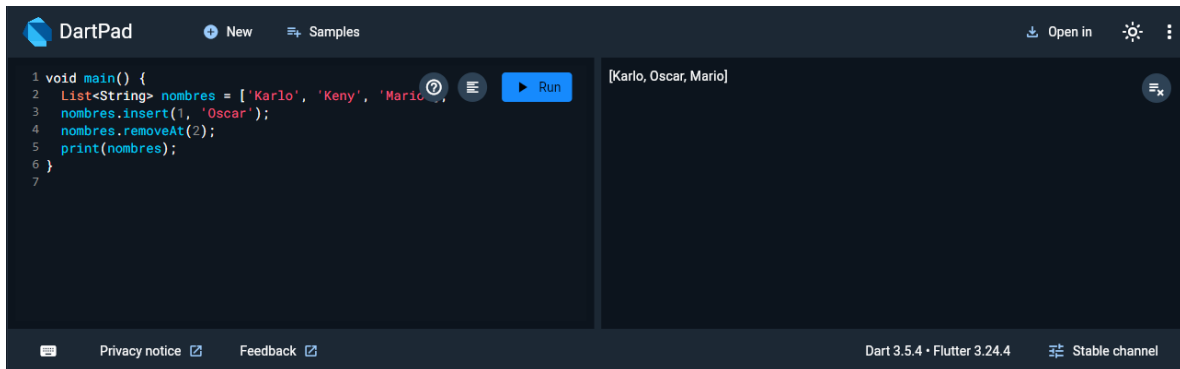


Listas

Son colecciones ordenadas de elementos en Dart.

Pueden ser fijas (tamaño fijo) o dinámicas (pueden cambiar de tamaño).

Permiten acceder a elementos por índice, modificarlos, agregar, eliminar o ordenar.



The screenshot shows the DartPad interface. On the left, the code editor contains the following Dart code:

```
1 void main() {  
2   List<String> nombres = ['Karlo', 'Kenya', 'Mario'];  
3   nombres.insert(1, 'Oscar');  
4   nombres.removeAt(2);  
5   print(nombres);  
6 }  
7
```

On the right, the output console displays the result of the code execution: `[Karlo, Oscar, Mario]`. The bottom status bar indicates the environment is Dart 3.5.4 • Flutter 3.24.4, Stable channel.

Sets

Son colecciones sin orden y sin elementos duplicados.

Útiles para verificar si algo existe o realizar operaciones como unión o intersección.



The screenshot shows the DartPad interface. On the left, the code editor contains the following Dart code:

```
1 void main() {  
2   Set<int> numerosA = {1, 2, 3};  
3   Set<int> numerosB = {3, 4, 5};  
4  
5   var union = numerosA.union(numerosB);  
6   var interseccion = numerosA.intersection(numerosB);  
7  
8   print(union);  
9   print(interseccion);  
10 }  
11
```

On the right, the output console displays the results of the code execution: `{1, 2, 3, 4, 5}` and `{3}`. The bottom status bar indicates the environment is Dart 3.5.4 • Flutter 3.24.4, Stable channel.

Maps

Son pares clave-valor.

Las claves son únicas y sirven para acceder a los valores.



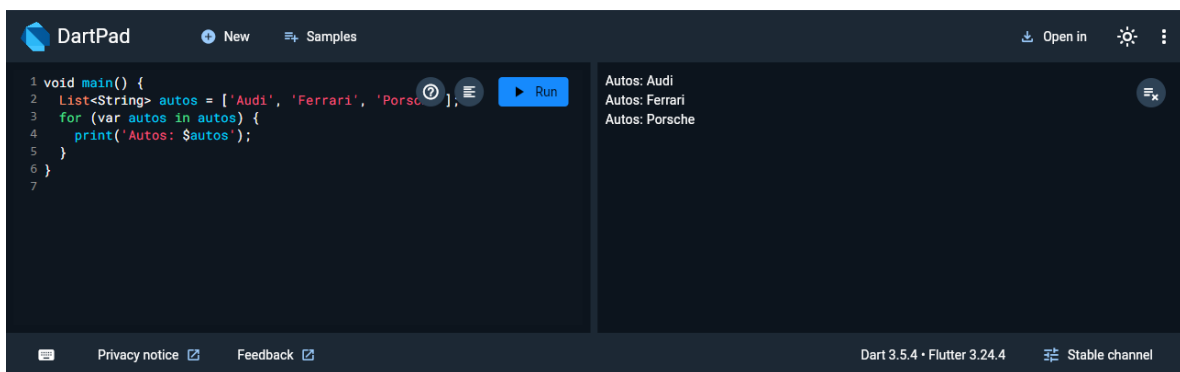
The screenshot shows the DartPad interface with a Dart script that creates a Map of student names and IDs. The code is as follows:

```
1 void main() {  
2   Map<int, String> estudiantes = {1: 'Karlo', 2: 'Mario', 3: 'Oscar'};  
3  
4   estudiantes[4] = 'Oscar';  
5   estudiantes.remove(2);  
6   print(estudiantes);  
7 }  
8
```

The output on the right shows the state of the Map after the operations: {1: Karlo, 3: Mario, 4: Oscar}.

Bucles

for: Ejecuta un número fijo de veces.

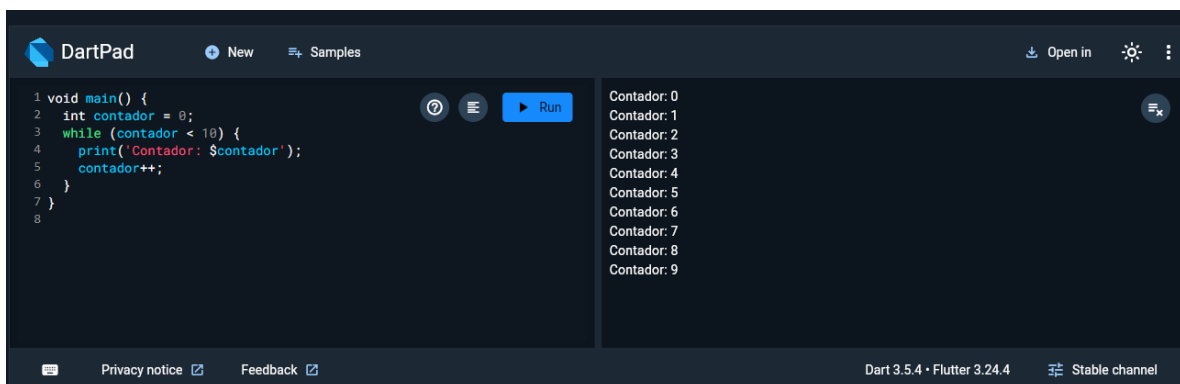


The screenshot shows the DartPad interface with a Dart script that uses a for loop to print car names. The code is as follows:

```
1 void main() {  
2   List<String> autos = ['Audi', 'Ferrari', 'Porsche'];  
3   for (var autos in autos) {  
4     print('Autos: $autos');  
5   }  
6 }  
7
```

The output on the right shows the results of the loop: Autos: Audi, Autos: Ferrari, Autos: Porsche.

while: Repite mientras una condición sea verdadera.

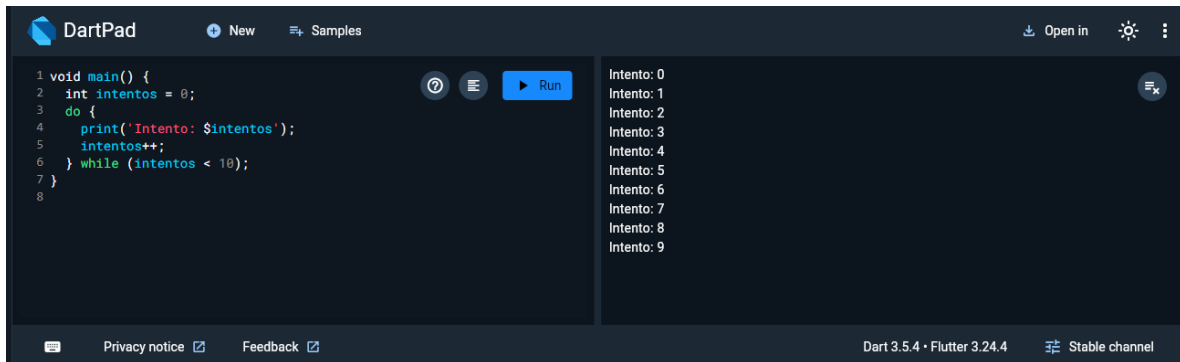


The screenshot shows the DartPad interface with a Dart script that uses a while loop to print a counter from 0 to 9. The code is as follows:

```
1 void main() {  
2   int contador = 0;  
3   while (contador < 10) {  
4     print('Contador: $contador');  
5     contador++;  
6   }  
7 }  
8
```

The output on the right shows the results of the loop: Contador: 0, Contador: 1, Contador: 2, Contador: 3, Contador: 4, Contador: 5, Contador: 6, Contador: 7, Contador: 8, Contador: 9.

do-while: Siempre ejecuta al menos una vez.



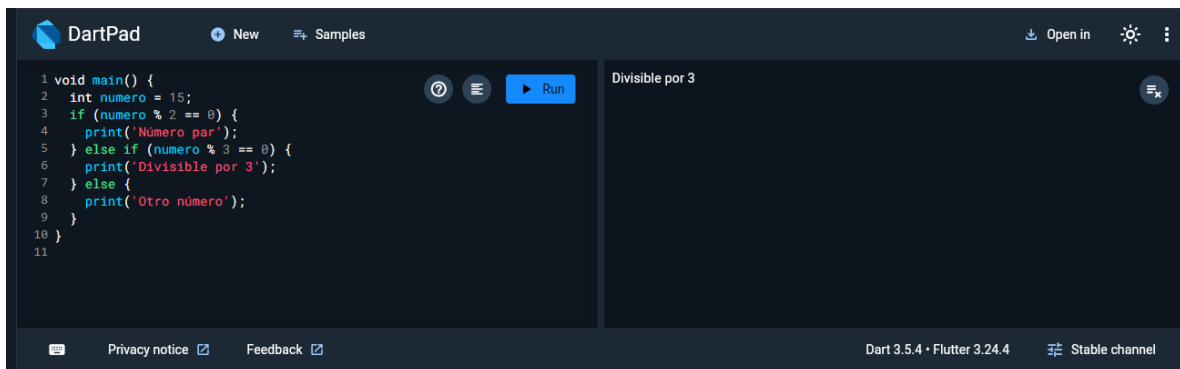
The screenshot shows the DartPad interface with a Dart code snippet using a `do-while` loop. The code prints the word "Intento" followed by a counter from 0 to 9. The output on the right side of the editor shows the results of the loop.

```
1 void main() {  
2   int intentos = 0;  
3   do {  
4     print('Intento: $intentos');  
5     intentos++;  
6   } while (intentos < 10);  
7 }  
8
```

Output: Intento: 0, Intento: 1, Intento: 2, Intento: 3, Intento: 4, Intento: 5, Intento: 6, Intento: 7, Intento: 8, Intento: 9

Condicionales

if: Ejecuta si una condición es verdadera.

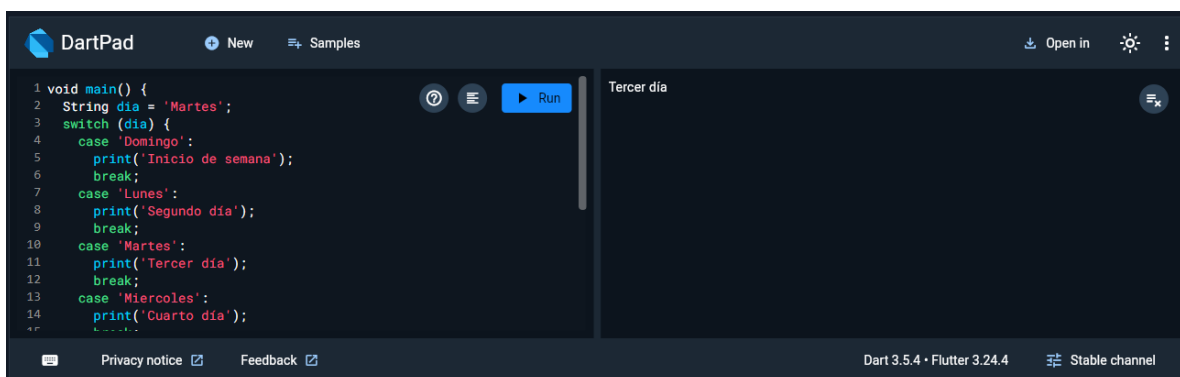


The screenshot shows the DartPad interface with a Dart code snippet using an `if-else` conditional. The code checks if a number is divisible by 2 or 3 and prints the appropriate message. The output on the right side of the editor shows the result for the number 15.

```
1 void main() {  
2   int numero = 15;  
3   if (numero % 2 == 0) {  
4     print('Número par');  
5   } else if (numero % 3 == 0) {  
6     print('Divisible por 3');  
7   } else {  
8     print('Otro número');  
9   }  
10 }  
11
```

Output: Divisible por 3

switch: Selecciona entre varios casos según un valor.



The screenshot shows the DartPad interface with a Dart code snippet using a `switch` statement. The code checks the day of the week and prints the corresponding message. The output on the right side of the editor shows the result for the day "Martes".

```
1 void main() {  
2   String dia = 'Martes';  
3   switch (dia) {  
4     case 'Domingo':  
5       print('Inicio de semana');  
6       break;  
7     case 'Lunes':  
8       print('Segundo día');  
9       break;  
10    case 'Martes':  
11      print('Tercer día');  
12      break;  
13    case 'Miercoles':  
14      print('Cuarto día');  
15      break;  
16    }  
17 }
```

Output: Tercer día