



UNIVERSIDAD POLITÉCNICA
DE TECÁMAC

TAREA 1

ANÁLISIS DE LA
PROGRAMACIÓN
VISUAL



Alumnos:

1. González
Clímaco Bruno
2. Joaquín Espejel
Jesús Elías
3. Juárez Galicia
Karim Izahi
4. Gámez Ledo
Irving Jonathan

Matriculas:

1. 1321124272
2. 1321124258
3. 1321124269
4. 1321124302

Grupo:

4322IS

Carrera:

Ingeniería en
Software

ANALISIS DE LA PROGRAMACIÓN VISUAL

- **Programación Orientada a Objetos:**

La Programación Orientada a Objetos (POO) es un paradigma de programación, es decir, un modelo o un estilo de programación que nos da unas guías sobre cómo trabajar con él. Se basa en el concepto de clases y objetos. Este tipo de programación se utiliza para estructurar un programa de software en piezas simples y reutilizables de planos de código (clases) para crear instancias individuales de objetos.

- **Características y aplicaciones de eventos.**

- Características

- Dependencia de eventos
- Orientada al servicio
- Eventos
- Controlador de eventos
- Funciones de activación
- Tiempo controlado

- Aplicaciones

Ya que la programación orientada a eventos resulta ser más un enfoque que un tipo de lenguaje, las aplicaciones que se basan en eventos pueden crearse en cualquier lenguaje de programación.

La programación orientada a eventos está transformando el mundo de los servicios de desarrollo de software y se está convirtiendo en parte integral del desarrollo actual. Su difusión fue estimulada por Windows y la difusión de entornos visuales.

- **Características de componentes y métodos visuales y no visuales.**

- Aplicaciones

Un componente de software es un elemento de un sistema de prestación de servicios predefinido que puede comunicarse con otros componentes y es un objeto escrito de acuerdo con las especificaciones. No importa cuál sea esa especificación, siempre que el objeto cumpla con la especificación.

- Visuales y No Visuales

Un componente es visual cuando tiene una representación gráfica en el diseño y el tiempo de ejecución (botones, controles deslizantes, cuadros de edición, etc.) y se dice que es no visual en caso contrario (temporizadores, diálogo que no aparece durante el diseño, etc.). Aparte de eso, no hay otras diferencias entre ellos, excepto, por supuesto, la que se deriva de la visualización del componente.

- **Procesos de desarrollo visual en proyectos distribuidos y de escritorio.**

El desarrollo de software moderno como herramienta de programación visual es una solución fácil de usar para expertos que no están especializados en codificación. El diseño visual simple de imágenes y bloques hace que sea fácil

de entender, diseñar e interpretar para el usuario promedio. Entonces, en lugar de ver líneas de código difíciles de descifrar, las personas pueden entender y explicar lógicamente conceptos complejos utilizando este lenguaje de programación visual. Debido a su relativa simplicidad, la programación visual es una forma ideal de familiarizar a los usuarios con la programación y la programación. Sin embargo, a pesar de la simplicidad del lenguaje, puede resultar engorroso porque tiene muchos gráficos. Este lenguaje de programación es más grande y por lo tanto ocupa más espacio en la computadora, lo que puede ralentizar el dispositivo debido a la cantidad de memoria que necesita en el disco. Los VPL son muy limitados en la gama de características expresivas que utilizan. Esto dificulta la realización de operaciones más complejas, por lo que los gigantes tecnológicos del mundo de la programación rara vez utilizan este tipo de lenguaje.

- **Requerimientos visuales de proyectos distribuidos y de escritorio.**

- Definir los objetivos del proyecto
- Seleccione el resultado
- Definir riesgos y límites
- Usar ayudas visuales para mejorar la estrategia del proyecto
- Calculo de presupuesto
- Hacer un plan de contingencia
- Documento de puntos de referencia

- **Herramientas y lenguajes de programación visual.**

- Python:

Es un lenguaje de programación orientado a objetos dinámicos y de alto nivel que se usa comúnmente para desarrollar aplicaciones rápidamente.

- C++:

Un lenguaje de programación y un entorno de desarrollo potente y flexible para crear aplicaciones para Windows, Linux, iOS y Android.

- JavaScript:

El editor de JavaScript en Visual Studio es compatible con EcmaScript 6 y cuenta con el motor IntelliSense más avanzado del mercado.

JavaScript es un lenguaje de primera clase en Visual Studio. Puede usar la mayoría de las herramientas de edición estándar (extractos, IntelliSense, etc.) al escribir código JavaScript en el IDE de Visual Studio.

- TypeScript:

Es un conjunto completo de JavaScript que se compila en JavaScript sin procesar y le permite crear un código más extensible.

- C#:

Es un marco multiplataforma gratuito, de código abierto y de alto rendimiento para crear aplicaciones modernas, basadas en la nube y conectadas a Internet. Con ASP.NET Core, puede crear aplicaciones y servicios web, aplicaciones IoT y servicios back-end móviles.

- Razor:

La arquitectura Razor de ASP.NET permite incluir C# directamente en sus vistas. Además, tiene acceso completo a los patrones y plantillas existentes de .NET Framework.