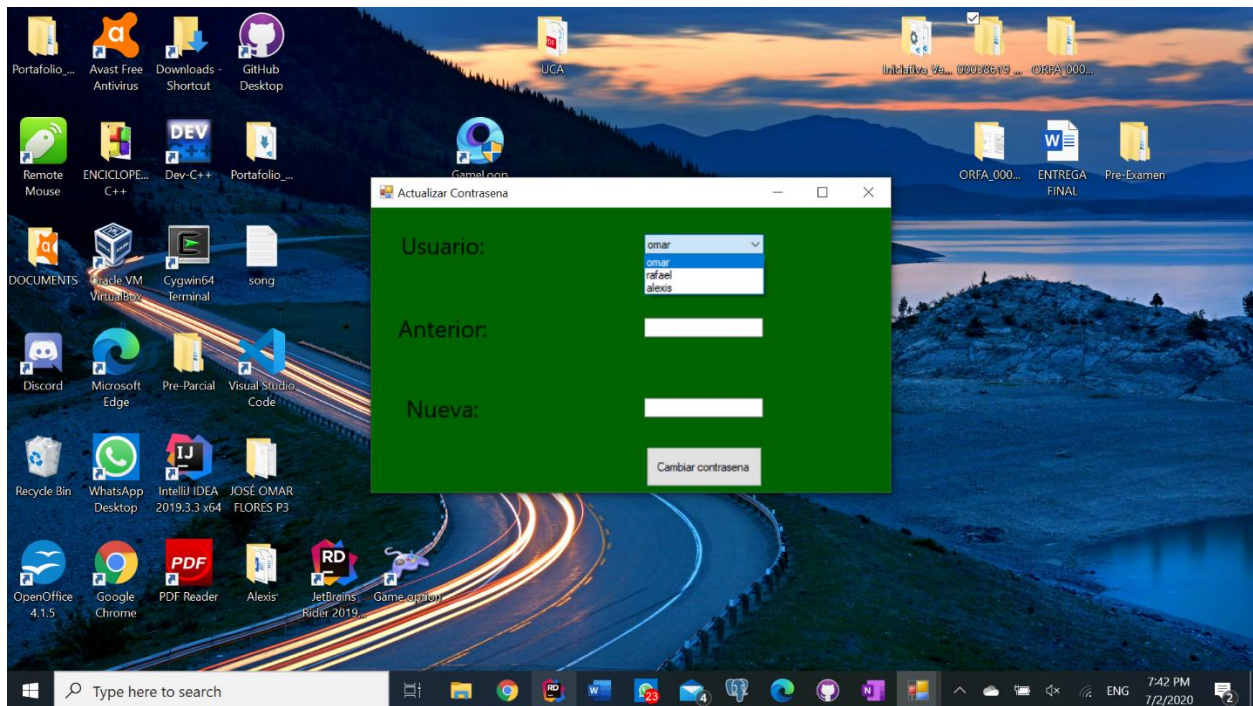
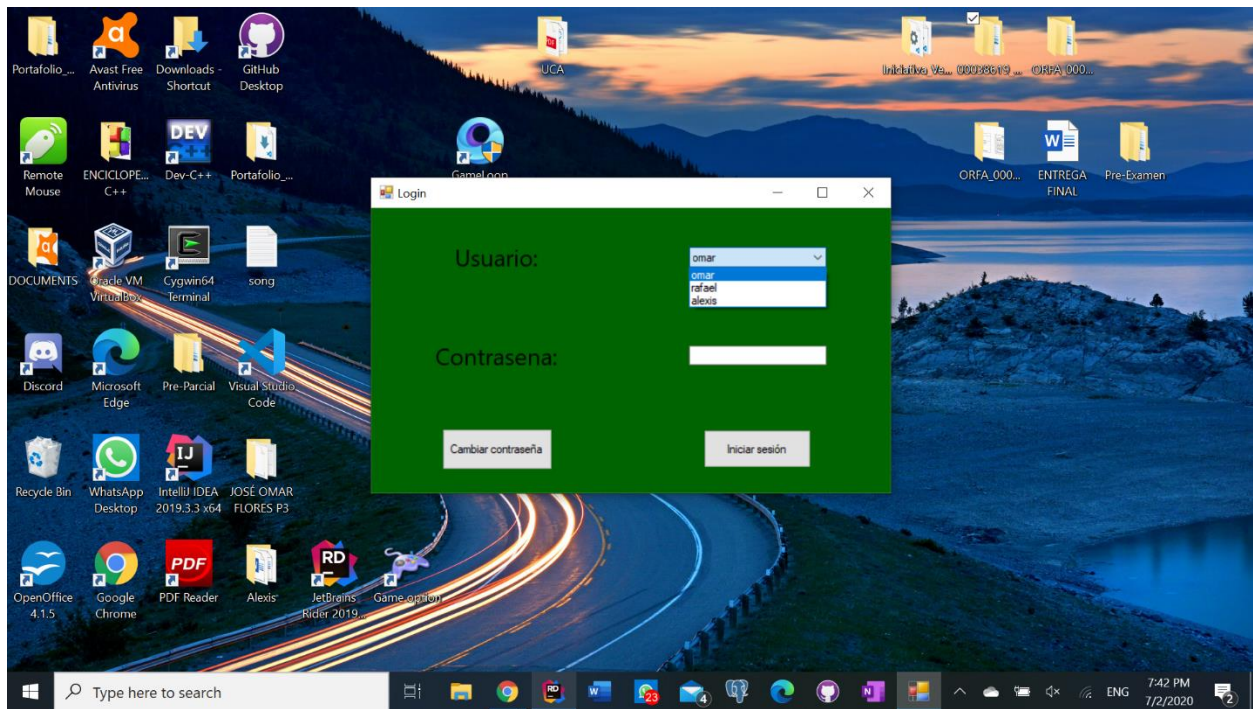
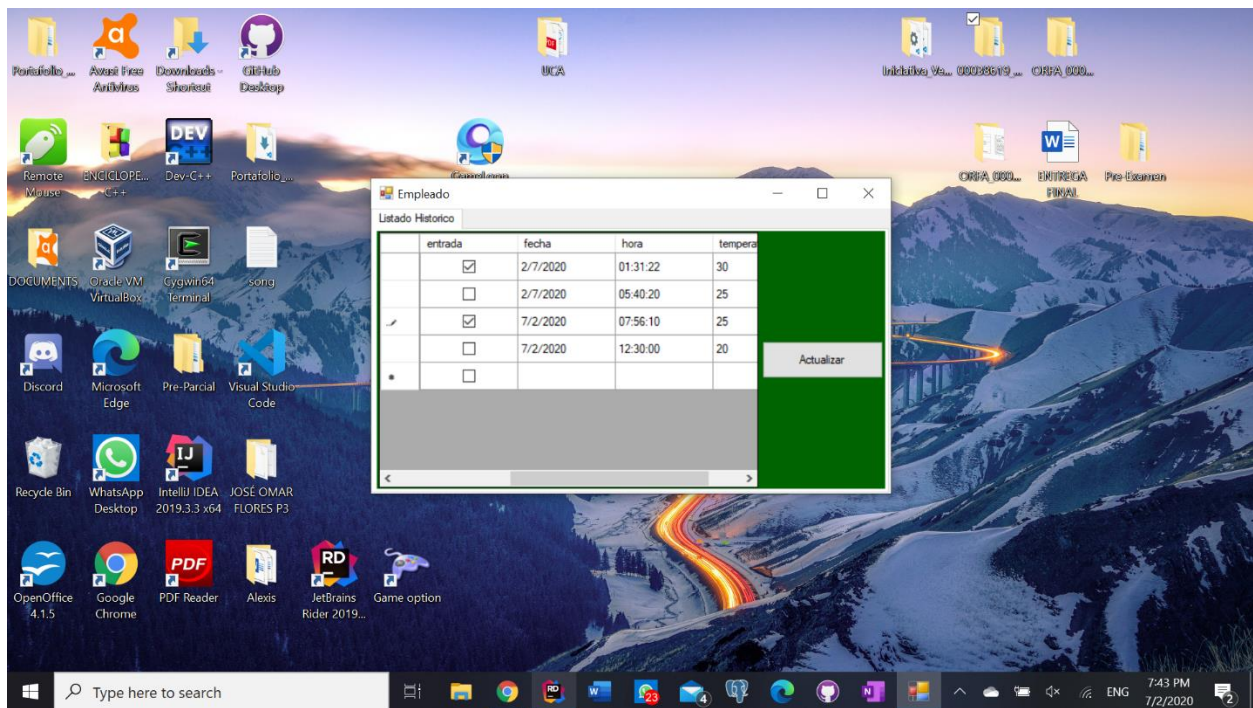
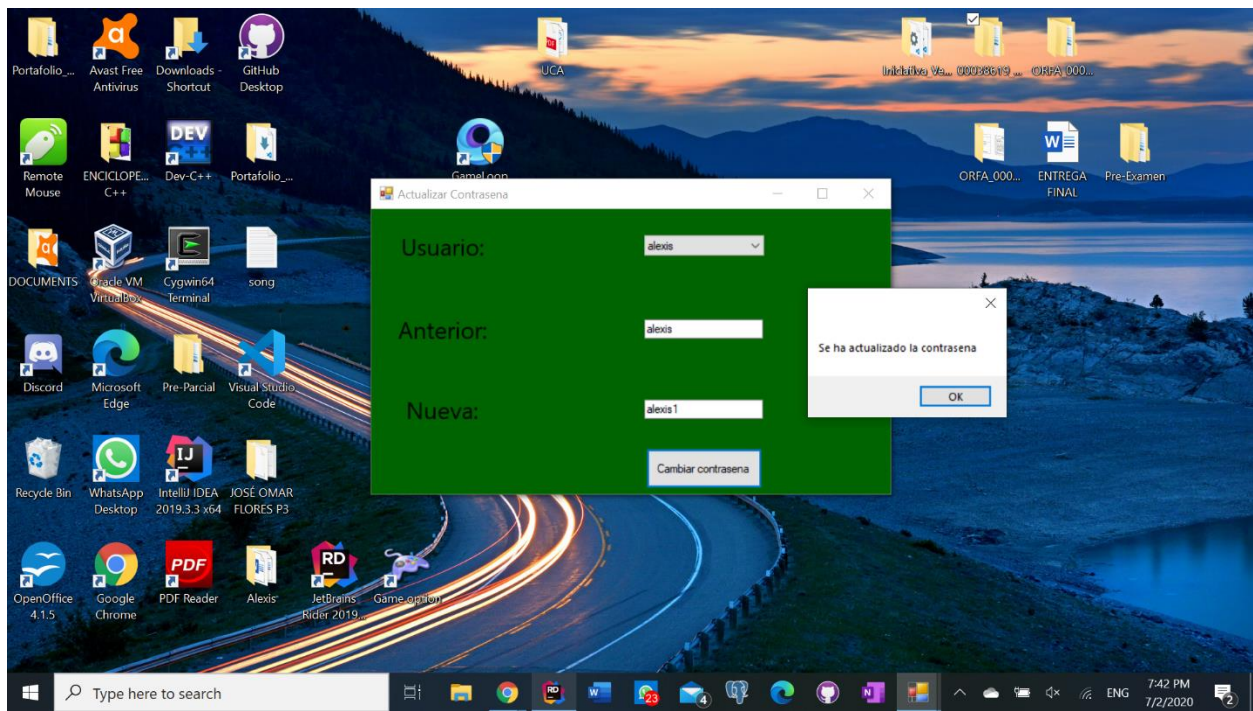
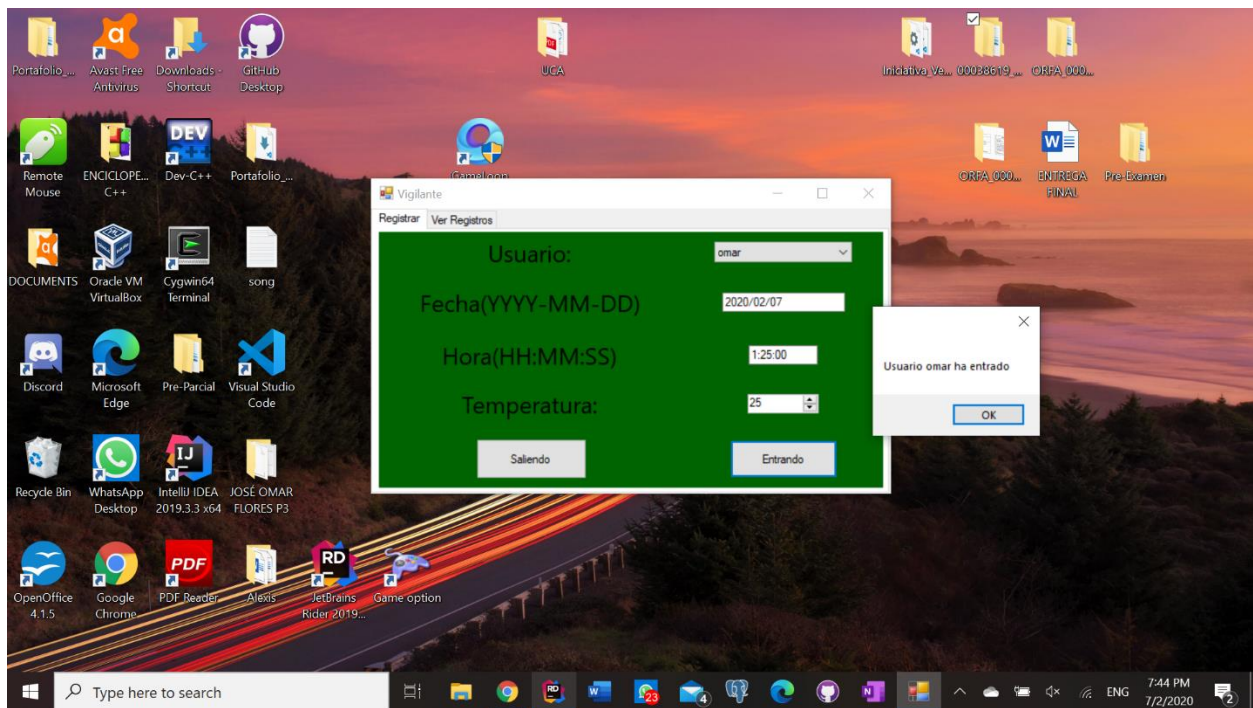
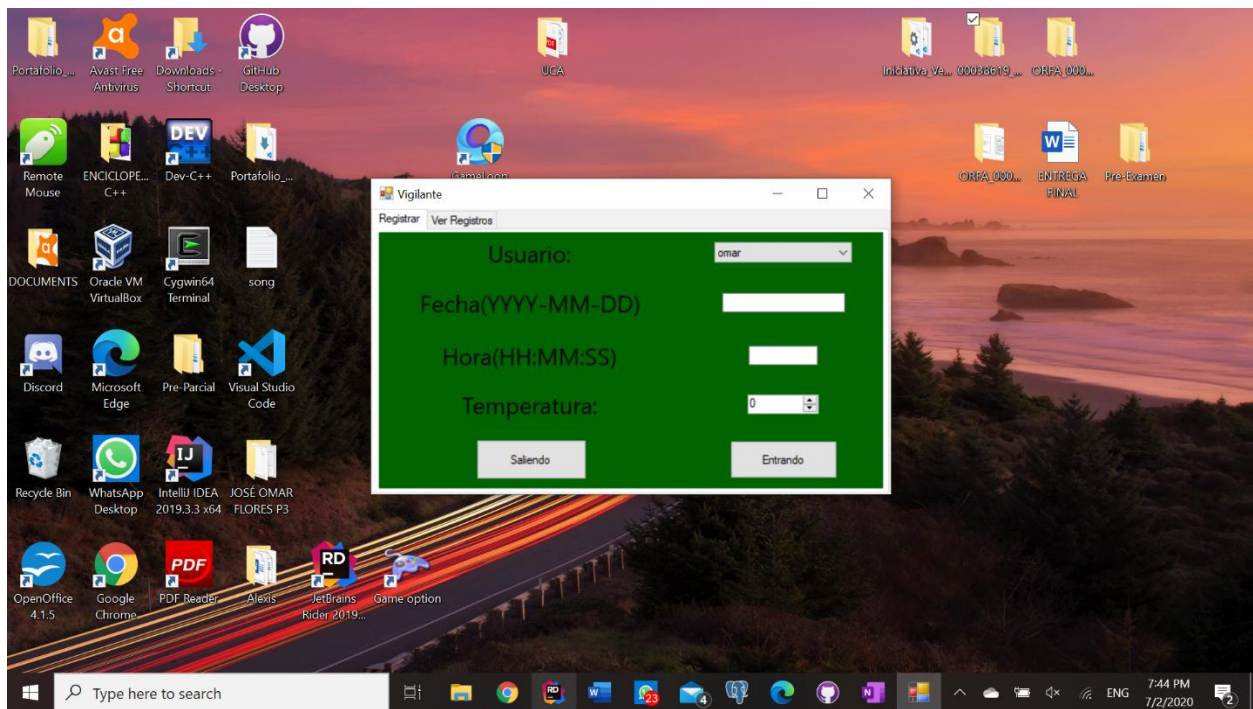
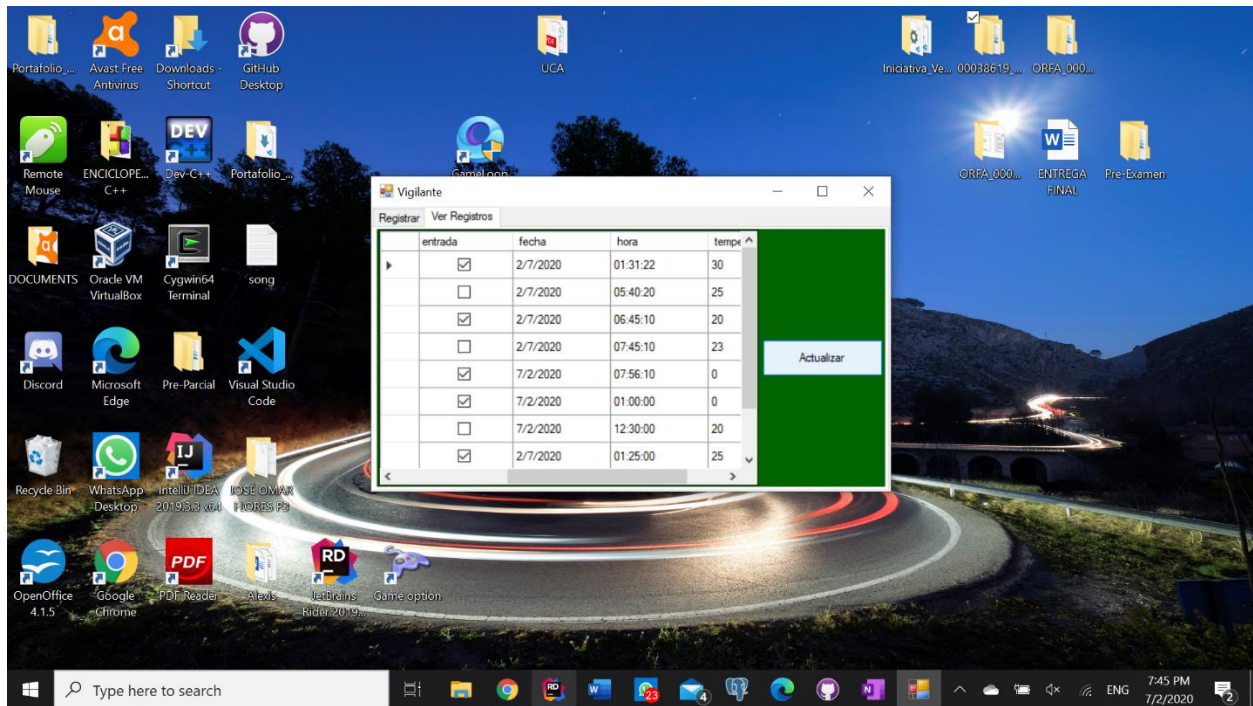
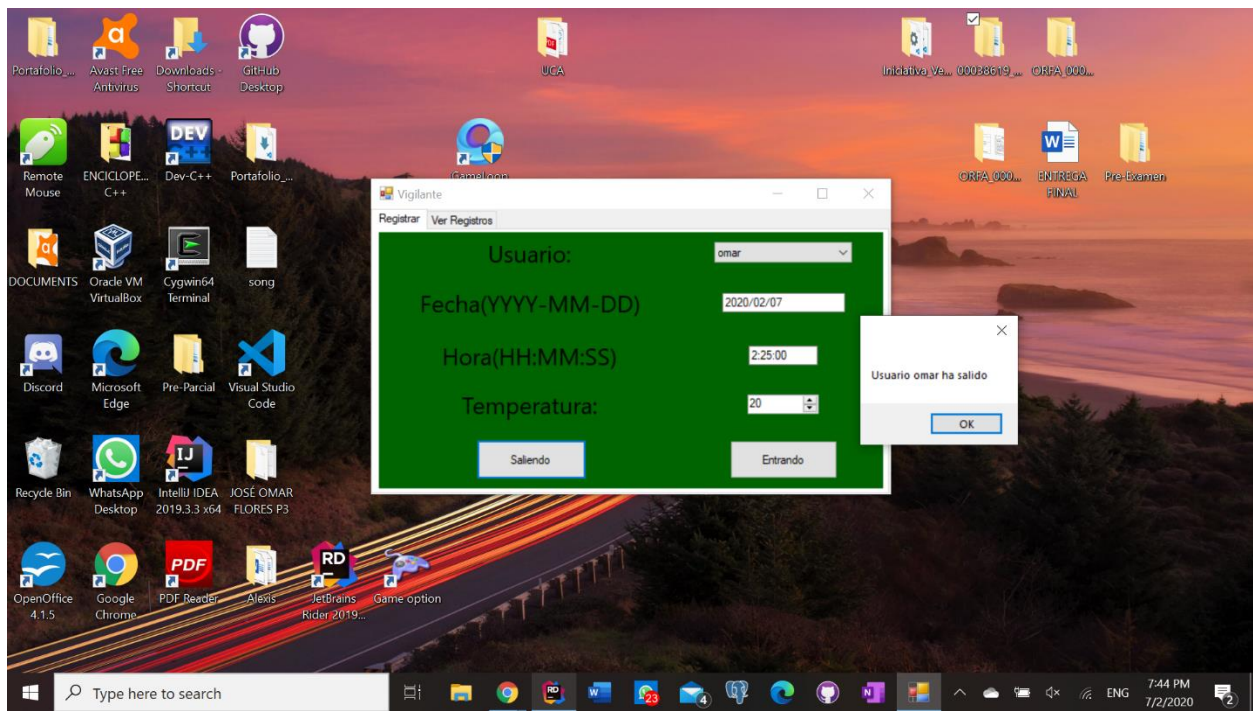


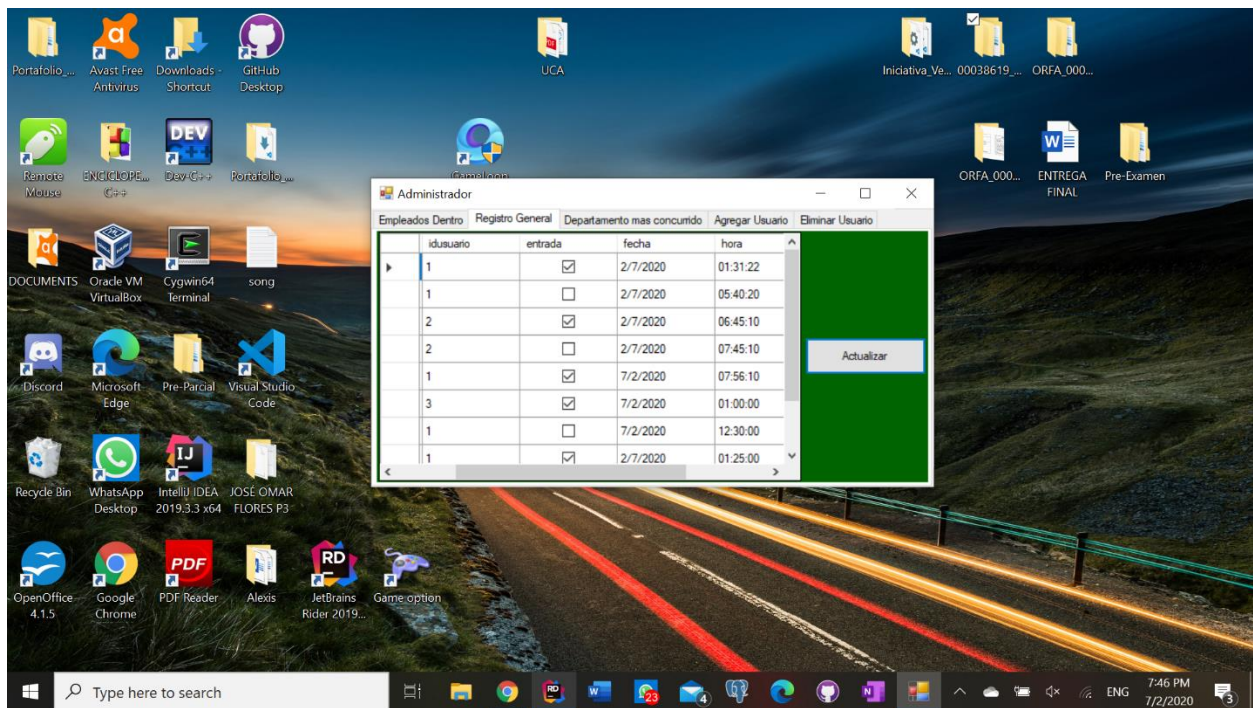
1. Capturas de pantalla:

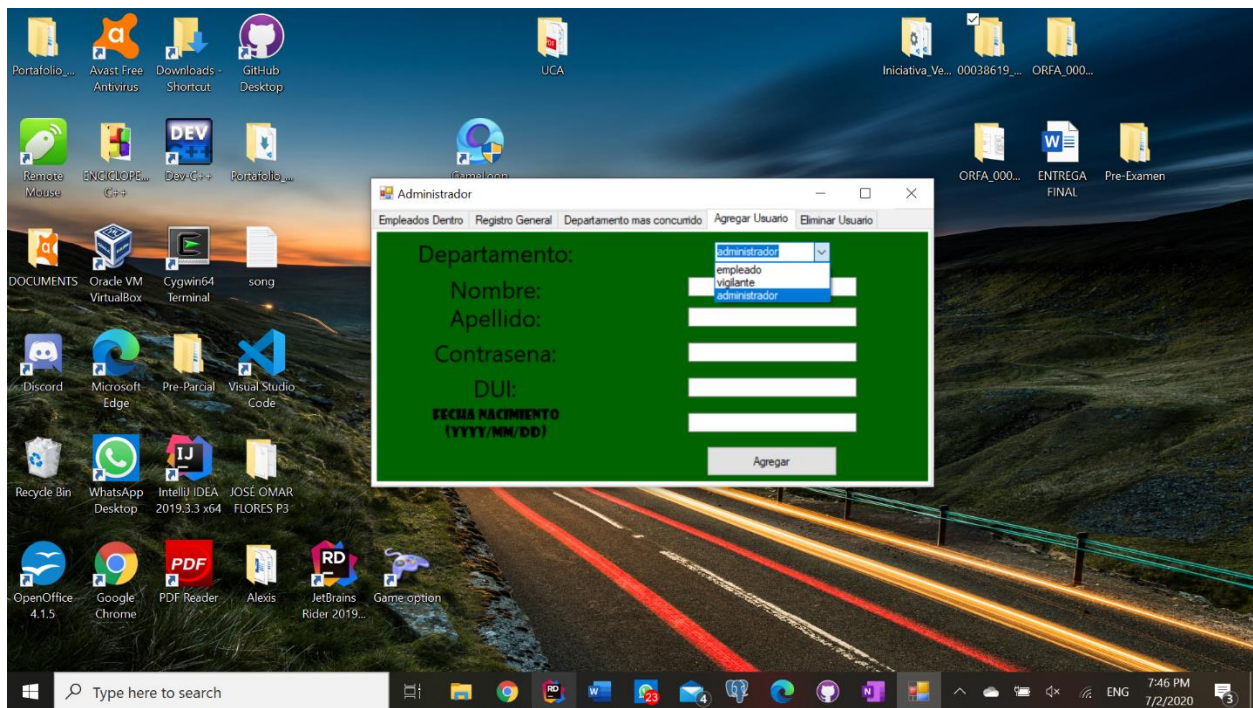
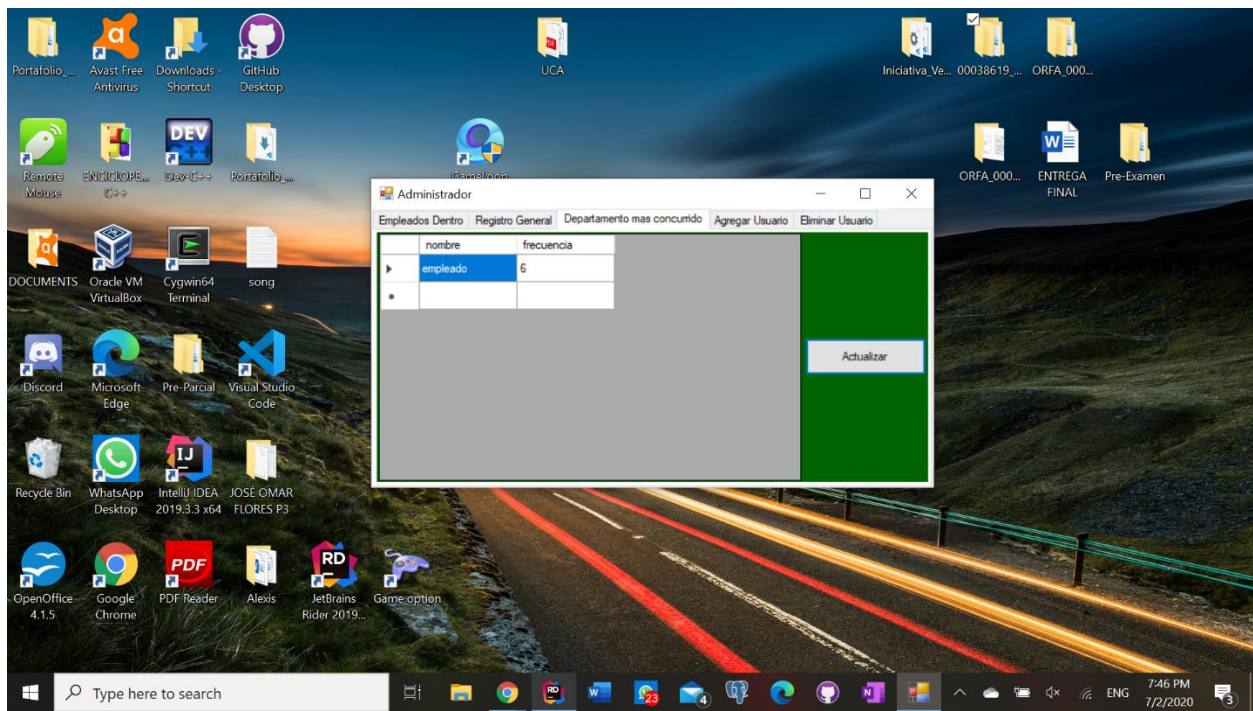


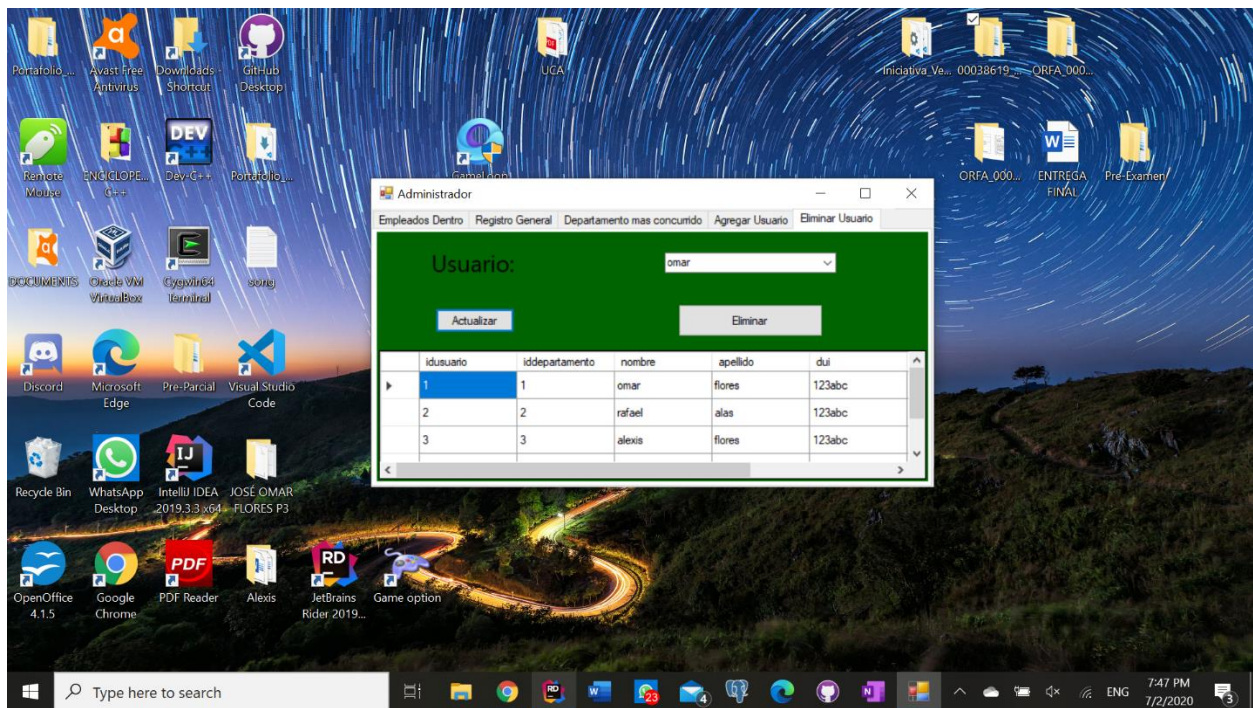
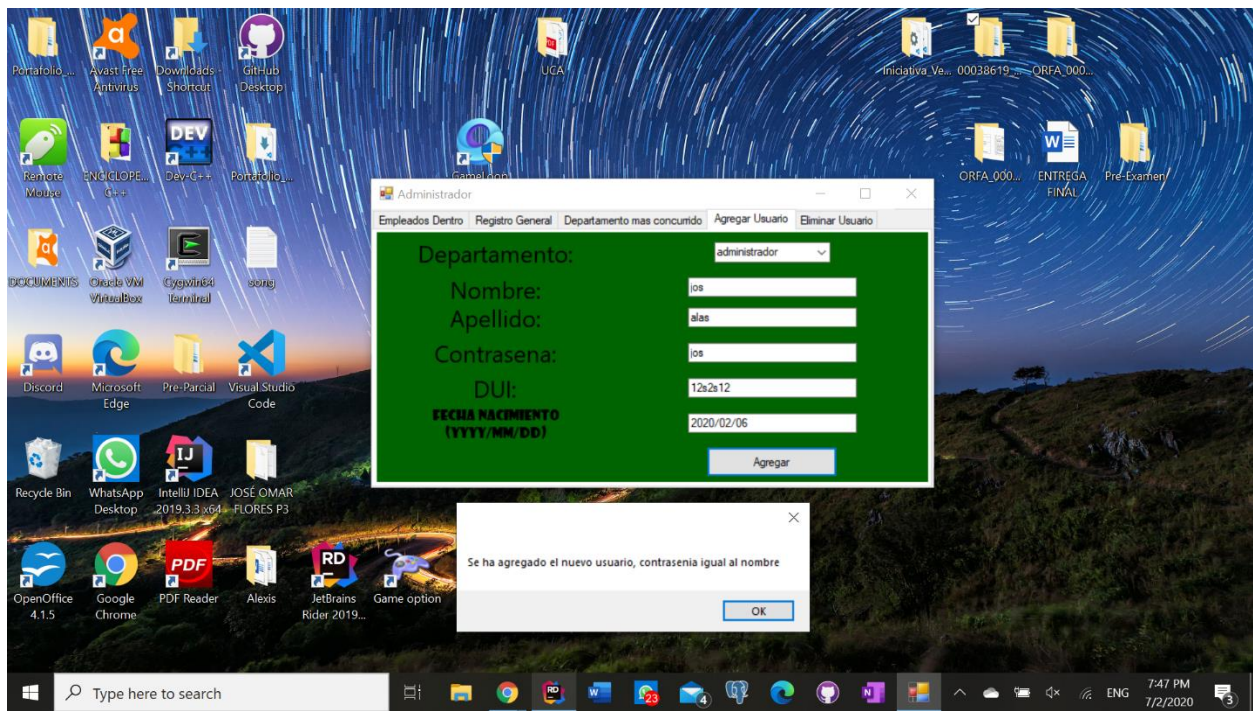


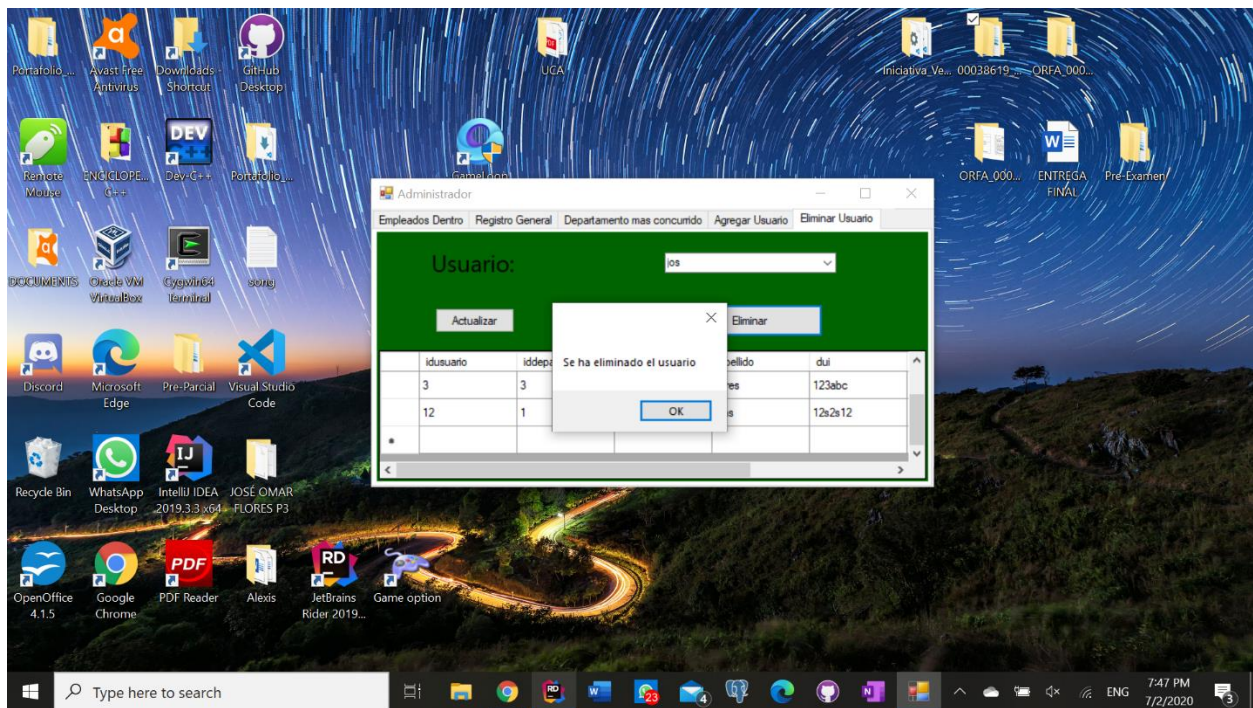
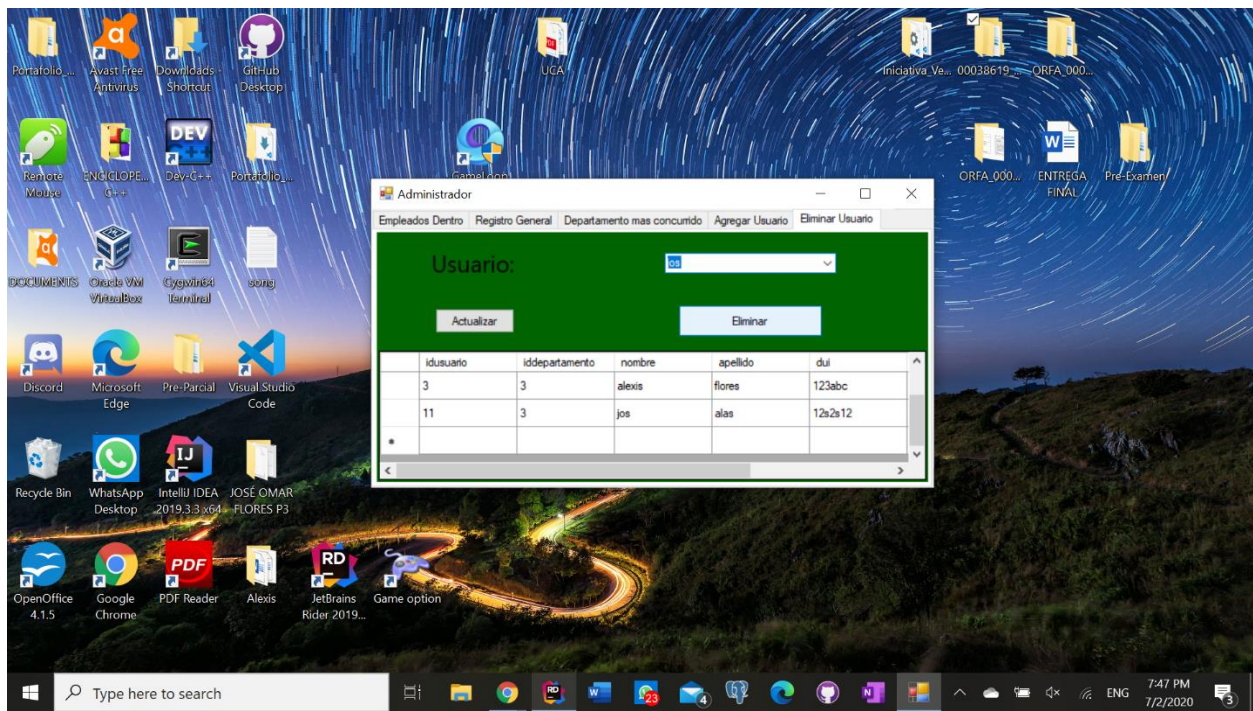


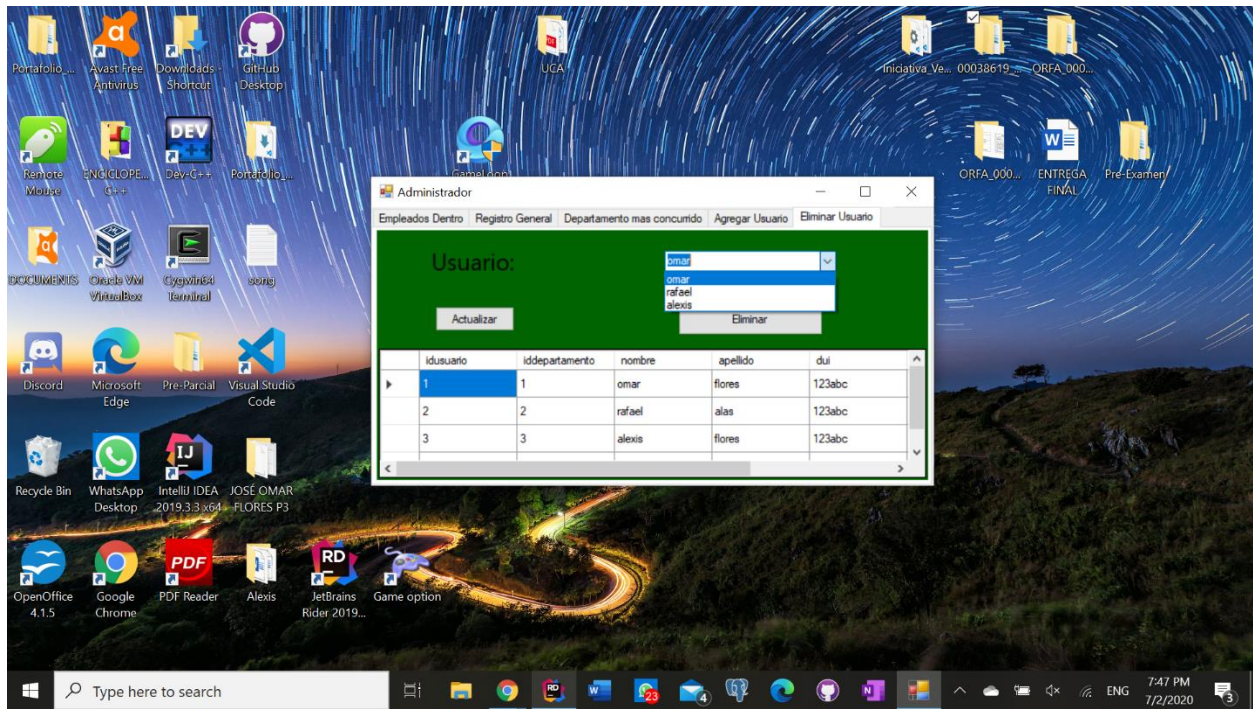






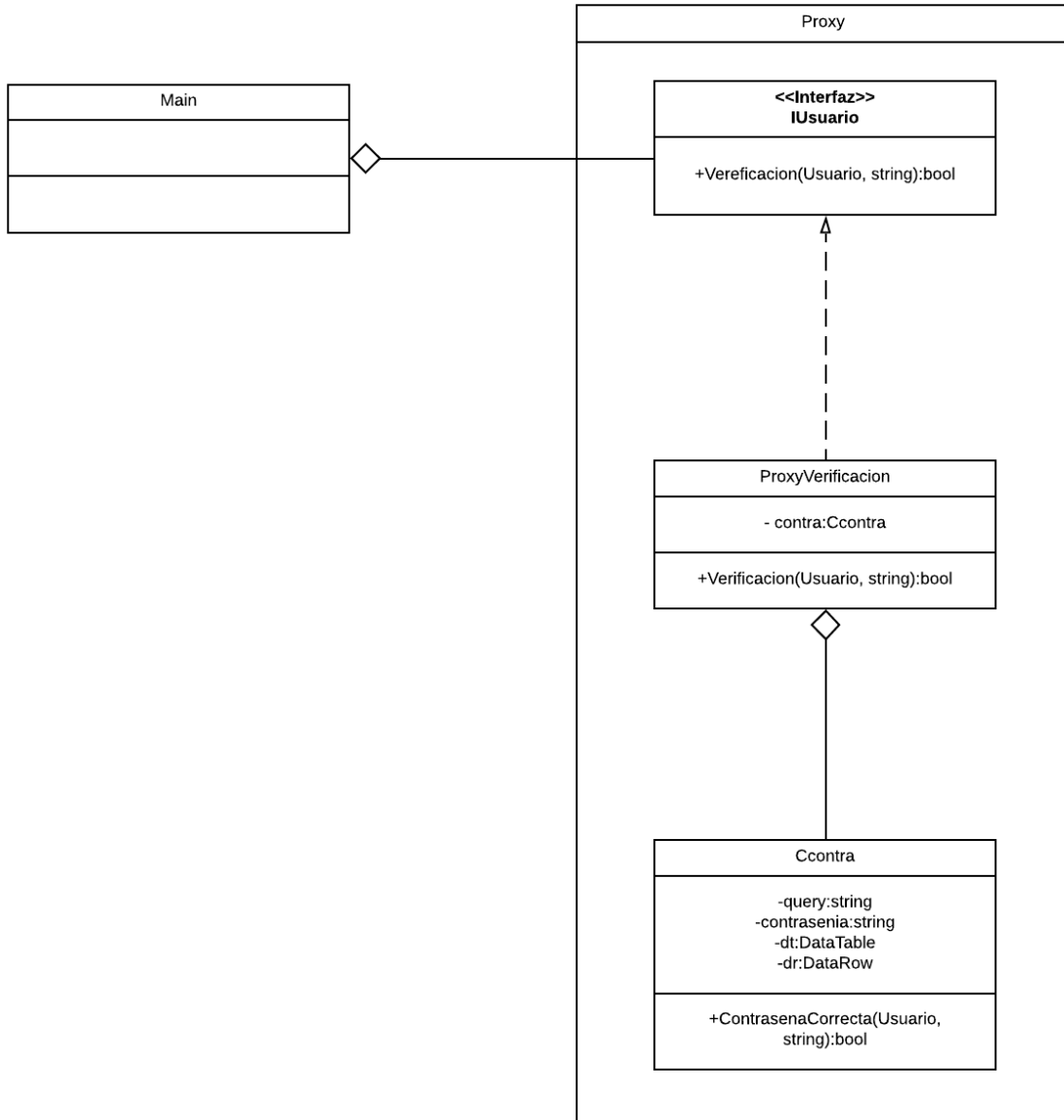




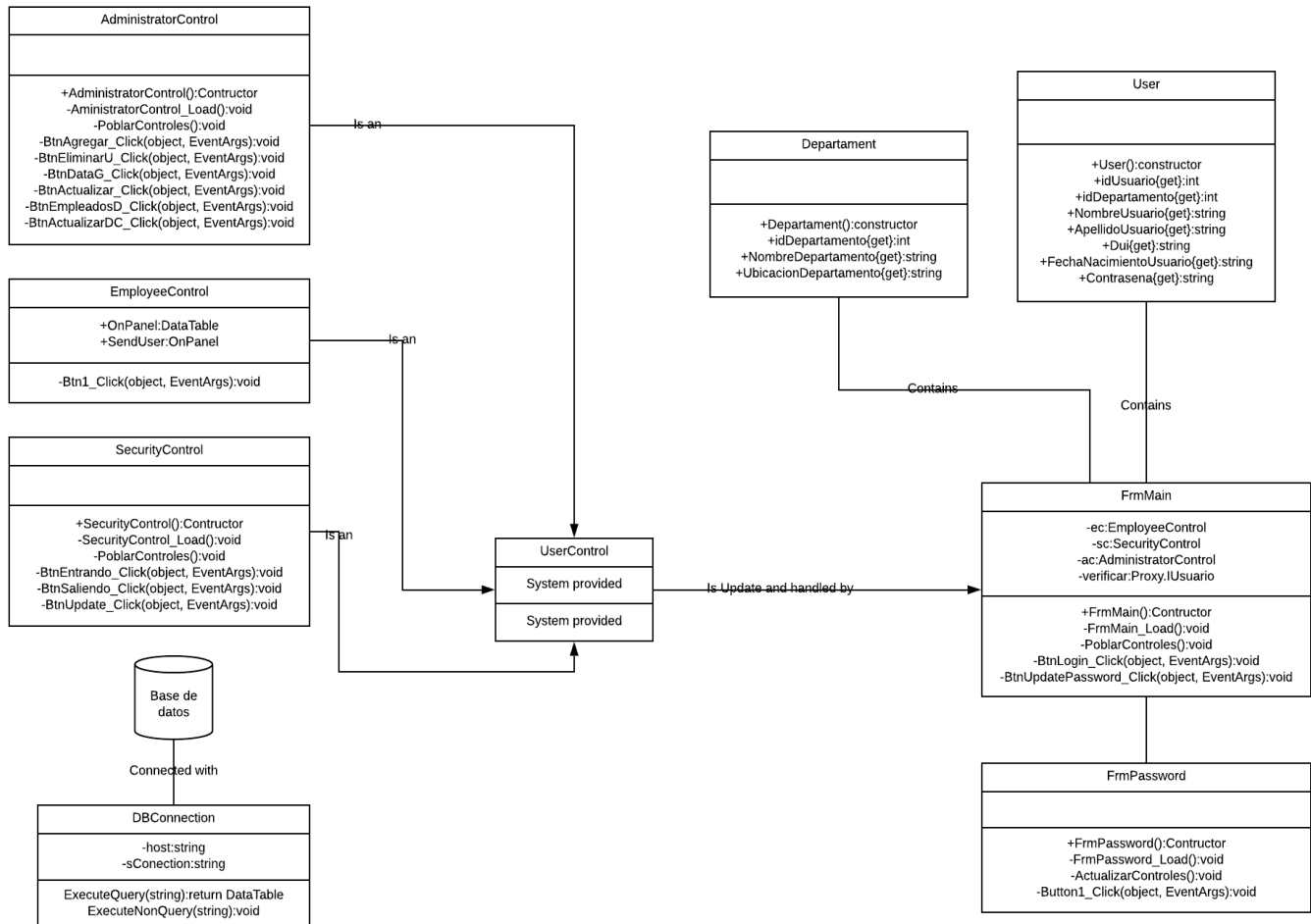


2. Diagramas UML:

-UML PROXY



-UML CLASES:



3. Patrones de diseño:

El patrón de diseño implementado fue el Patrón Proxy tipo Autenticación, se eligió este Patrón en específico debido a la facilidad que se tiene para implementarlo al proyecto y además del interés que generan las ventajas que tiene este Patrón de Software y a las constantes referencias que se le hacen. La mayor ventaja es que ofrece mayor seguridad a la clase interna (que se encarga de consultar la contraseña en la base de datos) a la hora del acceso del cliente a la clase, que no lo hace directamente, sino por medio del proxy, haciendo que el cliente haga solo determinadas acciones si cumple algunas condiciones, como lo es el caso que se usó (Proxy tipo Autenticación), si la contraseña es correcta validándola por medio de la clase privada entonces el cliente puede acceder. También mencionar que mantiene una referencia que permite al proxy acceder al objeto real. proporciona una interfaz igual que la del sujeto real.

La implementación se hizo creando:

- Una clase "Proxy"
- Una interface "IUsuario"
- Una clase "ProxyVerificacion" que implementa a la interface "IUsuario"
- Y una clase **privada** "Ccontra" que tiene el método "ContrasenaCorrecta"

La clase Ccontra, que es la clase protegida, tiene el método ContrasenaCorrecta que recibe como parámetros a un objeto tipo Usuario y un string que es la contraseña digitada en el textBox y que lo que hace es hacer una consulta en la Base de datos para verificar si la contraseña digitada y la contraseña de la base de datos de ese Usuario en específico son iguales, por lo cual devolvería un "true"

En el "FrmLogin" es donde se crea un proxy llamado "verificar" de tipo "IUsuario" que se encuentra dentro de "Proxy" y va a ser del tipo "ProxyVerificacion"

4. Script de la base de datos:

```
CREATE TABLE DEPARTAMENTO
```

```
(
```

```
    idDepartamento SERIAL PRIMARY KEY,
```

```
    nombre VARCHAR(20) NOT NULL,
```

```
    ubicacion VARCHAR(20) NOT NULL
```

```
);
```

```
CREATE TABLE USUARIO(
```

```
    idUsuario SERIAL PRIMARY KEY,
```

```
    idDepartamento INT NOT NULL,
```

```
    nombre VARCHAR(50) NOT NULL,
```

```
    apellido VARCHAR(50) NOT NULL,
```

```
    contrasena VARCHAR(50) NOT NULL,
```

```
    dui VARCHAR(20) NOT NULL ,
```

```
    fechaNacimiento DATE NOT NULL
```

```
);
```

```
CREATE TABLE REGISTRO(
```

```
    idRegistro SERIAL PRIMARY KEY,
```

```
    idUsuario INT NOT NULL,
```

```
    entrada BOOL NOT null,
```

```
    fecha DATE NOT null,
```

```
    hora TIME NOT null,
```

```
    temperatura INT NOT null
```

```
);
```

```
ALTER TABLE USUARIO
```

```
ADD CONSTRAINT fk_usuario_departamento
```

```
FOREIGN KEY (idDepartamento) REFERENCES DEPARTAMENTO (idDepartamento);
```



```
ALTER TABLE REGISTRO
```

```
ADD CONSTRAINT fk_registro_usuario
```

```
FOREIGN KEY (idUserio) REFERENCES USUARIO (idUserio);
```