

SESIÓN UNO:

1. Presentación de participantes
2. Ruta:

Ruta de Formación: Desarrollo de aplicaciones FullStack, utilizando bases de datos SQL y NoSQL.



Nivel Básico: Fundamentos y Exploración

- Bases esenciales del desarrollo
- Programación y conceptos básicos.
- Sintaxis de FrontEnd
- Configuración del entorno de desarrollo.
- Sintaxis Backend y acceso a datos.
- Introducción a Flutter y Dart para diseño de interfaces de usuario y bases de datos.

Nivel Intermedio: Profundización y Conexión

- Uso de frameworks y librerías populares.
- Creación y uso de APIs RESTful.
- Desarrollo backend avanzado.
- Exploración de bases de datos NoSQL.
- Conexión y configuración de bases de datos NoSQL.
- Dominio de operaciones CRUD en Backend

Nivel Avanzado: Maestría y Optimización

- Desarrollo backend con patrones de diseño.
- Implementación de servicios y microservicios.
- Desarrollo de aplicaciones complejas en Flutter.
- Integración de funcionalidades avanzadas (Sensores).

Nivel Experto: Testing, Seguridad y Más

- Testing avanzado y seguridad.
- Estrategias avanzadas de pruebas.
- Fortalecimiento de la seguridad en todas las capas de las aplicaciones.
- Perfeccionamiento de habilidades.
- Enfrentamiento de desafíos complejos en el desarrollo.
- Pruebas rigurosas para garantizar calidad.

Activar Windows
Ve a Configuración para activar Windows.

3. APP MONOLÍTICA VS MICROSERVICIOS



4. Principios fundamentales de Dart

<https://dartpad.dev/>

```

1 void main() {
2
3 /* IMPRIMIR DATOS PERSONALES, NOMBRE, REGIONAL,
4  * EDAD, FUNCIONARIO (TRUE, FALSE), ALTURA
5  (METROS)
6  * */
7
8 String name = "Albeiro Ramos";
9 String regional = "Distrito Capital";
10 int age = 41;
11 bool officer = false;
12 double height = 1.72;
13
14 print("Nombre:" + name);
15 print("Regional:" + regional);
16 print("Edad: " + age.toString());
17 print("Funcionario: " + officer.toString());
18 print("Altura: " + height.toString());
19
20

```

Nombre:Albeiro Ramos
Regional:Distrito Capital
Edad: 41
Funcionario: false
Altura: 1.72

```

1 void main() {
2
3 /* IMPRIMIR DATOS PERSONALES, NOMBRE, REGIONAL,
4  * EDAD, FUNCIONARIO (TRUE, FALSE), ALTURA
5  (METROS)
6  * */
7
8 String name = "Albeiro Ramos";
9 String regional = "Distrito Capital";
10 int age = 41;
11 bool officer = false;
12 double height = 1.72;
13
14 print("Nombre: $name");
15 print("Regional: $regional");
16 print("Edad: $age");
17 print("Funcionario: $officer");
18 print("Altura: $height");
19
20

```

Nombre: Albeiro Ramos
Regional: Distrito Capital
Edad: 41
Funcionario: false
Altura: 1.72

```

1 void main() {
2
3 /* IMPRIMIR DATOS PERSONALES, NOMBRE, REGIONAL,
4  * EDAD, FUNCIONARIO (TRUE, FALSE), ALTURA
5  (METROS)
6  * */
7
8 String name = "Albeiro Ramos";
9 String regional = "Distrito Capital";
10 int age = 41;
11 bool officer = false;
12 double height = 1.72;
13
14 print("Nombre: $name");
15 print("Regional: $regional");
16 print("Edad: $age");
17 print("Funcionario: $officer");
18 print("Altura: $height");
19
20 print("Mi nombre es $name de la Regional
21 $regional, tengo $age años y mi alura es de
22 $height metros");
23
24 }

```

Nombre: Albeiro Ramos
Regional: Distrito Capital
Edad: 41
Funcionario: false
Altura: 1.72
Mi nombre es Albeiro Ramos de la
Regional Distrito Capital, tengo 41
años y mi alura es de 1.72 metros

```

1 // Copyright 2015 the Dart project authors. All rights reserved.
2 // Use of this source code is governed by a BSD-style license
3 // that can be found in the LICENSE file.
4
5 void main() {
6   var datos = {
7     // Key: Value
8     'nombres': 'Diego Ojeda',
9     'Regional': 'Boyacá',
10    'Edad': 33,
11    'Funcionario': true,
12    'Altura': 1.65
13  };
14
15  print(datos);
16 }
17

```

(nombres: Di

```

void main() {

  /* condicional simple
  if(condicion){ }

  /////////// condicional simple con condicion doble
  if(condicion){}else{}

  /////////// condicional secuencia
  if(condicion){}
  else if (condicion2){}
  else if (condicion 3){}
  else if (condicion n){}

  /////////// condicional anidado
  if(condicion){
    if(subcondicon){
      if (subcondicion2){}
    }
  }else{ igual que if}

}

```

```

1 void main() {
2
3   /*Construya un software que permita indicar
4    *si una persona se encuentra apta para votar
5    o no*/
6
7   int age = 10;
8
9   if (age >= 18){
10    print("Si puede votar, ya que tiene $age años");
11  } else {
12    print("No puede votar, ya que tiene $age años");
13  }
14 }
15

```

No puede votar, ya que tiene 10 años

```
1 void main() {
2
3 /* operadores logicos
4  > mayor que
5  < menor que
6  >= mayor igual que
7  <= menor igual que
8  == igualdad
9  != diferente
10  && conector de condicion (y) = todo de ser verdad
11  || conector de condicion (o) = alguna condicion debe ser verdad
12  * */
13
14 }
15
```

```
1 void main() {
2
3 List ejemplo=[23,5432,56,647382,432];
4 List ejemplo2=[23,"Albeiro",true,647382,314123,"Henández"];
5 List ejemplo3=[true,11211,ejemplo];
6 print(ejemplo);
7 print(ejemplo2);
8 print(ejemplo3);
9 List<String> ejemplo4=["hola","Mundo","98"];
10 print(ejemplo4);
11 List<int> ejemplo5=[123,456,789];
12 print(ejemplo5);
13
14 }
15
```

[23, 5432, 5
[23, Albeiro,
[true, 11211
[hola, Mund
[123, 456, 7

TALLER 1:

1. **Suma de dos números:** Escribe un programa que pida al usuario dos números enteros y luego muestre la suma de esos dos números.
2. **Factorial de un número:** Desarrolla un algoritmo que calcule el factorial de un número entero ingresado por el usuario. El factorial de un número n se denota como $n!$ y se calcula como el producto de todos los enteros positivos desde 1 hasta n .
3. **Ordenar una lista:** Crea un programa que ordene una lista de números enteros en orden ascendente. Puedes implementar algún algoritmo de ordenamiento como el método de la burbuja o de selección.
4. **Buscar un elemento en una lista:** Escribe un algoritmo que busque un número dado en una lista de números enteros. Si el número está presente en la lista, el programa debe indicar su posición; de lo contrario, debe mostrar un mensaje indicando que el número no se encontró.
5. **Contar palabras en una cadena de texto:** Desarrolla un algoritmo que cuente el número de palabras en una cadena de texto ingresada por el usuario. Una palabra se define como una secuencia de caracteres separados por espacios en blanco.
6. **Calcular el promedio de una lista de números:** Escribe un programa que calcule el promedio de una lista de números enteros ingresados por el usuario.

7. **Convertir temperatura:** Crea un algoritmo que convierta una temperatura dada en grados Celsius a grados Fahrenheit. La fórmula de conversión es: $Fahrenheit = (Celsius \times 9/5) + 32$.
8. **Encontrar el número más grande en una lista:** Desarrolla un programa que encuentre el número más grande en una lista de números enteros ingresados por el usuario.
9. **Fibonacci:** Escribe un algoritmo que genere los primeros n números de la secuencia de Fibonacci, donde cada número es la suma de los dos números anteriores. Los primeros dos números de la secuencia son 0 y 1.
10. **Verificar si un número es primo:** Crea un programa que verifique si un número dado por el usuario es primo o no. Un número primo es aquel que es divisible únicamente por 1 y por sí mismo.

SESIÓN DOS:

1. Recordando sesión uno:
 - a. Arquitectura monolítica y microservicios (Front-End y Back-End)
 - b. Dart: Listas, maps, colecciones y funciones.
2. Descargar, descomprimir e instalar para Windows / Web [Flutter.dev](https://flutter.dev) en el disco local C:.
Dependiendo lo que escoja, descargará un instructivo o guía.
3. Editar variable de entorno en el PATH / Nuevo / 'C:\flutter\bin' / Aceptar
4. En el PowerShell, escribir 'flutter' - 'flutter doctor'. Nota: Si se tiene instalador 'Dart', es mejor desinstalarlo, porque 'Flutter' ya tiene 'Dart'
5. Instalar VSCode con las extensiones: Flutter - Flutter Widget Snippets - Awesome Widget Snippets - Flutter Color - Material Icon Theme
6. Crear proyecto: view / command palette (Ctrl + shift + p) / Flutter: New Project / Application
7. Iniciar el servidor (terminal):
 - a. Run / Star debbuging
 - b. "Parte superior derecha, Icono:" / "Run"
 - c. flutter run (**terminal**).
 - d. ctrl + c (Cerrar el servidor) (**terminal**).
8. Actividad Sesión_2:
 - a. Proyecto nuevo
 - b. Ejecutar Chrome
 - c. Visualización modo móvil
 - d. Cambiar el color morado
 - e. Cambiar la información del toolbar por tu nombre

SESIÓN TRES:

1. Recordando sesión dos:
 - a. Instalación de Flutter, mi primer proyecto
2. Estructura de carpetas y archivos
 - a. Todo lo que se ve en Flutter se llama 'widget' (Fragmentos de código que intervienen directamente con las interfaces gráficas). Existen dos tipos: 1] StatelessWidget, no tienen estado, no cambian en tiempo de ejecución (Ejp, etiquetas, textos); 2] StatefulWidget, si tienen estado (Ejp, botón).
 - b. La aplicación arranca desde 'lib/main.dart' en la función 'void main()'.
 - c. Agregar función 'main' / Dentro 'runApp'.
 - d. Agregar estructuras de widgets con 'sta' (Ejemplo, StatelessWidget).
 - e. En lugar del 'Container', reemplazar con MaterialApp'. Agregar los widget para 'pintar' la página inicial (Ejemplo, title y home: Text()).
 - f. Reemplazar en el home: 'Text()' con 'Column()'.
 - g. Crear un estándar de organización a través del widget 'Scaffold'.
 - h. Nota: Se aconseja tener servidor de archivos (imágenes), db, aplicaciones, etc.

SESIÓN CUATRO:

1. Recordando sesión tres:
 - a. Trabajando con el archivo 'lib/main'