

**Aplicaciones Móviles Multiplataforma**

**Mixins en Dart**

**Laboratorio: Introducción a DART**

**COMPETENCIAS**

* Comprender como funciona Dart
* Crear mixins en Dart

**EQUIPOS, MATERIALES, PROGRAMAS Y RECURSOS**

* PC Personal / PC de Labotatorio.
* Sistema operativo Windows 10/11.
* Material disponible desde Tecsup Virtual.
* DartPad Online

**SEGURIDAD**

* Colocar las mochilas en el gabinete al final del salón para evitar caídas en caso de sismo.
* No ingresar con bebidas ni comidas.
* Apagar los equipos y los monitores al culminar la sesión.

**INTRODUCCIÓN**

Existen diferentes lenguajes orientados a desarrollar en dispositivos móviles pero en este caso vamos a utilizar uno muy demandado actualmente y que fue creado por Google.

**PREPARACIÓN**

El Alumno debe revisar previamente el material del curso en Canvas y revisar su texto.

**ACTIVIDAD**

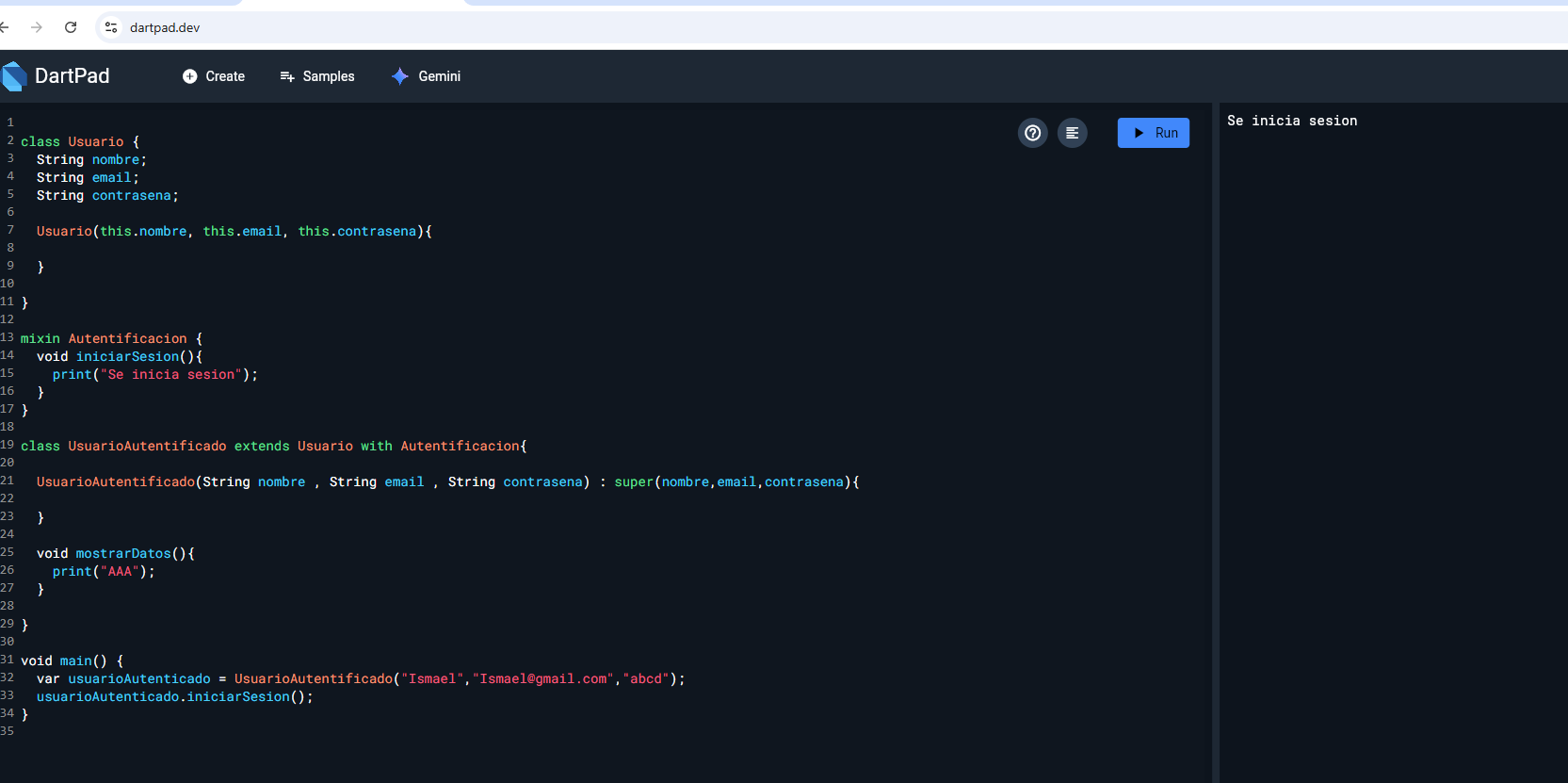
**LINK GITHUB: https://github.com/ismaeltec123/lab3\_Multiplataforma.git**

**Ejercicio 1: Control de acceso**

Crea una clase base llamada Usuario que tenga propiedades como nombre, email y contraseña. Luego, define un mixin llamado Autenticacion que tenga un método iniciarSesion para simular el inicio de sesión. Por último, crea una clase UsuarioAutenticado que herede de Usuario y utilice el mixin Autenticacion. En el programa principal, crea un objeto de tipo UsuarioAutenticado e invoca el método iniciarSesion







**Ejercicio 2: Funciones matemáticas**

Crea una clase Calculadora con métodos para realizar operaciones matemáticas básicas como suma, resta, multiplicación y división. Luego, define dos mixins llamados OperacionesAdicionales y OperacionesAvanzadas que agreguen funcionalidades como potencia y raíz cuadrada, respectivamente. Crea una clase CalculadoraAvanzada que use ambas mixins y sea capaz de realizar todas las operaciones definidas. En el programa principal, crea un objeto de tipo CalculadoraAvanzada y realiza algunas operaciones matemáticas.

**Ejercicio 3: Gestión de empleados**

Crea una clase base llamada Empleado con propiedades como nombre, apellido, salario y horasTrabajadas. Luego, define un mixin llamado Bonificaciones que tenga un método para calcular la bonificación basada en el salario del empleado. Después, crea una clase EmpleadoConBonificaciones que herede de Empleado y use el mixin Bonificaciones.

En el programa principal, crea una lista de objetos EmpleadoConBonificaciones y calcula la bonificación para cada empleado basada en su salario. Finalmente, muestra el nombre del empleado junto con su bonificación en la salida.

**Sección B**

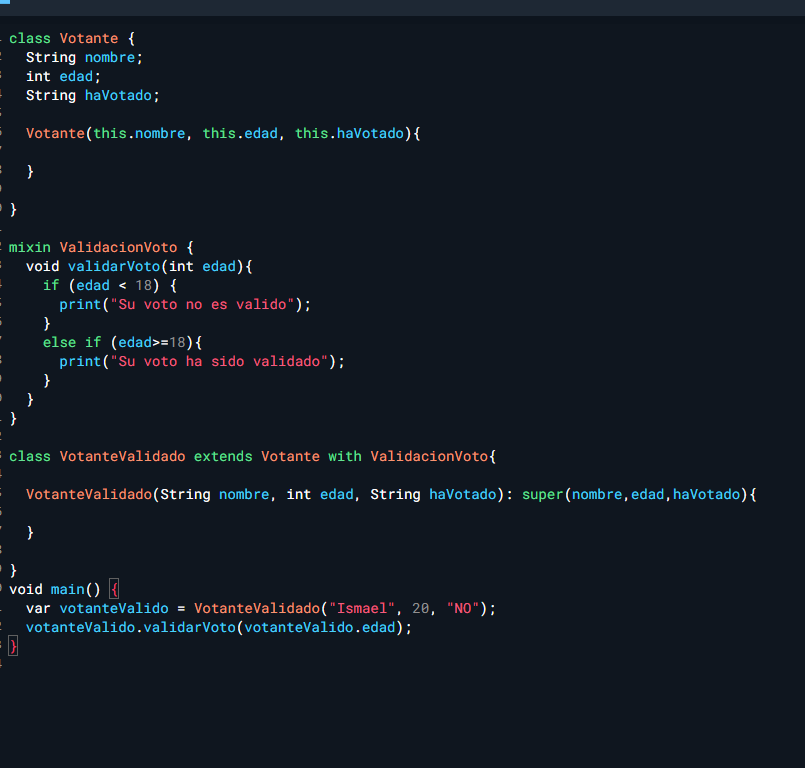
### Ejercicio 4: Sistema de gestión de inventarios

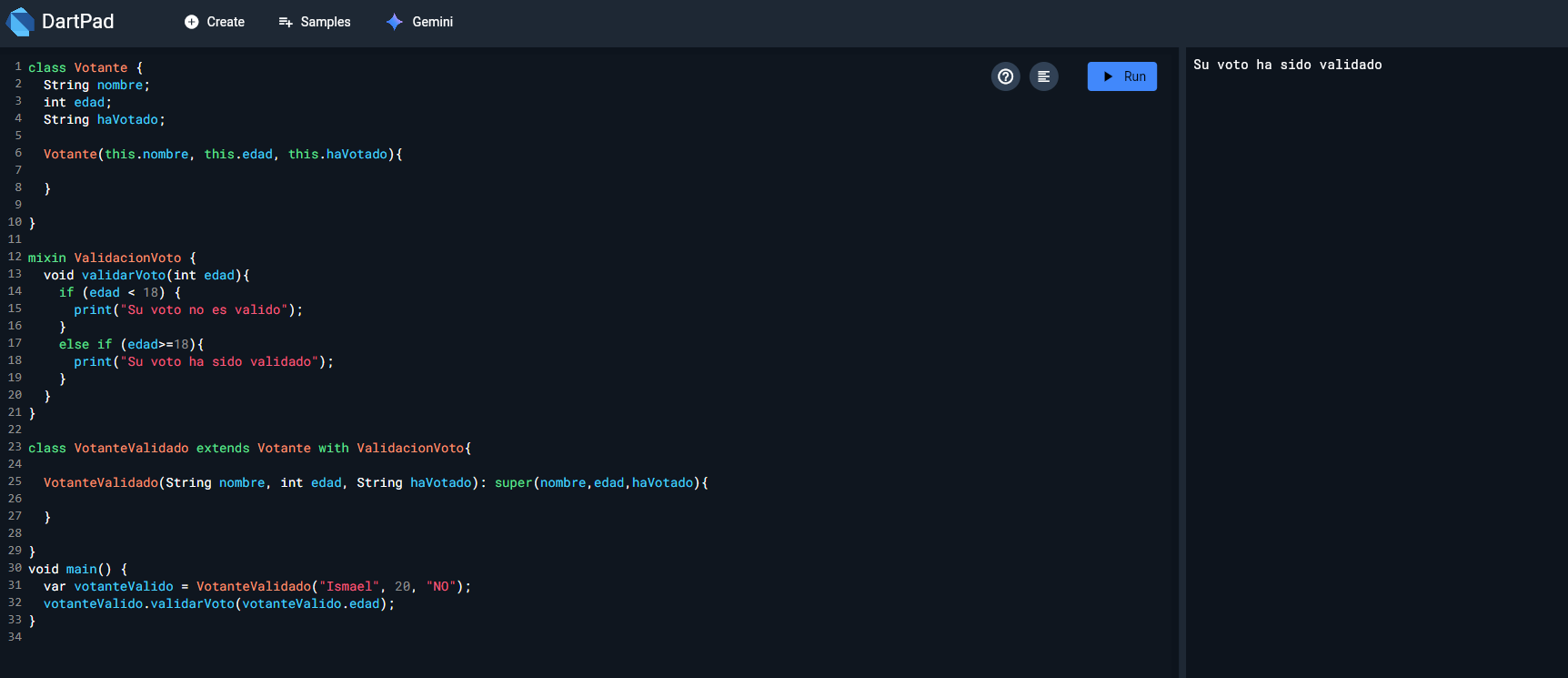
Crea una clase base llamada Producto que tenga propiedades como nombre, precio y cantidadEnStock. Luego, define un mixin llamado GestionInventario que tenga un método actualizarStock para simular la actualización del stock de productos. Por último, crea una clase ProductoConGestion que herede de Producto y utilice el mixin GestionInventario.

En el programa principal, crea un objeto de tipo ProductoConGestion, actualiza su stock y muestra el nuevo stock en la salida.

### Ejercicio 5: Sistema de votación

Crea una clase base llamada Votante que tenga propiedades como nombre, edad y haVotado. Luego, define un mixin llamado ValidacionVoto que tenga un método validarVoto para simular la validación del derecho a votar (por ejemplo, verificando la edad). Crea una clase VotanteValido que herede de Votante y use el mixin ValidacionVoto.

En el programa principal, crea un objeto de tipo VotanteValido, valida su derecho a votar, y muestra si el votante es elegible o no para votar. 



### Ejercicio 6: Sistema de reservas de hotel

Crea una clase base llamada Habitacion que tenga propiedades como numeroHabitacion, tipo, y estaDisponible. Luego, define un mixin llamado ReservaHabitacion que tenga un método reservar para simular la reserva de una habitación. Después, crea una clase HabitacionReservada que herede de Habitacion y utilice el mixin ReservaHabitacion.

En el programa principal, crea una lista de objetos HabitacionReservada, reserva algunas habitaciones, y muestra el estado de disponibilidad de cada habitación en la salid