

# TAREA - GESTIÓN DE TRABAJADORES

En esta tarea pondremos en práctica todo lo que hemos visto en clase a lo largo de estas semanas.

La tarea consta de varias partes que finalmente conectaremos entre ellas.:

1. **Creación de una interfaz de gestión de trabajadores.**
  - a. Utilizando datos en local.
  - b. Utilizando una base de datos externa (Supabase) para hacer persistente los datos.
2. **Creación de interfaces de diálogo para gestión de logueo / registro.**
3. **Integrar la gestión de acceso a la interfaz en la interfaz principal de gestión de trabajadores.**

## 1. CREACIÓN DE INTERFAZ PRINCIPAL: GESTIÓN DE TRABAJADORES.

Crear un programa para gestionar altas de trabajadores de una empresa ficticia.

La ventana empleada será la siguiente:

Sesión Acciones Modo

Identificación del trabajador

DNI 12345678A

Nombre Luis

Apellido 1 Perez

Apellido 2 Perez

Provincia del trabajador

PONTEVEDRA

Eliminar provincia

Añadir provincia

Profesión del trabajador

Carpintero

Dentista

Eliminar profesión

Añadir profesión

Añadir trabajador

Trabajadores disponibles

Nombre y apellidos	Provincia	Profesión
Luis Perez Perez	PONTEVEDRA	CARPINTERO

Eliminar trabajador

Detalle del trabajador

12345678A - Luis Perez Perez - PONTEVEDRA - CARPINTERO

Fija un tamaño mínimo para la interfaz que asegure que todos los componentes se ven correctamente:

The interface is titled 'Sesión Acciones Modo'. It features a left sidebar with the following sections:

- Identificación del trabajador:** Fields for DNI (12345678A), Nombre (Luis), Apellido 1 (Perez), and Apellido 2 (Perez).
- Provincia del trabajador:** A dropdown menu showing 'PONTEVEDRA' and buttons for 'Eliminar provincia' and 'Añadir provincia'.
- Profesión del trabajador:** A list box containing 'Carpintero' and 'Dentista', with an 'Eliminar profesión' button and an 'Añadir profesión' button at the bottom.
- Añadir trabajador:** A large blue button at the bottom of the sidebar.

The right main area is titled 'Trabajadores disponibles' and contains a table with the following data:

Nombre y apellidos	Provincia	Profesión
Luis Perez Perez	PONTEVEDRA	CARPINTERO

Below the table is an 'Eliminar trabajador' button. At the bottom of the right section is a 'Detalle del trabajador' box showing the text: '12345678A - Luis Perez Perez - PONTEVEDRA - CARPINTERO'.

Y cuando ampliamos toda la interfaz se debería ver así:

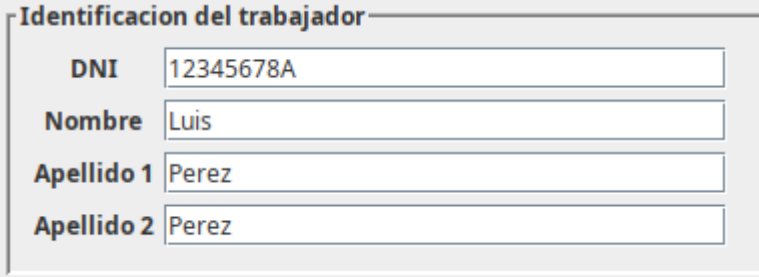
This screenshot shows the same interface as the previous one, but scaled to a larger size. The layout remains consistent, with the left sidebar and right main area clearly visible. The table in the 'Trabajadores disponibles' section still contains the same data: Luis Perez Perez, PONTEVEDRA, and CARPINTERO. The 'Detalle del trabajador' box at the bottom right still displays the same text: '12345678A - Luis Perez Perez - PONTEVEDRA - CARPINTERO'. The overall structure and components are preserved, demonstrating the importance of a minimum size for the interface.

Por lo tanto a hora de diseñar la interfaz evita utilizar el layout “absolute”, en su lugar, utiliza los layouts más convenientes para esta interfaz. Permitiendo ver correctamente el menú y dos partes de la interfaz claramente diferenciadas:

- En la parte izquierda podemos añadir un nuevo trabajador/provincia/trabajo.
- En la parte derecha para mostrar y/o eliminar un trabajador.

## 1.1. Partes y validaciones necesarias.

### 1.1.1. Identificación del trabajador.



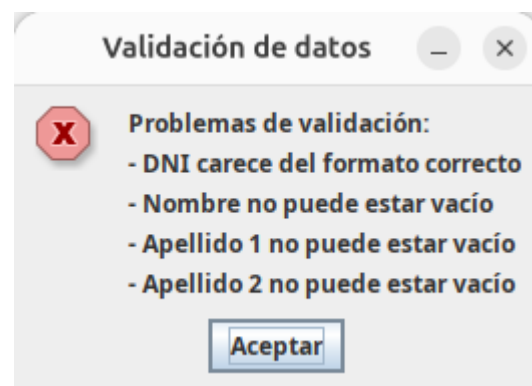
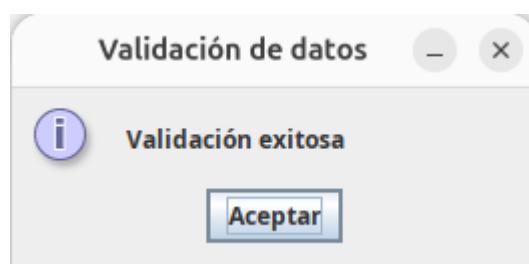
Formulario de identificación del trabajador con los siguientes campos:

DNI	12345678A
Nombre	Luis
Apellido 1	Perez
Apellido 2	Perez

Campos de texto para ingresar DNI, nombre y apellidos.

- Si uno de estos campos está vacío el botón de “Añadir trabajador” estará deshabilitado.
- VALIDACIÓN:** Cuando le demos al botón de “Añadir trabajador” o “Acciones>Trabajador>Validación de datos del trabajador” se inicia un proceso de validación de los campos de este formulario:
  - DNI:** Debe tener un formato correcto - 8 dígitos y una letra “MAYÚSCULA”. A mayores (y como extra) podemos comprobar que el “dígito de control” es correcto.  
<https://www.interior.gob.es/opencms/en/servicios-al-ciudadano/tramites-y-gestiones/dni/calculo-del-digito-de-control-del-nif-nie/>
  - Nombre, Apellido1 y Apellido 2** no puede estar vacío (**DEBE TENER AL MENOS UN CARÁCTER - no sólo espacios**), ni debe contener ningún número, sólo letras.

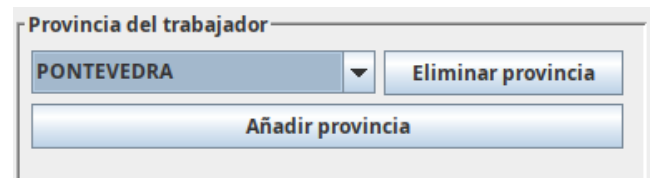
El resultado será una ventana emergente indicando si la validación ha sido satisfactoria o surgió algún problema, en ese caso indicar que ha pasado.



### 1.1.2. Provincia del trabajador.



The screenshot shows a window titled "Provincia del trabajador". It contains a dropdown menu that is currently empty, a button labeled "Eliminar provincia" to its right, and a button labeled "Añadir provincia" below the dropdown.



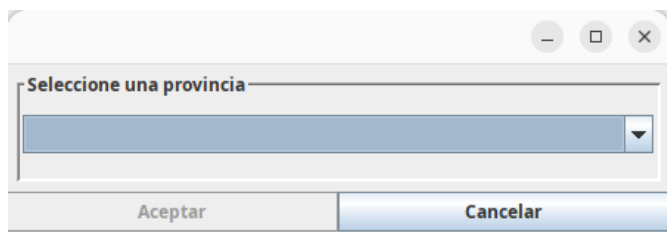
The screenshot shows the same window, but now the dropdown menu displays "PONTEVEDRA". The "Eliminar provincia" button is still present to the right, and the "Añadir provincia" button is below.

Al iniciar / resetear la interfaz, no debe tener seleccionado ningún elemento de la lista desplegable. En la lista desplegable tendremos las posibles provincias del trabajador.

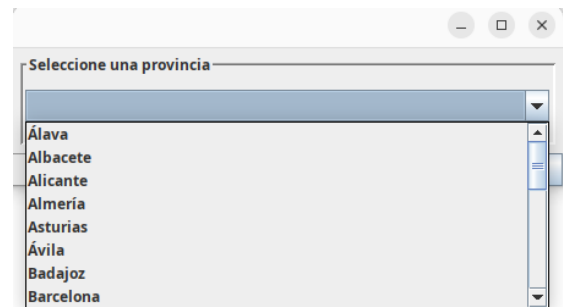
Este listado se construye dinámicamente mediante los botones de “Añadir provincia” y “Eliminar provincia”.

#### AÑADIR PROVINCIA:

1. **VENTANA DIÁLOGO PROVINCIAS.** Si pulsamos “añadir provincia” se **despliega una nueva ventana de diálogo** que incluye un conjunto extenso de provincias (todas las de España).



The screenshot shows a dialog box titled "Seleccione una provincia". It has a dropdown menu that is currently empty. At the bottom, there are two buttons: "Aceptar" and "Cancelar".



The screenshot shows the same dialog box, but the dropdown menu is now expanded, displaying a list of Spanish provinces: Álava, Albacete, Alicante, Almería, Asturias, Ávila, Badajoz, and Barcelona.

Fig. Ventana diálogo destina a añadir nuevas provincias al listado de la ventana principal

2. Al seleccionar una de estas podemos añadirla en el desplegable de provincias del trabajador, con el botón de “Aceptar”.  
**OJO:** Este botón está deshabilitado hasta que seleccionemos una provincia.
3. En caso de pulsar cancelar no se añade nada al desplegable de la ventana principal.

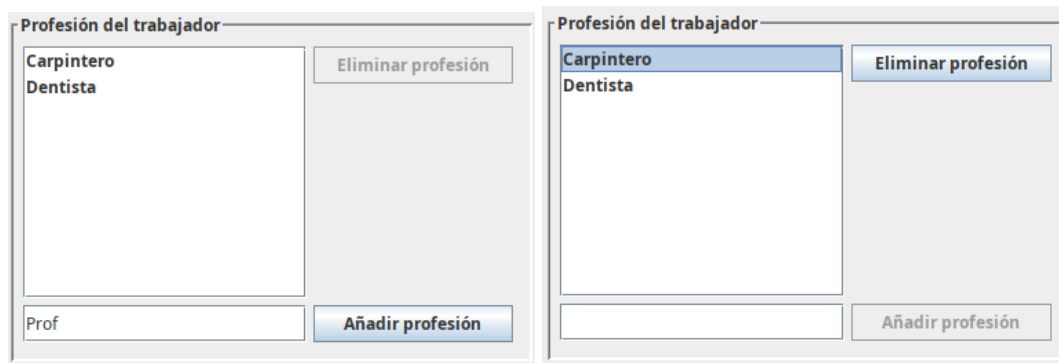
#### ELIMINAR PROVINCIA:

Si está seleccionada una provincia se puede eliminar del listado de provincias disponibles del trabajador, sino el botón estará deshabilitado.

// Provincias españolas

```
private final String[] provincias = {  
    "Álava", "Albacete", "Alicante", "Almería", "Asturias", "Ávila",  
    "Badajoz", "Barcelona", "Burgos", "Cáceres", "Cádiz", "Cantabria",  
    "Castellón", "Ciudad Real", "Córdoba", "Cuenca", "Girona", "Granada",  
    "Guadalajara", "Guipúzcoa", "Huelva", "Huesca", "Islas Baleares", "Jaén",  
    "La Coruña", "La Rioja", "Las Palmas", "León", "Lleida", "Lugo",  
    "Madrid", "Málaga", "Murcia", "Navarra", "Ourense", "Palencia",  
    "Pontevedra", "Salamanca", "Santa Cruz de Tenerife", "Segovia",  
    "Sevilla", "Soria", "Tarragona", "Teruel", "Toledo", "Valencia",  
    "Valladolid", "Vizcaya", "Zamora", "Zaragoza", "Ceuta", "Melilla"  
};
```

### 1.1.3. Profesión del trabajador.



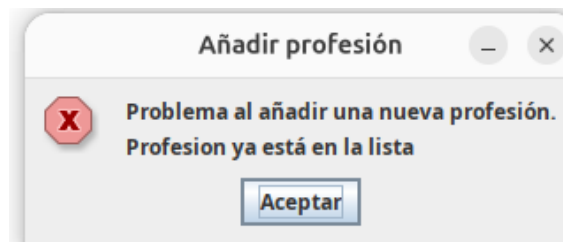
En este caso, usaremos una **lista desplegable [JList]** (ojo recuerda incluir el “scroll”).

#### **AÑADIR:**

En esta ocasión se añade elemento a lista directamente desde la interfaz principal.

Si el campo de texto no está vacío (contiene al menos un carácter alfanumérico) se puede añadir esa profesión.

**OJO! No puede estar duplicada una profesión en la lista, si se intenta añadir una profesión que ya está en la lista saldrá un aviso.**



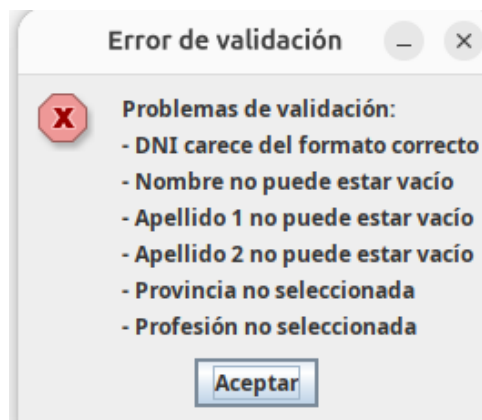
#### **ELIMINAR:**

Si pulsamos en un elemento de la lista podemos eliminar dicha profesión, mediante el botón de “Eliminar profesión” que sólo estará disponible cuando una profesión esté seleccionada.

### 1.1.4. BOTÓN AÑADIR TRABAJADOR.

Está disponible únicamente cuando todos los campos están cubiertos / seleccionados.

Al pulsar sobre el botón se iniciará un **proceso de validación**, en caso de no superar la validación nos indicará a qué se debe el fallo:



**2º Validación:** A mayores también se comprobará que el DNI del usuario no está ya registrado como un trabajador, siendo un campo que tiene ser único dentro del listado de trabajadores.

### 1.1.5. Trabajadores disponibles y detalles del trabajador.

The image displays two versions of a web interface side-by-side, illustrating the state of a worker management system before and after selecting a worker.

**Left Screenshot (Initial State):**

- Trabajadores disponibles:** A table with 3 columns: 'Nombre y apellidos', 'Provincia', and 'Profesión'. It contains one row: 'Luis Perez Perez', 'PONTEVEDRA', 'CARPINTERO'.
- Eliminar trabajador:** A button that is disabled (grayed out).
- Detalle del trabajador:** A section with a header and an empty content area.

**Right Screenshot (Selected State):**

- Trabajadores disponibles:** The same table as the left, but the first row is highlighted in blue, indicating it is selected.
- Eliminar trabajador:** A button that is now enabled (blue).
- Detalle del trabajador:** The section now displays the selected worker's information: '12345678A - Luis Perez Perez - PONTEVEDRA - CARPINTERO'.

Fig. Tabla de trabajadores disponible cuando seleccionamos o uno a un trabajador.

Tabla que agrupa todos los trabajados disponibles de la empresa.

**OJO! Hay que incluir también el scroll.**

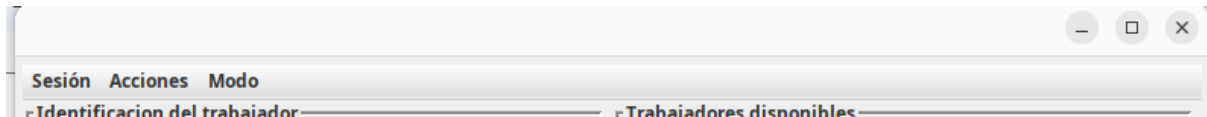
Cuando no hemos seleccionado a ningún trabajador el botón de eliminar estará deshabilitado.

Al seleccionar un trabajador, a parte de poder eliminarlo del listado, obtendremos un extracto de toda la información del trabajador:

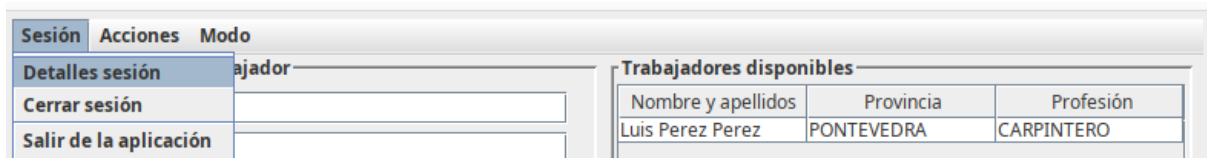
DNI - Nombre y apellidos - Provincia - Profesión.

## 1.2. Menú de la interfaz.

Consta de 3 menús principales:

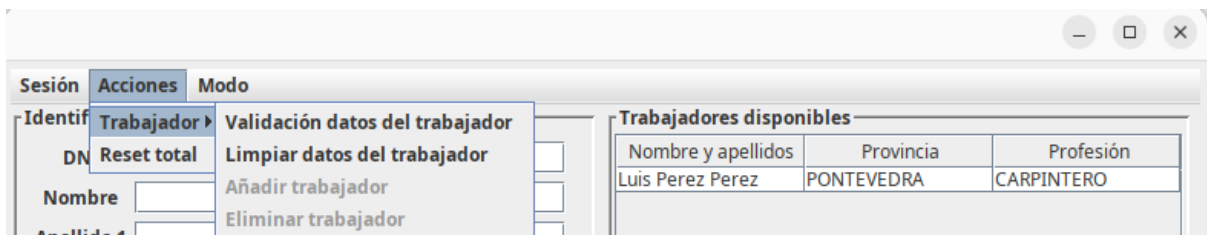


### 1.2.1 Sesión.



- **Detalles de sesión.** Nos indicará con que email y access\_token hemos entrado en la interfaz principal (*lo veremos cuando integremos los diálogos de login/registro con esta interfaz de gestión de trabajadores*).
- **Cerrar sesión.** Volvemos a la ventana de login/registro (*se pierde el access\_token, el usuario tiene que volver a loguearse*).
- **Salir de la aplicación.** Cerramos totalmente la aplicación.

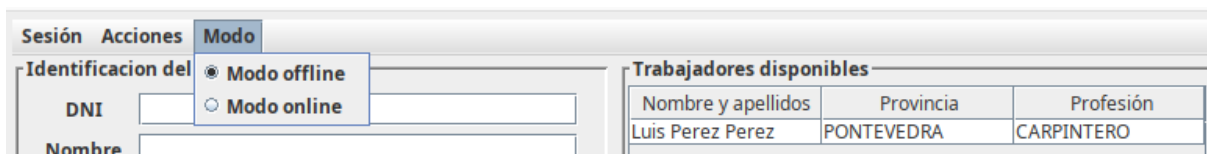
### 1.2.2. Acciones



- **Validación datos del trabajador.** Ejecuta el proceso de validación de los datos de identificación del trabajador, pero no envía la información únicamente informativo.
- **Limpiar datos del trabajador.** Borra los campos de datos del aparato de "Identificación del trabajador".
- **Añadir y eliminar trabajador.** Actúan sobre los mismos botones de la interfaz principal. Estarán disponibles o no disponibles siguiendo el mismo comportamiento que los botones en la interfaz principal. *Utilizar método doClick del botón:*

```
btnNewButton.doClick();
```

### 1.2.3. Modo



Permite cambiar la gestión de los datos de la interfaz de modo.

**Modo offline:** Utilizamos datos en local únicamente.

**Modo online:** Utilizamos y actualizamos una base externa en este caso de la plataforma de Supabase.

### 1.3. DATOS EN LOCAL. - MODO OFFLINE

Guarda la información en memoria (local) mediante las estructuras de datos que se consideren oportunas (vistas en las prácticas/actividades anteriores). La única información persistente es el listado completo de provincias que utilizaremos en el diálogo de selección de provincias.

**NOTA:** En este punto, no se utilizaría ninguna base de datos por lo que al cerrar la aplicación se perderá la información.

### 1.4. COMUNICACIÓN CON SUPABASE. - MODO ONLINE.

La interfaz funcionará exactamente igual pero nos apoyaremos en una base de datos que previamente hemos configurado en la plataforma Supabase. Estructura de tablas:

- **Trabajador.** Listado de trabajadores de la empresa.
- **Profesiones.** Listado disponible de profesiones que podemos seleccionar en la interfaz.
- **Provincias.** Listado disponible de provincias (recogidas en la interfaz principal).
- **ListadoCompletoProvincia.** Listado de todas las provincias posibles, esta lista es necesaria para la ventana de diálogo “de provincias”. De este listado podemos escoger las provincias que incluiremos en el comboBox de la interfaz principal.



Columns				About data types
Name	Type	Default Value	Primary	
dni	T text	NULL	<input checked="" type="checkbox"/>	
nombre	T text	NULL	<input type="checkbox"/>	
apellido1	T text	NULL	<input type="checkbox"/>	
apellido2	T text	NULL	<input type="checkbox"/>	
provincia	text	NULL	<input type="checkbox"/>	
profesion	text	NULL	<input type="checkbox"/>	
Add column				
Foreign keys				
Trabajador_profesion_fkey				
Foreign key relation to: public.Profesiones				
profesion → public.Profesiones.profesion				
Trabajador_provincia_fkey				
Foreign key relation to: public.Provincias				
provincia → public.Provincias.provincia				

Columns				About data types
Name	Type	Default Value	Primary	
provincia	T text	NULL	<input checked="" type="checkbox"/>	

Columns				About data types
Name	Type	Default Value	Primary	
profesion	T text	NULL	<input checked="" type="checkbox"/>	

Para comunicarte con la base de datos puedes utilizar el código proporcionado en el aula virtual. [SupabaseDB.java](#) y [SupabaseConfig.java](#).

**NOTA.** Para que sea más fácil trabajar con las tablas, en Supabase podemos deshabilitar RLS sino tendríamos que configurarlo correctamente para poder acceder. Ver: **ANEXO. DESHABILITAR RLS EN LAS TABLAS DE SUPABASE.**

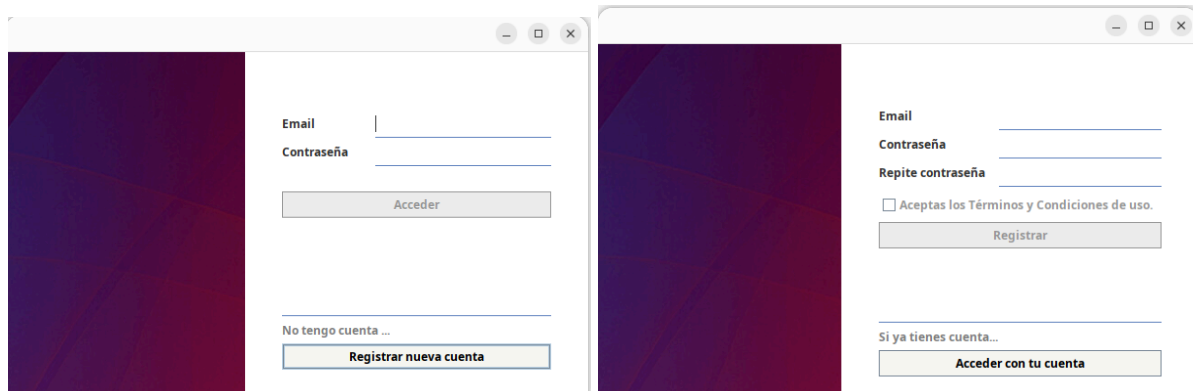


## 2. Creación de interfaces de diálogo para gestión de logueo / registro.

En este apartado llevaremos a cabo los siguientes pasos:

1. Creación de un wireframe de las ventanas de login / registro.
2. Creación de dos JDialog uno para el acceso y otro para el registro.
3. Comunicar con la plataforma de Supabase y su servicio gratuito de autenticación.  
Para ello puedes utilizar [SupabaseAuth.java](#) y [SupabaseConfig.java](#)

Si estás sin ideas, a la hora de diseñar tus interfaces, te muestro 2 posibles interfaces:



### 2.1. REQUISITOS.

- Dos diálogos diferentes: uno para la gestión del acceso y otro para el registro.
- Desde el diálogo de “login” se debe acceder al diálogo de “registro” y viceversa.
- Gestionar todas los posibles problemas a la hora de acceder o acceder.

**Nota:** Si el login o registro se ha hecho satisfactoriamente nos debería llegar un “access\_token” único & temporal que identifica de forma única al usuario dentro de la plataforma. Si no nos llega puede deberse a diversos errores, el código proporcionado muestra por terminal que errores van surgiendo, trata de personalizar diálogos emergentes de aviso para dar un feedback lo más completo posible al usuario de que está sucediendo.

- **Opcional:** Deshabilitar los botones cuando no hay datos.
  - **Ventana de acceso** - si no está cubierto los campos de email & contraseña deshabilitar botón.
  - **Ventana de registro** - si no está cubierto los campos de texto no habilitar el checkbox para aceptar los términos y condiciones de uso, a su vez hasta que no se active este campo no se habilita el botón de registro.

**Nota:** Recuerda que utilizamos un componente específico cuando trabajamos con contraseñas.

### **3. INTEGRAR EN LA INTERFAZ PRINCIPAL (GESTIÓN DE TRABAJADORES) LA GESTIÓN DEL ACCESO AL SERVICIO.**

En este punto deberíamos tener tanto la interfaz principal como las ventanas de diálogo de acceso/registro funcionales.

En este último punto debes integrar todas en una única aplicación.

Por ello:

1. Iniciamos la interfaz principal, sino no tenemos un access token operativo lanzamos el diálogo de "login".
2. En el caso de obtener un access\_token (con alguno de los dos diálogos) abrimos la ventana principal y le enviamos información del email & access\_token - información disponible / visible mediante el botón del menú: Sesión > Detalles sesion.
3. Desde la interfaz principal podemos salir de la sesión actual. En ese caso se pierde el access\_token y retornamos al diálogo de "login".

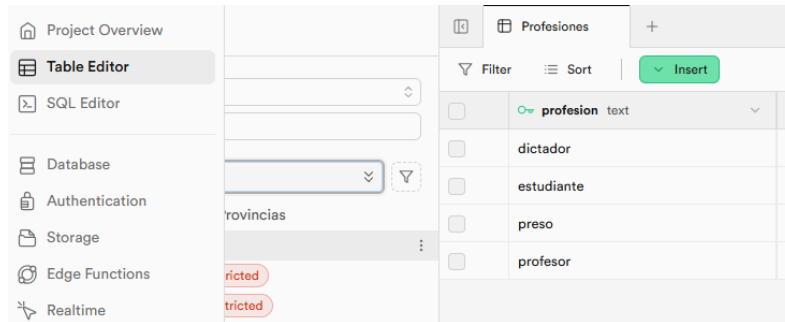
A tener en cuenta:

1. Solo tendremos un diálogo de login, registro en todo momento y no permitir abrir más de un diálogo. Lo mismo con la interfaz principal, sólomente será una.

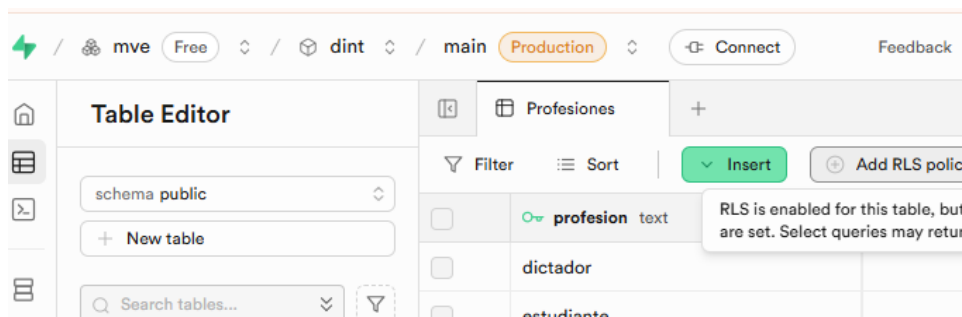
## ANEXO. DESHABILITAR RLS EN LAS TABLAS DE SUPABASE.

Dentro del editor de tablas, pulsaremos sobre cada una de las tablas y deshabilitaremos RLS. **Pasos:**

1. Acceder a supabase, entramos en Table editor



2. Pulsamos sobre la tabla y en Add RLS policy > Disable RLS y confirmamos el cambio.



3. Desprotegemos la tabla, pero su gestión/manejo será mucho más sencillo.

