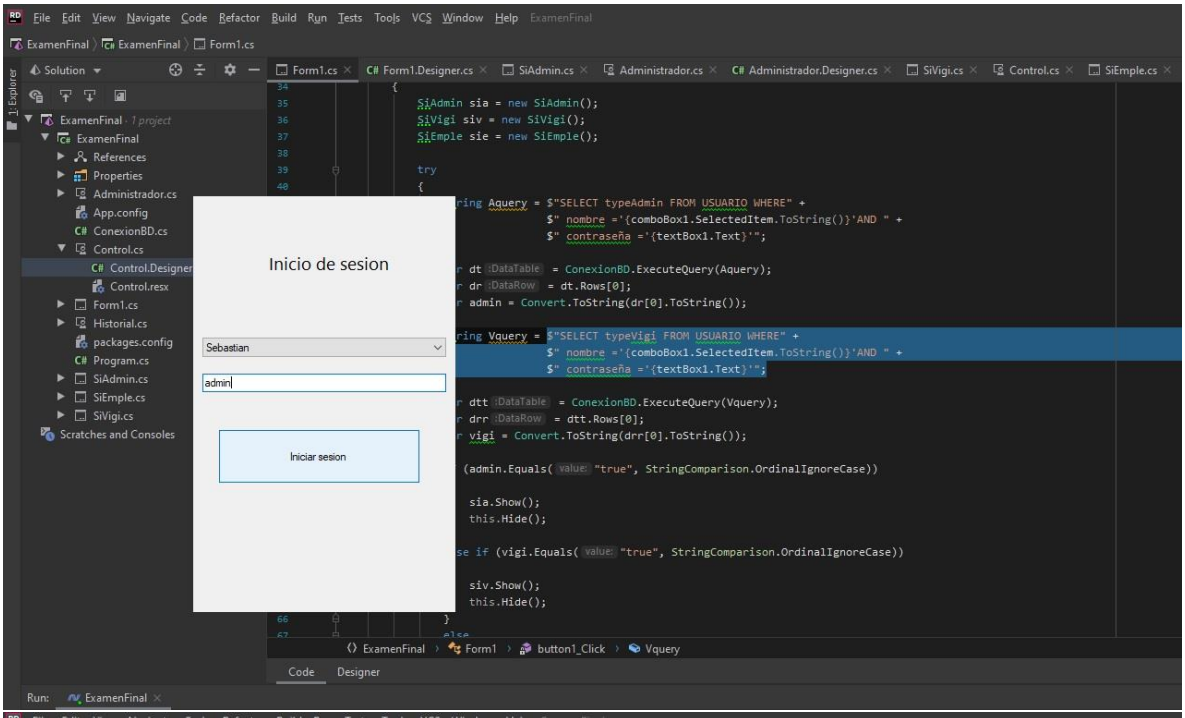
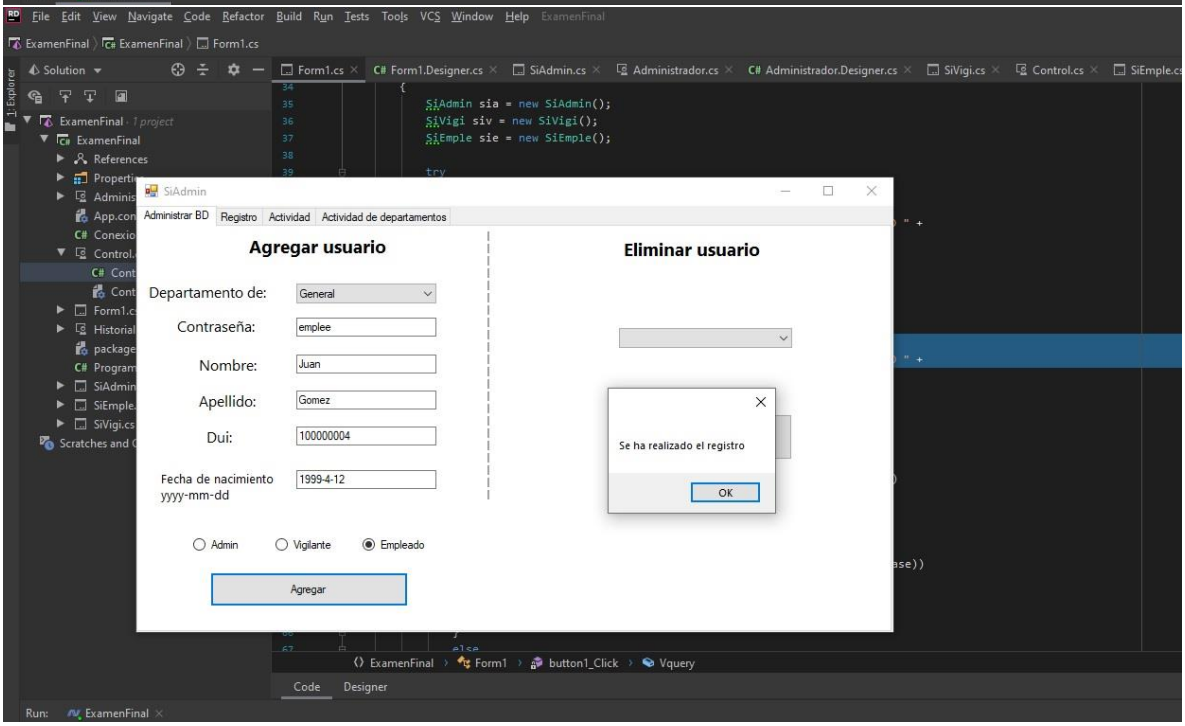
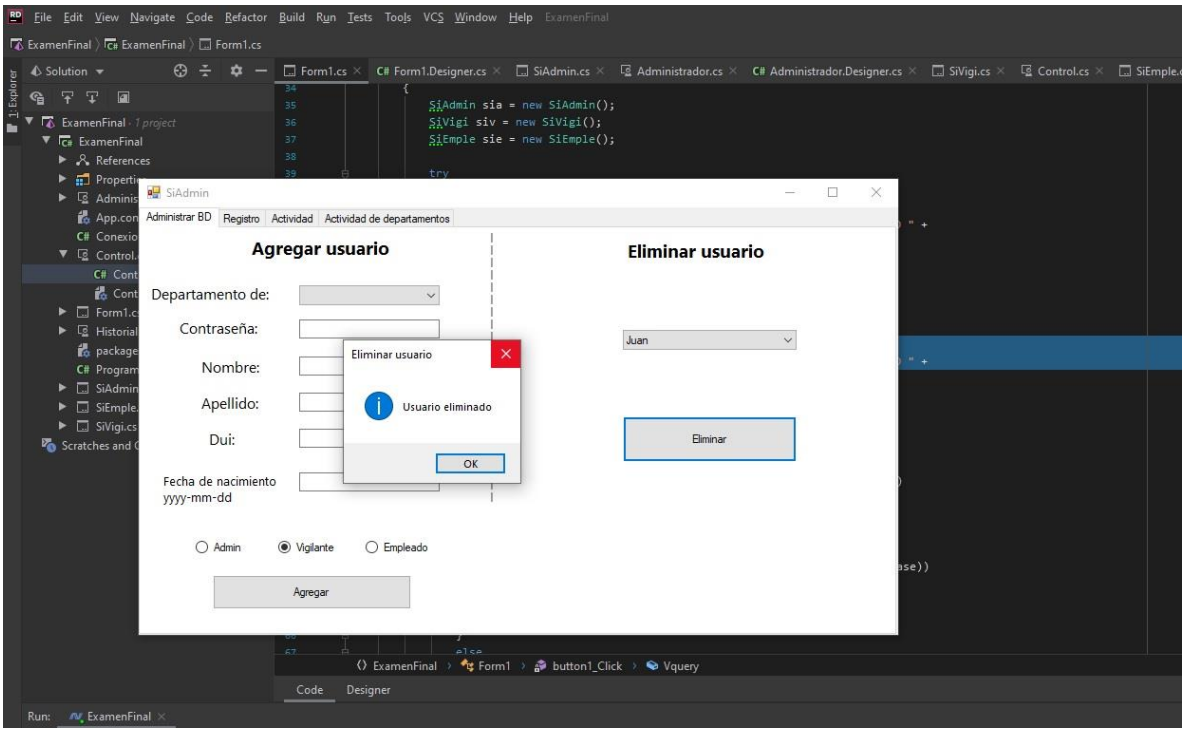
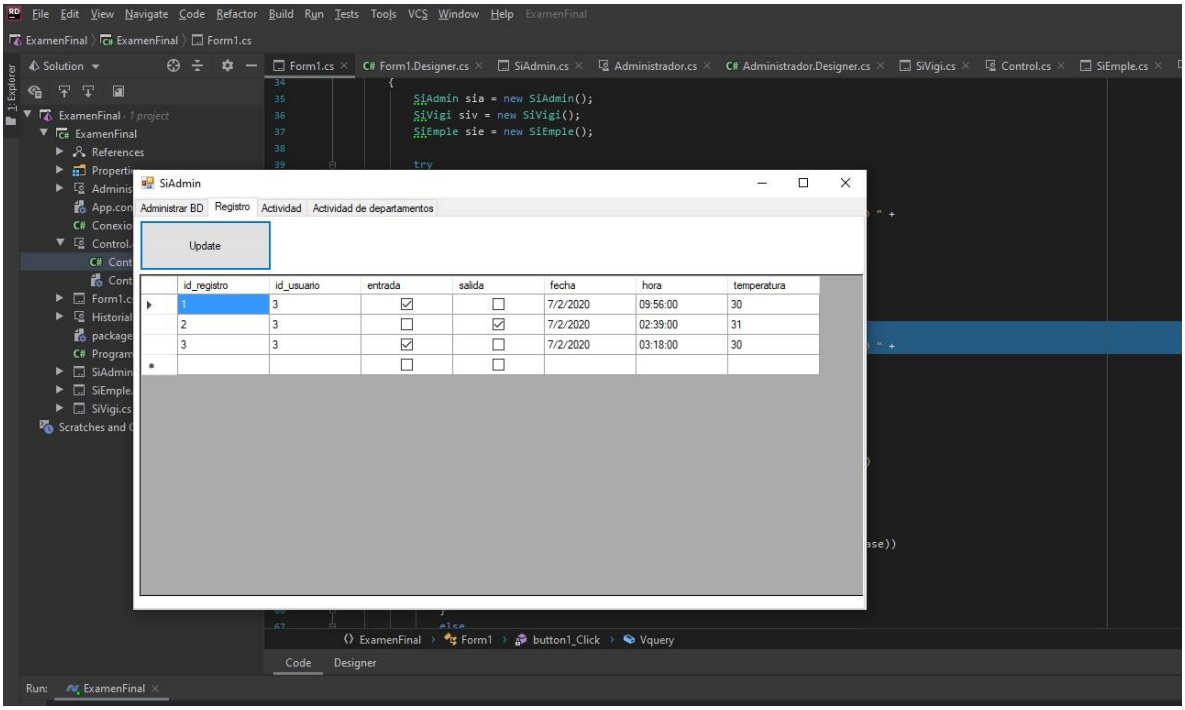
Manual

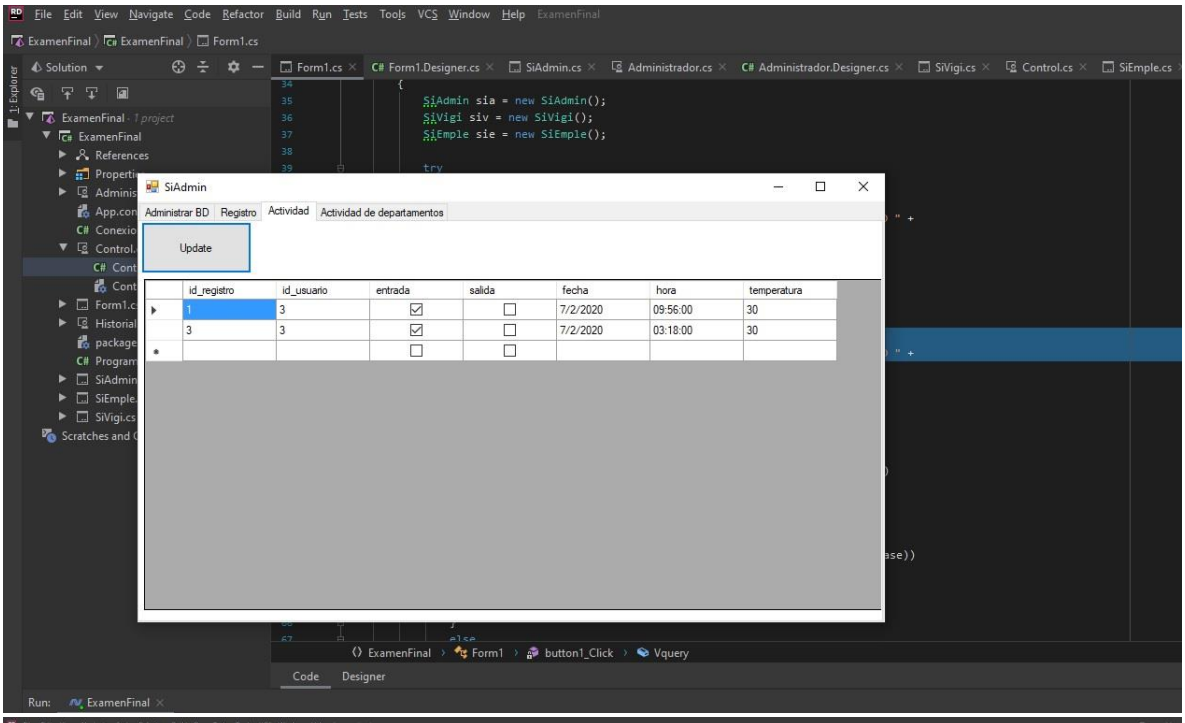
Admin:

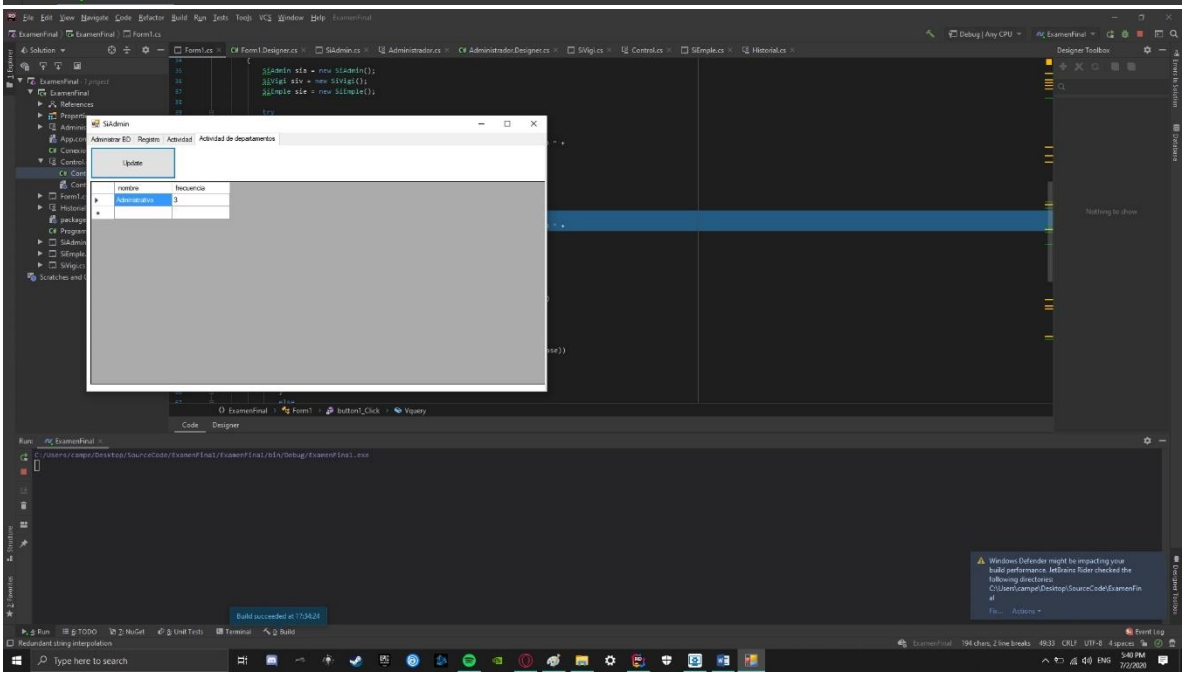




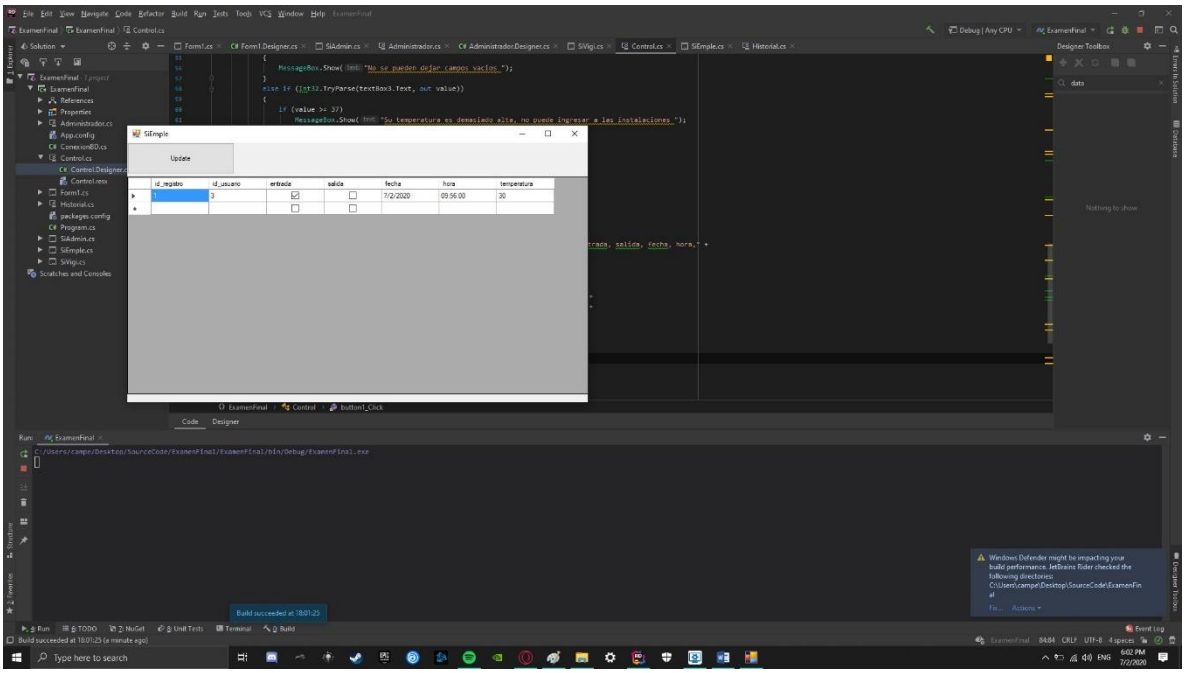

SiAdmin
Administrar BD
Registro
Actividad
Actividad de departamentos
Agregar usuario
Departamento de:
Contraseña:
Nombre:
Apellido:
Dui:
Fecha de nacimiento
yyyy-mm-dd
Admin
Vigilante
Empleado
Agregar
Eliminar usuario
Usuario eliminado
OK
Eliminar usuario
Juan
Eliminar


SiAdmin
Administrar BD
Registro
Actividad
Actividad de departamentos
Update
id_registro
id_usuario
entrada
salida
fecha
hora
temperatura
1 3 7/2/2020 09:56:00 30
2 3 7/2/2020 02:39:00 31
3 3 7/2/2020 03:18:00 30

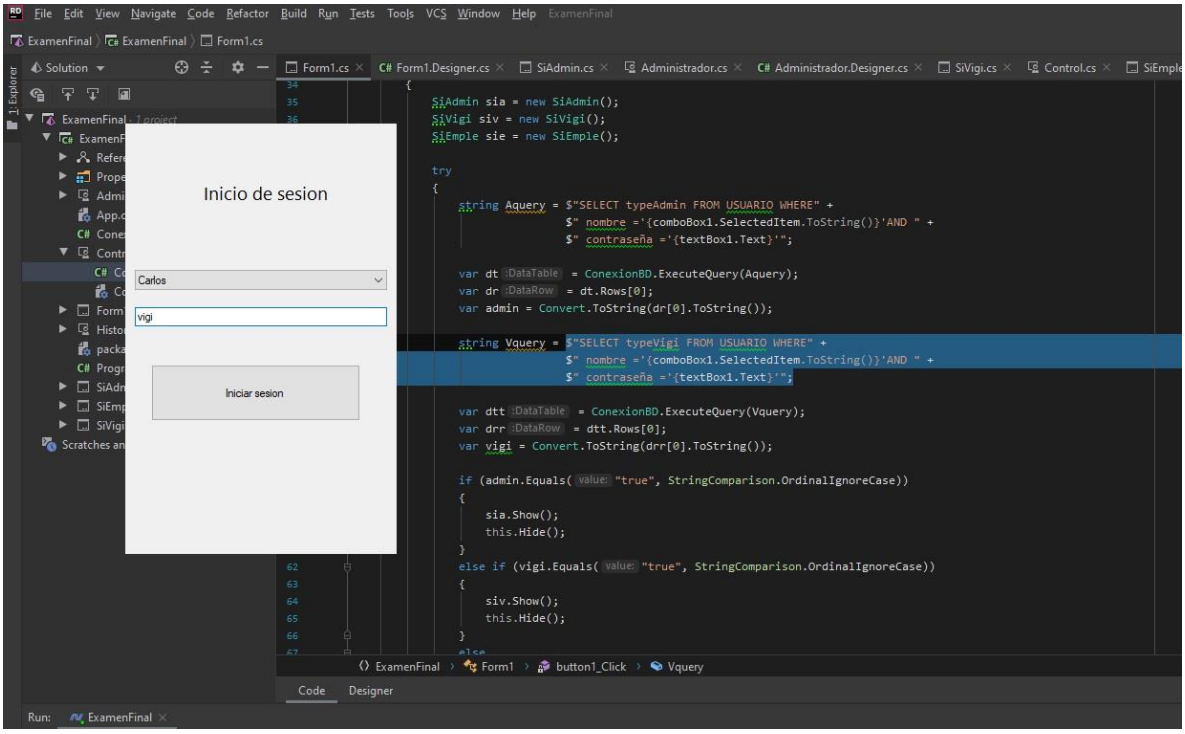

File Edit View Navigate Code Refactor Build Run Tests Tools VCS Window Help ExamenFinal
ExamenFinal ExamenFinal Form1.cs
Solution
Form1.cs
Form1.Designer.cs
SiAdmin.cs
Administrador.cs
Administrador.Designer.cs
SiVigi.cs
Control.cs
SiEmple.cs
ExamenFinal - 1 project
ExamenFinal
References
SiAdmin
Administrar BD
Registro
Actividad
Actividad de departamentos
Update
id_registro
id_usuario
entrada
salida
fecha
hora
temperatura
1
3
7/2/2020
09:56:00
30
3
3
7/2/2020
03:18:00
30
ExamenFinal Form1 button1_Click Vquery
Code
Designer
Run: ExamenFinal

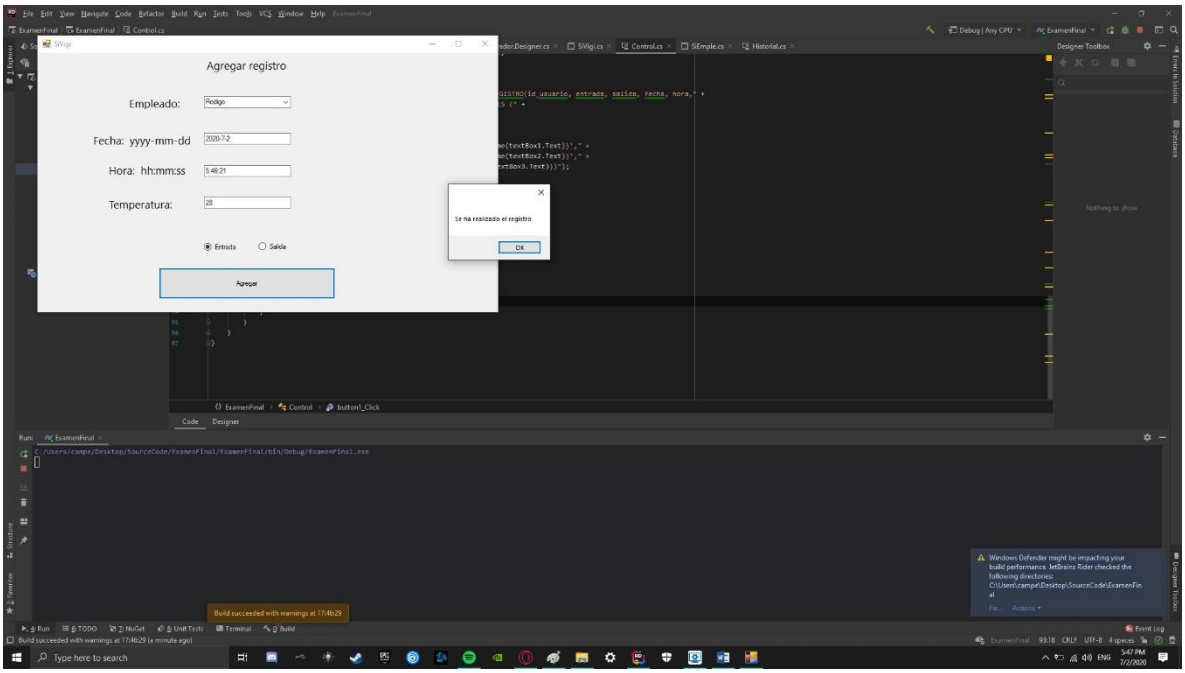

SiAdmin
Administrar BD
Registro
Actividad
Actividad de departamentos
Update
nombre
frecuencia
Administrativo
3
Code
Designer
Run: ExamenFinal
Build succeeded at 17:34:24
Type here to search

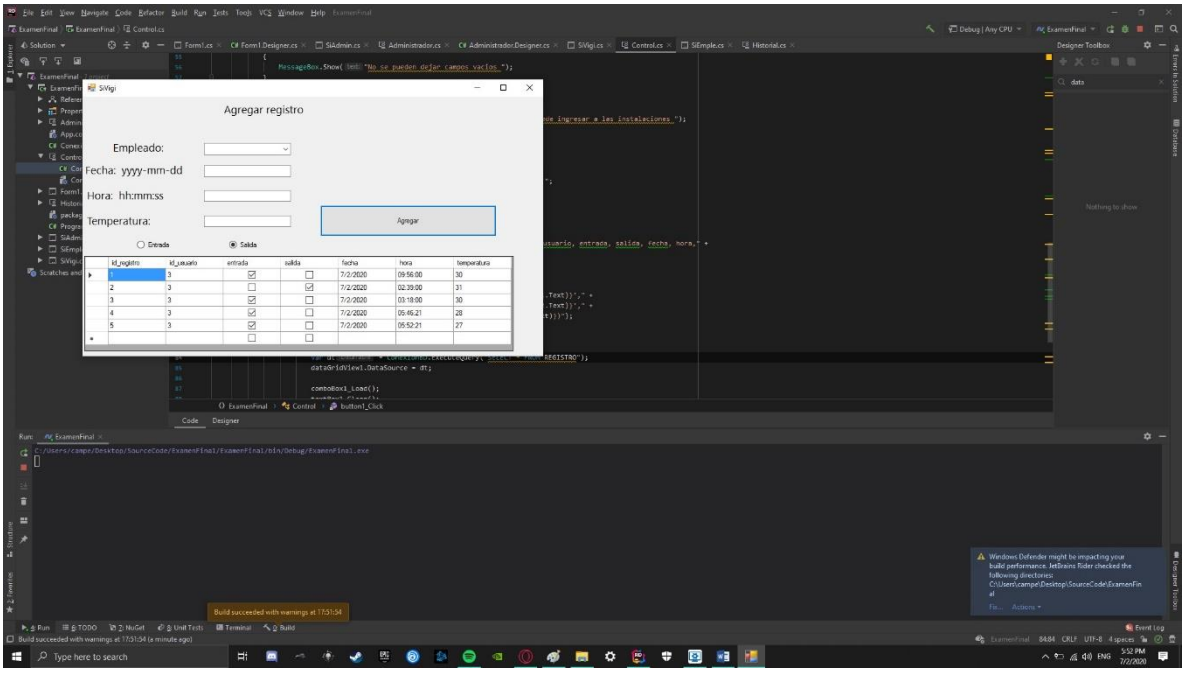
Empleado:



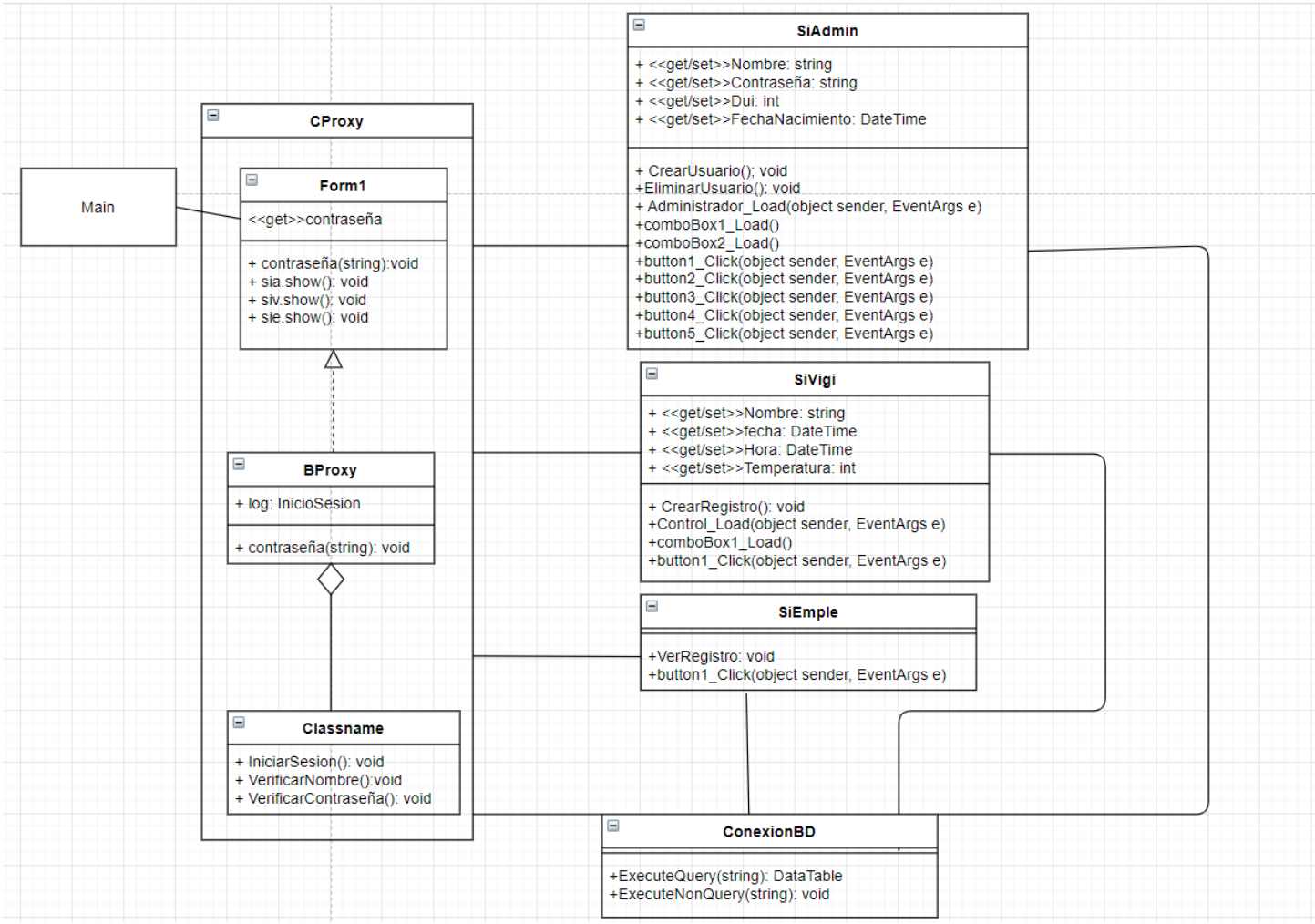
Vigilante:




Agregar registro
Empleado:
Rodigo
Fecha: yyyy-mm-dd
2020-7-2
Hora: hh:mm:ss
5:46:21
Temperatura:
28
Entrada
Salida
Agregar
Se ha realizado el registro
OK


Agregar registro
Empleado:
Fecha: yyyy-mm-dd
Hora: hh:mm:ss
Temperatura:
Agregar
Entrada
Salida
id_registro
id_usuario
entrada
salida
fecha
hora
temperatura

UML:



Patrones utilizados:

Proxy:

El patrón proxy se usa cuando se necesita una referencia a un objeto más flexible o sofisticada que un puntero. Tambien el patrón Proxy es un patrón estructural que tiene como propósito proporcionar un subrogado o intermediario de un objeto para controlar su acceso.

Mvc:

Es un patrón de arquitectura de software, que separa los datos y principalmente lo que es la lógica de negocio de una aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario. Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Script:

```
CREATE TABLE DEPARTAMENTO(
	id_departamento SERIAL PRIMARY KEY,
	nombre VARCHAR(100) NOT NULL,
	ubicacion VARCHAR(100) NOT NULL
);

CREATE TABLE USUARIO(
	id_usuario SERIAL PRIMARY KEY,
	id_departamento INT NOT NULL,
	contraseña VARCHAR(20) NOT NULL,
	nombre VARCHAR(50) NOT NULL,
	apellido VARCHAR(50) NOT NULL,
	dui INT NOT NULL,
	typeAdmin BOOL NOT NULL,
	typeVigi BOOL NOT NULL,
	typeEmple BOOL NOT NULL,
	fechaNacimiento DATE NOT NULL,
	CONSTRAINT fk_departamento_usuario
	FOREIGN KEY (id_departamento) REFERENCES DEPARTAMENTO(id_departamento)
);

CREATE TABLE REGISTRO(
	id_registro SERIAL PRIMARY KEY,
	id_usuario INT NOT NULL,
	entrada BOOL NOT NULL,
	salida BOOL NOT NULL,
	fecha DATE NOT NULL,
	hora TIME NOT NULL,
	temperatura DECIMAL NOT NULL,
	CONSTRAINT fk_usuario_registro
	FOREIGN KEY (id_usuario) REFERENCES USUARIO(id_usuario)
);

SELECT * FROM DEPARTAMENTO
SELECT * FROM USUARIO
SELECT * FROM REGISTRO

INSERT INTO DEPARTAMENTO VALUES (1, 'General', 'San Salvador'), (2, 'Vigilancia', 'San Salvador'), (3, 'Administrativo', 'San
Salvador')

INSERT INTO USUARIO VALUES (1, 1, 'admin', 'Sebastian', 'Ramirez', '100000001', TRUE, FALSE, FALSE, '1999-4-12'),
(2, 2, 'vigi', 'Carlos', 'Santana', '100000002', FALSE, TRUE, FALSE, '1999-4-12'),
(3, 3, 'emple', 'Rodigo', 'Gonzalez', '100000003', FALSE, FALSE, TRUE, '1999-5-12')

INSERT INTO REGISTRO VALUES (1, 3, TRUE, FALSE, '2020-7-2', '9:56:00', '30')

SELECT d.nombre, count(u.id_departamento) as frecuencia
FROM REGISTRO r, DEPARTAMENTO d, USUARIO u
WHERE r.id_usuario = u.id_usuario AND d.id_departamento = u.id_departamento
GROUP BY d.id_departamento
ORDER BY frecuencia DESC LIMIT 1;
```