

Configuración, instalación y administración del SDK

Autores

Alexis Chimba, Javier Ramos, Ariel Reyes, Anthony Villarreal

Ingeniera Doris Chicaiza

Universidad de las Fuerzas Armadas - ESPE

aschimba@espe.edu.ec, sjramos@espe.edu.ec, alreyes2@espe.edu.ec,
anvillarreal@espe.edu.ec.

Resumen

En este laboratorio se configuró el entorno de desarrollo para trabajar con Dart SDK y se desarrollaron dos programas básicos. Como primer paso, se utilizó PowerShell para instalar Chocolatey, un gestor de paquetes que facilitó la instalación de Dart mediante el comando `choco install dart-sdk`. Durante la instalación, se aceptaron automáticamente todas las condiciones requeridas y, posteriormente, se agregó la ruta del SDK (`C:\tools\dart-sdk`) a las variables de entorno del sistema para garantizar su correcto funcionamiento. Después de configurar el entorno, se desarrollaron dos programas usando Dart. El primero calcula el costo del consumo eléctrico en función de los kilowatts consumidos y una tarifa fija. El segundo determina el precio final de un producto aplicando un 20% de descuento y un 15% de IVA. Las pruebas se realizaron en la terminal (CMD) para verificar su correcta ejecución. Además, se instalaron las extensiones de Dart y Flutter desde el apartado de Plugins en Android Studio. Finalmente, se comprobó que el SDK de Dart estuviera correctamente vinculado al crear un nuevo proyecto de Flutter, asegurando que el entorno de desarrollo funcionara adecuadamente.

Palabras clave - Dart SDK, Chocolatey, Android Studio.

I. Introducción

Configurar un entorno de desarrollo adecuado es un paso esencial para cualquier programador que quiera comenzar a crear aplicaciones de forma eficiente. En este laboratorio nos enfocamos en instalar y preparar el SDK de Dart, un lenguaje de programación creado por Google, muy utilizado hoy en día junto con Flutter para desarrollar aplicaciones multiplataforma. Para lograrlo, utilizamos herramientas como PowerShell y Chocolatey, que simplificaron el proceso de instalación. Además, configuramos correctamente las variables de entorno y vinculamos Dart con Android Studio para facilitar el trabajo futuro. Como parte de la práctica, desarrollamos dos pequeños programas: uno para calcular el costo de un consumo eléctrico y otro para obtener el precio final de un producto aplicando descuento e IVA. Esta experiencia no solo ayudó a entender mejor la instalación y configuración de herramientas, sino que también fortaleció habilidades de programación y manejo de entornos de trabajo reales.

II. Trabajos Relacionados

El proceso de configuración de entornos de desarrollo como Dart SDK es fundamental en proyectos de desarrollo de aplicaciones móviles y web. Trabajos previos han abordado la instalación de Flutter, la gestión de dependencias mediante Chocolatey, y la configuración de IDEs como Android Studio para optimizar el flujo de trabajo. Además, la creación de programas básicos para cálculos matemáticos es una práctica común en cursos de introducción a la programación, ya que refuerza la lógica algorítmica y el uso correcto de variables y operaciones.

III. Materiales y Métodos

Materiales.

- Descargar e instalar Dart SDK
- Sistema Operativo Windows 11
- Dart SDK versión 3.x.x
- Editor de texto: Android Studio Code o Visual Studio Code.
- Terminal CMD o PowerShell

IV. Desarrollo

1. Descargar chocolate para instalarlo en PowerShell.
2. Descargar Dart por medio de Chocolatey con el siguiente comando: `choco install dart-sdk`
3. Aceptar todo durante el proceso de instalación.
4. Agregar la ruta de Dart en las variables de entorno del sistema (por defecto suele ser: `C:\tools\dart-sdk`).
5. Descargar las extensiones de Dart y Flutter en Android Studio desde el apartado de Plugins.

6. Verificar que el SDK de Dart sea visible al crear un nuevo proyecto de Flutter (entrar al apartado de Dart, colocar la dirección del SDK en el sistema y comprobar si aparecen las opciones correspondientes).

V. Resultados

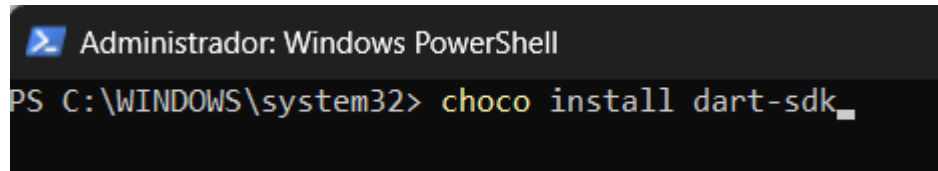


Figura 1. Comando de instalación de dart por medio de Power Shell

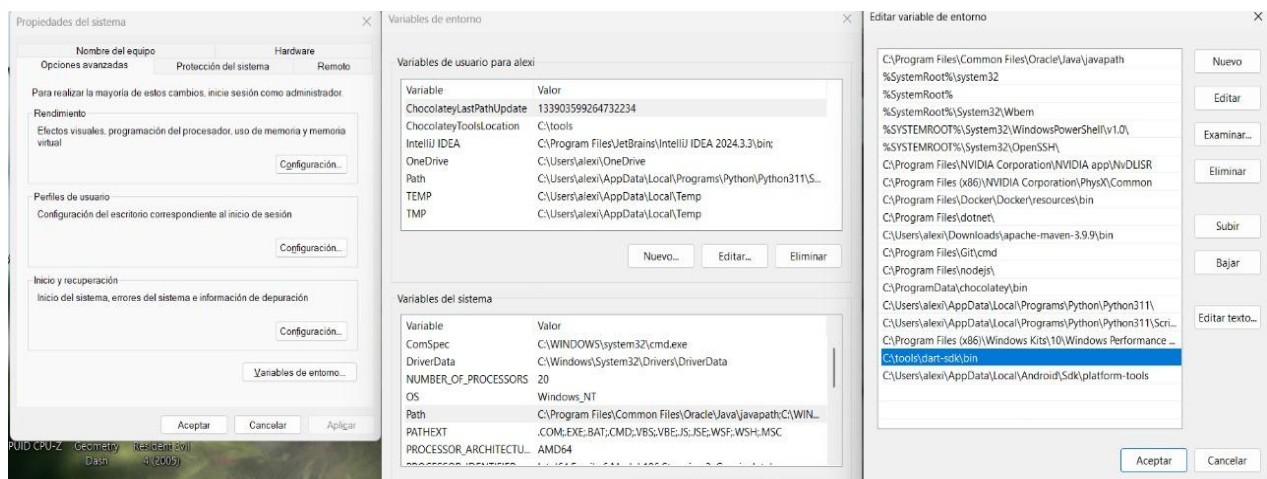


Figura 2. Inserción de la dirección de dart-sdk en las variables de entorno

```

main.dart x
C:\Users\alexi\Downloads> main.dart
1 import 'dart:io';
2 void main() {
3   print('--- Cálculo de Consumo de Energía Eléctrica (CLS) ---');
4   print('Tarifa por KW: \$0.85');
5   print('Ingrese el consumo en kilowatts (KW):');
6   // Leemos la entrada del usuario desde la consola.
7   // stdin es parte de dart:io y permite leer datos del teclado.
8   // El operador '!' (non-null assertion) asegura que la entrada no es nula, ya que confiamos en que el usuario ingresará algo.
9   double consumoKW = double.parse(stdin.readLineSync()!);
10  double tarifaPorKW = 0.85;
11  double costoConsumo = consumoKW * tarifaPorKW;
12  print('\nResultados del Consumo:');
13  print('Consumo ingresado: $consumoKW KW');
14  print('Tarifa aplicada por KW: \${tarifaPorKW.toStringAsFixed(2)}');
15  print('Costo total: \${costoConsumo.toStringAsFixed(2)}');
16  print('Cálculo verificado: Consumo ($consumoKW KW) x Tarifa (\${tarifaPorKW.toStringAsFixed(2)}) = \${costoConsumo.toStringAsFixed(2)}');
17  print('\n--- Cálculo de Precio de un Artículo ---');
18  print('Descuento: 20%');
19  print('IVA: 15%');
20  print('Ingrese el precio original del artículo:');
21  double precioOriginal = double.parse(stdin.readLineSync()!);
22  double porcentajeDescuento = 20.0;
23  double porcentajeIVA = 15.0;
24  double montoDescuento = precioOriginal * (porcentajeDescuento / 100);
25  double precioConDescuento = precioOriginal - montoDescuento;
26  double montoIVA = precioConDescuento * (porcentajeIVA / 100);
27  double precioFinal = precioConDescuento + montoIVA;
28  print('\nResultados del Artículo:');
29  print('Precio original: \${precioOriginal.toStringAsFixed(2)}');
30  print('Descuento (20%): \${montoDescuento.toStringAsFixed(2)}');
31  print('Precio con descuento: \${precioConDescuento.toStringAsFixed(2)}');
32  print('IVA (15%): \${montoIVA.toStringAsFixed(2)}');
33  print('Precio final: \${precioFinal.toStringAsFixed(2)}');
34  print('Cálculo verificado: Precio con descuento (\${precioConDescuento.toStringAsFixed(2)}) + IVA (\${montoIVA.toStringAsFixed(2)}) = \${precioFinal.toStringAsFixed(2)}');
35 }

```

Figura 3. Código de cálculo de precio de kilowatts y precio con descuento e iva

```

--- Cálculo de Consumo de Energía Eléctrica (CLS) ---
Tarifa por KW: $0.85
Ingrese el consumo en kilowatts (KW):
12

Resultados del Consumo:
Consumo ingresado: 12.0 KW
Tarifa aplicada por KW: $0.85
Costo total: $10.20
Cálculo verificado: Consumo (12.0 KW) x Tarifa ($0.85) = $10.20

--- Cálculo de Precio de un Artículo ---
Descuento: 20%
IVA: 15%
Ingrese el precio original del artículo:
56

Resultados del Artículo:
Precio original: $56.00
Descuento (20%): $11.20
Precio con descuento: $44.80
IVA (15%): $6.72
Precio final: $51.52
Cálculo verificado: Precio con descuento ($44.80) + IVA ($6.72) = $51.52

Proceso terminado con código de salida 0

```

Figura 4. Resultado del código de dart en consola.

VII. Conclusiones

- La instalación de Dart SDK mediante Chocolatey y su integración con Android Studio fue exitosa, permitiendo un entorno de desarrollo funcional.
- Se reforzaron habilidades en configuración de entornos, manejo de variables del sistema y programación básica en Dart.
- Para futuros trabajos, se recomienda seguir documentando el proceso y explorar herramientas avanzadas como Flutter para ampliar conocimientos.

Referencias

- [1] Dart SDK, Dart Programming Language, Recuperado de <https://dart.dev/get-dart>.
- [2] Chocolatey Software, Installing Chocolatey, Recuperado de <https://chocolatey.org/install>.