

			
	REQUERIMIENTO TÉCNICO		

1.- IDENTIFICACIÓN DEL REQUERIMIENTO			
Proyecto	Facturacion.cl		
Folio	3	Versión	1.0
Nombre	Creación interfaz listado de cuentas		
Nº de entrega	1		

2.- DATOS DE ENTREGA	
Fecha Entrega	23-10-2024
Nombre Analista	Jeremias Campos
Observación	implementación de la interfaz de listado de cuentas, incluyendo búsqueda y navegación a detalle de cuenta.

Modificar Archivo			
Acción	Crear Pantalla: listado_screen.dart – ListadoScreen	Nombre	Implementación de pantalla Listado en Flutter lib/listado_screen.dart
<div>Estructura del Formulario</div> <ul style="list-style-type: none">Campos incluidos: Nombre completo del usuario, número de teléfono, dirección de correo electrónico. <div>Diseño General</div> <ul style="list-style-type: none">Barra de navegación superior color azul (#1A90D9).Título "Listado" centrado en la barra de navegación.Icono de menú (tres líneas horizontales) en la esquina superior izquierda. <div>Barra de Búsqueda</div> <ul style="list-style-type: none">Ubicada justo debajo de la barra de navegación.Campo de entrada con placeholder "Nombre".Icono de lupa a la izquierda del campo de entrada. <div>Lista de Cuentas</div> <p>Cada entrada de la lista debe mostrar:</p> <ul style="list-style-type: none">Nombre completo del usuario.Número de teléfono.Dirección de correo electrónico. <div>Funcionalidad de la Lista</div> <ul style="list-style-type: none">Crear una lista desplazable utilizando un widget ListView.builder.Si la lista de usuarios está vacía:<ul style="list-style-type: none">Acción: Mostrar un mensaje en el centro de la pantalla que diga "No hay usuarios registrados".Si la lista contiene usuarios:<ul style="list-style-type: none">Acción: Mostrar cada usuario en un contenedor con las siguientes características:Mostrar el nombre completo en la parte superior.Mostrar el número de teléfono debajo del nombre, formateado adecuadamente (por ejemplo, (123) 456-7890).Mostrar el correo electrónico en la parte inferior con un color distinto (por ejemplo, azul).El contenedor debe alternar colores de fondo para diferenciar cada elemento. <div>Comportamiento de Búsqueda</div> <ul style="list-style-type: none">Crear una función <code>_filtrarUsuarios</code> que reciba el valor ingresado en el campo de búsqueda y realice lo siguiente:Si el campo de búsqueda está vacío:<ul style="list-style-type: none">Acción: Mostrar la lista completa de usuarios.Si el campo contiene texto:<ul style="list-style-type: none">Acción: Filtrar los usuarios cuyo nombre contenga el texto ingresado (ignorando mayúsculas y minúsculas).Si hay coincidencias:<ul style="list-style-type: none">Acción: Mostrar la lista filtrada con los usuarios que coinciden con la búsqueda.Si no hay coincidencias:<ul style="list-style-type: none">Acción: Mostrar un mensaje en el centro de la pantalla que diga "No se encontraron resultados".			

IMAGEN 1

```
64
65 void _filtrarUsuarios() {
66     String query = _searchController.text.toLowerCase();
67     setState(() {
68         usuariosFiltrados = usuarios
69             .where((usuario) => usuario['nombre'].toLowerCase().contains(query))
70             .toList();
71     });
72 }
```

IMAGEN 2

```
211 : ListView.builder(
212     itemCount: usuariosFiltrados.length,
213     itemBuilder: (context, index) {
214         final usuario = usuariosFiltrados[index];
215         return Container(
216             decoration: BoxDecoration(
217                 color: index.isEven
218                     ? Color(0xFFD5E5ED)
219                     : Color(0xFFEFF3F6),
220             ), // BoxDecoration
221             child: ListTile(
222                 contentPadding: EdgeInsets.symmetric(
223                     vertical: 0,
224                     horizontal: 12.0,
225                 ), // EdgeInsets.symmetric
226                 title: Text(
227                     usuario['nombre'] ?? 'Sin nombre',
228                     style: TextStyle(
229                         fontSize: 16.0,
230                     ), // TextStyle
231                 ), // Text
232                 subtitle: Row(
233                     children: [
234                         Flexible(
235                             flex: 2,
236                             child: FittedBox(
237                                 fit: BoxFit.scaleDown,
238                                 alignment: Alignment.centerLeft,
239                                 child: Text(
240                                     formatPhoneNumber(usuario['celular'] ?? '(No encontrado)'),
241                                     style: TextStyle(fontSize: 12.0),
242                                 ), // Text
243                             ), // FittedBox
244                         ), // Flexible
```

```

245         SizedBox(width: 4),
246         Flexible(
247           flex: 3,
248           child: FittedBox(
249             fit: BoxFit.scaleDown,
250             alignment: Alignment.centerLeft,
251             child: Text(
252               usuario['correo'] ?? 'Correo no disponible',
253               style: TextStyle(color: Color(0xFF2A1CC), fontSize: 16.0),
254             ), // Text
255           ), // FittedBox
256         ), // Flexible
257       ],
258     ), // Row
259     trailing: Container(
260       width: 30,
261       height: 30,
262       decoration: BoxDecoration(
263         color: Color(0xFFF4F0F0),
264         shape: BoxShape.circle,
265         boxShadow: [
266           BoxShadow(
267             color: Colors.grey.withOpacity(0.5),
268             blurRadius: 5,
269             spreadRadius: 1,
270           ), // BoxShadow
271         ],
272       ), // BoxDecoration
273       child: Icon(
274         Icons.arrow_forward_ios,
275         size: 15.0,
276         color: Color(0xFF606163),
277       ), // Icon
278     ), // Container

```

IMAGEN 3

```

      child: TextField(
        controller: _searchController,
        decoration: InputDecoration(
          prefixIcon: Icon(Icons.search, color: Colors.grey),
          hintText: 'Nombre',
          border: InputBorder.none,
          contentPadding: EdgeInsets.symmetric(vertical: 12),
        ), // InputDecoration
      ), // TextField
    ), // Container

```

IMAGEN 4

```

Future<void> _initDatabase() async {
  final documentsDirectory = await getApplicationDocumentsDirectory();
  final path = p.join(documentsDirectory.path, 'usuarios.db');

  database = await openDatabase(
    path,
    version: 1,
  );
  _cargarUsuarios();
}

```