

UD2. Programación de interfaces



Ciclo: Desarrollo de Aplicaciones Multiplataforma
Módulo: Desarrollo de interfaces.

Objetivos

- Generar interfaces gráficas de usuario mediante editores visuales utilizando la funcionalidad del editor y adaptando el código generado.
- Crear una interfaz gráfica utilizando los asistentes de un editor visual.
- Utilizar las funciones del editor para localizar los componentes de la interfaz.
- Modificar las propiedades de los componentes para adecuar a las necesidades de la aplicación.
- Analizar y modificar el código generado por el editor visual.
- Asociar a los eventos las acciones correspondientes.

LENGUAJES DE PROGRAMACIÓN. TIPOS.

PARADIGMAS DE PROGRAMACIÓN

- Un lenguaje de programación es un lenguaje formal diseñado para expresar procesos que pueden ser llevados a cabo por máquinas como las computadoras.
- Pueden usarse para crear programas que controlen el comportamiento físico y lógico de una máquina, para expresar algoritmos con precisión, o como modo de comunicación humana.
- Está formado por un conjunto de símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.
- Al proceso por el cual se escribe, se prueba, se depura, se compila (de ser necesario) y se mantiene el código fuente de un programa informático se le llama programación.

- Permite especificar de manera precisa sobre qué datos debe operar una computadora, cómo deben ser almacenados o transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias.
- Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural.

TIPOS DE LENGUAJES DE PROGRAMACIÓN

- Existen dos tipos de lenguajes claramente diferenciados: los lenguajes de bajo nivel y los de alto nivel.
- El ordenador solo entiende un lenguaje conocido como código binario o código máquina, consistente en ceros y unos. Es decir, solo utiliza 0 y 1 para codificar cualquier acción.

Lenguajes de bajo nivel

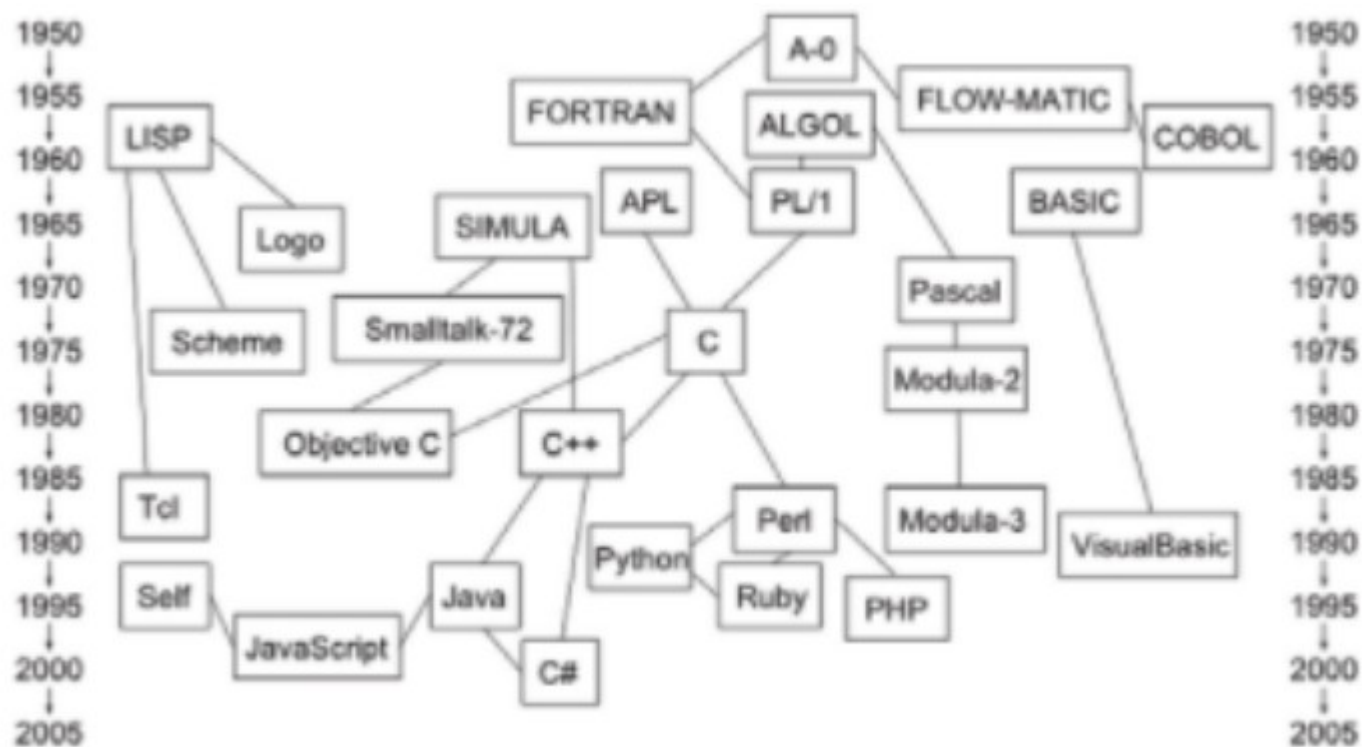
- Son lenguajes totalmente dependientes de la máquina, es decir, que el programa que se realiza con este tipo de lenguajes no se pueden migrar o utilizar en otras máquinas.
- Al estar prácticamente diseñados a medida del hardware, aprovechan al máximo las características del mismo.

Dentro de este grupo se encuentran:

- El lenguaje máquina: este lenguaje ordena a la máquina las operaciones fundamentales para su funcionamiento. Consiste en la combinación de ceros y unos para formar las órdenes entendibles por el hardware de la máquina.
- El lenguaje ensamblador: es un derivado del lenguaje máquina y está formado por abreviaturas de letras y números llamadas nemotécnicos.

Lenguajes de alto nivel

- Son aquellos que se encuentran más cercanos al lenguaje natural que al lenguaje máquina.
- Dirigidos a solucionar problemas mediante el uso de EDD (Estructuras Dinámicas de Datos).
- Se trata de lenguajes independientes de la arquitectura del ordenador.
- Estos lenguajes permiten al programador olvidarse por completo del funcionamiento interno de las máquinas para las que está diseñando el programa.



Evolución de los lenguajes de programación.

PARADIGMAS DE PROGRAMACIÓN

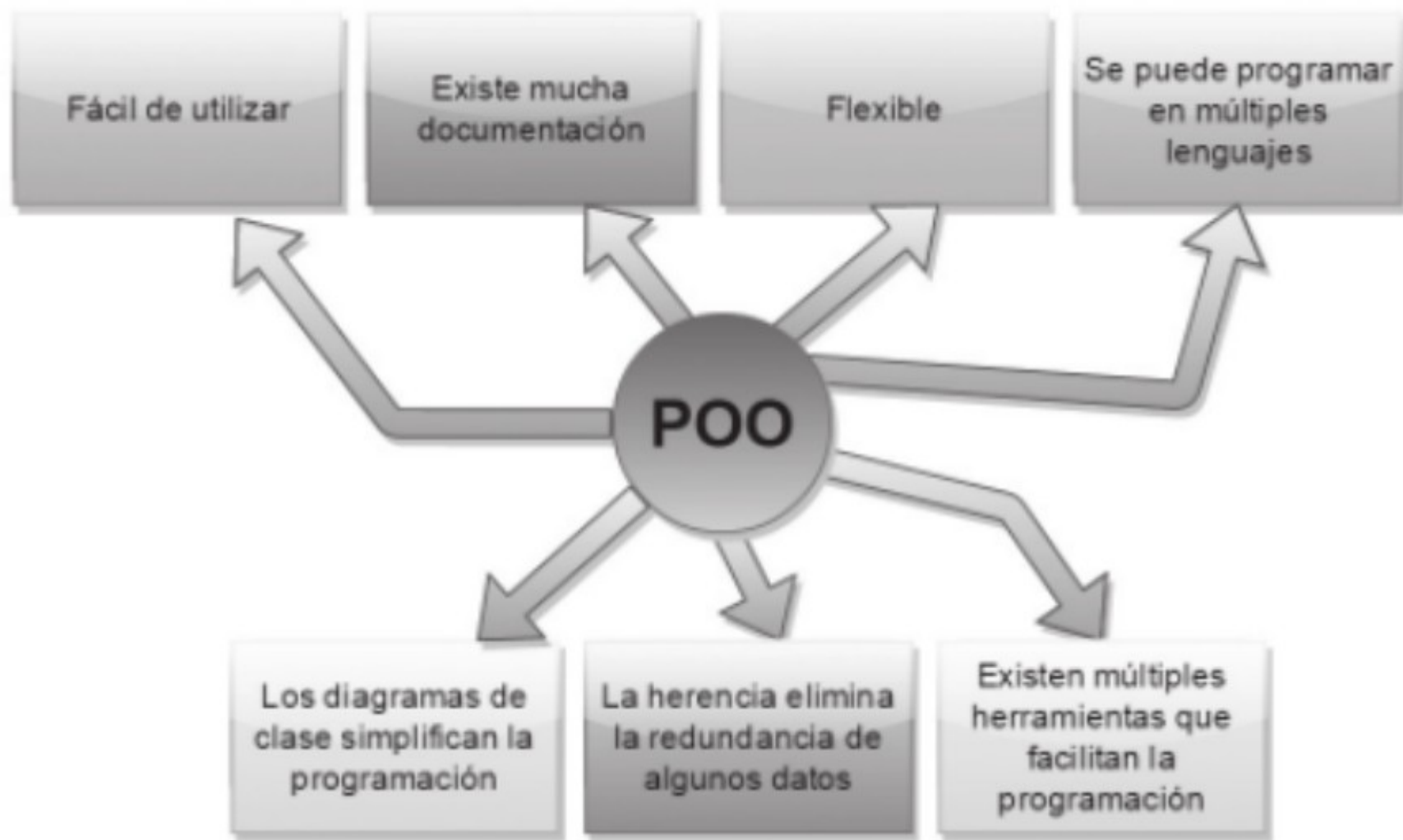
- Un paradigma de programación es una propuesta tecnológica que es adoptada por una comunidad de programadores cuyo núcleo central es resolver uno o varios problemas claramente delimitados.
- Es un estilo de programación empleado.
- La resolución de estos problemas debe suponer consecuentemente un avance significativo en al menos un parámetro que afecte a la ingeniería de software,

PROGRAMACIÓN ORIENTADA A OBJETOS

- La programación orientada a objetos o POO (OOP según sus siglas en inglés) es un paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.
- Está basado en varias técnicas, incluyendo herencia, cohesión, abstracción, polimorfismo, acoplamiento y encapsulamiento.

Los objetos son entidades que tienen un determinado estado, comportamiento (método) e identidad:

- El estado está compuesto de datos o informaciones, serán uno o varios atributos a los que se habrán asignado unos valores concretos (datos).
- El comportamiento está definido por los métodos o mensajes a los que sabe responder dicho objeto, es decir, que operaciones se pueden realizar con él.
- La identidad es una propiedad de un objeto que lo diferencia del resto; dicho con otras palabras, es su identificador (concepto análogo al de identificador de una variable o una constante).



- Un objeto contiene toda la información que permite definirlo e identificarlo frente a otros objetos pertenecientes a otras clases e incluso frente a objetos de una misma clase, al poder tener valores bien diferenciados en sus atributos.
- A su vez, los objetos disponen de mecanismos de interacción llamados métodos, que favorecen la comunicación entre ellos.
- Esta comunicación favorece a su vez el cambio de estado en los propios objetos.
- Esta característica lleva a tratarlos como unidades indivisibles, en las que no se separa el estado y el comportamiento

PROGRAMACIÓN DIRIGIDA POR 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.
- El creador de un programa dirigido por eventos debe definir los eventos que manejarán su programa y las acciones que se realizarán al producirse cada uno de ellos, lo que se conoce como el administrador de evento.
- En la programación dirigida por eventos, al comenzar la ejecución del programa se llevarán a cabo las inicializaciones y demás código inicial y, a continuación, el programa quedará bloqueado hasta que se produzca algún evento. Cuando alguno de los eventos esperados por el programa tenga lugar, el programa pasará a ejecutar el código del correspondiente administrador de evento.

- Por ejemplo, si el evento consiste en que el usuario ha hecho clic en el botón Play de un reproductor de películas, se ejecutará el código del administrador de evento, que será el que haga que la película se muestre por pantalla.
- La programación dirigida por eventos 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

PROGRAMACIÓN BASADA EN COMPONENTES

- La programación orientada a componentes (que también es llamada basada en componentes) es una rama de la ingeniería del software, con énfasis en la descomposición de sistemas ya conformados, en componentes funcionales o lógicos, con interfaces bien definidas usadas para la comunicación entre componentes
- Se considera que el nivel de abstracción de los componentes es más alto que el de los objetos y, por tanto, no comparten un estado y se comunican intercambiando mensajes que contienen datos.

- Un componente de software es un elemento de un sistema que ofrece un servicio predefinido, y es capaz de comunicarse con otros componentes.
- Una definición más simple puede ser un componente es un objeto escrito de acuerdo a unas especificaciones. No importa que especificación sea esta, siempre y cuando el objeto se adhiera a la especificación.
- Solo cumpliendo correctamente con esa especificación es que el objeto se convierte en componente y adquiere características, como reusabilidad.
- La capacidad de ser reutilizado (reusability), es una característica importante de los componentes de software de alta calidad. Un componente debe ser diseñado e implementado, de tal forma que pueda ser reutilizado en muchos programas diferentes.

Conclusión....

Programación orientada a objetos.

- El desarrollo de interfaces gráficas permite la creación del canal de comunicación entre el usuario y la aplicación.
- Tener en cuenta la importancia del diseño debido a ello.
- Las herramientas de desarrollo permiten la implementación del código relativo al proceso de creación.
- La programación orientada a objetos permite utilizar entidades o componentes que tienen su propia identidad y comportamiento.

HERRAMIENTAS DE EDICIÓN DE INTERFACES

IDE

- Para desarrollar software por su alta productividad se utilizan entornos de desarrollo o IDE. Un entorno de desarrollo (IDE) suele tener los siguientes componentes, aunque no necesariamente todos:
 - Un editor de texto.
 - Un compilador.
 - Un intérprete.
 - Un depurador
 - Un cliente.
 - Posibilidad de ofrecer un sistema de control de versiones.
 - Facilidad para ayuda en la construcción de interfaces gráficas de usuario.

Microsoft Visual Studio

- Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual JH y Visual Basic .NET, al igual que entornos de desarrollo web como ASP. NET. Aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.
- Es un producto comercial, aunque Microsoft tiene versiones gratuitas pero no libres con ciertas limitaciones en la explotación de las aplicaciones a desarrollar.
- Actualmente está en su versión 2019 (Vídeo aules...)

Interfaces gráficas. WPF

- WPF (Windows Presentation Foundation)
- Sucesor de Windows Forms.
- WPF es una API (Applications Programming Interfaz) perteneciente al framework .NET para la creación de interfaces de usuario (aplicaciones gráficas de escritorio) bajo Windows.

- Con WPF se pueden crear interfaces gráficas utilizando un lenguaje de marcado (etiquetas) similar a HTML (XAML).
- Con Visual Studio se pueden crear interfaces de usuario de forma visual generando código XAML por detrás (code behind).
- Las interfaces gráficas WPF son vectoriales.
- WPF utiliza Data binding. MVC. MVVM