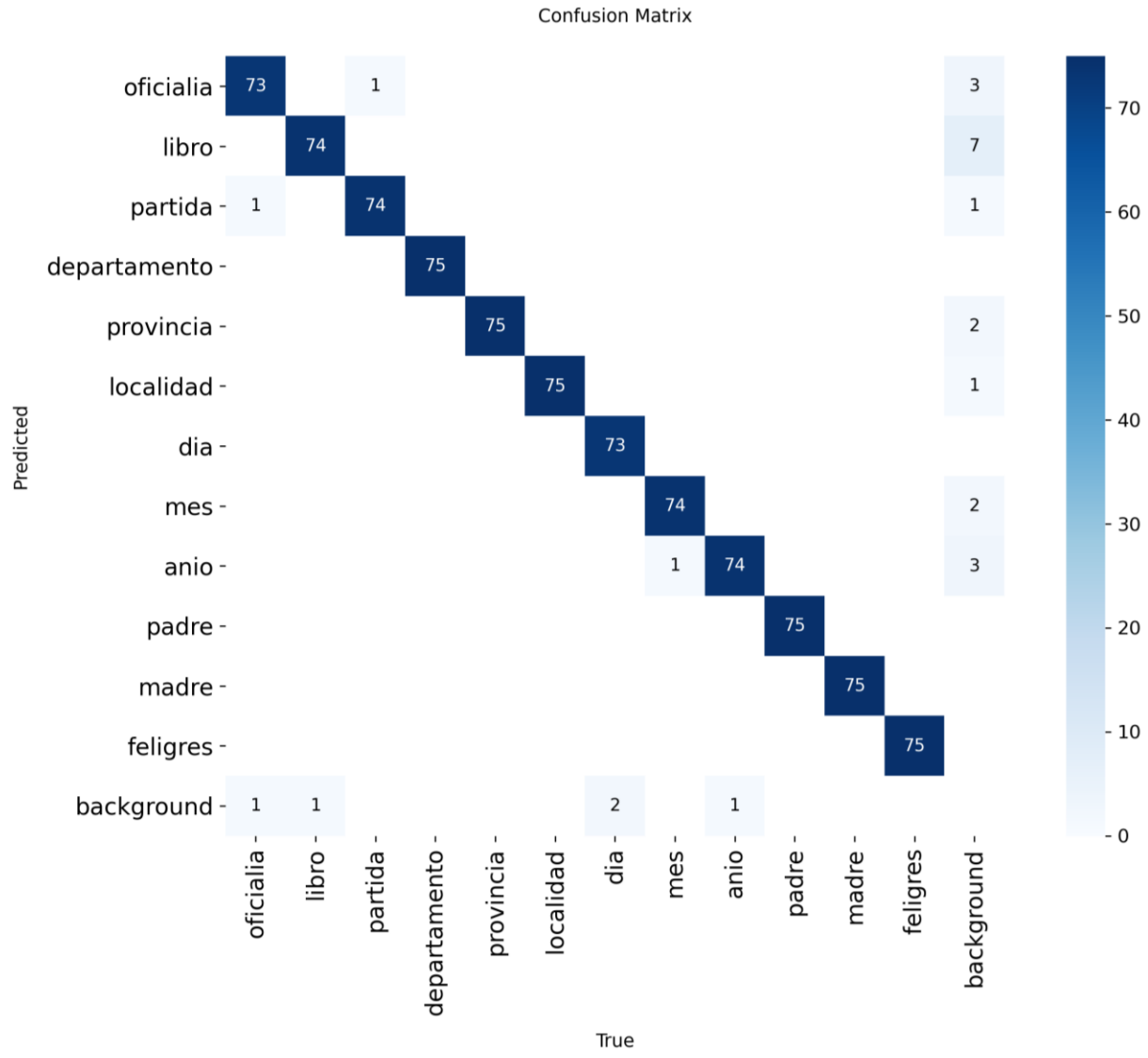
d) Evaluación del Desempeño mediante mAP y Métricas de Clasificación

Una vez finalizado el entrenamiento de los modelos, se generaron las matrices de confusión a partir del conjunto de prueba. En el caso de los certificados de nacimiento, el conjunto estuvo

conformado por 75 imágenes, cada una con una instancia de cada clase. La Figura 46 presenta la matriz obtenida, donde los tonos más oscuros representan una mayor cantidad de predicciones correctas por clase.

Figura 46

Matriz de confusión - YOLO certificados de nacimiento



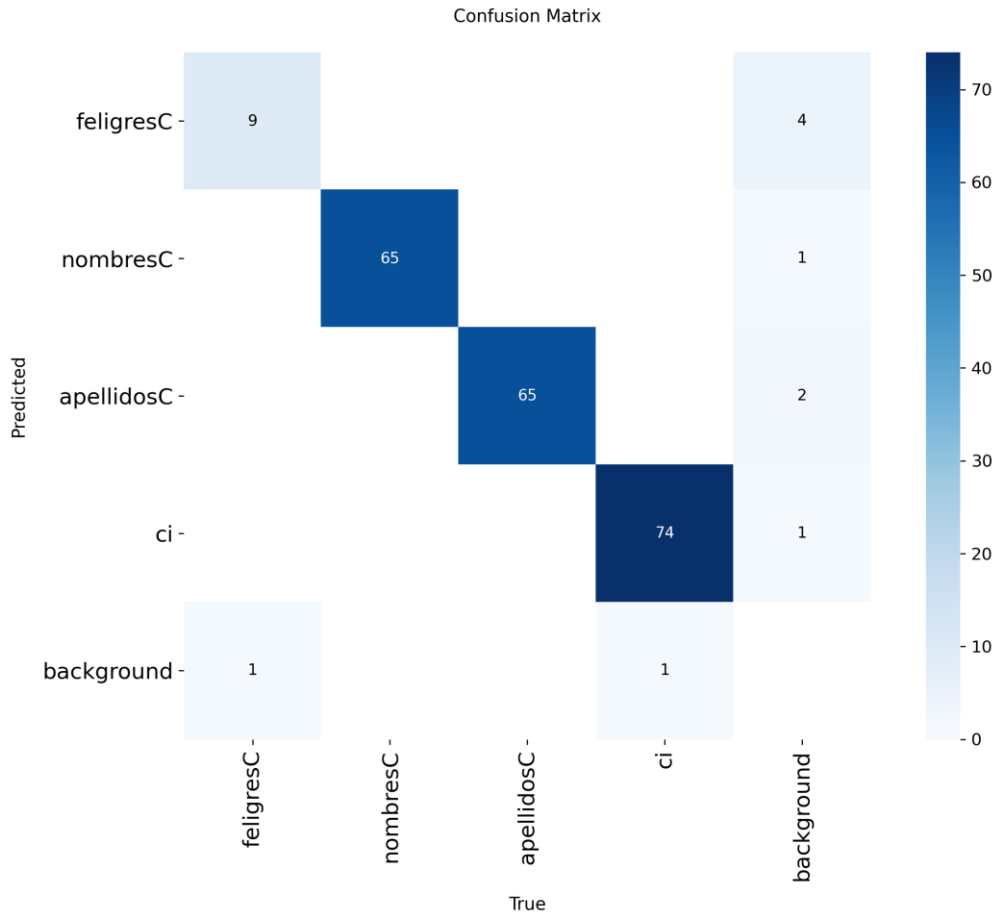
Nota: La figura muestra la matriz de confusión del modelo YOLOv8s en certificados de nacimiento, generada a partir del conjunto de prueba usando la librería Ultralytics. Elaboración propia mediante Jupyter Notebook.

De la misma manera, se generó la matriz de confusión para el conjunto de prueba correspondiente a las cédulas de identidad, el cual estuvo conformado por 10 instancias de la

clase feligresaC, 65 de nombresC, 65 de apellidosC y 75 de la clase ci. La Figura 47 presenta la matriz obtenida, donde se muestra la distribución de las predicciones del modelo en relación con las clases reales.

Figura 47

Matriz de confusión - YOLO cédulas de identidad



Nota: La figura muestra la matriz de confusión del modelo YOLOv8s, para cédulas de identidad, generada a partir del conjunto de prueba utilizando la librería Ultralytics. Elaboración propia mediante Jupyter Notebook.

A partir de las matrices de confusión se calcularon las métricas de clasificación precisión, recall y F1 Score para cada tipo de documento. Por otro lado, para evaluar la capacidad del modelo para detectar correctamente las regiones de texto, se obtuvieron directamente mediante la librería Ultralytics las métricas de mean Average Precision (mAP) a los umbrales 0.5 y 0.5:0.95 sobre el conjunto de prueba. En la Tabla 25 se presentan los resultados desglosados por documento, incluyendo todas las métricas mencionadas.

Tabla 25

Métricas de evaluación en detección de texto con YOLOv8s

Documento	Precisión	Recall	F1 Score	mAP@0.5	mAP@0.5:0.95
Certificado de Nacimiento	0.976	0.991	0.983	0.992	0.737
Cédula de Identidad	0.964	0.991	0.977	0.975	0.752

Nota: La tabla presenta las métricas utilizadas para evaluar el desempeño del modelo YOLOv8s en la detección de regiones de texto sobre el conjunto de prueba, incluyendo precisión, recall, F1 Score y mean Average Precision (mAP) a los umbrales 0.5 y 0.5:0.95. Elaboración propia.

3.4.1.5. FASTER RCNN

Se seleccionó el modelo Faster R-CNN con MobileNetV3-Large FPN, que cuenta con aproximadamente 18 millones de parámetros para entrenamiento. Este modelo pertenece a la categoría de detectores two-stage, ya que realiza la detección y clasificación en dos fases separadas: primero genera propuestas de regiones y luego evalúa y clasifica cada propuesta, refinando las cajas delimitadoras.

a) Arquitectura

Para adaptar el modelo Faster R-CNN a nuestro caso, se reemplazó la cabeza de predicción para que reconociera únicamente las clases de interés, diferenciando entre certificados de nacimiento y cédulas de identidad, como se ilustra en la Figura 48.

Figura 48

Adaptación de arquitectura Faster R-CNN

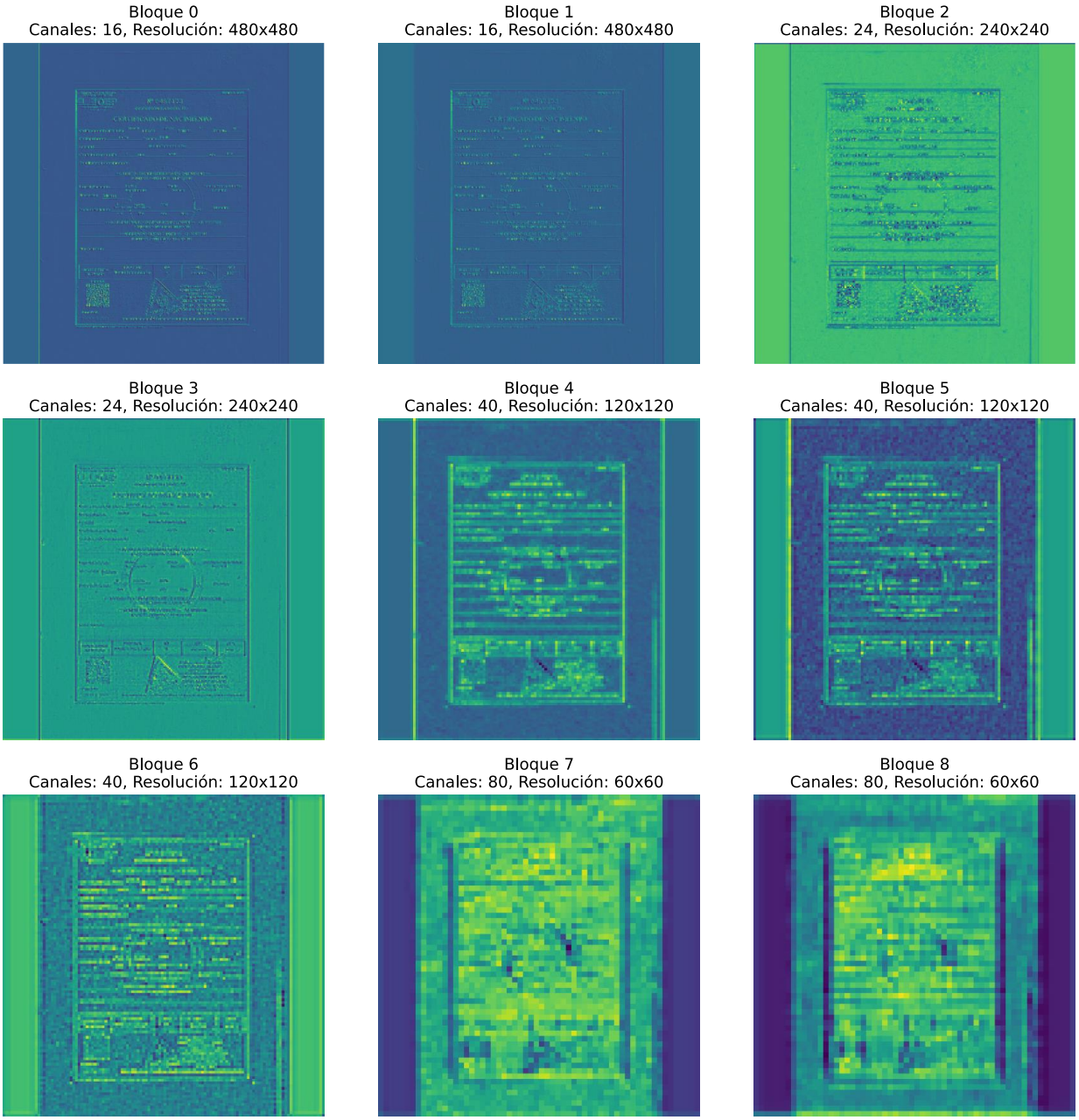
```
model = models.detection.fasterrcnn_mobilenet_v3_large_fpn(pretrained=True)
in_features = model.roi_heads.box_predictor.cls_score.in_features
model.roi_heads.box_predictor = models.detection.faster_rcnn.FastRCNNPredictor(in_features, n_classes+1)
```

Nota: La figura muestra el código donde se carga el modelo Faster R-CNN preentrenado y se reemplaza la cabeza de predicción para adaptarlo a clases requeridas, con n_classes=12 para certificados de nacimiento y n_ classes=4 para cédulas de identidad. Elaboración propia mediante Jupyter Notebook.

Una vez definida la arquitectura, se analizaron los feature maps generados por el backbone, como se muestra en la Figura 49, observando las salidas de algunos bloques de la arquitectura.

Figura 49

Salida de los bloques de la arquitectura Faster R-CNN

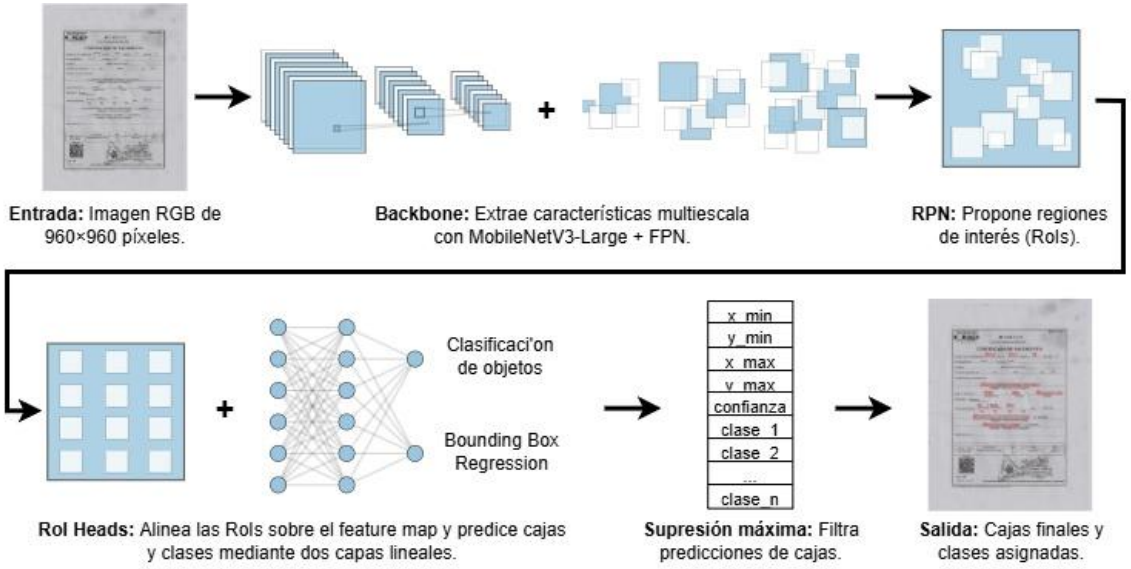


Nota: La figura muestra de forma resumida, mediante un promedio de los feature maps de nueve de los diecisiete bloques del backbone de Faster R-CNN, cómo varían la resolución y el número de canales según los parámetros de la arquitectura. Elaboración propia mediante Jupyter Notebook.

A diferencia de YOLO, la salida de Faster R-CNN está organizada por regiones propuestas, y cada predicción incluye las coordenadas de la caja delimitadora (x_min, y_min, x_max, y_max), la puntuación de confianza de la región (object_conf) y las probabilidades de clase. Estas predicciones se generan mediante el bloque (cls_score), que devuelve el puntaje de la clase, y (bbox_pred), que proporciona las coordenadas refinadas de la caja. La Figura 50 resume de manera clara el flujo completo seguido por el modelo Faster R-CNN instanciado.

Figura 50

Proceso de detección de la arquitectura Faster R-CNN



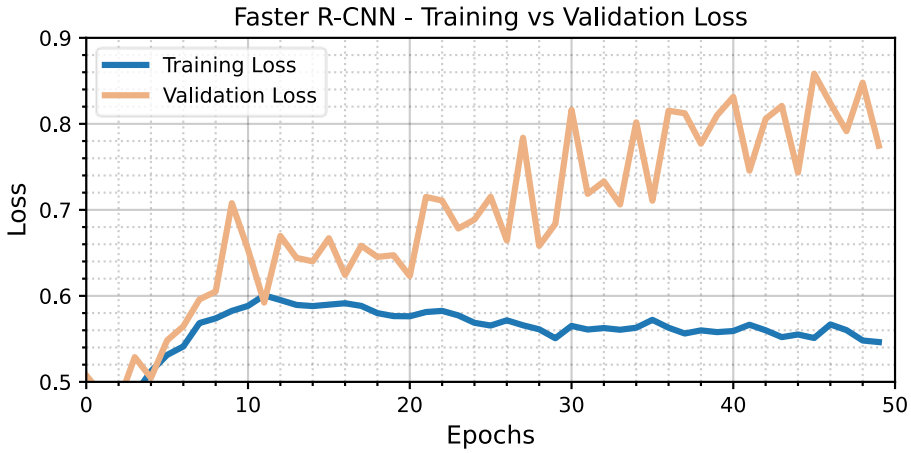
Nota: La figura ilustra cómo la red extrae características, genera propuestas de regiones, procesa las ROIs y produce predicciones finales de cajas y clases mediante los componentes backbone, RPN y RoI heads. Elaboración propia mediante draw.io.

b) Entrenamiento en Certificados de Nacimiento

Durante el entrenamiento del modelo Faster R-CNN, se registraron la pérdida y el mAP. La Figura 51 muestra que la pérdida presenta algunas fluctuaciones que podrían sugerir sobreajuste. Sin embargo, la Figura 52 evidencia que el mAP aumenta y se estabiliza, permaneciendo alto en el conjunto de validación, lo que indica que el modelo mantiene una buena capacidad de generalización en la detección y clasificación de los campos textuales en certificados de nacimiento.

Figura 51

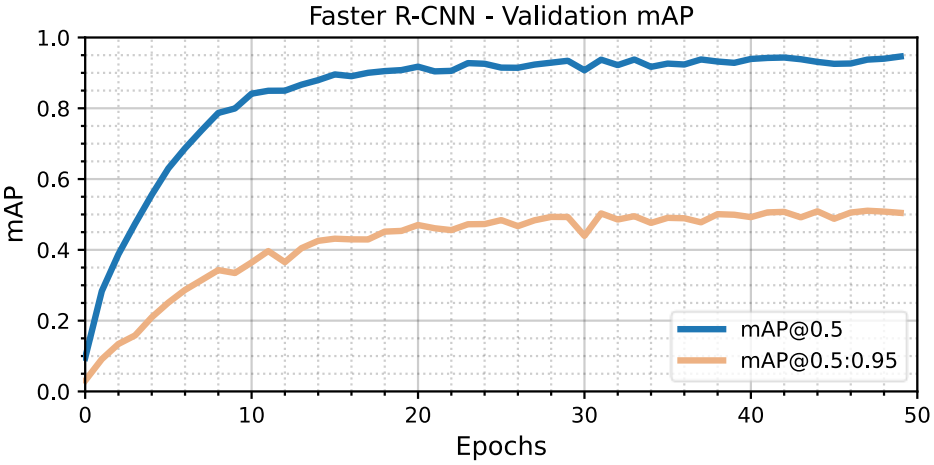
Evolución de la pérdida - Faster R-CNN certificados de nacimiento



Nota: La gráfica muestra la evolución de las pérdidas durante 50 épocas, con la pérdida de validación ligeramente superior a la de entrenamiento, con una diferencia de aproximadamente 0.2. Elaboración propia.

Figura 52

Evolución del mAP - Faster R-CNN certificados de nacimiento



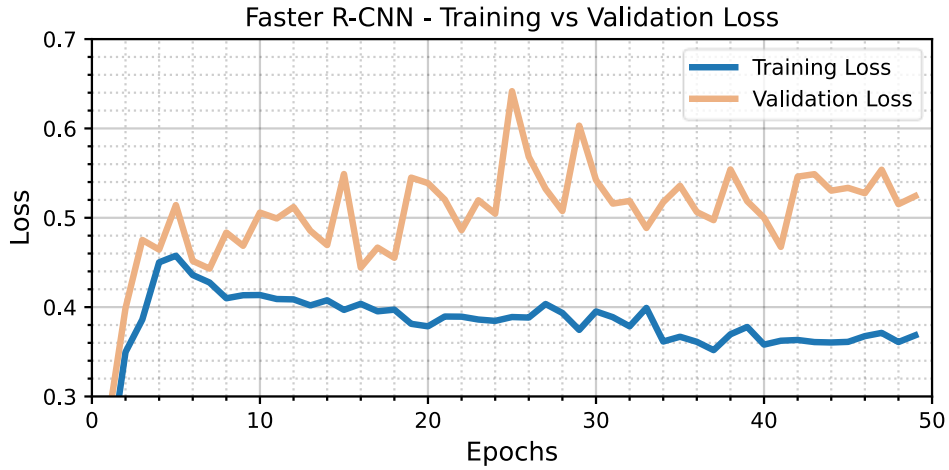
Nota: La gráfica muestra la evolución del mAP durante 50 épocas, con mAP@0.5 cercano a 1 y mAP@0.5:0.95 alrededor de 0.5, reflejando que el mAP a umbrales más estrictos es más exigente. Elaboración propia.

c) Entrenamiento en Cedulas de Identidad

Se registraron la pérdida y el mAP. La Figura 53 muestra la pérdida de validación ligeramente por encima de la de entrenamiento, mientras que la Figura 54 evidencia el aumento y estabilización del mAP, reflejando la mejora en la detección y clasificación de los campos.

Figura 53

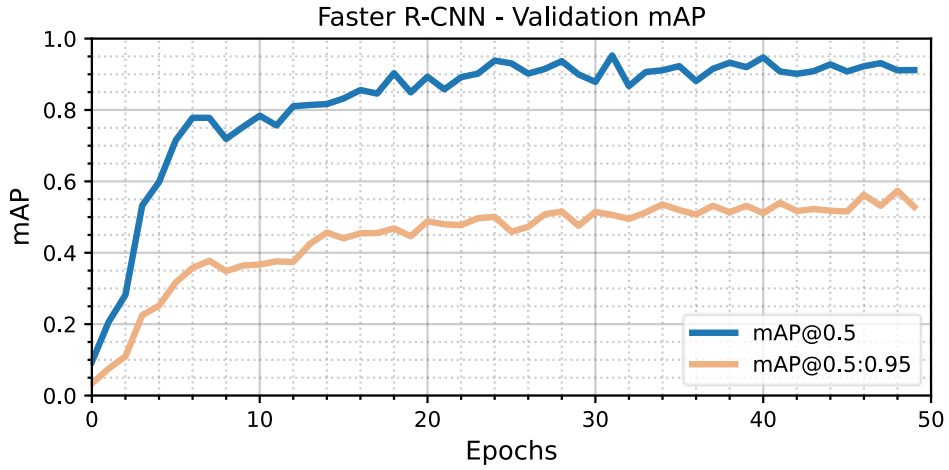
Evolución de la pérdida - Faster R-CNN cédulas de identidad



Nota: La gráfica muestra la evolución de las pérdidas durante 50 épocas, con la pérdida de validación ligeramente superior a la de entrenamiento, con una diferencia de aproximadamente 0.15. Elaboración propia.

Figura 54

Evolución del mAP - Faster R-CNN cédulas de identidad



Nota: La gráfica muestra la evolución del mAP durante 50 épocas, con mAP@0.5 cercano a 0.9 y mAP@0.5:0.95 alrededor de 0.5, reflejando que el mAP a umbrales más estrictos es más exigente. Elaboración propia.

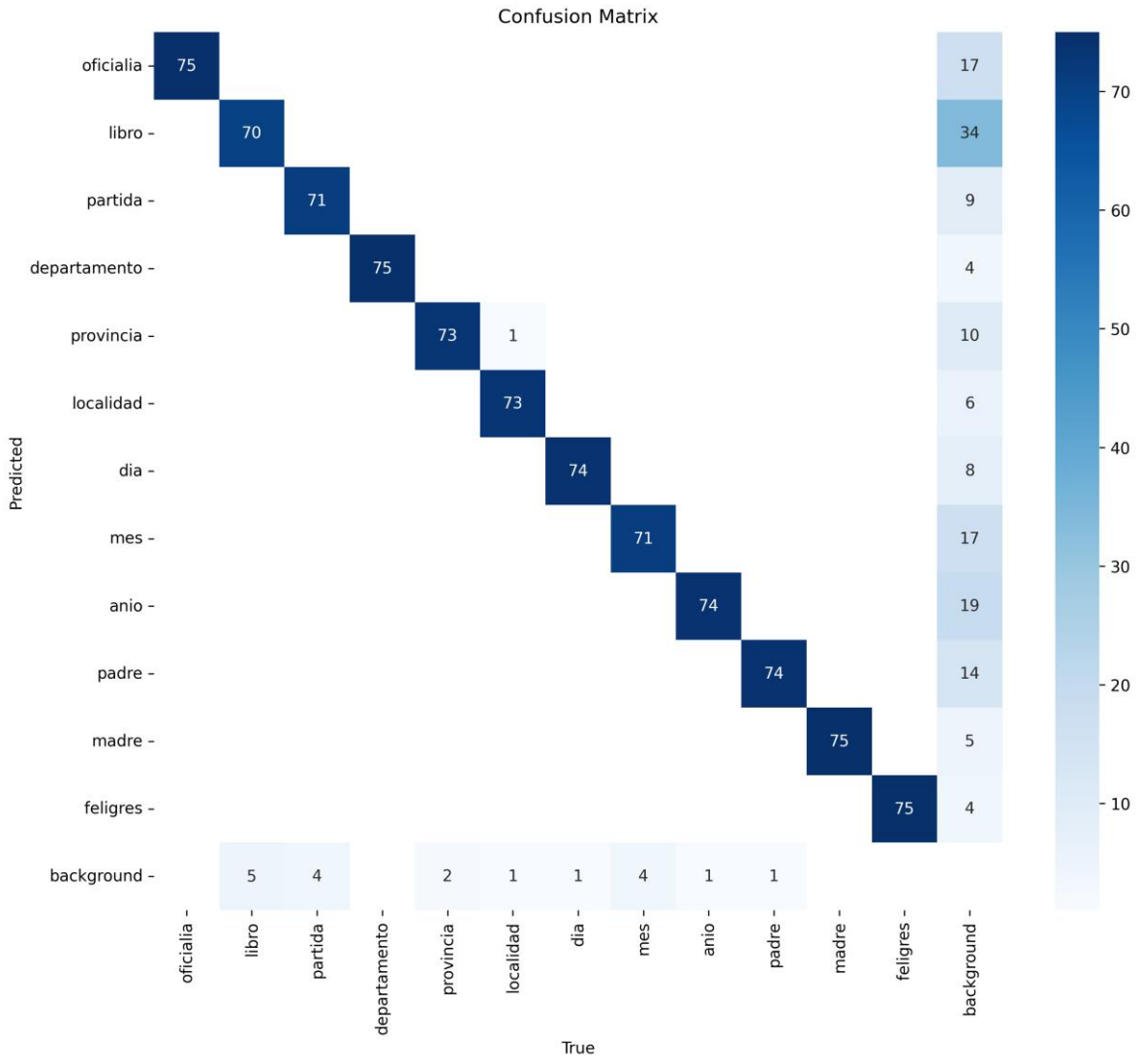
d) Evaluación del Desempeño mediante mAP y Métricas de Clasificación

Una vez finalizado el entrenamiento de los modelos, se generaron las matrices de confusión a utilizando el conjunto de prueba. Para los certificados de nacimiento, este conjunto estuvo

compuesto por 75 imágenes, cada una con una instancia de cada clase. La Figura 55 muestra la matriz resultante, donde los tonos más oscuros indican un mayor número de predicciones correctas por clase.

Figura 55

Matriz de confusión - Faster R-CNN certificados de nacimiento



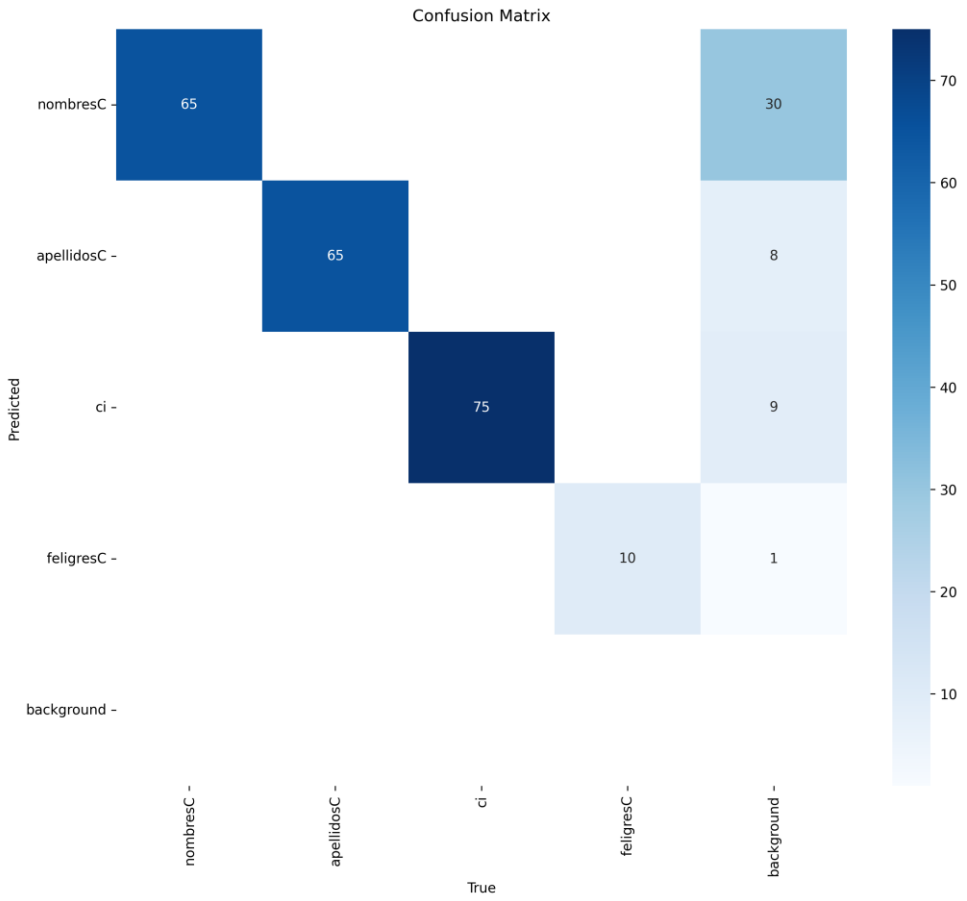
Nota: La figura muestra la matriz de confusión del modelo Faster R-CNN para certificados de nacimiento, generada a partir del conjunto de prueba mediante un script en Python. Elaboración propia mediante Jupyter Notebook.

De manera similar, se construyó la matriz de confusión para el conjunto de prueba correspondiente a las cédulas de identidad, compuesto por 10 instancias de la clase feligresC,

65 de nombresC, 65 de apellidosC y 75 de la clase ci. La Figura 56 muestra la matriz resultante, junto a la distribución de las predicciones del modelo en comparación con las clases reales.

Figura 56

Matriz de confusión - Faster R-CNN cedulas de identidad



Nota: La figura muestra la matriz de confusión del modelo Faster R-CNN para cédulas de identidad, generada a partir del conjunto de prueba mediante un script de Python. Elaboración propia mediante Jupyter Notebook.

A partir de las matrices de confusión se calcularon las métricas de clasificación incluyendo precisión, recall y F1 Score para cada tipo de documento. Por otro lado, para evaluar la capacidad del modelo de detectar correctamente las regiones de texto, se empleó la librería pycocotools.cocoeval para obtener las métricas de mean Average Precision (mAP) a los umbrales 0.5 y 0.5:0.95 sobre el conjunto de prueba. En la Tabla 26 se presentan los resultados desglosados por documento, incluyendo todas las métricas mencionadas.

Tabla 26

Métricas de evaluación en detección de texto con Faster R-CNN

Documento	Precisión	Recall	F1 Score	mAP@0.5	mAP@0.5:0.95
Certificado de Nacimiento	0.856	0.978	0.913	0.946	0.520
Cédula de Identidad	0.817	1.000	0.900	0.931	0.519

Nota: La tabla presenta las métricas utilizadas para evaluar el desempeño del modelo Faster R-CNN en la detección de regiones de texto sobre el conjunto de prueba, incluyendo precisión, recall, F1 Score y mean Average Precision (mAP) a los umbrales 0.5 y 0.5:0.95. Elaboración propia.

3.4.1.6. Análisis Comparativo de Resultados

Durante el entrenamiento, Faster R-CNN mostró una mayor fluctuación en la pérdida de validación, con diferencias de 0.15–0.20 respecto al entrenamiento. A pesar de estas oscilaciones, el mAP continuó aumentando, mostrando que el modelo seguía aprendiendo de manera efectiva, aunque de forma menos estable. En contraste, YOLOv8s presentó curvas de pérdida más estables, con la validación ligeramente inferior al entrenamiento, lo que reflejaba un aprendizaje más consistente.

Esta diferencia en la estabilidad podría explicarse por la naturaleza de la arquitectura de Faster R-CNN. Al ser un detector two-stage que utiliza una Region Proposal Network para generar propuestas de regiones antes de clasificarlas, el proceso de aprendizaje puede ser más sensible a variaciones, especialmente en documentos con texto de distintos tamaños. En cambio, YOLOv8s, al ser un detector one-stage, realiza la detección y clasificación de manera simultánea, lo que le da una mayor consistencia durante el entrenamiento.

Estos comportamientos se reflejaron también en las métricas de desempeño resumidas en la Tabla 27. En términos de F1 Score, YOLOv8s alcanzó valores superiores, con 0.983 en certificados de nacimiento y 0.977 en cédulas de identidad, mientras que Faster R-CNN obtuvo 0.913 y 0.900 respectivamente. Esto indica que YOLO logra un mejor equilibrio entre precisión y recall, ofreciendo mayor exactitud y cobertura en la clasificación de campos de texto.

De manera similar, en la métrica de mAP@0.5:0.95, YOLOv8s superó nuevamente a Faster R-CNN, con 0.737 y 0.752 frente a 0.520 y 0.519, respectivamente, lo que indica una detección más precisa de los límites de los campos de texto en los documentos escaneados.

Tabla 27

Comparación de métricas YOLOv8s y Faster R-CNN

Modelo	Documento	F1 Score	mAP@0.5:0.95
YOLOv8s	Certificado de Nacimiento	0.983	0.737
	Cédula de Identidad	0.977	0.752
Faster R-CNN	Certificado de Nacimiento	0.913	0.520
	Cédula de Identidad	0.900	0.519

Nota: La tabla resume los valores de F1 Score y mAP@0.5:0.95 obtenidos por los modelos YOLOv8s y Faster R-CNN en certificados de nacimiento y cédulas de identidad, permitiendo observar de manera directa las diferencias de desempeño entre ambas arquitecturas. Elaboración propia.

En conclusión, aunque ambos modelos cumplen con la tarea de detección y clasificación de regiones de texto, YOLOv8s sobresale como la opción más adecuada, gracias a su combinación de alta precisión en el procesamiento de documentos.

3.4.2. Comparativa de Modelos para la Extracción de Texto

3.4.2.1. Selección de Modelos

En las áreas del reconocimiento y extracción de texto, se encontraron investigaciones en las cuales redes híbridas CNN + RNN lograban altos niveles de precisión en documentos con textos impresos y manuscritos. En particular, la investigación de Mahadevkar, Patil y Kotecha (2024) mostró que las LSTM bidireccionales, combinadas con CNN y decodificación CTC, alcanzan precisiones superiores al 98 % en el reconocimiento de palabras. Por otro lado, las GRU presentan un rendimiento similar al de LSTM, mostrando un buen desempeño en tareas secuenciales, como señalan Zulqarnain, Ghazali, Mohmad Hassim y Rehan (2020).

Estas investigaciones nos permitieron seleccionar las arquitecturas que serían evaluadas en el proyecto, considerando su desempeño reportado y su idoneidad para cada tarea. La Tabla 28 presenta las arquitecturas preseleccionadas, destacando su respectiva justificación.

Tabla 28

Arquitecturas preseleccionadas para evaluación

Tarea	Arquitectura	Justificación
Reconocimiento de texto	CNN + LSTM + bidireccional + CTC	Alta precisión en textos manuscritos e impresos.
	CNN + GRU + bidireccional + CTC	Rendimiento similar al de LSTM.

Nota: La tabla muestra las arquitecturas preseleccionadas para cada tarea, junto con la justificación basada en estudios previos y desempeño reportado en la literatura. Elaboración propia.

3.4.2.2. Arquitecturas

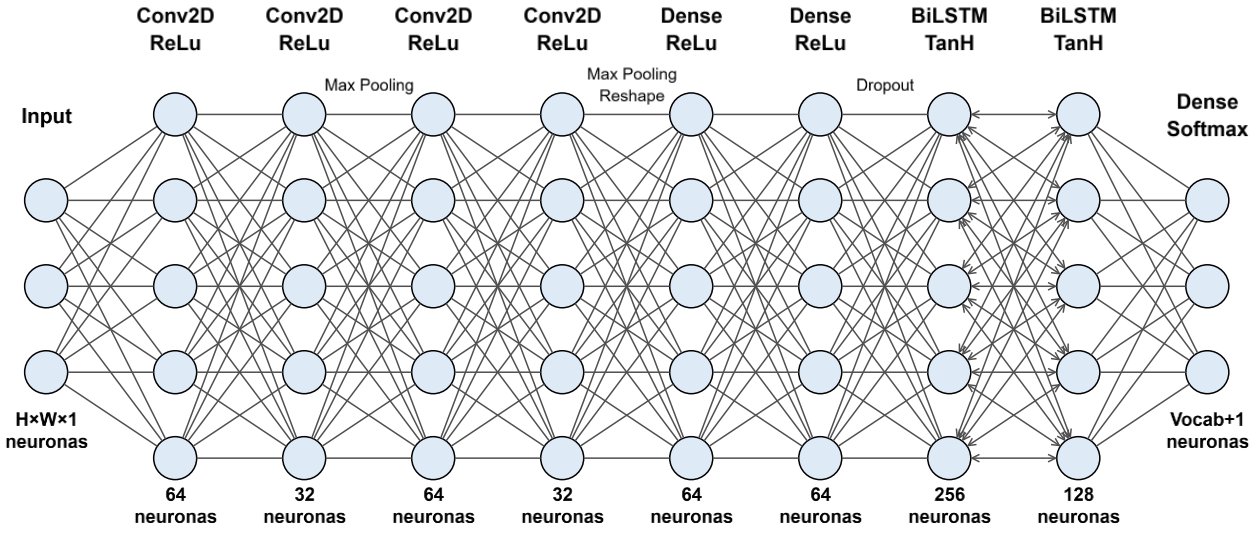
La arquitectura de los modelos híbridos de reconocimiento de texto puede organizarse en fases para facilitar su comprensión. En la fase de entrada, se reciben las imágenes del texto y, durante el entrenamiento, las etiquetas correspondientes. A continuación, la fase de extracción de características utiliza convoluciones Conv2D y operaciones de MaxPooling para transformar las imágenes en representaciones jerárquicas que capturan patrones visuales relevantes.

En la fase de codificación, las representaciones extraídas se reorganizan en secuencias adecuadas para el procesamiento secuencial mediante capas Dense, permitiendo que el decodificador capture las dependencias temporales entre los caracteres a lo largo de la secuencia. Para ello, se emplean redes recurrentes, como LSTM o GRU, que procesan la información en ambas direcciones. Finalmente, en la fase de salida, una capa Dense con activación Softmax transforma la información en probabilidades para cada carácter.

Par este caso, se emplearon redes bidireccionales LSTM y GRU, y sus arquitecturas se muestran en los diagramas de las Figuras 57 y 58. La bidireccionalidad implica que cada secuencia se procesa en ambos sentidos (hacia adelante y hacia atrás), lo que permite capturar dependencias pasadas y futuras dentro de la secuencia de caracteres. Las LSTM cuentan con tres compuertas internas (entrada, olvido y salida) que regulan el flujo de información, mientras que las GRU poseen dos compuertas (actualización y reinicio), simplificando la estructura recurrente. Debido a estas diferencias en el funcionamiento interno, no se definió la misma cantidad de neuronas en las capas de LSTM y GRU (ver Anexos G y H), en cambio, la cantidad de neuronas se seleccionó considerando el número de parámetros, asegurando que ambas arquitecturas tuvieran una capacidad de representación similar y un entrenamiento equilibrado.

Figura 57

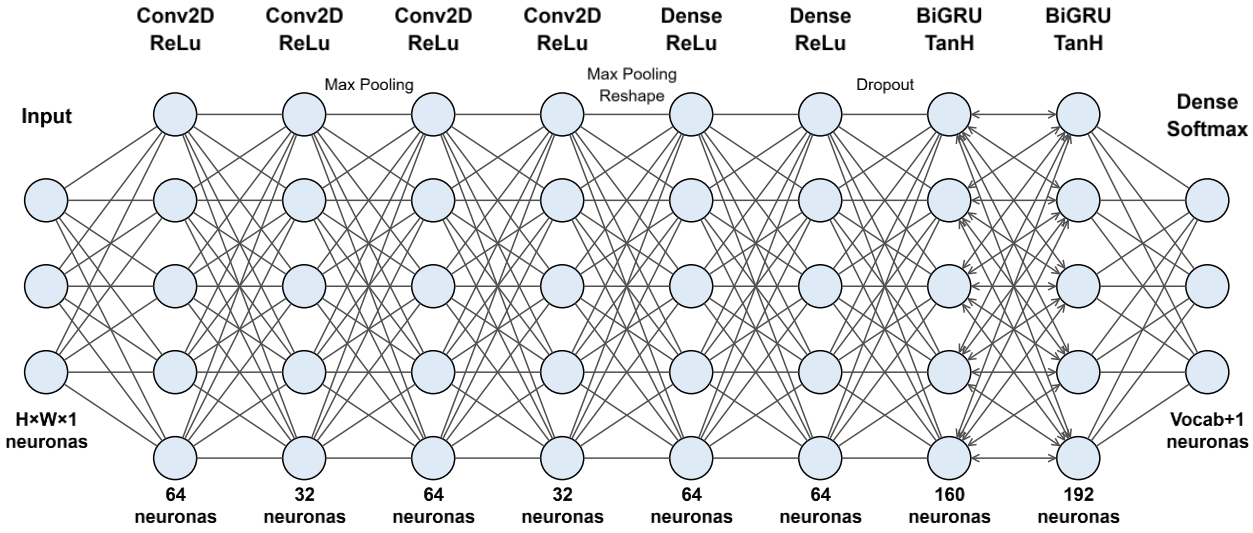
Diagrama de red neuronal híbrida CNN-BiLSTM



Nota: La figura ilustra la arquitectura CNN-BiLSTM propuesta, con entrada (H, W, 1), donde la CNN extrae características que las capas BiLSTM procesan para finalmente pasar por una capa Dense Softmax que genera la salida (Vocab+1), siendo Vocab el número de caracteres posibles. Elaboración propia en draw.io.

Figura 58

Diagrama de red neuronal híbrida CNN-BiGRU



Nota: La figura ilustra la arquitectura CNN-BiGRU propuesta, con entrada (H, W, 1), donde la CNN extrae características que las capas BiGRU procesan para finalmente pasar por una capa Dense Softmax que genera la salida (Vocab+1), siendo Vocab el número de caracteres posibles. Elaboración propia en draw.io.

3.4.2.3. Implementación de la Capa CTC

En tareas de OCR, las secuencias de salida tienen longitudes variables y no siempre existe una alineación directa entre los caracteres de la imagen y las etiquetas correspondientes. Para abordar este problema, se utiliza la función de pérdida Connectionist Temporal Classification (CTC), que permite entrenar redes neuronales para generar secuencias de caracteres sin requerir alineación previa. La CTC calcula la probabilidad de todas las posibles correspondencias entre las predicciones del modelo y las secuencias reales.

Para entrenar los modelos propuestos se utilizó TensorFlow/Keras, implementando una versión personalizada de la función de pérdida CTC, ya que Keras no la proporciona como capa estándar de forma directa. La capa CTC permite calcular la pérdida entre las secuencias de salida predichas por la red y las secuencias reales, en la Figura 59 se ilustra el flujo del cálculo de la pérdida CTC dentro de los modelos.

Figura 59

Implementación de la capa CTC para reconocimiento de texto

```
class CTCLayer(Layer):
    def __init__(self, **kwargs) -> None:
        super().__init__(**kwargs)
        self.loss_function = tf.keras.backend.ctc_batch_cost

    def call(self, y_true, y_hat):
        batch_len = tf.cast(tf.shape(y_true)[0], dtype="int64")
        input_len = tf.cast(tf.shape(y_hat)[1], dtype='int64') * tf.ones(shape=(batch_len, 1), dtype='int64')
        label_len = tf.cast(tf.shape(y_true)[1], dtype='int64') * tf.ones(shape=(batch_len, 1), dtype='int64')
        loss = self.loss_function(y_true, y_hat, input_len, label_len)
        self.add_loss(loss)
        return y_hat
```

Nota: La figura muestra el código de la capa personalizada CTC en TensorFlow/Keras, utilizada para calcular la pérdida CTC y permitir el entrenamiento de los modelos OCR. Elaboración propia mediante Jupyter Notebook.

3.4.2.4. Condiciones de Entrenamiento

Para el entrenamiento se consideraron únicamente los caracteres presentes en la documentación recolectada. El vocabulario del modelo incluye letras mayúsculas y minúsculas, números, ciertos símbolos y caracteres especiales como 'Ñ' y 'ñ', resultando en un total de 66 unidades en la capa de salida, incluyendo un token adicional necesario para la función de pérdida CTC. Este vocabulario define explícitamente todas las posibles clases que la red puede predecir.

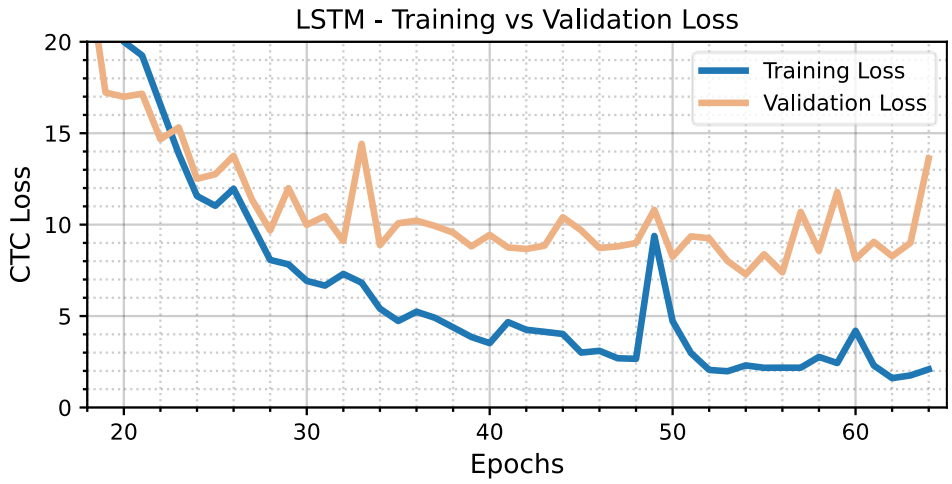
Las imágenes de entrada fueron redimensionadas a 1024 × 64 píxeles y normalizadas para mantener consistencia en la entrada esperada por los modelos definidos. El entrenamiento se realizó con batches de 16 muestras durante 100 épocas, utilizando el optimizador Adam con una tasa de aprendizaje de 0.001.

3.4.2.5. Evolución del Entrenamiento

Durante el entrenamiento del modelo LSTM bidireccional, se registró la evolución de la pérdida, presentada en la Figura 60. La gráfica muestra fluctuaciones en la pérdida de validación, con picos cercanos a 15, mientras que la pérdida de entrenamiento se mantiene alrededor de 5. Este comportamiento refleja la complejidad del aprendizaje secuencial y la sensibilidad del modelo a ciertas secuencias de entrada.

Figura 60

Evolución de la pérdida - Modelo LSTM bidireccional

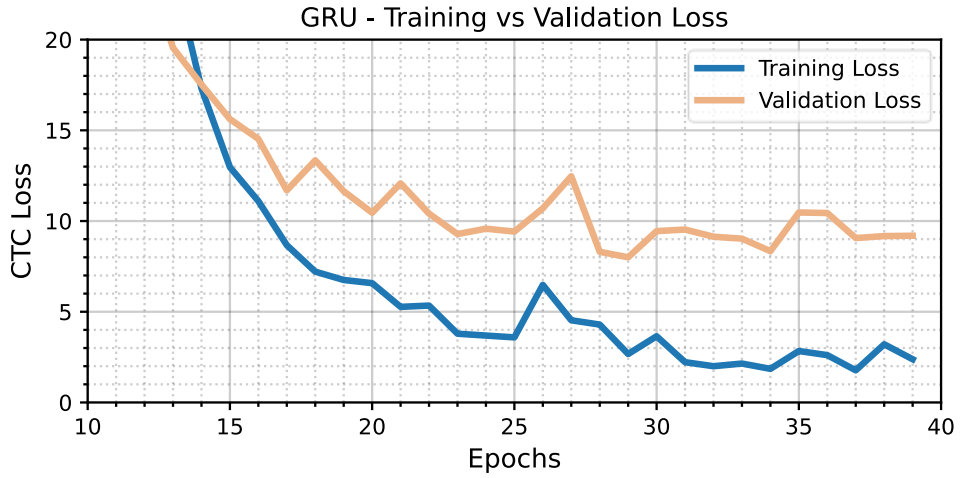


Nota: La gráfica muestra la pérdida durante 65 épocas de entrenamiento del modelo LSTM bidireccional para reconocimiento de texto, donde se distinguen fluctuaciones en la pérdida de validación y entrenamiento. Elaboración propia mediante Jupyter Notebook.

De la misma manera, durante el entrenamiento del modelo GRU bidireccional, se registró la evolución de la pérdida, presentada en la Figura 61. La gráfica muestra fluctuaciones en la pérdida de validación, con picos que alcanzan valores superiores a los de entrenamiento, mientras que la pérdida de entrenamiento se mantiene más estable.

Figura 61

Evolución de la pérdida - Modelo GRU bidireccional



Nota: La gráfica muestra la pérdida durante 65 épocas de entrenamiento del modelo LSTM bidireccional para reconocimiento de texto, donde se distinguen fluctuaciones en la pérdida. Elaboración propia mediante Jupyter Notebook.

3.4.2.6. Métricas de Evaluación

Las métricas de evaluación seleccionada fueron el CER (Character Error Rate) y el WER (Word Error Rate), y se definen en la Tabla 29. Se optó por estas métricas porque permiten cuantificar directamente la calidad del reconocimiento de texto, mientras que otras métricas como Precisión, Recall o la función de pérdida no reflejan de manera adecuada los errores de secuencia ni la correcta alineación de caracteres en tareas de reconocimiento y extracción de texto.

Tabla 29

Métricas de evaluación para reconocimiento de texto

Métrica	Descripción
CER (Character Error Rate)	Porcentaje de caracteres incorrectamente reconocidos respecto al total de caracteres.
WER (Word Error Rate)	Porcentaje de palabras incorrectamente reconocidas respecto al total de palabras.

Nota: La tabla presenta las métricas empleadas para evaluar la precisión de los modelos de reconocimiento de texto, considerando errores a nivel de caracteres y palabras. Elaboración propia.

3.4.2.7. Análisis Comparativo de Resultados

Los modelos de reconocimiento de texto evaluados presentaron un desempeño muy alto en la predicción de caracteres y palabras, como se observa en la Tabla 30. La variante con LSTM bidireccional logró una mayor precisión, alcanzando un CER de aproximadamente 0.50 % y un WER de 2.36 %. Por su parte, la variante con GRU bidireccional obtuvo un CER de 0.81 % y un WER de 3.72 %, mostrando un rendimiento ligeramente inferior.

Tabla 30

Evaluación de modelos en reconocimiento de texto

Modelo	CER (%)	WER (%)
CNN + BiLSTM bidireccional	0.50	2.36
CNN + GRU bidireccional	0.81	3.72

Nota: La tabla muestra los valores de Character Error Rate y Word Error Rate obtenidos para las variantes LSTM y GRU bidireccionales, evaluadas sobre el conjunto de prueba. Elaboración propia.

En conclusión, la variante LSTM bidireccional resultó ser la más efectiva. Un CER de 0.50 % indica que, en promedio, solo 5 de cada 1000 caracteres se predicen incorrectamente, mientras que un WER de 2.36 % refleja que aproximadamente 2 de cada 100 palabras contienen al menos un error de predicción, evidenciando una alta precisión del modelo en la transcripción de texto.

3.5. DESARROLLO DEL SISTEMA

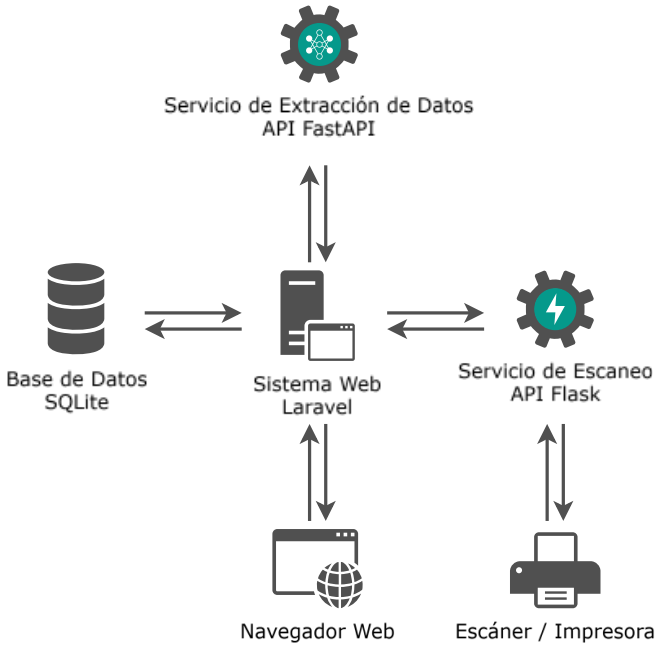
Una vez finalizado el entrenamiento y la selección de las redes neuronales, se procedió a definir la arquitectura general del sistema y un diseño preliminar de la base de datos, con el propósito de establecer la estructura sobre la cual se desarrollaría el sistema. A partir de esta base, se dio inicio al desarrollo del sistema empleando la metodología XP, ejecutando el plan de iteraciones establecido en la Tabla 18, en concordancia con las historias de usuario previamente definidas.

3.5.1. Arquitectura del Sistema

La arquitectura del sistema se definió a partir de la separación de la solución en tres componentes independientes, como se muestra en la Figura 62: la aplicación web desarrollada en Laravel, el servicio de escaneo implementado mediante Flask y el servicio de extracción de datos construido con FastAPI, este último encargado de ejecutar los modelos de redes neuronales.

Figura 62

Arquitectura del sistema



Nota: La figura muestra los componentes del sistema y el flujo de comunicación entre la aplicación web, el servicio de escaneo y el servicio de extracción de datos, así como su interacción con la base de datos y los dispositivos externos. Elaboración propia mediante draw.io.

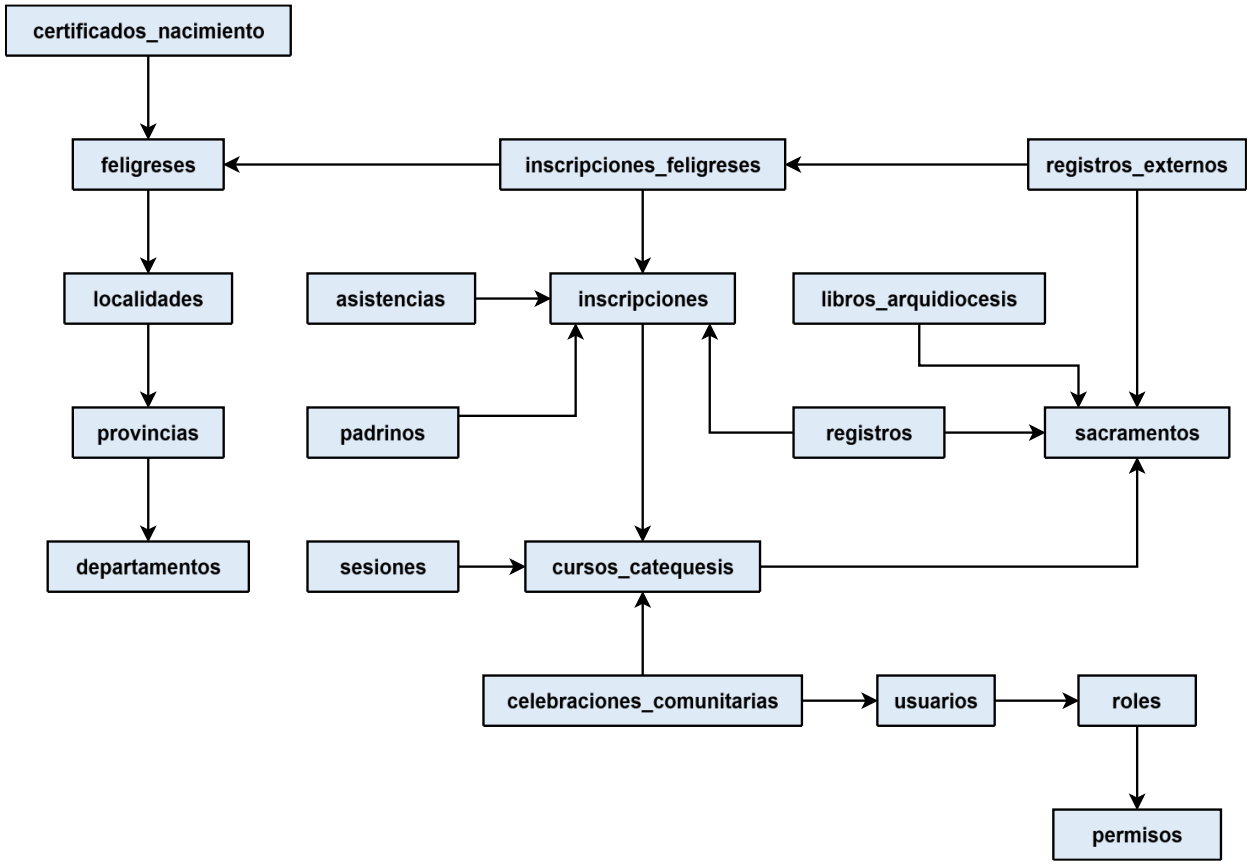
La comunicación entre los componentes del sistema se realiza mediante solicitudes HTTP. La aplicación web interactúa con el servicio de escaneo para iniciar la digitalización de documentos, mientras que este servicio mantiene la comunicación con el dispositivo físico a través de la red local y retorna las respuestas en formato JSON. Posteriormente, la imagen obtenida es enviada al servicio de extracción de datos, donde los resultados son devueltos igualmente en formato JSON.

3.5.2. Diseño de la Base de Datos

El diseño de la base de datos se realizó tomando como referencia la información gestionada en los procesos parroquiales. El modelo fue definido bajo un esquema relacional, identificando las entidades principales del dominio, entre ellas feligreses, inscripciones, sacramentos, usuarios, roles y permisos. La Figura 63 muestra una vista general de las tablas previstas y sus relaciones conceptuales, mientras que la definición detallada de estas estructuras se desarrolló de manera progresiva a lo largo de las iteraciones.

Figura 63

Estructura conceptual de la base de datos del sistema



Nota: La figura presenta una vista general de las tablas consideradas en el sistema y sus relaciones conceptuales. Elaboración propia mediante draw.io.

Para la implementación se adoptó SQLite como motor de base de datos, debido a su facilidad de despliegue, bajo requerimiento de configuración y adecuación al contexto operativo del sistema. Dado que la carga transaccional prevista es reducida, este motor ofrece un desempeño suficiente sin requerir un servidor de base de datos independiente, simplificando la administración y operación de la solución propuesta.

3.5.3. Seguridad del Sistema

Al desarrollar el sistema se tuvieron en cuenta mecanismos de seguridad orientados a proteger la información personal gestionada por la parroquia. Las medidas que se implementaron se

enfocaron en la autenticación de usuarios, el control de acceso y la protección de la base de datos frente a vulnerabilidades comunes.

3.5.3.1. Autenticación y control de acceso

La autenticación fue implementada utilizando Laravel Breeze, paquete oficial del framework Laravel que ofrece una estructura básica para el registro y el inicio de sesión de los usuarios. Este enfoque permitió integrar el sistema de autenticación propio del framework sin necesidad de desarrollar una solución personalizada, evitando posibles errores durante la implementación.

Las contraseñas fueron almacenadas utilizando funciones de hash criptográfico (bcrypt), evitando su registro en texto plano. De esta manera, el sistema compara el hash generado durante el inicio de sesión con el valor previamente almacenado en la base de datos.

El manejo de sesiones se realizó con el propio framework, impidiendo el acceso a rutas internas cuando no existe una sesión autenticada. Asimismo, se definieron roles de usuario para limitar determinadas acciones dentro del sistema, restringiendo el acceso a funcionalidades específicas según el usuario autenticado.

3.5.3.2. Protección de la base de datos

La interacción con la base de datos se realizó mediante el ORM Eloquent de Laravel, el cual utiliza consultas parametrizadas. Este mecanismo evita que los datos ingresados por el usuario formen parte directa de la estructura de la consulta SQL.

La base de datos SQLite se mantuvo fuera del directorio público del proyecto, impidiendo su acceso directo desde el navegador. Además, se establecieron relaciones y restricciones entre las tablas del modelo relacional, así como validaciones previas al almacenamiento de la información, con el fin de conservar la coherencia de los datos registrados en la base de datos.

3.5.4. Iteración 1

3.5.4.1. Planificación

La Iteración 1 está conformada por las historias de usuario: HU-01 (Acceso al sistema) y HU-02 (Gestión de roles y permisos). En la Tablas 31 se detalla la planificación de tareas para ambas

historias de usuario, donde se detallan las actividades necesarias para cumplir los criterios de aceptación de cada historia.

Tabla 31

Plan de tareas - Iteración 1

Código	Tarea	Prioridad	Descripción
IT1-T1	Autenticación	Alta	Implementar el sistema de inicio de sesión con validación de credenciales.
IT1-T2	Gestión de usuarios (crear/editar/eliminar)	Alta	Permitir al párroco administrar cuentas: registrar nuevos usuarios, modificar información y eliminar usuarios del sistema.
IT1-T3	Gestión de roles	Media	Implementar la asignación, modificación y eliminación de roles dentro del sistema.
IT1-T4	Configuración de permisos por rol	Alta	Cada usuario debe recibir un rol y sus permisos respectivos.
IT1-T5	Restricción de funcionalidades según rol	Alta	Los usuarios solo acceden a las funciones permitidas por su rol.
IT1-T6	Validación de acceso seguro	Alta	Permitir que los usuarios ingresen con correo y contraseña según su rol.
IT1-T7	Interfaz del módulo de usuarios	Media	Crear las vistas de inicio de sesión, administración de usuarios, roles y permisos.
IT1-T8	Interfaz del módulo de roles	Alta	Validar campos requeridos, formato de correo y contraseñas seguras.
IT1-T9	Pruebas funcionales de acceso y roles	Alta	Verificar que todas las funciones de usuario funcionen correctamente.

Nota: La tabla muestra las tareas planificadas y ejecutadas durante la Iteración 1 correspondientes a las historias asignadas. Elaboración propia.

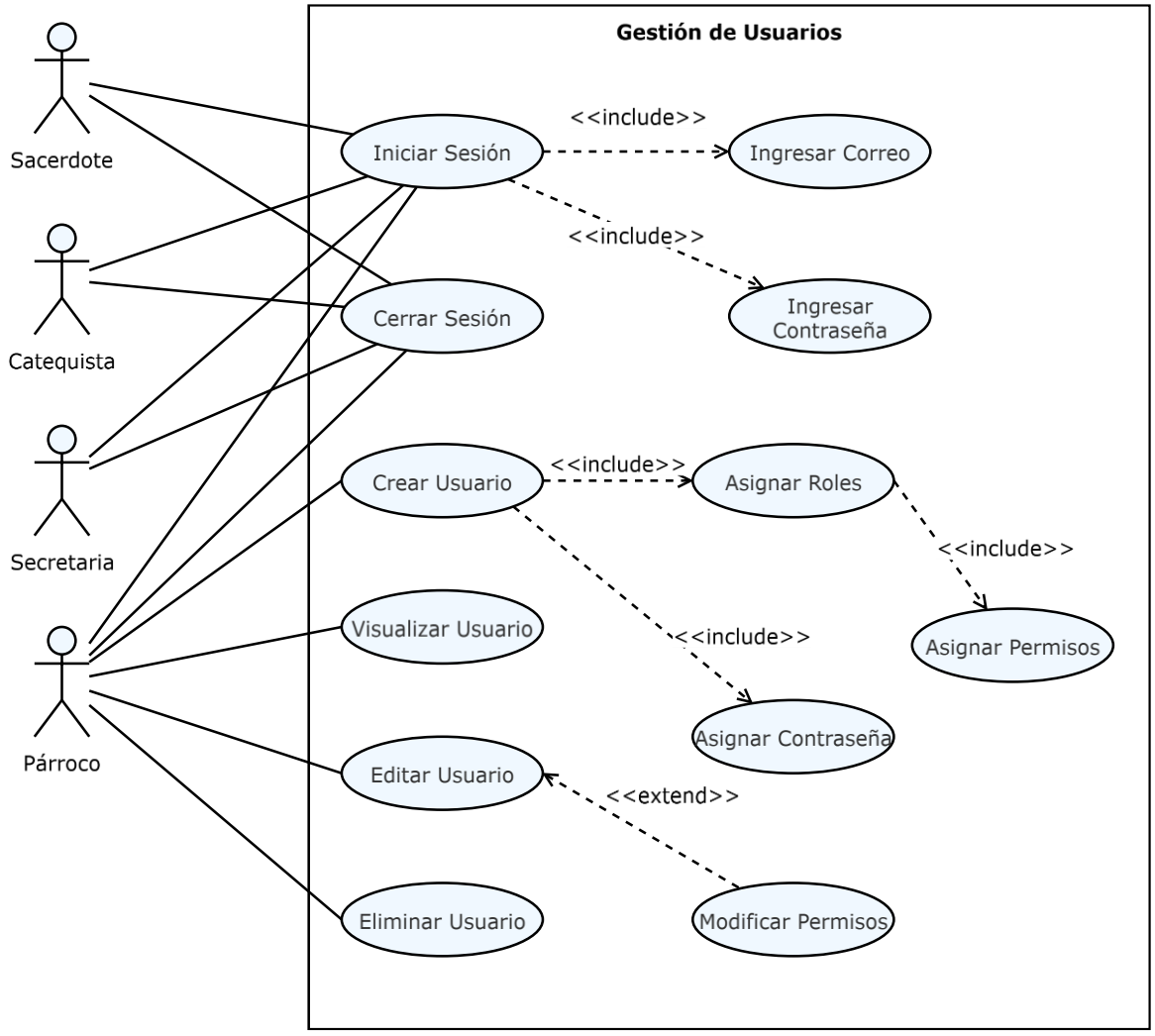
3.5.4.2. Diseño

Para la gestión de usuarios, se elaboró el diagrama de casos de uso que se muestra en la Figura 64, el cual contempla dos áreas principales: la sesión de usuario y la gestión de usuarios. En la primera, ambos actores, Administrador y los usuarios que no tienen privilegios de administración (Secretaría, Sacerdote o Catequista), comparten las funcionalidades de Iniciar sesión y Cerrar sesión, siendo obligatorio ingresar las credenciales correspondientes. En la segunda, exclusiva del Administrador, se incluyen las operaciones de Crear, Visualizar, Editar y Eliminar usuario, donde la creación de un usuario implica obligatoriamente asignar un rol y establecer una

contraseña, y la asignación de roles incluye la definición de los permisos correspondientes. La edición de un usuario puede implicar la modificación de permisos, acción que se realiza reasignando un rol al usuario o modificando los permisos del rol.

Figura 64

Diagrama de casos de uso - Gestión de usuarios



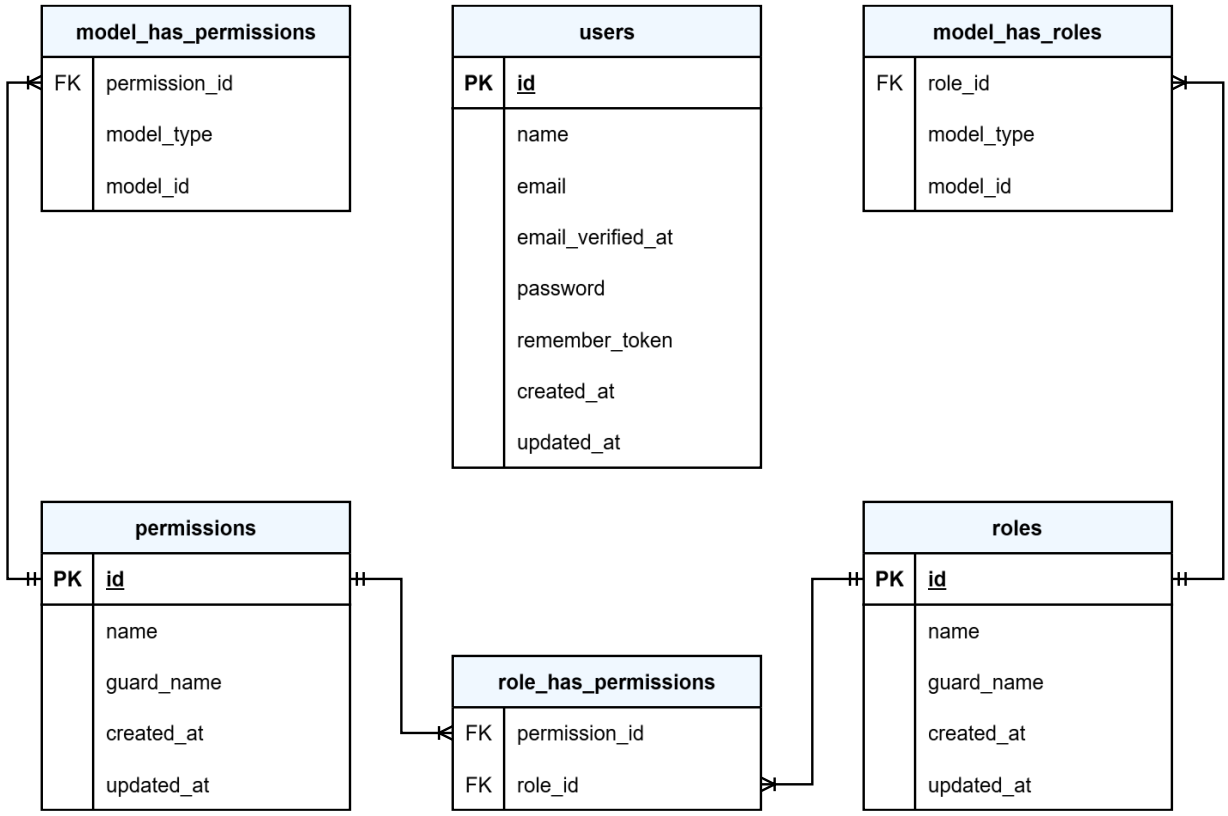
Nota: En el diagrama de casos de uso se representan las principales funcionalidades del sistema relacionadas con la gestión de usuarios, mostrando la interacción entre los distintos roles, incluyendo Administrador, Secretaría, Sacerdote y Catequista. Elaboración propia mediante draw.io.

Para el manejo de los datos, fue necesario representar cómo se almacenaría la información en el sistema, por lo que se elaboró el diagrama relacional de la base de datos, como se muestra en la Figura 65. En él se destacan las entidades principales, incluyendo usuarios, roles y

permisos para la gestión de usuarios, relacionadas mediante tablas intermedias que permiten asignar roles y permisos de forma flexible.

Figura 65

Diagrama relacional - Gestión de usuarios

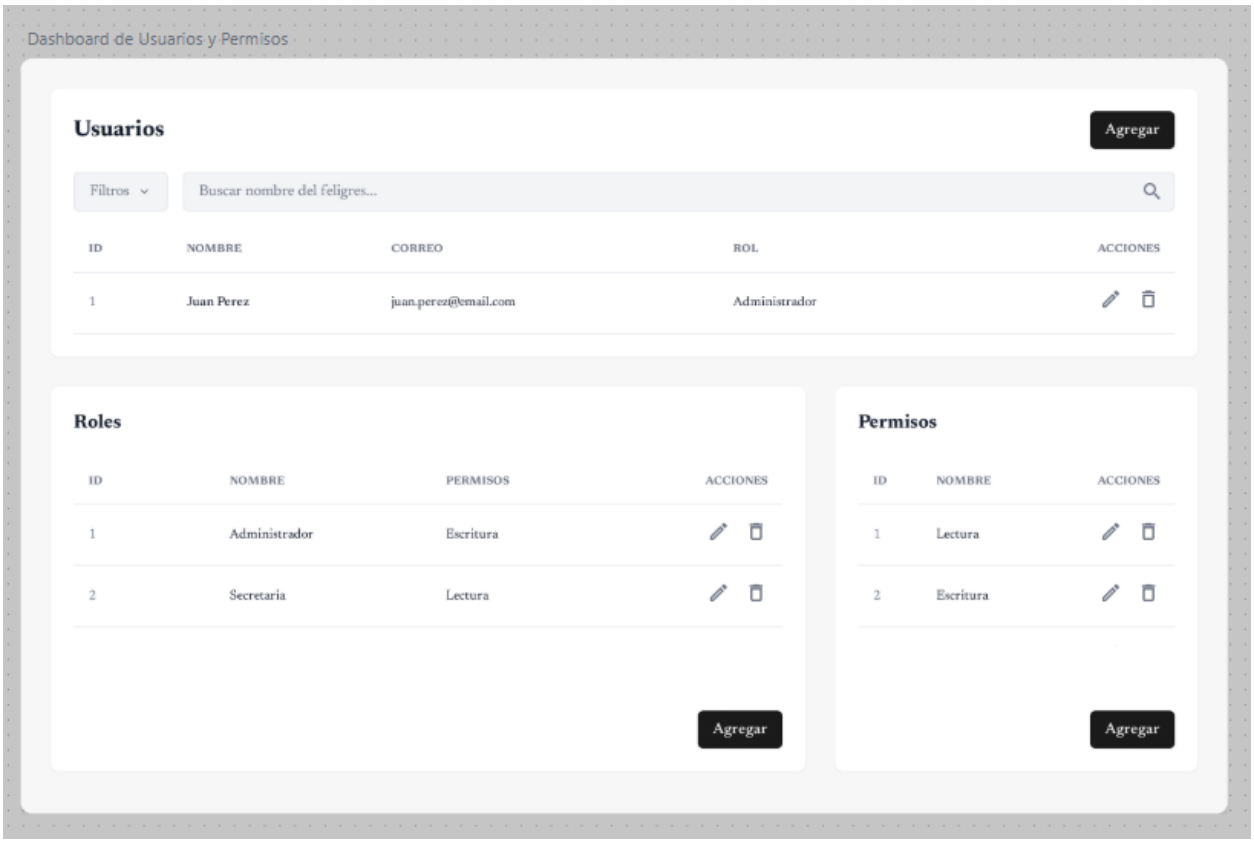


Nota: El diagrama muestra la estructura de la base de datos y las relaciones entre usuarios, roles y permisos. Elaboración propia mediante draw.io.

Finalmente se elaboraron los wireframes correspondientes a la primera iteración, con el objetivo de definir de manera visual la estructura, distribución de elementos y la navegación entre pantallas. En la Figura 66 se muestra el wireframe de la gestión de usuarios, roles y permisos, permitiendo identificar cómo se organizarán las funcionalidades y cómo los usuarios podrán interactuar con ellas una vez finalizada la fase de desarrollo.

Figura 66

Wireframe - Gestión de usuarios, roles y permisos



Nota: La figura muestra la distribución de los elementos en la interfaz para la gestión de usuarios, roles y permisos. Elaboración propia mediante Stitch.

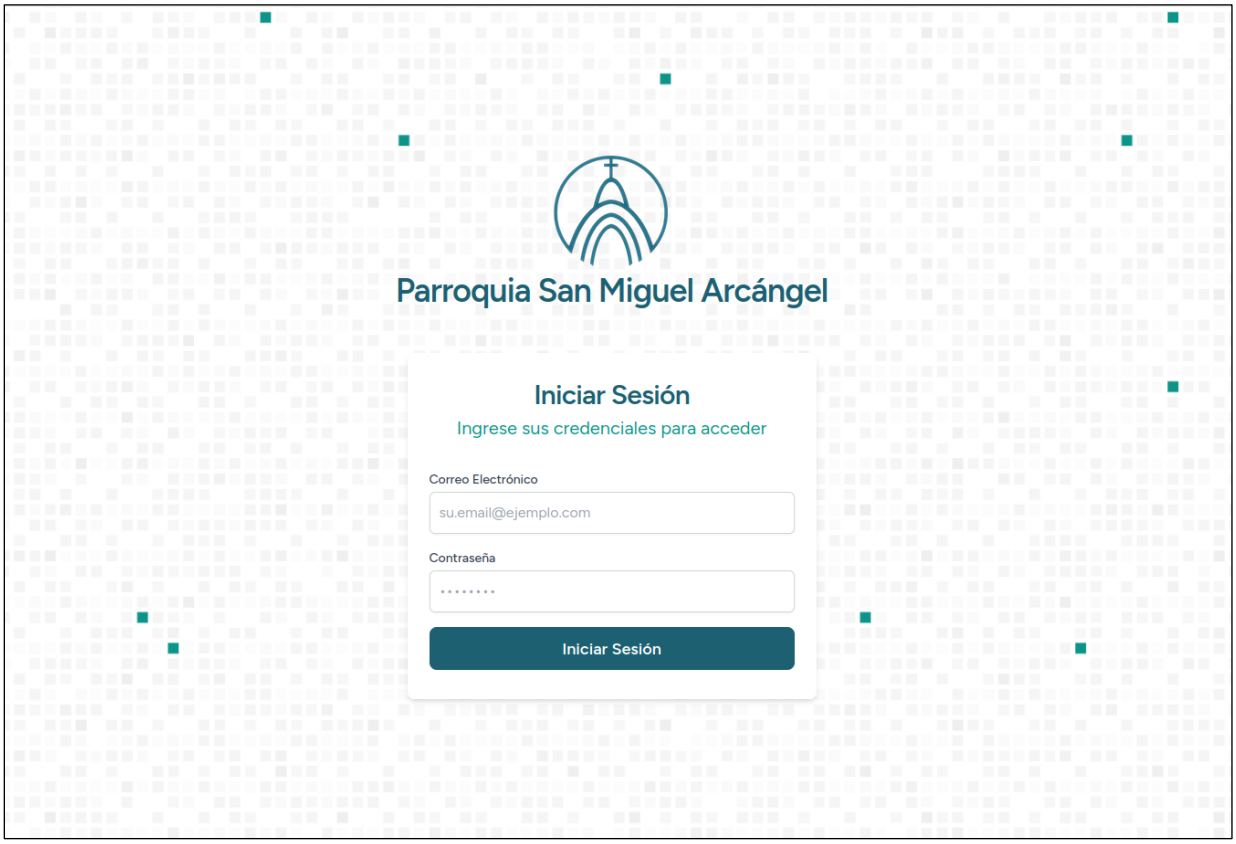
3.5.4.3. Desarrollo

Durante la fase de desarrollo se implementaron las funcionalidades de inicio de sesión y la gestión de usuarios, roles y permisos. Para una autenticación segura, se utilizó la librería Breeze, que maneja las sesiones y protege la información de los usuarios. Asimismo, se validó que los usuarios no puedan acceder a secciones restringidas del sistema web sin haber iniciado sesión, incluso si intentan ingresar directamente a través de la URL

Para el inicio de sesión, como se observa en la Figura 67, se diseñó una interfaz que incluye el logotipo de la institución, los campos de correo electrónico y contraseña, y un botón de acceso. Además, se implementó hashing de contraseñas para asegurar que estas no sean visibles en la base de datos ni durante la autenticación.

Figura 67

Pantalla de inicio de sesión del sistema

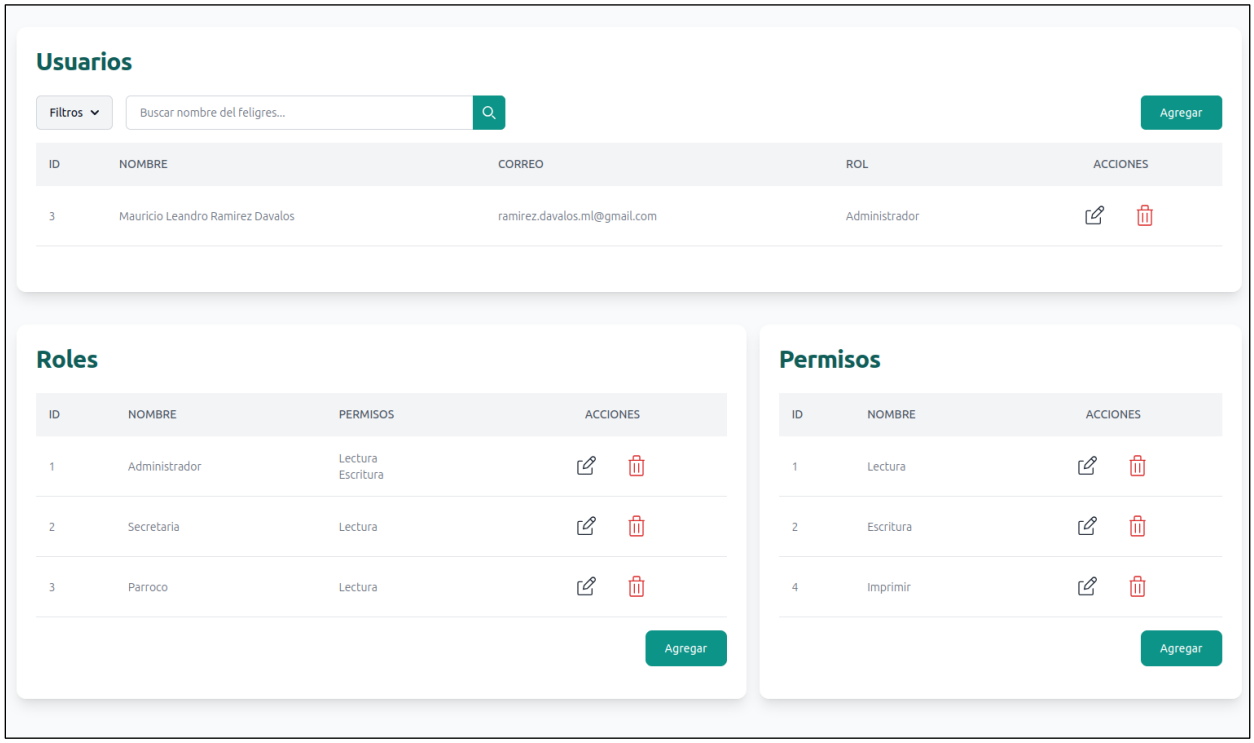


Nota: La figura muestra la interfaz de inicio de sesión del sistema, donde los usuarios ingresan su correo electrónico y contraseña para acceder. Elaboración propia.

De la misma manera, en la Figura 68 se muestra la pantalla de gestión de usuarios, donde en la parte superior central se encuentra una tabla principal titulada "Usuarios", que permite visualizar los registros existentes. Cada registro incluye los campos ID, nombre, correo, rol y acciones para editar o eliminar usuarios. Debajo de la tabla de usuarios, se localizan las secciones "Roles" y "Permisos", que permiten gestionar los roles disponibles y los permisos asociados a cada uno, utilizando la librería Spatie para asegurar una administración segura y flexible. Cada tabla incluye un botón de "Agregar" para incorporar nuevos elementos, así como opciones para editar o eliminar registros existentes.

Figura 68

Pantalla de gestión de usuarios, roles y permisos



Nota: La figura muestra la interfaz del sistema utilizada para administrar usuarios, asignar roles y definir permisos según el perfil de cada usuario. Elaboración propia.

3.5.4.4. Pruebas

Durante la Iteración 1 se realizaron pruebas funcionales y de seguridad para asegurar que los distintos módulos del sistema operen según lo esperado. Estas pruebas se enfocaron en validar los procesos de gestión de usuarios, roles y permisos, verificando que funcionalidades clave como inicio de sesión, creación de usuarios, asignación de roles y protección frente a accesos no autorizados funcionen correctamente.

Los casos de prueba se presentan en las Tablas 32 a 36, donde se detalla la estructura de cada prueba, incluyendo precondiciones, pasos a seguir, datos utilizados, postcondiciones, resultados obtenidos y observaciones.

Tabla 32

Caso de prueba – Inicio de sesión

<table>
<tr><th colspan="5">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</th></tr>
<tr><td colspan="2">Caso de prueba: CP-01</td><td colspan="3">Caso de Uso: Gestión de usuarios</td></tr>
<tr><td colspan="5">Descripción del caso de prueba: Verificar que un usuario registrado pueda acceder al sistema ingresando un correo y contraseña válidos.</td></tr>
<tr><td colspan="5">Precondiciones</td></tr>
<tr><td colspan="5">– El usuario debe existir en la base de datos.
– El usuario debe tener correo y contraseña válidos.</td></tr>
<tr><td colspan="5">Pasos de la prueba</td></tr>
<tr><td colspan="5">1. Ingresar a la pantalla de inicio de sesión.
2. Introducir correo en el campo Usuario.
3. Introducir contraseña en el campo Contraseña.
4. Presionar el botón “Iniciar sesión”.</td></tr>
<tr><td colspan="2">Datos de entrada</td><td rowspan="2">Respuesta esperada del sistema</td><td colspan="2">Coincide</td></tr>
<tr><td>Campo</td><td>Valor</td><td>Si</td><td>No</td></tr>
<tr><td>Usuario</td><td>ramirez.davalos.ml@gmail.com</td><td>Verificar que el correo exista en los usuarios en la base de datos.</td><td>X</td><td></td></tr>
<tr><td>Contraseña</td><td>123456</td><td>Validar que la contraseña coincida con la registrada para ese usuario.</td><td>X</td><td></td></tr>
<tr><td colspan="5">Postcondiciones</td></tr>
<tr><td colspan="5">– El sistema mantiene activa la sesión del usuario autenticado.</td></tr>
<tr><td colspan="5">Resultados de la prueba</td></tr>
<tr><td colspan="3">Observaciones: Funcionamiento correcto del proceso de autenticación.</td><td colspan="2">Veredicto: Pasó</td></tr>
</table>

Nota: La tabla presentada corresponde al caso de prueba CP-01, relacionado con el proceso de inicio de sesión y ejecutado durante la Iteración 1. Elaboración propia.

Tabla 33

Caso de prueba – Crear usuario

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA	
Caso de prueba: CP-02	**Caso de Uso:** Gestión de usuarios
Descripción del caso de prueba: Validar que se pueda registrar un usuario ingresando nombre, correo, contraseña y rol.	
Precondiciones	
– Haber iniciado sesión. – Tener permisos sobre el módulo de usuarios. – Debe existir al menos un rol registrado en el sistema.	

Pasos de la prueba				
1. Ingresar al módulo “Usuarios”. 2. Presionar el botón “Agregar usuario”. 3. Completar los campos obligatorios. 4. Seleccionar un rol. 5. Presionar “Guardar”.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Nombre	Juan Pérez	Validar que el nombre no esté vacío.	X	
Correo	juan.perez@gmail.com	Verificar que el correo no exista previamente en la base de datos.	X	
Contraseña	123456	Validar que cumpla con la política de seguridad del sistema (longitud, caracteres, hash)	X	
Confirmar contraseña	123456	Validar que coincida con el campo contraseña.	X	
Rol	Sacerdote	Verificar que el rol exista en la base de datos y asignarlo correctamente.	X	
Postcondiciones				
– El usuario aparece en la tabla principal. – El usuario queda registrado en la base de datos con contraseña cifrada y rol asignado.				
Resultados de la prueba				
Observaciones: Spatie asigna correctamente el rol.		**Veredicto:** Pasó		

Nota: La tabla presentada corresponde al caso de prueba CP-02, relacionado con el proceso de creación de usuarios. Elaboración propia.

Tabla 34

Caso de prueba – Inyección SQL en login

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA	
Caso de prueba: CP-03	**Caso de Uso:** Gestión de usuarios
Descripción del caso de prueba: Comprobar que el sistema no sea vulnerable a inyecciones SQL en los campos de usuario y contraseña.	
Precondiciones	
– El sistema debe estar operativo.	
Pasos de la prueba	
1. Ingresar en el campo Usuario: ' OR '1'='1 2. Ingresar cualquier contraseña. 3. Presionar “Iniciar sesión”.	

Datos de entrada		Respuesta esperada del sistema	Coincide	
Campo	**Valor**		**Si**	**No**
Usuario	' OR '1'='1	Acceso denegado, el sistema no permite login.	X	
Contraseña	cualquier valor	Acceso denegado, el sistema no permite login.	X	
Postcondiciones				
– El sistema sigue protegido frente a inyección SQL.				
Resultados de la prueba				
Observaciones: La autenticación valida correctamente los inputs.		**Veredicto:** Pasó		

Nota: La tabla muestra el caso de prueba enfocado en la validación de los datos de entrada para prevenir ataques de inyección SQL. Elaboración propia.

Tabla 35

Caso de prueba – Acceso sin autenticación

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA				
Caso de prueba: CP-04	**Caso de Uso:** Gestión de usuarios			
Descripción del caso de prueba: Verificar que un usuario no autenticado no pueda acceder a módulos protegidos mediante URL directa.				
Precondiciones				
– El sistema debe estar operativo. – El usuario no debe haber iniciado sesión.				
Pasos de la prueba				
1. Abrir una URL interna del sistema. 2. Intentar acceder a los módulos protegidos sin iniciar sesión.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
URL	/feligreses	Redirige automáticamente a la página de inicio de sesión	X	
URL	/cursos	Redirige automáticamente a la página de inicio de sesión	X	
Postcondiciones				
– El sistema mantiene protegidas las rutas restringidas.				
Resultados de la prueba				
Observaciones: El sistema bloquea el acceso sin autenticación.		**Veredicto:** Pasó		

Nota: Se presenta el caso de prueba para el acceso a módulos mediante URL sin autenticación. Elaboración propia.

Tabla 36

Caso de prueba – Restricción de permisos por rol

<table>
<tr><th colspan="5">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</th></tr>
<tr><td colspan="2">Caso de prueba: CP-05</td><td colspan="3">Caso de Uso: Gestión de usuarios</td></tr>
<tr><td colspan="5">Descripción del caso de prueba: Verificar que un usuario con rol no pueda ejecutar acciones ajenas a su rol.</td></tr>
<tr><th colspan="5">Precondiciones</th></tr>
<tr><td colspan="5">– Usuario autenticado con rol “Sacerdote”.
– Existen roles y permisos definidos en el sistema.</td></tr>
<tr><th colspan="5">Pasos de la prueba</th></tr>
<tr><td colspan="5">1. Iniciar sesión con usuario de rol “Sacerdote”.
2. Intentar acceder a funcionalidades del párroco (crear/editar/eliminar usuarios).</td></tr>
<tr><th colspan="2">Datos de entrada</th><th rowspan="2">Respuesta esperada del sistema</th><th colspan="2">Coincide</th></tr>
<tr><th>Campo</th><th>Valor</th><th>Si</th><th>No</th></tr>
<tr><td>Rol</td><td>Sacerdote</td><td>No permite acceder a funciones del párroco.</td><td>X</td><td></td></tr>
<tr><td>Función</td><td>Crear usuario</td><td>La opción “Crear usuario” no debe mostrarse en el menú ni la interfaz cuando el usuario es sacerdote.</td><td>X</td><td></td></tr>
<tr><th colspan="5">Postcondiciones</th></tr>
<tr><td colspan="5">– El usuario limitado solo puede interactuar con las funciones permitidas por su rol.</td></tr>
<tr><th colspan="5">Resultados de la prueba</th></tr>
<tr><td colspan="3">Observaciones: Restricciones aplicadas correctamente, el control de acceso funciona.</td><td colspan="2">Veredicto: Pasó</td></tr>
</table>

Nota: La tabla presentada corresponde al caso de prueba que verifica la restricción de acciones de los usuarios según los permisos asignados a sus roles. Elaboración propia.

3.5.5. Iteración 2

3.5.5.1. Planificación

La Iteración 2 está conformada por las historias de usuario HU-03 (Registro de feligreses), HU-04 (Integración de redes neuronales) y HU-05 (Integración del servicio de escaneo), cuya finalidad es permitir el registro, edición y gestión de los feligreses, así como automatizar la identificación y extracción de información desde documentos escaneados. En la Tabla 37 se presentan las tareas definidas para cumplir los criterios de aceptación.

Tabla 37

Plan de tareas - Iteración 2

Código	Tarea	Prioridad	Descripción
IT2-T1	Registro manual de feligreses	Alta	Implementar el formulario para ingresar datos personales: nombres, fecha y lugar de nacimiento, libro, partida y oficialía.
IT2-T2	Validación de formularios de registro	Alta	Aplicar validaciones de campos obligatorios, formatos de fecha y consistencia de datos al momento del registro.
IT2-T3	Edición de registros de feligreses	Alta	Permitir la modificación de datos ya registrados, respetando los permisos del rol asignado.
IT2-T4	Gestión de documentos adjuntos	Media	Permitir asociar documentación digitalizada al registro del feligrés (imágenes).
IT2-T5	Detección automática de información según tipo de documento	Alta	Integrar el modelo de detección para los documentos y su información correspondiente.
IT2-T6	Extracción automática de información	Alta	Integrar el modelo de extracción (OCR) para la digitalización de la información detectada.
IT2-T7	Previsualización y corrección de datos extraídos	Alta	Mostrar los datos detectados para que el usuario revise, corrija y confirme antes de registrarlos.
IT2-T8	Carga automática de campos sugeridos	Media	Completar automáticamente los campos del formulario con la información extraída.
IT2-T9	Interfaz de registro asistido	Media	Diseñar las vistas para mostrar simultáneamente el documento escaneado y los campos sugeridos para edición.
IT2-T10	Pruebas funcionales del registro y de las redes neuronales.	Alta	Validar el registro manual, edición, reconocimiento de documentos y extracción de información.

Nota: La tabla muestra las tareas planificadas y ejecutadas durante la Iteración 2 correspondientes a las historias asignadas. Elaboración propia.

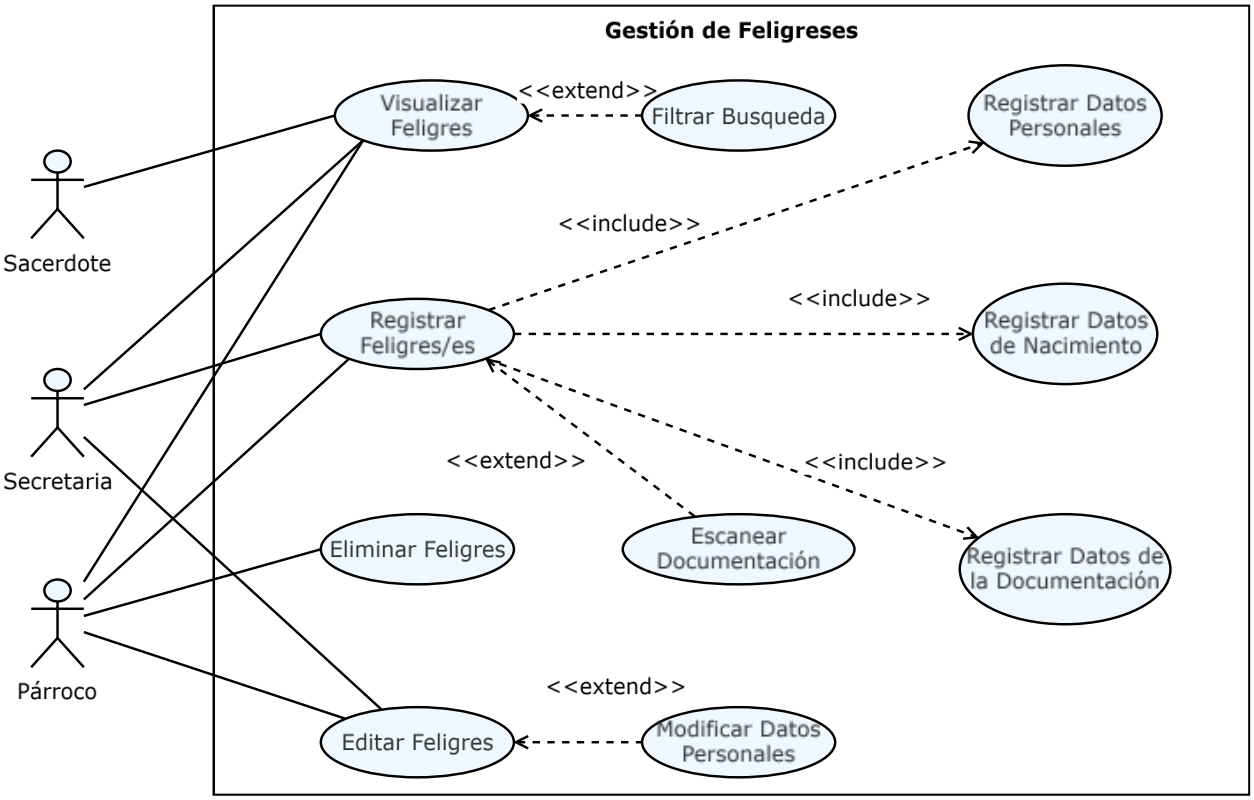
3.5.5.2. Diseño

Para la gestión de feligreses se elaboró un diagrama de casos de uso, como se muestra en la Figura 69, que modela las interacciones de los actores clave con el sistema: Párroco, Secretaria y Sacerdote, quienes acceden a las funcionalidades según sus responsabilidades. Cada actor tiene un nivel de acceso diferenciado: la Secretaria y el Párroco pueden registrar y editar

feligreses, mientras que el Sacerdote tiene privilegios limitados a la visualización y eliminación de registros. La función de Visualizar Feligreses permite consultar información detallada y puede complementarse con filtros por nombre, padres o fecha de nacimiento, y la acción de registrar nuevos feligreses puede incluir opcionalmente el escaneo de documentación en caso de no realizar el registro manualmente.

Figura 69

Diagrama de casos de uso - Gestión de feligreses

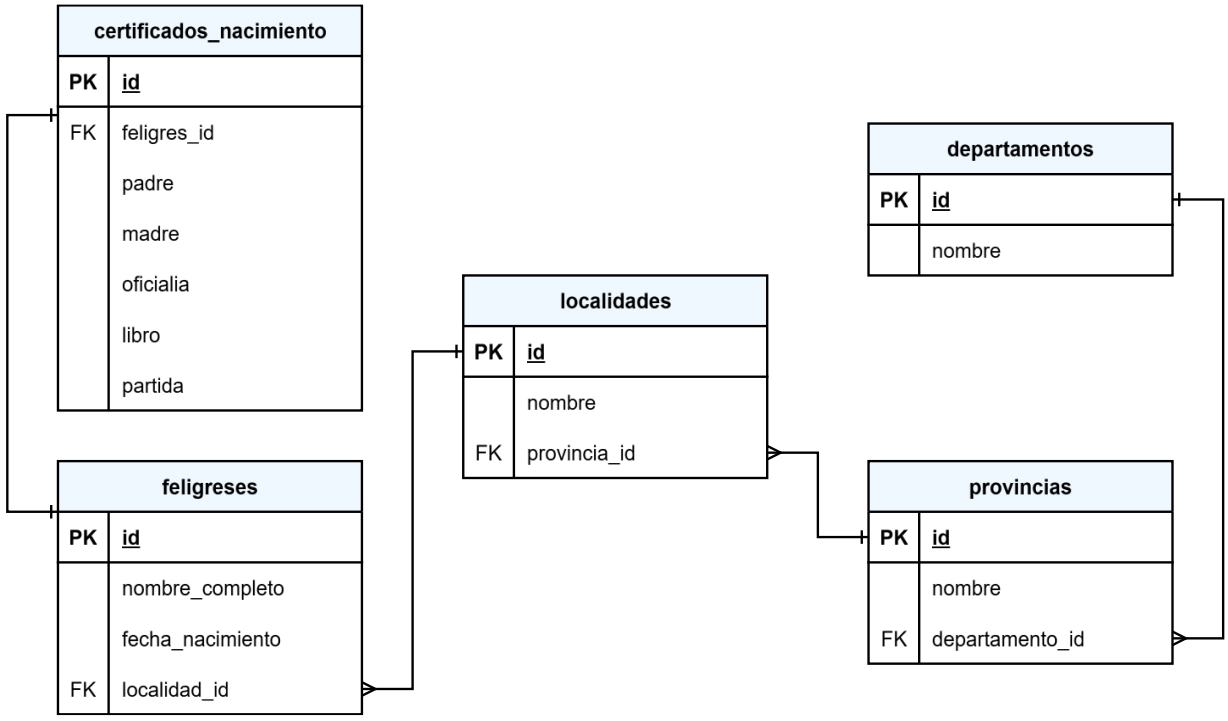


Nota: El diagrama ilustra las funcionalidades esenciales del sistema para la gestión de feligreses, destacando cómo cada actor interactúa con las diferentes funciones según su rol. Se representan las acciones de registrar, editar, visualizar y eliminar registros, así como las opciones complementarias de filtrado y escaneo de documentación. Elaboración propia mediante draw.io.

De la misma manera, para el manejo de los datos, fue necesario representar cómo se almacenaría la información en el sistema, por lo que se elaboró el diagrama relacional de la base de datos, como se muestra en la Figura 70. En él se destacan las entidades principales, incluyendo feligreses, certificados_nacimiento y localidades para la gestión de feligreses.

Figura 70

Diagrama relacional - Gestión de feligreses

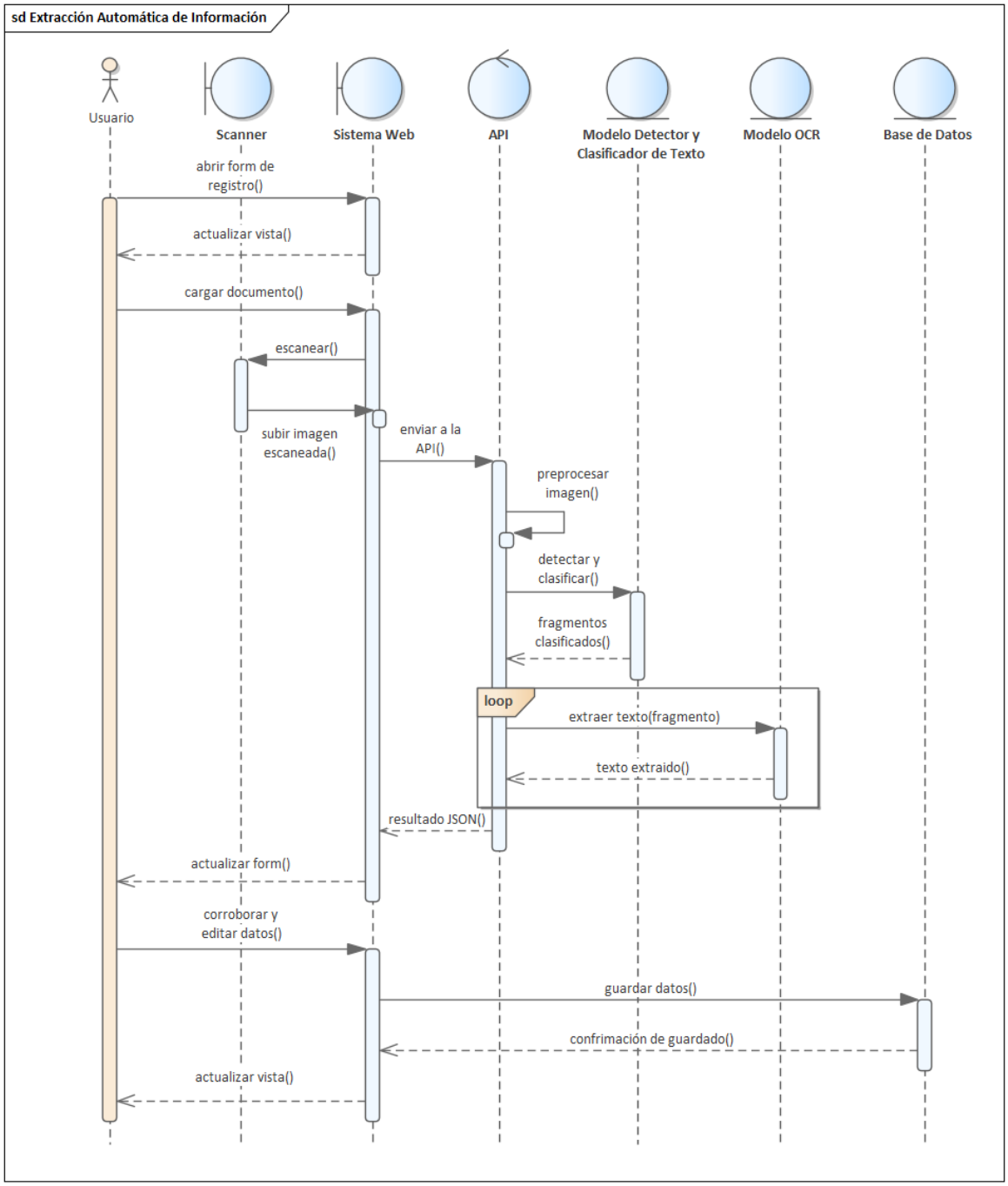


Nota: El diagrama muestra la estructura de la base de datos y las relaciones entre usuarios, roles y permisos. Elaboración propia mediante draw.io.

Para la integración de los modelos de detección y OCR, se elaboró el diagrama de secuencia que se muestra en la Figura 71, el cual describe la interacción entre el Usuario, el Sistema Web, la API y los modelos de inteligencia artificial. El proceso inicia cuando el Usuario abre el formulario de registro y carga un documento; el Sistema Web envía la información a la API, que solicita al Modelo Detector y Clasificador de Texto identificar las secciones relevantes y, en un ciclo iterativo, envía cada fragmento al Modelo OCR para extraer el texto. Finalmente, la API consolida la información y la devuelve al Sistema Web, donde el Usuario puede verificar y corregir los datos antes de almacenarlos en la Base de Datos.

Figura 71

Diagrama de secuencia - Procesamiento automático de documentos

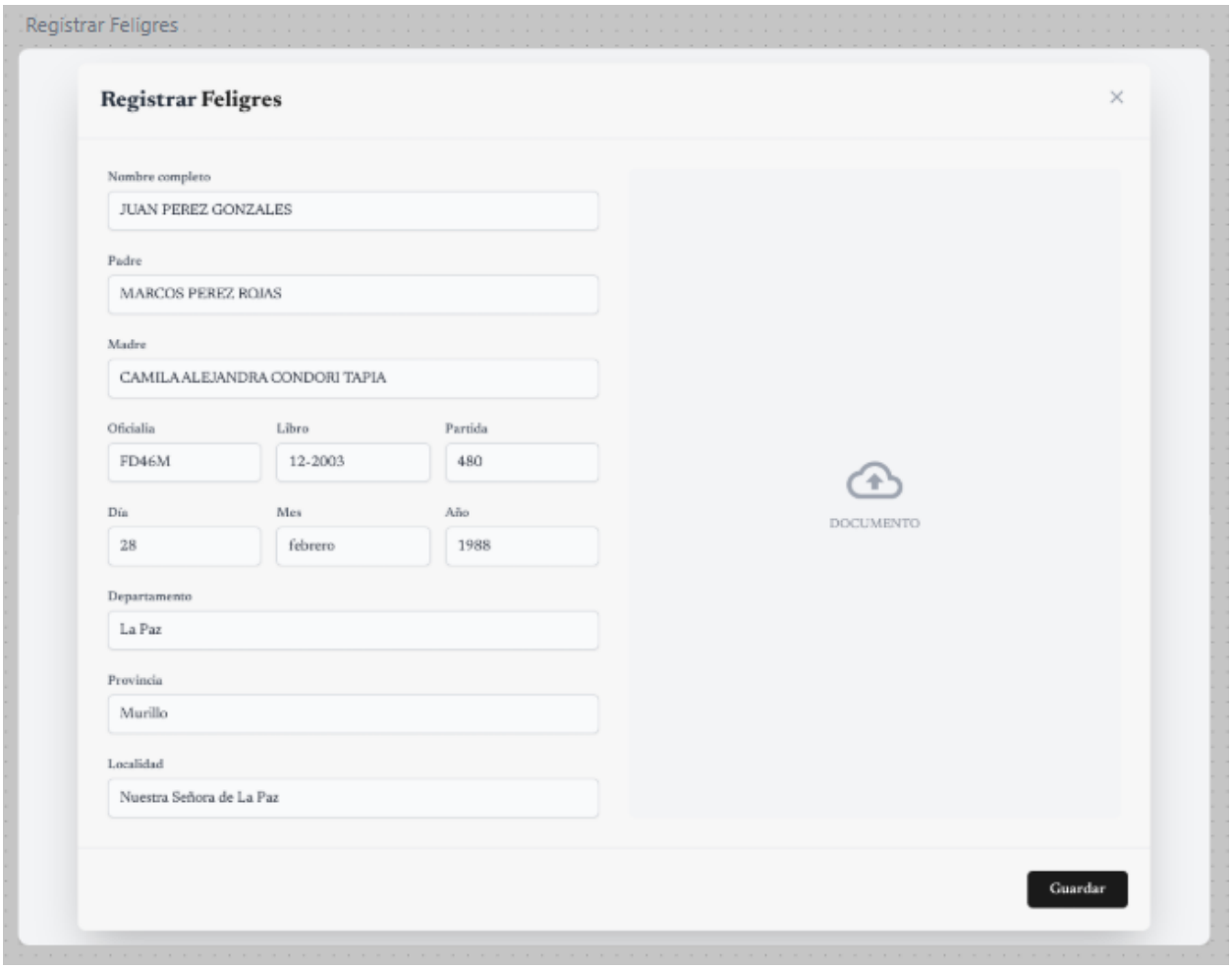


Nota: El diagrama muestra la secuencia de procesamiento de documentos en el sistema web, destacando la interacción entre la interfaz, el scanner y la API y los modelos de redes neuronales para extraer y devolver la información al usuario. Elaboración propia mediante Enterprise Architect.

Finalmente se elaboraron los wireframes correspondientes a la segunda iteración, con el objetivo de definir de manera visual la estructura, distribución de elementos y la navegación entre pantallas. En la Figura 72 se muestra el wireframe para el registro de un feligrés, permitiendo identificar los campos de entrada manual junto con el panel lateral que permitirá la visualización del documento escaneado.

Figura 72

Wireframe – Registro de feligreses



Nota: La figura muestra la distribución de los elementos en la interfaz para el registro de un feligrés. Elaboración propia mediante Stitch.

3.5.5.3. Desarrollo

Durante la fase de desarrollo se implementó un servicio en FastAPI para gestionar la comunicación entre el sistema web y los modelos entrenados. En la Figura 73 se presenta una de las funciones incluidas en la API, la cual ejecuta un proceso en dos etapas: primero, emplea el modelo de detección de objetos para identificar y recortar las regiones relevantes de la imagen; posteriormente, aplica el modelo de OCR sobre cada fragmento recortado para extraer el texto contenido. Los resultados de cada detección incluyen coordenadas, puntuación de confianza, clase detectada, texto extraído y la ruta del archivo recortado.

Figura 73

Función de procesamiento de documentos en FastAPI

```python
def predict(image_path):
    results = yolo_model(image_path)
    output = []
    img = cv2.imread(image_path)
    crop_count = 0
    for r in results:
        boxes = r.boxes.xyxy.cpu().numpy()
        scores = r.boxes.conf.cpu().numpy()
        class_ids = r.boxes.cls.cpu().numpy().astype(int)
        for b, s, c in zip(boxes, scores, class_ids):
            x1, y1, x2, y2 = map(int, b)
            crop = img[y1:y2, x1:x2]
            crop_filename = os.path.join(CROP_DIR, f"crop_{crop_count}.jpg")
            cv2.imwrite(crop_filename, crop)
            crop_count += 1
            text = single_sample_prediction(ocr_model, crop_filename)
            output.append({
                "box": [x1, y1, x2, y2],
                "score": float(s),
                "class_id": int(c),
                "class_name": CLASSES[int(c)],
                "text": text,
                "crop_file": crop_filename
            })
    return output
```

Nota: La figura muestra la función implementada en la API que permite detectar automáticamente regiones de interés en los documentos y extraer el texto mediante OCR. Elaboración propia mediante Jupyter Notebook.

Para habilitar la comunicación entre el sistema web y el escáner, fue necesario desarrollar un programa, mostrado en la Figura 74, que permite al sistema web enviar una solicitud de escaneo. Una vez completado el escaneo, el programa devuelve la imagen al sistema web, que luego la envía a la API correspondiente para su procesamiento.

Figura 74

Servicio de escaneo Flask-Laravel

```python
from flask import Flask, jsonify
from flask_cors import CORS
import subprocess, requests, os

UPLOAD_URL = "http://127.0.0.1:8001/api/scan/upload"
SCAN_FILE = "/tmp/scan.jpg"

app = Flask(__name__)
CORS(app)

@app.route('/scan', methods=['POST'])
def scan():
    try:
        subprocess.run([
            "scanimage","--format=jpeg","--mode=Gray",
            "--resolution","300","--output-file",SCAN_FILE
        ], check=True)

        with open(SCAN_FILE, "rb") as f:
            resp = requests.post(UPLOAD_URL, files={"archivo": f})

        os.remove(SCAN_FILE)

        try: data = resp.json()
        except: data = resp.text

        return jsonify({"mensaje":"Escaneo completado","respuesta_laravel":data})

    except Exception as e:
        return jsonify({"error":str(e)}), 500

if __name__ == "__main__":
    app.run(host="0.0.0.0", port=9000)
```

Nota: La figura el programa que permite la comunicación entre el sistema web y el escáner: recibe la solicitud de escaneo, obtiene la imagen resultante y la devuelve al sistema web. Elaboración propia mediante Jupyter Notebook.

Una vez implementada la API, establecida la conexión con el sistema web y el scanner, se adaptaron los formularios de registro, como se observa en la Figura 75, de manera que los

campos se completen automáticamente con los datos extraídos por la API. Además, se incorporó un panel de visualización del documento, que permite al usuario corroborar y, en caso necesario, corregir la información antes de guardarla en la Base de Datos.

Figura 75

Formulario de registro con datos extraídos automáticamente



Nota: La figura muestra la interfaz del formulario de registro de feligreses adaptada para recibir los datos extraídos automáticamente por la API con los modelos entrenados. Elaboración propia.

3.5.5.4. Pruebas

Durante la Iteración 2 se realizaron pruebas funcionales enfocadas en validar tanto el registro manual de información de feligreses como el registro automatizado mediante el escaneo de documentos y su procesamiento a través de las redes neuronales integradas. Los casos de prueba correspondientes se presentan en las Tablas 38 a 40.

Tabla 38

Caso de prueba – Registro manual de feligreses

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA				
Caso de prueba: CP-06	**Caso de Uso:** Gestión de feligreses			
Descripción del caso de prueba: Validar que el sistema registre un feligrés manualmente, verificando cada campo obligatorio.				
Precondiciones				
– Usuario autenticado como Secretaria o Párroco. – El usuario debe contar con la documentación personal para registrar la información.				
Pasos de la prueba				
1. Abrir el formulario de registro de feligreses. 2. Completar todos los campos obligatorios: nombres, fecha y lugar de nacimiento, libro, partida, oficialía y demás. 3. Presionar “Guardar”.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Nombre completo	EUGENIO ROJAS CUBA	Validar que el nombre esté y aceptar solo caracteres permitidos.	X	
Padre	ANGELINO ROJAS GUTIERREZ	Validar que el padre esté y aceptar solo caracteres permitidos.	X	
Madre	SANTUSA CUBA CONDORI	Validar que la madre esté y aceptar solo caracteres permitidos.	X	
Oficialía	D4SM	Validar que la oficialía esté y aceptar solo caracteres permitidos.	X	
Libro	12-2003	Validar que el libro no se encuentre vacío.	X	
Partida	100	Validar que la partida no se encuentre vacía.	X	
Día	11	Validar que el día sea un valor numérico permitido (1–31).	X	
Mes	Enero	Validar que el mes sea válido.	X	
Año	1988	Validar que el año sea numérico.	X	
Departamento	La Paz	Validar que el departamento no se encuentre vacío y aceptar solo caracteres permitidos.	X	
Provincia	Murillo	Validar que la provincia no se encuentre vacía y aceptar solo caracteres permitidos.	X	

Localidad	Nuestra Señora de La Paz	Validar que la localidad no se encuentre vacía y aceptar solo caracteres permitidos.	X	
Postcondiciones				
– El feligrés queda registrado en la base de datos.				
Resultados de la prueba				
Observaciones: Registro manual completado correctamente.			**Veredicto:** Pasó	

Nota: La tabla presentada corresponde al caso de prueba CP-06, relacionado con el registro manual de feligreses y ejecutado durante la Iteración 2. Elaboración propia.

Tabla 39

Caso de prueba – Escaneo de documentos

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA				
Caso de prueba: CP-07		**Caso de Uso:** Gestión de feligreses		
Descripción del caso de prueba: Validar que el sistema pueda solicitar un escaneo al hardware y recibir la imagen correctamente.				
Precondiciones				
– Escáner conectado y configurado. – La API desarrollada con Flask para el escaneo de documentos debe estar activa. – Usuario autenticado con permisos de registro de feligreses.				
Pasos de la prueba				
1. Abrir el formulario de registro de feligreses. 2. Presionar “Escanear documento”. 3. Esperar a que la imagen sea capturada y enviada al sistema. 4. Verificar que se muestre en el panel de previsualización.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Escaneo	Certificado de Nacimiento	Imagen recibida y mostrada en el formulario correctamente	X	
Postcondiciones				
– Imagen del documento queda disponible para OCR y registro.				
Resultados de la prueba				
Observaciones: Comunicación con el escáner exitosa, la imagen se muestra correctamente.			**Veredicto:** Pasó	

Nota: La tabla presentada corresponde al caso de prueba CP-07, relacionado con el proceso de escaneo de documentos y la recepción de la imagen por parte del sistema. Elaboración propia.

Tabla 40

Caso de prueba – Procesamiento de imagen con redes neuronales

<table>
<tr><th colspan="4">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</th></tr>
<tr><td colspan="2">Caso de prueba: CP-08</td><td colspan="2">Caso de Uso: Gestión de feligreses</td></tr>
<tr><td colspan="4">Descripción del caso de prueba: Validar que el sistema envíe correctamente una imagen a la API con los modelos de detección y extracción, y que la respuesta devuelta sea un objeto JSON que contenga la información extraída etiquetada con su respectiva clase.</td></tr>
<tr><th colspan="4">Precondiciones</th></tr>
<tr><td colspan="4">– La API desarrollada con Flask para el escaneo de documentos debe estar activa.
– La API desarrollada en FastAPI debe estar activa y conectada al sistema web.
– Los modelos de detección y extracción deben estar cargados correctamente.
– Los documentos pueden ser únicamente certificados de nacimiento o cédulas de identidad.</td></tr>
<tr><th colspan="4">Pasos de la prueba</th></tr>
<tr><td colspan="4">1. Abrir el formulario de registro de feligreses.
2. Presionar “Escanear documento”.
3. El sistema envía la imagen a la API mediante una solicitud POST.
4. Esperar la respuesta de la API.
5. Verificar que el JSON recibido incluya:
– Clase detectada
– Texto extraído
– Puntuación de confianza
– Coordenadas del recorte
6. Verificar que los campos del formulario se completen con el texto devuelto.</td></tr>
<tr><th colspan="2">Datos de entrada</th><th rowspan="2">Respuesta esperada del sistema</th><th colspan="2">Coincide</th></tr>
<tr><th>Campo</th><th>Valor</th><th>Si</th><th>No</th></tr>
<tr><td>Imagen</td><td>scan.jpeg</td><td>El sistema envía la imagen a la API.</td><td>X</td><td></td></tr>
<tr><td>Petición a la API</td><td>POST /predict/scan.jpeg</td><td>Validar que el servidor responda con código 200 OK.</td><td>X</td><td></td></tr>
<tr><td>Respuesta JSON</td><td>Generado por la API</td><td>{ "detections": [{ "class": "nombre", "text": "Juan Pérez", "score": 0.92 }, { "class": "fecha_nacimiento", "text": "12/03/2012", "score": 0.89 }] }</td><td>X</td><td></td></tr>
<tr><th colspan="4">Postcondiciones</th></tr>
<tr><td colspan="4">– Los datos del JSON se muestran automáticamente en el formulario de registro.
– La imagen procesada queda almacenada temporalmente.</td></tr>
<tr><th colspan="4">Resultados de la prueba</th></tr>
<tr><td colspan="3">Observaciones: La API devolvió correctamente el JSON con las detecciones de los modelos.</td><td colspan="2">Veredicto: Pasó</td></tr>
</table>

Nota: La tabla presentada corresponde al caso de prueba CP-08, enfocado en el envío de la imagen a la API de procesamiento mediante redes neuronales. Elaboración propia.

3.5.6. Iteración 3

3.5.6.1. Planificación

La Iteración 3 está conformada por las historias de usuario HU-06 (Gestión de cursos sacramentales) y HU-07 (Programación de sesiones de cursos), cuya finalidad es permitir la creación y administración de los cursos sacramentales, así como la programación de sus sesiones de manera organizada. En la Tabla 41 se presentan las tareas definidas para cumplir los criterios de aceptación.

Tabla 41

Plan de tareas - Iteración 3

Código	Tarea	Prioridad	Descripción
IT3-T1	Creación de cursos sacramentales	Alta	Permitir registrar nuevos cursos incluyendo descripción, horarios, costo y sacramento.
IT3-T2	Edición de cursos	Alta	Permitir modificar los detalles de cursos ya existentes.
IT3-T3	Eliminación de cursos	Media	Permitir eliminar cursos de forma segura, garantizando la integridad de la información.
IT3-T4	Consulta de lista de cursos	Media	Implementar la funcionalidad para mostrar la lista completa de cursos con sus detalles.
IT3-T5	Consulta individual de curso	Media	Permitir acceder a la información detallada de un curso específico.
IT3-T6	Creación de sesiones de cursos	Media	Permitir registrar sesiones de cursos con tema, fecha, horario y lugar.
IT3-T7	Edición de sesiones de cursos	Media	Permitir modificar la programación de sesiones existentes.
IT3-T8	Interfaz de gestión de cursos y sesiones	Media	Diseñar las vistas para crear, editar, eliminar y consultar cursos y sus sesiones.
IT3-T9	Pruebas funcionales de cursos y sesiones	Alta	Validar la correcta creación, edición, eliminación y visualización de cursos y sesiones según criterios de aceptación.

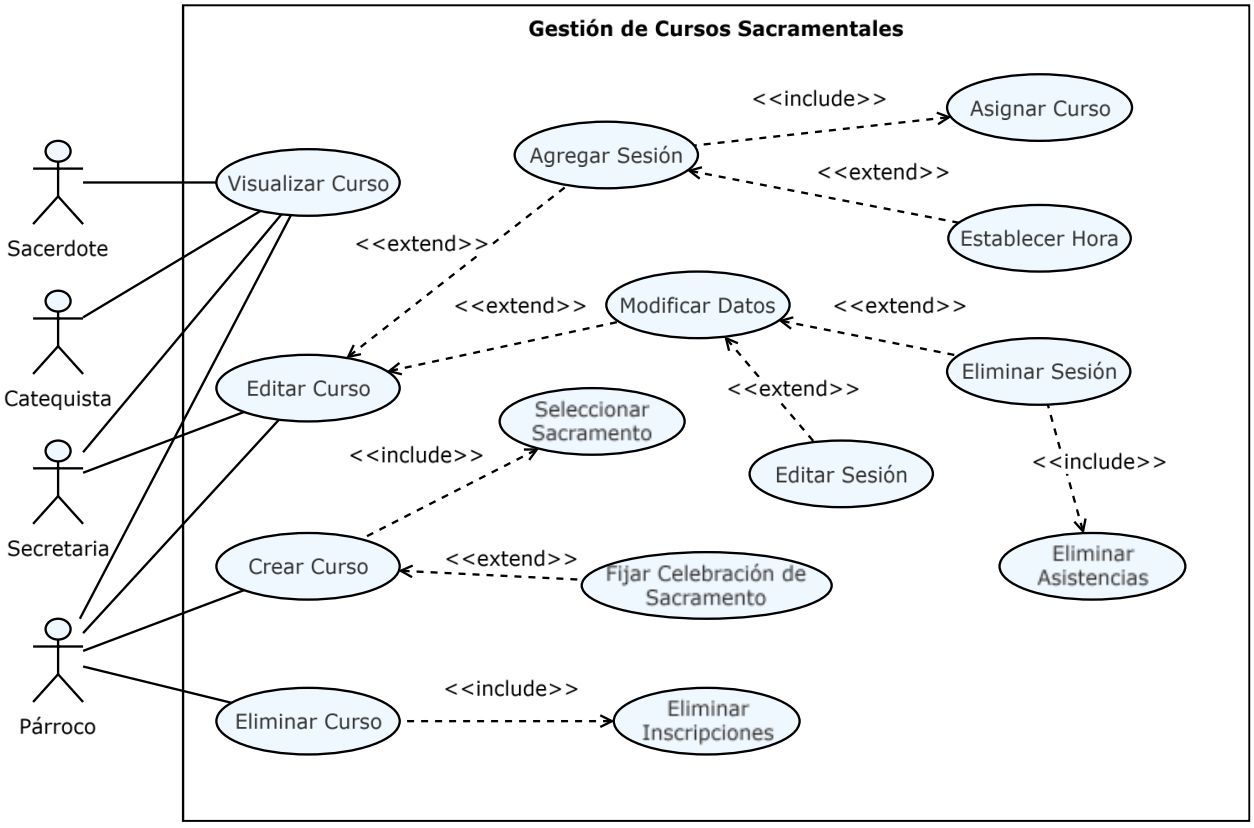
Nota: La tabla muestra las tareas planificadas y ejecutadas durante la Iteración 3 correspondientes a las historias asignadas. Elaboración propia.

3.5.6.2. Diseño

Para la gestión de cursos sacramentales se elaboró un diagrama de casos de uso, como se muestra en la Figura 76, que modela las interacciones de los actores con el sistema, quienes acceden a las funcionalidades según sus roles. Este diagrama permite visualizar de forma clara las operaciones disponibles para cada tipo de usuario, como la creación, edición de cursos. Además, se observa como la creación y administración de sesiones está íntimamente relacionada con la edición de un curso, por lo que implica repercusiones en el caso de su eliminación, pudiendo llegar a afectar en el control de asistencia de los cursos sacramentales.

Figura 76

Diagrama de casos de uso - Gestión de cursos sacramentales

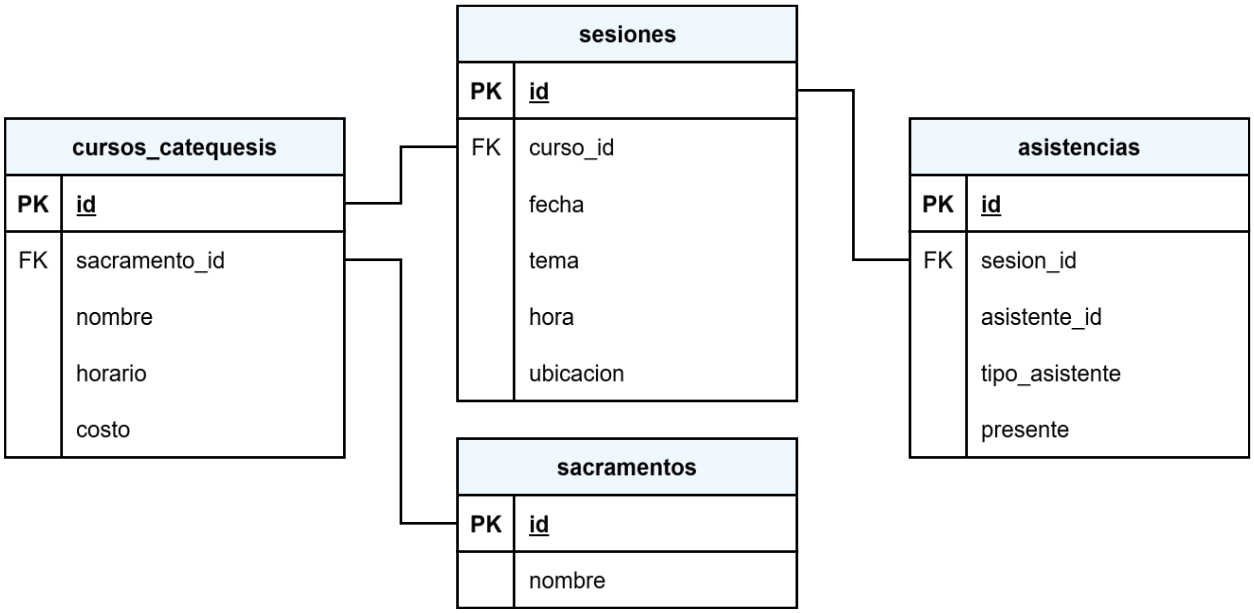


Nota: El diagrama ilustra las funcionalidades esenciales del sistema para la gestión de cursos sacramentales, junto con las sesiones que pueden ser asignadas a los mismos. Destacando cómo cada actor interactúa con las diferentes funciones según su rol. Elaboración propia mediante draw.io.

Para el manejo de los datos, fue necesario representar cómo se almacenaría la información en el sistema, por lo que se elaboró el diagrama relacional de la base de datos, como se muestra en la Figura 77. En él se destaca la relación entre los cursos y las sesiones, donde una sesión puede registrar múltiples asistencias de los feligreses. Además, los cursos pueden asociarse con la celebración comunitaria del respectivo sacramento.

Figura 77

Diagrama relacional- Gestión de cursos sacramentales

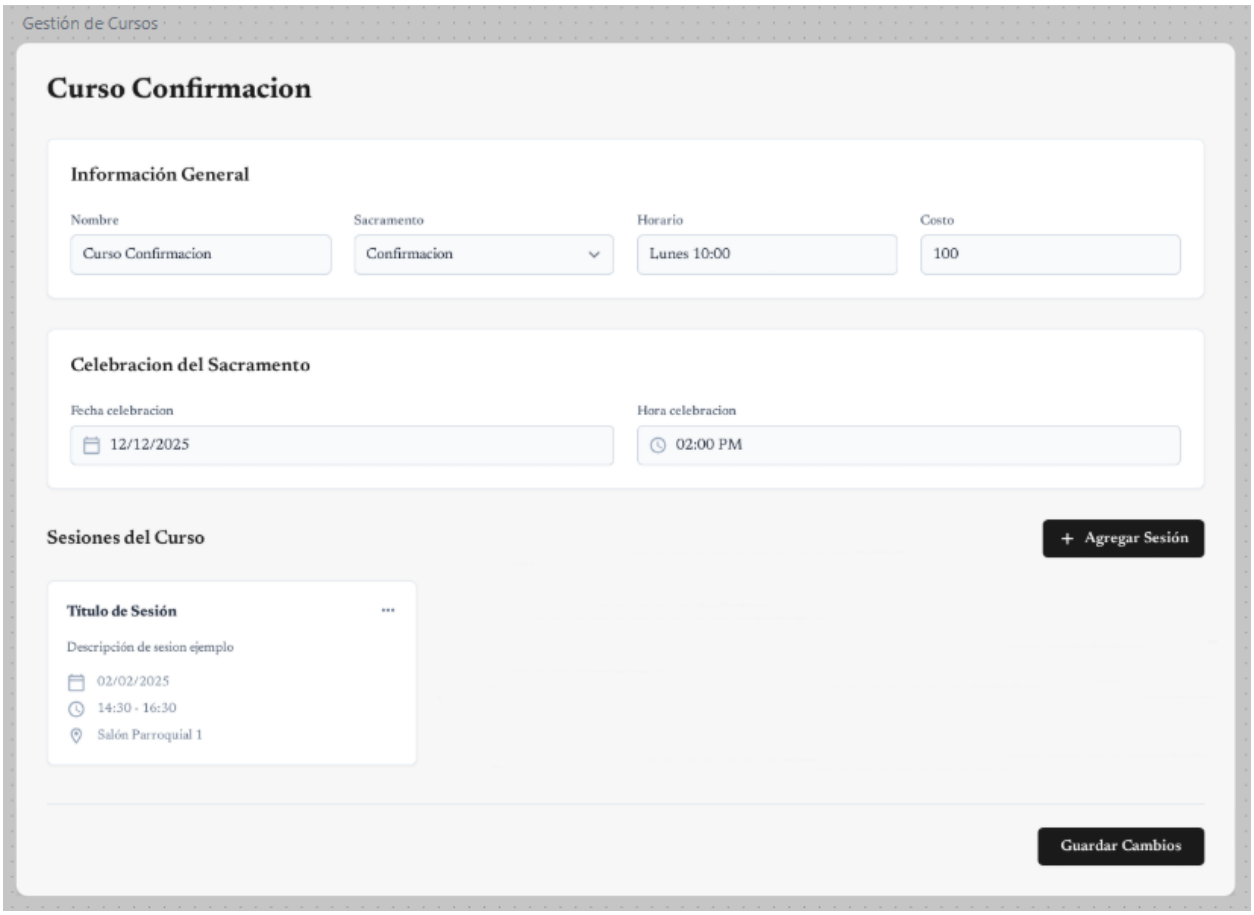


Nota: El diagrama muestra la estructura de la base de datos y las relaciones entre cursos de catequesis, sesiones, sacramentos, asistencias y celebraciones comunitarias. Elaboración propia mediante draw.io.

Finalmente se elaboraron los wireframes correspondientes a la tercera iteración, con el objetivo de definir de manera visual la estructura, distribución de elementos y la navegación entre pantallas. En la Figura 78 se muestra el wireframe para la gestión de cursos sacramentales, en el cual se observa la presentación de la información general, celebración del sacramento y las sesiones propias del curso.

Figura 78

Wireframe – Cursos y sesiones



Nota: La figura muestra la distribución de los elementos en la interfaz para la creación de cursos y sus respectivas sesiones. Elaboración propia mediante Stitch.

3.5.6.3. Desarrollo

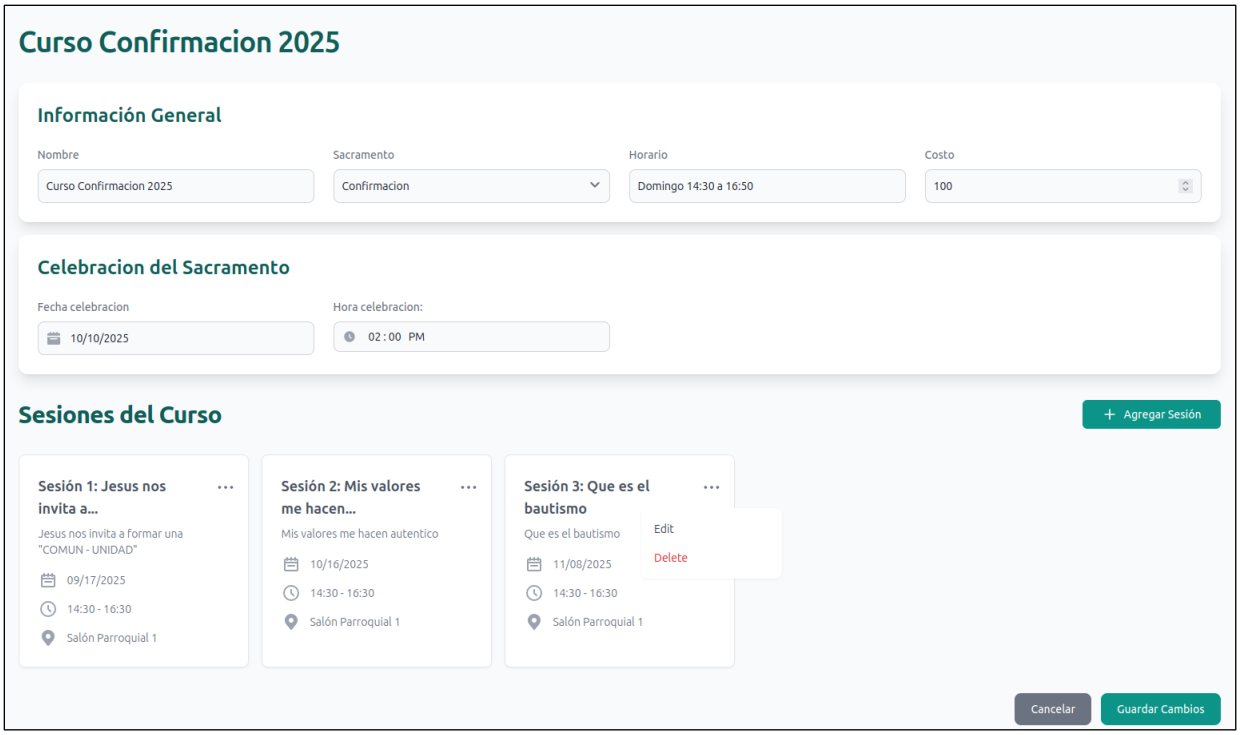
Durante la fase de desarrollo se implementaron las interfaces para la gestión de cursos y sesiones. En su diseño se consideró que no todos los usuarios debían tener permisos de edición, por lo que se desarrollaron dos versiones: una con la capacidad de modificar los datos y otra con acceso únicamente a la visualización de las sesiones y de los inscritos en los cursos sacramentales.

En la vista individual de cada curso, como se muestra en la Figura 79, se presenta la información general del curso, incluyendo el nombre, el sacramento, el horario y el costo. Además, para

aquellos cursos que cuentan con celebraciones comunitarias, se incorpora una sección específica denominada “Celebración del sacramento”. Finalmente, en la parte inferior se listan las sesiones correspondientes al curso, junto con un botón que permite agregar nuevas sesiones, así como opciones para editar o eliminar las existentes.

Figura 79

Pantalla de cursos y sesiones



Nota: La figura muestra la interfaz de usuario con permisos de edición y creación para la gestión de cursos y sesiones. En ella se observan las secciones de información general del curso, celebración del sacramento y listado de sesiones. Elaboración propia.

3.5.6.4. Pruebas

Durante la Iteración 3 se realizaron pruebas funcionales enfocadas en validar la gestión de cursos y sesiones. Se evaluaron la creación, edición y eliminación de cursos, así como la asignación de celebraciones. Además, se verificó la creación, actualización y eliminación de sesiones. Los casos de prueba correspondientes se presentan en las Tablas 42 y 43.

Tabla 42

Caso de prueba – Creación de cursos sacramentales

<table>
<tr><td colspan="5">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</td></tr>
<tr><td colspan="2">Caso de prueba: CP-09</td><td colspan="3">Caso de Uso: Gestión de cursos sacramentales.</td></tr>
<tr><td colspan="5">Descripción del caso de prueba: Validar que el sistema permita registrar un nuevo curso incluyendo nombre, sacramento, costo y horario.</td></tr>
<tr><td colspan="5">Precondiciones</td></tr>
<tr><td colspan="5">– Usuario autenticado como Párroco.
– El sacramento debe estar registrado en el sistema.</td></tr>
<tr><td colspan="5">Pasos de la prueba</td></tr>
<tr><td colspan="5">1. Abrir el formulario “Crear curso”.
2. Completar los campos obligatorios.
3. Presionar el botón “Guardar”.
4. Verificar que el curso aparezca en la lista.</td></tr>
<tr><td colspan="2">Datos de entrada</td><td rowspan="2">Respuesta esperada del sistema</td><td colspan="2">Coincide</td></tr>
<tr><td>Campo</td><td>Valor</td><td>Si</td><td>No</td></tr>
<tr><td>Nombre completo</td><td>Curso Confirmación Tras sus Huellas.</td><td>Validar que el nombre no esté vacío y aceptar solo caracteres permitidos.</td><td>X</td><td></td></tr>
<tr><td>Sacramento</td><td>Confirmación</td><td>Validar que el sacramento exista y asignarlo.</td><td>X</td><td></td></tr>
<tr><td>Horario</td><td>Domingo 14:30 a 16:30.</td><td>Validar que el horario no esté vacío y aceptar solo caracteres permitidos.</td><td>X</td><td></td></tr>
<tr><td>Costo</td><td>100</td><td>Validar que el valor sea numérico.</td><td>X</td><td></td></tr>
<tr><td colspan="5">Postcondiciones</td></tr>
<tr><td colspan="5">– El curso queda registrado en la base de datos.
– Se muestra en la lista general de cursos.</td></tr>
<tr><td colspan="5">Resultados de la prueba</td></tr>
<tr><td colspan="3">Observaciones: El curso fue registrado correctamente.</td><td colspan="2">Veredicto: Pasó</td></tr>
</table>

Nota: La tabla presentada corresponde al caso de prueba CP-09, relacionado con la creación de cursos sacramentales. Elaboración propia.

Tabla 43

Caso de prueba – Creación de sesiones

<table>
<tr><td colspan="2">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</td></tr>
<tr><td>Caso de prueba: CP-10</td><td>Caso de Uso: Gestión de cursos sacramentales.</td></tr>
<tr><td colspan="2">Descripción del caso de prueba: Validar que se puedan registrar nuevas sesiones para un curso.</td></tr>
<tr><td colspan="2">Precondiciones</td></tr>
</table>

– Curso registrado en la base de datos. – Usuario con permisos para crear sesiones.				
Pasos de la prueba				
1. Abrir un curso. 2. Presionar “Agregar sesión”. 3. Completar los campos obligatorios. 4. Presionar el botón “Guardar”.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Tema	Jesús nos invita a formar una COMUN-UNIDAD	Validar que el tema no esté vacío y aceptar solo caracteres permitidos.	X	
Fecha	11-12-2025	Validar formato fecha.	X	
Hora	14:30 -16:30.	Validar formato de rango de horas.	X	
Lugar	Salón parroquial 1	Validar que el lugar no esté vacío y aceptar solo caracteres permitidos.	X	
Postcondiciones				
– La sesión queda registrada y aparece en la interfaz del usuario.				
Resultados de la prueba				
Observaciones: Sesión creada y asignada correctamente.		**Veredicto:** Pasó		

Nota: La tabla presentada corresponde al caso de prueba CP-10, relacionado con la creación de sesiones dentro de los cursos sacramentales. Elaboración propia.

3.5.7. Iteración 4

3.5.7.1. Planificación

La Iteración 4 se centra en las historias de usuario HU-08 (Inscripción a cursos) y HU-09 (Monitoreo de feligreses inscritos). Su objetivo principal es optimizar la gestión de los cursos sacramentales, permitiendo que los feligreses sean inscritos en los cursos sacramentales y que el personal autorizado pueda administrar de forma organizada la información de cada feligrés participante.

Asimismo, la Iteración 4 incluye funcionalidades para supervisar la asistencia, los pagos y la documentación de los feligreses inscritos. Con el fin de mantener un registro completo y actualizado de cada participante, facilitando el seguimiento de los feligreses en los cursos sacramentales. En la Tabla 44 se presentan las tareas definidas para cumplir los criterios de aceptación de las historias abordadas.

Tabla 44

Plan de tareas - Iteración 4

Código	Tarea	Prioridad	Descripción
IT4-T1	Registro de inscripciones a cursos	Alta	Permitir inscribir feligreses a uno o varios cursos de manera simultánea.
IT4-T2	Vista de cursos disponibles	Media	Mostrar al usuario la lista de cursos abiertos para inscripción.
IT4-T3	Consulta de inscritos por curso	Alta	Permitir visualizar los feligreses inscritos en cada curso.
IT4-T4	Cancelación de inscripciones	Media	Permitir anular inscripciones de feligreses según los permisos de rol asignados.
IT4-T5	Registro de pagos de feligreses	Media	Permitir ingresar y actualizar los montos pagados por los feligreses en cada curso.
IT4-T6	Registro de asistencia	Media	Permitir registrar y actualizar la asistencia de los feligreses en cada sesión de curso.
IT4-T7	Registro de documentación de feligreses	Alta	Permitir registra la información más relevante de los documentos proporcionados por los feligreses, como información de padrinos y testigos.
IT4-T8	Interfaz de gestión de inscripciones y monitoreo	Alta	Diseñar vistas para gestionar inscripciones, pagos, asistencia y visualización de documentación.
IT4-T9	Pruebas funcionales de inscripciones y monitoreo	Alta	Validar el correcto funcionamiento del registro de inscripciones, monitoreo de asistencia, pagos y visualización de documentación.

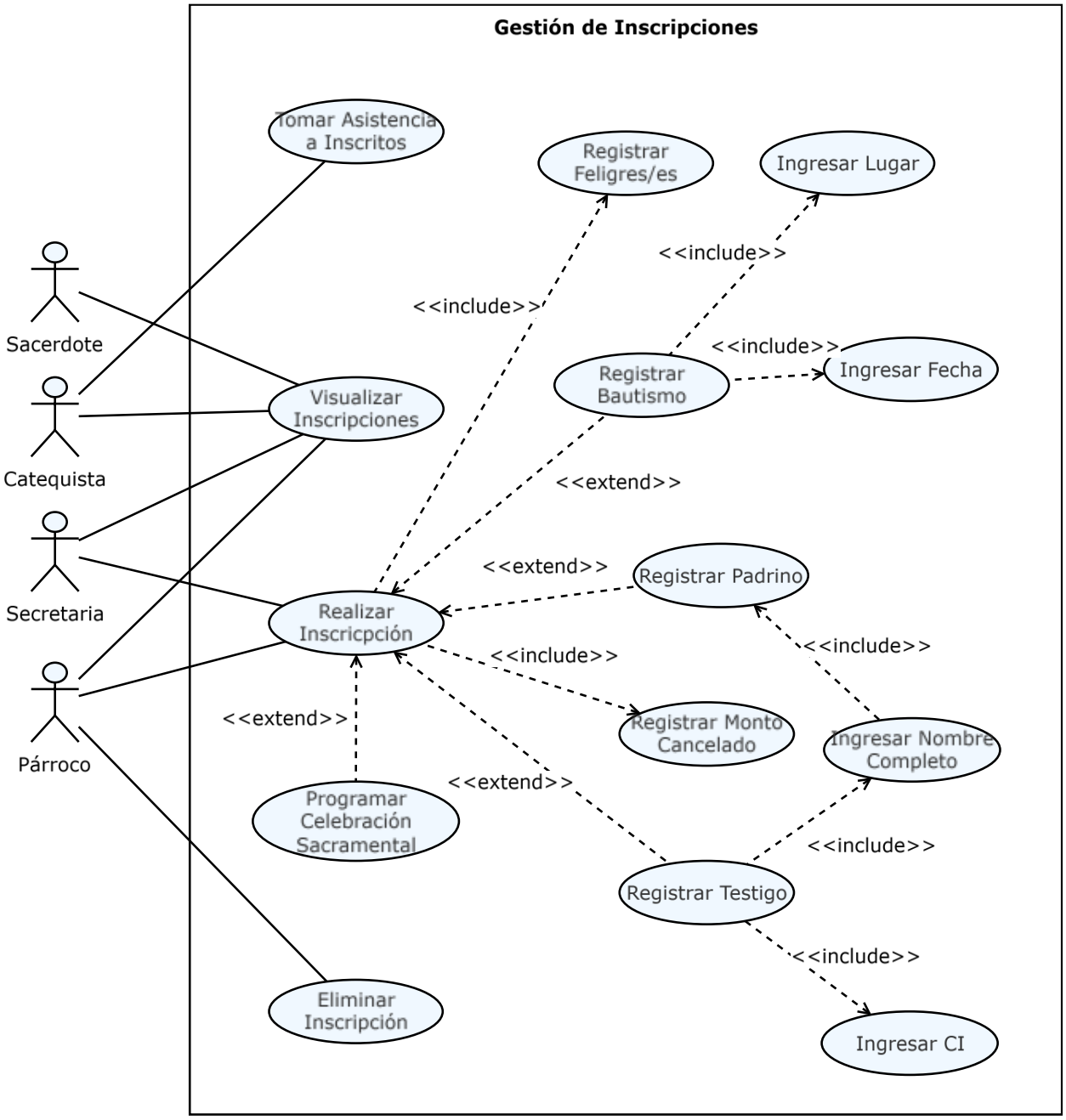
Nota: La tabla muestra las tareas planificadas y ejecutadas durante la Iteración 4 correspondientes a las historias asignadas. Elaboración propia.

3.5.7.2. Diseño

Para la gestión de inscripciones a los cursos sacramentales se elaboró un diagrama de casos de uso, como se muestra en la Figura 80. Este diagrama refleja las interacciones de los distintos actores con el sistema, quienes acceden a las funcionalidades de acuerdo con los permisos establecidos para su rol. Entre las acciones principales se destacan la realización de inscripciones, la visualización de inscripciones registradas, la eliminación de inscripciones y la toma de asistencia de los participantes inscritos en los cursos.

Figura 80

Diagrama de casos de uso – Gestión de inscripciones

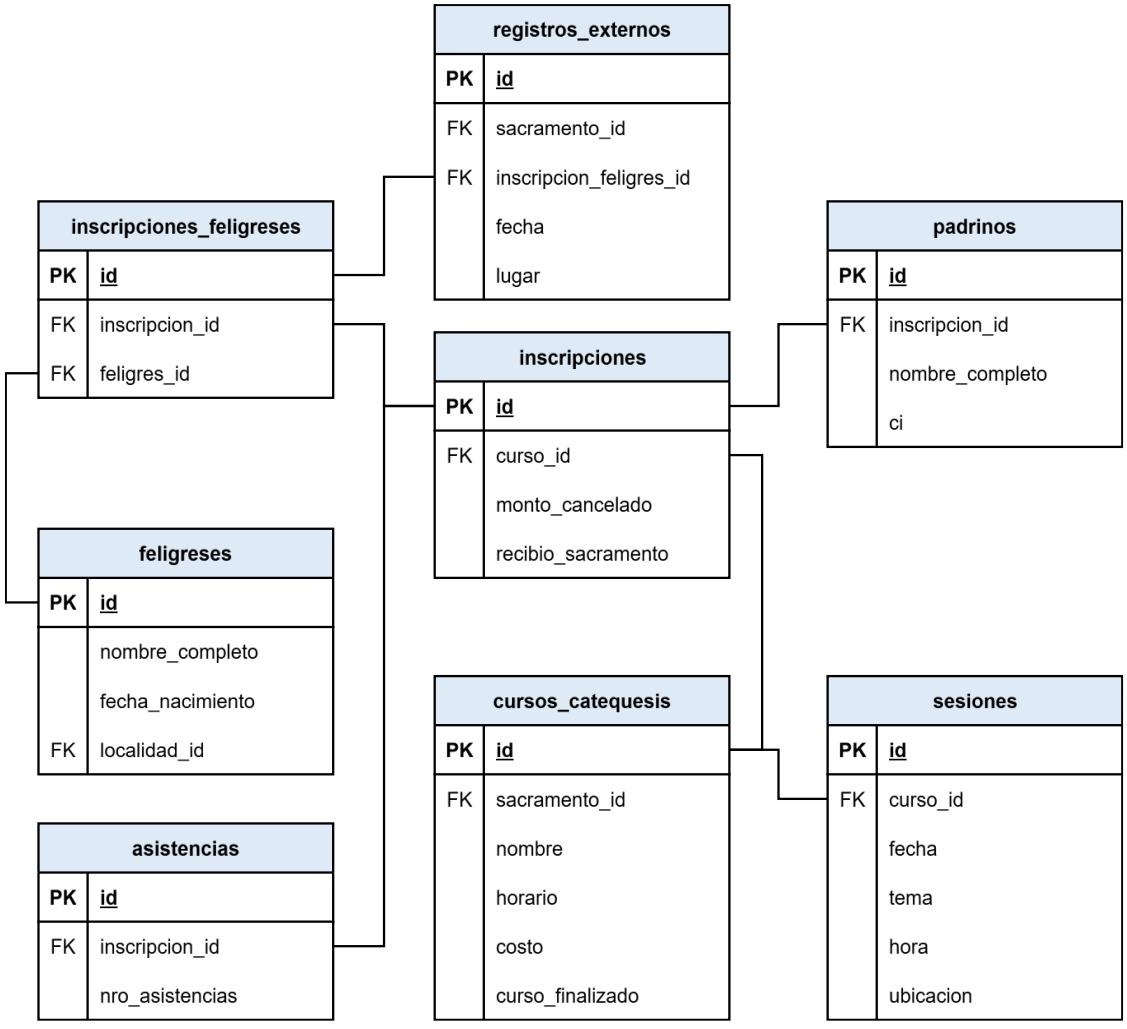


Nota: El diagrama muestra las funcionalidades esenciales del sistema para la gestión de inscripciones en los cursos sacramentales, destacando cómo los actores pueden inscribir, consultar y eliminar inscripciones según el rol que desempeñan. Elaboración propia mediante draw.io.

En cuanto al manejo de los datos, fue necesario representar cómo se almacenaría la información en el sistema, por lo que se elaboró un diagrama relacional de la base de datos, como se muestra en la Figura 81. En este diagrama se destacan las relaciones entre las inscripciones y los feligreses, incluyendo la vinculación de dos feligreses en el caso del sacramento del matrimonio, así como el manejo de la información de los padrinos. También se representan la asistencia a los cursos y los registros externos de sacramentos recibidos en otras parroquias.

Figura 81

Diagrama relacional - Gestión de inscripciones

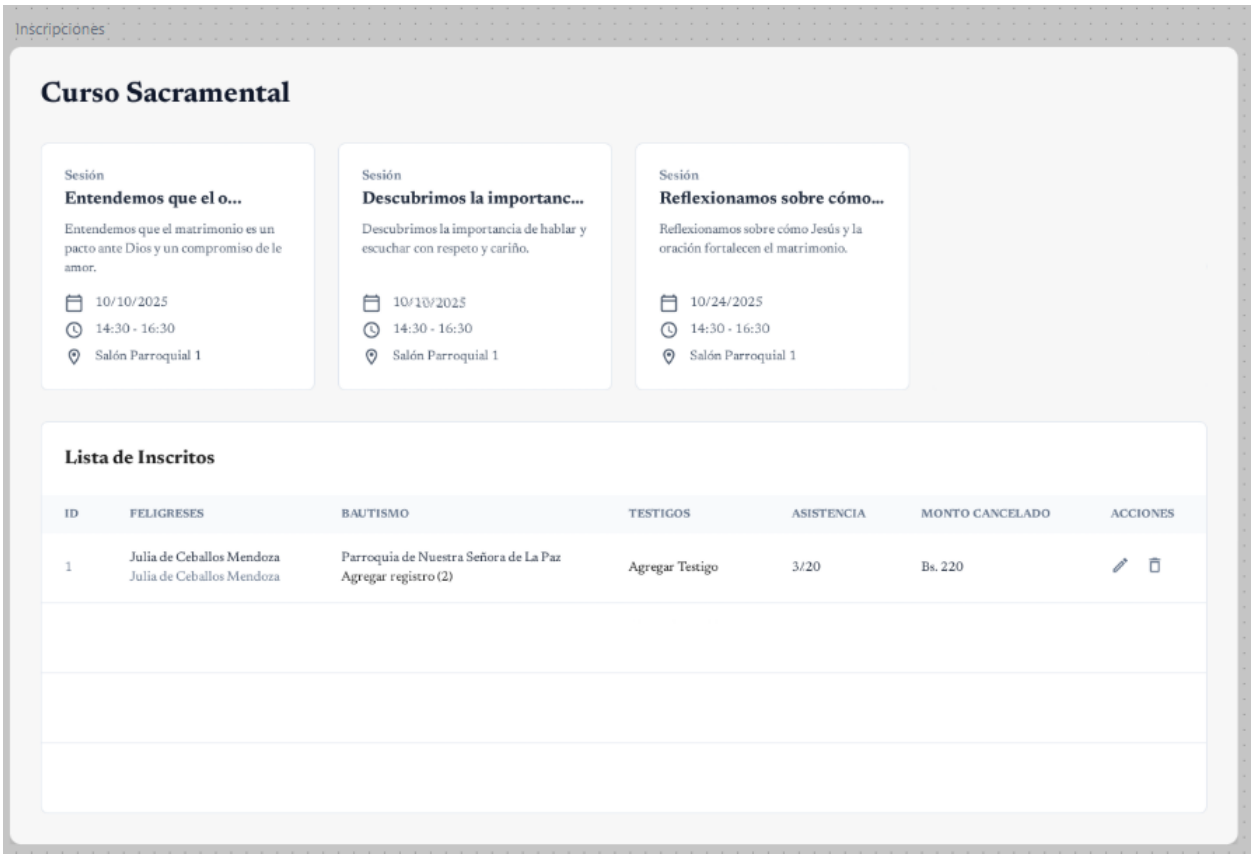


Nota: El diagrama muestra la estructura de la base de datos y las relaciones entre las inscripciones, los feligreses, los padrinos, la asistencia y los registros externos de los sacramentos. Elaboración propia mediante draw.io.

Finalmente, se elaboraron los wireframes correspondientes a la cuarta iteración, con el propósito de definir visualmente la estructura, la distribución de los elementos y la navegación entre las distintas pantallas. En la Figura 82 se presenta el wireframe del módulo de inscripciones a los cursos sacramentales, donde se aprecia la organización de la información más relevante de cada feligrés inscrito, así como los datos necesarios para el proceso de preparación y recepción del sacramento.

Figura 82

Wireframe – Inscripciones de un curso



Nota: La figura muestra la distribución de los elementos en la interfaz para la gestión de las inscripciones en cada curso. Elaboración propia mediante Stitch.

3.5.7.3. Desarrollo

Durante la fase de desarrollo se implementaron las interfaces de inscripción a cursos y de monitoreo de feligreses inscritos. Tal como se definió en los requerimientos del usuario, se incluyó la inhabilitación de botones y acciones según el rol y los permisos de cada usuario.

Para la inscripción se agregó un modal en la vista de los feligreses, que permite registrar su participación en los cursos una vez que se ha ingresado su información personal. La inscripción se realiza en dos etapas, como se muestra en la Figura 83. En la primera etapa, se selecciona el curso en el que se inscribirá el feligrés. En la segunda etapa, se visualiza un resumen de la información de inscripción, incluyendo los nombres completos, el sacramento, los horarios de los cursos y el monto total a cancelar, el cual puede modificarse en caso de que no se realice el pago completo. Para los sacramentos individuales, solo se muestra un feligrés; sin embargo, en el caso del matrimonio, que es un sacramento en pareja, aparece un campo adicional para registrar al segundo feligrés.

Figura 83

Pantalla de formulario inscripciones



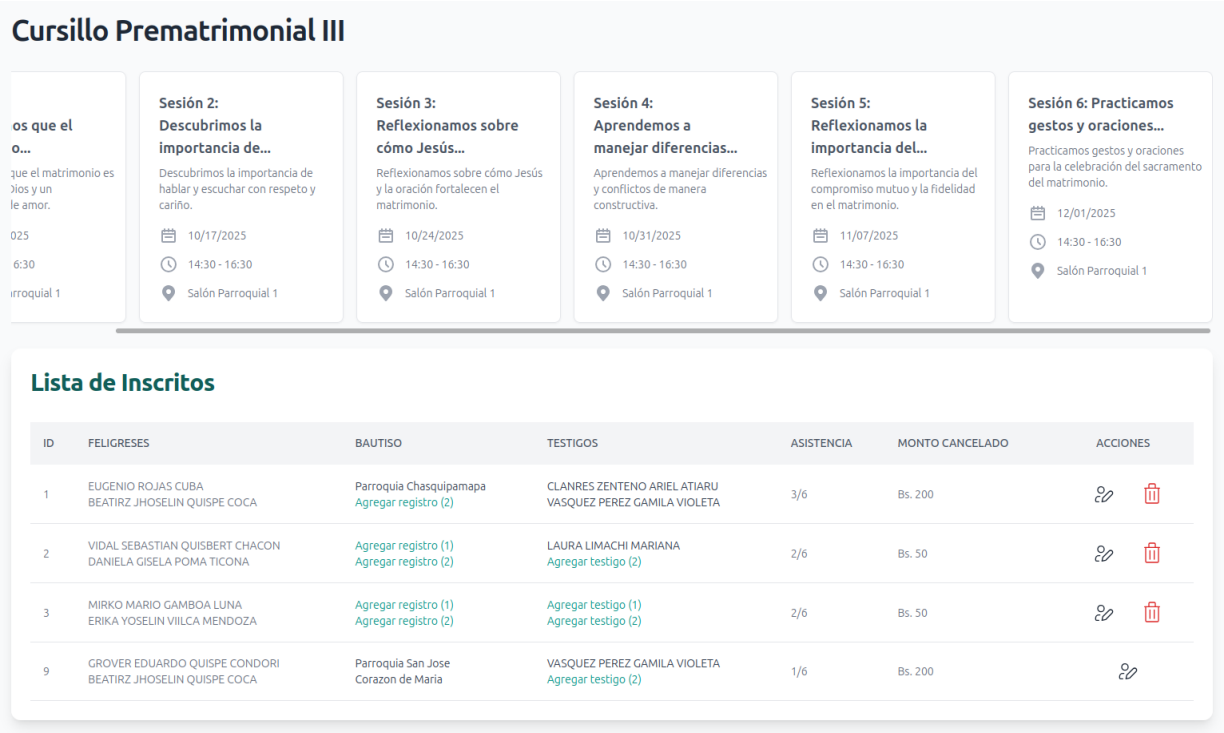
Nota: La figura muestra la interfaz de usuario para la inscripción de feligreses en cursos, con un proceso en dos etapas: selección del curso y visualización del resumen de inscripción. Muestra datos personales, sacramento, horarios, monto a pagar y permite ajustes. Elaboración propia.

Para el monitoreo de las inscripciones y de la información no disponible en la documentación personal del feligrés, como se observa en la Figura 84, en la vista individual de cada curso con sus inscritos se agregó, junto a cada registro de inscripción, la opción de ingresar la información de bautismos realizados en otras parroquias, así como los datos de testigos o padrinos. Esta

información puede registrarse de manera manual o mediante la extracción automática utilizando las redes neuronales implementadas. Además, se muestra la asistencia total a los cursos según el número de sesiones, así como el monto cancelado, acompañado de un botón que permite editar dicho monto.

Figura 84

Pantalla de inscripciones de un curso



Nota: La figura muestra la interfaz de usuario con la vista individual de un curso y sus inscritos, incluyendo opciones para registrar bautismos externos y datos de padrinos, ya sea de manera manual o mediante redes neuronales. También se muestra la asistencia total por sesiones y el monto pagado, con la opción de editarlo. Elaboración propia.

3.5.7.4. Pruebas

Durante la Iteración 4 se realizaron pruebas funcionales para validar la inscripción a cursos y el seguimiento de los feligreses inscritos. Se evaluaron la inscripción de feligreses en los cursos sacramentales, el registro de información de padrinos y testigos, así como la validación de registros de bautismo provenientes de otras parroquias. Los casos de prueba correspondientes se presentan en las Tablas 45 y 46.

Tabla 45

Caso de prueba – Inscripción de feligreses a un curso

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA				
Caso de prueba: CP-11		**Caso de Uso:** Gestión de inscripciones.		
Descripción del caso de prueba: Validar que el sistema permita inscribir a un feligrés en uno o varios cursos de manera correcta.				
Precondiciones				
– Usuario autenticado con permisos de inscripción. – Feligrés registrado en la base de datos. – Cursos disponibles para inscripción.				
Pasos de la prueba				
1. Abrir el modal de inscripción desde la vista de feligreses. 2. Seleccionar el curso. 3. Verificar el resumen de inscripción. 4. Confirmar inscripción. 5. Revisar que el feligrés aparezca en la lista de inscritos del curso.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Feligrés	EUGENIO ROJAS CUBA	Validar que el feligrés exista en la base de datos.	X	
Curso seleccionado	Bautismo 2025	Mostrar datos del curso seleccionado en la ventana resumen.	X	
Monto a pagar	40	Validar que el valor sea numérico y no vacío.	X	
Postcondiciones				
– El feligrés queda inscrito en el curso. – La inscripción aparece en el listado general del curso.				
Resultados de la prueba				
Observaciones: Feligrés inscrito correctamente en el curso.		**Veredicto:** Pasó		

Nota: La tabla presentada corresponde al caso de prueba CP-11, relacionado con la inscripción de feligreses a cursos sacramentales. Elaboración propia.

Tabla 46

Caso de prueba – Registro de documentación

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA	
Caso de prueba: CP-12	**Caso de Uso:** Gestión de inscripciones.
Descripción del caso de prueba: Validar que el sistema permita registrar información de padrinos, testigos y documentos relevantes del feligrés.	

Precondiciones				
– Usuario autenticado con permisos de inscripción. – Feligrés registrado en la base de datos. – Feligrés inscrito en curso de Confirmación.				
Pasos de la prueba				
1. Abrir inscripción del feligrés. 2. Ingresar datos de padrinos. 3. Ingresar datos de bautismo. 4. Guardar cambios. 5. Verificar que la información se muestre en el listado.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Padrino	MAURICIO RAMIREZ DAVALOS	Validar que el padrino no esté vacío y aceptar solo caracteres permitidos.	X	
Madrina	LUCIA MELINA DAVALOS	Validar que la madrina no esté vacía y aceptar solo caracteres permitidos.	X	
Fecha de bautismo	15-02-2002	Validar que el formato sea de fecha.	X	
Lugar de bautismo	Parroquia Inmaculada Concepción	Validar que el lugar no esté vacío y aceptar solo caracteres permitidos.	X	
Postcondiciones				
– Información de padrinos registrada correctamente. – Información de bautismo registrada correctamente.				
Resultados de la prueba				
Observaciones: Registro de documentación completado correctamente.		**Veredicto:** Pasó		

Nota: La tabla presentada corresponde al caso de prueba CP-12, relacionado con el registro de documentación de feligreses. Elaboración propia.

3.5.8. Iteración 5

3.5.8.1. Planificación

La Iteración 5 se centra en las historias de usuario HU-11 (Programación de celebraciones sacramentales) y HU-13 (Monitoreo de celebraciones sacramentales). El objetivo principal de esta iteración es organizar las celebraciones sacramentales, de manera que se pueda monitorear tanto las celebraciones de los cursos sacramentales como las celebraciones diarias de eucaristías y las unciones de los enfermos a domicilio. En la Tabla 47 se presentan las tareas definidas para cumplir los criterios de aceptación de las historias abordadas.

Tabla 47

Plan de tareas - Iteración 5

Código	Tarea	Prioridad	Descripción
IT5-T1	Programación de celebraciones	Alta	Permitir establecer fecha, hora y lugar de cada celebración sacramental.
IT5-T2	Diferenciación de tipo de celebración	Alta	Clasificar celebraciones en comunitarias (Primera Comunión, Confirmación) y particulares (Matrimonio, Bautizo).
IT5-T3	Reprogramación de celebraciones	Media	Permitir cambiar fechas y horarios según disponibilidad de espacios.
IT5-T4	Asociación de celebraciones a cursos	Media	Vincular celebraciones a los cursos correspondientes de feligreses.
IT5-T5	Visualización del calendario de celebraciones	Baja	Mostrar de manera organizada las celebraciones próximas, con fecha, hora y lugar.
IT5-T6	Actualización de información de celebraciones	Media	Permitir modificar los datos de las celebraciones existentes en caso de cambios.
IT5-T7	Interfaz de gestión de celebraciones	Media	Implementar vistas para programar, modificar, asociar y monitorear celebraciones sacramentales.
IT5-T8	Pruebas funcionales de celebraciones	Media	Validar la correcta programación, asociación, visualización y actualización de celebraciones según criterios de aceptación.

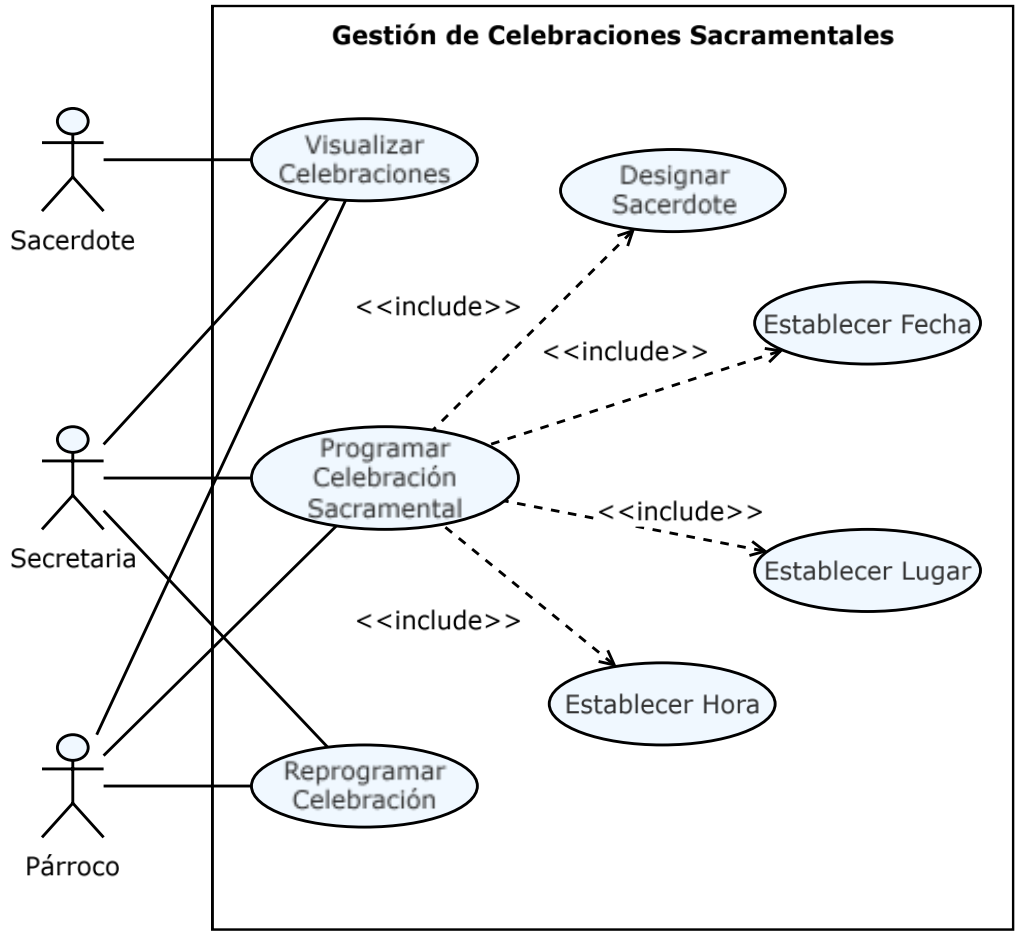
Nota: La tabla muestra las tareas planificadas y ejecutadas durante la Iteración 5 correspondientes a las historias asignadas. Elaboración propia.

3.5.8.2. Diseño

Para la programación y el monitoreo de las celebraciones sacramentales se elaboró un diagrama de casos de uso, como se muestra en la Figura 85. Este diagrama indica que los roles de Párroco y Secretaria son los únicos capaces de agendar y reprogramar celebraciones sacramentales, ya sean comunitarias o particulares. Por otro lado, el rol de Sacerdote únicamente puede visualizar las celebraciones programadas, ya que su actuación se rige por las disposiciones del Párroco.

Figura 85

Diagrama de casos de uso – Gestión de celebraciones sacramentales



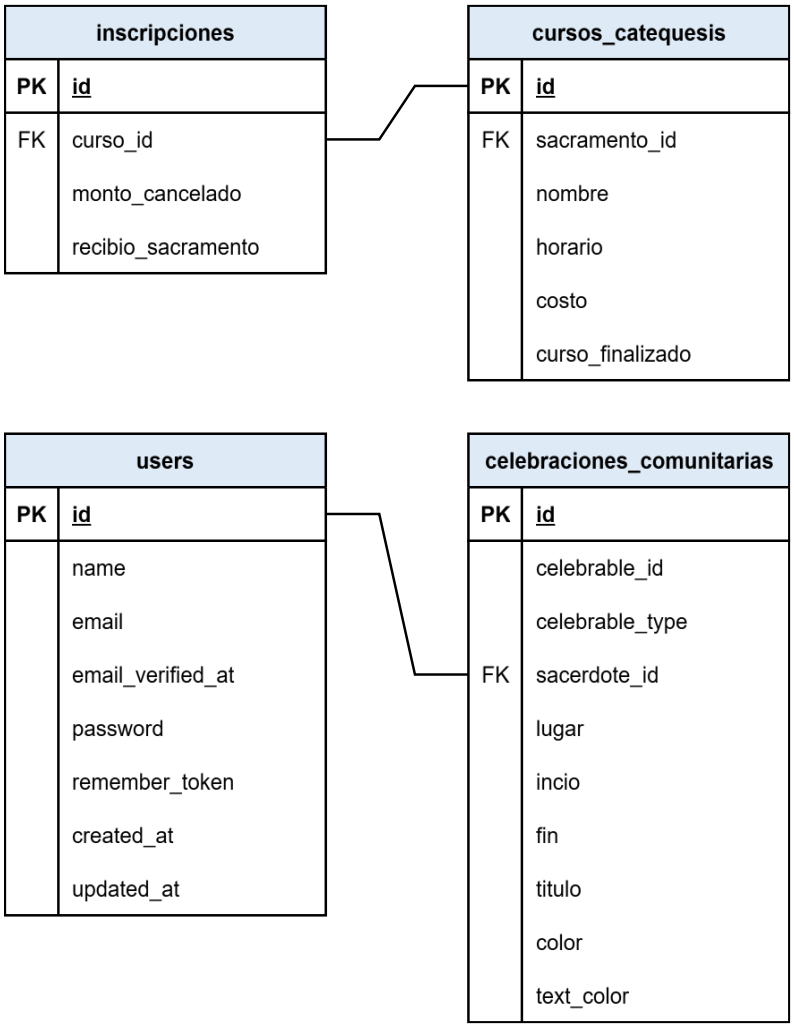
Nota: El diagrama muestra las funcionalidades principales del sistema para la programación de celebraciones sacramentales, destacando cómo cada actor interactúa con las diferentes funciones según su rol. Elaboración propia mediante draw.io.

En cuanto al manejo de los datos, fue necesario representar cómo se almacenaría la información en el sistema, por lo que se elaboró un diagrama relacional de la base de datos, como se muestra en la Figura 86. En este diagrama se destacan las relaciones entre las inscripciones, los cursos y las celebraciones. La relación que vincula las celebraciones con los cursos e inscripciones es polimórfica, lo que permite que existan celebraciones tanto particulares como comunitarias, dependiendo del sacramento. Esto se maneja mediante los campos celebrable_id, que apunta a la clave primaria correspondiente, y celebrable_type, que indica el tipo de celebración. Además,

se representa la relación entre los usuarios y las celebraciones, ya que solo los sacerdotes con ese rol en el sistema podrán realizar celebraciones sacramentales.

Figura 86

Diagrama relacional – Gestión de celebraciones sacramentales



Nota: El diagrama muestra la estructura de la base de datos con relaciones polimórficas entre inscripciones, cursos y celebraciones, y la vinculación de usuarios, donde solo sacerdotes autorizados realizan las celebraciones sacramentales. Elaboración propia mediante draw.io.

3.5.8.3. Desarrollo

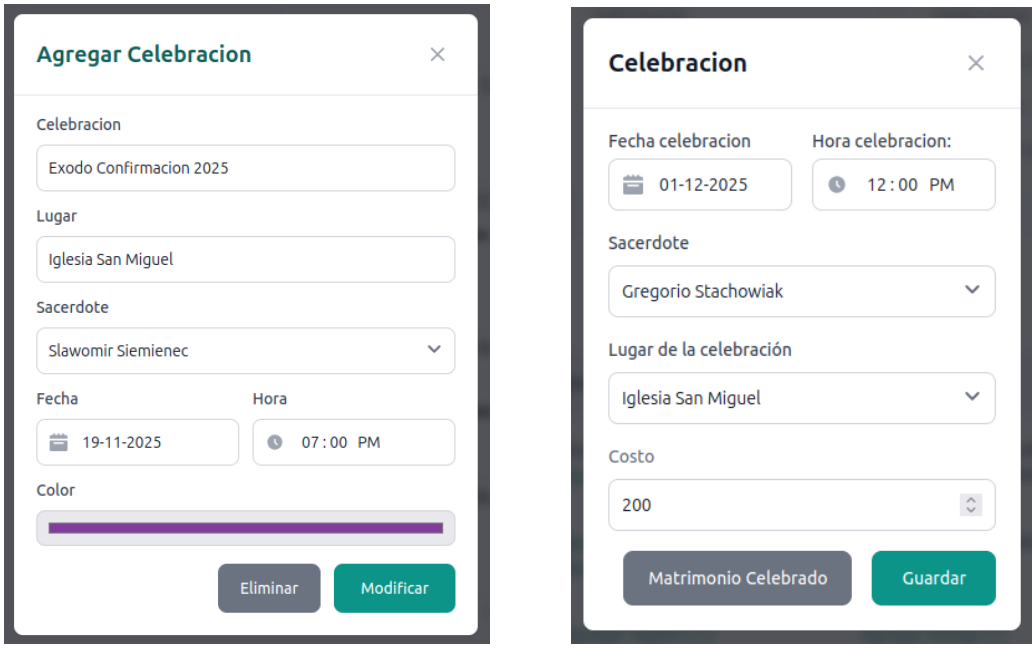
Durante la fase de desarrollo se implementaron las interfaces para la programación y el monitoreo de las celebraciones sacramentales. Tal como se definió en los requerimientos del

usuario, se incluyó la restricción de acceso a ciertas funcionalidades según el rol y los permisos de cada usuario.

Para la programación de celebraciones se implementaron dos modales, como se muestra en la Figura 87. El primero se encuentra en la pantalla de inicio, donde se muestran todas las celebraciones. Este modal permite registrar la celebración indicando el lugar, el sacerdote responsable, la fecha y la hora, además de seleccionar un color para diferenciar visualmente cada celebración al momento de mostrarlas al usuario. El segundo modal se implementó en las vistas de los inscritos en los cursos, para la programación de celebraciones particulares o comunitarias.

Figura 87

Pantallas para la programación de celebraciones



Nota: Se muestran los modales implementados para la programación de celebraciones: uno en la pantalla principal para registrar lugar, sacerdote, fecha, hora y color de la celebración; y otro en la vista de inscritos para programar celebraciones comunitarias o particulares. Elaboración propia.

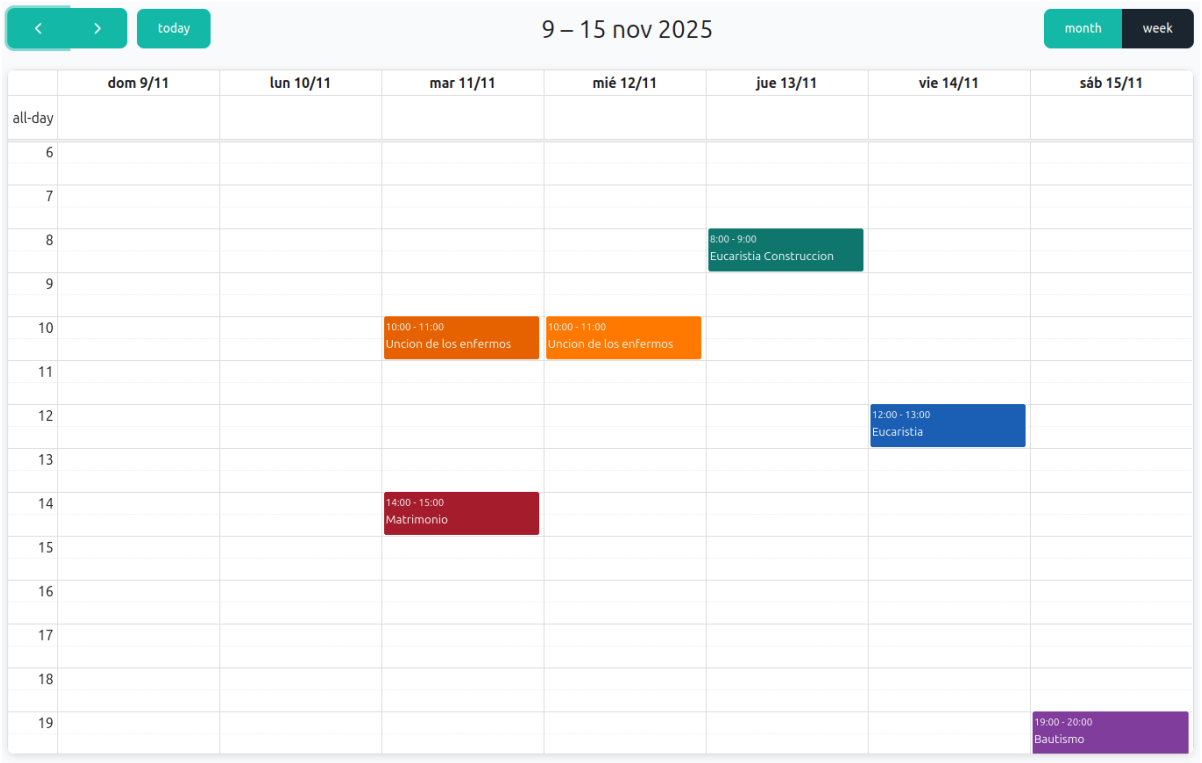
Para el monitoreo de las celebraciones sacramentales se implementó una agenda, como se observa en la Figura 88. En esta agenda se muestran todas las celebraciones programadas, tanto las comunitarias correspondientes a Confirmación o Primera Comunión, como las celebraciones particulares de Bautismo o Matrimonio. Su funcionamiento permite que, al hacer

clic sobre una celebración, se muestre en detalle toda su información. Además, si se hace clic sobre una fecha y hora disponibles, el sistema permite programar una nueva celebración.

La agenda cuenta con dos vistas: una vista mensual que muestra el mes completo y una vista semanal que incluye los días y las horas de las celebraciones.

Figura 88

Pantalla de gestión de celebraciones



Nota: La figura muestra la interfaz de usuario con la agenda para el monitoreo de celebraciones sacramentales, con vista mensual y semanal. Permite visualizar sus detalles al seleccionarlas o programar nuevas celebraciones haciendo clic en una fecha y hora disponibles. Elaboración propia.

3.5.8.4. Pruebas

Durante la Iteración 5 se realizaron pruebas funcionales para validar la programación y reprogramación de celebraciones sacramentales. Los casos de prueba correspondientes se presentan en las Tablas 48 y 49.

Tabla 48

Caso de prueba – Programación de celebraciones sacramentales

<table>
<tr><th colspan="5">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</th></tr>
<tr><td colspan="2">Caso de prueba: CP-13</td><td colspan="3">Caso de Uso: Gestión de celebraciones sacramentales</td></tr>
<tr><td colspan="5">Descripción del caso de prueba: Validar que el sistema permita registrar correctamente una celebración sacramental.</td></tr>
<tr><td colspan="5">Precondiciones</td></tr>
<tr><td colspan="5">– Usuario autenticado con permisos de programación (Párroco o Secretaria).
– Al menos un Sacerdote registrado en el sistema.
– Fecha y hora disponible en la agenda.</td></tr>
<tr><td colspan="5">Pasos de la prueba</td></tr>
<tr><td colspan="5">1. Hacer clic en alguna fecha en la agenda para programar una celebración.
2. Completar campos: sacramento, lugar, fecha, hora, sacerdote y color (opcional).
3. Guardar celebración.
4. Verificar que la celebración aparezca en la agenda.</td></tr>
<tr><td colspan="2">Datos de entrada</td><td rowspan="2">Respuesta esperada del sistema</td><td colspan="2">Coincide</td></tr>
<tr><td>Campo</td><td>Valor</td><td>Si</td><td>No</td></tr>
<tr><td>Sacramento</td><td>Unción de los enfermos</td><td>Validar que el campo no este vacío y que contenga caracteres válidos.</td><td>X</td><td></td></tr>
<tr><td>Fecha</td><td>25-12--2025</td><td>Validar formato de fecha.</td><td>X</td><td></td></tr>
<tr><td>Hora</td><td>14:00</td><td>Validar que la hora este en un formato valido.</td><td>X</td><td></td></tr>
<tr><td>Lugar</td><td>Los Pinos, Calle 1</td><td>Validar que el campo no este vacío y que contenga caracteres válidos.</td><td>X</td><td></td></tr>
<tr><td>Sacerdote</td><td>Rvdo. P. Slawomir Siemienec</td><td>Validar que el sacerdote exista en la base de datos y asociarlo a la celebración.</td><td>X</td><td></td></tr>
<tr><td colspan="5">Postcondiciones</td></tr>
<tr><td colspan="5">– Celebración registrada correctamente y visible en la agenda.</td></tr>
<tr><td colspan="5">Resultados de la prueba</td></tr>
<tr><td colspan="3">Observaciones: Celebración agregada y mostrada correctamente.</td><td colspan="2">Veredicto: Pasó</td></tr>
</table>

Nota: Se presenta el caso de prueba que valida la programación de celebraciones sacramentales. Elaboración propia.

Tabla 49

Caso de prueba – Asociación de celebración a curso

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA	
Caso de prueba: CP-14	**Caso de Uso:** Gestión de celebraciones sacramentales
Descripción del caso de prueba: Validar que una celebración pueda asociarse correctamente a un curso sacramental.	

<table>
<tr><td colspan="4">Precondiciones</td></tr>
<tr><td colspan="4">– Usuario autenticado con permisos en la edición de cursos.
– Curso creado sin celebración asignada.
– Al menos un Sacerdote registrado en el sistema.</td></tr>
<tr><td colspan="4">Pasos de la prueba</td></tr>
<tr><td colspan="4">1. Ir al módulo de cursos y seleccionar uno.
2. Abrir la opción de asociar celebración sacramental.
3. Completar los campos obligatorios: fecha, hora, lugar y sacerdote.
4. Guardar los cambios.
5. Verificar que la celebración aparezca vinculada correctamente al curso.</td></tr>
<tr><td colspan="2">Datos de entrada</td><td rowspan="2">Respuesta esperada del sistema</td><td colspan="2">Coincide</td></tr>
<tr><td>Campo</td><td>Valor</td><td>Si</td><td>No</td></tr>
<tr><td>Fecha</td><td>25-10--2025</td><td>Validar que la fecha esté en un formato valido.</td><td>X</td><td></td></tr>
<tr><td>Hora</td><td>10:00</td><td>Validar que la hora esté en un formato valido.</td><td>X</td><td></td></tr>
<tr><td>Lugar</td><td>Capilla Santa Rosa</td><td>Validar que se haya seleccionado un lugar existente.</td><td>X</td><td></td></tr>
<tr><td>Sacerdote</td><td>Rvdo. P. Slawomir Siemienec</td><td>Validar que el sacerdote exista en la base de datos y asociarlo correctamente a la celebración.</td><td>X</td><td></td></tr>
<tr><td colspan="4">Postcondiciones</td></tr>
<tr><td colspan="4">– Celebración vinculada correctamente al curso.</td></tr>
<tr><td colspan="4">Resultados de la prueba</td></tr>
<tr><td colspan="3">Observaciones: Asociación realizada sin errores.</td><td colspan="2">Veredicto: Pasó</td></tr>
</table>

Nota: La tabla corresponde al caso de prueba CP-13, relacionado con la asociación de celebraciones a cursos sacramentales. Elaboración propia.

3.5.9. Iteración 6

3.5.9.1. Planificación

La Iteración 6 se centra en las historias de usuario HU-10 (Creación de registros sacramentales) y HU-12 (Emisión de certificados sacramentales). Su objetivo principal es generar los registros oficiales una vez que el curso haya finalizado y el feligrés haya recibido el sacramento. A partir de estos registros, se podrá emitir el certificado correspondiente en el formato actualmente utilizado por la parroquia. En la Tabla 50 se presentan las tareas definidas para cumplir los criterios de aceptación de las historias abordadas.

Tabla 50

Plan de tareas - Iteración 6

Código	Tarea	Prioridad	Descripción
IT6-T1	Generación de registros sacramentales oficiales	Alta	Tras finalizar los cursos, transferir la información de los feligreses a los registros oficiales de sacramentos.
IT6-T2	Correlación con libros físicos	Alta	Asegurar que cada registro digital mantenga correspondencia con los libros físicos exigidos por el Arzobispado.
IT6-T3	Inclusión de observaciones	Media	Permitir agregar observaciones en los registros de Bautismo y Matrimonio según sea necesario.
IT6-T4	Búsqueda de registros	Baja	Implementar funcionalidad para buscar registros por distintos criterios: nombre, fecha, sacramento, curso, etc.
IT6-T5	Generación de certificados sacramentales	Alta	Permitir generar certificados oficiales de sacramentos siguiendo los formatos de la parroquia.
IT6-T6	Previsualización de certificados	Media	Mostrar una vista previa del certificado antes de o enviarlo a la impresora.
IT6-T7	Impresión de certificados	Alta	Permitir enviar certificados a la impresora seleccionada, registrando fecha y responsable de la emisión.
IT6-T8	Interfaz de registros y certificados	Media	Diseñar vistas para gestionar registros sacramentales y generar certificados.
IT6-T9	Pruebas funcionales de registros y certificados	Media	Validar que los registros oficiales y la emisión de certificados funcionen correctamente según los criterios de aceptación.

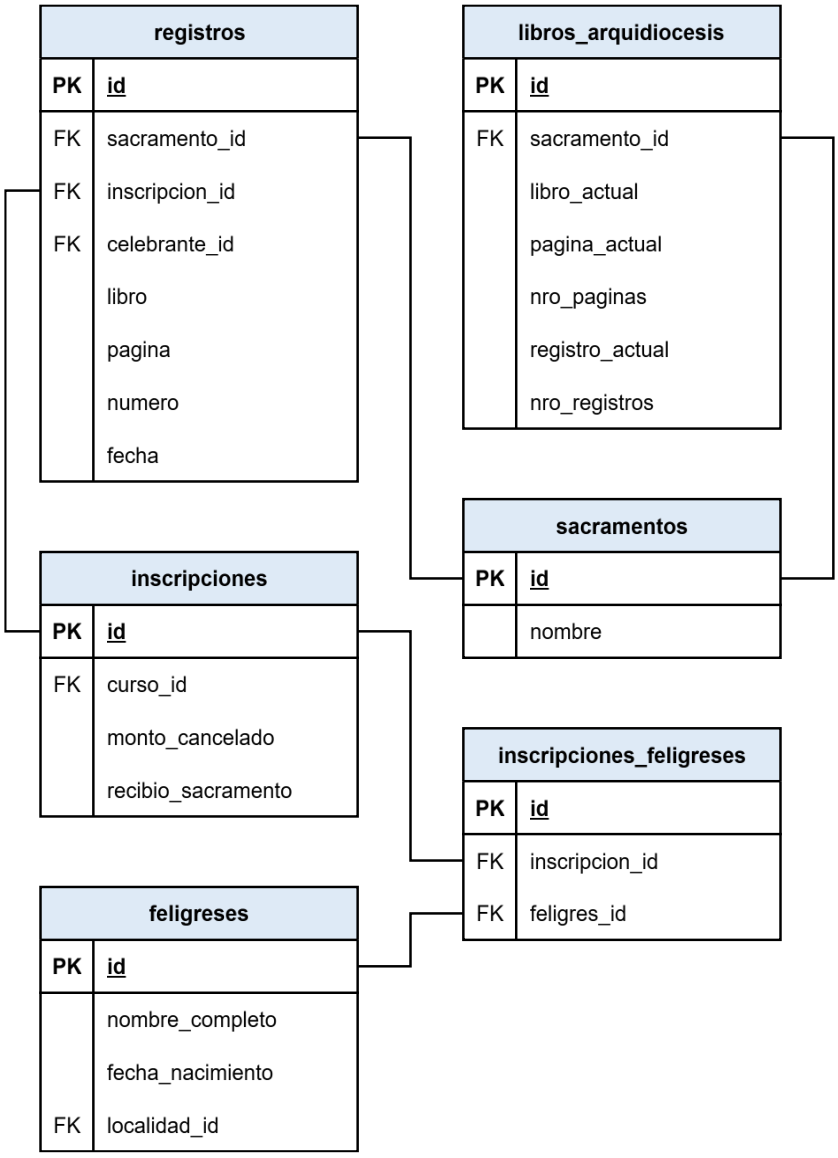
Nota: La tabla muestra las tareas planificadas y ejecutadas durante la Iteración 6 correspondientes a las historias asignadas. Elaboración propia.

3.5.9.2. Diseño

En la etapa de diseño fue necesario representar cómo se almacenaría la información en el sistema mediante un diagrama relacional de la base de datos, como se muestra en la Figura 89. En el diagrama se destacan las relaciones entre los registros, las inscripciones y los libros de la arquidiócesis. Una vez que el feligrés recibe el sacramento, el atributo recibio_sacramento se establece en verdadero, lo que genera automáticamente un registro según la correlación definida en la tabla libros_arquidocesis.

Figura 89

Diagrama relacional – Gestión de registros sacramentales

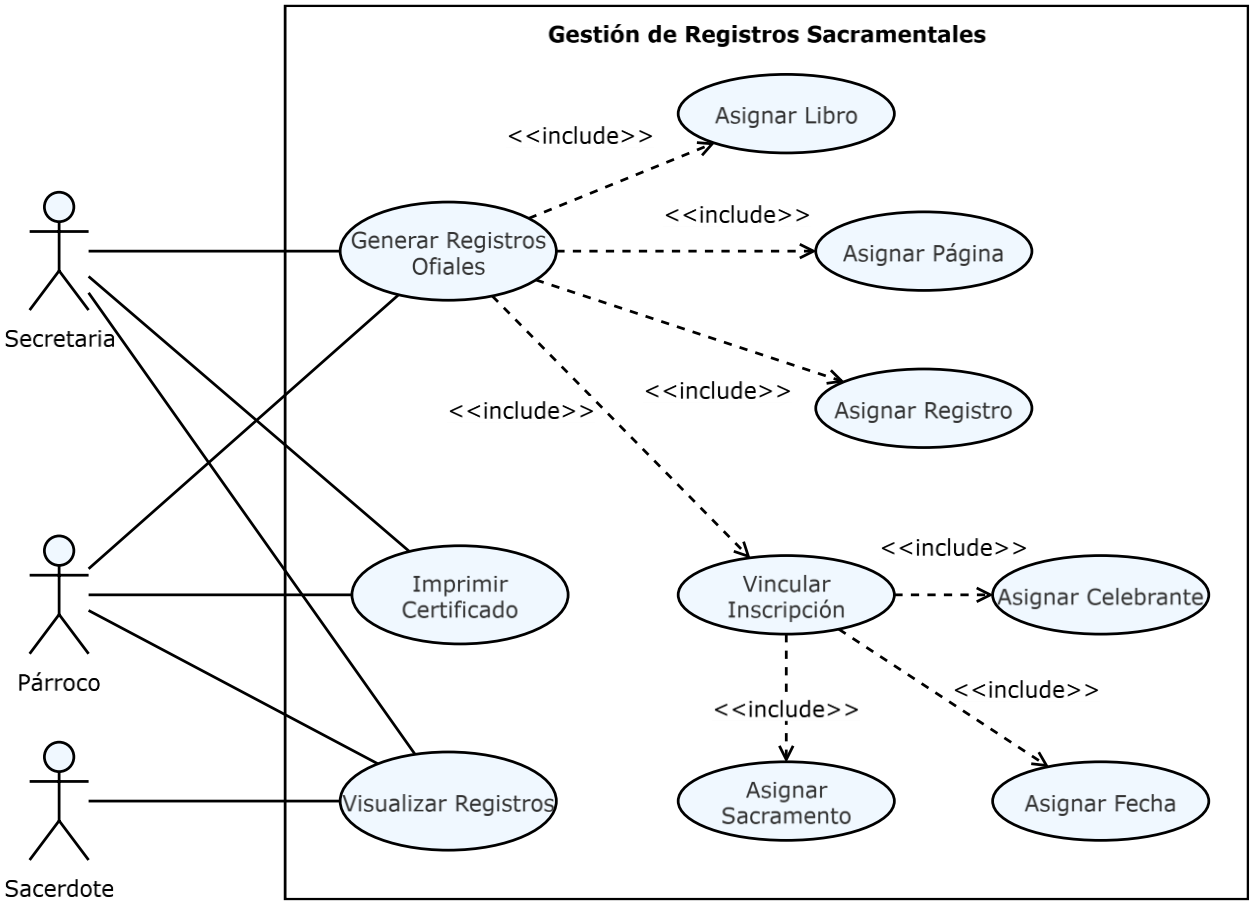


Nota: El diagrama muestra la estructura de la base de datos con las relaciones entre registros, inscripciones y libros de la arquidiócesis. Elaboración propia mediante draw.io.

Para la gestión de registros sacramentales se elaboró un diagrama de casos de uso, como se muestra en la Figura 90. Este diagrama indica que los roles de Párroco y Secretaria son los únicos capaces de generar los registros oficiales de los sacramentos e imprimir certificados que validen la recepción del sacramento. Por otro lado, el rol de Sacerdote únicamente puede visualizar los registros oficiales de los sacramentos.

Figura 90

Diagrama de casos de uso – Gestión de registros sacramentales



Nota: El diagrama muestra las funcionalidades principales del sistema para la gestión de registros sacramentales, destacando cómo cada actor interactúa con las diferentes funciones según su rol. Elaboración propia mediante draw.io.

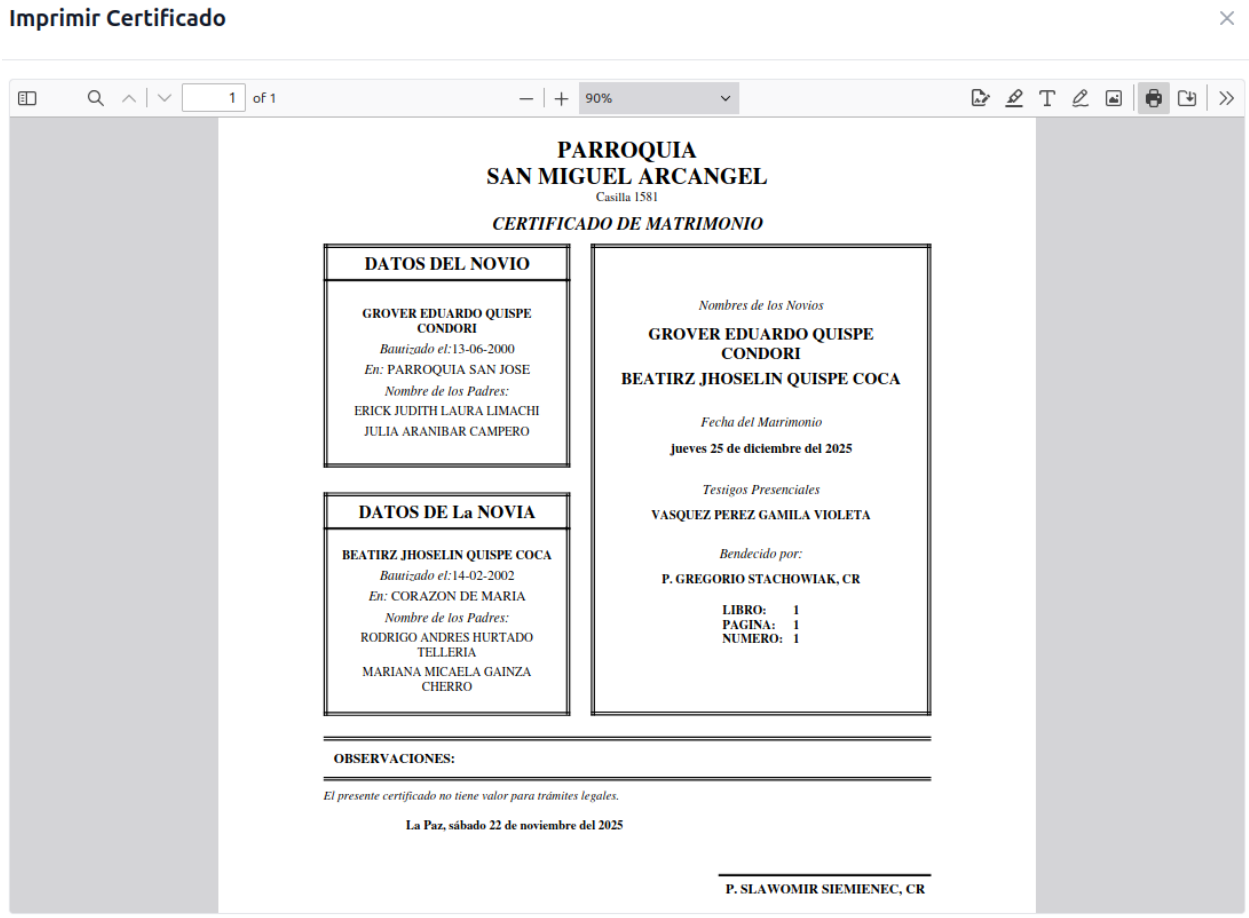
3.5.9.3. Desarrollo

Durante la fase de desarrollo se implementaron las interfaces para los registros sacramentales y la emisión de certificados. Tal como se definió en los requerimientos del usuario, se incluyó la restricción de acceso a ciertas funcionalidades según el rol y los permisos de cada usuario.

Para los registros sacramentales se diseñaron tablas específicas para cada sacramento, con el fin de visualizar los registros correspondientes. En cuanto a la generación de certificados, como se muestra en la Figura 91, junto a cada registro se incluyó un botón que abre un modal con la previsualización del certificado listo para imprimir.

Figura 91

Pantallas para la impresión de certificados



Nota: Se muestra el modal desplegado para la previsualización de los certificados en el formato de la parroquia, listo para impresión. Elaboración propia.

3.5.9.4. Pruebas

Durante la Iteración 6 se realizaron pruebas funcionales enfocadas en validar tanto la generación de los registros oficiales de los sacramentos como la emisión de certificados en formato PDF listos para impresión. Estas pruebas aseguraron que los registros se generen correctamente una vez recibido el sacramento y que los certificados respeten el formato oficial de la parroquia. Los casos de prueba correspondientes se presentan en las Tablas 51 y 52.

Tabla 51

Caso de prueba – Generación de registro sacramental

<table>
<tr><th colspan="4">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</th></tr>
<tr><td colspan="2">Caso de prueba: CP-15</td><td colspan="2">Caso de Uso: Gestión de registros sacramentales</td></tr>
<tr><td colspan="4">Descripción del caso de prueba: Validar que el sistema genere un registro sacramental oficial una vez que el feligrés haya recibido el sacramento.</td></tr>
<tr><th colspan="4">Precondiciones</th></tr>
<tr><td colspan="4">– Inscripción marcada como recibió sacramento = verdadero.
– Usuario con permisos de creación de registros (Párroco o Secretaria).
– Libro correspondiente disponible en el sistema.</td></tr>
<tr><th colspan="4">Pasos de la prueba</th></tr>
<tr><td colspan="4">1. Acceder al módulo del sacramento correspondiente.
2. Entrar al curso del sacramento celebrado.
3. Verificar que la información de la inscripción este completa
4. Presionar el botón “Generar Registro”.
5. Verificar que el registro aparezca en los registros oficiales del sacramento.</td></tr>
<tr><th colspan="2">Datos de entrada</th><th rowspan="2">Respuesta esperada del sistema</th><th colspan="2">Coincide</th></tr>
<tr><th>Campo</th><th>Valor</th><th>Si</th><th>No</th></tr>
<tr><td>Inscripción</td><td>Feligrés, Curso y Celebración</td><td>Validar que exista toda la información necesaria para generar el registro.</td><td>X</td><td></td></tr>
<tr><td>Correlación libro</td><td>El sistema los asigna automáticamente.</td><td>Identificar el libro correspondiente al que se asociará el registro sin intervención del usuario.</td><td>X</td><td></td></tr>
<tr><th colspan="4">Postcondiciones</th></tr>
<tr><td colspan="4">– El registro sacramental queda registrado en la base de datos.</td></tr>
<tr><th colspan="4">Resultados de la prueba</th></tr>
<tr><td colspan="3">Observaciones: El registro se generó correctamente sin requerir ingreso manual de información.</td><td colspan="2">Veredicto: Pasó</td></tr>
</table>

Nota: La tabla corresponde al caso de prueba CP-15, relacionado con la generación del registro sacramental y ejecutado durante la Iteración 6. Elaboración propia.

Tabla 52

Caso de prueba – Impresión de certificado

<table>
<tr><th colspan="2">INFORMACIÓN GLOBAL DEL CASO DE PRUEBA</th></tr>
<tr><td>Caso de prueba: CP-16</td><td>Caso de Uso: Gestión de registros sacramentales</td></tr>
<tr><td colspan="2">Descripción del caso de prueba: Validar que el sistema envíe correctamente el certificado a la impresora seleccionada.</td></tr>
</table>

Precondiciones				
– Registro oficial de un sacramento. – Impresora configurada y encendida. – Usuario con permisos de impresión (Párroco o Secretaria).				
Pasos de la prueba				
1. Acceder al módulo de registros del sacramento correspondiente. 2. Seleccionar un registro y abrir la previsualización del certificado. 3. Presionar el botón “Imprimir”. 4. Seleccionar la impresora deseada. 5. Confirmar la impresión. 6. Verificar que el sistema envíe la orden a la impresora.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Fecha de emisión	Asignada automáticamente por el sistema.	Cargar correctamente la fecha actual en la previsualización y en la impresión del certificado.	X	
Responsable del certificado	El sistema asigna el nombre del Párroco	Mostrar el responsable en la previsualización e incluirlo en la impresión.	X	
Impresora	EPSON L3250	Validar que la impresora esté disponible y aceptar la orden de impresión.	X	
Postcondiciones				
– Certificado enviado a la impresora y listo para ser retirado.				
Resultados de la prueba				
Observaciones: Impresión realizada sin inconvenientes.		**Veredicto:** Pasó		

Nota: La tabla corresponde al caso de prueba CP-16, relacionado con la impresión de certificados sacramentales y ejecutado durante la Iteración 6. Elaboración propia.

3.6. EVALUACIÓN DEL SISTEMA EN LA DOCUMENTACIÓN PERSONAL

La evaluación del sistema desarrollado se centra en analizar cómo la detección y extracción de información de documentación personal mediante redes neuronales impacta directamente en los procesos de registro de información de los feligreses en la Parroquia San Miguel Arcángel. Esta sección aborda los dos aspectos críticos planteados en la formulación del problema: la reducción de demoras en la obtención de datos y la minimización de inconsistencias en los registros. Para cada aspecto, se describe el propósito de su evaluación, la metodología, los resultados obtenidos y la interpretación de estos, siguiendo un enfoque estructurado y metodológico.

3.6.1. Reducción de Demoras en el Ingreso de Datos

3.6.1.1. Propósito

El objetivo de esta evaluación es determinar el impacto del sistema en la eficiencia del registro de información. Antes de la implementación, el proceso manual implicaba múltiples transcripciones: la documentación se ingresaba primero en Excel, y posteriormente, al finalizar el curso, se transcribía a Access tal como se observa en las Figuras 18 y 19.

3.6.1.2. Metodología

La metodología aplicada para evaluar la reducción de demoras se basa en un enfoque cuantitativo, enfocado en comparar los tiempos de los procesos manuales frente al sistema. Para lo cual, se definieron las siguientes métricas:

- Tiempo por registro a un sacramento: Incluye la inscripción al curso, la transcripción al registro oficial y el tiempo de revisión.

- Tiempo promedio por sacramento: Calculado a partir de las mediciones individuales, permite evaluar la eficiencia del sistema en cada tipo de sacramento.

Para el registro manual, se midieron directamente los tiempos de ingreso inicial en Excel y de la transcripción posterior a Access. Debido a las limitaciones de tiempo del caso de estudio, se tomaron únicamente 10 mediciones por cada sacramento, considerando la documentación necesaria para cada feligrés o conjunto de feligreses en casos como el Matrimonio. Posteriormente, se compararon los tiempos manuales frente a los obtenidos con el sistema mediante la siguiente fórmula:

$$Reducción(\%) = \frac{Tiempo\ manual - Tiempo\ sistema}{Tiempo\ maual} * 100$$

3.6.1.3. Resultados

Los tiempos promedio obtenidos para cada sacramento se presentan en las Tablas 53, 54, 55 y 56, mostrando claramente la reducción de tiempos lograda con el sistema en comparación con el proceso manual.

Tabla 53

Tiempos de registro – Sacramento de bautismo

Nro.	Tiempo Manual (min)	Tiempo Sistema (min)	Reducción (%)
1	2.96	2.25	23.99
2	2.83	2.12	25.09
3	2.06	2.11	-2.43
4	3.01	2.21	26.58
5	3.38	2.27	32.84
6	2.85	2.29	19.65
7	3.03	2.31	23.76
8	2.79	2.33	16.49
9	2.93	2.32	20.82
10	2.23	2.21	0.90
Promedio	**2.81**	**2.24**	**20.13**

Nota: La tabla muestra los tiempos medidos en el registro del sacramento de Bautismo mediante la toma de información a partir de un certificado de nacimiento y una cédula de identidad por cada medición. Elaboración propia.

Tabla 54

Tiempos de registro – Sacramento de primera comunión

Nro.	Tiempo Manual (min)	Tiempo Sistema (min)	Reducción (%)
1	2.08	1.80	13.46
2	2.97	1.97	33.67
3	1.89	1.86	1.59
4	1.65	1.90	-15.15
5	2.06	1.82	11.65
6	2.05	1.96	4.39
7	1.85	1.90	-2.70
8	1.65	1.80	-9.09
9	1.83	2.08	-13.66
10	2.15	1.84	14.42
Promedio	**2.02**	**1.89**	**6.19**

Nota: La tabla muestra los tiempos medidos en el registro del sacramento de Primera Comunión mediante la toma de información a partir de un certificado de nacimiento por cada medición. Elaboración propia.

Tabla 55

Tiempos de registro – Sacramento de confirmación

Nro.	Tiempo Manual (min)	Tiempo Sistema (min)	Reducción (%)
1	4.11	3.12	24.09
2	4.82	2.97	38.38
3	3.58	3.10	13.41
4	3.53	2.92	17.28
5	4.27	3.08	27.87
6	4.01	2.79	30.42
7	3.99	3.17	20.55
8	4.29	3.08	28.21
9	3.79	3.11	17.94
10	3.77	2.74	27.32
Promedio	**4.02**	**3.01**	**25.10**

Nota: La tabla muestra los tiempos medidos en el registro del sacramento de Confirmación mediante la toma de información a partir de un certificado de nacimiento y dos cédulas de identidad por cada medición. Elaboración propia.

Tabla 56

Tiempos de registro – Sacramento de matrimonio

Nro.	Tiempo Manual (min)	Tiempo Sistema (min)	Reducción (%)
1	6.94	4.43	36.17
2	5.57	5.02	9.87
3	5.88	4.63	21.26
4	6.50	4.89	24.77
5	6.13	4.77	22.19
6	5.41	4.95	8.50
7	6.25	4.98	20.32
8	5.34	5.15	3.56
9	6.07	4.62	23.89
10	5.47	5.04	7.86
Promedio	**5.96**	**4.85**	**17.84**

Nota: La tabla muestra los tiempos medidos en el registro del sacramento de Matrimonio mediante la toma de información a partir de dos certificados de nacimiento y dos cédulas de identidad por cada medición. Elaboración propia.

Los resultados evidenciaron una reducción notable en los tiempos de registro al comparar el proceso manual con el sistema automatizado. No obstante, es importante considerar que, al tratarse de un sistema recién desarrollado, la secretaria encargada del registro aún no estaba completamente familiarizada con su uso, a diferencia del proceso manual que ha sido aplicado durante décadas. Durante la evaluación, también brindó sugerencias valiosas para mejorar la fluidez de las inscripciones y optimizar la eficiencia en futuros registros. Aun así, la Tabla 57 muestra de forma clara el porcentaje de reducción de demoras alcanzado con la implementación del sistema.

Tabla 57

Reducción porcentual de demoras en el ingreso de datos

Sacramento	Tiempo Manual Promedio (min)	Tiempo Sistema Promedio (min)	Reducción (%)
Bautismo	2.81	2.24	20.13
Primera Comunión	2.02	1.89	6.19
Confirmación	4.02	3.01	25.10
Matrimonio	5.96	4.85	17.84
Promedio General	**-**	**-**	**17.32**

Nota: La tabla presenta el porcentaje de reducción de tiempos en el registro de información para cada sacramento, comparando el proceso manual (ingreso en Excel y transcripción a Access) con el sistema automatizado. Elaboración propia.

3.6.2. Minimización de Inconsistencias en los Registros

3.6.2.1. Propósito

La presente evaluación analiza cómo el sistema desarrollado contribuye a minimizar las inconsistencias en los registros de los feligreses. La duplicación de información y la falta de validaciones en el proceso manual incrementan la probabilidad de errores en el proceso de registro. Con la implementación del sistema, que incluye validaciones y un flujo simplificado, se busca reducir inconsistencias y garantizar registros más confiables.

3.6.2.2. Metodología

La metodología aplicada para evaluar la minimización de inconsistencias se basa en un enfoque cuantitativo y de pruebas funcionales, que permite identificar, clasificar y cuantificar las

inconsistencias presentes en los registros oficiales y verificar cómo el sistema contribuye a reducirlas. Tal como se muestra en la Figura 92, se seleccionó una muestra de 100 registros por cada sacramento, con el fin de analizar la calidad de la información registrada.

Figura 92

Registros oficiales con inconsistencias

FECHABAUTI	APELLIDOS	NOMBRES	INFO	FECHANACIM	LUGAR	PADRE	MADRE	PADRINO	MADRINA	OFICIALIA
		CONSTANZA	NO CANCELARC		LA PAZ, BOLIVIA					
06/01/2000	MAMANI QUILL	ANA VERONICA	NO CANCELARC	15/10/1998	LA PAZ, BOLIVIA	MAMANI LADIS	QUILLI AIDA		ANA ?	
15/01/2000	BAUTISTA CHIN	KAREM IGNACIA		22/11/1998	LA PAZ, BOLIVIA	BAUTISTA RUBI	CHINO FLORA	ELSNER HANS	ELSNER BEATE	1095
15/01/2000	BONIFACIO PO(	ERICK DAVID		14/01/1999	LA PAZ, BOLIVIA	BONIFACIO TOI	POCA MARIA	MOLINA ALVAR	MOLINA CONSU	1208
15/01/2000	URQUIZO HURT	NATHALIA		09/09/1999	LA PAZ, BOLIVIA	URQUIZO MARI	HURTADO Ma.	HURTADO ROG	HURTADO LIND	1216
15/01/2000	PAZ ANDERSON	PAMELA LUCIAI		08/12/1999	LA PAZ, BOLIVIA	PAZ MARCELO	ANDERSON HEA	PAZ RODRIGO	PAZ MARICARM	1054
16/01/2000	AYALA MAMAN	MARIA CRISTIN		13/11/1999	LA PAZ, BOLIVIA	AYALA JAIME	MAMANI MARC	MAMANI PASCU	MAMANI DION	1053
22/01/2000	GAMBOA VARG	ALISON SORAYA		09/05/1996	LA PAZ, BOLIVIA	GAMBOA GERM	VARGAS SALOM		VARGAS REMEI	2312
22/01/2000	GAMBOA VARG	YANELLIS MIRE		11/11/1997	LA PAZ, BOLIVIA	GAMBOA, GERM	VARGAS, SALOM		QUISBERT, ILDA	1211
22/01/2000	RAMOS QUISBI	JOSE SILVERIO		19/05/1979	LA PAZ, BOLIVIA	RAMOS JOSE	QUISBERT GUA		GUACHALLA SU	
29/01/2000	PEREZ CALLE	JONATHAN		17/02/1999	LA PAZ, BOLIVIA	PEREZ, VICTOR	CALLE, FANNY L	MENDIETA, ANT	MENDIETA, IVO	
22/01/2000	VILLARREAL CA	NICOLAS IGNAC		05/04/1999	LA PAZ, BOLIVIA	VILLARREAL, JO	CARDENAS, ISA	CARDENAS, REN	CARDENAS, KAT	1054
22/01/2000	ALARCON CONI	CHRISTIAN EUG		29/07/1996	LA PAZ, BOLIVIA	ALARCON EUGE	CONDORI SOFIA	TICONA JESUS		1054
22/01/2000	MARTINEZ DAV	MATEO GABRIE		14/09/1999	LA PAZ, BOLIVIA	MARTINEZ MIG	DAVILA PATRIC	RIOS JORGE	RIOS REBECA D	1054
29/01/2000	QUISPE CONDC	GROVER EDUAF		23/03/1999	LA PAZ, BOLIVIA	QUISPE LUIS	CONDORI MOD	MMANI EDDY		2284

Nota: La figura muestra registros oficiales del sacramento de Bautismo revisados para la identificación de inconsistencias. Elaboración propia.

Tras la revisión, las inconsistencias detectadas fueron clasificadas en dos categorías principales las cuales se detallan de manera detallada a continuación:

- Inconsistencias de coherencia: Intercambio en los nombres de los padres, fechas incorrectas, caracteres inválidos o desajustes entre los apellidos del feligrés y los de sus padres.

- Inconsistencias por pérdida de información: Campos obligatorios incompletos o ausentes, como la falta de nombres o fechas de nacimiento, que comprometen la integridad del registro del sacramento.

A partir de esta clasificación, se contabilizó el número de registros que presentaban al menos una inconsistencia y se calculó la probabilidad de incidencia por sacramento utilizando la siguiente fórmula:

$$Porcentaje_{Inconsistencia}(\%) = \frac{Nro.\,de\ \ inconsistencias}{Total\ de\ registros\ revisados} * 100$$

Para evaluar la reducción de errores con el sistema, se diseñaron pruebas funcionales bajo condiciones reales del flujo de registro. Estas pruebas incluyeron el ingreso de registros

duplicados para verificar la detección de duplicados, el intento de registrar información incompleta para comprobar las validaciones, y la modificación de datos existentes para asegurar la coherencia sin generar duplicados.

3.6.2.3. Resultados

Los resultados obtenidos de la revisión de los registros oficiales se presentan en la Tabla 58, indicando la probabilidad de que un registro contenga al menos una inconsistencia por sacramento.

Tabla 58

Probabilidad de inconsistencias por registro

Sacramento	Registros con inconsistencias	Total revisados	Probabilidad (%)
Bautismo	17	100	17%
Primera Comunión	10	100	10%
Confirmación	9	100	9%
Matrimonio	9	100	9%
Promedio Total	**45**	**400**	**11.25%**

Nota: La tabla presenta la probabilidad de que un registro oficial contenga al menos una inconsistencia, calculada a partir del número de registros con inconsistencias en una muestra de 100 registros por sacramento. Elaboración propia.

Para validar cómo el sistema contribuye a reducir estas inconsistencias, se diseñaron casos de prueba funcionales basados en situaciones reales del flujo de registro.

Tabla 59

Caso de prueba – Validación de campos obligatorios

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA	
Caso de prueba: CP-17	**Caso de Uso:** Gestión de feligreses
Descripción del caso de prueba: Validar que el sistema no permita registrar un feligrés si los campos obligatorios están incompletos.	
Precondiciones	
– Usuario autenticado como Secretaria o Párroco.	
Pasos de la prueba	
1. Abrir el formulario de registro de feligreses. 2. Dejar uno o varios campos obligatorios vacíos.	

3. Presionar “Guardar”.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Oficialía	(vacío)	Mostrar mensaje: “El campo es obligatorio”	X	
Libro	(vacío)	Mostrar mensaje: “El campo es obligatorio”	X	
Partida	(vacío)	Mostrar mensaje: “El campo es obligatorio”	X	
Mes	(vacío)	Mostrar mensaje: “El campo es obligatorio”	X	
Postcondiciones				
– Registro no guardado en la base de datos.				
Resultados de la prueba				
Observaciones: El sistema bloqueó correctamente el registro de información incompleta.		**Veredicto:** Pasó		

Nota: La tabla corresponde al caso de prueba CP-17, relacionado con la validación de campos obligatorios en el registro de feligreses. Elaboración propia.

Tabla 60

Caso de prueba – Modificación de datos de feligreses

INFORMACIÓN GLOBAL DEL CASO DE PRUEBA				
Caso de prueba: CP-18		**Caso de Uso:** Gestión de feligreses		
Descripción del caso de prueba: Validar que el sistema permita modificar datos existentes sin crear registros duplicados.				
Precondiciones				
– Feligrés registrado previamente en el sistema. – Usuario autenticado con permisos de edición.				
Pasos de la prueba				
1. Abrir el registro de un feligrés existente. 2. Modificar el campo del Padre. 3. Modificar el campo de la Madre. 4. Presionar “Guardar”.				
Datos de entrada		**Respuesta esperada del sistema**	**Coincide**	
Campo	**Valor**		**Si**	**No**
Nombre completo	EUGENIO ROJAS CUBA	Actualizar el campo sin generar un nuevo registro duplicado.	X	

Padre	ANGELINO ROJAS GUTIERREZ	Actualizar el campo sin generar un nuevo registro duplicado.		
Madre	SANTUSA CUBA CONDORI	Actualizar el campo sin generar un nuevo registro duplicado.	X	
Día	11	Actualizar el campo sin generar un nuevo registro duplicado.		
Mes	Enero	Actualizar el campo sin generar un nuevo registro duplicado.		
Año	1988	Actualizar el campo sin generar un nuevo registro duplicado.		
Postcondiciones				
– Modificación reflejada correctamente en la base de datos.				
Resultados de la prueba				
Observaciones: Modificación realizada correctamente			**Veredicto:** Pasó	

Nota: La tabla corresponde al caso de prueba CP-18, relacionado con la modificación de datos de feligreses sin generar registros duplicados. Elaboración propia.

De la revisión de los registros oficiales se observa que la probabilidad promedio de inconsistencias es de 11.25%, siendo Bautismo el sacramento con mayor incidencia. El sistema desarrollado, mediante validaciones de campos y un flujo simplificado, permite bloquear registros incompletos y mantener coherencia al modificar datos en los registros existentes. Esto indica que las medidas implementadas contribuyen de manera efectiva a minimizar errores en los registros oficiales.

3.7. EVALUACIÓN TÉCNICA

Para la evaluación técnica del sistema se toma como referencia la norma ISO 9126, la cual brinda un marco completo para evaluar la calidad del software, definiendo cinco factores clave: usabilidad, funcionalidad, confiabilidad, mantenibilidad y portabilidad. A continuación, se presenta un análisis detallado de cada uno de estos factores en relación con el sistema.

3.7.1. Usabilidad

La usabilidad se centra en el esfuerzo que los usuarios deben realizar para utilizar el sistema de manera efectiva, eficiente y satisfactoria. Según la ISO/IEC 9126, sus características son:

- Entendimiento: La facilidad para comprender cómo usar el sistema.

- Aprendizaje: El esfuerzo necesario para dominar el uso del sistema.
- Operabilidad: La eficiencia en el uso del sistema una vez que se ha aprendido.
- Atracción: Capacidad del sistema para ser atractivo y mantener el interés.
- Conformidad de uso: Cumplimiento con los estándares de usabilidad aplicables.

En la Tabla 61, se muestran los resultados de la encuesta aplicada a los usuarios de la Parroquia San Miguel Arcángel, indicando los niveles de porcentaje alcanzados por el sistema en cuanto a su comprensibilidad y manejo, y presentando finalmente el porcentaje total correspondiente a la usabilidad del sistema.

Tabla 61

Encuesta de Usabilidad del Sistema

Nro.	Pregunta	Respuestas (8)		Porcentaje
		SI	NO	
1	¿El acceso al sistema es complicado?	2	6	75.0%
2	¿Las respuestas del sistema son de su comprensibilidad?	7	1	87.5%
3	¿Tiene alguna dificultad en realizar los procesos del sistema?	3	5	62.5%
4	¿La interfaz del sistema es amigable y entendible a su parecer?	7	1	87.5%
5	¿Son comprensibles y satisfactorios los datos que se muestran en el sistema?	7	1	87.5%
6	¿La aplicación coadyuva de manera eficiente los procesos que se realizaban anteriormente?	6	2	75.0%
7	¿El sistema satisface los requerimientos establecidos?	7	1	87.5%
8	¿El sistema es de fácil uso bajo su criterio?	6	2	75.0%
PROMEDIO TOTAL				**79.69%**

Nota: La tabla muestra los resultados de la encuesta de usabilidad para el Sistema de Extracción de Información Documental. Elaboración propia.

El promedio final de usabilidad del sistema alcanzó el 79.69%, indicando un nivel adecuado de aceptación por parte de los usuarios.

$$USABILIDAD = 79.69\%$$

3.7.2. Funcionalidad

La funcionalidad evalúa el grado en que el sistema cumple con los requisitos y expectativas de los usuarios, para lo cual, en la Tabla 62, se presentan los parámetros utilizados para calcular el Punto Función (PF).

Tabla 62

Parámetros Punto Función

Parámetros	Cantidad	Peso			Factor de Peso
		Simple	Medio	Complejo	
Número de Entradas de Usuario	14	3	4	6	56
Número de Salidas de Usuario	8	4	5	7	40
Número de Peticiones de Usuario	7	3	4	6	21
Número de Archivos Lógicos	22	7	10	15	154
Número de Interfaces Externas	3	5	7	10	30
TOTAL					**301**

Nota: La tabla muestra los parámetros obtenidos mediante el método de Punto Función, el cual se utiliza para medir la funcionalidad proporcionada por el sistema de software, considerando la cantidad y complejidad de entradas, salidas, archivos y consultas. Para el sistema evaluado, se obtuvo un total de 301 puntos. Elaboración propia.

La funcionalidad del sistema se evalúa considerando su complejidad. En la Tabla 63 se presentan los valores de complejidad utilizados para el cálculo del Punto Función.

Tabla 63

Valores de Ajuste de Complejidad

Datos de ajuste	Valor de Ajuste
Sin influencia	0
Menor importancia	1
Moderado	2
Medio	3
Significativo	4
Esencial	5

Nota: La tabla presenta los valores de ajuste de complejidad utilizados para calcular el Punto Función del sistema, según la influencia de cada factor en la funcionalidad del software. Tomado de Ingeniería de software: un enfoque práctico", por Pressman, 2010.

En la Tabla 64 se muestran los valores de ajuste de complejidad aplicados a los distintos factores del sistema, los cuales permiten calcular el ajuste total del Punto Función.

Tabla 64

Ajuste de Complejidad del Sistema

f_i	Factor de ajuste	Valor
1	¿El sistema requiere copias de seguridad y recuperación de datos fiable?	5
2	¿Se requiere comunicación de datos?	5
3	¿Requiere la entrada de datos interactiva y que se realicen operaciones?	4
4	¿Es crítico el rendimiento?	4
5	¿Se ejecutará el rendimiento en un entorno operativo existente y fuertemente utilizado?	4
6	¿Requiere el sistema de datos iterativos?	4
7	¿Se actualizan los archivos de forma interactiva?	3
8	¿Son complejas las entradas y salidas o las peticiones?	4
9	¿Es complejo el procesamiento interno?	5
10	¿Se ha diseñado el código para ser reutilizable?	4
11	¿Están incluidas en el diseño la instalación?	5
12	¿Se diseñó el sistema para que se pueda tener compatibilidad de pantallas?	5
13	¿Se ha diseñado el sistema para ser fácilmente utilizado por el usuario?	5
TOTAL		**57**

Nota: La tabla presenta los factores de ajuste de complejidad, obteniéndose un total de 50 para el Sistema de Extracción de Información Documental. Elaboración Propia.

A partir de los datos obtenidos y considerando un grado de confiabilidad del 65 %, se calculó, mediante la Ecuación 6, el valor del Punto de Función del sistema.

Ecuación 6: Punto de Función

$$PF = C_{total} * \left(G + T * \sum f_i\right)$$
$$PF = 301 * (0.65 + 0.01 * 57) = 367.22 \quad (5)$$

Donde:

C_{total}: suma de los puntos de función.

G: es el grado de confiabilidad.

T: es la taza de error.

f_i: factores de ajuste de complejidad.

Para calcular el punto máximo considerando un 100 % de los factores, se tiene:

$$PF_{maximo} = 301 * (0.65 + 0.01 * 70) = 406.35$$

La relación entre el punto real y el máximo se expresa como porcentaje:

$$PF_{real} = \left(\frac{367.22}{406.35}\right) * 100 = 90.37\%$$

Por lo tanto, la funcionalidad del sistema se estima en 90.37 %, tomando como referencia el Punto de Función máximo posible.

$$FUNCIONALIDAD = 90.37\%$$

3.7.3. Mantenibilidad

La mantenibilidad del sistema hace referencia a la facilidad con la que el software puede ser modificado, corregido o ampliado. Esta métrica permite estimar el esfuerzo requerido para realizar ajustes en el sistema. Para ello, se emplea el Índice de Mantenibilidad (IMS), calculado mediante la Ecuación 7, el cual considera parámetros como la cantidad de módulos de la versión actual, los módulos modificados, añadidos y eliminados.

Ecuación 7: Índice de Mantenibilidad

$$IMS = \frac{M_t - (F_a + F_m + F_e)}{M_t} \quad (2)$$

Donde:

M_t: es el número de módulos en la versión actual.
F_a: es el número de módulos en la versión actual que han sido modificados.
F_m: es el número de módulos en la versión actual que han sido agregados.
F_e: es el número de módulos en la versión anterior que se han eliminado.

Reemplazando con los datos del sistema:

$$IMS = \frac{20 - (5 + 2 + 1)}{20} = 0.6$$

Dentro de los criterios específicos de mantenibilidad del sistema, un valor del 60% podría interpretarse como aceptable, aunque podría requerir ajustes adicionales según las exigencias del proyecto.

$$MANTENIBILIDAD = 60\%$$

3.7.4. Portabilidad

La portabilidad del sistema es un factor clave para garantizar su eficacia y versatilidad. Esta característica determina la facilidad con la que el sistema puede adaptarse a diferentes entornos y plataformas sin perder funcionalidad ni precisión. Los datos relacionados con este aspecto están registrados en la Tabla 65.

Tabla 65

Portabilidad del Sistema

Factor	Valor %
¿El sistema puede ser transferido de un entorno a otro?	90
¿Se puede adaptar con facilidad?	80
¿Es fácil de instalar?	65
¿Es capaz de reemplazar a una aplicación similar?	90
TOTAL	**81.25**

Nota: En la tabla se muestran los principales factores de portabilidad identificados para poder realizar un análisis teniendo en cuenta el funcionamiento del sistema. Elaboración Propia.

El sistema tiene la capacidad de ser ejecutado en diferentes entornos, estableciendo una portabilidad de 81.25%.

El promedio final para la capacidad del sistema de ser ejecutado en diferentes entornos alcanzó el 81.25%, indicando un nivel adecuado en cuanto a portabilidad.

$$PORTABILIDAD = 81.25\%$$

3.7.5. Resultados

Tras el análisis técnico del sistema y considerando los resultados obtenidos, es posible determinar la calidad global del software en función de los parámetros evaluados previamente. La Tabla 66 muestra los puntajes alcanzados en cada una de las características principales analizadas.

Tabla 66

Resultados de la Evaluación Técnica

Características	Resultados
Usabilidad	79.69%
Funcionalidad	90.37%
Mantenibilidad	60.00%
Portabilidad	81.25%
Total	77.83%

Nota: La tabla presenta los resultados obtenidos para cada característica evaluada dentro del análisis técnico del sistema. Elaboración Propia.

Como resultado de la evaluación se obtuvo un puntaje del 77.83%, lo que evidencia que el sistema cumple de manera aceptable con los principales factores de calidad. No obstante, este nivel de desempeño también señala que existen áreas susceptibles de mejora para elevar la calidad general del sistema.

3.8. EVALUACIÓN ECONÓMICA

Para la evaluación económica del presente proyecto se consideraron los costos de desarrollo y los costos de operación del software, obteniéndose así el costo total del proyecto.

3.8.1. Costos del Desarrollo de Software

Para estimar los costos del desarrollo de software se utilizará el modelo COCOMO Intermedio, que ofrece una metodología para calcular de manera aproximada el esfuerzo, el tiempo y los recursos requeridos en cada fase del desarrollo. De esta forma, es posible obtener una estimación confiable de los costos asociados a la creación del producto.

3.8.1.1. Líneas de Código

El primer paso consiste en calcular el esfuerzo mediante la determinación de la variable KDLC (Kilos de Líneas de Código). Para este cálculo se considera que PF = 301, valor previamente conocido, y que la cantidad de líneas por punto de función depende del lenguaje de programación utilizado, cuyos valores se presentan en la Tabla 67 mediante el factor de líneas de código por punto de función (LDC/PF).

Tabla 67

Líneas de Código por Punto Función

Lenguaje	Factor LDC/PF
Python	25
PhP	25
Javascript	35

Nota: La tabla muestra el factor de líneas de código por punto de función (LDC/PF) correspondiente a cada lenguaje de programación considerado para el desarrollo del sistema. Elaboración Propia.

Siguiendo la Ecuación 8, y una vez establecidos los factores de los lenguajes empleados, se procede a calcular los KDLC para Python, PHP y JavaScript.

Ecuación 8: Ecuación KLDC por PF

$$KLDC = \frac{PF * Factor\left(\frac{LDC}{PF}\right)}{1000} \quad (2)$$

$$KLDC_{python} = \frac{301 * 25}{1000} = 7.52$$

$$KLDC_{php} = \frac{301 * 25}{1000} = 7.52$$

$$KLDC_{javascript} = \frac{301 * 35}{1000} = 10.53$$

Al haberse utilizado más de un lenguaje, es necesario obtener el total de KDLC, lo cual se logra mediante la suma de los valores calculados para cada lenguaje.

$$KLDC = KLDC_{python} + KLDC_{php} + KLDC_{javascript}$$

$$KLDC = 7.52 + 7.52 + 10.53$$

$$KLDC = 25.57$$

3.8.1.2. Tipo de Proyecto

En el modelo COCOMO, tanto en sus versiones básica, intermedia y avanzada, es necesario seleccionar el tipo de proyecto según el valor de KDLC. En la Tabla 68 se presentan los diferentes tipos de proyecto y sus rangos correspondientes.

Tabla 68

Tipos de Proyectos en el Modelo COCOMO

Tipo	Tamaño del Proyecto
Orgánico	Menor a 50 KLDC
Semi-acoplado	Entre 50-300 KLDC
Empotrado	Mayor a 300 KLDC

Nota: La tabla presenta los distintos tipos de proyecto según el modelo COCOMO, clasificados de acuerdo con el tamaño del proyecto medido en KLDC. Elaboración Propia.

Dado que el valor de KDLC obtenido para el proyecto no supera las 50, el proyecto se clasifica como de tipo orgánico. En la Tabla 69 se presentan los coeficientes que se utilizarán para el cálculo del esfuerzo y del tiempo de desarrollo.

Tabla 69

Coeficientes COCOMO

Tipo de Proyecto	a	b	c	d
Orgánico	3.2	1.05	2.5	0.38

Nota: La tabla muestra los coeficientes del modelo COCOMO intermedio empleados para la estimación de costos en proyectos de software de tipo orgánico. Los valores de a, b, c y d corresponden a los coeficientes estándar definidos por el modelo, utilizados para calcular el esfuerzo y el tiempo de desarrollo. Elaboración Propia.

3.8.1.3. Factor de Ajuste de Esfuerzo

Para calcular el valor del factor de ajuste de esfuerzo (FAE), se consideran los 15 conductores de costo mostrados en la Tabla 70. El FAE se obtiene mediante la multiplicación de los valores asignados a cada uno de estos conductores.

Tabla 70

Conductores de Costo en el Modelo COCOMO

CONDUCTORES DE COSTE	VALORACIÓN					
	Muy bajo	**Bajo**	**Nominal**	**Alto**	**Muy alto**	**Extra alto**
Fiabilidad requerida del software	0.75	0.88	1.00	**1.15**	1.40	-
Tamaño de la base de datos	-	0.94	1.00	**1.08**	1.16	-
Complejidad del producto	1.70	0.85	1.00	1.15	**1.30**	1.65
Restricciones del tiempo de ejecución	-	-	1.00	**1.11**	1.30	1.66
Restricciones del almacenamiento principal	-	-	**1.00**	1.06	1.21	1.56
Volatilidad de la máquina virtual	-	0.87	**1.00**	1.15	1.30	-
Tiempo de respuesta del ordenador	-	0.87	1.00	**1.07**	1.15	-
Capacidad del analista	1.46	1.19	**1.00**	0.86	0.71	-
Experiencia en la aplicación	1.29	1.13	1.00	**0.91**	0.82	-
Capacidad de los programadores	1.42	1.17	1.00	**0.86**	0.70	-
Experiencia en S.O. utilizado	1.21	1.10	**1.00**	0.90	-	-
Experiencia en el lenguaje de programación	1.14	1.07	**1.00**	0.95	-	-
Prácticas de programación modernas	1.24	1.10	1.00	**0.91**	0.82	-
Utilización de herramientas software	1.24	1.10	1.00	**0.91**	0.83	-
Limitaciones de planificación del proyecto	1.23	1.08	**1.00**	1.04	1.10	-

Nota: La tabla presenta los coeficientes de los conductores de costo del modelo COCOMO intermedio, correspondientes a distintos factores del desarrollo de software, clasificados según sus niveles de valoración. Elaboración Propia.

Con los valores de los conductores de costo se obtiene el FAE mediante la multiplicación ellos.

$$FAE = 1.15 * 1.08 * 1.30 * 1.11 * 1 * 1 * 1.07 * 1 * 0.91 * 0.86 * 1 * 1 * 0.91 * 0.91 * 1$$

$$FAE = 1.24278$$

3.8.1.4. Esfuerzo

Para determinar el esfuerzo requerido en el desarrollo del proyecto, se emplea la Ecuación 9 del modelo COCOMO intermedio. Esta ecuación permite calcular el esfuerzo en personas por mes a partir de los KLDC y del factor de ajuste de esfuerzo (FAE), considerando los coeficientes específicos del tipo de proyecto.

Ecuación 9: Ecuación de Esfuerzo

$$E = a * KLDC^{b} * FAE \quad (2)$$

$$E = 3.2 * 25.57^{1.05} * 1.24278$$

$$E = 119.58\ personas/mes$$

3.8.1.5. Tiempo

Para estimar la duración del desarrollo del proyecto, se utiliza la Ecuación 10 del modelo COCOMO intermedio. Esta ecuación permite calcular el tiempo de desarrollo en meses, considerando los coeficientes específicos del tipo de proyecto y el esfuerzo previamente calculado.

Ecuación 10: Ecuación de Tiempo

$$T = c * E^{d} \quad (2)$$

$$T = 2.5 * 119.58^{\,0.38}$$

$$T = 15.39\ meses$$

3.8.1.6. Costo de Software

Para el desarrollo del proyecto se consideró la inversión correspondiente a un solo desarrollador durante un período aproximado de 15 meses. En la Ecuación 11 se realiza el cálculo del costo total del desarrollo, suponiendo un sueldo mensual de BS. 2500.

Ecuación 11: Ecuación de Costo de Software

$$C = T * D * S$$

Donde:

C: es el costo de desarrollo del software.

T: es el tiempo requerido para el desarrollo del software.

D: es el número de desarrolladores.

S: es el sueldo de los desarrolladores.

Reemplazando cada uno de los valores en la ecuación:

$$C = 15.39 * 1 * 2500$$

$$C = 38475$$

Como resultado del cálculo, se determina que el costo estimado para el desarrollo del software es de Bs. 38475, correspondiente a un período de trabajo de aproximadamente 15 meses con un desarrollador.

3.8.2. Costos de Operación del Software

Para mantener en funcionamiento el sistema una vez desarrollado, es necesario considerar los costos de operación del software, que incluyen los recursos de infraestructura y servicios asociados, los cuales se detallan en la Tabla 71. Estos costos permiten asegurar la disponibilidad, seguridad y correcto desempeño del sistema durante su uso.

Tabla 71

Costos de Operación del Software

Concepto	**Descripción**	**Costo (Bs.)**
Servicio VPS (Hostinger KVM 2)	VPS con 2 vCPU, 8 GB RAM, 100 GB SSD NVMe, 8 TB de transferencia.	568.0
Dominio	Nombre de dominio utilizado para acceder al sistema smaparroquia.cloud.	-
Certificado SSL	Certificado Let’s Encrypt instalado.	-
TOTAL		**568.0**

Nota: La tabla muestra los costos de operación del software: incluye VPS con 2 vCPU, 8 GB RAM, 100 GB SSD NVMe y 8 TB de transferencia por 568 Bs. Dominio y certificado SSL incluidos sin costo adicional. Elaboración Propia.

3.8.3. Costo Total del Proyecto

Una vez realizada la evaluación económica de los Costos de Desarrollo de Software y los Costos de Operación del Software, en la Tabla 72 se presenta el costo total estimado del presente proyecto. Esta tabla resume la inversión necesaria para la implementación y mantenimiento del sistema, incluyendo tanto los gastos asociados al desarrollo como los requeridos para su operación.

Tabla 72

Costos Total del Proyecto

Descripción	**Costo (Bs.)**
Costos de Desarrollo de Software	38475
Costos de Operación del Software	568
TOTAL	**39043**

Nota: La tabla muestra el costo total estimado del proyecto, incluyendo los costos de desarrollo y de operación del software. Elaboración Propia.

CAPÍTULO 4.
CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

A partir del desarrollo y evaluación del sistema, se pueden establecer las siguientes conclusiones específicas sobre el cumplimiento de cada objetivo:

- Se realizó un análisis exhaustivo mediante entrevistas al personal de la parroquia, identificándose los puntos críticos de demora y error en la transcripción manual, lo que permitió definir con precisión los requerimientos del usuario, reflejados en las 12 historias de usuario y en sus criterios de aceptación correspondientes.

- Se recopiló y se anotó un conjunto representativo de documentos reales de feligreses, incluyendo cédulas de identidad y certificados de nacimiento. Además, se generaron documentos sintéticos para conformar datasets adecuados para la detección y extracción de información, los cuales sirvieron como base sólida para el entrenamiento de los modelos.

- Se compararon distintas arquitecturas de redes neuronales: YOLOv8s y Faster R-CNN para la detección, y redes híbridas LSTM y GRU para la extracción de información en celdas de recurrencia. Finalmente, se seleccionaron aquellas que lograron los mejores indicadores de precisión, recall y F1-score en detección, así como WER y CER en extracción.

- Se desarrolló el sistema web y se integraron los modelos entrenados mediante una API construida con FastAPI, que conecta el sistema con los modelos de redes neuronales. Además, se implementó un servicio local de escaneo usando Flask, el cual permite la comunicación entre el escáner y el sistema web, facilitando que la documentación de los feligreses se envíe automáticamente al sistema para su procesamiento y extracción de datos.

- Se evaluó el sistema en condiciones operativas reales mediante el escaneo de documentación auténtica de los feligreses. A partir de esta evaluación, se concluyó que el sistema logra reducir los tiempos de los procesos administrativos.

Finalmente, el desarrollo del sistema web permitió automatizar de manera efectiva la detección y extracción de información desde documentos físicos, como cédulas de identidad y certificados de nacimiento, logrando una reducción del 17.32% en los tiempos de los procesos administrativos para el registro de datos. Asimismo, mediante validaciones automáticas y un flujo de registro optimizado, el sistema contribuye a reducir inconsistencias en la información de los feligreses en la Parroquia San Miguel Arcángel.

4.2. RECOMENDACIONES

El presente trabajo presenta recomendaciones finales para optimizar el sistema, mejorar la precisión en la gestión de registros y facilitar su escalabilidad, basadas en los hallazgos obtenidos durante su desarrollo y evaluación.

- Incorporación progresiva de nuevos documentos: Se sugiere añadir gradualmente nuevos tipos de documentos parroquiales, como certificados de bautismo, certificados de confirmación de otras parroquias. Esto permitirá que el sistema continúe automatizando procesos que actualmente requieren transcripción manual, ampliando su cobertura y utilidad.

- Implementación de un flujo de aprendizaje continuo: Se recomienda crear un mecanismo de “aprendizaje en producción” mediante el cual los documentos corregidos manualmente por los usuarios se almacenen para su uso posterior. Estos datos podrían utilizarse periódicamente para reentrenar los modelos de detección y reconocimiento, mejorando progresivamente la precisión sin necesidad de intervención técnica constante.

- Enriquecimiento del dataset y preprocesamiento: Se sugiere ampliar el dataset incorporando caracteres con acentos, ya que actualmente solo incluye letras y números estándar, así como la letra Ñ en mayúscula y minúscula. Esta mejora permitirá una mayor precisión en la extracción de información, especialmente en nombres y apellidos de los feligreses.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Amazon Web Services. (s.f.). *¿Qué es Python?* Recuperado el 9 de marzo de 2026, de https://aws.amazon.com/es/what-is/python/

Andreu, R., Ricart, J., & Valor, J. (1991). *Estrategia y Sistemas de Información.* España: McGraw-Hill.

Arana, C. (2021). Redes neuronales recurrentes: Análisis de los modelos especializados en datos secuenciales. *Serie Documentos de Trabajo*(797). Obtenido de https://www.econstor.eu/bitstream/10419/238422/1/797.pdf

Arzola Saavedra, A. (2025). Redes LSTM: Memoria a Largo Plazo en Secuencias. *Medium*. Obtenido de https://medium.com/@Alejandro.D.A.S/3-redes-lstm-memoria-a-largo-plazo-en-secuencias-0160d39ef600

Beynon-Davies, P. (2018). *Sistemas de información. Introducción a la Informática en las Organizaciones.* Reverté. Obtenido de https://www.google.com.bo/books/edition/Sistemas_de_informaci%C3%B3n/5jbeDwAAQBAJ?hl=es-419&gbpv=1

Bobadilla, J. (2020). *Machine Learning y Deep Learning: Usando Python, Scikit y Keras.* Bogotá: Ediciones de la U. Obtenido de https://books.google.es/books?hl=es&lr=lang_es&id=iAAyEAAAQBAJ&oi=fnd&pg=PA11&dq=Machine+Learning+ingenier%C3%ADa+de+sistemas&ots=Qiz5u_pM9p&sig=PvTAfkLVfcAqSSTfBtUwI2A76Kg#v=onepage&q=Machine%20Learning%20ingenier%C3%ADa%20de%20sistemas&f=false

Dobilas, S. (2022). Redes neuronales recurrentes GRU – una forma inteligente de predecir secuencias en Python. *towards data science*. Obtenido de https://towardsdatascience.com/gru-recurrent-neural-networks-a-smart-way-to-predict-sequences-in-python-80864e4fe9f6/

Duque Domingo, J., Gómez García-Bermejo, J., & Zalama Casanova, E. (2024). *Visión Artificial. Componentes de los sistemas de visión y nuevas tendencias en Deep Learning.* RA-MA. Obtenido de

https://books.google.es/books?hl=es&lr=lang_es&id=Rc4KEQAAQBAJ&oi=fnd&pg=PT309&dq=visi%C3%B3n+artificial&ots=Dr6GcmdVio&sig=sb2zFnJW05EI329s48ZM-YJLN8c#v=onepage&q=visi%C3%B3n%20artificial&f=false

Fernández González, J. (s.f.). Introducción a las metodologías ágiles. *Universitat Oberta de Catalunya*. Obtenido de https://gc.scalahed.com/recursos/files/r161r/w25941w/S02R02.pdf

Flórez, R., & Fernández, J. M. (2008). *Las Redes Neuronales Artificiales.* Netbiblo. Obtenido de https://www.google.com.bo/books/edition/Las_Redes_Neuronales_Artificiales/X0uLwi1Ap4QC?hl=es-419&gbpv=1

Folgado Galache, L., & Gallego Cano, J. C. (2024). *Digitalización Aplicada a los Sectores Productivos.* Editex. Obtenido de https://www.google.com.bo/books/edition/Digitalizaci%C3%B3n_aplicada_a_los_sectores/l5kPEQAAQBAJ?hl=es-419&gbpv=1&kptab=overview

Gao, R., Chen, M., Zhao, Z., & Ren, J. (2024). Real-time Detection Algorithm of Aircraft Landing Gear based on improved YOLOv8. *Research Gate*. Obtenido de https://www.researchgate.net/publication/381458105_Real-time_Detection_Algorithm_of_Aircraft_Landing_Gear_based_on_improved_YOLOv8

García, O., & Olivera, O. (16 de Septiembre de 2019). *Redes Neuronales artificiales: Qué son y cómo se entrenan*. Recuperado el 5 de Abril de 2025, de Xeridia: https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i

Google Open Source. (s.f.). *TensorFlow*. Recuperado el 9 de marzo de 2026, de https://opensource.google/projects/tensorflow?hl=es-419

Hagan, M., Demuth, H., Beale, M., & De Jesús, O. (2014). *Neural Network Design* (Segunda ed.). Obtenido de https://hagan.okstate.edu/NNDesign.pdf

Hernández Trasobares, A. (2003). Los sistemas de información: evolución y desarrollo. *Proyecto Social*.

Hidayatullah, P., Syakrani, N., Sholahuddin, M. R., Gelar, T., & Tubagus, R. (2025). YOLOv8 to YOLO11: A Comprehensive Architecture In-depth Comparative Review. *arXiv*. Obtenido de https://arxiv.org/abs/2501.13400

Hochreiter, S., & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*. Obtenido de https://www.researchgate.net/publication/13853244_Long_Short-Term_Memory

Holy See. (1983). *Code of Canon Law.* Obtenido de https://www.vatican.va/archive/cod-iuris-canonici/eng/documents/cic_lib4-cann834-878_en.html#PART_I.

IONOS. (10 de Agosto de 2020). *Keras: biblioteca de código abierto para crear redes neuronales*. Obtenido de https://www.ionos.com/es-us/digitalguide/online-marketing/marketing-para-motores-de-busqueda/que-es-keras/

IONOS. (3 de Enero de 2023). *¿Qué es Flask Python? Un breve tutorial sobre este microframework*. Obtenido de https://www.ionos.com/es-us/digitalguide/paginas-web/desarrollo-web/flask/#c574986

Jain, A. (2025). *Fast RCNN*. Obtenido de Medium: https://medium.com/@abhishekjainindore24/fast-rcnn-4fbe954294ef

Laravel. (s.f.). *Laravel documentation*. Obtenido de https://laravel.com/docs/12.x

Laudon, K., & Laudon, J. (1996). *Administración de los Sistemas de Información.* Prentice Hall.

Lubinus, F., Rueda, C., Marconi, B., & Arias, Y. (2021). Redes neuronales convolucionales: un modelo de Deep Learning en imágenes diagnósticas. *Universidad Autónoma de Bucaramanga*. Obtenido de https://rcr.acronline.org/index.php/rcr/article/view/161/219

Mahadevkar, S., Patil, S., & Kotecha, K. (2024). Enhancement of handwritten text recognition using AI-based. *MethodsX*. Obtenido de https://pdf.sciencedirectassets.com/306208/1-s2.0-S2215016123X0003X/1-s2.0-S2215016124001080/main.pdf?X-Amz-Security-Token=IQoJb3JpZ2luX2VjEJz%2F%2F%2F%2F%2F%2F%2F%2F%2F%2FwEaCXVzLWVhc3QtMSJIMEYCIQC8oeav9lODJQ9mKgwa9LS51ZPcIP6MYaWbm%2FQA54MptAIhAJhWS6fOtU

Marques de Sá, J. (2012). *Pattern Recognition: Concepts, Methods and Applications.* Springer Science+Business Media. Obtenido de https://www.google.com.bo/books/edition/Pattern_Recognition/CtipCAAAQBAJ?hl=es-419&gbpv=1

McCaffrey, J. (2018). Comprensión de las células LSTM mediante C#. *MSDN*. Obtenido de https://learn.microsoft.com/en-us/archive/msdn-magazine/2018/april/test-run-understanding-lstm-cells-using-csharp

Mercier, M. (14 de Enero de 2025). *¿Qué es una red neuronal profunda?* Obtenido de botpress: https://botpress.com/es/blog/deep-neural-network

Mora, M., Wang, F., Marx Gomez, J., & Duran-Limon, H. (2023). *Development Methodologies for Big Data Analytics Systems: Plan-driven, Agile, Hybrid, Lightweight Approaches.* Springer International Publishing. Obtenido de https://www.google.com.bo/books/edition/Development_Methodologies_for_Big_Data_A/h0jhEAAAQBAJ?hl=es-419&gbpv=0

Pandey, P. (s.f.). *Introducing CRAFT: Character Region Awareness for Text Detection*. Obtenido de YARSA VISION: https://www.yarsavision.com/2022/01/craft-ocr.html

Pérez Orozco, B. (2018). Inteligencia artificial. *INCyTU*. Obtenido de https://www.foroconsultivo.org.mx/INCyTU/documentos/Completa/INCYTU_18-012.pdf

Pérez, I., & Gegúndez, M. (2021). *Deep learning: fundamentos, teoría y aplicación.* Obtenido de https://www.google.com.bo/books/edition/DEEP_LEARNING/kzsvEAAAQBAJ?hl=es-419&gbpv=1

Pradel, J., & Raya, J. (2013). *Introducción a la ingeniería de software.* Universitat Oberta de Catalunya. Obtenido de https://openaccess.uoc.edu/bitstream/10609/69245/5/Ingenier%C3%ADa%20del%20software_M%C3%B3dulo%201_%20Introducci%C3%B3n%20a%20la%20ingenier%C3%ADa%20del%20software.pdf

Pradhan, A., Pradhan, M., & Pradhan, R. (2022). Comparative Analysis of RCNN and Faster R-CNN for Text Detection in a Poor Quality Topographic Map. *ResearchGate*. Obtenido de

https://www.researchgate.net/publication/366451804_Comparative_Analysis_of_RCNN_and_Faster_R-CNN_for_Text_Detection_in_a_Poor_Quality_Topographic_Map

Pravinkumar, S., Rushabh, J., Rutuj, R., Kunal, D., & Ashutosh, B. (2023). Novel Study on Localization in Scene Text Detection. *INTERNATIONAL JOURNAL OF NEXT-GENERATION COMPUTINGINTERNATIONAL JOURNAL OF NEXT-GENERATION COMPUTING*. Obtenido de https://www.researchgate.net/publication/368999369_Novel_Study_on_Localization_in_Scene_Text_Detection

Pressman, R. (2010). *Ingeniería de software: un enfoque práctico.* McGraw-Hill. Obtenido de https://www.javier8a.com/umb/analisis/Libro_Pressman_7.pdf

Quispe Mamani, N. U. (2017). Redes neuronales artificiales, metodología de desarrollo y aplicaciones. *ResearchGate*. Obtenido de https://www.researchgate.net/publication/359512026_Redes_neuronales_artificiales_metodologia_de_desarrollo_y_aplicaciones

Raschka, S., & Mirjalili, V. (2019). *Python Machine Learning.* Marcombo. Obtenido de https://books.google.es/books?hl=es&lr=lang_es&id=5EtOEAAAQBAJ&oi=fnd&pg=PT5&dq=Machine+Learning&ots=erl-SxZGG7&sig=VQkZJkpDEfOV6Cy_dMcCAsnkgZQ#v=onepage&q=Machine%20Learning&f=false

Redmon, J., Santosh, D., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. *arXiv*. Obtenido de https://arxiv.org/pdf/1506.02640

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv*. Obtenido de https://arxiv.org/abs/1506.01497

Restrepo, D., Viloria, J., & Robles, C. (2021). *El camino a las redes neuronales artificiales.* Unimagdalena. Obtenido de https://books.google.es/books?hl=es&lr=lang_es&id=hTSDEAAAQBAJ&oi=fnd&pg=PT12&dq=redes+neuronales&ots=ytpH3a1OfP&sig=vpvC19AH6sL3VA_fQ7RfNp2j3yg#v=onepage&q=redes%20neuronales&f=false

Rouhiainen, L. (2018). *Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro.* Planeta. Obtenido de https://planetadelibrosar0.cdnstatics.com/libros_contenido_extra/40/39307_Inteligencia_artificial.pdf

Sancho Caparrini, F. (s.f.). *Redes Neuronales: una visión superficial*. Recuperado el 5 de Abril de 2025, de https://www.cs.us.es/~fsancho/Blog/posts/Redes_Neuronales.md

Sommerville, I. (2011). *Ingeniería de Software* (Novena ed.). Pearson. Obtenido de https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf

Sotaquirá, M. (3 de Septiembre de 2018). *¿Qué es una Red Neuronal?* Recuperado el 5 de Abril de 2025, de Codificando Bits: https://codificandobits.com/blog/que-es-una-red-neuronal/

Sotil, D. A. (28 de Abril de 2022). *Aprendizaje Profundo: Redes Neuronales Convolucionales CNN*. Recuperado el 6 de Abril de 2025, de RPubs: https://rpubs.com/dsotilccama/CNN_Markdown

Ultralytics. (s.f.). Recuperado el 2025, de Documentación de Ultralytics YOLO: https://docs.ultralytics.com/es/

Valbuena Castro, R. E. (2021). *Inteligencia Artificial: Investigación Científica Avanzada Centrada en Datos.* Obtenido de https://www.google.com.bo/books/edition/Inteligencia_Artificial/SoMTEAAAQBAJ?hl=es-419&gbpv=1

Wang, X., Wei, G., Chen, S., & Liu, J. (2023). An efficient weakly semi-supervised method for object automated annotation. Obtenido de https://link.springer.com/article/10.1007/s11042-023-15305-0

Yadav, N. (8 de Enero de 2024). *Introducción a FastAPI*. Obtenido de Medium: https://medium.com/@yaduvanshineelam09/introduction-to-fastapi-123c0b2778a5

Zulqarnain, M., Ghazali, R., Mohmad Hassim, Y. M., & Rehan, M. (2020). A comparative review on deep learning models. *Indonesian Journal of Electrical Engineering and Computer Science*. Obtenido de https://d1wqtxts1xzle7.cloudfront.net/77654250/13839-

libre.pdf?1640841929=&response-content-disposition=inline%3B+filename%3DA_comparative_review_on_deep_learning_mo.pdf&Expires=1756673176&Signature=abchHOpiNpQVEd3dV4setiI2LoTTwfwQkNhqIcPsEYlLLOYH6bC8nn~L

GLOSARIO DE TÉRMINOS

GLOSARIO

- **Capas de pooling**

Capas que reducen la dimensionalidad seleccionando valores representativos de regiones específicas, como máximos o promedios.

- **Casos de prueba**

Escenarios definidos para verificar el funcionamiento del software mediante condiciones de entrada, procesos y resultados esperados.

- **Casos de uso**

Modelos que describen interacciones entre actores y el sistema, representando funcionalidades y objetivos del usuario.

- **CER**

Métrica que mide el porcentaje de errores a nivel de carácter en una transcripción comparada con el texto real.

- **Conjunto de entrenamiento**

Subconjunto de datos utilizado para ajustar los parámetros de un modelo durante el proceso de aprendizaje.

- **Conjunto de prueba**

Subconjunto de datos reservado para evaluar el rendimiento final de un modelo una vez entrenado.

- **Conjunto de validación**

Datos empleados para ajustar hiperparámetros y monitorear el rendimiento durante el entrenamiento del modelo.

- **Dataset**

Colección estructurada de datos empleada para entrenar, validar y evaluar modelos de aprendizaje automático.

- **Épocas**

Número de veces que un modelo recorre por completo el conjunto de entrenamiento durante el aprendizaje.

- **Función de activación**

Función matemática que determina la salida de una neurona e introduce no linealidad en las redes.

- **Kernel**

Matriz empleada en operaciones de convolución para extraer características específicas de una entrada.

- **OCR**

Tecnología que convierte imágenes con texto en contenido digital editable mediante reconocimiento automático de caracteres.

- **Padding**

Relleno aplicado a los bordes de una matriz para conservar dimensiones en operaciones de convolución.

- **Sobreajuste**

Situación donde un modelo aprende patrones del entrenamiento de forma excesiva y presenta bajo desempeño en datos nuevos.

- **WER**

Métrica que mide errores a nivel de palabra en una transcripción comparada con el texto de referencia.

ANEXOS

ANEXO A

CARTA DE SOLICITUD DE TUTORÍA ACADÉMICA

La Paz, 23 de febrero de 2026

Señor

Ing. Roberto Carlos Mamani Mayta
Presente. -

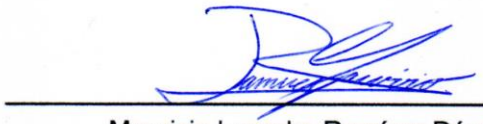
Ref.: **NOMBRAMIENTO DE TUTORÍA DE PROYECTO DE GRADO**

Distinguido Ingeniero Mamani:

Mediante la presente y en mérito a su amplio conocimiento y experiencia tengo a bien hacer llegar a su persona, la solicitud para que acepte la TUTORIA de mi trabajo de investigación titulado **"SISTEMA INTELIGENTE DE RECONOCIMIENTO ÓPTICO DE CARACTERES BASADO EN REDES NEURONALES PARA LA DETECCIÓN Y EXTRACCIÓN DE INFORMACIÓN EN DOCUMENTOS DE IDENTIDAD"**, con el correspondiente compromiso mío de cumplir con:

- Presentaciones dentro de los plazos establecidos por la universidad.
- Realización de presentaciones a mi revisor, considerando que cada uno de los avances será previamente consultado con su persona.
- Informar a usted, de todas las actividades y problemas en relación a mi investigación.
- Evitar todo acto arbitrario sobre las actividades de mi investigación

Presentado mi compromiso y responsabilidad, para con usted y con mi trabajo, me despido quedando a la espera de una respuesta positiva a la presente solicitud.



Mauricio Leandro Ramírez Dávalos

ANEXO B

CARTA DE ACEPTACIÓN DE TUTORÍA

La Paz, 23 de febrero de 2026

Señora
Lic. Martha Heredia Poma
COORDINADORA DE LA CARRERA DE INGENIERÍA DE SISTEMAS
UNIVERSIDAD LA SALLE BOLIVIA
Presente. -

Ref.: **ACEPTACIÓN DE TUTORÍA**

Distinguida Licenciada Heredia:

En respuesta a la solicitud realizada por el estudiante **MAURICIO LEANDRO RAMÍREZ DÁVALOS** de la carrera de Ingeniería de Sistemas, quien me presentó el documento de su proyecto de grado titulado: **"SISTEMA INTELIGENTE DE RECONOCIMIENTO ÓPTICO DE CARACTERES BASADO EN REDES NEURONALES PARA LA DETECCIÓN Y EXTRACCIÓN DE INFORMACIÓN EN DOCUMENTOS DE IDENTIDAD",** y después de haber realizado el análisis del tema y aceptado los compromisos propuestos por el estudiante, acepto hacerme cargo de la tutoría y me comprometo a cumplir las responsabilidades inherentes a esta función.

Sin otro particular, saludo a usted atentamente.



Ing. Roberto Carlos Mamani Mayta

ANEXO C

CARTA DE ACEPTACIÓN DE LA INSTITUCIÓN PARA CASO DE ESTUDIO



PARROQUIA SAN MIGUEL ARCANGEL
Congregación de la Resurrección de Nuestro Señor Jesucristo

La Paz,
15 de abril del 2025

Señores
Universidad LA SALLE BOLIVIA
Presente.-

De mi mayor consideración:

Por medio del presente tenemos a bien informarles que el Sr. Mauricio Leandro Ramírez Dávalos, a quien corresponde la cédula de identidad 6740048 estudiante de séptimo semestre de la carrera de INGENIERA DE SISTEMAS, de la UNIVERSIDAD LA SALLE, ha sido aceptado para desarrollar su Proyecto de Grado: **SISTEMA DE EXTRACCIÓN DE INFORMACIÓN DOCUMENTAL UTILIZANDO RECONOCIMIENTO ÓPTICO DE CARACTERES CON REDES NEURONALES**, en la parroquia San Miguel Arcángel, bajo mi supervisión Rvdo. P. Slawomir Siemieniec, CR.

Para este efecto, la parroquia colaborará con afabilidad al Sr. Ramírez, proporcionando la información necesaria para el buen desarrollo del proyecto presentado.

Atentamente,

PARROQUIA DE SAN MIGUEL ARCANGEL
LA PAZ - BOLIVIA
CONGREGACIÓN DE LA RESURRECCIÓN

Rvdo. Slawomir Siemieniec, CR
PÁRROCO

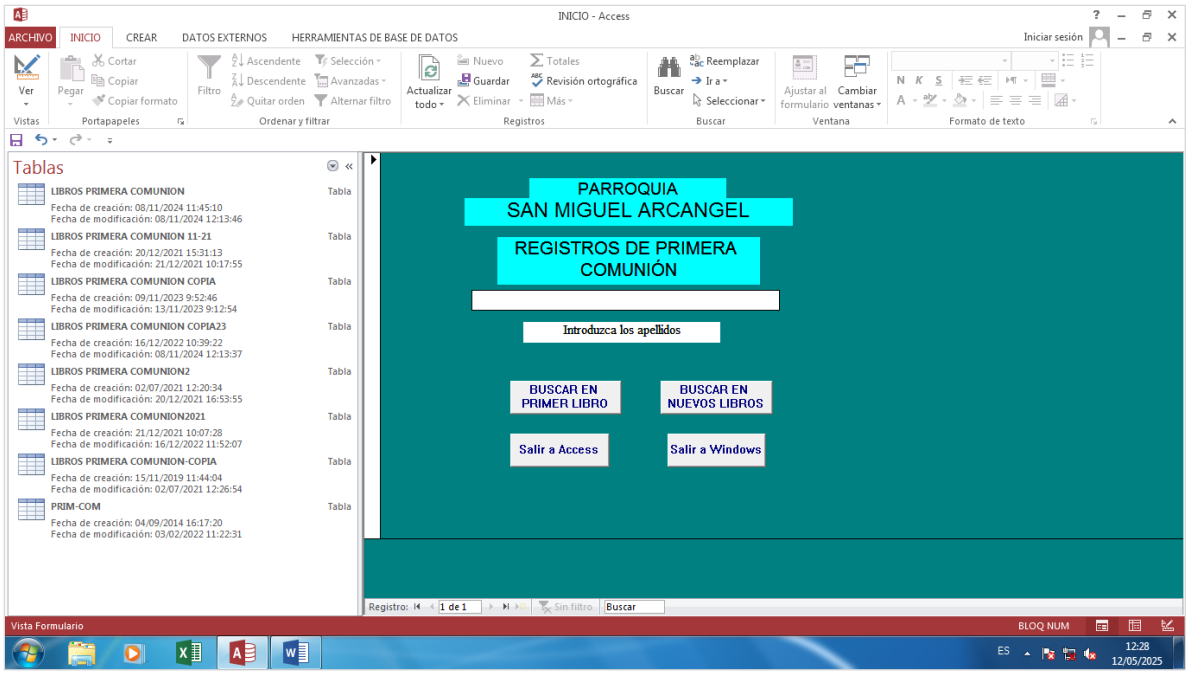
Calacoto Calle 21 N° 8066
La Paz - Bolivia

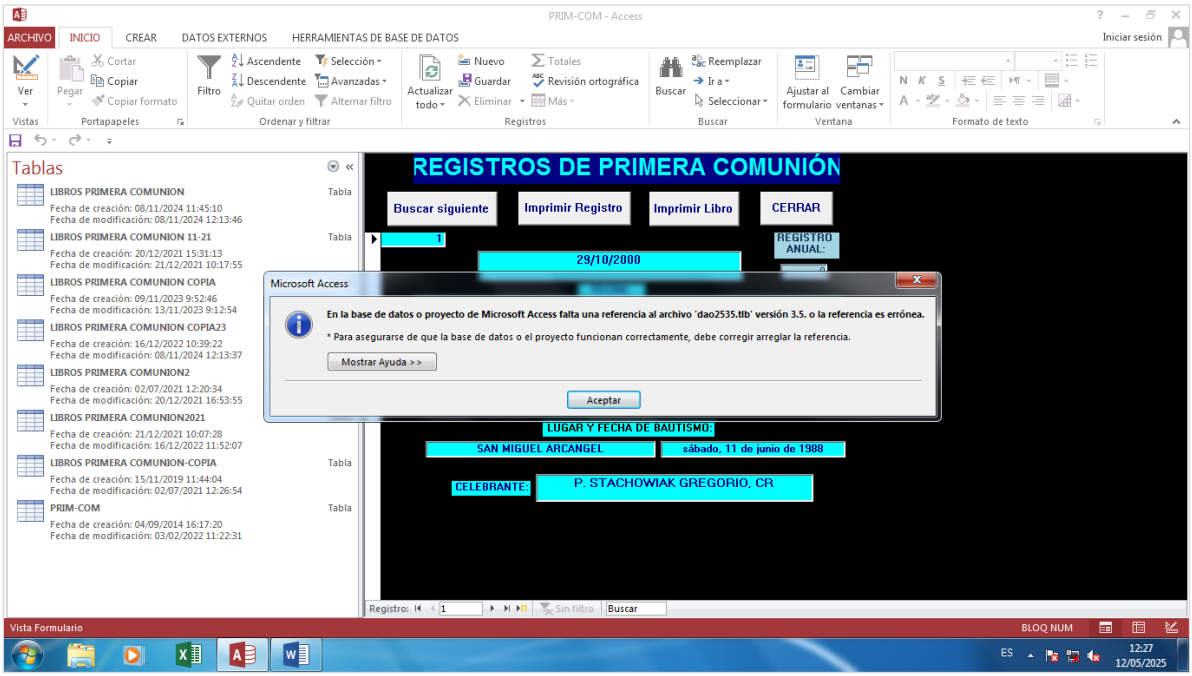
2792002
2792386

www.facebook.com/
ParroquiaSanMiguelArcangel.mx/

ANEXO D

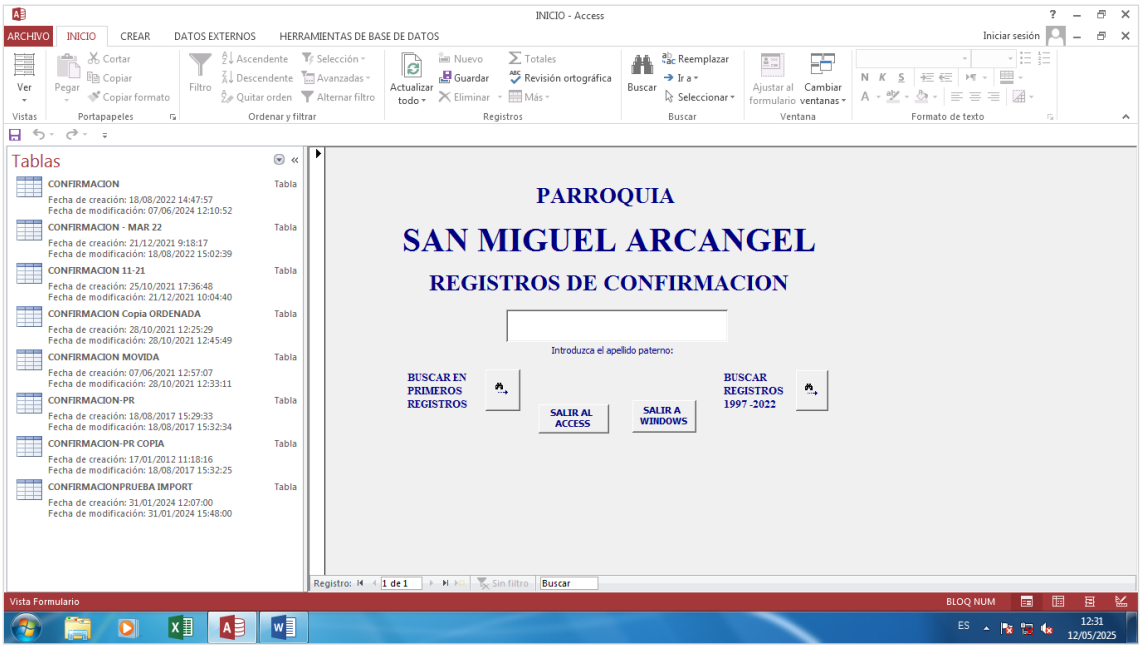
CAPTURAS DE REGISTROS DE PRIMERA COMUNIÓN DEL SISTEMA ACTUAL

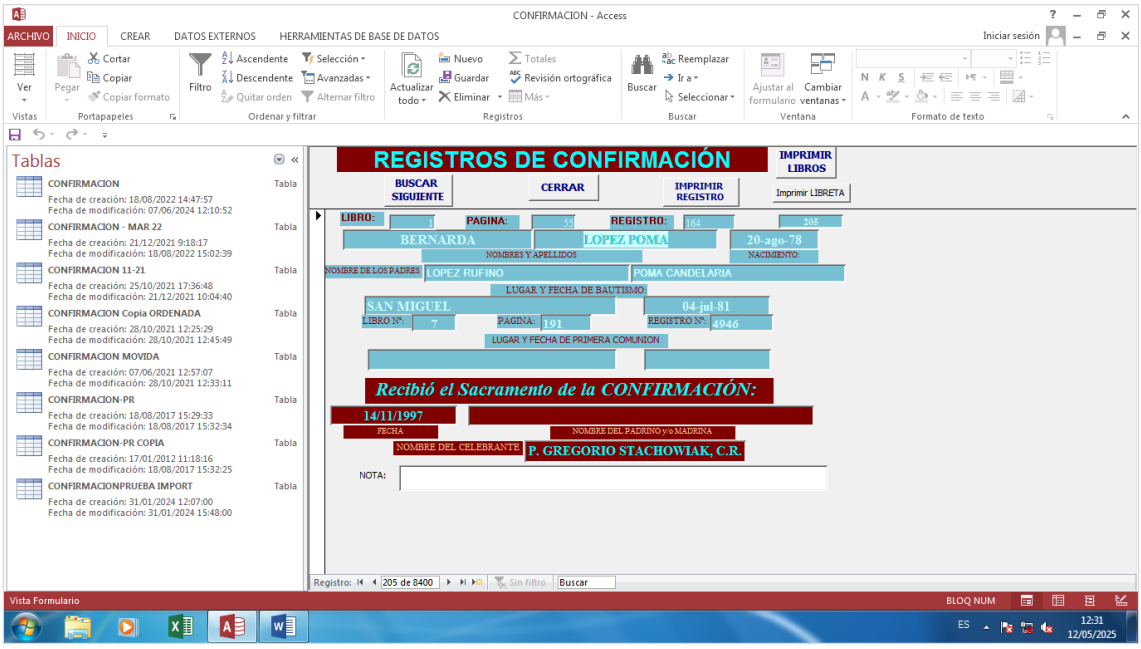




ANEXO E

CAPTURA DE REGISTROS DE CONFIRMACIÓN DEL SISTEMA ACTUAL





ANEXO F

ACTA DE VALIDACIÓN Y APROBACIÓN DE REQUERIMIENTOS DE USUARIO

ACTA DE VALIDACIÓN Y APROBACIÓN DE REQUERIMIENTOS DE USUARIO

Proyecto:	Sistema de Detección y Extracción de Información de Documentación Personal mediante Redes Neuronales
Institución:	Parroquia San Miguel Arcángel
Desarrollador:	Univ. Mauricio Leandro Ramírez Dávalos
Lugar y fecha:	La Paz, 25 de julio, 2025

Por la presente acta se deja constancia de que, tras las entrevistas realizadas con el personal de la parroquia y la revisión conjunta de los requerimientos funcionales levantados, las partes involucradas validan y aprueban los requerimientos descritos a continuación, declarando que representan las necesidades actuales en el manejo de información y documentación parroquial.

OBJETIVO DEL LEVANTAMIENTO

Establecer y confirmar los requerimientos funcionales necesarios para la construcción del sistema, asegurando que reflejen fielmente las necesidades reales de gestión documental, registro sacramental y automatización administrativa de la Parroquia San Miguel Arcángel.

REQUERIMIENTOS VALIDADOS

- El sistema debe permitir el acceso únicamente al personal con credenciales válidas y permitir la gestión de usuarios.
- El sistema debe permitir asignar y modificar roles a los usuarios, cada uno con permisos específicos.
- Permitir el registro manual de feligreses, incluyendo datos personales y documentación adjunta, con validaciones y edición de registros según permisos.
- Automatizar la identificación y extracción de datos de documentos escaneados (certificados de nacimiento y cédulas de identidad).
- Permitir crear, editar, eliminar y consultar cursos sacramentales con descripción, horarios, costo y sacramento correspondiente.
- Permitir crear y modificar sesiones de cursos con tema, fecha, horario y lugar.
- Permitir inscribir feligreses a uno o varios cursos y consultar cursos disponibles e inscritos.

- Registrar pagos y asistencia de feligreses, con visualización de información relevante de cada participante.
- Establecer fecha, hora y lugar de celebraciones (comunitarias o particulares) y permitir reprogramación.
- Registrar y mantener actualizado un calendario de celebraciones sacramentales.
- Migrar información de inscripciones a registros oficiales solo para feligreses que culminaron el curso.
- Generar certificados de sacramentos según formatos oficiales, con previsualización e impresión.

DECLARACIÓN DE CONFORMIDAD

Las partes certifican que:

- Los requerimientos arriba detallados fueron explicados, revisados y comprendidos en su totalidad.
- Su contenido refleja fielmente las necesidades actuales de la parroquia.
- Se autoriza su utilización como base oficial para el diseño y desarrollo del sistema.

PARROQUIA DE "SAN MIGUEL ARCANGEL" LA PAZ - BOLIVIA CONGREGACIÓN DE LA RESURRECCIÓN

Rvdo. P. Slawomir Siemieniec, CR
PARROCO SAN MIGUEL ARCANGEL

Univ. Mauricio Leandro Ramírez Dávalos
DESARROLLADOR DEL PROYECTO

ANEXO G

IMPLEMENTACIÓN DEL MODELO HÍBRIDO CNN-GRU PARA EXTRACCIÓN

```
# Entradas
input_images = Input(shape=(IMG_WIDTH, IMG_HEIGHT, 1), name="image")
input_labels = Input(shape=(None, ), name="label")

# Capas convolucionales
conv_1 = Conv2D(64, 3, strides=1, padding="same",
kernel_initializer="he_normal", activation="relu", name="conv_1")(input_images)

conv_2 = Conv2D(32, 3, strides=1, padding="same",
kernel_initializer="he_normal", activation="relu", name="conv_2")(conv_1)
max_pool_1 = MaxPool2D(pool_size=(2, 2), strides=(2, 2))(conv_2)
conv_3 = Conv2D(64, 3, strides=1, padding='same', activation='relu',
kernel_initializer='he_normal', name="conv_3")(max_pool_1)
conv_4 = Conv2D(32, 3, strides=1, padding='same', activation='relu',
kernel_initializer='he_normal', name="conv_4")(conv_3)
max_pool_2 = MaxPool2D(pool_size=(2, 2), strides=(2, 2))(conv_4)

# Codificación
reshape = Reshape(target_shape=((IMG_WIDTH//4), (IMG_HEIGHT//4)*32),
name="reshape_layer")(max_pool_2)
dense_encoding = Dense(64, kernel_initializer="he_normal", activation="relu",
name="enconding_dense")(reshape)
dense_encoding_2 = Dense(64, kernel_initializer="he_normal", activation="relu",
name="enconding_dense_2")(dense_encoding)
dropout = Dropout(0.4)(dense_encoding_2)

# Decodificador RNN bidireccional
gru1 = Bidirectional(GRU(160, return_sequences=True,
kernel_initializer='he_normal'), merge_mode='sum', name="gru1")(dropout)
gru2 = Bidirectional(GRU(96, return_sequences=True,
kernel_initializer='he_normal'), merge_mode='concat', name="gru2")(gru1)

# Salida
output = Dense(len(char_to_num.get_vocabulary())+1, activation="softmax",
name="output_dense")(gru2)

# Capa de pérdida CTC
ctc_loss_layer = CTCLayer()(input_labels, output)

# Modelo de entrenamiento
train_model = Model(inputs=[input_images, input_labels], outputs=
[ctc_loss_layer])

# Modelo de predicción
prediction_model = Model(inputs=[input_images], outputs=output)
```

ANEXO H

IMPLEMENTACIÓN DEL MODELO HÍBRIDO CNN-LSTM PARA EXTRACCIÓN

```
# Entradas
input_images = Input(shape=(IMG_WIDTH, IMG_HEIGHT, 1), name="image")
input_labels = Input(shape=(None, ), name="label")

# Capas convolucionales
conv_1 = Conv2D(64, 3, strides=1, padding="same",
kernel_initializer="he_normal", activation="relu", name="conv_1")(input_images)

conv_2 = Conv2D(32, 3, strides=1, padding="same",
kernel_initializer="he_normal", activation="relu", name="conv_2")(conv_1)
max_pool_1 = MaxPool2D(pool_size=(2, 2), strides=(2, 2))(conv_2)
conv_3 = Conv2D(64, 3, strides=1, padding='same', activation='relu',
kernel_initializer='he_normal', name="conv_3")(max_pool_1)
conv_4 = Conv2D(32, 3, strides=1, padding='same', activation='relu',
kernel_initializer='he_normal', name="conv_4")(conv_3)
max_pool_2 = MaxPool2D(pool_size=(2, 2), strides=(2, 2))(conv_4)

# Codificación
reshape = Reshape(target_shape=((IMG_WIDTH//4), (IMG_HEIGHT//4)*32),
name="reshape_layer")(max_pool_2)
dense_encoding = Dense(64, kernel_initializer="he_normal", activation="relu",
name="enconding_dense")(reshape)
dense_encoding_2 = Dense(64, kernel_initializer="he_normal", activation="relu",
name="enconding_dense_2")(dense_encoding)
dropout = Dropout(0.4)(dense_encoding_2)

# Decodificador RNN bidireccional
lstm_1 = Bidirectional(LSTM(128, return_sequences=True, dropout=0.25),
name="bidirectional_lstm_1")(dropout)
lstm_2 = Bidirectional(LSTM(64, return_sequences=True, dropout=0.25),
name="bidirectional_lstm_2")(lstm_1)

# Salida
output = Dense(len(char_to_num.get_vocabulary())+1, activation="softmax",
name="output_dense")(lstm_2)

# Capa de pérdida CTC
ctc_loss_layer = CTCLayer()(input_labels, output)

# Modelo de entrenamiento
train_model = Model(inputs=[input_images, input_labels], outputs=
[ctc_loss_layer])

# Modelo de predicción
prediction_model = Model(inputs=[input_images], outputs=output)
```

ANEXO I

ACTA DE VALIDACIÓN DE PRUEBAS DEL SISETMA

ACTA DE VALIDACIÓN DE PRUEBAS DEL SISTEMA

Proyecto:	Sistema de Detección y Extracción de Información de Documentación Personal mediante Redes Neuronales
Institución:	Parroquia San Miguel Arcángel
Desarrollador:	Univ. Mauricio Leandro Ramírez Dávalos
Lugar y fecha:	La Paz, 21 de enero, 2026

OBJETIVO DE LA VALIDACIÓN

Evaluar el funcionamiento del sistema en la captura y procesamiento automatizado de datos provenientes de documentos de identidad, verificando su desempeño en condiciones reales de uso dentro de la parroquia.

RESULTADOS DE LA EVALUACIÓN

Durante las pruebas se comparó el proceso manual tradicional (registro en Excel y posterior transcripción a Access) con el uso del sistema, evidenciándose una reducción en los tiempos de ingreso de datos:

- Bautismo: 20.13% de reducción
- Primera Comunión: 6.19% de reducción
- Confirmación: 25.10% de reducción
- Matrimonio: 17.84% de reducción

Reducción promedio general: 17.32%

DECLARACIÓN DE CONFORMIDAD

Las partes certifican que:

- Las pruebas fueron realizadas en condiciones reales dentro de la parroquia.
- El sistema fue utilizado por el personal encargado del registro.
- Los resultados obtenidos evidencian una mejora en la eficiencia del ingreso de datos.



Rvdo. P. Sławomir Siemieniec, CR
PÁRROCO SAN MIGUEL ARCÁNGEL

Univ. Mauricio Leandro Ramírez Dávalos
DESARROLLADOR DEL PROYECTO