

			
	REQUERIMIENTO TÉCNICO		

1.- IDENTIFICACIÓN DEL REQUERIMIENTO			
Proyecto	Facturacion.cl		
Folio	Actividad 5	Versión	1.0
Nombre	Crear una interfaz que permita iniciar sesión dentro de la aplicación. El usuario debe estar guardado en la base de datos para ingresar. En caso de que ingrese un dato de manera incorrecta entonces se le debe mostrar un mensaje de error.		
N° de entrega	1		

2.- DATOS DE ENTREGA	
Fecha Entrega	23-10-2024
Nombre Analista	Nicolás Hidalgo
Observación	

Crear Archivo en Flutter

<i>Acción</i>	- Crear Archivo	<i>Nombre</i>	login_screen
----------------------	-----------------	----------------------	--------------

Como se indica en la imagen 1:

- Se debe crear una vista que permita a un usuario ingresar dentro de la aplicación con su correo y contraseña. Debe haber un botón que se comuniquen con la base de datos para verificar que el usuario es correcto, en caso de que no lo sea, entonces debe mostrarse un mensaje de error.

Definición de parámetros a recibir o utilizar:

- BuildContext context
- TextEditingController _emailController
- TextEditingController _passController
- bool isVisible
- DBSqlite database

Definición de lógica a realizar

- **Navegación entre vistas:** Al presionar el botón de “Ingresar” si el usuario es correcto, entonces debe navegar a la vista de la actividad 1.
- **Verificar datos:** Al presionar el botón de “Ingresar” se debe buscar al usuario en la base de datos y devolver si existe o no.

IMAGEN 1

```

class LoginScreenState extends State<LoginScreen> {
  Widget build(BuildContext context) {
    // ignore
    body: SingleChildScrollView(
      child: Padding(
        padding: const EdgeInsets.symmetric(
          horizontal: 25.0,
          vertical: 155.0,
        ), // EdgeInsets.symmetric
        child: Column(
          children: [
            TextFormField(
              controller: _emailController,
              decoration: const InputDecoration(
                contentPadding: EdgeInsets.symmetric(vertical: 10.0),
                icon: Icon(
                  Icons.person,
                ), // Icon
                hintText: "Usuario"), // InputDecoration
            ), // TextFormField
            const SizedBox(
              height: 10,
            ), // SizedBox
            TextFormField(
              controller: _passController,
              obscureText: isVisible,
              decoration: InputDecoration(
                icon: const Icon(Icons.key),
                hintText: "Contraseña",
                suffixIcon: IconButton(
                  icon: Icon(isVisible
                    ? Icons.visibility_outlined
                    : Icons.visibility_off_outlined), // Icon
                  onPressed: () => setState(() {
                    isVisible = !isVisible;
                  })),
            ), // IconButton
            ), // InputDecoration
            ), // TextFormField
            const SizedBox(
              height: 20,
            ), // SizedBox
            CustomButton(
              onPressed: () {
                login();
              },
              label: "Ingresar",
              width: MediaQuery.of(context).size.width,
              color: const Color(0xFF1A90D9),
            ) // CustomButton
          ],
        ), // Column
      ), // Padding
    ), // SingleChildScrollView
  ); // Scaffold
}

```

Crear función para obtener un usuario en Flutter

Requerimiento #####

Acción	- Crear función	Nombre	getUser(email, password)
---------------	-----------------	---------------	--------------------------

Definición de parámetros a recibir o utilizar:

- Database
- String email
- String password

Definición de lógica a realizar

- **Obtener un usuario:** Se consulta a la base de datos si es que existe un usuario según los datos ingresados, si es que existe entonces se devuelve el usuario, en caso contrario se devuelve un nulo.

IMAGEN 1

```
Future<Map<String, Object?>>> getUser(String email, String password) async {
  final db = await database;
  final user = await db.query(
    'users',
    where: 'email = ? AND password = ?',
    whereArgs: [email, password],
  );

  if (user.isNotEmpty) {
    return user.first;
  } else {
    return null;
  }
}
```

Crear función de inicio de sesión en Flutter

Acción	- Crear función	Nombre	logIn()
---------------	-----------------	---------------	---------

Definición de parámetros a recibir o utilizar:

- database

Definición de lógica a realizar

- **Inicio de sesión:** Se consulta la función creada anteriormente para verificar si es que el usuario existe en la base de datos, si es que no existe se muestra un mensaje de error indicando que los datos ingresados son incorrectos, en caso contrario se redirige a la vista de la actividad 1.

IMAGEN 1

```

void logIn() async {
  final user =
    await database.getUser(_emailController.text, _passController.text);
  if (user != null) {
    if (mounted) {
      Navigator.push(
        context,
        MaterialPageRoute(
          builder: (context) {
            return const MainScreen();
          },
        ), // MaterialPageRoute
      );
    }
  } else {
    if (mounted) {
      MessagesStatus.showStatusMessage(
        context, "Correo y/o contraseña incorrectos", true);
    }
  }
}

```

Crear mensaje de información en Flutter

Acción	- Crear función	Nombre	showStatusMessage(context, status, isError)
---------------	-----------------	---------------	---

Definición de parámetros a recibir o utilizar:

- BuildContext context
- String status
- bool isError

Definición de lógica a realizar

- **Mensaje de información:** Se debe crear un Snackbar que muestre al usuario información sobre el estado en caso de un inicio de sesión incorrecto.

IMAGEN 1

```
static void showStatusMessage(
    BuildContext context, String status, bool isError) {
    final scaffold = ScaffoldMessenger.of(context);

    scaffold.showSnackBar(
        SnackBar(
            behavior: SnackBarBehavior.floating,
            shape: RoundedRectangleBorder(
                borderRadius: BorderRadius.circular(10.0),
            ), // RoundedRectangleBorder
            backgroundColor: isError ? Colors.red[300] : Colors.green[300],
            content: Text(
                status,
                style: const TextStyle(color: Colors.white, fontSize: 15),
            ), // Text
            action: SnackBarAction(
                label: 'Ok', onPressed: scaffold.hideCurrentSnackBar), // SnackBarAction
        ), // SnackBar
    );
}
```