



FUNDAMENTOS DE PROGRAMACIÓN

**ENTORNOS DE DESARROLLO**

## ENTORNOS DE DESARROLLO

Los entornos de desarrollo son herramientas esenciales en la programación, ya que proporcionan un espacio estructurado para escribir, probar y depurar código. Estos entornos permiten a los desarrolladores mejorar su productividad y eficiencia al contar con herramientas integradas como editores de texto, depuradores y sistemas de control de versiones. En esta temática, se explicarán los diferentes tipos de entornos de desarrollo, sus componentes fundamentales y su impacto en la industria del software. (Casado Iglesias, 2015)

### Definición y propósito de un entorno de desarrollo

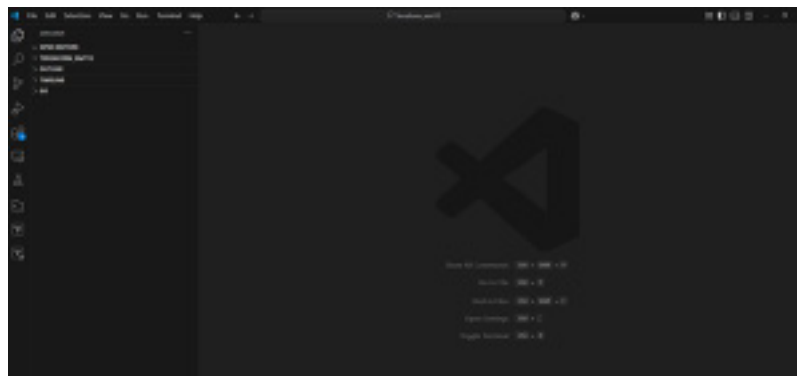
Un entorno de desarrollo es un conjunto de herramientas y configuraciones diseñadas para facilitar la creación de software. Su propósito es optimizar el flujo de trabajo del programador, brindándole un espacio adecuado para la escritura, compilación y ejecución del código. (Casado Iglesias, 2015)

#### ✓ Entornos de desarrollo locales

Estos entornos están instalados directamente en la computadora del programador y permiten desarrollar software sin necesidad de conexión a internet. Algunos ejemplos incluyen:

#### ✓ **Visual Studio Code:** Un editor de código ligero pero potente.

Figura 1. Visual Studio Code



#### ✓ **IntelliJ IDEA:** Un IDE optimizado para Java y otros lenguajes.

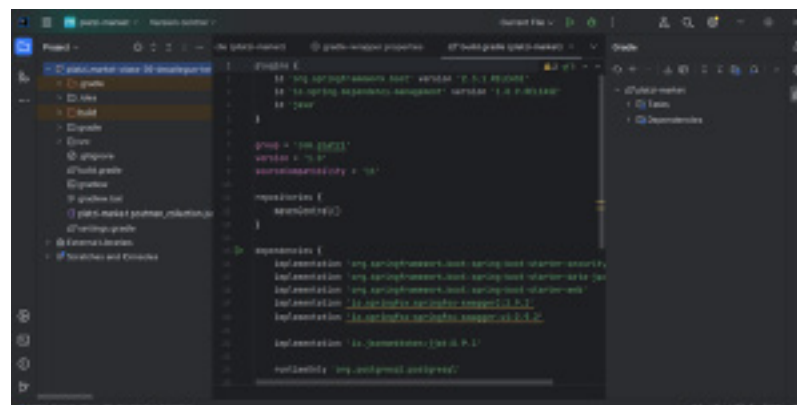


Imagen 5 - IntelliJ Entorno - Fuente: A, Fernando. 2025

✓ **Eclipse:** IDE ampliamente utilizado en la comunidad Java.

### ✓ **Entornos de Desarrollo en la Nube**

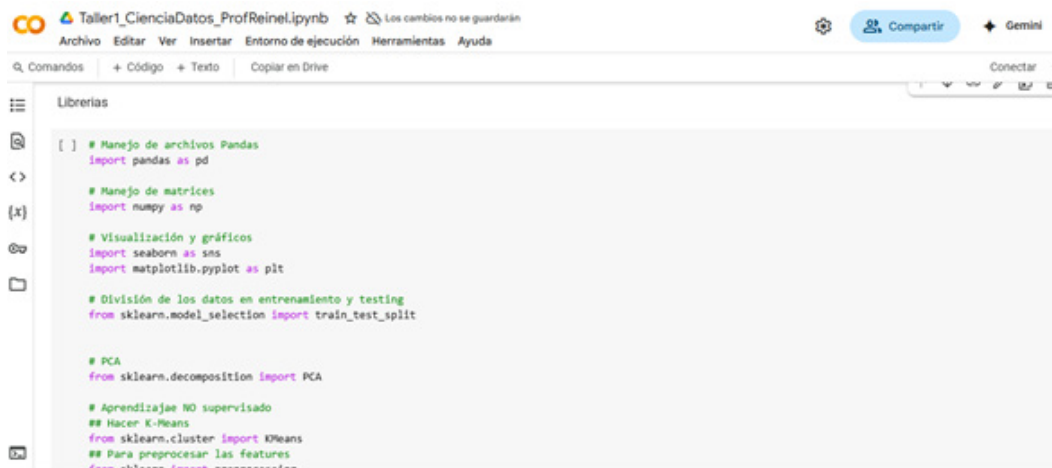
Con el avance de la tecnología, muchas herramientas han migrado a la nube, permitiendo el acceso desde cualquier lugar sin necesidad de instalación local. Algunos ejemplos destacados son:

✓ **AWS Cloud9:** Un entorno basado en la web con soporte para múltiples lenguajes de programación.

✓ **GitHub Codespaces:** Permite desarrollar directamente desde GitHub con una máquina virtual en la nube.

✓ **Google Colab:** Muy utilizado en inteligencia artificial y ciencia de datos.

Figura 2. Google Colab



```
Taller1_CienciaDatos_ProfReinel.ipynb  Los cambios no se guardarán
Archivo  Editar  Ver  Insertar  Entorno de ejecución  Herramientas  Ayuda

Comandos  + Código  + Texto  Copiar en Drive  Conectar

Librerías

[ ] # Manejo de archivos Pandas
import pandas as pd

# Manejo de matrices
import numpy as np

# Visualización y gráficos
import seaborn as sns
import matplotlib.pyplot as plt

# División de los datos en entrenamiento y testing
from sklearn.model_selection import train_test_split

# PCA
from sklearn.decomposition import PCA

# Aprendizaje NO supervisado
## Hacer K-Means
from sklearn.cluster import KMeans
## Para preprocesar las features
from sklearn.feature_extraction import TfidfVectorizer
```

## Entornos de desarrollo virtualizados y contenedorizados

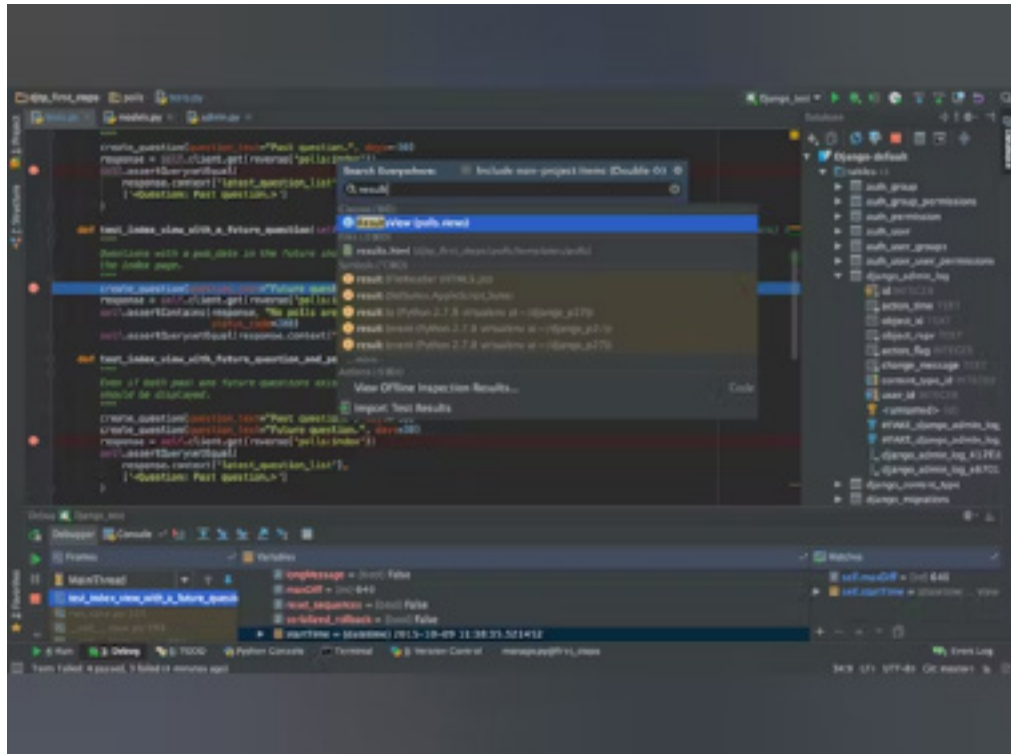
El uso de máquinas virtuales y contenedores ha permitido una mayor flexibilidad en el desarrollo de software. Tecnologías como Docker y Vagrant facilitan la creación de entornos replicables y escalables. (Casado Iglesias, 2015)

## Editores de Código y IDEs

Los entornos de desarrollo integrados (IDEs) ofrecen herramientas avanzadas para la edición de código, depuración y compilación. Algunos IDEs populares incluyen:

- ✓ **JetBrains PyCharm:** Ideal para desarrollo en Python.

Figura 3. Pycharm



- ✓ **Microsoft Visual Studio:** Potente para desarrollo en .NET y C++.

Figura 4. Inicio de Microsoft Visual Studio



Figura 5. Entorno Microsoft visual studio



## Compiladores e intérpretes

Dependiendo del lenguaje de programación, el código puede requerir un compilador o un intérprete:

- ✓ **Compiladores:** Transforman el código fuente en código máquina (Ejemplo: GCC para C/C++, Java Compiler para Java).
- ✓ **Intérpretes:** Ejecutan el código línea por línea sin necesidad de compilación previa (Ejemplo: Python Interpreter, Node.js para JavaScript).

## Control de Versiones

Los sistemas de control de versiones permiten gestionar cambios en el código y colaborar con otros desarrolladores. Herramientas clave:

- ✓ **Git:** El sistema de control de versiones más utilizado.
- ✓ **GitHub/GitLab/Bitbucket:** Plataformas para alojar repositorios de código.

## Depuradores

El depurado es esencial para identificar y corregir errores en el código. Los IDEs suelen incluir herramientas integradas para este propósito, como:

- ✓ **Debugger de Visual Studio Code**
- ✓ **Depurador de IntelliJ IDEA**

## Configuración y Personalización

Cada desarrollador puede ajustar su entorno según sus necesidades:

- ✓ **Instalación de extensiones y plugins.**
- ✓ **Configuración de temas y atajos de teclado.**



☑ **Uso de herramientas de automatización y linters.**

## **Impacto de los entornos de desarrollo en la industria del software**

El uso de entornos de desarrollo eficientes permite a las empresas entregar software de mayor calidad en menos tiempo. Algunas tendencias recientes incluyen:

- Desarrollo colaborativo en la nube.
- Automatización de entornos con infraestructura como código.
- Integración de IA en IDEs para mejorar la productividad.

## **Bibliografía**

Casado Iglesias, C. (2015). Entornos de desarrollo. RA-MA Editorial.  
<https://elibro.net/es/lc/tecnologicadeloriente/titulos/62495>