

# FUNDAMENTOS DE SOFTWARE DESARROLLO O IMPLEMENTACIÓN

## DESARROLLO O IMPLEMENTACIÓN

La fase de **desarrollo o implementación**, es donde el diseño se traduce en un sistema funcional. En esta etapa, los desarrolladores codifican, prueban y ensamblan los componentes del **software**, utilizando los lineamientos definidos en las fases previas. Es el momento de dar vida al proyecto.

La fase de desarrollo o implementación, consiste en transformar el diseño del **software** en código ejecutable. Utilizando lenguajes de programación, **frameworks** y herramientas de desarrollo, se construyen las funcionalidades especificadas en los requisitos.

### Actividades principales de la fase de desarrollo

- **Codificación.** El equipo de desarrollo convierte el diseño en código fuente utilizando el lenguaje y las tecnologías seleccionadas. Durante esta actividad, se enfocarán en:

- o **Modularidad:** dividir el código en componentes independientes.
- o **Calidad:** asegurar que el código sea limpio, documentado y reutilizable.

**Ejemplo práctico:** en un sistema de gestión de inventarios, se desarrollan módulos específicos como:

- o Gestión de productos (crear, actualizar, eliminar).
- o Generación de reportes.
- o Integración con sistemas externos.

- **Integración.** Los módulos individuales se ensamblan para formar un sistema completo. Es importante verificar que los componentes trabajen juntos sin problemas.

**Ejemplo práctico:** en una aplicación bancaria, se integran los módulos de autenticación, consulta de saldo y transferencias, para garantizar un flujo de usuario continuo.

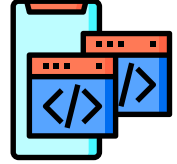
- **Pruebas unitarias.** Durante el **desarrollo**, se realizan pruebas unitarias para validar que cada módulo funcione correctamente, de manera aislada.





**Ejemplo práctico:** en un módulo de cálculo de intereses bancarios, se realizan pruebas con valores específicos para confirmar que los resultados sean precisos.

- **Documentación.** La documentación del código y las funcionalidades son esenciales para el mantenimiento futuro del sistema. Incluye comentarios en el código, manuales técnicos, y guías para desarrolladores.



## Herramientas utilizadas en la fase de desarrollo

En esta etapa, las herramientas de desarrollo desempeñan un papel crucial para garantizar la productividad, la calidad del código y la colaboración entre los equipos.

- **Entornos de desarrollo integrados (IDE).** Son herramientas que facilitan la escritura, depuración y prueba del código.

### Ejemplos:

- o **Visual Studio Code:** popular por su flexibilidad y extensiones para múltiples lenguajes.
- o **Eclipse:** ideal para proyectos en Java.
- o **IntelliJ IDEA:** recomendado para desarrollo en Java y Kotlin.

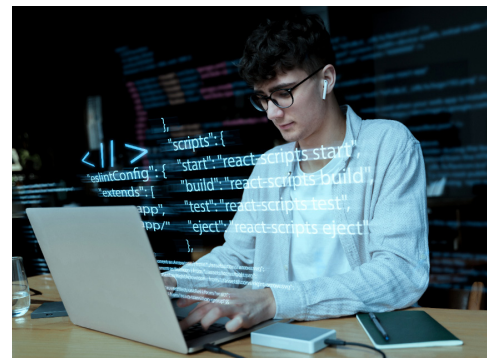
- **Sistemas de control de versiones.** Permiten a los desarrolladores gestionar cambios en el código y colaborar de manera eficiente.

### Ejemplos:

- o **Git:** utilizado junto con plataformas como GitHub, GitLab o Bitbucket.
- o **Subversion (SVN):** aunque menos común, sigue siendo útil en proyectos específicos.
- **Marcos de desarrollo.** Estructuras predefinidas que aceleran el desarrollo.

### Ejemplos:

- o **Django para Python:** desarrollo rápido de aplicaciones web.
- o **Laravel para PHP:** creación de sistemas web robustos y seguros.
- o **Spring para Java:** ideal para aplicaciones empresariales.
- **Herramientas de construcción y automatización.** Facilitan tareas repetitivas como la compilación, el empaquetado y la ejecución de pruebas.



### Ejemplos:

- o **Maven:** gestión de proyectos en Java.
- o **Gradle:** automatización de tareas en múltiples lenguajes.
- o **Jenkins:** automatización de procesos de integración continua.
- **Herramientas de pruebas unitarias.** Permiten a los desarrolladores validar el correcto funcionamiento de sus módulos.

### Ejemplos:

- o **JUnit:** marco de pruebas para Java.
- o **PyTest:** ideal para pruebas en Python.
- o **xUnit:** compatible con proyectos en .NET.

## Artefactos generados en la fase de desarrollo

Durante esta etapa, se producen varios artefactos que son fundamentales para el desarrollo y la posterior implementación del sistema.

- **Código fuente.** Es el producto principal de esta etapa. El código debe estar estructurado, documentado y optimizado, para facilitar su mantenimiento.

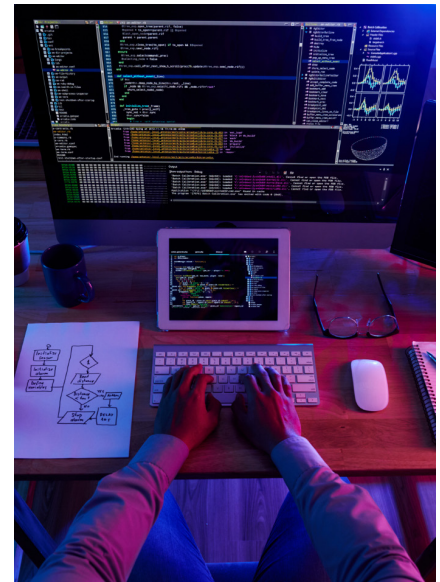
**Ejemplo práctico:** en un sistema de comercio electrónico, el código incluye funciones como:

- o Filtrar productos por categoría.
- o Pagos de procesadores.
- o Gestionar inventarios.

- **Bibliotecas y componentes reutilizables.** Se desarrollan módulos que pueden ser reutilizados en otros proyectos.

**Ejemplo práctico:** un módulo de autenticación que maneje el inicio de sesión, mediante redes sociales.

- **Manual técnico.** Se documenta el diseño del sistema y cómo los desarrolladores pueden trabajar con él. Incluye:
  - o Estructura de archivos.



- o Detalles sobre cómo configurar y ejecutar el sistema.

- o Explicación de claves de algoritmos.

- **Casos de prueba.** Se generan scripts y documentos que especifican cómo se deben probar los módulos del sistema.

**Ejemplo práctico:** un caso de prueba para el módulo de “Carrito de compras” podría incluir:

- o Verificar que los productos se añadan correctamente al carrito.

- o Comprobar que los precios totales se calculan de manera precisa.