

Proyecto

Gestor de Usuario en Qt



Karen Acosta Tse
Edgar Javier Quesada Morales
Nayra Ramírez Jorge

Índice

1. Desarrollo de la base de datos	3
1.1 Creación de las tablas	3
1.2 SQL Generar consultas necesarias.....	3
2. Análisis y diseño	5
2.1 Creación diagrama entidad-relación de la base de datos	5
2.2 Creación diagrama de clases de la aplicación	6
2.3 Propuesta de la interfaz gráfica	7
3. Desarrollo de la aplicación	9
3.1 Creación de datos y objetos	9
3.1.1 Lógica	9
3.1.2 Tipos de usuarios	9
3.2 Generación de modelos necesarios	10
3.3 Desarrollo de la interfaz gráfica	12

1. Desarrollo de la base de datos

Se ha creado una base de datos relacional compuesta por tres tablas principales: **empresa**, **departamento** y **usuario**. Estas están relacionadas mediante claves foráneas para mantener la integridad entre los datos.

La tabla *empresa* almacena los datos de cada empresa; *departamento* se vincula a una empresa mediante *id_empresa*; y *usuario* contiene la información de cada usuario, asociado a un departamento mediante *id_dpto*.

Además de la creación de tablas, se han generado consultas SQL para insertar datos de prueba y visualizar toda la información unificada a través de sentencias JOIN.

1.1 Creación de las tablas

Creación de la tabla *empresa*:

```
CREATE TABLE "empresa" ("id_empresa" INTEGER, "nombre" TEXT NOT NULL, PRIMARY KEY("id_empresa" AUTOINCREMENT))
```

Creación de la tabla *departamento*:

```
CREATE TABLE "departamento" ("id_dpto" INTEGER, "nombre" TEXT NOT NULL, "id_empresa" INTEGER, PRIMARY KEY("id_dpto" AUTOINCREMENT), FOREIGN KEY("id_empresa") REFERENCES "empresa")
```

Creación de la tabla *usuario*:

```
CREATE TABLE "usuario" ("id_usuario" INTEGER, "nombre" TEXT NOT NULL, "apellidos" INTEGER NOT NULL, "dni" TEXT NOT NULL UNIQUE, "telefono" TEXT, "email" TEXT UNIQUE, "estado" TEXT, "id_dpto" INTEGER, "puesto" TEXT, "contrasenha" TEXT, PRIMARY KEY("id_usuario" AUTOINCREMENT), FOREIGN KEY("id_dpto") REFERENCES "departamento")
```

1.2 SQL Generar consultas necesarias

Insertar valores en la tabla *empresa*:

```
INSERT INTO empresa (nombre) VALUES ('Aerolaser');  
SELECT * FROM empresa;
```

Insertar valores en la tabla *departamento*:

```
INSERT INTO departamento (nombre) VALUES ('Informatica');  
SELECT * FROM departamento;
```

Insertar valores en la tabla *usuario*:

```
INSERT INTO usuario (nombre, apellidos, dni, telefono, email, estado, puesto, id_dpto, contrasenha)  
VALUES ('David', 'Leon', '1111111A', '111222333', 'davidleon@gmail.com', 'vacaciones', 'trabajador',  
1, '1234');  
SELECT * FROM usuario;
```

Mostrar todos los datos en una sola tabla:

SELECT e.id_empresa, e.nombre AS empresa, d.id_dpto, d.nombre AS departamento, u.id_usuario, u.nombre AS nombre_usuario, u.apellidos, u.dni, u.telefono, u.email, u.estado, u.puesto FROM empresa e JOIN departamento d ON e.id_empresa = d.id_empresa JOIN usuario u ON d.id_dpto = u.id_dpto;

```
1  INSERT INTO empresa (nombre) VALUES ('Aerolaser');
2  SELECT * FROM empresa;
3
4  INSERT INTO departamento (nombre) VALUES ('Informatica');
5  SELECT * FROM departamento;
6
7  INSERT INTO usuario (nombre, apellidos, dni, telefono, email, estado, puesto,
8  id_dpto, contraseña) VALUES ('David', 'Leon', '1111111A', '111222333',
9  'davidleon@gmail.com', 'vacaciones', 'trabajador', 1, '1234');
10 SELECT * FROM usuario;
11
12 SELECT e.id_empresa, e.nombre AS empresa, d.id_dpto, d.nombre AS departamento,
13 u.id_usuario, u.nombre AS nombre_usuario, u.apellidos, u.dni, u.telefono, u.email,
14 u.estado, u.puesto FROM empresa e JOIN departamento d ON e.id_empresa = d.id_empresa
15 JOIN usuario u ON d.id_dpto = u.id_dpto;
```

id_empresa	empresa	id_dpto	departamento	id_usuario	nombre_usuario	apellidos	dni	telefono	email	estado	puesto
1	Aerolaser	1	Informatica	1	David	Leon	1111111A	111222333	davidleon@gmail.com	vacaciones	trabajador

2. Análisis y diseño

En este apartado se abordan las fases de análisis y diseño de la aplicación, fundamentales para establecer una base sólida antes del desarrollo. Primero, se define la estructura de la base de datos mediante un diagrama entidad-relación, que permite visualizar las entidades principales y sus relaciones. A continuación, se presenta el diagrama de clases, el cual representa la estructura lógica de los objetos y su comportamiento dentro del sistema. Por último, se incluye una propuesta de interfaz gráfica que servirá como guía visual para el desarrollo del formulario de usuario, basada en una interfaz clara e intuitiva.

2.1 Creación de diagrama entidad-relación base de datos

Al comenzar con la creación del diagrama, realizamos un primer modelo pero al iniciar el desarrollo nos dimos cuenta que teníamos que realizar algunos cambios para mejorar el diagrama.

Diagrama E/R inicial

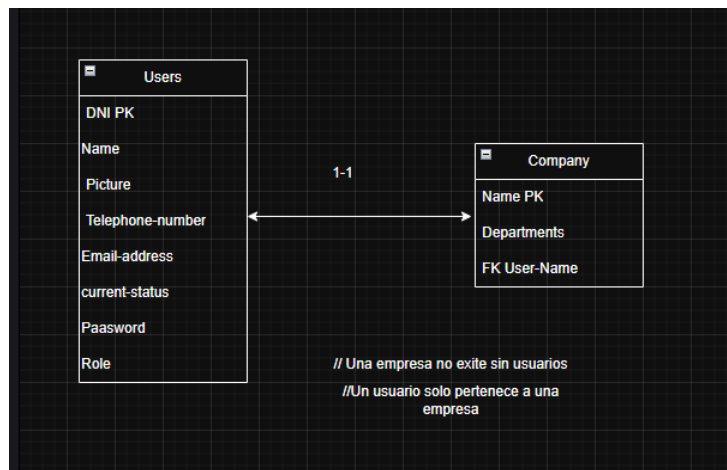
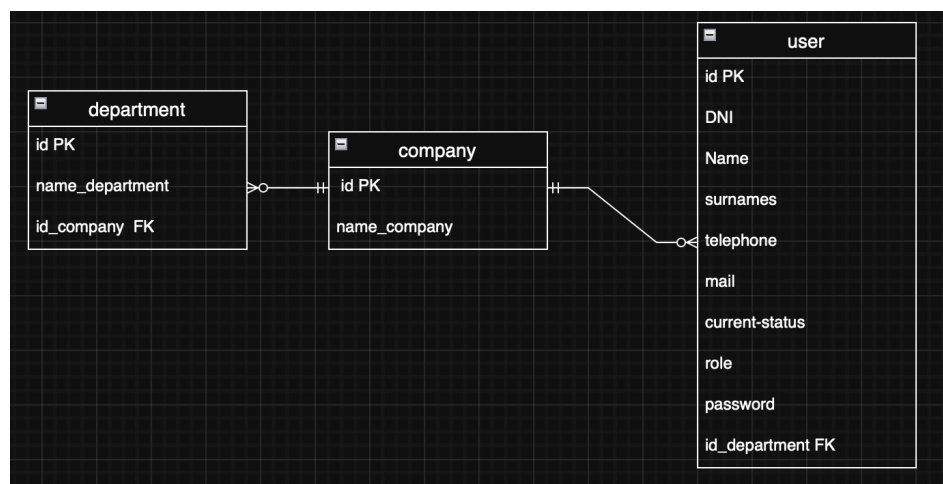


Diagrama E/R final



2.2 Creación diagrama de clases de la aplicación

En un principio realizamos un UML, el cual al final decidimos cambiar:

Diagrama UML inicial

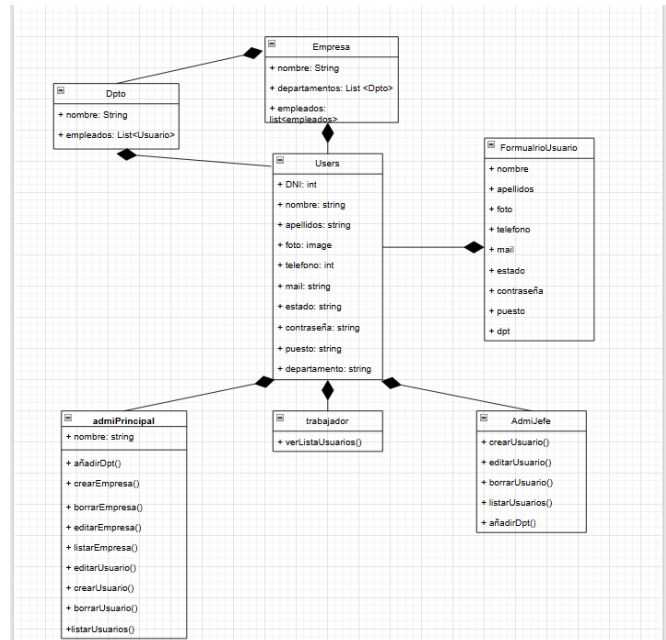
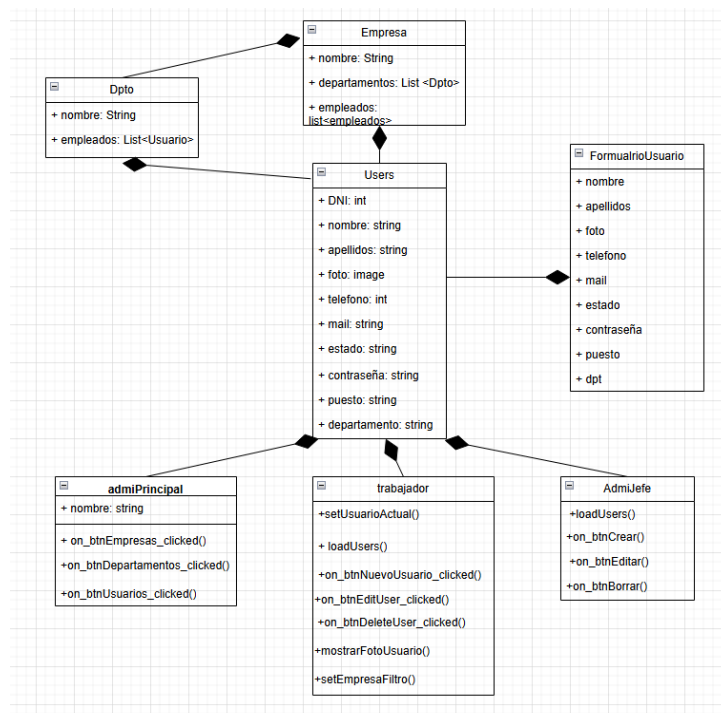


Diagrama UML final



2.3 Propuesta de interfaz gráfica

Como referencia para el diseño del formulario de creación de usuarios, se ha tomado como inspiración la interfaz de Windows Server 10. Esta interfaz se considera adecuada por su estructura clara, ordenada y fácil de entender.

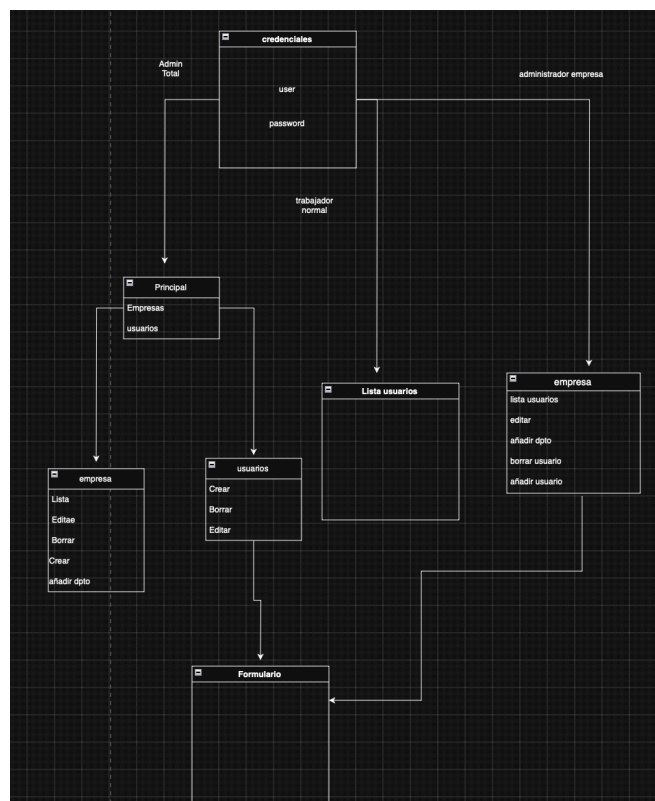
A continuación, se muestra una captura que ejemplifica el estilo visual que se busca adoptar, donde los campos están bien organizados y la navegación es intuitiva para el usuario.

La imagen muestra una ventana de diálogo titulada "Nuevo objeto: Usuario". En la parte superior, indica "Crear en: solvetic.com/Solvetic IT". El formulario contiene los siguientes campos:

- Nombre de pila:** Solvetic
- Iniciales:** (campo vacío)
- Apellidos:** Internet
- Nombre completo:** Solvetic Internet
- Nombre de inicio de sesión de usuario:** solvetic1 (con un menú desplegable a la derecha que muestra @solvetic.com)
- Nombre de inicio de sesión de usuario (anterior a Windows 2000):** SOLVETIC\ (con un campo adicional a la derecha que contiene solvetic1)

En la parte inferior, hay tres botones: "< Atrás", "Siguiente >" (destacado con un recuadro azul) y "Cancelar".

Propuesta de interfaz



Esta es la propuesta de la interfaz que permite que cada tipo de usuario solo acceda a los paneles necesarios para desarrollar su actividad. Posteriormente, en el *desarrollo de la aplicación*, llevamos a cabo un cambio, el cual el usuario no necesita contraseña, solo el nombre del usuario.

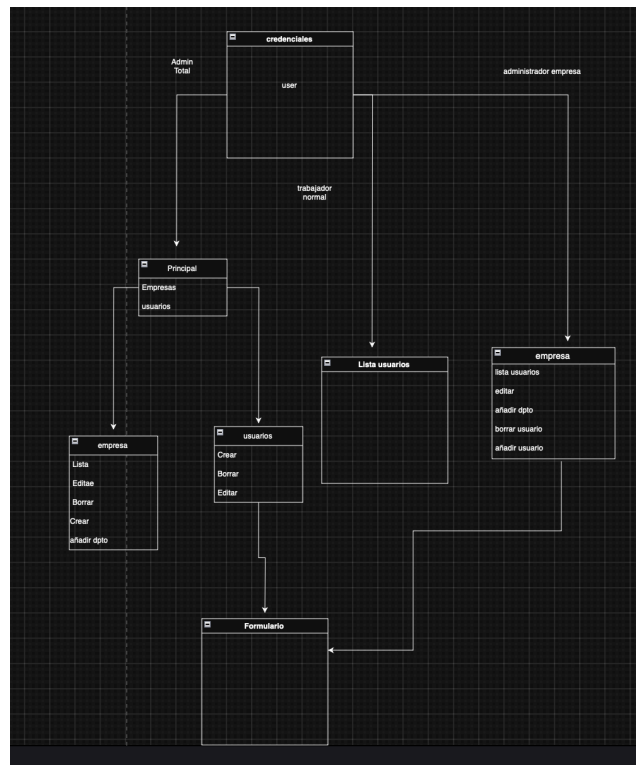
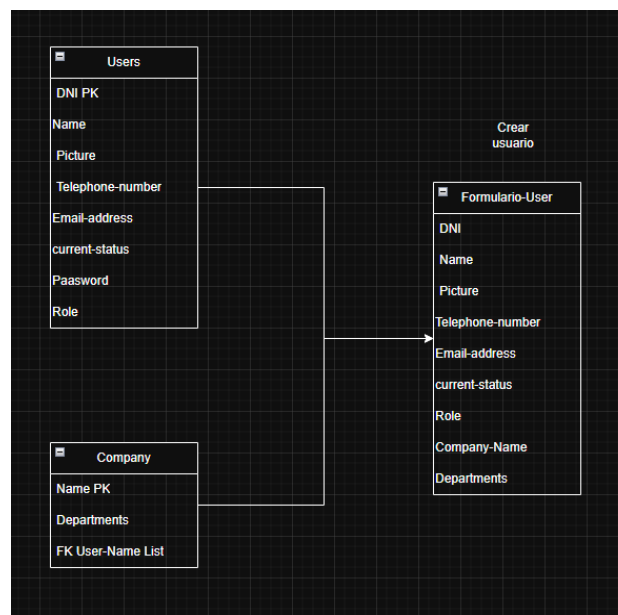


Diagrama de uso del formulario de creación de usuario



3. Desarrollo de la aplicación

En este apartado se detalla el proceso de desarrollo de la aplicación, comenzando con la definición y creación de los datos necesarios para su funcionamiento. Inicialmente se generan los objetos e instancias que permiten realizar pruebas básicas. A continuación, se desarrollan los modelos que estructuran la lógica del sistema, mediante la implementación de clases y relaciones previamente definidas. Finalmente, se construye la interfaz gráfica, siguiendo el diseño propuesto, con el objetivo de ofrecer una experiencia de usuario clara, intuitiva y bien organizada.

Para ello, se utilizó el diagrama de la interfaz como guía durante el desarrollo.

La estructura del proyecto se organizó en módulos: gestión de datos, lógica del sistema y presentación visual. Cada clase representa una entidad concreta (como usuarios, empresas o departamentos), y se establece una comunicación directa entre la lógica y la interfaz para responder a las acciones del usuario.

La información se almacena en una base de datos integrada, lo que permite utilizar la aplicación sin depender de conexiones externas.

3.1 Creación de datos y objetos

En esta sección se abordan los elementos fundamentales necesarios para el funcionamiento de la aplicación. Se definen los objetos principales que representan a las empresas, departamentos y usuarios, y se establecen sus relaciones lógicas y jerárquicas.

A partir de estos objetos se construye la lógica del sistema, basada en roles de usuario, y se organiza la interfaz gráfica para gestionar cada entidad de forma estructurada. Esta base permite realizar las validaciones, gestionar permisos y controlar el acceso a las distintas secciones del programa en función del tipo de usuario autenticado.

3.1.1 Lógica

Para la lógica del programa se definieron cuatro tipos de roles, de los cuales solo tres son roles reales que participan en el funcionamiento normal de la aplicación:

- **Admin_total:** Administrador total del sistema.
- **Admin_empresa:** Administrador específico de cada empresa cliente.
- **Usuario:** Usuario con acceso limitado, generalmente trabajador o usuario estándar.
- **rol_invalido:** Rol lógico utilizado para facilitar la programación. Cuando un usuario tiene asignado este rol, significa que no existe o que está deshabilitado.

3.1.2 Tipos de usuarios:

En nuestro programa tenemos 3 tipos de usuarios: *Administrador total* (administrador de la empresa que vende el software), *Administrador de Empresa* (administrador propio de cada empresa cliente), *Usuario estándar o trabajador* (poco acceso o ninguna a la gestión del programa).

- **Administrador total:** Es el administrador principal de la empresa. Este administrador puede:
 - Crear, editar, listar y borrar empresas.
 - Añadir departamentos a una empresa.
 - Crear usuarios mediante un formulario con los siguientes campos:
 - Nombre
 - Apellidos
 - Foto
 - Teléfono
 - Correo electrónico
 - Estado (activo/inactivo)
 - Contraseña
 - Puesto
 - Departamento
 - Editar y borrar usuarios, utilizando el mismo formulario para la edición.

- **Administrador de empresa:** Es el administrador propio de cada empresa cliente y puede:
 - Crear y editar usuarios de su empresa mediante el formulario antes descrito.
 - Borrar usuarios.
 - Añadir departamentos dentro de su empresa.
 - Gestionar la lista de usuarios asignados a su empresa.
 - Editar información de la empresa.

- **Usuario estándar:** Este tipo de usuario es un trabajador o usuario con acceso limitado, que generalmente no tiene permisos para gestionar el sistema. Para ellos, es importante que aparezca la lista de todos los usuarios dentro de su sección, para consultas o referencias.

3.2 Generación de modelos necesarios

El proyecto está organizado en una estructura modular que facilita el desarrollo y mantenimiento de la aplicación. La estructura general es la siguiente:

- Página principal: Contiene los archivos principales que gestionan la ventana principal y la interfaz general de la aplicación:
 - main.cpp
 - mainwindow.ui
 - mainwindow.h
 - mainwindow.cpp

- Models: Representan la lógica de negocio y las entidades centrales del sistema, incluyendo empresas, departamentos y usuarios:
 - empresa.h
 - empresa.cpp
 - departamento.h
 - departamento.cpp
 - usuario.h
 - usuario.cpp

- **Widgets:** Incluyen los formularios y componentes gráficos específicos para la interacción con el usuario, tales como:

- empresaForm.h	- empresaForm.cpp	- empresaForm.ui
- userForm.h	- userForm.cpp	- userForm.ui
- dptoForm.h	- dptoForm.cpp	- dptoForm.ui
- guess.h	- guess.cpp	- guess.ui
- MainAdmi.h	- MainAdmi.cpp	- MainAdmi.ui
- AdminEmpresa.h	- AdminEmpresa.cpp	- AdminEmpresa.ui
- mainwindow.h	- mainwindow.cpp	- mainwindow.ui
- users.h	- users.cpp	- users.ui

Además, el programa cuenta con métodos clave dentro de la clase *MainWindow* que permiten gestionar la lógica y la interacción con el usuario, entre ellos:

- **validarUsuario()** → Verifica si el usuario existe en la base de datos, si su cuenta está activa, y determina su rol. Devuelve un *ResultadoValidacion* que indica el estado (activo/inactivo/no existe) y el tipo de rol.
- **onbtnEntrarClicked()** → Este es el slot del botón "Entrar" (login). Muestra la pestaña correspondiente con *mostrarPestanaSegunRol()*.
- **mostrarPestanaSegundoRol()** → Solamente activa la pestaña correspondiente al rol del usuario autenticado, mediante:
 - Pestaña 1: Admin total.
 - Pestaña 2: Admin empresa.
 - Pestaña 3: Usuario estándar.
- **onBtnAdminEmpresaClicked()** → Muestra la ventana de gestión de empresas (*EmpresaForm*).
- **onBtnAdminDepartamentosClicked()** → Muestra la ventana para administrar departamentos (*DptoForm*).
- **onBtnAdminTrabajadoresClicked()** → Muestra la ventana para gestionar trabajadores o usuarios (*Users*).
- **onBtnAdminCompanyClicked()** → Muestra la ventana de *admicompany*, que también puede estar en la pestaña 2 para admin de empresa.

Esta estructura modular permite mantener una clara separación entre la lógica, la gestión de datos y la interfaz gráfica, facilitando el desarrollo escalable y el mantenimiento futuro.

3.3 Desarrollo de la interfaz gráfica

- **Explicación la estructura de la interfaz:**

Archivo	Descripción
mainwindow.ui	Pantalla de login
MainAdmi.ui	Panel principal del administrador total
AdminEmpresa.ui	Panel del administrador de empresa
users.ui	Vista general de usuarios
empresaForm.ui	Formulario para crear/editar empresas
usuarioForm.ui	Formulario para crear/editar usuarios
dptoForm.ui	Formulario para crear/editar departamentos

- **Explicación de lo qué hace cada ventana y su función.**

Ventanas principales:

- **MainWindow:** Pantalla de inicio de sesión.
- **MainAdmi:** Contiene pestañas y botones para administrar empresas, departamentos y usuarios.
- **AdminEmpresa:** Similar a MainAdmi pero restringido a usuarios de una empresa específica.
- **UserForm / EmpresaForm / DptoForm:** Formularios reutilizables para crear y editar elementos.

Campos de los formularios:

- **DNI**
- **Nombre:** incluyendo apellidos
- **Foto:** se carga desde imágenes del ordenador
- **Teléfono**
- **Email**
- **Departamento**
- **Empresa**
- **Estado:** activo/ inactivo/ suspendido
- **Rol:** (admi total, admi empresa o usuario)

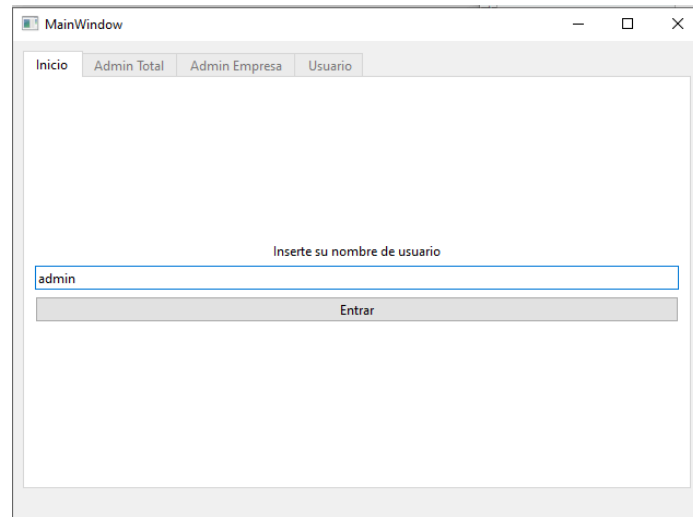
- **Explicación de la navegación entre ventanas y la gestión de pestañas según rol**

Al iniciar sesión, según el rol que tenga el usuario, se abrirá una pestaña específica, por ejemplo, si eres un administrador total tendrás acceso a todo mientras que el usuario solo puede visualizar su perfil.

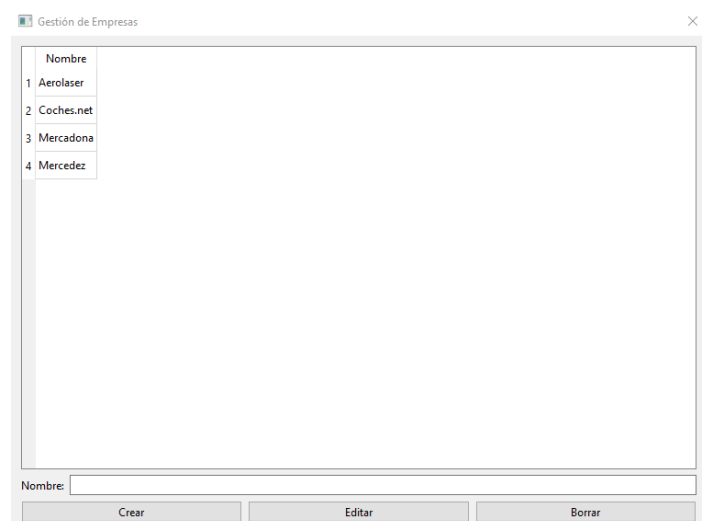
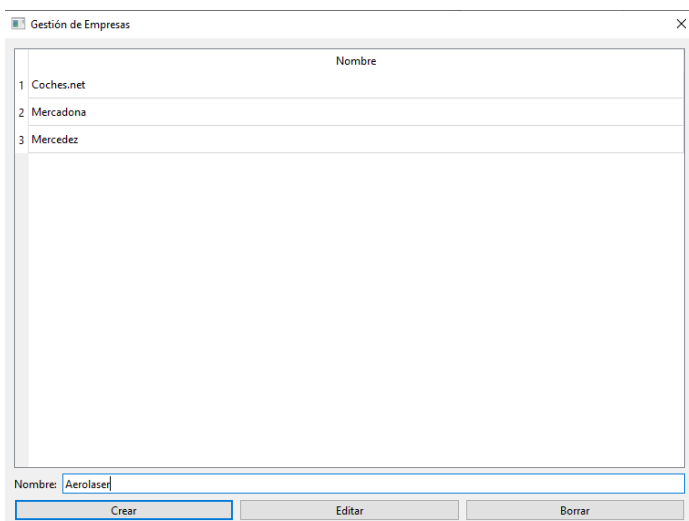
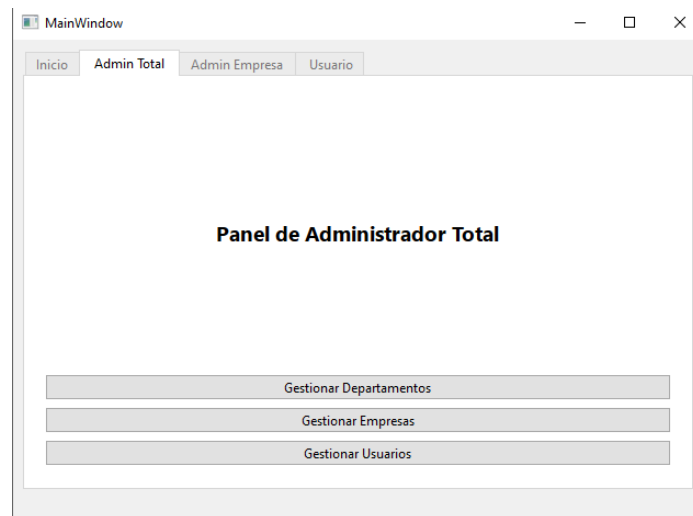
La parte final del proyecto es el desarrollo de la interfaz gráfica. Es la parte visual, la que verá el usuario, donde se encuentra la distribución de los botones interactivos del programa.

En las siguientes imágenes se puede ver como que la interfaz gráfica finalmente:

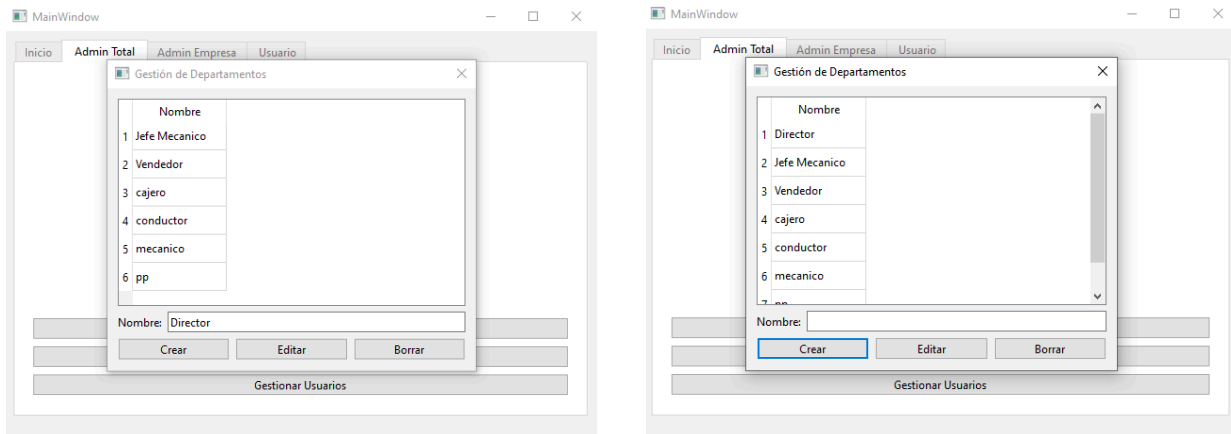
En este caso, al contar con acceso total, el administrador puede gestionar todos los niveles de la organización: empresas, departamentos y usuarios. Asimismo, la estructura de permisos se visualizará de la siguiente manera:



Al seleccionar "Gestionar empresa" y completar la creación de la empresa, la interfaz mostrará el siguiente diseño:



Por otro lado, si creamos un departamento la pestaña se verá de la siguiente manera:



Asimismo, para añadir un nuevo usuario, será necesario completar el siguiente formulario con toda la información requerida. Los campos a incluir son los siguientes:

DNI: 49518911A

Nombre: David

Teléfono: 111222333

Email: davidleon@gmail.com

Departamento: Director

Empresa: Aerolaser

Estado: Activo

Rol: Admin de Empresa

Foto (ruta): ptos de prueba para proyecto/diferencia-entre-gerente-y-director.jpg

Seleccionar Foto

Guardar Cancelar

Si el DNI introducido es incorrecto, la casilla se mostrará en rojo:

DNI: 49518911B

Nombre: David

Teléfono: 111222333

Email: davidleon@gmail.com

Departamento: Director

Empresa: Aerolaser

Estado: Activo

Rol: Admin de Empresa

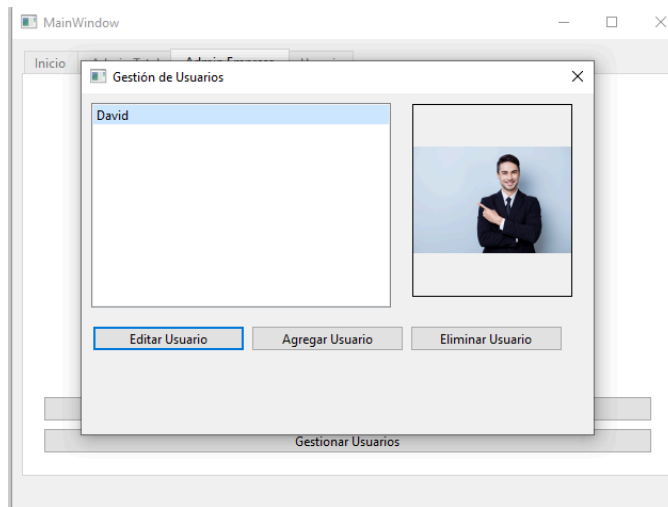
Foto (ruta):

Seleccionar Foto

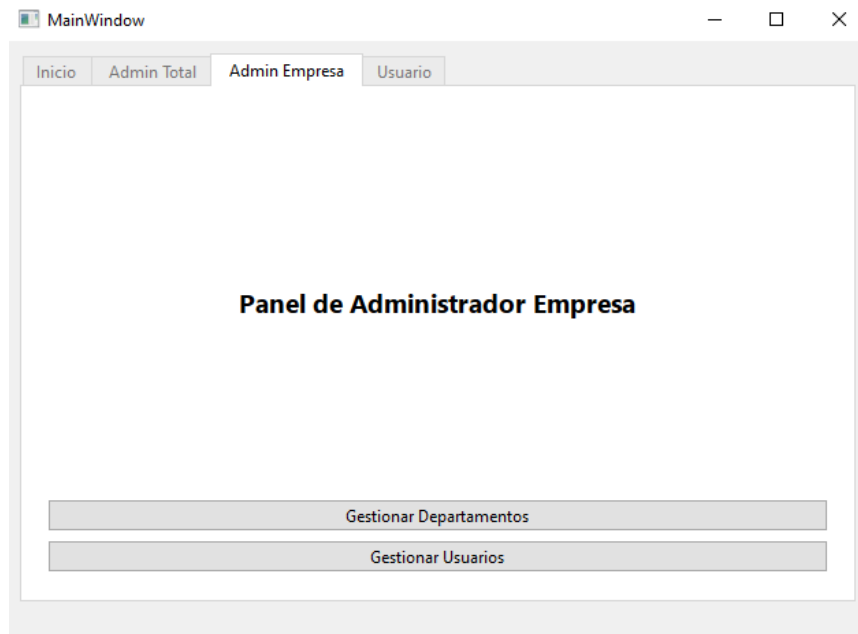
Previsualización

Guardar Cancelar

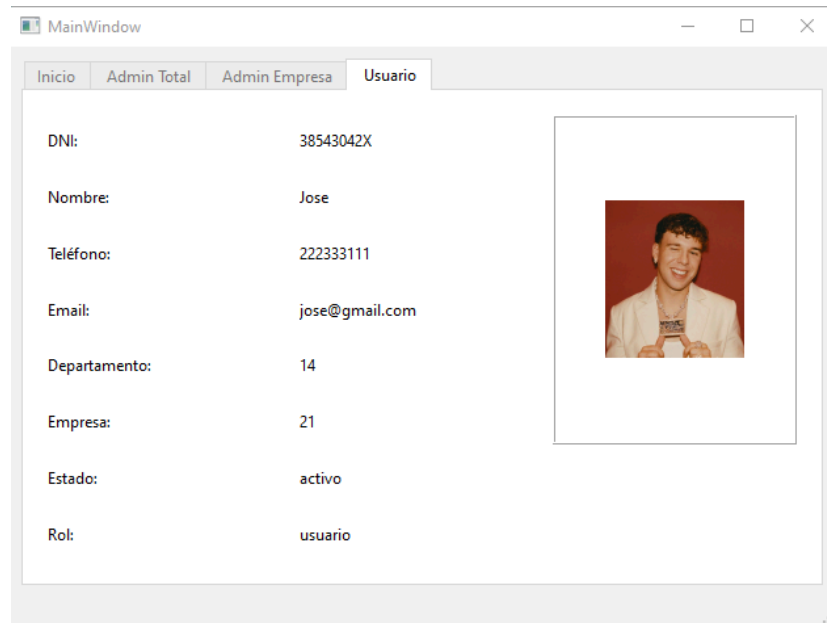
Asimismo, si pulsamos el botón de gestionar usuarios se verá de la siguiente forma:



Si iniciamos con David, al haber gestionado que sea admin de empresa solo tendrá derecho a gestionar los departamentos y los usuarios:



Por último, si se inicia sesión como un usuario solo podrá visualizar su propia información, la cual se verá de esta forma:



DNI:	38543042X
Nombre:	Jose
Teléfono:	222333111
Email:	jose@gmail.com
Departamento:	14
Empresa:	21
Estado:	activo
Rol:	usuario

