



UNIVERSIDAD POLITÉCNICA DE TECÁMAC.

Materia: Programación Visual

Docente: Emmanuel Torres Servín

Grupo: 5322IS

Actividad: Análisis de la programación visual

Integrantes:

- ✓ Alemán Pérez Natali Joselin
- ✓ Ángel Velasco Marco Joel
- ✓ Isaac de León Carbajal

Matriculas:

1321124050
1321124015
1321124054

Contenido

Conceptos de la Programación Orientada a Objetos	1
Clases.....	1
Objetos	1
Mensajes	1
Los Pilares de la POO	1
Encapsulamiento.....	1
Abstracción	1
Herencia.....	1
Polimorfismo	2
Características y aplicaciones de eventos	2
Características de componentes y métodos visuales y no visuales	2
Componentes visuales	2
Componentes no visuales.....	3
Métodos.....	3
Procesos de desarrollo visual en proyectos distribuidos y de escritorio	3
Requerimientos visuales de proyectos distribuidos y de escritorio.....	4
Herramientas y lenguajes de programación visual.....	5
Fuentes de información	6



Conceptos de la Programación Orientada a Objetos

Clases

Una clase es una plantilla. Define de manera genérica cómo van a ser los objetos de determinado tipo. Es un molde del que luego se crean múltiples objetos con características similares.

Objetos

Un objeto es una entidad concreta que se crea a partir de la plantilla que es la clase. Este nuevo objeto tiene ya "existencia" real, puesto que ocupa memoria y se puede utilizar en el programa. Es el conjunto de variables o datos y métodos o funciones relacionadas entre sí.

Mensajes

Es la petición de un objeto a otro para solicitar la ejecución de alguno de sus métodos o para obtener el valor de uno de sus atributos públicos. Siempre hay un receptor, lo que no ocurre a una llamada a procedimiento.

Los Pilares de la POO

Encapsulamiento

Permite que todo lo referente a un objeto quede aislado dentro de éste. Es decir, que todos los datos referentes a un objeto queden "encerrados" dentro de éste y sólo se puede acceder a ellos a través de los miembros que la clase proporcione (propiedades y métodos).

Abstracción

Implica que la clase debe representar las características de la entidad hacia el mundo exterior, pero ocultando la complejidad que conllevan. Esto es, abstraer la complejidad que haya dentro dando una serie de atributos y comportamientos (propiedades y funciones) específicos de un objeto, los que los distinguen de los demás.

Herencia

Es el resultado de crear un modelo a partir de otro. Es decir, cuando una clase hereda de otra obtiene todos los rasgos que tuviese la primera. La clase que hereda de otra obtiene todos los rasgos de la primera y añade otros nuevos y además también puede modificar algunos de los que ha heredado.

A la clase de la que se hereda se le llama clase base, y a la clase que hereda de ésta se le llama clase derivada.



Polimorfismo

El concepto de polimorfismo se refiere al hecho de que varios objetos de diferentes clases, pero con una base común, se pueden usar de manera indistinta, sin tener que saber de qué clase exacta son para poder hacerlo.

También se refiere al a posibilidad de definir métodos diferentes denominados de forma idéntica dentro de una misma clase, con alguna característica que los identifique, como su tipo de retorno o cantidad y/o tipo de parámetros.

Características y aplicaciones de eventos

Los eventos son las acciones sobre el programa. Cuando se produce o dispara un evento sobre un determinado componente, se da inicio a un conjunto de acciones programadas por el programador para ese evento concreto. Algunos ejemplos de los eventos son dar clic a un botón, arrastrar un icono, pulsar una tecla o una combinación de teclas, escribir en una caja de texto, etc.

Los eventos son recogidos por un bucle exterior permanente y distintos procesos se encargan de tratarlos. Habitualmente, este bucle externo permanece oculto al programador que simplemente se encarga de tratar los eventos, aunque en algunos entornos de desarrollo (IDE) será necesaria su construcción.

Los eventos permiten crear código por partes o módulos, que se ejecutarán sólo cuando se dispare cierto evento. Esto logra un mantenimiento más sencillo, ya que si alguno de los eventos no funciona como se espera, sólo se debe revisar y corregir el fragmento de código que corresponde a ese evento, en lugar de tener que buscar en todo el programa. Además, la escritura de código por fragmentos también facilita la reutilización de estos fragmentos.

Características de componentes y métodos visuales y no visuales

Componentes visuales

Un componente es visual cuando tiene una representación gráfica en tiempo de diseño y ejecución (botones, barras de desplazamiento, cuadros de edición, etc.)



Componentes no visuales

Caso contrario, es decir, no cuentan con una representación gráfica (temporizadores, cuadros de diálogo -no visibles en la fase de diseño-, etc).

Métodos

Un método es un bloque de código que tiene definido en su interior un conjunto de instrucciones, las cuales realizan una determinada tarea. Un programa hace que se ejecuten las instrucciones al llamar al método y especificando los argumentos de método necesarios.

Procesos de desarrollo visual en proyectos distribuidos y de escritorio

Un sistema distribuido es un conjunto o grupo de equipos que son independientes entre sí y que actúan como un único equipo de forma transparente y que tienen como objetivo la descentralización del procesamiento o el almacenamiento de información. La distribución permite obtener grandes prestaciones con un coste razonablemente bajo. En la actualidad, la mayoría de los sistemas informáticos son distribuidos y no dependen de un único nodo o equipo para funcionar.

Las aplicaciones de escritorio se pueden descargar desde el Microsoft Store, descargar desde Internet o instalarse con algún tipo de medio (como un CD, DVD o dispositivo de almacenamiento USB). Se abren con un archivo .EXE, .MSI o .DLL y normalmente se ejecutan en el dispositivo, a diferencia de las aplicaciones basadas en web (que se ejecutan en la nube).

Las fases comunes del desarrollo de software son:

- Planificación: El propósito de este paso es averiguar el alcance del problema y determinar las soluciones. En esta etapa se deben considerar los recursos, costos, tiempo, beneficios y otros elementos.
- Requerimientos: Se consideran los requisitos funcionales del proyecto o la solución. También es donde tiene lugar el análisis del sistema y de las necesidades de los usuarios finales para garantizar que el nuevo sistema pueda satisfacer sus expectativas.
- Diseño: Describe en detalle las especificaciones, características y operaciones necesarias que lograrán los requisitos funcionales del sistema propuesto que se implementará.
- Desarrollo: Es cuando se contrata a un programador, ingeniero de redes y / o desarrollador de bases de datos para que realice el trabajo principal en el proyecto.



- Integración y pruebas: Implica la integración de sistemas y las pruebas del sistema (de programas y procedimientos), normalmente realizadas por un profesional de aseguramiento de la calidad (QA)
- Implementación: Esta fase implica la instalación real del sistema recientemente desarrollado. Este paso pone el proyecto en producción moviendo los datos y componentes del sistema antiguo y colocándolos en el nuevo sistema a través de una transición directa.
- Operaciones y mantenimiento: Conlleva el mantenimiento y las actualizaciones periódicas que el sistema requiera según las necesidades de los usuarios finales.

Requerimientos visuales de proyectos distribuidos y de escritorio

Un requerimiento define las funciones, capacidades o atributos intrínsecos de un sistema de software, en otras palabras, describe como un sistema debe comportarse.

Los requerimientos se pueden dividir en dos grandes categorías:

- Requerimientos funcionales: Son aquellos que describen cualquier actividad que el sistema deba realizar, es decir, el comportamiento o función particular de un sistema o software cuando se cumplen ciertas condiciones.
- Requerimientos no funcionales: Definen características o cualidades generales que se esperan de un sistema y establecen restricciones sobre el producto, el proceso de desarrollo de software y restricciones externas que el software debe lograr.

Algunos de los requerimientos visuales para cualquier software con estos elementos son:

- Simplificar: Muchas aplicaciones pecan de querer poner demasiadas cosas, pero un diseño limpio y cuidado, con lo estrictamente necesario, hace que la experiencia de usuario sea mejor.
- Identidad: Hay que saber diferenciar cada una de las cosas, pero todo tiene que mantener una misma identidad.
- Buen uso de los iconos: Aunque no hay que abusar de ellos, el uso de iconos es una buena idea. Además, tienen que ser suficientemente grandes para que se puedan tocar con facilidad.
- Tipografía: No debe ser muy grande ni muy pequeña, ser fácilmente legible y con buena resolución. El contraste con el fondo y la distancia de interlineado son factores fundamentales para que sea fácil de leer.



- Código de colores: Tenemos asimilado que un elemento rojo hace referencia a un error, así como uno verde a un acierto o confirmación. Mal usados pueden generar reticencia o falta de entendimiento por parte de los usuarios.
- Lenguaje visual: El usuario tiene que poder entender las opciones disponibles con una sola imagen, o con el uso de diferentes colores, entre otras técnicas.

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. La apariencia y el etiquetado de los módulos permite identificar qué tarea en el flujo del programa pueden resolver. Los pictogramas sirven para orientar al usuario. Así, no se necesitan estructuras muy complejas ni un alto grado de abstracción.

Algunos de los lenguajes de programación visual son:

- Scratch: Proyecto del Grupo Lifelong Kindergarten del MIT Media Lab. Se ofrece de forma gratuita.
- NEPO, es un software de programación libre inspirado en Scratch. Fue creado por el MIT (Massachusetts Institute of Technology).
- Blockly: Proporciona un editor de programación visual al que se añaden aplicaciones Android, iOS y web. Blockly también utiliza bloques gráficos que encajan entre ellos.
- Grape: Es un entorno de desarrollo gráfico. Permite incluso a los principiantes programar con microcontroladores en pasos simples.
- App Inventor: Esta interfaz gráfica permite programar aplicaciones para teléfonos móviles Android con bloques gráficos.
- Ardublock: Este lenguaje de programación gráfica está especialmente diseñado para programar el microcontrolador Arduino sin introducir texto.
- Pure Data: Este lenguaje de programación visual está orientado tanto a flujos de datos como a los entornos de desarrollo.
- Lego Mindstorms: Su núcleo es la pieza de Lego programable: los motores eléctricos, sensores y piezas de tecnología propios de Lego permiten construir y programar robots y otros sistemas interactivos.



Fuentes de información

Alarcón, J. M. (Julio de 2021). Los conceptos fundamentales sobre Programación Orientada Objetos explicados de manera simple. Campus MVP. Recuperado de: <https://www.campusmvp.es/recursos/post/los-conceptos-fundamentales-sobre-programacion-orientada-objetos-explicados-de-manera-simple.aspx>

Andres (s / f). ¿Qué es la programación orientada a eventos? TIC. Recuperado de: <http://contenidos.sucerman.com/nivel3/dispositivos/unidad1/leccion2.html>

Cortijo, F. (s / f). Componentes. Elvex. Recuperado de: <https://elvex.ugr.es/decsai/builder/intro/3.html>

Desarrollo Web (Septiembre de 2020). Programación visual: la entrada más sencilla al mundo digital. IONOS. Recuperado de: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/programacion-visual/>

Sistemas (Junio de 2020). La importancia de una arquitectura distribuida. ILimit. Recuperado de: <https://www.ilimit.com/blog/importancia-arquitectura-distribuida/>

Gómez, C. (Junio de 2020). Requerimientos de software. Diario de QA. Recuperado de: <https://www.diariodeqa.com/post/quiero-ser-qa-que-debo-aprender-requerimientos-de-software>

Unitel (s / f). Los 10+1 Requisitos a tener en cuenta en el diseño de una APP. Unitel. Recuperado de: <https://unitel-tc.com/10-requisitos-diseno-crear-app/>