# UF1

Entornos de Desarrollo

Olimpia Olguín Tinoco











- Desarrollo de Software
- Instalación y uso de entornos de desarrollo





 Entorno de Desarrollo: es el conjunto de procedimientos y herramientas utilizadas para la creación o mantenimiento de programas informáticos (software).



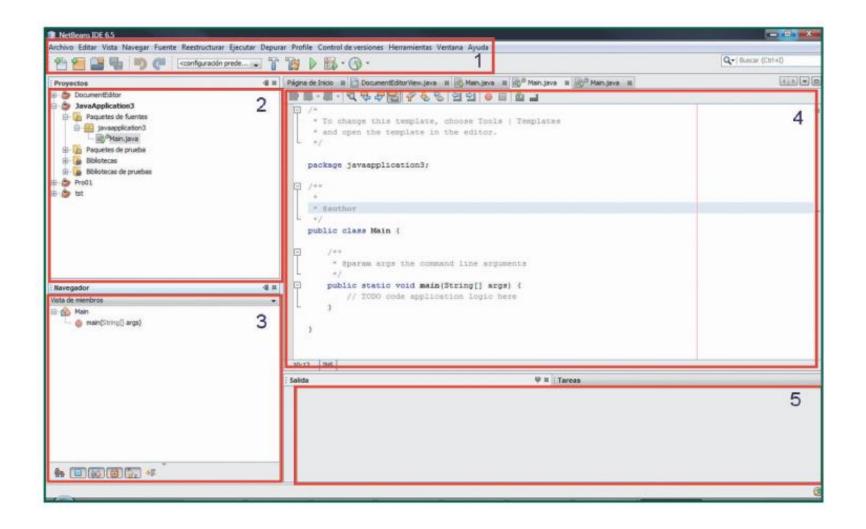
IDE: (Integrated Development Environment o Entorno de Desarrollo Integrado) al entorno de programación que contiene un conjunto de procesos, instrumentos y normas que trabajan de forma coordinada para facilitar al programador el control de las diferentes etapas del desarrollo en las que está directamente implicado







### 









### IDE

- **Área 1.** Menú principal, desde donde se realizan las principales acciones del entorno (fcheros, depuración y control del proyecto).
- Área 2. Ventana de proyecto. Muestra una visión de conjunto del proyecto (fcheros, librerías necesarias para el código entrado, etcétera).
- Área 3. Ventana de navegación. Permite acceder de forma rápida a los ele-mentos de las clases que tengamos seleccionadas.
- Área 4. Ventana de edición. Editor de texto a través del cual entramos en el código del desarrollo de nuestro proyecto.
- Área 5. Ventana de salida / Tareas. Muestra los errores de compilación e información seleccionada por el programador mediante palabras clave del código.





- Desarrollo de software es el proceso que tiene lugar desde que un usuario plantea una mejora de su actual sistema informático hasta que dicha solución (nuevo software creado o modificado de uno ya existente) le es entregado para su inmediata utilización.
- También incluye aspectos relacionados con el negocio del cliente, como por ejemplo: las fechas de entrega, la valoración económica, el cálculo de recursos necesarios y acciones de ajuste de personal, y el presupuesto.

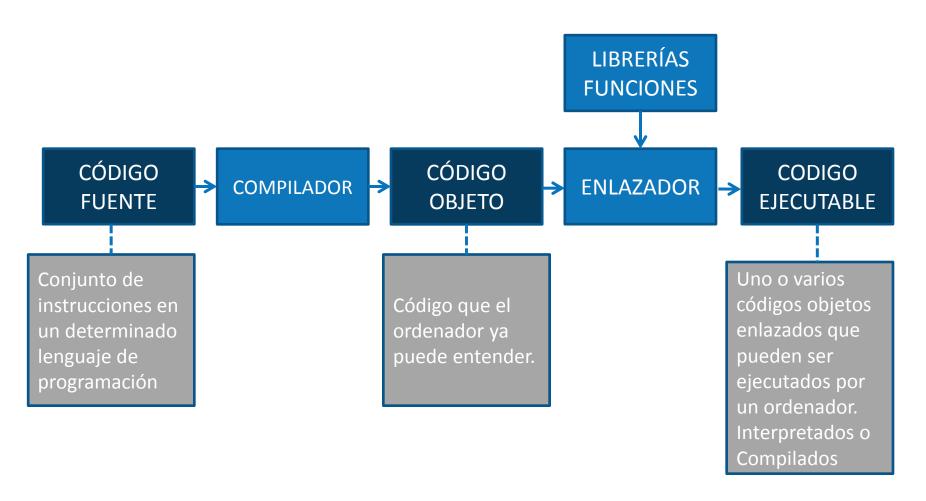
### Desarrollo de software = aspectos tecnológica + aspectos de negocio

- **Un programa informático** es un conjunto de instrucciones que indican y permiten a un ordenador realizar una tarea determinada.
- Algoritmo: conjunto de instrucciones con normas claras, finitas y ordenadas en el tiempo. Puede ser sencillo o complicado



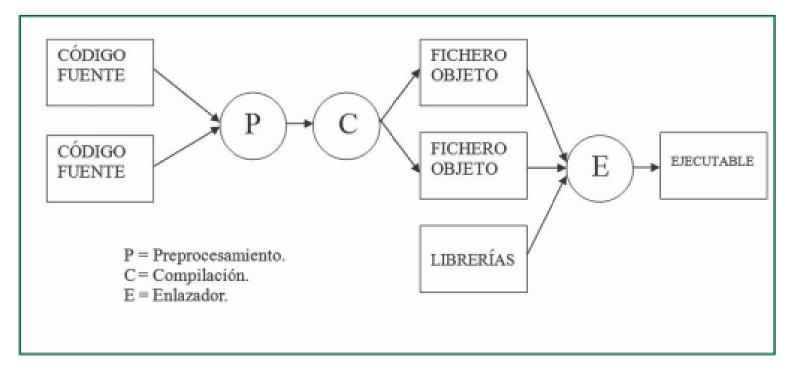


## Programa informático









**Preprocesamiento**: Se hace un rastreo del código fuente para detectar todos los posibles errores de entrada de código (paréntesis sin cerrar, asignaciones de variables inválidas, etcétera).





### Máquina virtual

**Una máquina virtual** es un software que simula un entorno de ordenador concreto en el que podemos instalar un sistema operativo, o programas, y ejecutarlos.

Es como un ordenador simulado dentro de otro ordenador

La enfocaremos a el lenguaje Java, que utiliza una máquina virtual para ejecutarse, que es la llamada JVM (Máquina Virtual de Java).







Dividimos los lenguajes de programación en tres grandes grupos:

- Lenguaje procedural (lista de acciones a realizar siempre en el mismo orden). Éste se caracteriza por: un punto de comienzo, un proceso y un final. Los procedimientos pueden ser funciones, rutinas o métodos. Basic y Fortran.
- Lenguaje estructurado (variante del anterior con mayor organización interna)
  Romper la estructura monolítica de los programas en módulos más pequeños y
  manejables así como diferenciar variables globales y locales visibles desde un módulo en
  concreto. C y Pascal
- Lenguaje orientado a objetos (Se crean estructuras de datos y operaciones que aceptan dichas estructuras)

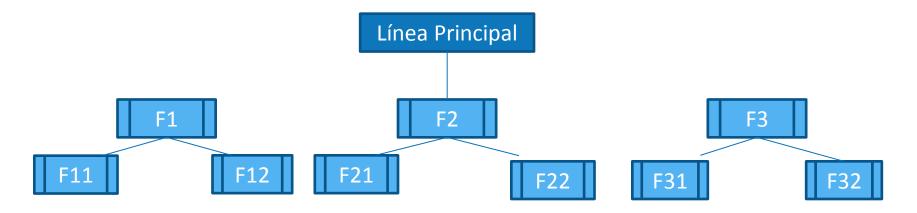
El programador diseña la estructura de datos y el tipo de operaciones que acepta dicha estructura. Al conjunto de estructura de dato y operación se le llama objeto. El programa ya no es una lista de acciones de orden fijo si no un conjunto de objetos (datos y posibles operaciones que puede realizar) que cooperan entre ellos para conseguir un objetivo. Java, C, C++, PHP, C# y Visual Basic.





## Programación modular

Consiste en dividir el programa en una línea principal y varias funciones

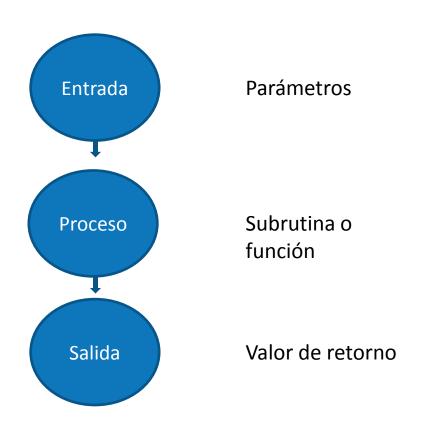


- La línea principal controla la lógica
- Uso de break para romper un secuencia y loops
- En cada función utilizar la programación estructurada





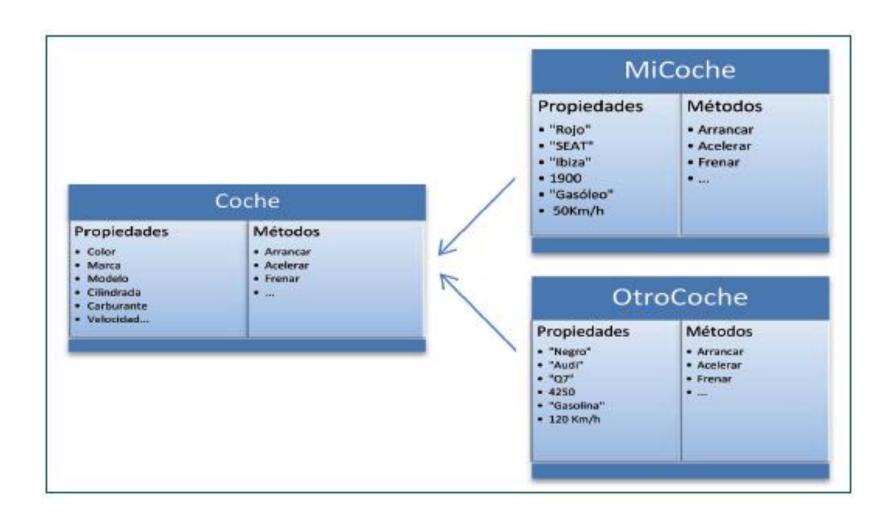
# Programación estructurada







### **POO**







### Características de los lenguajes más difundidos

#### JAVA:

- Lenguaje orientado a objetos.
- Más sencillo de manejar que otros lenguajes orientados a objetos.
- Manejo de computación distribuida (varios ordenadores trabajando juntos en una red).
- Portabilidad (la Máquina Virtual de Java permite que un mismo programa pueda ser utilizado en diferentes ordenadores con distintos sistemas operativos).
- Buenas perspectivas profesionales.

### <u>C</u>

- Lenguaje estructurado.
- Reducido número de comandos.
- Existencia de compiladores para casi cualquier ordenador.
- Larga presencia en el tiempo.





### Características de los lenguajes más difundidos

#### **C++**

- Lenguaje orientado a objetos.
- Concisión: utiliza muchos caracteres especiales en vez de comandos, por lo que un programador codifica a mucha velocidad.
- Portabilidad: para compilar en casi todos los ordenadores y sistemas operativos.
- Programación modular: varias fuentes son compiladas independientemente para luego enlazarlas.
- Compatibilidad: código escrito en C puede ser incluido con facilidad en C++.
- Eficiencia en la ejecución de programas

#### PHP

- Lenguaje de script (permite el control de una o más aplicaciones).
- Utilizado en el desarrollo web (Ámbito del Servidor).
- Práctico: su evolución se orientó hacia la utilidad de cara al usuario.
- Muy bajo coste económico.
- Páginas web dinámicas (uso embebido en código HTML).





# Características de los lenguajes más difundidos

#### C#

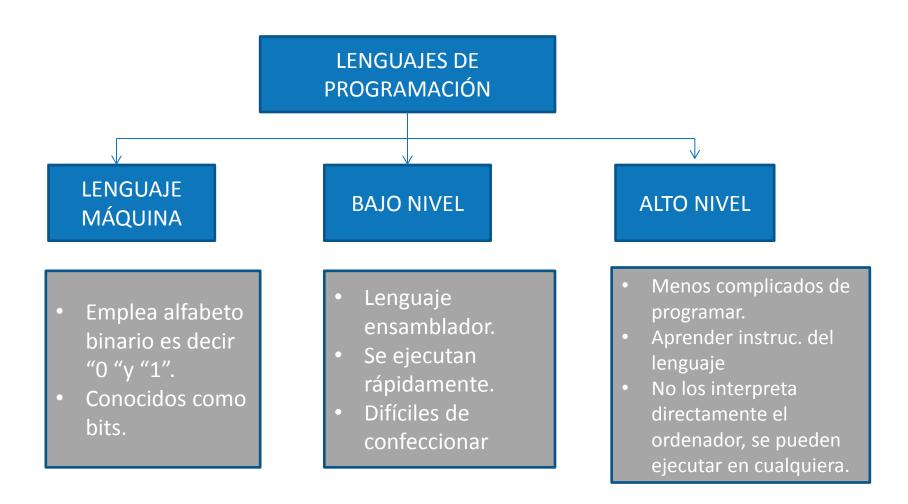
- Lenguaje orientado a objetos.
- Incorpora características de otros lenguajes (Java, Visual Basic).
- Forma parte de la plataforma .NET de Microsoft.
- Su sintaxis deriva de C y C++.

### **Visual Basic**

- Lenguaje orientado a objetos.
- Fácil de utilizar.
- Deriva del BASIC de procedimiento.
- Forma parte de la plataforma .NET de Microsoft.











Clasificación basada en la forma de trabajar de los programas y la filosofía de su creación es la siguiente:

- Lenguajes de programación imperativos: entre ellos tenemos el Cobol, Pascal, C y Ada.
- Lenguajes de programación declarativos: el Lisp y el Prolog.
- Lenguajes de programación orientados a objetos: el Smalltalk y el C++.
- Lenguajes de programación orientados al problema: son aquellos lenguajes específicos para gestión.
- Lenguajes de programación naturales: son los nuevos lenguajes que pretender aproximar el diseño y la construcción de programas al lenguaje de las personas.





Clasificación teniendo en cuenta el desarrollo de los ordenadores según sus diferentes generaciones:

- Lenguajes de programación de primera generación: el lenguaje máquina y el ensamblador.
- Lenguajes de programación de segunda generación : los primeros lenguajes de programación de alto nivel imperativo (FORTRAN, COBOL).
- Lenguajes de programación de tercera generación: son lenguajes de programación de alto nivel imperativo pero mucho más utilizados y vigentes en la actualidad (ALGOL 8, PL/I, PASCAL, MODULA).
- Lenguajes de programación de cuarta generación: usados en aplicaciones de gestión y manejo de bases de dados (NATURAL, SQL).
- Lenguajes de programación de quinta generación: creados para la inteligencia artificial y para el procesamiento de lenguajes naturales (LISP, PROLOG).





## Fases de Desarrollo de una Aplicación

- Planificación del proyecto. Estudio de viabilidad.
  - Documento del presupuesto. Después de la planificación, el responsable del proyecto debe valorarlo económicamente.
  - Firma del documento de iniciación del proyecto.
  - Análisis del sistema y recogida de requerimientos
  - Diseño del sistema.
- Codificación
- Pruebas técnicas
- Pruebas de usuario.
- Documentación.
- Implementación en entorno de explotación (Real).
- Plan de mantenimiento





### **PRINCE2**

- Un Entorno de Desarrollo debe encajar dentro de una metodología de trabajo.
- PRINCE2(Projects IN Controlled Environments o Proyectos en Entornos Controlados) es una buena metodología de gestión de proyectos con certificación de calidad.



https://es.slideshare.net/evergreenpm/acis-xii-jornadasprince2





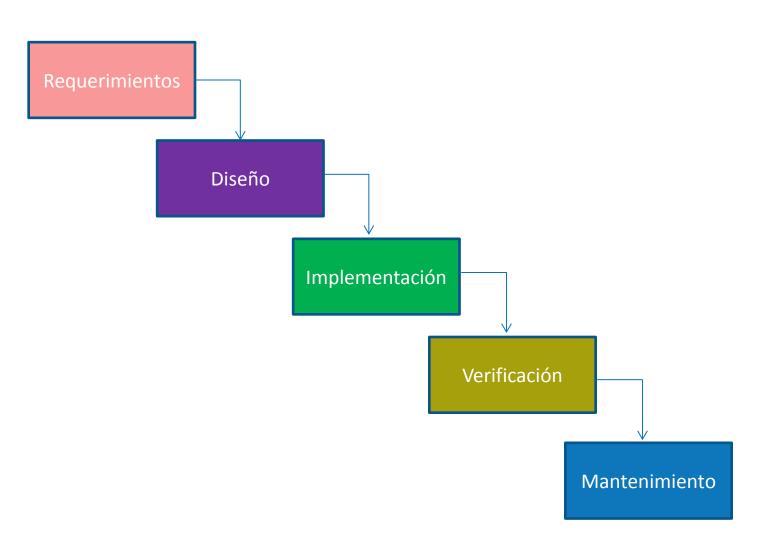
# Diferencia entre Metodología ágil y tradicional

| Metodologías Agiles  | Metodologías Tradicionales   |
|--|--|
| Especialmente preparados para cambios durante el proyecto        | Cierta resistencia a los cambios                                       |
| Impuestas internamente(por el equipo)                            | Impuestas externamente   |
| Proceso menos controlado, con pocos principios                   | Proceso mucho más controlado, con numerosas políticas/normas           |
| No existe contrato tradicional o es más flexible si existe       | Existe un contrato prefijado   |
| El cliente es parte del equipo de desarrollo                     | El cliente interactúa con el equipo de desarrollo mediante reuniones   |
| Grupos pequeños (<10 integrantes) y trabajando en el mismo sitio | Grupos grandes y posiblemente distribuidos                             |
| Pocos roles  | Más roles  |
| Menos énfasis en la arquitectura del software                    | La arquitectura del software es esencial y se expresa mediante modelos |





# Metodología tradicional (waterfall-en cascada)







### Metodología tradicional (waterfall-en cascada)

#### Ventajas:

- Simple y fácil de entender
- Aproximación "disciplina"
  - Definición exhaustiva del trabajo
  - Revisión sistemática en hitos
  - Énfasis en control y documentación
- Nivel de incertidumbre del proyecto bajo.

### Desventajas

- Cambios más caros cuanto más tarde se descubren
- Riesgo para cliente y proveedor
  - Seguimiento sobre calendario engañoso: "Efecto 90%"
  - Validación al final del proyecto: funcionalidades y arquitectura
  - Fomenta la "deuda técnica





# Metodología Ágil (SCRUM)

### Agilidad es:

- Una alternativa al desarrollo secuencial tradicional
- Concebida para ayudar en la respuesta a lo imprevisto
- Incremental, trabajo interactivo con retroalimentación empírica
- Focalizada en incrementos potenciales de productos entregables en cortos periodos de tiempo
- Revisando contantemente cada aspecto del desarrollo a través del ciclo de vida
- Continuamente re-evaluando la dirección con las posibilidades de cambio
- Nivel de incertidumbre del proyecto alto



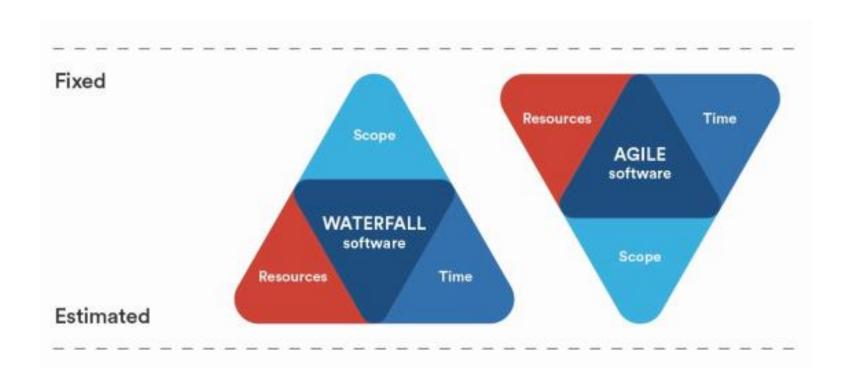








# Metodologías agiles





La mejor forma de aprender es "haciendo". Educación y empresas forman un binomio inseparable