**Documentación Clases Abstractas Dart.**

**Presentado Por:**

**Ximena Jaramillo Villada.**

**Juan Esteban Ramirez Jimenez.**

**Juan Sebastian Aristizabal.**

**Sebastian Ramirez Ramirez.**

**Yefry Alejandro Marin Serrano.**

**Presentado A:**

**Edison Sandoval.**

**Ficha - 2711993.**

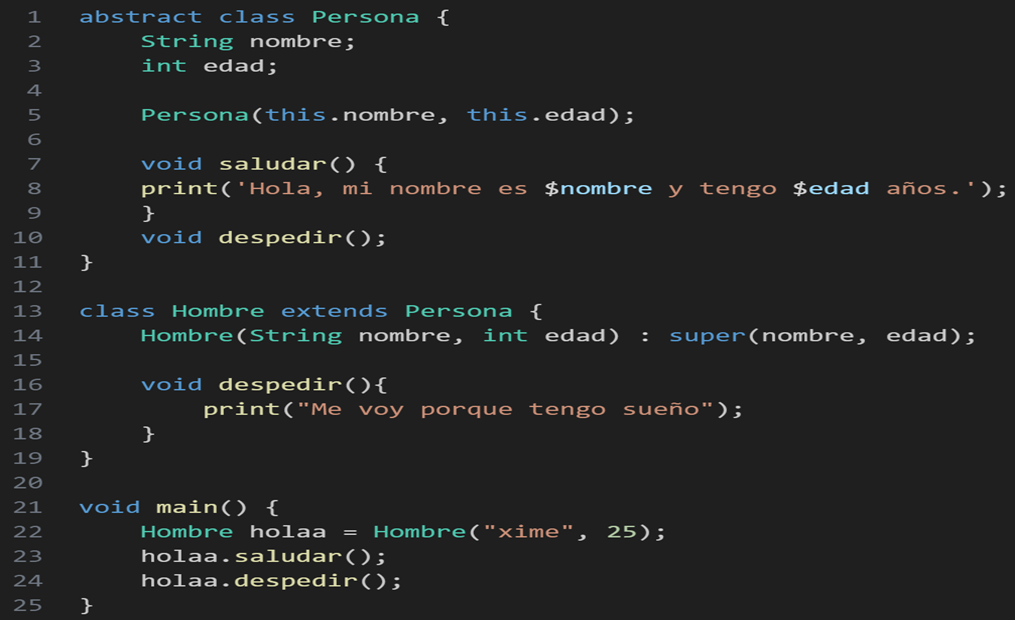
**2024.**

**Clase Abstracta.**

Una clase abstracta en Dart se define utilizando la palabra clave **abstract**. Las clases abstractas no pueden ser instanciadas directamente; están destinadas a ser extendidas por otras clases. Pueden contener métodos abstractos (métodos sin implementación) que deben ser implementados por las subclases, así como métodos con implementación.

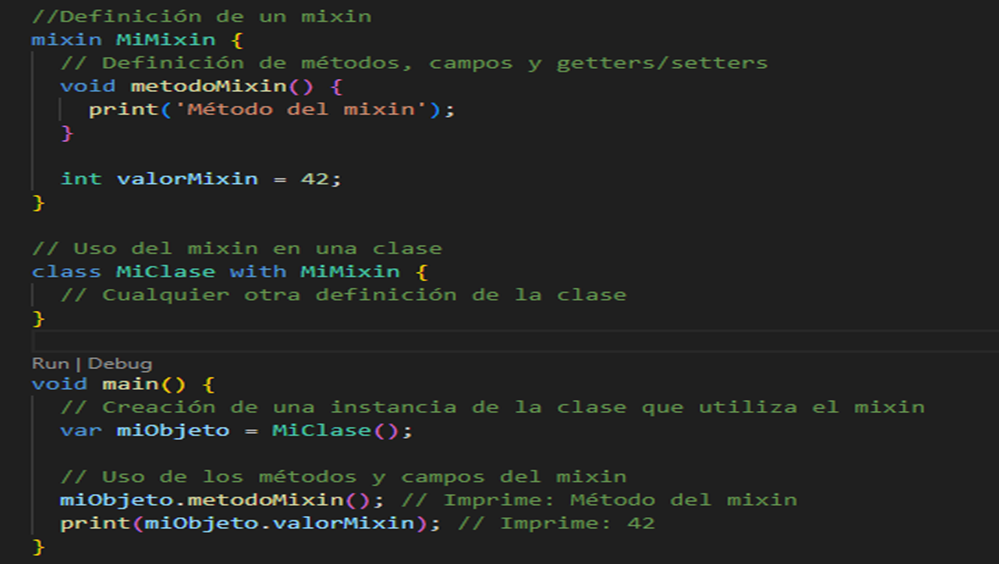
**Métodos Abstractos.**

Un método abstracto en Dart es un método que se declara sin una implementación en una clase abstracta. Este tipo de método define una firma que las subclases concretas deben implementar. Los métodos abstractos sólo pueden existir en clases abstractas y no pueden tener un cuerpo (bloque de código). Las subclases concretas que extienden la clase abstracta están obligadas a proporcionar una implementación para estos métodos abstractos



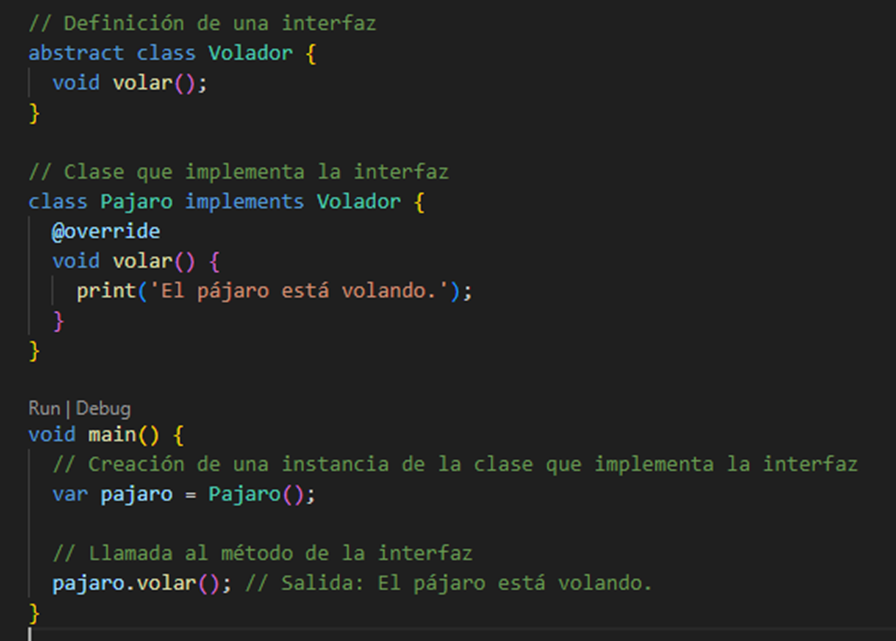
**Mixins.**

Un mixin en Dart es una forma de reutilizar el código en múltiples clases sin utilizar la herencia múltiple. Los mixins permiten compartir métodos y propiedades entre varias clases. En Dart, los mixins se definen con la palabra clave **mixin** y pueden ser utilizados por cualquier clase mediante la palabra clave **with**.



**Interfaz.**

En Dart, una interfaz es un contrato que define un conjunto de métodos y propiedades que una clase debe implementar. Aunque Dart no tiene una sintaxis explícita para definir interfaces como otros lenguajes de programación (como Java), cualquier clase puede actuar como una interfaz en Dart. Una clase que actúa como interfaz define métodos y propiedades, y cualquier clase que implemente esta interfaz debe proporcionar implementaciones para esos métodos y propiedades.



**Características Técnicas de las Interfaces en Dart:**

1. **Definición:**

* Cualquier clase puede actuar como una interfaz.
* Una clase puede implementar múltiples interfaces.

1. **Implementación:**

* Se utiliza la palabra clave implements para implementar una interfaz.
* La clase que implementa una interfaz debe proporcionar implementaciones para todos los métodos y propiedades definidos en la interfaz.

1. **Propósito:**

* Proporcionar un contrato que asegura que una clase implementará ciertos métodos y propiedades.
* Facilitar la polimorfia al permitir que diferentes clases implementen la misma interfaz.