

Práctica 2: Instalación y uso de entornos de desarrollo

Instrucciones para Elegir las propuestas:

De las dos propuestas 1, 2 que tienes debajo, elije una de las siguientes opciones para completarla:

Propuesta 1: Configuración de Eclipse para Python y creación de una calculadora de conversión de monedas (USD y RUB a EUR) en Python.

Propuesta 2: Configuración de Eclipse para C++ y creación de una calculadora de conversión de monedas (USD y RUB a EUR) en C++.

Entregas:

Evidencia de Configuración del Entorno

Para verificar que realmente han configurado Eclipse correctamente, puedes necesito:

- **Capturas de pantalla** de la configuración del entorno:
 - En Python: capturas del menú de configuración de PyDev y de la configuración del intérprete de Python en Eclipse.
 - En C++: capturas de la configuración de CDT y de los ajustes del compilador de C++.
- Un **informe breve** (de 1 o 2 párrafos) donde expliques los pasos que siguistes para configurar Eclipse y cualquier problema que haya encontrado durante la instalación y configuración.

Unifica en un documento en PDF todo

Que tienen que hacer:

Propuesta 1: Desarrollo en Python con Eclipse (usando PyDev)

Objetivo:

Configurar Eclipse para desarrollar en Python y crear una aplicación que convierta dólares y rublos a euros.

Pasos:

Configurar Eclipse para Python:

Instalar PyDev:

Práctica 2: Instalación y uso de entornos de desarrollo

Abre Eclipse y ve a Help > Eclipse Marketplace, busca "PyDev" y haz clic en Go.

Luego instalar PyDev.

Configurar Python Interpreter: Asegúrate de que Eclipse tenga configurado el intérprete de Python:

Ve a Window > Preferences > PyDev > Interpreters > Python Interpreter y configura la ubicación de tu instalación de Python.

Prueba su funcionamiento

Crea un archivo Python conversor_moneda.py que permita ingresar una cantidad en dólares o rublos y convierta el valor a euros. Las tasas de cambio son las siguientes:

1 USD = 0.92 EUR

1 RUB = 0.010 EUR

El programa debe pedir al usuario que ingrese el valor y la moneda (USD o RUB), luego debe mostrar el resultado en euros.

Exploración de Menús:

Una vez configurado PyDev, en los menús de Eclipse, bajo PyDev, encontrarás opciones específicas para ejecutar y depurar programas Python. Por ejemplo, en el menú Run verás Run As > Python Run y Debug As > Python Debug.

También tendrás opciones para configurar y gestionar entornos virtuales de Python, ver el árbol de archivos y gestionar dependencias.

Propuesta 2: Desarrollo en C++ con Eclipse (usando CDT)

Objetivo:

Configurar Eclipse para el desarrollo en C++ y crear una calculadora de conversión de monedas de dólares y rublos a euros.

Pasos:

Configurar Eclipse para C++ (CDT):

Instalar CDT (C/C++ Development Tools): Ve a Help > Eclipse Marketplace, busca "CDT" y haz clic en Go para instalar el plugin para C/C++.

Configurar el compilador C++: Asegúrate de tener instalado un compilador como MinGW en Windows o GCC en Linux/Mac. Configura Eclipse para usarlo en Window > Preferences > C/C++ > Build > Settings.

Prueba su funcionamiento

Crea un archivo C++ conversor_moneda.cpp que permita ingresar una cantidad en dólares o rublos y la convierta a euros. Las tasas de cambio son las siguientes:

1 USD = 0.92 EUR

1 RUB = 0.010 EUR

Práctica 2: Instalación y uso de entornos de desarrollo

El programa debe pedir al usuario que ingrese el valor y la moneda (USD o RUB), luego debe mostrar el resultado en euros.

Exploración de Menús:

Una vez configurado el CDT, en el menú Project encontrarás opciones específicas como Build Project, Clean, y Build Configurations para gestionar el proceso de compilación en C++.

En el menú Run, verás opciones como Run As > Local C/C++ Application, además de herramientas de depuración específicas para C++ como Debug As > Local C/C++ Application.

Orientación de cómo puede ser el fichero en PYTHON

```
def convertir_a_euros(cantidad, moneda):
    tasas = {"USD": 0.92, "RUB": 0.010}
    if moneda in tasas:
        return cantidad * tasas[moneda]
    else:
        return None

# Entrada del usuario
cantidad = float(input("Ingresa la cantidad: "))
moneda = input("Ingresa la moneda (USD o RUB): ").upper()

# Conversión
resultado = convertir_a_euros(cantidad, moneda)
if resultado is not None:
    print(f"{cantidad} {moneda} equivale a {resultado:.2f} EUR.")
else:
    print("Moneda no soportada. Por favor, usa USD o RUB.")
```

Orientación de cómo puede ser el fichero en C++

```
#include <iostream>
#include <string>
using namespace std;

float convertirAEuros(float cantidad, string moneda) {
    if (moneda == "USD") {
        return cantidad * 0.92;
    } else if (moneda == "RUB") {
        return cantidad * 0.010;
    } else {
        return -1; // Indicador de error
    }
}

int main() {
```

Práctica 2: Instalación y uso de entornos de desarrollo

```
float cantidad;  
string moneda;  
  
// Entrada del usuario  
cout << "Ingresa la cantidad: ";  
cin >> cantidad;  
cout << "Ingresa la moneda (USD o RUB): ";  
cin >> moneda;  
  
// Conversión  
float resultado = convertirAEuros(cantidad, moneda);  
if (resultado != -1) {  
    cout << cantidad << " " << moneda << " equivale a " <<  
resultado << " EUR." << endl;  
} else {  
    cout << "Moneda no soportada. Por favor, usa USD o RUB." <<  
endl;  
}  
  
return 0;  
}
```