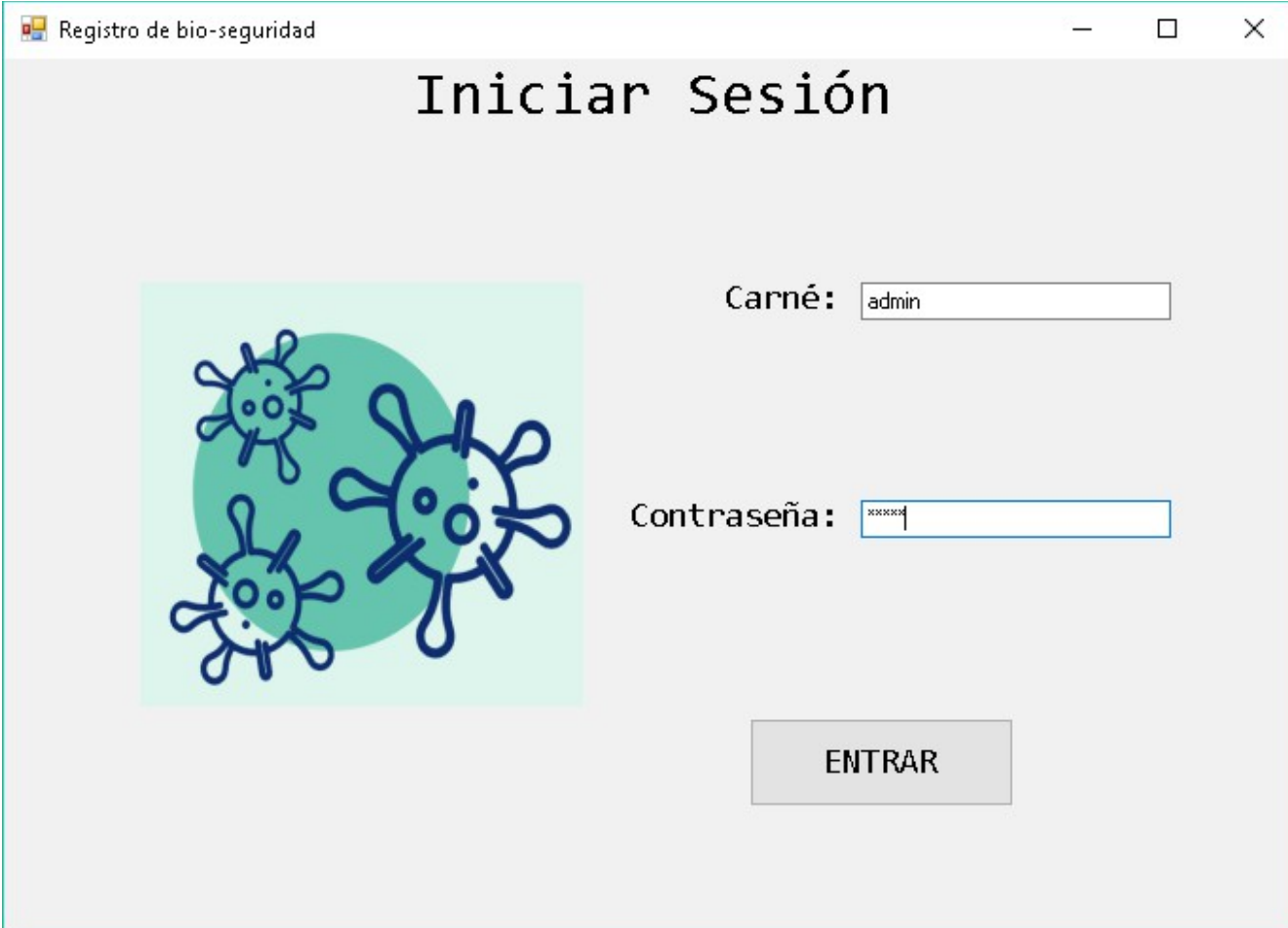


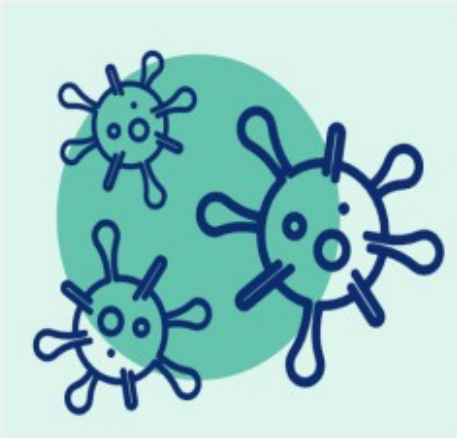
Manual tercer parcial.

Capturas de pantalla



Registro de bio-seguridad

Iniciar Sesión



Carné:

Contraseña:

ENTRAR

Iniciar sesión.

Registro de bio-seguridad

REGRESAR Bienvenid@ usuario [Administrador]

Mantenimiento de usuarios Resúmenes Resúmenes extra

Agregar usuario

idUsuario: wario123 Contraseña: wario123

Nombre: wilberto Apellido: salazar

DUI: 97943543 Fecha de nacimiento: 4/ 1/1950

Departamento: Administracion Limpiar AGREGAR

Eliminar usuario

Usuario agregado exitosadamente

Aceptar

ELIMINAR

Agregar usuario como administrador.

Registro de bio-seguridad

REGRESAR Bienvenid@ usuario [Administrador]

Mantenimiento de usuarios Resúmenes Resúmenes extra

Agregar usuario

idUsuario: Contraseña:

Nombre: Apellido:

DUI: Fecha de nacimiento:

Departamento: Administracion Limpiar

Eliminar usuario

Usuario borrado exitosadamente

Aceptar

idUsuario: wario123 ELIMINAR

Eliminar usuario como administrador.

Registro de bio-seguridad

REGRESAR

Bienvenid@ usuario [Administrador]

Mantenimiento de usuarios Resúmenes Resúmenes extra

Empleados dentro de la empresa

	idUsuario	nombre	apellido
▶	juan123	juan	gallardo
	mario300	mario	aurelio
	frank400	franciso	flores

Departamento más concurrido:

Vigilancia

5

Registro General

	idRegistro	entrada	fechaYhora	temperatura	idUsuario
▶	1	<input checked="" type="checkbox"/>	7/2/2020	38	admin
	2	<input checked="" type="checkbox"/>	7/3/2020	37	juan123
	3	<input checked="" type="checkbox"/>	7/22/2020	38	juan123
	4	<input checked="" type="checkbox"/>	7/28/2020	39	marco100
	5	<input type="checkbox"/>	7/8/2020	40	marco100
	6	<input checked="" type="checkbox"/>	7/8/2020	36	mauri200

Ver resúmenes como administrador.

Registro de bio-seguridad

REGRESAR

Bienvenid@ mauricio [Administrador]

Mantenimiento de usuarios Resúmenes Resúmenes extra

Mayores temperaturas

	nombre	frecuencia
▶	frank400	41
	admin	40
	mario300	40
	marco100	40
	mario300	40

Empleados mayores de 60 años

	nombre	frecuencia
▶	mario300	63
	frank400	73

Ver resúmenes extra como administrador.

Registro de bio-seguridad

REGRESAR

Bienvenid@ juan [Seguridad]

Registro de temperaturas

Escoger

☒ Entrada ☐ Salida

Fecha y hora: 10:33:58 PM

Temperatura: 30

idUsuario: frank400

AGREGAR

Empleados actualmente dentro

	idUsuario	nombre	apellido
▶	juan123	juan	gallardo
	mario300	mario	aurelio
	frank400	franciso	flores

Registro agregado exitosamente

Aceptar

Registrar temperaturas como vigilante.

Registro de bio-seguridad

REGRESAR

Bienvenid@ marco [Personal]

Mayor temperatura registrada: 40

Histórico de Entradas/Salidas

	idRegistro	entrada	fechaYhora	temperatura	idUsuario
▶	5	<input type="checkbox"/>	7/8/2020	40	marco100
	4	<input checked="" type="checkbox"/>	7/28/2020	39	marco100

Interfaz del personal: mayor temperatura e histórico de sus entradas/salidas.

Redimensionamiento dinámico implementado.

Patrones aplicados

1) Proxy:

Apliqué el patrón Proxy en el inicio de sesión de mi programa.

El patrón proxy permite controlar el acceso a un objeto específico, y en este caso, controla el primer acceso a la base de datos para verificar si el identificador de usuario y la contraseña corresponden a las que se encuentran guardadas en la base y así iniciar sesión.

Se podría decir que la utilización del patrón es como proxy de protección, para proteger las credenciales de acceso del cliente.

El proxy se ha implementado usando la clase ProxyAcceso en la cual se anida la clase Acceso que contiene los métodos necesario para el inicio de sesión. Así mismo, contiene una clase ProxySeguro que hereda el método de PeticiónAcceso que se encarga de realizar la petición si no encuentra problemas con las credenciales ingresadas por el usuario.

2) Estrategia.

Apliqué el patrón estrategia para usar tres diferentes algoritmos de verificación que permiten detectar el departamento al que pertenece el usuario que inicia sesión y luego poder abrir la interfaz indicada.

El patrón estrategia permite escoger de entre varios algoritmos similares pero con algunos comportamientos diferentes.

El objetivo de la estrategia en este caso es también detallar en el código la lógica de la empresa que se

ha tratado de simular, lo que facilita la lectura y modificación de la interfaz.

La estrategia se ha implementado usando una interfaz IDepartamento que deja como herencia un método de verificación a tres otras clases CAdministración, CPersonal y CVigilancia.

En dichas clases es que se aplican los métodos concretos cuyo objetivo es buscar el departamento al cual pertenece el usuario actual.

Delegates

Solamente logré implementar dos instancias de delegate, las dos con la misma función pero en dos UserControls diferentes.

La implementación hace referencia a un método ActualizarControles y a un método ActualizarDataGrids para invocar el delegate ActualizaciónGeneral en varios niveles del código.

Script de la base de datos

```
CREATE DATABASE "POOF"
```

```
WITH
```

```
OWNER = postgres
```

```
ENCODING = 'UTF8'
```

```
LC_COLLATE = 'English_United States.1252'
```

```
LC_CTYPE = 'English_United States.1252'
```

```
TABLESPACE = pg_default
```

```
CONNECTION LIMIT = -1;
```

```
CREATE TABLE departamento
```

```
(
```

```
idDepartamento serial primary key,
```

```
nombre varchar(50) not null,
```

```
ubicacion varchar(200) not null
```

```
);
```

```
ALTER TABLE departamento
```

```
OWNER to postgres;
```

```
CREATE TABLE usuario
```

```
(
  idUsuario varchar(50) primary key,
  contrasena varchar(50) not null,
  nombre      varchar(50) not null,
  apellido varchar(50) not null,
  dui      int unique not null,
  fechaNacimiento date not null,
  idDepartamento int references departamento(idDepartamento)
```

```
ON DELETE CASCADE
```

```
ON UPDATE CASCADE
```

```
);
ALTER TABLE usuario
  OWNER to postgres;
```

```
CREATE TABLE registro
```

```
(
  idRegistro serial primary key,
  entrada bool not null,
  fechaYhora date not null,
  temperatura int not null,
  idUsuario varchar(50) references usuario(idUsuario)
```

```
ON DELETE CASCADE
```

```
ON UPDATE CASCADE
```

```
);
ALTER TABLE registro
  OWNER to postgres;
```

```
insert into departamento (nombre, ubicacion)
values ('Administracion', 'Edificio anexo');
```

```
insert into departamento (nombre, ubicacion)
values ('Vigilancia', 'Porton delantero');
```

```
insert into departamento (nombre, ubicacion)
values ('Personal', 'Edificio de oficinas');
```

```
insert into usuario
values ('admin', 'admin', 'usuario', 'administrador', '93424785', '1995-04-23', 1);
```

Problemas encontrados

- Al cargar la aplicación sin ningún registro en la base de datos, el código cae en una excepción controlada porque no pudo cargar las listas necesarias para los Resúmenes. Ya no tuve tiempo de resolverlo.
- Utilicé una clase objeto Frecuencia para realizar los DataGrids de los Resúmenes Extra pero olvidé que temperatura y edad aparecerían como frecuencias en los Grids respectivos.