

Programación Orientada a Objetos

Introducción y brevísima historia.

Introducción a la POO

2

En el transcurso de estas dos clases, vamos a ir familiarizándonos con todos estos conceptos.

Y vamos a poder usarlos!

Ahora sobrevolálos nomás...



Introducción a la POO

3

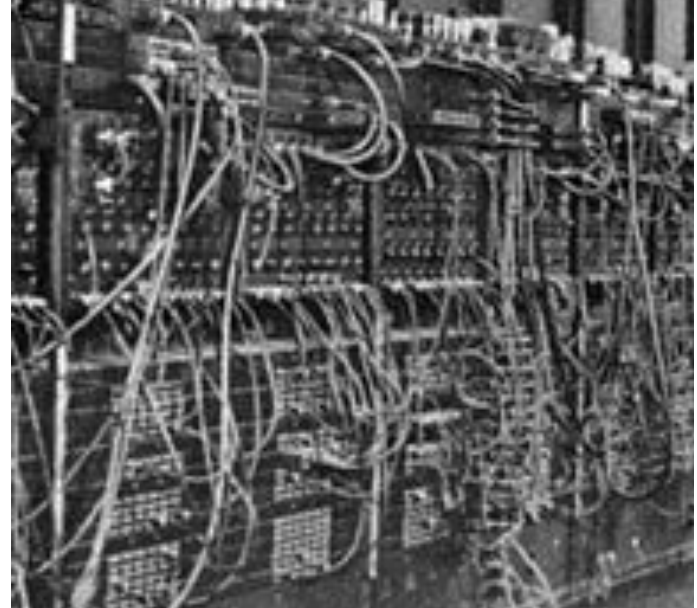
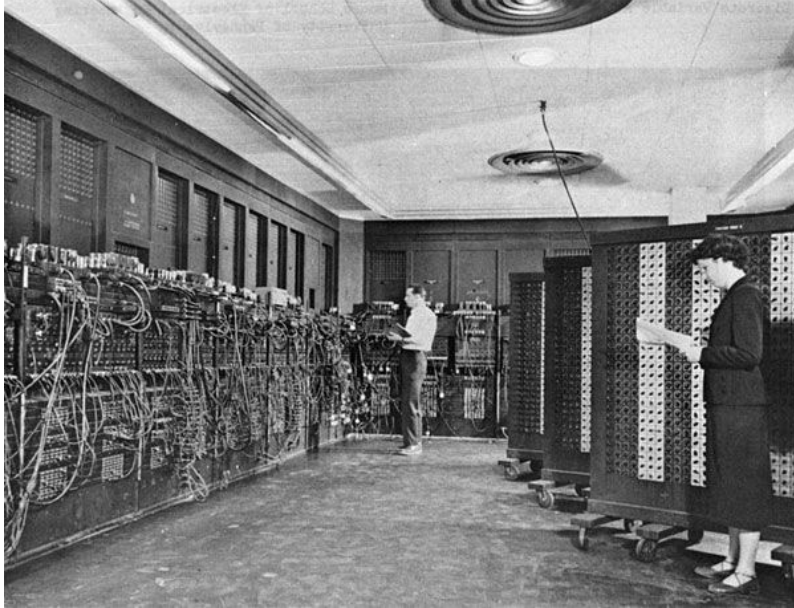
Imperdonable brevísima historia.

En los orígenes de la computación, las máquinas eran muy poco amigables.

Para enseñarles a hacer algo, había directamente que “cablearlas”. Modificar sus circuitos mismos.

Y de esta forma, podían hacer sólo la tarea para la que se las había “cableado”.

Introducción a la POO



Introducción a la POO

Con el paso de los años, las computadoras comenzaron a ser **programables**.

De alguna forma se les podía “incorporar” una **secuencia de instrucciones**, para que hagan una tarea, u otra secuencia de instrucciones distinta, para que hagan alguna otra tarea...

Introducción a la POO

En un principio, esta secuencia de instrucciones estaba estrechamente ligada a la ingeniería de la máquina.

En un momento, las máquinas se programaron en **Assembler**. Un lenguaje muy poco amigable, que hablaba casi directamente con los circuitos.

Este lenguaje es rapidísimo.

Introducción a la POO

Las máquinas fueron más y más rápidas, y la necesidad de optimizar tanto la programación fue cediendo, frente a la necesidad de resolver más y más problemas, cada vez más complejos y variados.

Pasamos de enormes máquinas que estaban dedicadas a “romper” encriptaciones en tiempos de guerra, por ejemplo, a máquinas que podían serle de asistencia en su trabajo a grandes empresas.

Introducción a la POO

Los ingenieros y programadores, comenzaron entonces a diseñar lenguajes que fuesen más amigables. De más alto nivel.

Bajo nivel → el circuito | Alto nivel → el habla

Y en algún momento aparecieron las Computadoras Personales.

Introducción a la POO

Con la llegada de las PC. Las máquinas pasaron de resolver problemas de guerra... a resolver problemas de grandes empresas... a resolver problemas cotidianos del día a día. Incluso en los hogares.

Y años más tarde, se harían presentes en los bolsillos de cada uno de nosotros.

Introducción a la POO

Así en esta *imperdonable brevísima historia*, les contamos cómo evolucionaron las computadoras y los lenguajes de programación en lo que son hoy en día.

Ahora veámos cómo evolucionó la programación un poco más en detalle...

Introducción a la POO

Assembler: Programación a muy **bajo nivel**. Sobre el circuito de la máquina.

Programación secuencial/estructurada: Programación en la que una instrucción sigue a la anterior, y dónde el programador debe manejar y controlar todos los procesos “a mano”.

Todo el código está en el mismo lugar. Corregir un error, significa rever y volver a testear todo el código.

Introducción a la POO

Programación procedimental: Se extrapola de la programación secuencial que muchas veces se repiten líneas de código. Se extraen estas repeticiones en procedimientos y funciones. Corregir un error puede ser más sencillo al tener “partes” más reconocibles en el código. Pero de todas formas el código se ve como un todo.

Módulos: Aparecen agrupaciones de procedimientos y funciones. Esto presenta algún grado de abstracción.

Se “modulariza” aún más y mejora el tema de corrección de errores y testeos.

Introducción a la POO

Programación Orientada a Objetos - Cambia el paradigma.

En este paradigma, se plantea **dividir el problema a resolver, en problemas menores**, que se representarán con **Objetos**.

Estos **Objetos** tendrán relación a elementos del **contexto real del problema** a resolver.

A estos **Objetos** se los dotará de **características** y de **responsabilidades** para que puedan operar y responder a la necesidad particular que nos interese.

Estos **Objetos** podrán **interactuar** entre ellos.

Introducción a la POO

Cuando organicemos/pensemos/diseñemos el código según el paradigma Orientado a Objetos, el resultado en el momento de ejecución será el mismo que de otra forma, pero el código será:

- » **mucho más mantenible**
- » **legible**
- » **escalable**
- » **fácil de corregir y actualizar.**

Introducción a la POO

En un ejemplo de un sistema para una **Veterinaria**. Seguramente identificaré objetos como **AlimentoBalanceado**, **AnimalDomestico**, **Perro**, **Gato**, **Canario**, **Empleado**, **Cliente**, **Veterinario**...

Estos Objetos podrán relacionarse entre si.

Por ejemplo el **Veterinario**, podrá alimentar al **Perro**.

Introducción a la POO

Estos **Objetos** podrán hacer cosas (porque tendrán **responsabilidades**), y podrán responder según su Estado.

Por ejemplo, un **Canario**, “podrá decirnos su nombre”, porque tendrá un *String nombre* = “Tweety” como una de sus características.

Corrección de Errores / Testeo

Con este diseño, por ejemplo si tengo algún inconveniente con el funcionamiento de mis Gatos. Sabré exactamente en qué código ir a buscar. No tendré que modificar (y volver a testear), el código de los otros Objetos.

Introducción a la POO

Modelado

Con POO, modelamos aspectos de la vida cotidiana que nos interesen según el contexto del problema que tenemos que resolver.

Usaremos los ingredientes concretos y cotidianos que están involucrados en el problema, y modelaremos a partir de ellos y de sus interacciones.

Al utilizar en el modelado elementos de la vida cotidiana, podremos hablar con mayor facilidad con personas no coders (cliente, alguien de coordinación, alguien del equipo de otra disciplina...).

Introducción a la POO

Abstracción

En el momento de modelar, debemos poder extrapolar de la realidad - del contexto del problema que tenemos que resolver - **las piezas fundamentales.**

Los actores principales.

Y de estas piezas, tenemos que lograr **generalizar (abstraer)** qué **características y comportamientos** las podrán representar.

En el sistema ejemplo de la **Veterinaria**, vimos claramente **Perros** y **Gatos** (objetos que representan *cosas* bastante concretas), pero también podríamos modelar **Ventas**, **Turnos...**