

Editorial

Para la revista de Programación II de la carrera de Ingeniería de Sistemas llegar al número 7 es un gran logro, con la característica de tener esta vez 12 artículos, describiendo los programas que se llevaron a cabo en la gestión 02/2012.

En este número hay diversos tipos de programas desde test 's de verificación de verificación si las redes sociales son un vicio, juegos o un tutorial de lenguajes de programación.

Se felicita a los estudiantes que coadyuvaron en la realización de los artículos y los programas durante este semestre, animándoles a continuar con el mismo empeño y las ganas demostradas.

Continúa el compromiso de ir mejorando cada vez más, en la edición de la revista.

Atte.

Walter Carvajal

ORDENADOR EN CARPETAS

Omar Castro Quisbert

Universidad La Salle

Ommy:osi@hotmail.com

RESUMEN

Con los conocimientos en programación II he elaborado un programa que ordene todos los archivos de diferentes categorías como mp3, videos, Word, aplicaciones e imágenes.

I. INTRODUCCION

El programa tiene como aplicación copiar a diferentes carpetas ordenadamente asignados.

II. MARCO TEORICO

A la hora de hacer backups, copiar ficheros o tener que liberar espacio de disco, es cuando uno se dé cuenta de la cantidad de archivos que día a día y año a año vamos acumulando en el ordenador. Archivos del sistema, del trabajo (equipo, clientes...), de ocio (videos, música...) y personales (imágenes...). Tener una consistente organización a la hora de nombrarlos y clasificarlos te facilita el trabajo y la vida. Pero es algo que hay que hacer

expresamente, poniendo deliberado cuidado en ello.

Una buena organización-clasificación te ahorra tiempo, separa lo importante de lo menos importante, evita quebraderos de cabeza y hace que te desgastes en las cosas que de verdad merecen la pena. Organizar bien los archivos de tu ordenador es como cuidar la presión de los neumáticos de tu coche. Un *mal* necesario que hace que lo demás vaya mejor y con este programa será más fácil hacerlo.

III. ESTRUCTURA DE UN OBJETO

La estructura de carpetas hace referencia a la forma en que están organizadas las carpetas a partir de un determinado directorio (que puede ser el directorio raíz o cualquier otro).

Es decir, el nombre de las carpetas y las diferentes subcarpetas. La mejor

35

forma de visualizar la estructura de las carpetas es utilizando un diagrama de árbol. (Ver Árbol)

La definición de Directorio raíz aquí:
Directorio raíz.

En cuanto a la búsqueda de archivos, ésta puede hacerse con herramientas especiales, como las que ofrece Windows presionando F3, permitiéndonos buscar por nombre de archivo, fecha de creación, modificación, y otras propiedades.

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus

descendientes a través de la herencia.

IV. CLASES Y OBJETOS

Estos objetos fueron utilizados:

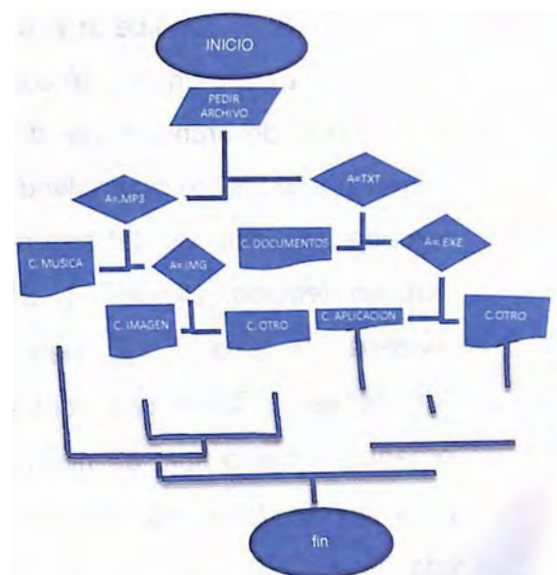
- Listbox
- Label
- System
- Button
- PictureBox
- Form
- Richtext

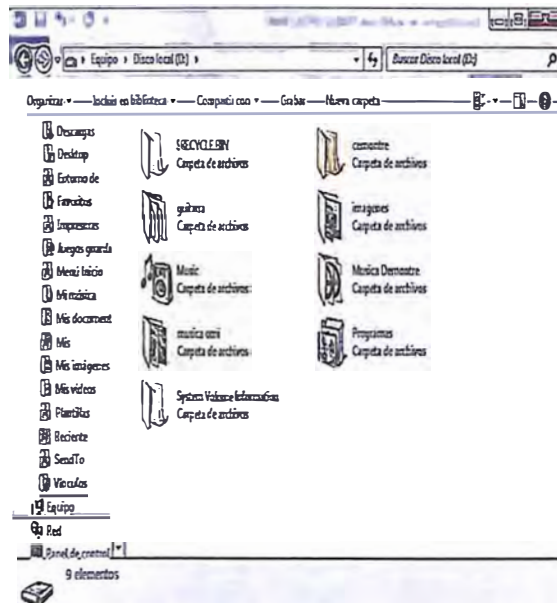
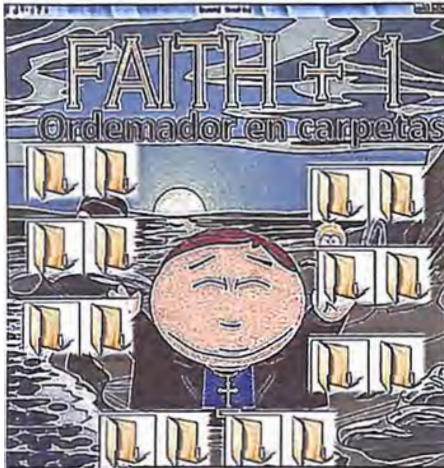
36

Con sus respectivas clases predefinidas en el lenguaje.

ALGORITMO

Sistema autenticación:





V. CONCLUSION

37

Ayuda a facilitar ordenar la pc también en encontrar virus y algún archivo que no podemos encontrar.

Toda nuestra actividad sobre el ordenador, por muy pequeña que sea, va dejando rastros que con la acumulación de meses o incluso años, se vuelven algo terrible. Esos rastros se denominan los archivos temporales, que se depositan en una carpeta con ese nombre. Con tan solo unas horas de uso y ya quedan archivos temporales depositados en la carpeta de Windows.

Debes imaginar a los archivos temporales como un monstruo que se hace más grande cada vez que le das de comer. Pero hay otra cosa adicional que no puedes tomar a la ligera: virus y troyanos también se alojan en la carpeta de archivos temporales para esconderse del antivirus o de cualquier medida de seguridad, ya que siempre buscan huecos en nuestro sistema operativo para ocultarse y hacer de las suyas.

VI. RECOMENDACIONES

No tener archivos ocultos.

Necesitas visual studio 2010 para
que arranque el programa.

VII. BIBLIOGRAFIA

<http://thinkwasabi.com/2010/10/truco>

s-

Hecho por Omar Castro Quisbert

http://www.aulaclic.es/winxp/b_4_1_1.htm

TUTORIALES DE LENGUAJES DE PROGRAMACIÓN

Cristhian Rodrigo Callizaya Quenta

rodrigo_crisamix@hotmail.com

RESUMEN

Terminando el semestre de programación II (Programación orientada a objetos c#) con el Ing. Walter Carvajal, se realizó presente proyecto tutoriales de los lenguajes de programación (que ayudara a fortalecer el conocimiento de la sintaxis, trucos, artificios, secretos, teoría, etc., de algunos lenguajes de programación c++, Ruby, HTML 5, android, eclipse, python, Delphi.

I. INTRODUCCIÓN

Lenguaje de programación es el idioma utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Hay muchos lenguajes de programación, pero para programar no es necesario conocer todos los lenguajes, es igual que cuando hablamos, podemos comunicarnos en español aunque no sepamos

Alemán. Aunque la palabra debería ser idioma ya que lenguaje realmente abarca todos los idiomas pero en computación equivocadamente se usa el término lenguaje cuando el término correcto es idiomas de programación.

En la actualidad los lenguajes de programación están escritos para ser comprensibles por el ser humano, a este código se le llama **código fuente**, pero no es comprendido por la máquina ya que esta solo maneja el lenguaje binario.

La compilación es el proceso de traducir un programa en código fuente a programa en **código objeto** (que usa el lenguaje binario), el programa encargado de compilar se llama **compilador**. La mayoría de software de programación trae su propio compilador.

La solución de problemas mediante el uso de un computador nos lleva a desarrollar.

Programas o aplicaciones, la construcción de estos programas debe ser realizada mediante

Una metodología, los pasos para la construcción de programas son:

1. Diseñar el algoritmo, consiste en analizar el problema y describir los instrucciones de

forma ordenada que resuelven el problema.

2. Codificar de forma sistemática la secuencia de instrucciones en un lenguaje.

3. Ejecutar y validar el programa en un computador (verificación)

Para codificar de forma sistemática un algoritmo se necesita de un lenguaje

programación que la computadora comprenda.

II. MARCO TEÓRICO

• Paradigma Imperativo:

Describe la programación como una secuencia instrucciones o comandos que cambian el

Estado de un programa. El código máquina en general está basado en el paradigma

Imperativo. Su contrario es el paradigma declarativo. En este paradigma se incluye el

Paradigma procedimental (procedural) entre otros. [2]

• Paradigma Declarativo:

No se basa en el cómo se hace algo (cómo se logra un objetivo paso a paso), sino que describe

(Declara) cómo es algo. En otras palabras, se enfoca en describir las propiedades de la

Solución buscada, dejando indeterminado el algoritmo (conjunto de instrucciones) usado

Para encontrar esa solución. Es más complicado de implementar que el paradigma

Imperativo, tiene desventajas en la eficiencia, pero ventajas en la solución de determinados

Problemas.

• Paradigma Estructurado:

La programación estructurada se basa en una metodología de desarrollo de programas

llamada refinamientos sucesivos: Se plantea una operación como un todo y se divide en

Segmentos más sencillos o de menor complejidad. Una vez terminado todos los

Programadores. Si se ha utilizado adecuadamente la programación estructurada, esta

integración debe ser sencilla y no presentar problemas al integrar la misma, y de

Presentar algún problema, será rápidamente detectable para su corrección.

La representación gráfica de la programación estructurada se realiza a través de diagramas de

flujo o chart, el cual representa el programa con sus entradas, procesos y salidas.

La programación estructurada propone segregar los procesos en estructuras lo más simple posibles, las cuales se conocen como secuencia, selección e interacción. Ellas están disponibles en todos los lenguajes modernos de programación imperativa en forma de sentencias. Combinando esquemas sencillos se pueden llegar a construir sistemas amplios y

Complejos pero de fácil entendimiento.

- Paradigma Orientado a Objetos:

La programación orientada a objetos, intenta simular el mundo real a través del

Significado de objetos que contiene características y funciones. Los lenguajes orientados a objetos se clasifican como lenguajes de quinta generación.

En la Programación Orientada a Objetos (POO según siglas en inglés) se definen los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos) comportamiento (esto es, procedimientos o métodos) identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa

UNIVERSIDAD LA SALLE 02/2012

como un conjunto de estos objetos, que colaboran entre

III. CLASES Y OBJETOS

Estos objetos fueron utilizados:

-Button

-PictureBox

-Form

-RichText

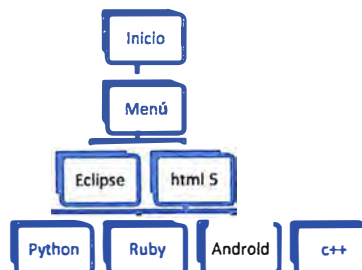
-Listbox

-Label

-System

Con sus respectivas clases predefinidas en el lenguaje.

IV. ALGORITMO

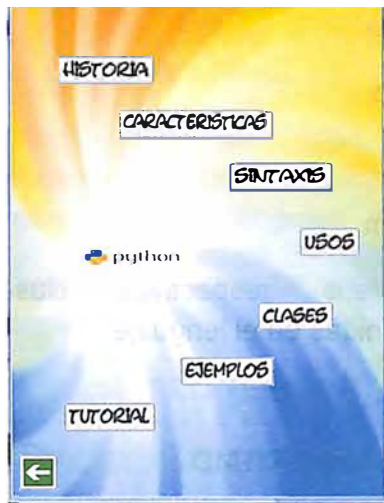


V. PRUEBA DE CORRIDO

Menú de opciones



Ventana de cada tema



VI. CONCLUSIÓN

Lenguajes compilados:

Naturalmente, un programa que se escribe en un lenguaje de alto nivel también tiene que

Traducirse a un código que pueda utilizar la máquina.

Los programas traductores que pueden realizar esta operación se llaman compiladores. Éstos,

como los programas ensambladores avanzados, pueden generar muchas líneas de código

de máquina por cada proposición del programa fuente. Se requiere una corrida de

compilación antes de procesar los datos de un problema.

Los compiladores son aquellos cuya función es traducir un programa escrito en un

Determinado lenguaje a un idioma que la computadora entienda (lenguaje máquina con

Código binario).

Al usar un lenguaje compilado (como lo son los lenguajes del popular Visual Studio de

Microsoft), el programa desarrollado nunca se ejecuta mientras haya errores, sino hasta

que luego de haber compilado el programa, ya no aparecen errores en el código.

- Lenguajes interpretados:

Se puede también utilizar una alternativa diferente de los compiladores para traducir

Lenguajes de alto nivel. En vez de traducir el programa fuente y grabar en forma

Permanente el código objeto que se produce durante la corrida de compilación para

Utilizarlo en una corrida de producción futura, el programador sólo carga el programa

Fuente en la computadora junto con los datos que se van a procesar. A continuación, un

Programa intérprete, almacenado en el sistema operativo del disco, o incluido de manera

Permanente dentro de la máquina, convierte cada proposición del programa fuente en lenguaje de máquina conforme vaya siendo necesario durante el proceso de los datos. No se

graba el código objeto para utilizarlo posteriormente.

La siguiente vez que se utilice una instrucción, se le debe interpretar otra vez y traducir

VII. RECOMENDACIONES

Se sugiere implementar usando google para complementar sobre mas lenguajes de programación, video tutoriales, audio libros y otros materiales didácticos para mejorar el aprendizaje.

VIII. BIBLIOGRAFÍA

http://es.wikipedia.org/wiki/Lenguaje_de_programacion

<http://www.alegsaonline.com/art/13.php>

<http://lenguajes-de-programacion.com/programacion-estructurada.shtml>

<http://lenguajes-de-programacion.com/programacion-orientada-a-objetos.shtml>

TEST ADICCIONES II

Luigi Pedro Enriquez Paz

luigivenyi@hotmail.com

Universidad La Salle

RESUMEN

Con los conocimientos que he adquirido en la materia de sistemas de información programación 2, y con el manejo del lenguaje C# y con la colaboración del docente Lic. Walter Carvajal, se presentara un programa que nos pone a disposición dos test de adicciones al facebook y a los celulares.

I. INTRODUCCIÓN

Bueno para hablar de los test primero se tiene que enterar que una adicción (del latín *addictio*) es una enfermedad física y psicoemocional, según la Organización Mundial de la Salud. En el sentido tradicional es una dependencia hacia una sustancia, actividad o relación (codependencia) y en este caso se selecciono el Facebook y los Celulares como tema principal ya que son las adicciones del siglo XXI

II. MARCO TEÓRICO

Actualmente una de las áreas más candentes en la industria y en el ámbito académico es la orientación

a objetos. La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases. Muchos lenguajes cumplen uno o dos de estos puntos; muchos menos cumplen los tres. La barrera más difícil de sortear es usualmente la herencia.

El concepto de programación orientada a objetos (OOP) no es nuevo, lenguajes clásicos como SmallTalk se basan en ella. Dado que la OOP. Se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que

está basado en objetos si soporta objetos como una característica fundamental del mismo.

El elemento fundamental de la OOP es, como su nombre lo indica, el objeto. Podemos definir un objeto como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo.

¿Qué son las Adicciones?

Son una conducta impulsiva e irresistible a ejecutar algo irracional o contrario a la voluntad de quien lo ejecuta. Es un padecimiento o síndrome, que presenta un cuadro muy florido de signos y síntomas característicos dependiendo de la sustancia adictiva. Existen adicciones tanto a sustancias químicas, vegetales, así como a actividades y hasta a ciertas relaciones interpersonales.

¿Cuáles son las causas de las Adicciones?

El origen de la adicción es multifactorial: Existen factores biológicos, genéticos, psicológicos y sociales. Los estudios demuestran que pueden existir cambios neuroquímicos en las personas adictas y que además, es posible que exista predisposición biogenética a desarrollar esta enfermedad. La naturaleza exacta de la adicción continúa siendo motivo de análisis.

III. ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - METODOS

Cada uno de estos componentes desempeña un papel totalmente independiente:

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

IV. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

PictureBox

Form

List box

Label

System

Con sus respectivas clases predefinidas en el lenguaje.

V. ALGORITMO

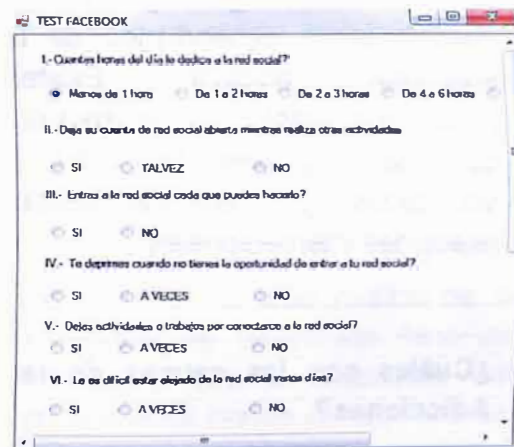


VI. PRUEBA DE CORRIDO

Menú de opciones



Ventana de cada tema



VII. CONCLUSIÓN

Al igual que TEST I en C++ esta versión esta mejorada ya que con C# se pudo implementar imágenes y una forma mas llamativa al de la consola de aplicaciones pero al igual que todo programa siempre habrá algo que mejorar

VIII. RECOMENDACIONES

Se recomienda aumentar videos para ayudar de una manera más grafica lo dañino de las adicciones ya mencionas en los test.

IX. BIBLIOGRAFÍA

<http://www.meta-adicciones.com/Test.php>

<http://es.finance.yahoo.com/blogs/fin-technologyayredeses/test-saberes-adicto-facebook-100552696.html>

http://www.entornomedico.org/enfermedadesdelaalaz/index.php?option=com_content&view=article&id=79:adicciones&catid=35:enfermedades&Itemid=56

TEST PARA SABER CUAL ES EL PESO IDEAL DE UNA PERSONA

Mariela Alisson Gutiérrez Callejas

mariela.gutierrez-70@hotmail.com

Universidad La Salle

RESUMEN

Con los conocimientos adquiridos en la materia de programación II (Programación Orientada a objetos) y la colaboración del docente: Lic. Walter Carvajal se realizó el presente trabajo que enfatiza en el tema de la nutrición ayudando a facilitar a una persona el modo de vida que lleva. Sabiendo cual es su peso adecuado.

I. INTRODUCCIÓN

El presente proyecto muestra como resultado el índice de masa corporal para ello se da a conocer en el informe una parte introductoria sobre el tema:

El índice de masa corporal, conocido también como BMI (body mass index) indica el estado nutricional de la persona considerando dos factores elementales: Su peso actual y su altura.

Este índice es el primer paso para conocer el estado nutricional de cualquier persona. Su cálculo arroja como resultado un valor que indica si la persona de la cual se habla se encuentra por debajo, dentro o

excedida del peso establecido como normal para su tamaño físico.

El índice de masa corporal establece la relación entre la masa y la talla de la persona. La ecuación para calcular el IMC es: masa corporal ("peso", expresada en kilogramos) dividida entre el cuadrado de la estatura (expresada en metros).

IMC de 18,5-24,9 se considera un peso saludable.

IMC de 25,0-29,9 se considera sobrepeso.

IMC de 30,0-39,9 se considera obesidad.

IMC de 40,0 o más se considera obesidad severa u obesidad mórbida).

II. MARCO TEÓRICO

La programación orientada a objetos es un paradigma que utiliza objetos como elementos fundamentales en la construcción de la solución. Surge en los años 70. Un objeto es una abstracción de algún hecho o ente del mundo real que tiene atributos que representan sus características o propiedades y métodos que representan su

comportamiento o acciones que realizan. Todas las propiedades y métodos comunes a los objetos se encapsulan o se agrupan en clases. Una clase es una plantilla o un prototipo para crear objetos, por eso se dice que los objetos son instancias de clases.

Hoy en día la tecnología orientada a objetos ya no se aplica solamente a los lenguajes de programación, además se viene aplicando en el análisis y diseño con mucho éxito, al igual que en las bases de datos. Es que para hacer una buena programación orientada a objetos hay que desarrollar todo el sistema aplicando esta tecnología, de ahí la importancia del análisis y el diseño orientado a objetos.

Los objetos, clases y estructuras tienen las propiedades siguientes:

Los objetos son instancias de un tipo de datos determinado. El tipo de datos proporciona un plano para el objeto que se crea, o del cual se crean instancias, cuando se ejecuta la aplicación.

Los nuevos tipos de datos se definen mediante clases y estructuras.

Las clases y estructuras conforman las unidades de creación de aplicaciones de C#, ya que contienen código y datos. Una aplicación de C# siempre contendrá al menos una clase.

Una estructura se puede considerar una clase ligera, idónea para crear

tipos de datos que almacenan pequeñas cantidades de datos, y no representa un tipo que después se podría extender por medio de la herencia.

Las clases de C# son compatibles con la herencia, lo que significa que se pueden derivar de una clase previamente definida.

III. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

PictureBox

Form

RichText

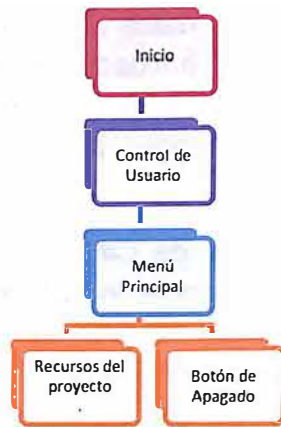
ComboBox

Label

System

Con sus respectivas clases predefinidas en el lenguaje.

IV. ALGORITMO



VI. CONCLUSIÓN

Al implementar el código y el diseño de nuestro proyecto relacionado con la nutrición observamos que se presentaron problemas

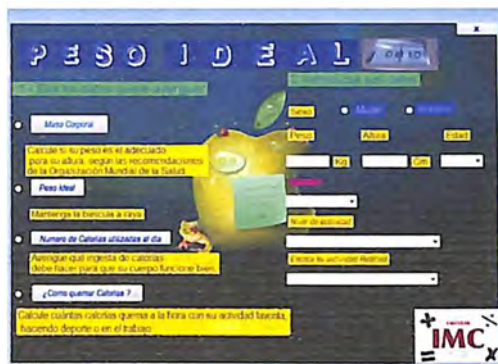
Al mandar datos de una ventana a otra además de buscar una correcta información en el estudio del tema ya que se quiso ampliar mas las aplicaciones que tiene el proyecto características especiales que nos permiten tener un diseño personalizado del programa.

Se pudo observar que el lenguaje Visual C# maneja una gran variedad de clases predeterminadas que facilitan en gran manera el manejo de los problemas que se fueron presentando a medida que realizamos el proyecto.

Se espera que este proyecto permita a muchas personas facilitar su modo de vida con la ayuda del programa diseñado que señala cual debe ser el peso adecuado de una persona para que esta no tenga consecuencias funestas posteriormente

V. PRUEBA DE CORRIDO

Menú de opciones



Ventana de cada tema

VII. RECOMENDACIONES

El presente programa se codifico en c ++ pero en esta ocacion trabajamos en c #, una forma que nos permite trabajar con diseños.

Se sugiere implementar videos especiales de como llevar una adecuada salud y ventanas de diseño que muestre tips según la OMS sobre dietas nutritivas adecuadas o incluso informes relacionados con el tema para personas con enfermedades diferentes

VIII. BIBLIOGRAFÍA

www.indicemasacorporal.org/

www.alimentacionsana.com.ar/informaciones/.../pesoideal.htm

[-http://msdn.microsoft.com](http://msdn.microsoft.com)

[-paginaweb2.com/post-introduccion-y-ventajas-de-la-programacion-o.](http://paginaweb2.com/post-introduccion-y-ventajas-de-la-programacion-o)

java.ciberaula.com/articulo/tecnologia_orientada_objetos/

NUMEROLOGÍA

Raúl Franz Villca Yampa

Franz_himura@hotmail.com

Universidad La Salle

RESUMEN

Con los conocimientos que he adquirido en la materia de programación II, y con el manejo del lenguaje C# y con la colaboración del docente Lic. Walter Carvajal, el presente trabajo se muestra el proyecto de numerología basado en los astros y los signos zodiacales.

I. INTRODUCCIÓN

La numerología sostiene, y prueba, que tus cualidades, tus defectos, tus sentimientos, tus inquietudes y tus vivencias, vienen determinadas por los muchos números que aparecen al hacer tu cuadro numerológico completo. Gracias a estos números y las múltiples combinaciones que ellos forman, podemos adentrarnos más y más en tu vida y en tu carácter, hasta tal punto que en tu estudio numerológico completo puedes reconocerte plenamente.

Se trata de una práctica que nos ayuda a hacer un análisis de personalidad increíblemente exacto que abarca no sólo el carácter, sino también la vocación, las posibilidades económicas, el amor y

las afinidades de la persona, y que además nos revela sus metas o finalidad de vida, sus posibilidades de destino y sus ciclos de vida.

Sin embargo, un estudio numerológico completo es algo complejo y difícil para el que necesitamos, además de la fecha de nacimiento, los nombres y la firma o nombre común o el apodo por el que nos conocen. Estos datos, analizados separada y conjuntamente, llegan a definirte con toda exactitud, y explican a la vez las enormes diferencias que pueden existir entre personas con un mismo número de nacimiento, e incluso gemelos nacidos con escasos minutos entre sí.

La numerología, significado e influencia de los números, es una de las ciencias ocultas que la humanidad ha cultivado desde el más lejano pasado. Siempre se creyó que los números tienen en sí mismos un principio activo. En su aspecto humano, el número es el símbolo que expresa la relación de nuestra vida y nuestra mente con la naturaleza. Es decir, nuestra existencia y nuestras posibilidades y

facultades dependen en cierto modo de ellos. Las vibraciones numéricas establecen así una relación existente entre los seres y el Universo.

La numerología nos permite conocer los mensajes de los números, no solo en su aspecto cuantitativo, apreciable a primera vista, si no también en sus aspectos cualitativos, percibiendo que detrás de cada cifra vibra una energía que nos trae una valiosa información psicológica y espiritual.

Los números son tal vez uno de los conceptos humanos mas perfectos y elevados. La numerología es la disciplina que estudia la vibración secreta de este maravilloso código y nos enseña a utilizarlo en nuestro beneficio

Del uno al nueve, cada número, por si mismo, encierra un significado en la naturaleza que es aplicable al ser humano, revelándonos misteriosa, pero certeramente, una esencia temperamental y caracterológica. No podemos negar la influencia de los números en nuestras casas, fechas de nacimiento y tampoco lo que son capaces de revelarnos, aplicándolos a nuestros nombres, que "casualmente" nos representarán inexorablemente.

Sin embargo, no debemos olvidar que la numerología es un estudio profundo, detallista y minucioso que se basa en un

UNIVERSIDAD LA SALLE 02/2012

sistema coordinado de muchos elementos y todos los números que integran nuestra vida, especialmente el de nuestro nombre y nuestra fecha de nacimiento. Por esto, aquí presentamos predicciones generales para cada número que seguramente te ayudarán a recapacitar sobre algunos puntos a tener en cuenta este año, pero será tu carta numerológica particular la única que te mostrará el camino con la eficacia deseada.

II. MARCO TEÓRICO

Culturas como los egipcios, los babilónicos y los griegos dieron mucha importancia a los números; a los cuales consideraban sagrados.

Una de las principales escuelas fue la pitagórica.

Pitágoras fue un famoso matemático que fundo su propia escuela en donde uno de los principales objetivos era comprender el universo a través de los números.

Ellos creían que el universo se basaba en el poder de los números y que este es un todo armónico cuyo funcionamiento se puede entender a través de los números.

Los pitagóricos vinculaban los números a la sabiduría y a la creación ya que las matemáticas pueden describir el universo físico y los principios de todos los seres..

Los números son símbolos sagrados que guardan profundos conocimientos sobre las leyes que rigen el universo, a través de la numerología.

III. ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - METODOS

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

IV. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

Picture box

Form

Rich text

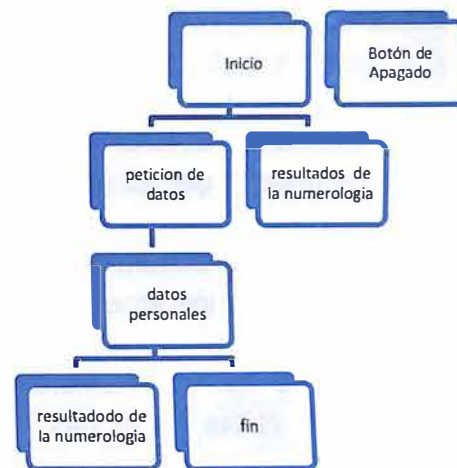
List box

Label

System

Con sus respectivas clases predefinidas en el lenguaje.

V. ALGORÍTMO



VI. PRUEBA DE CORRIDO

Sistema autenticación:



AVERIGUA TU SUERTE

PRIMER NOMBRE
SEGUNDO NOMBRE
APELLIDO PATERNO
APELLIDO MATERNO

FECHA DE NACIMIENTO 1 - ENERO - 2013 -

SEXO HOMBRE -

RESULTADO



VII. CONCLUSIÓN

El universo y sus partes son un número y cada ser es un resultado numérico, las fechas de nacimiento de una persona pueden indicar las características de la misma.

VIII. RECOMENDACIONES

Se sugiere averiguar tu suerte en los astros de la numerología.

IX. BIBLIOGRAFÍA

Numerología Sagrada, por Evelyn Murillo, Gómez Pérez Marco Antonio, "Pitágoras", en: Los grandes, Colombia, grupo editorial Tomo, 2002.

Aristóteles, Obras Selectas, edimat libros.

JUEGO DE LA VIBORITA

Iván Rodrigo Hilari Nina

Rodrigo065@hotmail.com

Universidad La Salle

RESUMEN

Con los conocimientos que he adquirido en la materia de programación II, y con el manejo del lenguaje C# y con la ayuda del docente Lic. Walter Carvajal, les presento una aplicación para pc el juego de la viborita

I. INTRODUCCIÓN

Tanelli Armanto, el ingeniero que realizó la adaptación en 1997 del viejo juego Snake para el Nokia 1100, habló con lanacion.com sobre cómo renació el título en 350 millones de móviles.

El juego creado por Alexey Pazhitnov contabiliza unas 118 millones de unidades vendidas gracias al Game Boy, que incluía el título junto a la consola portátil de Nintendo. Por su

Parte, la saga del simpático plomero italiano, emblema de la compañía

Japonesa de videojuegos, tiene en su haber 240 Millones de unidades vendidas en todo el mundo. Entonces, ¿es Súper Mario Bross el juego más popular del mundo? No, hay un título de muy bajo perfil, pero que fue precursor de las aplicaciones de entretenimiento en móviles: el Snake. Preinstalado de

UNIVERSIDAD LA SALLE 02/2012

fábrica y sin costo alguno, inició su presencia en los celulares de la compañía finlandesa en 1997 con su primera versión en el modelo 6110.

Las reglas del juego eran muy simples: cuatro teclas permitían guiar en un escenario a una hambrienta serpiente hacia su alimento, evitando tocar las paredes o su propio cuerpo, que crecía a medida que comía y se incrementaba el desafío del juego. Parte del impacto de Snake en la audiencia masiva fue gracias al Nokia 1100, un equipo de prestaciones básicas que vendió unas 200 millones de unidades entre 2003 y 2007.

II. MARCO TEÓRICO

Actualmente una de las áreas más candentes en la industria y en el ámbito académico es la orientación a objetos. La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código

que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases. Muchos lenguajes cumplen uno o dos de estos puntos; muchos menos cumplen los tres. La barrera más difícil de sortear es usualmente la herencia.

El concepto de programación orientada a objetos (OOP) no es nuevo, lenguajes clásicos como SmallTalk se basan en ella. Dado que la OOP se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

El elemento fundamental de la OOP es, como su nombre lo indica, el objeto. Podemos definir un objeto como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma

parte de una organización jerárquica o de otro tipo.

III. ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - METODOS

Cada uno de estos componentes desempeña un papel totalmente independiente:

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que

se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

IV. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

PictureBox

Form

RichText

Listbox

Label

System

V. ALGORITMO

INICIO

Comenzar

Salir

VI. PRUEBA DE CORRIDO



VII. CONCLUSIÓN

Observamos que se presentaron problemas principalmente de estudio del tema, ya que se quiso implementar varias características especiales que nos permiten tener un diseño personalizado del programa.

Se pudo observar que el lenguaje Visual C# maneja una gran variedad de clases predeterminadas que facilitan en gran manera el manejo de los problemas que se fueron presentando a medida que realizamos el proyecto.

En conclusión, este trabajo realizado durante este tiempo en esta materia nos sirve para mejorar el manejo rápido de los dedos.

VIII. BIBLIOGRAFÍA

<http://www.google.com/>

MIS TARJETAS

Jesus Angel Huayhua Monasterios

jack.tenreck@hotmail.com

Universidad La Salle

RESUMEN

Con el aprendizaje que he adquirido en la materia de programación II, y con lo aprendido del manejo del lenguaje C# y con la colaboración del docente (Lic. Walter Carvajal), se presenta un programa de aplicación de creación de tarjetas virtuales.

I. INTRODUCCIÓN

“Mis Tarjetas” es una aplicación que más que darle uso pretende mostrar cómo se realiza un proyecto de estas características. La aplicación hace de uso de conceptos de programación orientada a objetos con interfaz gráfica de usuario (GUI), manipulación de imágenes y componentes avanzados de las librerías de C#, además de conceptos de persistencia de datos (se utilizaron archivos físicos).

Mis Tarjetas” es una que gestiona tarjetas, estas tarjetas pueden ser de cumpleaños, de amistad, de navidad y de amor, con el programa podemos crear tarjetas y rediseñarla a nuestro gusto, cambiar el color del fondo, cambiar la imagen y cambiar la dedicatoria, además se puede agregar remitente

y agregar uno o varios destinatarios, posteriormente se puede almacenar la tarjeta en un archivo de tarjetas.

II. MARCO TEÓRICO

Actualmente una de las áreas más candentes en la industria y en el ámbito académico es la orientación a objetos. La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases. Muchos lenguajes cumplen uno o dos de estos puntos; muchos menos cumplen los tres. La barrera más difícil de sortear es usualmente la herencia.

El concepto de programación orientada a objetos (OOP) no es nuevo, lenguajes clásicos como SmallTalk se basan en ella. Dado que la OOP se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

El elemento fundamental de la OOP es, como su nombre lo indica, el objeto. Podemos definir un objeto como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo.

ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - METODOS

Cada uno de estos componentes desempeña un papel totalmente independiente:

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un

objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

III. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

PictureBox

Form

RichText

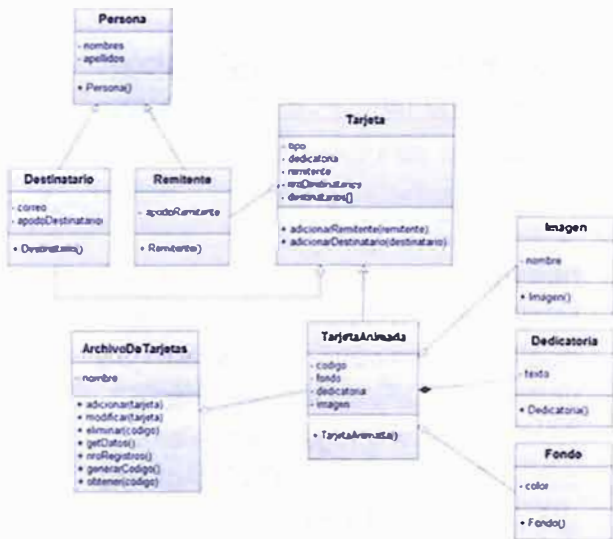
Listbox

Label

System

Con sus respectivas clases predefinidas en el lenguaje.

I. ALGORITMO

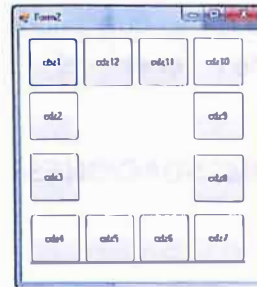


IV. PRUEBA DE CORRIDO

Sistema autenticación:



Menú de opciones:



Ventana de cada tema:



V. CONCLUSIÓN

El estudio del tema y el programa ya realizado se puso en marcha, ya que se quiso implementar varias características especiales que nos permiten tener un diseño personalizado del programa.

Se pudo observar que el lenguaje Visual C# maneja una gran variedad de clases predeterminadas que facilitan en gran manera el manejo de los problemas que se fueron presentando a medida que realizamos el proyecto.

Se espera que este proyecto permita a muchos programadores conocer más acerca del lenguaje C# su utilidad, y su funcionamiento, y

principalmente que el desarrollo de proyectos para uso personal, es una gran opción de personalización ya que uno decide, que puede, y que no puede tener el software diseñado.

VI. RECOMENDACIONES

Se recomienda mejorar, implementar imágenes y otros materiales didácticos para mejorar la aplicación.

VII. BIBLIOGRAFÍA

<http://www.google.com.bo/url?url=http://www.youtube.com/watch%3Fv%3DmzTx0vWrbG8&rct=j&sa=X&ei=aSWhUMvuDoS-8ASy6lGIDg&ved=0CB0QuAlwAA&q=como+hacer+una+tarjeta+virtual&>

[usg=AFQjCNFLHL7ldDBRWTSBk7xQUQ58gi1zIA&cad=rja](http://www.educacontic.es/blog/como-crear-una-tarjeta-de-presentacion-virtual?usg=AFQjCNFLHL7ldDBRWTSBk7xQUQ58gi1zIA&cad=rja)

<http://www.educacontic.es/blog/como-crear-una-tarjeta-de-presentacion-virtual>

<http://www.educacontic.es/blog/como-crear-una-tarjeta-de-presentacion-virtual>

http://www.google.com.bo/url?sa=t&rct=j&q=como+hacer+una+tarjeta+virtual&source=web&cd=22&cad=rja&ved=0CCQQFjABOBQ&url=http%3A%2F%2Fwww.regalosvirtuales.com%2F&ei=HCahULQu75DRAdbhgcgl&usg=AFQjCNEXwxmblvwPp_9KjGxbspDSeo8vdw

JUEGO DEL LABERINTO

Héctor Matías Arhuata

math_terrestre@hotmail.com

Universidad La Salle

RESUMEN

Para crear este proyecto se obtuvo la ayuda del Lic. Walter Carbajal y en base al materia de programación II y a todos los conocimientos adquiridos creamos este juego.

I. INTRODUCCIÓN

Los laberintos de forma cuadrada o rectangular son los más antiguos que existen; la primera representación conocida de un laberinto de este tipo se encuentra en una tablilla de Pilo y también la encontramos, como sello, en las tumbas del antiguo Egipto. Los laberintos de forma redonda o circular aparecieron a fines del siglo VII a.C. en la Italia etrusca; más tarde, los encontramos en las monedas de Cnosos, a finales del siglo III y se cree que eran usadas como mapa del célebre Laberinto de Creta.

Los laberintos se clasifican básicamente en dos grandes grupos "*según la relación que existe con el centro y la salida del mismo*". El primer grupo de estos laberintos es el laberinto clásico o laberinto univiarío:

es el que nos hace recorrer, al ingresar en él, todo el espacio para llegar al centro mediante una única vía, camino o sendero; es decir, no nos ofrece la posibilidad de tomar caminos alternativos, no hay bifurcaciones, sino que hay una sola puerta de salida, que es la misma por la que se entra al laberinto. Por el hecho de tener un solo camino o sendero que seguir a medida que avanzamos dentro de él, no nos podemos perder en su interior.

El segundo grupo de laberintos son los laberintos de mazes (perdederos, laberinto de caminos alternativos) en donde al recorrer el interior del laberinto seguiremos un camino correcto o uno incorrecto, que nos llevará o no a la salida del mismo. Los mazes se comenzaron a utilizar en los jardines de setos en la Inglaterra del siglo XII, ya que eran el lugar propicio para una cita amorosa; luego de allí se extendieron progresivamente por toda Europa, especialmente en Francia e Italia. Se

destacan en este sentido los jardines laberínticos de Andre le Notre en Versalles y el de Caboni en la Villa Pisani en Italia.

Por otro lado, cada uno de estos dos grandes grupos se dividen a su vez en subcategorías, atendiendo a "*la forma en que fue construido el laberinto*":

Laberinto clásico o cretense: Es un laberinto univariario de forma ovoidal y de diseño muy sencillo.

Laberinto romano: Laberinto univariario, que en un principio era de forma cuadrada, dividido en cuatro cuadrantes alrededor del centro; más tarde, estaba formado de círculos concéntricos, con la misma subdivisión de cuadrantes o zonas enmarcando el centro del laberinto.

Laberinto barroco: Es un laberinto del tipo maze que tiene varias "*vías muertas*" o "*camino sin salida*", además de poseer una sola vía correcta para salir de él.

Laberinto manierista: Laberinto con estructura arboréa, es decir, al final de un corredor encontraremos una bifurcación en Y.

Laberinto rizoma: Laberinto de ramificaciones infinitas.

Laberinto de Hampton Court.

Laberinto de Stolp.

Laberintos medievales: Son laberintos univariarios típicos de los usados en la decoración del suelo de las catedrales. Tienen un diseño complejo.

Laberinto de Boughton Green.

Laberinto de Altjessnitz.

Laberinto ruso (llamados "*Ciudad de Troya*").

Laberintos modernos: Aquel laberinto en donde todos los corredores que lo conforman se interconectan entre sí y no posee caminos o senderos de "*círculo cerrado*", es decir, aquel corredor que llega de nuevo al mismo punto de partida.

II. MARCO TEÓRICO

Laberinto de Creta

Artículo principal: *Laberinto de Creta*.

El laberinto debe su nombre a la legendaria construcción diseñada por el inventor Dédalo a pedido del rey Minos de Creta para mantener preso a su hijo Minotauro (monstruo mitad hombre, mitad toro), que acabó muerto por Teseo, quien se adentró en los inextricables pasillos dejando una huella de hilo (que le había dado la princesa Ariadna, hermana del monstruo).

Aunque no ha sido identificado positivamente ningún sitio en Creta como el laberinto del Minotauro, en

Cnosos se encontraron monedas del siglo III antes de Cristo con el símbolo del laberinto en ellas. El formato típico durante este período es un circuito de siete meandros o vías, conocido como el "laberinto clásico".

III. ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - METODOS

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a

UNIVERSIDAD LA SALLE 02/2012

disposición de sus descendientes a través de la herencia.

IV. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

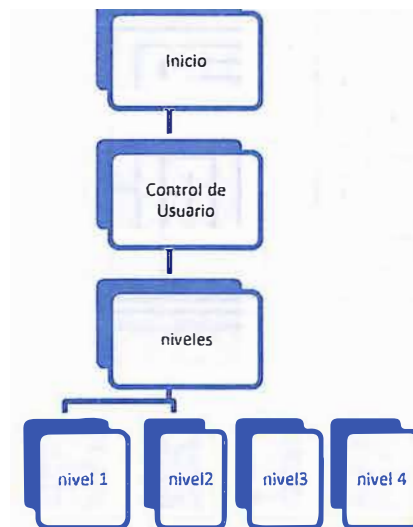
Form

Label

SystemCon

sus respectivas clases predefinidas en el lenguaje.

V. ALGORITMO



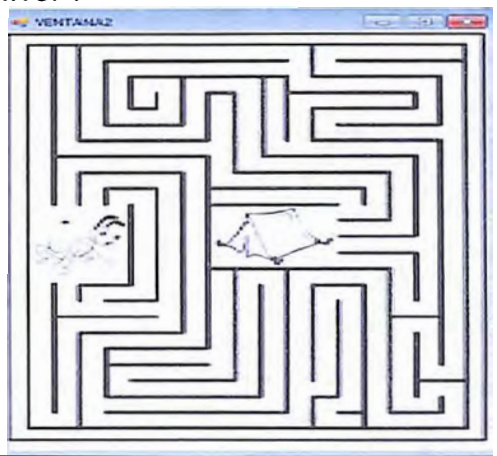
VI. PRUEBA DE CORRIDO

Sistema autenticación:

Niveles:



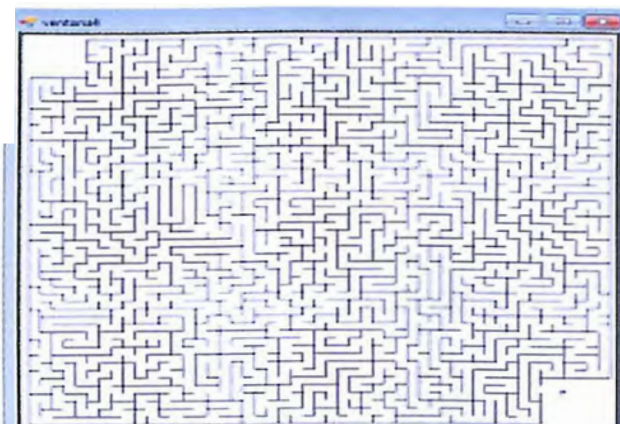
Nivel 1



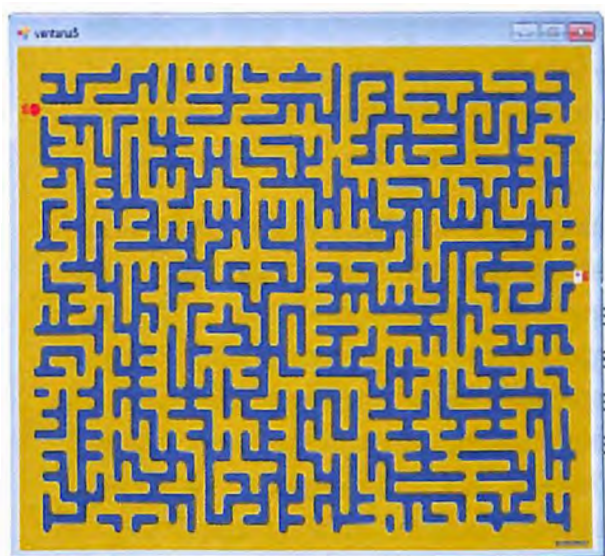
Nivel 2



Nivel 3



Nivel 4



VIII. RECOMENDACIONES

Se sugiere implementar nuevos y mejores niveles para este juego y en otros casos mejorarlo para ser un mejor proyecto.

IX. BIBLIOGRAFÍA

Wikimedia Commons alberga
contenido multimedia sobre
Laberinto.

Laberintos.net - Laberintos para jugar
gratis on-line (Español)

Fassier.fr/labyx/labys.html Philippe
Fassier, dibujante de laberintos
(francés)

Georgia Labyrinths (inglés)

The World-Wide Labyrinth Locator
(inglés)

Resources for Medieval Studies
(inglés)

Labyrinthos (inglés)

Laberinto virtual (español)

Exactly One Billion Mazes to solve
(inglés)

Laberintos en España - Spain
Labyrinths

NUMEROLOGIA DE LA VIDA

Jimmy Alejandro Ponce Paredes

jimale_16@hotmail.com

Universidad La Salle

RESUMEN

Con los conocimientos que he adquirido en la materia de programación 2 y la información, en el manejo del lenguaje C# y con la colaboración del docente Lic. Walter Carvajal, se presentara un programa que nos facilitará el aprendizaje del lenguaje Python

I. INTRODUCCION

“Numerología de la vida” en una aplicación que más que darle uso pretende mostrar cómo se realiza un proyecto de estas características. La aplicación hace de uso de conceptos aprendidos en el transcurso del semestre de programación orientada a objetos con interfaz gráfica de usuario (GUI), manipulación de imágenes y componentes avanzados de las librerías de C#, además de conceptos de persistencia de datos (se utilizaron archivos físicos).

“Numerología de la vida” es la lectura de la suerte con la fecha de nacimiento de una persona y esos sumar lo números y de ahí poder sacar por su fecha de nacimiento el numero que rige la su vida el numero

no puede ser mayor que 9 y menor que 0

II. MARCO TEORICO

Actualmente una de las áreas más candentes en la industria y en el ámbito académico es la orientación a objetos. La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento de l software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases múltiples. Muchos lenguajes cumplen uno o dos de estos puntos; muchos menos cumplen los tres. La barrera

más difícil de sortear es usualmente la herencia.

El concepto de programación orientada a objetos (OOP) no es nuevo, lenguajes clásicos como SmallTalk se basan en ella. Dado que la OOP se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

El elemento fundamental de la OOP es, como su nombre lo indica, el objeto. Podemos definir un objeto como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo.

ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - METODOS

Cada uno de estos componentes desempeña un papel totalmente independiente:

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

IV. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

Picture box

Form

Rich text

List box

Label

System

Con sus respectivas clases
predefinidas en el lenguaje.

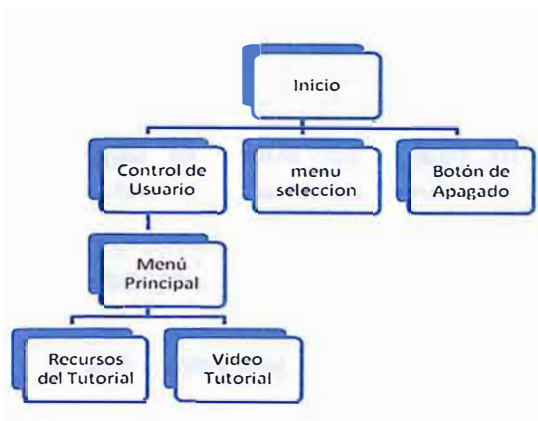
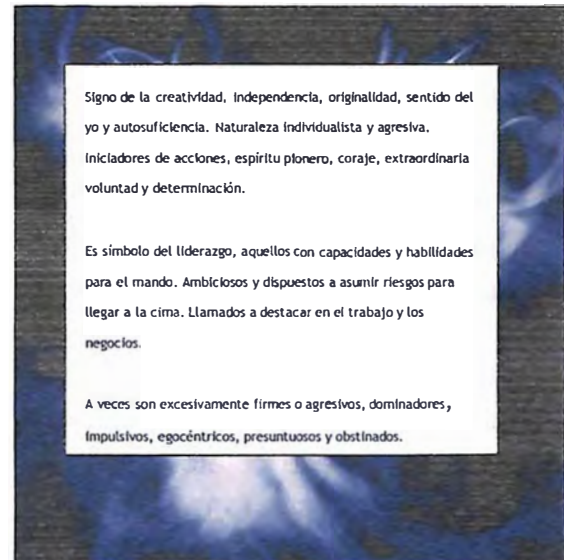


Escriba aquí la ecuación.

Menú de opciones



Ventana de cada tema



PRUEBA DE CORRIDO

Sistema autenticación:

UNIVERSIDAD LA SALLE 02/2012

CONCLUSION

Al implementar el código y el diseño de nuestro tutorial, observamos que se presentaron problemas principalmente de estudio del tema,

ya que se quiso implementar varias. Se pudo observar que el lenguaje Visual C# maneja una gran variedad de clases y objetos y herramientas que facilitan en gran manejo del programa y problemas que se fueron presentando a medida que realizamos el proyecto.

Se espera que este proyecto permita a muchos programadores conocer más acerca del lenguaje su utilidad, y su funcionamiento y diversión, ganar mayor conocimiento y principalmente que el desarrollo de proyectos, es una gran opción de personalización ya que uno decide, que puede y no puede implementar en sus proyectos, puede tener el software diseñado.

VI. RECOMENDACIONES

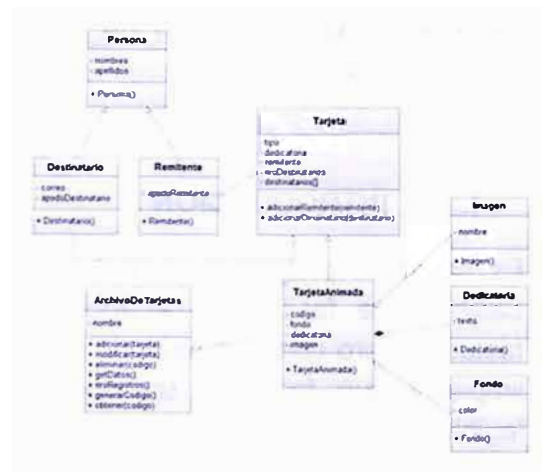
Se sugiere implementar video tutoriales y audio libros y otros y la ayuda de un tutor para el mayor

conocimiento en el manejo del lenguaje c materiales didácticos para mejorar el aprendizaje.

VII. BIBLIOGRAFIA

<http://www.programacion.com>

<http://www.lenguaje en c++.com>



FACTURACION DE UNA MINI EMPRESA

MIGUEL ANGEL JEMIO AZURDUY

Micky_j666@hotmail.com

Universidad La Salle

RESUMEN

Con los conocimientos que he adquirido en la materia de sistemas de información programación 2, y con el manejo del lenguaje C# y con la colaboración del docente Lic. Walter Carvajal, se presentara un programa que nos facilitará al sistematizar la facturación de una mini empresa.

I. INTRODUCCION

Es hacer un programa creando un sistema que facilite el llenado de facturas, pero con una sintaxis muy limpia y que favorece un código legible.

Se trata de no perder ningún tipo de dato, que este respaldado, a la maquina.

II. MARCO TEORICO

Actualmente una de las áreas más candentes en una mini empresa y en el ámbito académico es la orientación a objetos. La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento del software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo

UNIVERSIDAD LA SALLE 02/2012

de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases. Muchos lenguajes cumplen uno o dos de estos puntos; muchos menos cumplen los tres. La barrera más difícil de sortear es usualmente la herencia.

El concepto de programación orientada a objetos (OOP) no es nuevo, lenguajes clásicos como SmallTalk se basan en ella. Dado que la OOP se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

El elemento fundamental de la OOP es, como su nombre lo indica, el objeto. Podemos definir un objeto como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo.

III. ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - METODOS

Cada uno de estos componentes desempeña un papel totalmente independiente:

Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la herencia.

IV. CLASES Y OBJETOS

Estos objetos fueron utilizados:

Button

PictureBox

Form

Richtext

Listbox

Label

System

Con sus respectivas clases predefinidas en el lenguaje.

V. CONCLUSIÓN

Al implementar el código y el diseño de nuestro tutorial, observamos que se presentaron problemas principalmente de pérdida de datos, ya que se quiso implementar varias

características especiales que nos permiten tener un diseño personalizado del programa.

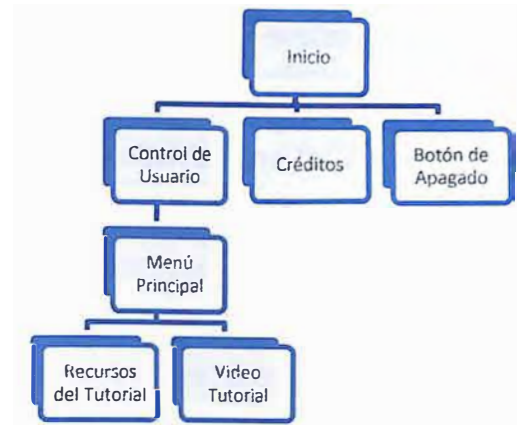
Se pudo observar que el lenguaje Visual C# maneja una gran variedad de clases predeterminadas que facilitan en gran manera el manejo de los problemas que se fueron presentando a medida que realizamos el proyecto.

Se espera que este proyecto permita a muchos programadores conocer más acerca de la sistematizar una facturación, su utilidad, y su funcionamiento, y principalmente que el desarrollo de proyectos para uso personal, es una gran opción de personalización ya que uno decide, que puede, y que no puede tener el software diseñado.

VI. RECOMENDACIONES

Se sugiere implementar video tutoriales y audio libros y otros materiales didácticos para mejorar el aprendizaje.

VII. ALGORÍTMO



VIII. BIBLIOGRAFÍA

Guía de aprendizaje de
Pythonsistematización

Release 2.0

Guido van Rossum

Fred L. Drake, Jr., editor

[Video tutorialescomosistematizar](#)

<http://www.youtube.com>

JUEGO DE MEMORIA

Cristian Fernando Mamani Maydana

lomejormio@hotmail.com

Universidad La Salle

RESUMEN

Con los conocimientos que he adquirido en la materia de programación II , y con el manejo del lenguaje C# he decidido realizar un juego de memoria en la que utilizare varios de mis conocimientos adquiridos en dicha materia y que me hará fácil el desarrollo del mismo.

I. INTRODUCCIÓN

Es un juego de cartas que requiere de gran habilidad mental o capacidad intelectual.

El número de cartas pueden variar debido a la dificultad, cada par de cartas (dos cartas) tienen una figura al igual que todas las demás cartas las cuales deben tener otra figura.

Para empezar el juego, se colocan diferentes cartas boca abajo sobre una mesa (no importa la superficie donde se coloquen las cartas, en este caso en la pantalla), luego se mezclara aleatoriamente para empezar con el juego , el jugador deberá poner boca arriba dos cartas al azar, si las dos cartas tienen la misma figura el jugador mantendrá esas dos cartas visibles las cuales

le ayudaran a acabar con el, pero si las dos cartas tienen diferentes figuras las cartas volverán a ponerse boca abajo y deberá levantar nuevamente dos cartas, El jugador acabara cuando este haya encontrado todos los pares de todas las cartas asignadas y resulte ganador del juego.

II. MARCO TEÓRICO

Este programa esta diseñado para su elaboración en el lenguaje de C# (C Sharp) del Visual C#.

En este articulo, se hará un juego de formar parejas en el que el jugador debe buscar las coincidencias entre pares de iconos ocultos.

Aprenderá a:

Usar un objeto **List** para contener objetos.

Usar un bucle **Foreach** en Visual C#

Realizar el seguimiento del estado de un formulario mediante variables de referencia.

Compilar un controlador de eventos que se pueda usar con varios objetos.

Hacer que un temporizador se active una vez después de iniciarse.

Paso 1: Crear un proyecto y agregar una tabla a un formulario

Para este proyecto es necesario iniciar una aplicación en **Windows Forms** una vez creado la aplicación se da el tamaño necesario que se va a utilizar y agregar un **TableLayoutPanel** esta tabla servirá como contenedor de las etiquetas (**Labels**) una vez editada con las filas y columnas necesarias en estos se debe arrastrar un **Label** y se edita sus propiedades cosa de que este **Label** pueda contener un icono utilizando una fuente de iconos (**Webdings**). Se debe seleccionar en la propiedad de texto la letra c para que el icono sea un cuadrado y sea mas fácil la contención de iconos mas adelante.

Nota

La fuente **Webdings** es una fuente de iconos que se distribuye con el sistema operativo Microsoft Windows.

Paso 2: Agregar un objeto aleatorio y una lista de iconos

En el editor de códigos es necesario primero agregar un objeto **Random** este maneja los mecanismos aleatorios del juego

Luego agregar un objeto **List** este objeto contendrá una lista de caracteres que gracias a la fuente **Webdings** estos adoptaran formas de iconos

Paso 3: Asignar un icono aleatorio a cada Label

Si el juego siempre oculta los mismos iconos en las mismas ubicaciones, no supone ningún reto. Debe asignar los iconos de forma aleatoria a los controles **Label** del formulario.

Para esto es necesario crear un método que sirva para asignar los iconos a los cuadrados aleatoriamente en este paso es necesario utilizar un bucle **Foreach** se usa siempre que se desea realizar la misma acción una y otra vez.

El método de asignar los iconos a los cuadrados recorre cada control **Label** del **TableLayoutPanel** y ejecuta las mismas instrucciones para cada uno de ellos. Esas instrucciones extraen un icono aleatorio de la lista que se agregó en **Paso 2: Agregar un objeto aleatorio y una lista de iconos**. (Para eso se necesita incluir dos iconos de cada uno en la lista, para que hubiera un par de iconos asignado a los controles **Label** aleatorios.)

Paso 4: Agregar un controlador de eventos Click a cada Label

Se inicia un evento de **Click** este creara un método que servirá para que cada vez que se haga click a un cuadro **Label** este muestre su icono oculto así cada vez que se vaya haciendo click ira mostrando todos sus iconos ocultos haciéndolos visibles (Es necesario utilizar la propiedad **ForeColor** para que cambie el color de la fuente a otro color se recomienda usar el negro).

Paso 5: Agregar referencias a los **Label**

El programa necesita realizar un seguimiento de los controles **Label** en los que el usuario hace click. Después de que se haga clic en el primer **Label**, el programa muestra el correspondiente icono. Después de que se haga click en el segundo **Label**, el programa muestra ambos iconos durante unos instantes y, a continuación, los vuelve a ocultar. El programa realiza un seguimiento del control **Label** en el que se hace click en primer lugar y del control en el que se hace clic en segundo lugar mediante variables de referencia.

Las variables de referencia parecen similares a las instrucciones usadas para agregar objetos (por ejemplo, objetos **Timer**, objetos **List** y objetos **Random**) a un formulario. Sin embargo, estas instrucciones no hacen aparecer dos controles **Label** adicionales en el formulario porque ninguna de las dos instrucciones incluye **new**. Sin **new**, no se crea ningún objeto. Por ello, **PrimerClick** y **SegundoClick** se denominan

variables de referencia: simplemente realizan un seguimiento o hacen referencia a objetos **Label**.

Cuando una variable no realiza el seguimiento de ningún objeto, se establece en un valor especial: **null** en Visual C#. Por lo tanto, cuando se inicia el programa, el valor de **PrimerClick** y el valor de **SegundoClick** están establecidos en **null**, lo que significa que las variables no realizan ningún tipo de seguimiento.

Paso 6: Agregar un temporizador

Del cuadro de herramientas se agrega un **timer**. El controlador de eventos **Tick** (del **timer**) realiza tres acciones: primero, detiene el temporizador mediante una llamada al método. A continuación, usa las dos variables de referencia, **PrimerClick** y **SegundoClick**, para recuperar los dos **Label** en las que el jugador hizo click y volver a ocultar sus iconos. Finalmente, restablece las variables de referencia **PrimerClick** y **SegundoClick** en **null**. Esto es importante, porque así es como se restablece el propio programa. Ahora no realiza el seguimiento de ningún control **Label** y vuelve a estar listo para el primer click del jugador.

Paso 7: Mantener visibles los pares

Agregue la instrucción **if** al método de control de eventos para que este reaccione cuando el texto (icono) sea igual haciendo una comparación entre los iconos del primero en hacer click y segundo en hacer click y los mantenga visibles.

Cuando dos iconos sean diferentes estos desaparecerán gracias al **timer** agregado mas antes reiniciando el proceso designado y realizándolo hasta que se cumpla la la condición del **if** de que los iconos sean iguales.

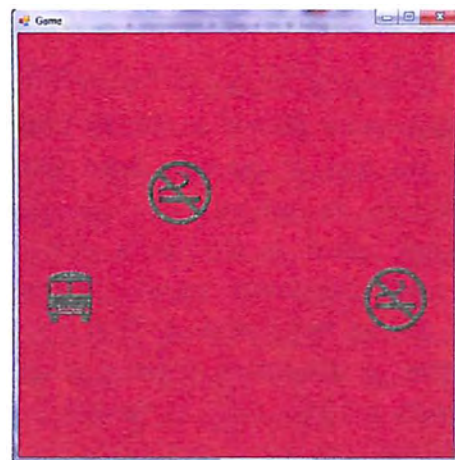
Paso 8: Finalizacion del juego

Una vez encontrado todos los pares en este aparecerá una ventana emergente que indicara que se han encontrado todos los pares y este se cerrara.

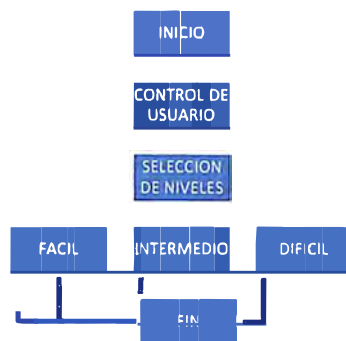
Ventana de Inicio del juego



Ventana de corrido del juego



III. ALGORITMO



IV PRUEBA DE CORRIDO

UNIVERSIDAD LA SALLE 02/2012

VI. CONCLUSIÓN

Al utilizar el código del Visual C#, observamos que se puede realizar grandes avances ya que este lenguaje es del todo practico y útil en varios aspectos que no todos los lenguajes de programación son capaces de llegar principalmente en el aspecto de diseño de programas.

Espero que este Mini-tutorial sea de ayuda para todos aquellos que quieran realizar algo parecido y que

sean capaces de mejorar a lo que
quieran y necesiten

VII. RECOMENDACIONES

Se recomienda examinar todo tipo
de video tutoriales acerca del
lenguaje C#, y su uso en el Visual
C# para mejorar su utilidad de este
software

BIBLIOGRAFÍA

[http://msdn.microsoft.com/es-
es/library/dd553235.aspx](http://msdn.microsoft.com/es-es/library/dd553235.aspx)

Tutorial 4: Crear un juego de formar
parejas

Video tutoriales C#
Sharp) <http://www.youtube.com>

MI CALENDARIO

AYRTON QUENTA HUAYCHO

dreicom666@hotmail.com

UNIVERSIDAD LA SALLE

RESUMEN

Resumiendo que es el calendario en las diferentes librerías el calendario tiene una simple vista pero tenemos que mejorarla verdad y cambiarla a un reloj por que no tiene esa función y también con la ayuda del Ing. Walter Carvajal.

I. INTRODUCCION

El programa tiene la aplicación de la personalización del calendario para una mejor vista y agrada a las personas lo que ven un cambio de calendario.

II.MARCOTEORICO

Actualmente una de las áreas más candentes en la industria y en el ámbito académico es la orientación a objetos. La orientación a objetos promete mejoras de amplio alcance en la forma de diseño, desarrollo y mantenimiento de

software ofreciendo una solución a largo plazo a los problemas y preocupaciones que han existido desde el comienzo en el desarrollo

de software: la falta de portabilidad del código y reusabilidad, código que es difícil de modificar, ciclos de desarrollo largos y técnicas de codificación no intuitivas.

Un lenguaje orientado a objetos ataca estos problemas. Tiene tres características básicas: debe estar basado en objetos, basado en clases y capaz de tener herencia de clases. Muchos lenguajes cumplen uno o dos de estos puntos; muchos menos cumplen los tres. La barrera más difícil de sortear es usualmente la herencia.

El concepto de programación orientada a objetos (OOP) no es nuevo, lenguajes clásicos como SmallTalk se basan en ella. Dado que la OOP se basa en la idea natural de la existencia de un mundo lleno de objetos y que la resolución del problema se realiza en términos de objetos, un lenguaje se dice que está basado en objetos si soporta objetos como una característica fundamental del mismo.

El elemento fundamental de la OOP es, como su nombre lo indica, el objeto. Podemos definir un objeto

UNIVERSIDAD LA SALLE 02/2012

como un conjunto complejo de datos y programas que poseen estructura y forman parte de una organización.

Esta definición especifica varias propiedades importantes de los objetos. En primer lugar, un objeto no es un dato simple, sino que contiene en su interior cierto número de componentes bien estructurados. En segundo lugar, cada objeto no es un ente aislado, sino que forma parte de una organización jerárquica o de otro tipo.

ESTRUCTURA DE UN OBJETO

Un objeto puede considerarse como una especie de cápsula dividida en tres partes:

1 - RELACIONES

2 - PROPIEDADES

3 - MÉTODOS

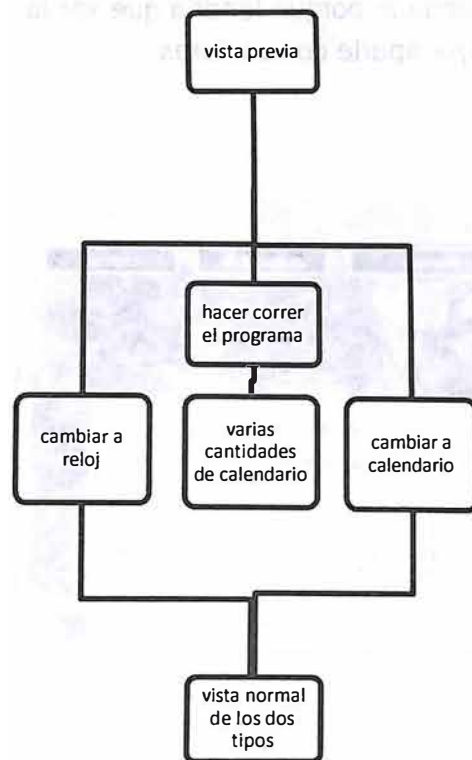
Las relaciones permiten que el objeto se inserte en la organización y están formadas esencialmente por punteros a otros objetos.

Las propiedades distinguen un objeto determinado de los restantes que forman parte de la misma organización y tiene valores que dependen de la propiedad de que se trate. Las propiedades de un objeto pueden ser heredadas a sus descendientes en la organización.

Los métodos son las operaciones que pueden realizarse sobre el objeto, que

normalmente estarán incorporados en forma de programas (código) que el objeto es capaz de ejecutar y que también pone a disposición de sus descendientes a través de la programación con los diferentes órdenes.

III. ALGORITMO



IV. PRUEBA DE CORRIDO

Sistema autenticación: que funcione las opciones y los botones

de

control.

VII. BIBLIOGRAFIA

http://www.youtube.com/watch?v-q40HDBre_F4

V. CONCLUSIÓN

ayuda a la persona que quiere ver su calendario de un mes a seis meses si usted lo desea.

<http://www.youtube.com/watch?v-T4juqPDa5U0&feature=related>

VI. RECOMENDACIONES

no varía de imágenes en el calendario porque tendría que ser la mayor aparte con vectores.

por algunas mejoras de la presentación.

