



## REQUERIMIENTO TÉCNICO

### 1.- IDENTIFICACIÓN DEL REQUERIMIENTO

<b>Proyecto</b>	Facturacion.cl		
<b>Folio</b>	Actividad 3	<b>Versión</b>	1.0
<b>Nombre</b>	Crear una interfaz que liste los usuarios registrados y guardados en la base de datos. Se debe agregar un filtro en la lista según el nombre del usuario.		
<b>Nº de entrega</b>	1		

### 2.- DATOS DE ENTREGA

<b>Fecha Entrega</b>	23-10-2024
<b>Nombre Analista</b>	Nicolás Hidalgo
<b>Observación</b>	

### *Crear Archivo en Flutter*

<b><i>Acción</i></b>	- Crear Archivo	<b><i>Nombre</i></b>	list_screen
----------------------	-----------------	----------------------	-------------

Como se indica en la imagen 1:

- Se debe crear una vista que liste a los usuarios guardados en la base de datos, debe permitir la búsqueda según el nombre del usuario.

#### **Definición de parámetros a recibir o utilizar:**

- BuildContext context

#### **Definición de lógica a realizar**

- **Navegación entre vistas:** Implementar la lógica para navegar entre un Usuario de la lista a una vista de detalle.
- **Menú Hamburguesa (Drawer):** Se debe añadir un menú en la barra que permita navegar entre las distintas vistas.

***IMAGEN 1***

```

class _UserListScreenState extends State<UserListScreen> {
  Widget build(BuildContext context) {
    body: Column(
      children: [
        Padding(
          padding: const EdgeInsets.symmetric(
            horizontal: 16.0,
            vertical: 5.0,
          ), // EdgeInsets.symmetric
          child: TextFormField(
            controller: _searchController,
            decoration: InputDecoration(
              prefixIcon: const Icon(
                Icons.search,
              ), // Icon
              label: const Text("Nombre"),
              fillColor: Colors.grey[100],
              filled: true,
              border: const OutlineInputBorder(
                borderSide: BorderSide.none,
                borderRadius: BorderRadius.all(
                  Radius.circular(10),
                ), // BorderRadius.all
              ), // OutlineInputBorder
            ), // InputDecoration
          ), // TextFormField
        ), // Padding
        Expanded(
          child: ListView.builder(
            itemCount: filteredUsers.length,
            itemBuilder: (context, index) {
              return Padding(
                padding: const EdgeInsets.symmetric(
                  horizontal: 8.0,
                ), // EdgeInsets.symmetric
                child: ListTile(
                  tileColor:
                    (index % 2 == 0) ? Colors.blue[100] : Colors.blue[50],
                  title: Text(filteredUsers.elementAt(index).name),
                  subtitle: Row(
                    children: [
                      Text(filteredUsers.elementAt(index).phoneNumber),
                      const SizedBox(
                        width: 10,
                      ), // SizedBox
                      Expanded(
                        child: Text(
                          filteredUsers.elementAt(index).email,
                          style: TextStyle(color: Colors.purple[600]),
                        ), // Text
                      ), // Expanded
                    ],
                  ), // Row
                  trailing: GestureDetector(
                    child: Container(

```

IMAGEN 2

```

trailing: GestureDetector(
  child: Container(
    padding: const EdgeInsets.all(5),
    decoration: BoxDecoration(
      borderRadius: BorderRadius.circular(50),
      color: Colors.grey[200],
      boxShadow: const [
        BoxShadow(
          color: Colors.black,
          blurRadius: 0.5,
          spreadRadius: 0.0,
          offset: Offset(0.5, 0.5),
        ) // BoxShadow
      ],
    ), // BoxDecoration
    child: const Icon(
      Icons.arrow_forward_ios,
    ), // Icon
  ), // Container
  onTap: () {
    Navigator.push(context, MaterialPageRoute(
      builder: (context) {
        return UserDetails(
          user: filteredUsers.elementAt(index),
        ); // UserDetails
      },
    )); // MaterialPageRoute
  },
), // GestureDetector
// ListTile

```

### *Crear función de carga de datos en Flutter*

Acción	- Crear función	Nombre	loadUsers()
--------	-----------------	--------	-------------

#### **Definición de parámetros a recibir o utilizar:**

- Ninguno

#### **Definición de lógica a realizar**

- **Carga de datos:** Se debe crear una función en el archivo de configuración de la base de datos para obtener a los usuarios guardados.

**IMAGEN 1**

```

Future<List<User>> getUsers() async {
  final db = await database;
  final List<Map<String, dynamic>> maps = await db.query('users');

  return List.generate(maps.length, (i) {
    return User(
      id: maps[i]['id'],
      name: maps[i]['name'],
      email: maps[i]['email'],
      phoneNumber: maps[i]['phoneNumber'],
      birthDate: maps[i]['birthDate'],
      address: maps[i]['address'],
      password: maps[i]['password'],
    ); // User
  }); // List.generate
}

```

### *Crear función de carga de datos en Flutter*

<b>Acción</b>	- Crear función	<b>Nombre</b>	loadUsers()
---------------	-----------------	---------------	-------------

#### **Definición de parámetros a recibir o utilizar:**

- List<User> filteredUsers
- List<User> users

#### **Definición de lógica a realizar**

- **Carga de datos:** Se debe crear una función para obtener los usuarios guardados en la base de datos utilizando la función anterior.

**IMAGEN 1**

```

@override
void initState() {
  loadUsers();
  _searchController.addListener(_filterUsers);
  super.initState();
}

Future<void> loadUsers() async {
  final usersData = await database.getUsers();
  setState(() {
    users = usersData;
    filteredUsers = users;
  });
}

```

### Crear función de filtro en Flutter

<b>Acción</b>	- Crear función	<b>Nombre</b>	_filterUsers()
---------------	-----------------	---------------	----------------

#### Definición de parámetros a recibir o utilizar:

- List<User> filteredUsers

#### Definición de lógica a realizar

- **Filtro de usuarios:** Se debe crear una función para filtrar a los usuarios según su nombre.

#### IMAGEN 1

```

void _filterUsers() {
  String query = _searchController.text.toLowerCase();
  setState(() {
    filteredUsers = users.where((user) {
      return user.name.toLowerCase().contains(query);
    }).toList();
  });
}

```