

Lección: Creando una Aplicación Móvil Simple Usando Dart

Objetivos de la Lección:

- Comprender los conceptos básicos de Dart para desarrollar una aplicación móvil sin frameworks.
- Escribir un código estructurado en Dart para crear una aplicación móvil paso a paso.
- Desarrollar una aplicación que muestre una interfaz gráfica básica.

Introducción de la Lección

Dart es un lenguaje de programación desarrollado por Google, conocido por su uso en la creación de aplicaciones móviles. Aunque normalmente se utiliza con Flutter, en esta lección trabajaremos con Dart de manera independiente. Aprenderás a crear una aplicación móvil simple que muestre un mensaje de saludo en pantalla.

Desarrollo del Tema Central

Vamos a construir una aplicación sencilla que muestre "¡Hola, Mundo!" en la pantalla. Para ello, utilizaremos solo Dart, sin ningún framework como Flutter.

Pasos para Crear la Aplicación:

1. Configuración Inicial

- Asegúrate de que las herramientas de desarrollo ya están configuradas y funcionando correctamente.
- Si no lo has hecho, abre tu navegador web y accede al sitio web "[TutorialsPoint Execute Dart Code](#)" y crea un nuevo archivo Dart, por ejemplo, `main.dart`.

2. Estructura Básica de una Aplicación en Dart

Una aplicación móvil simple en Dart necesita algunos componentes clave: la entrada, la lógica, y la salida visual.

Comienza creando la estructura básica del programa:

```
void main() {  
    // Aquí comienza tu aplicación  
    print('¡Hola, Mundo!');  
}
```

Este código es un buen punto de partida, ya que nos permite ver si el entorno está funcionando correctamente. Ejecútalo para asegurarte de que se muestra el mensaje en la consola.

3. Crear una Clase para la Interfaz de Usuario

Vamos a mejorar el código para que muestre algo en una pantalla móvil. Creamos una clase que controle la interfaz gráfica:

```
class MiAplicacion {  
    void ejecutar() {  
        print('Ejecutando la aplicación...');  
        mostrarPantalla();  
    }  
  
    void mostrarPantalla() {  
        print('Pantalla: ¡Hola, Mundo!');  
    }  
}
```

Este es un ejemplo básico de cómo estructurar tu código en Dart. La clase `MiAplicacion` contiene dos métodos:

- `ejecutar`: Inicia la aplicación.
- `mostrarPantalla`: Simula mostrar algo en la pantalla, por ahora con un `print`.

4. Ejecutar la Aplicación

Ahora, modifica el método `main` para crear y ejecutar la aplicación:

```
void main() {  
    MiAplicacion app = MiAplicacion();  
    app.ejecutar();  
}
```

Este código instancia la clase `MiAplicacion` y llama al método `ejecutar` para comenzar la ejecución.

5. Implementar Interacción con el Usuario

Para hacer la aplicación más interactiva, vamos a permitir que el usuario introduzca su nombre, y que la aplicación muestre un saludo personalizado. Utiliza el siguiente código para capturar la entrada del usuario:

```
import 'dart:io';  
  
class MiAplicacion {  
    void ejecutar() {  
        print('Por favor, introduce tu nombre:');  
        String? nombre = stdin.readLineSync();  
        mostrarPantalla(nombre);  
    }  
}
```

```

    }

    void mostrarPantalla(String? nombre) {
        print('Pantalla: ¡Hola, $nombre!');
    }
}

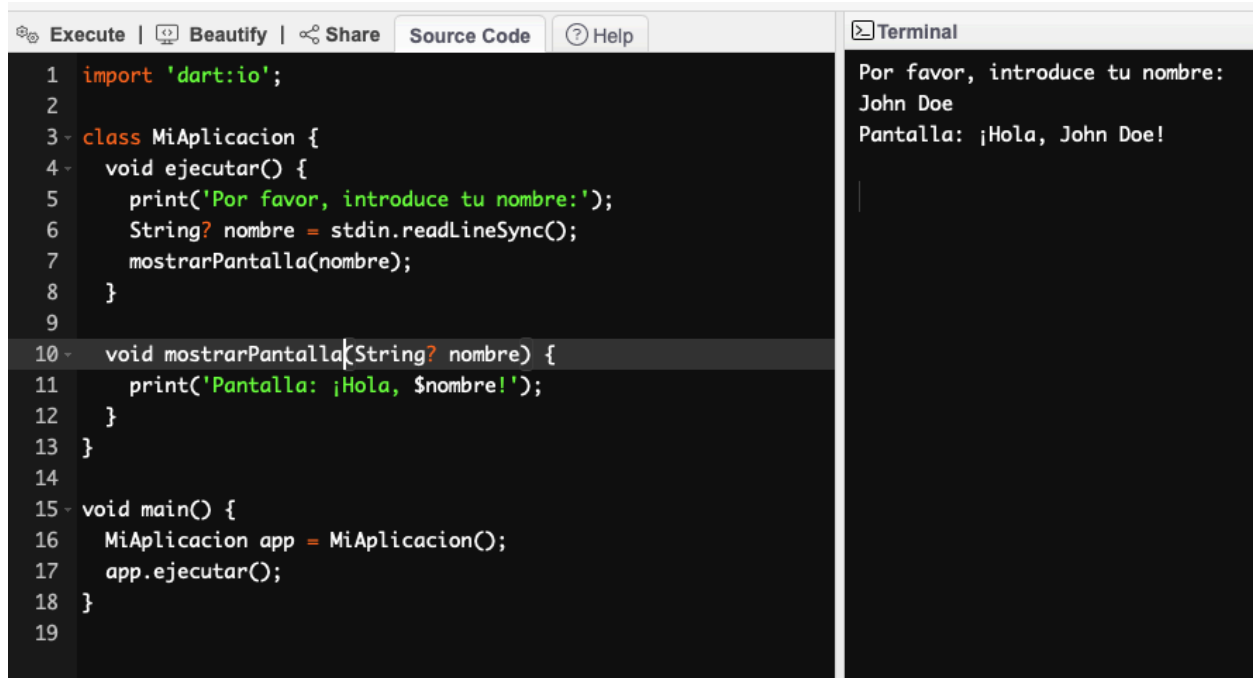
void main() {
    MiAplicacion app = MiAplicacion();
    app.ejecutar();
}

```

En este código, el programa solicita al usuario que introduzca su nombre mediante `stdin.readLineSync()` y luego lo pasa al método `mostrarPantalla` para personalizar el mensaje.

Ejemplos y Explicaciones Detalladas:

- **Entrada de datos:** Aquí usamos `stdin.readLineSync()` para leer la entrada desde la consola, lo que nos permite interactuar con el usuario.
- **Interpolación de strings:** El uso de `$nombre` en `mostrarPantalla` es un ejemplo de cómo insertar variables dentro de strings en Dart.

The image shows a code editor window with a dark theme. The top bar has tabs for 'Execute', 'Beautify', 'Share', 'Source Code', and 'Help'. The main area contains Dart code for a simple application. The code defines a class 'MiAplicacion' with two methods: 'ejecutar()' and 'mostrarPantalla()'. The 'ejecutar()' method prompts the user for their name and calls 'mostrarPantalla()'. The 'mostrarPantalla()' method prints a personalized greeting. The 'main()' function creates an instance of 'MiAplicacion' and calls 'ejecutar()'. To the right of the code editor is a terminal window. It shows the output of the program: 'Por favor, introduce tu nombre:', followed by the user input 'John Doe', and finally the output 'Pantalla: ¡Hola, John Doe!'.

```
1 import 'dart:io';
2
3 class MiAplicacion {
4   void ejecutar() {
5     print('Por favor, introduce tu nombre:');
6     String? nombre = stdin.readLineSync();
7     mostrarPantalla(nombre);
8   }
9
10  void mostrarPantalla(String? nombre) {
11    print('Pantalla: ¡Hola, $nombre!');
12  }
13 }
14
15 void main() {
16   MiAplicacion app = MiAplicacion();
17   app.ejecutar();
18 }
19
```

Por favor, introduce tu nombre:
John Doe
Pantalla: ¡Hola, John Doe!

Ejecución del código de DART presentado en esta lección. Imagen creada por J. Dastas (2024)

Relación con Otros Conceptos o Lecciones

Esta lección introduce conceptos fundamentales de programación en Dart, como el uso de clases, funciones, e interacción con el usuario. Estos conceptos se relacionan con otras lecciones que involucren estructuras de control, manejo de datos y diseño de interfaces en aplicaciones móviles.

Resumen de la Lección

En esta lección, hemos desarrollado una aplicación móvil simple utilizando Dart. La aplicación solicita el nombre del usuario y muestra un saludo personalizado en la pantalla. Hemos cubierto conceptos esenciales como la creación de clases, la captura de entrada de datos y la personalización de la salida. Este es un paso inicial en la creación de aplicaciones móviles sin frameworks.

Actividad de la Lección

1. Además de copiar el código anterior en DART y asegurarte de que el código corra, modifica la aplicación para que también muestre la edad del usuario y calcule cuántos años tendrá en 5 años.
2. Entrega todo el código generado en un documento con formato PDF.