Unidad I

Fundamentos del lenguaje

1.1. Programación orientada a eventos.

Los lenguajes visuales orientada al evento y con manejo de componentes dan al usuario que no cuenta con mucha experiencia en desarrollo, la posibilidad de construir sus propias aplicaciones utilizando interfaces gráficas sobre la base de ocurrencia de eventos.

Para soportar este tipo de desarrollo interactuan dos tipos de herramientas, una que permite realizar diseños gráficos y , un lenguaje de alto nivel que permite codificar los eventos. Con dichas herramientas es posible desarrollar cualquier tipo de aplicaciones basadas en el entorno.

Visual Basic es uno de los lenguajes de programación que más entusiasmo despiertan entre los programadores de computadoras, tanto expertos como novatos. En el caso de los programadores expertos por la facilidad con la que desarrollan aplicaciones complejas en poquísimo tiempo (comparado con lo que cuesta programar en ***Visual C++***, por ejemplo). En el caso de los programadores novatos por el hecho de ver de lo que son capaces a los pocos minutos de empezar su aprendizaje. El precio que hay que pagar por utilizar ***Visual Basic*** es una menor velocidad o eficiencia en las aplicaciones.

Visual Basic es un lenguaje de programación visual, también llamado lenguaje de 4ta. generación. Esto quiere decir que un gran número de tareas se realizan sin escribir código, simplemente con operaciones gráficas realizadas con el ratón sobre la pantalla.

Visual Basic es también un programa *basado en objetos*, aunque no *orientado a objetos* como ***Visual C++***. La diferencia está en que ***Visual Basic*** utiliza ***objetos*** con ***propiedades*** y ***métodos***, pero carece de los mecanismos de ***herencia*** y ***polimorfismo*** propios de los verdaderos lenguajes orientados a objetos como ***Java*** y *C++.*

En este trabajo se presentará las características generales de ***Visual Basic***, junto con algunos ejemplos sencillos que den idea de la potencia del lenguaje orientado a eventos, y del modo en que se utiliza.

1.2. Objetos, controles y componentes.

Un objeto es todo lo que se puede colocar en la mesa de trabajo. Una vez agregado un elemento visual (por ejemplo, una imagen, archivo multimedia, control u objeto 3D) a la mesa de trabajo, éste se convierte en un objeto del prototipo. Los objetos funcionan del mismo modo en los prototipos SketchFlow que en otros proyectos de Microsoft Expression Blend.

Trabajar con objetos

Existen varias formas de agregar objetos a un prototipo. Puede dibujar objetos directamente en la mesa de trabajo, agregar objetos desde el panel **Herramientas** o insertar objetos que haya importado a Expression Blend. Una vez que haya agregado objetos a la mesa de trabajo, puede modificarlos u organizarlos de la forma que le resulte más cómoda.

Para obtener más información, vea Dibujar objetos.
Trabajar con controles
Los controles (o elementos de diseño de la interfaz de usuario) son los componentes visibles de un prototipo. Puede agregar controles directamente a la mesa de trabajo, incluidos controles personalizados. También puede convertir objetos existentes en controles.
Puede crear un UserControl a partir de un objeto o grupo de objetos en la mesa de trabajo y, a continuación, agregarlo como un nodo componente al flujo de la aplicación. Esto resulta especialmente útil si tiene un UserControl (por ejemplo, una barra de navegación) que desea usar en varias pantallas diferentes.

1.2. Tecnología .NET.

La tecnología .Net en general ha revolucionado el mundo del desarrollo de software cumpliendo con su premisa de conectar personas, dispositivos y sistemas sobre internet. La respuesta de .Net a las necesidades del mercado son prácticamente en tiempo real, las empresas y los profesionales deben estar al día con los avances y nuevas técnicas que introduce esta plataforma. Pero, ¿por qué tantas versiones del Framework? ¿Por qué todo va tan rápido? hmmm prefiero quedarme con lo que tengo....

Desde el principio .Net ha sido la apuesta de Microsoft por una plataforma de desarrollo que permitiera sacar el máximo provecho de los protocolos abiertos (W3C, HTTP, SOAP, etc.), la conectividad que da internet y la integración de XML. Con todo esto en mente fue desarrollada la primera versión del .Net Framework en el año 2002, la cual marcó el inicio de una forma de trabajo mucho más flexible que soportaba diferentes lenguajes de programación y un aumento significativo en la productividad gracias al nuevo Ambiente de Desarrollo Integrado, Visual Studio.

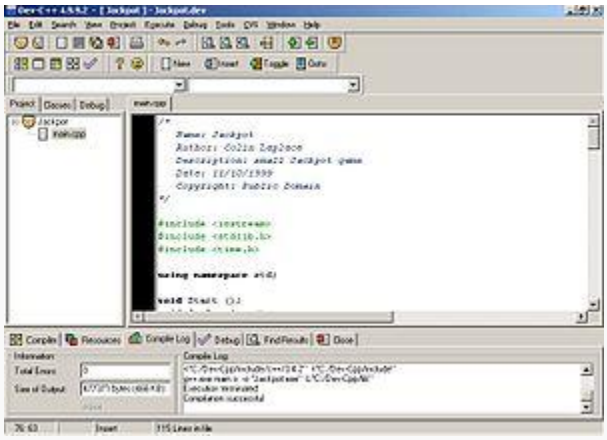
Con las primeras versiones del .Net Framework (1.0 y 1.1) pudimos hacer los primeros desarrollos y pruebas de la tecnología, surgieron mejoras y nuevas necesidades que fueron cubiertas en la Versión 2.0. En esta versión encontramos herramientas completas que nos permiten realizar tareas comunes de forma rápida, un ejemplo de estas herramientas que viene en forma de Provider es el MembershipProvider, un sistema completo de seguridad para aplicaciones ASP.Net, totalmente configurable y parametrizable con un modelo de desarrollo fácil y con controles de servidor disponibles. Esta versión marcó la base de lo que seria la evolución en capas del .Net Framework.

Entre la versión 1.0 y 2.0 del .Net Framework fuimos testigos de cambios significativos en la estructura principal de la tecnología, aquí vimos cómo se adaptó mejor la sintaxis este cambio lo sufrimos todos sobre todo a la hora de intentar migrar aplicaciones desarrolladas sobre la versión 1.0 y abrirlas con Visual

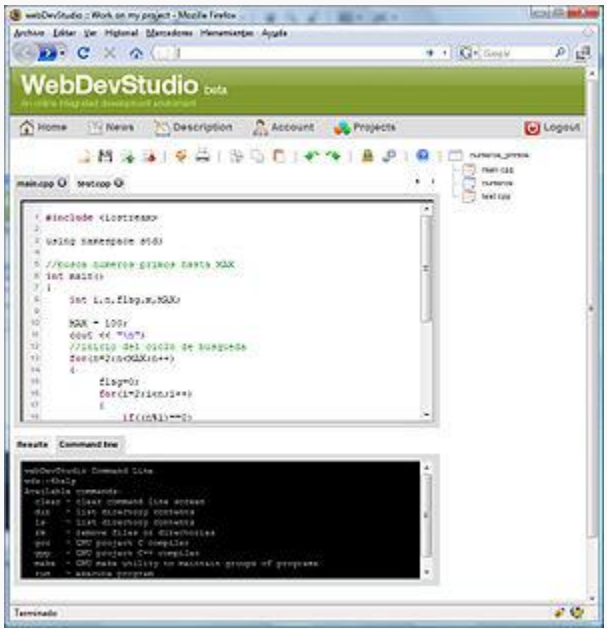
Studio 2005 (¿recuerdan el asistente de migración? ¡Qué horror! , nos cambiaba todo). Esta experiencia hizo que muchos de nosotros no quisiésemos utilizar nuevas versiones del .Net Framework.

1.3. Entorno integrado de desarrollo.

Un **entorno de desarrollo integrado**, llamado también **IDE** (sigla en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas deprogramación. Puede dedicarse en exclusiva a un solo lenguaje de programación o bien puede utilizarse para varios.



Dev C++, un entorno para el lenguaje de programaciónC++.



WebDevStudio, un IDE en línea para el lenguaje de programación C/C++.

Un IDE es un entorno de programación que ha sido empaquetado como un programa de aplicación; es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, PHP, Python, Java,C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk u Objective-C.

1.4. Tipos de proyectos.

Proyectos sociales

Son proyectos para lograr alguna obra que beneficie a la comunidad, pueden ser:
Con pequeña subvención. El apoyo económico es poco y proviene del mismo equipo de investigación, es manejado por la comunidad que aprenden unos de otros el manejo de grupo, la ejecución y supervisión de proyectos, se reúnen para establecer reglas, ellos mismos administran sus fondos. Los funcionarios ayudan a la comunidad en su proyecto.
Proyectos apoyados por pequeñas subvenciones. Las ayudas económicas provienen del equipo de investigación y el gobierno. Los funcionarios del gobierno intervienen para que las actividades se cumplan y a si vez opinan sobre cómo se debe administrar el proyecto. Se emplean algunos elementos del proyecto central.
Proyectos apoyados exclusivamente por el gobierno. El apoyo económico solo proviene del estado pero incluye algunos elementos del proyecto central. La comunidad igualmente presta ayuda en la ejecución de las actividades.

Proyectos de investigación

Tiene relaciones con la teoría existente en el tema y a su vez con el mundo empírico, de esta forma se planea lo que se pretende investigar. Sus partes son: planteamiento o formulación del problema, antecedentes, importancia o justificación del estudio, elementos teóricos que fundamenten la investigación,

objetivos (generales y específicos), metodología, esquema o plan de trabajo, cronograma y referencias.

Proyectos de inversión

Están relacionadas con la empresa y la parte comercial los hay de varias clases:
Inversión privada: consiste en crear un plan que permita obtener una rentabilidad económica a partir de la inversión de un capital.
Inversión pública: El estado invierte recursos para lograr el bienestar social de una comunidad a la vez que beneficio económico.
Inversión social: Se busca invertir bienes en el desarrollo exclusivamente social sin esperar remuneración económica, sino que los beneficios permanezcan después de acabado el proyecto.

Proyectos de infraestructura

Se invierte en obras civiles, se construye infraestructura que aporte beneficios económicos o sociales.

Proyectos sociales

Su único fin es mejorar la calidad de vida de una comunidad en sus necesidades básicas como salud, educación, empleo y vivienda. El proyecto pronostica y orienta una serie de actividades para conseguir unos determinados objetivos. Debe contener una descripción de lo que quiere conseguir, debe ser adaptado al entorno en que se piensa desarrollar, los recursos necesarios para desarrollarlo y el cronograma en el que se establece el plazo de su ejecución.

Proyectos de desarrollo sostenible

Es un proyecto social y económico de una comunidad que incluye ecología o del medio ambiente como un elemento importante tanto para mejorar la economía como para ser protegido durante un largo periodo. Este tipo de proyectos surgió en torno al deterioro en el medio ambiente y la intención de que la producción humana no lo impacte de forma negativa. También busca la participación equitativa de la sociedad en estos procesos.

1.5. Espacios de nombres.

En programación, un **espacio de nombres** (del inglés *namespace*), en su acepción más simple, es un conjunto de nombres en el cual todos los nombres son únicos.

Un espacio de nombres es un contenedor abstracto en el que un grupo de uno o más identificadores únicos pueden existir. Un identificador definido en un espacio de nombres está asociado con ese espacio de nombres. El mismo identificador puede independientemente ser definido en múltiples espacios de nombres, eso es, el sentido asociado con un identificador definido en un espacio de nombres es independiente del mismo identificador declarado en otro espacio de nombres. Los lenguajes que manejan espacio de nombres especifican las reglas que determinan a qué espacio de nombres pertenece una instancia de un identificador.

Por ejemplo, Pedro trabaja para la compañía X y su número de empleado es 123. María trabaja para la compañía Y y su número de empleada también es 123. La razón por la cual Pedro y María pueden ser identificados con el mismo número de empleado es porque trabajan para compañías diferentes. Diferentes compañías simbolizan en este caso diferentes espacios de nombres.

En programas grandes o en documentos no es infrecuente tener cientos o miles de identificadores. Los espacios de nombres (o técnicas similares como la emulación de espacios de nombres) disponen de un mecanismo para ocultar los identificadores locales. Proporcionan los medios para agrupar lógicamente los identificadores relacionados en sus correspondientes espacios de nombres, haciendo así el sistema más modular.

Muchos lenguajes de programación manejan espacios de nombres. En algunos lenguajes, como C++, PHP o Python, estos identificadores nombrando espacios de nombres están asociados con un espacio de nombres que los agrupa. Así pues, en estos lenguajes, los espacios de nombres se pueden anidar formando un árbol de espacios de nombres. En la raíz de este árbol se encuentra el espacio de nombres anónimo global.