

Universidad Politécnica de Tecámac

INGENIERIA EN SOFTWARE

ASIGNATURA: Programación Visual

PROFESORA: Emmanuel Torres Servín

TAREA PROGRAMACIÓN VISUAL

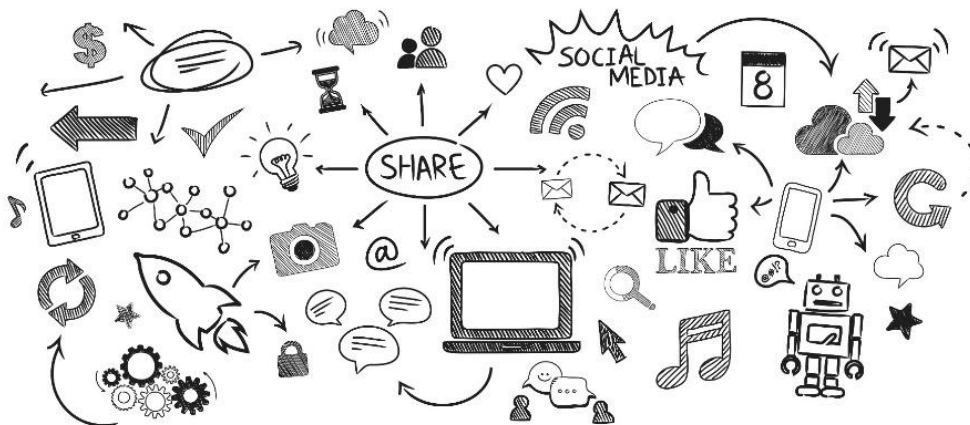
INTEGRANTES:

Ramírez George Samuel – 1321124307

Romero Hurtado Leslie Estefanía – 1321124300

GRUPO: 4322IS

ANÁLISIS DE LA PROGRAMACIÓN VISUAL



Conceptos de programación orientada a objetos.

La programación orientada a objetos es un paradigma de programación que se modela con objetos de hecho este paradigma se basa en el concepto de clases y objetos creando instancias individuales a los objetos. Algunos de los conceptos principales de POO son los siguientes:

- Objeto: Es un conjunto de variables y métodos que permite separar diferentes componentes de entidades de la vida real teniendo como resultado la instanciación de una clase en forma de un objeto
- Mensajes: Es el texto o petición que se le realiza para la ejecución de una tarea que requiera un objeto obteniendo valores de un atributo publico
- Método: Son todas las funciones que ayudan a que un objeto pueda efectuar un servicio respondiendo al mismo.
- Variable: es una palabra que representa algo que va a cambiar constantemente, que es inestables, inconsistente y mudable
- Atributos: características que diferencian a un objeto de otro, se puede decir que son las cualidades que se guardan en variables y estas pueden tener valores distintos,
- Instancia: es un modelo que procesa información, lee o especifica información, las instancias pueden ser todos los objetos de algún otro

Existen 4 pilares de la programación orientada a objetos que se describe en lo siguiente:

1. Encapsulación: es una característica de la programación orientada a objetos permite acceder a todo lo referente a un objeto a través de propiedades y métodos. En pocas palabras es cuando limitamos el acceso o damos un acceso restringido de una propiedad a los elementos que necesita a un solo miembro
2. Abstracción: La clase debe representar las características específicas de un objeto, en pocas palabras nos abstraer de la complejidad del mundo exterior dándonos una serie de atributos y comportamientos, es decir, propiedades y funciones, pudiendo crear objetos a partir de dicha clase
3. Herencia: Es cuando una clase obtiene rasgos (métodos y atributos) heredados desde otra clase existente. Teniendo una relación entre la clase general es decir una clase base y la clase más específica haciendo posible que podamos compartir métodos y datos entre clases, subclases y objetos
4. Polimorfismo: Es una cualidad de tener muchas formas, se refiere al hecho de tener varios objetos de diferentes clases base, pero se pueden utilizar de manera diferente sin tener que saber la clase exacta para poder hacerlo, teniendo la posibilidad de definir varias clases indistintas denominados de forma idéntica con métodos y atributos iguales

Características y aplicaciones de eventos.

Alguna de las características que podemos encontrar en la programación orientada a objetos y programación orientada a eventos son las siguientes:

- Dependencia de eventos: Acciones que se le dan al usuario con el objetivo de separar la lógica de procesamiento de eventos.
- Orientada al servicio: Se ocupa para programar vaya la redundancia programas que son utilizados para el servicio ejecutando así los servicios dados por el sistema
- Eventos: Es una condición que surge al ir haciendo la ejecución del programa y que requiere alguna acción por parte del sistema. Estos incluyen la interfaz y las acciones que deben estar en el programa cuando se vaya ejecutando, interactuando con un objeto (un claro ejemplo es cuando realizamos usar el mouse para seleccionar un botón)
- Controlador de eventos: Podemos decir, que es una función o método que ejecuta una acción o tarea específica cuando se activa un evento determinado
- Funciones de activación: Se utilizar para seleccionar que controlador de eventos ocupar, es decir, también son funciones que deciden que ejecutar cuando se produce dicho evento
- Tiempo controlado: Es una tarea preestablecida porque el código es ejecutado en un momento determinado

Algunas de sus aplicaciones son las siguientes:

- Clic sobre un botón
- Doble clic sobre el nombre de un fichero para abrirlo
- Arrastrar un icono
- Pulsar una tecla o una combinación de teclas
- Elegir una opción de un menú
- Escribir en una caja de texto
- Mover el ratón

Características de componentes y métodos visuales y no visuales.

Los componentes en programación visual es una clase que implementa una interfaz, o implementa una clase que permiten solicitar y/o brindar un servicio o funcionalidad y estos a su vez se subdividen en dos partes; componentes visuales y componentes no visuales.

Una característica importante de los componentes es que se pueden diseñar utilizando una clase que es un componente en el entorno del desarrollador

Los métodos o componentes visuales son aquellos que cumplen con ser una representación gráfica de tiempo, diseño y ejecución (ejemplo: ventanas, botones, cuadros de texto, barras scroll, etc.) de las páginas web, que este a su vez se subdivide en dos tipos:

1. Componentes interactivos: esto quiere decir que el usuario va a manipular los datos implícitos en el programa y los eventos que contenga
2. Componentes gráficos: esto se hace mediante librerías, funciones básicas como por ejemplo TCanvas, no suelen recibir eventos del usuario, pero si del programador

Los métodos o componentes no visuales (cuadros de dialogo no visibles, formato de diseño de los paneles, temporizadores, formularios dentro del sitio web, controladores, etc.).

Procesos de desarrollo visual en proyectos distribuidos y de escritorio.

Un proceso de desarrollo de software es la descripción de secuencias de múltiples actividades que son asignadas por el jefe de equipo para generar un proyecto coherente, el objetivo básico de este es hacer predecible el trabajo requiriendo de predecir el costo, mantener calidad y predecir el tiempo de desarrollo, pero también definiendo las actividades que cada persona va a cumplir, en las fases del proceso (especificación, análisis, etc.) o tareas mas concretas (crear diagramas, escribir código, etc.) facilitando el modelado visual junto con la capacidad de apreciar diferentes elementos e interacciones del sistema en distintos niveles de abstracción de los datos.

Requerimientos visuales de proyectos distribuidos y de escritorio.

Es importante que la gestión visual en los proyectos nos brinde la posibilidad de tomar un punto de vista beneficiando mas que nada a hacer un software de calidad y con una buena interfaz, para esto podemos utilizar lo siguiente:

- Cronogramas y gráficos de proyectos: por ejemplo, un diagrama de Gantt que ayuda a crear un cronograma de actividades, asignando mejor los recursos y tiempos a los programadores
- Tableros Kanban: trabajan con sprints que permiten ver el progreso del proyecto ayudando en el seguimiento de errores, planificación de sprints, análisis de sprints, reuniones diarias de actualización ágiles y gestión de producción web
- Se considera que debemos definir los objetivos tanto específicos como generales del proyecto, determinar resultados iniciales, identificar riesgos y limitaciones, crear una ruta visual, evaluar recursos, trazar un plan y documentar

Herramientas y lenguajes de programación visual.

- ✚ C# / ASP.NET Core: compila aplicaciones web y servicios conectados a internet y basadas a la nube, aplicaciones de IoT y servicios de back-end móviles utilizando los lenguajes HTML y JavaScript
- ✚ Razor: La sintaxis Razor de ASP.NET le permite alinear C# directamente en sus vistas. Además, tiene acceso completo al modelo actual y a los tipos de .NET Framework.
- ✚ Javascript: nos permite crear aplicaciones web interactivas, con cambios en los contenidos en tiempo real, efectos personalizados y muchas características más.
- ✚ CSS, Less, Sass y Font Awesome en ASP.NET Core: posibilidad de definir y mantener de manera eficaz los estilos y los archivos de hojas de estilo (CSS), además de tener un fácil acceso a los iconos sin imagen que ayudan a que la interfaz del sitio sea más intuitiva. Ahí es donde entran los lenguajes y herramientas que admiten Less y Sass, y bibliotecas como Font Awesome.
- ✚ Python: Lenguaje de programación dinámico y de alto nivel orientado a objetos, se usa para el desarrollo rápido de aplicaciones.
- ✚ Visual Basic: orientado a objetos, está dirigido a la plataforma .NET, lo cual le permitirá crear aplicaciones de Windows sin dificultad.
- ✚ C++: Un lenguaje de programación y entorno de desarrollo eficaz y flexible para crear aplicaciones para Windows, Linux, iOS y Android.
- ✚ F#: con funcionalidades orientadas a objetos para la plataforma .NET, y que se centra en facilitar la programación funcional de cualquier tarea.

REFERENCIAS

1. José M. Drake. (-). Microsoft PowerPoint - OO_08_I2_Proceso.ppt. 04/08/2022, de Master de Computación Sitio web: https://www.ctr.unican.es/asignaturas/MC_OO/Doc/OO_08_I2_Proceso.pdf
2. José Manuel Alarcón. (26 de julio de 2021). Los conceptos fundamentales sobre Programación Orientada Objetos explicados de manera simple. 05/08/2022, de Campus MVP Sitio web: <https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>
3. Helmut Sy Corvo. (30 de marzo de 2020). Programación orientada a eventos: características, ejemplos, ventajas, aplicaciones. 05/08/2022, de Lifeder Sitio web: <https://www.lifeder.com/programacion-orientada-a-eventos/>
4. Lizbeth Ramos. (6 de abril del 2016). COMPONENTES VISUALES Y NO VISUALES. 05/08/2022, de Blogspot Sitio web: http://temasliz.blogspot.com/2016/04/componentes-visuales-y-no-visuales_6.html
5. Alejandro Bedini González. (-). Gestión de Proyectos de Software.pdf. 05/08/2022, de - Sitio web: <https://www.inf.utfsm.cl/~guerra/publicaciones/Gestion%20de%20Proyectos%20de%20Software.pdf>
6. Jesús Moreno León. (octubre 26, 2016). ¿LENGUAJES VISUALES?. 05/08/2022, de Programamos videojuegos y "apps" Sitio web: <https://programamos.es/lenguajes-de-programacion-visuales-o-lenguajes-basados-en-texto/>