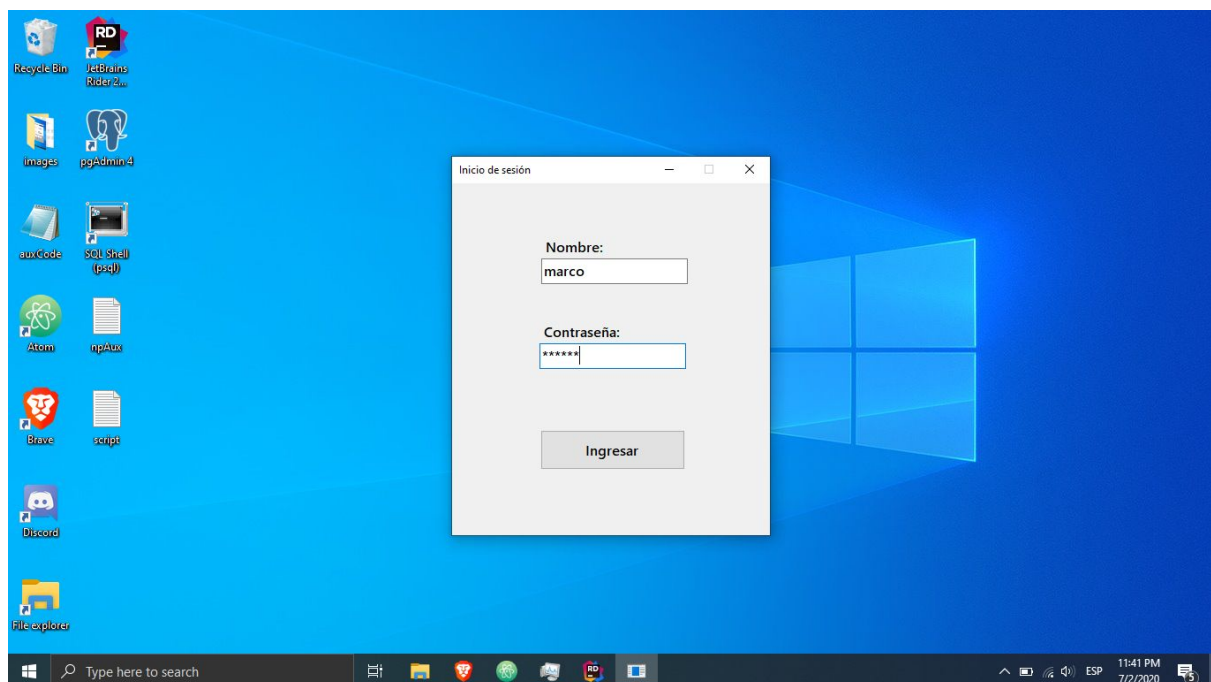
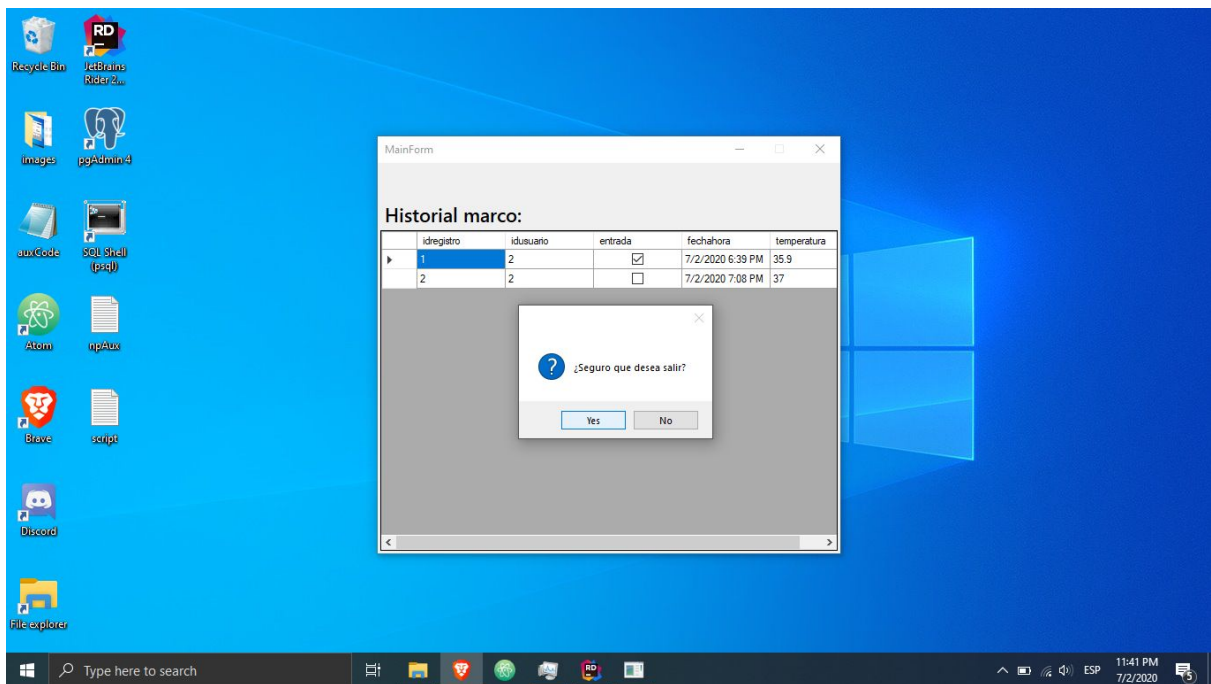
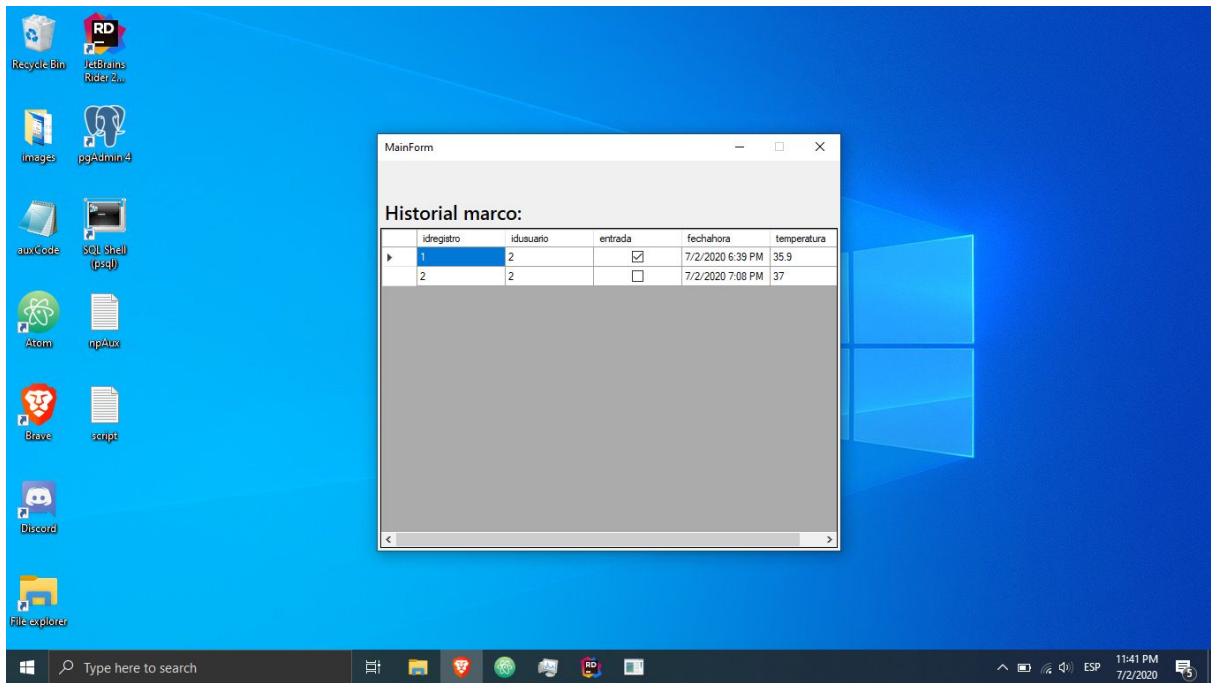


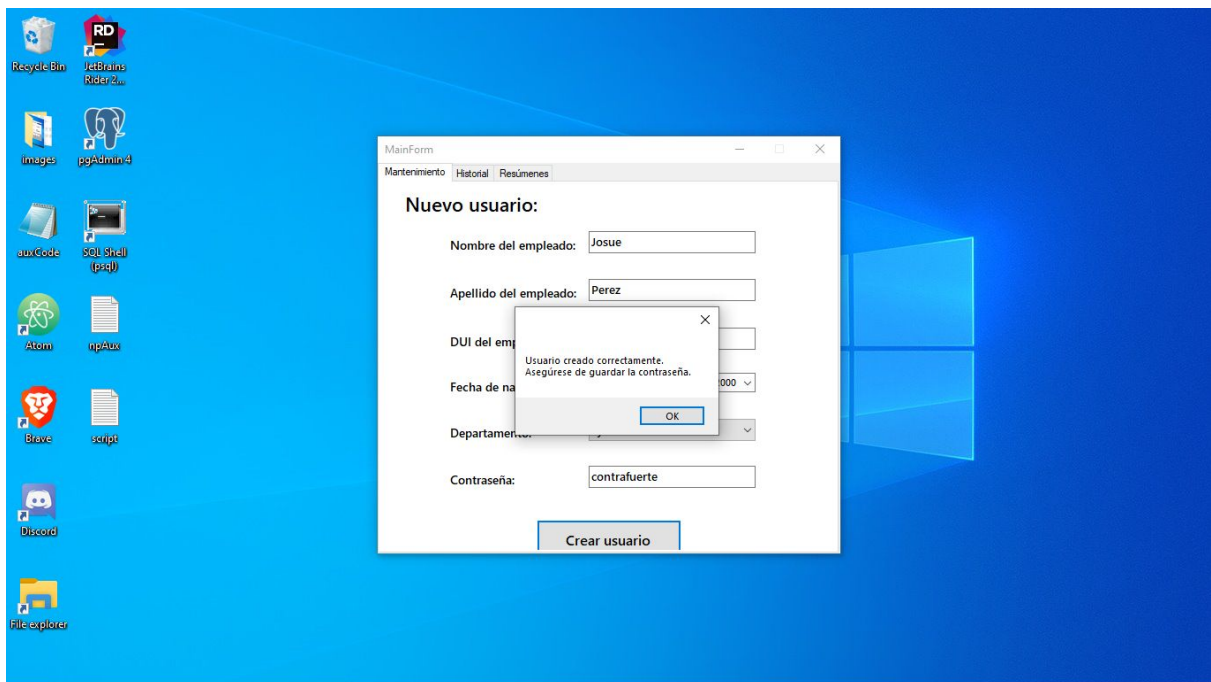
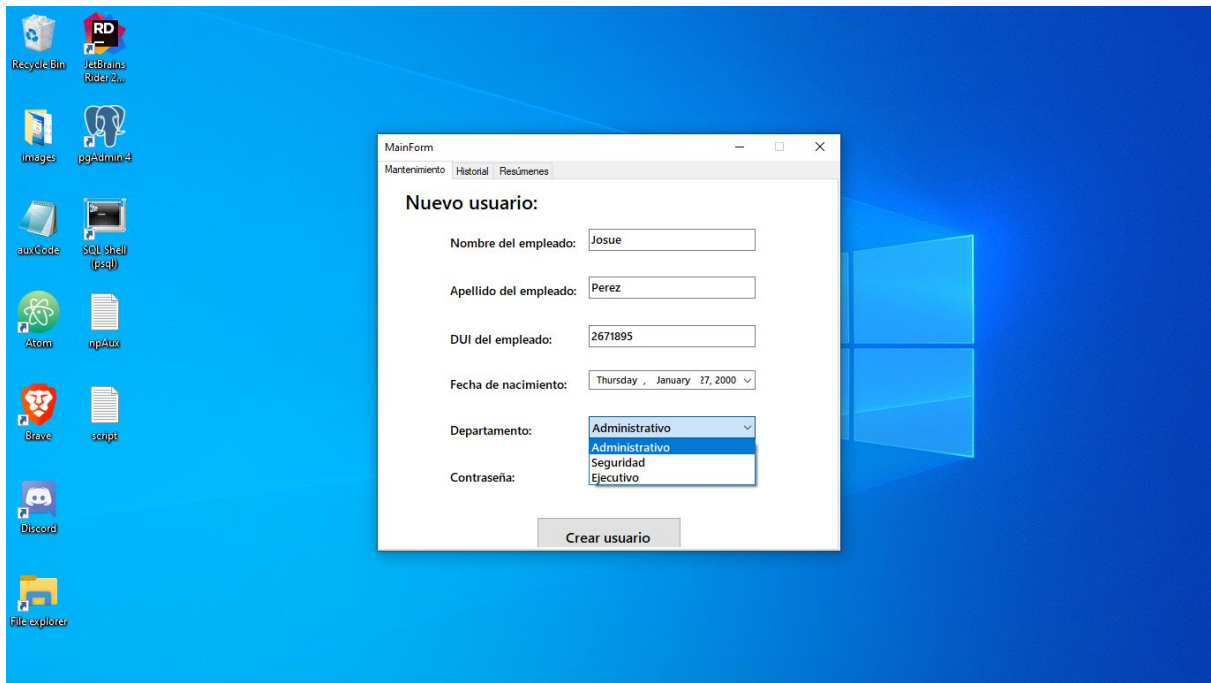
# Examen final Programación orientada a objetos

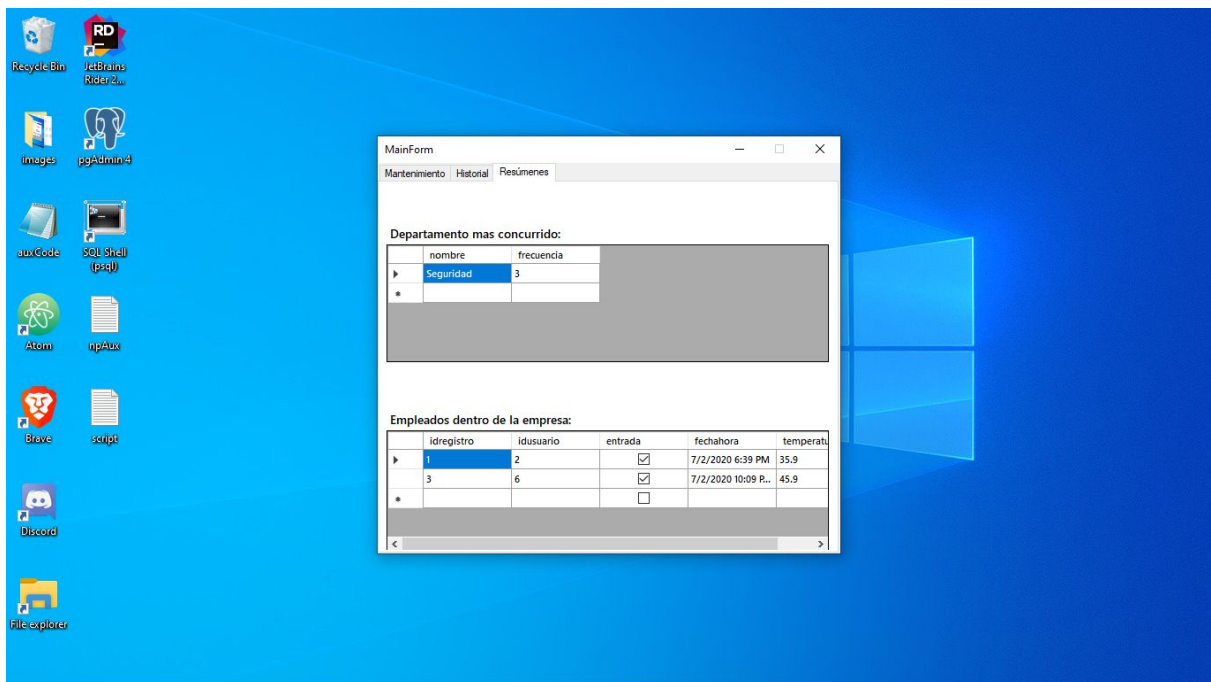
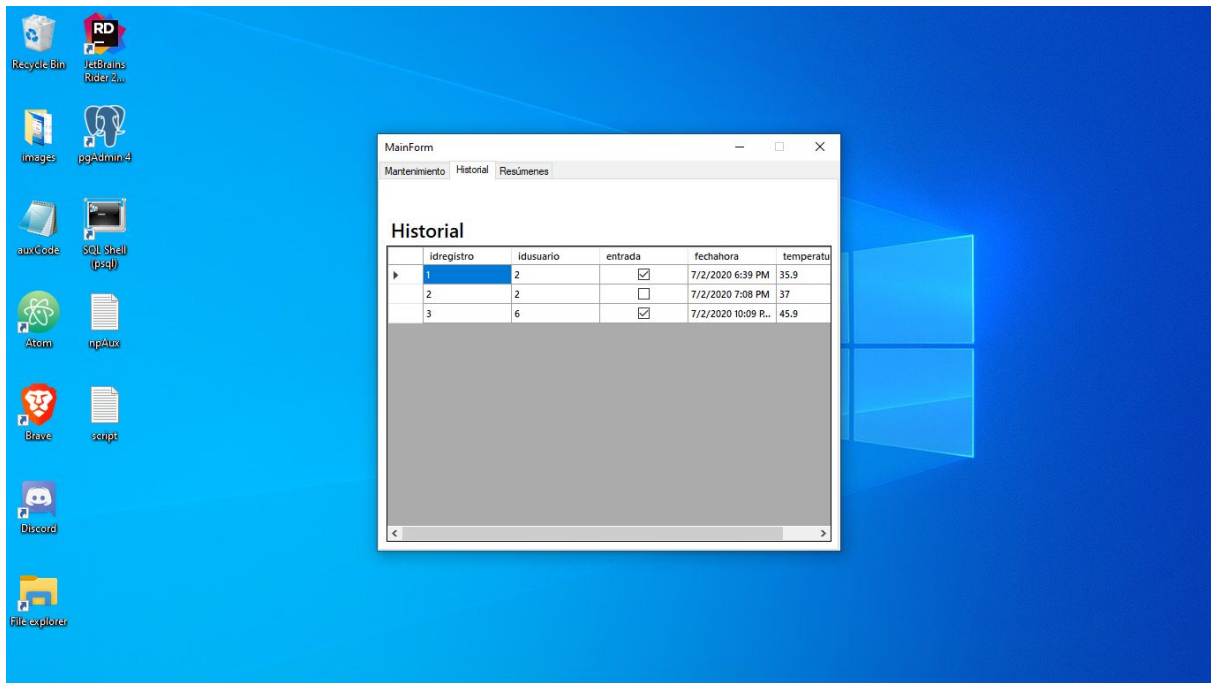
## Patrón de diseño “Estrategia”

El patrón de diseño estrategia es uno de los más fáciles de implementar, muy comprensible y útil, se utiliza especialmente para casos en los que se lleva a cabo una tarea pero esta requiere ser realizada con diferentes técnicas y herramientas. Para el caso de las aplicaciones con interfaz gráfica con conexión a base de datos, en las cuales existen múltiples tipos de usuarios con diferentes funciones, considero que el patrón estrategia proporciona un diseño óptimo. El patrón se implementó en el momento de desplegar los controles para los distintos tipos de usuarios.

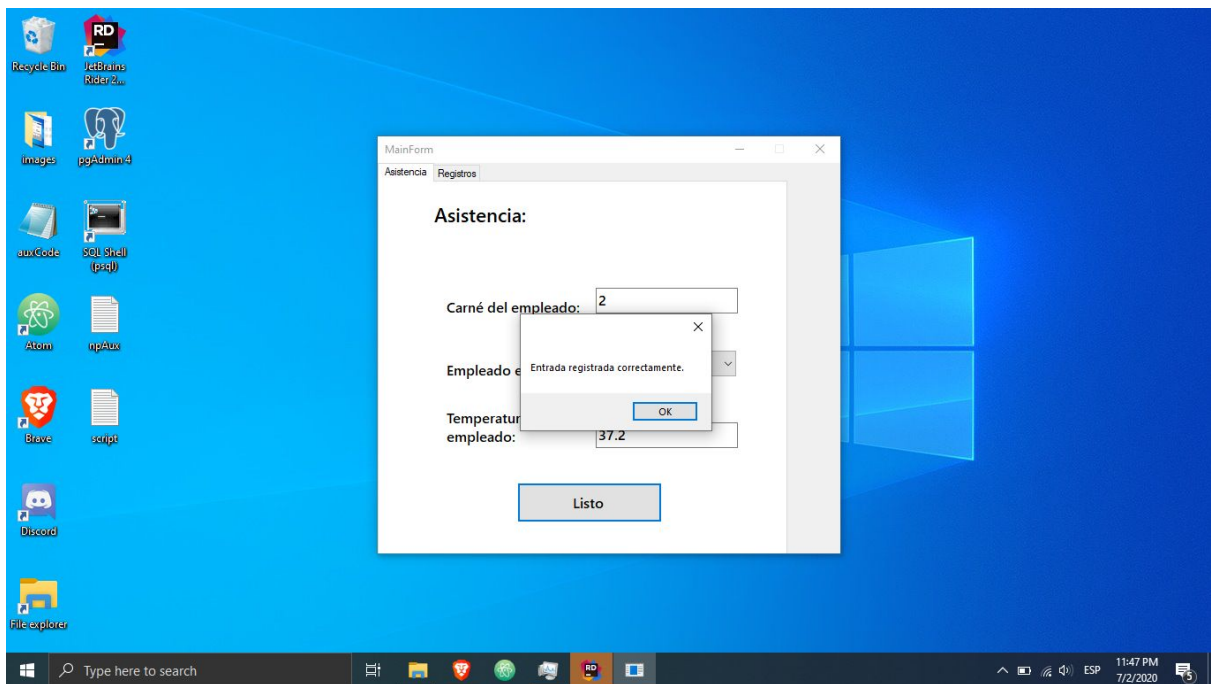
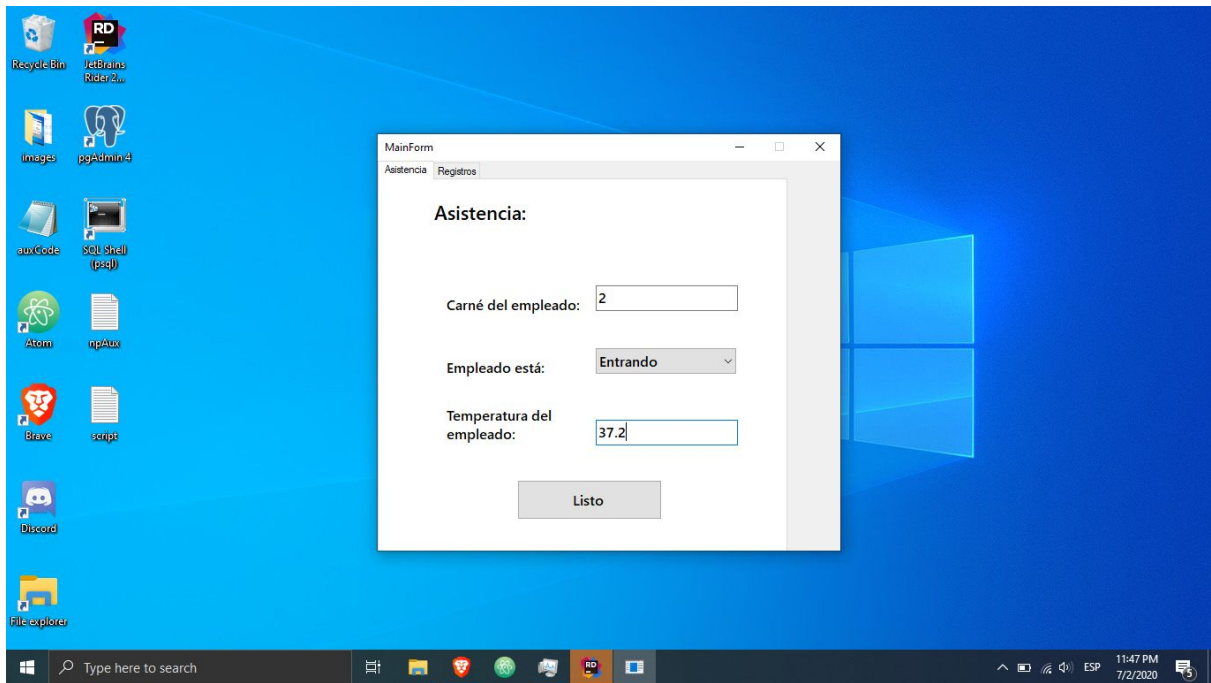


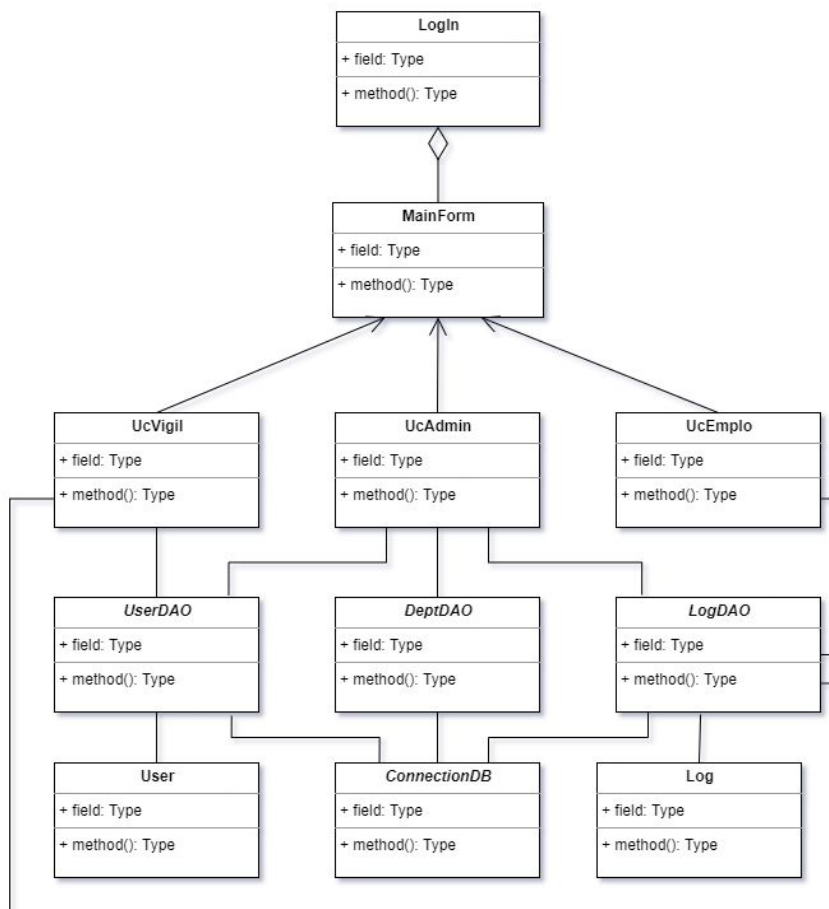
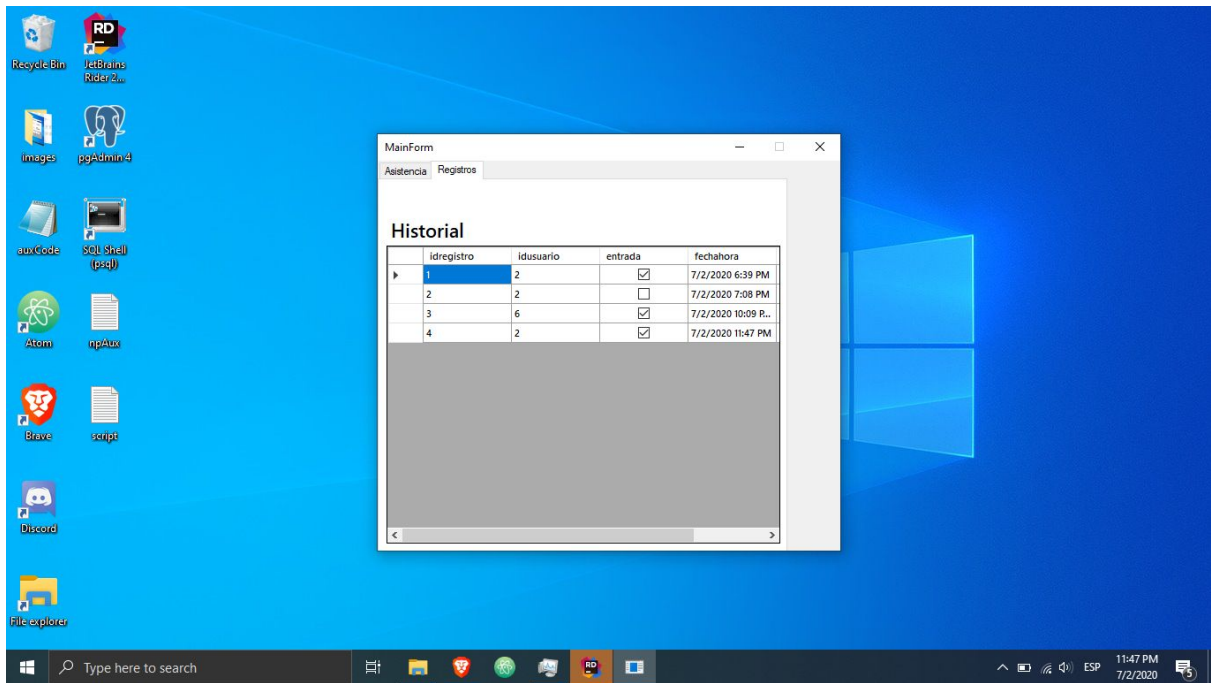












# Script de la base de datos

```
CREATE DATABASE poof;
```

```
CREATE TABLE departamento(  
idDepartamento SERIAL NOT NULL PRIMARY KEY,  
nombre VARCHAR(30) NOT NULL,  
ubicacion VARCHAR(30) NOT NULL);
```

```
CREATE TABLE usuario(  
idUsuario SERIAL NOT NULL PRIMARY KEY,  
idDepartamento INT NOT NULL,  
contrasena VARCHAR(15) NOT NULL,  
nombre VARCHAR(30) NOT NULL,  
apellido VARCHAR(30),  
dui VARCHAR(9) NOT NULL,  
fechaNacimiento VARCHAR(10) NOT NULL,  
CONSTRAINT fk_departamento_usuario FOREIGN KEY(idDepartamento)  
REFERENCES departamento(idDepartamento));
```

```
CREATE TABLE registro(  
idRegistro SERIAL NOT NULL PRIMARY KEY,  
idUsuario INT NOT NULL,  
entrada BOOLEAN NOT NULL,  
fechaHora TIMESTAMP NOT NULL,  
temperatura DECIMAL NOT NULL,  
CONSTRAINT fk_usuario_registro FOREIGN KEY(idUsuario) REFERENCES  
usuario(idUsuario));
```