

ANALISIS DE LA PROGRAMACIÓN VISUAL

(PROGRAMACIÓN VISUAL)

GRUPO: 4322-IS

INGENIERIA EN SOFTWARE

NOMBRE DEL PROFESOR: EMMANUEL TORRES SERVIN

ALUMN@S: MARQUEZ RODRIGUEZ JOCELYN

EDRIC YAMIL SANCHEZ AVILA

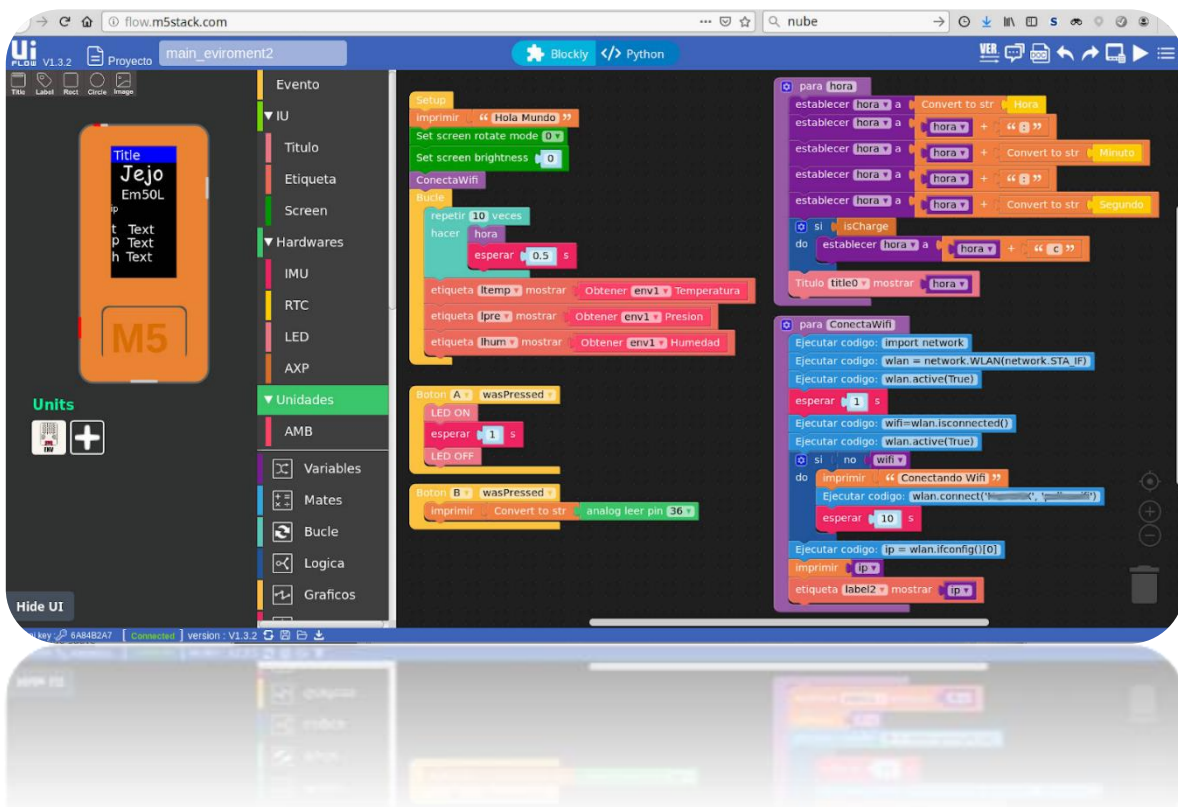
JONATHAN GAMEZ LEDO

INTRODUCCIÓN

En este proyecto se darán a conocer unos cuantos conceptos importantes que nos ayudan sobre este tema y podremos darnos cuenta que gran parte de los lenguajes de programación visual permiten a los usuarios crear programas mediante la manipulación de elementos gráficos, en lugar de especificarlos exclusivamente de manera textual, en donde se pueden utilizar componentes gráficos como, iconos, botones, y símbolos de codificación.

De la misma manera se notará que la programación visual es una forma perfecta de introducir a los usuarios en la codificación y la programación.

Ya que el desarrollo de software moderno como herramienta de programación visual es una solución amigable para los usuarios que no son expertos en codificación.



Conceptos de programación orientada a objetos.

Uno de los casos es el tema de la programación orientada a objetos, la cual se basa en la creación de un modelo del problema de destino en sus programas.

La programación orientada a objetos disminuye los errores y promueve la reutilización del código.

Bien se dice que 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.

La programación dirigida por eventos, es un paradigma de programación en el que tanto la estructura como la ejecución de los programas van determinados por los sucesos que ocurran en el sistema, definidos por el usuario o que ellos mismos provoquen.

De la misma manera es la base de lo que llamamos interfaz de usuario, aunque puede emplearse también para desarrollar interfaces entre componentes de Software o módulos del núcleo.

Dependencia de eventos

El flujo del programa viene dado por eventos que pueden ser acciones del usuario, mensajes de otros programas, etc., separando la lógica de procesamiento de los eventos del resto del código de un programa, contrastando así con el procesamiento por lotes.

Orientada al servicio

Se utiliza para escribir programas diseñados para el servicio sin ralentizar la computadora, ya que la orientación al servicio solo consume poco poder de procesamiento.

Eventos

Es una condición que surge durante la ejecución de un programa y que requiere alguna acción por parte del sistema.

Controlador de eventos

Es una determinada unidad del programa que se activa para reaccionar ante un evento.

Es decir, es un tipo de función o método que ejecuta una acción específica cuando se activa un evento determinado.

Funciones de activación

Son funciones que deciden qué código ejecutar cuando se produce un evento específico.

Se utilizan para seleccionar qué controlador de eventos emplear al producirse un evento.

Tiempo controlado

Es un código específico que se ejecuta en un momento determinado.

Características de componentes y métodos visuales y no visuales.

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (botones, barras de scroll, cuadros de edición, etc.), y se dice no visual en caso contrario (temporizadores, cuadros de diálogo no visibles en la fase de diseño, etc).

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

Un proceso de desarrollo de software es la descripción de una secuencia de actividades que deben ser seguida por un equipo de trabajadores para generar un conjunto coherente de productos, uno de los cuales es el programa del sistema deseado.

Específicamente un proceso de desarrollo debe conseguir:

- Proporcionar una plantilla de desarrollo del proyecto en el que quede definido lo que cada trabajador que interviene debe realizar y los productos que debe generar a lo largo de él.
- Mejorar la calidad del producto que se genera en función de: Disminuir el número de defectos que se producen y que deben ser corregidos.
- Disminuir la severidad de los defectos residuales que al final pueden permanecer en el producto final.
- Mejorar la reusabilidad, de forma que gran parte del trabajo que se realiza pueda ser reutilizado en próximos proyectos.
- Mejorar la estabilidad del proceso de forma que se minimicen las reelaboraciones del producto.
- Generar un producto que sea de fácil mantenimiento posterior.
- Mejorar la predictibilidad del proyecto en función de: La cantidad de esfuerzo humano y de recursos que requiera.
- Disminuir los plazos de desarrollo y llegada al mercado.
- Generar a lo largo del desarrollo de la información adecuada y diferenciada para que los diferentes responsables del proyecto puedan hacer su seguimiento de forma efectiva.

Herramientas y lenguajes de programación visual.

En la programación visual, los elementos del lenguaje de programación están disponibles en forma de bloques diseñados de manera gráfica, por lo que también se la llama programación gráfica.

Los cuales se utilizan como base:

- Python. Este es un lenguaje de programación dinámico y de alto nivel orientado a objetos, que a menudo se usa para el desarrollo rápido de aplicaciones.
- Visual Basic. Este lenguaje de programación orientado a objetos es moderno, muy fácil de aprender y está dirigido a la plataforma.
- C++
- F#

CONCLUSIÓN

Como conclusión nos podemos dar cuenta que el entender la programación visual nos ayuda a mejorar nuestra capacidad de comprender fácilmente la estructura de lenguajes de programación.

REFERENCIAS:

Sáez López, J. M., & Cózar Gutiérrez, R. (2017). Programación visual por bloques en Educación Primaria: Aprendiendo y creando contenidos en Ciencias Sociales. *Revista Complutense de Educación*, 28(2), 409-426.

Duilio, M. C. E., Jorge, A. N. C., Walter, C. C., & Adan, N. Z. W. Programación li-CC67-201601.

Eduardo, B. G. W., Jorge, A. N. C., & Adan, N. Z. W. Programación li-CC67-201502.

Luna, F., Millahual, C. P., & lacono, M. (2018). *PROGRAMACION WEB Full Stack 12-Sitios multiplataforma con Bootstrap: Desarrollo frontend y backend-Curso visual y práctico* (Vol. 12). RedUsers.