

UNIVERSIDAD CENTROAMERICANA JOSE SIMEON CAÑAS UCA

Manual Técnico Parcial 3 Programación Orientada Objetos

Realizado por:

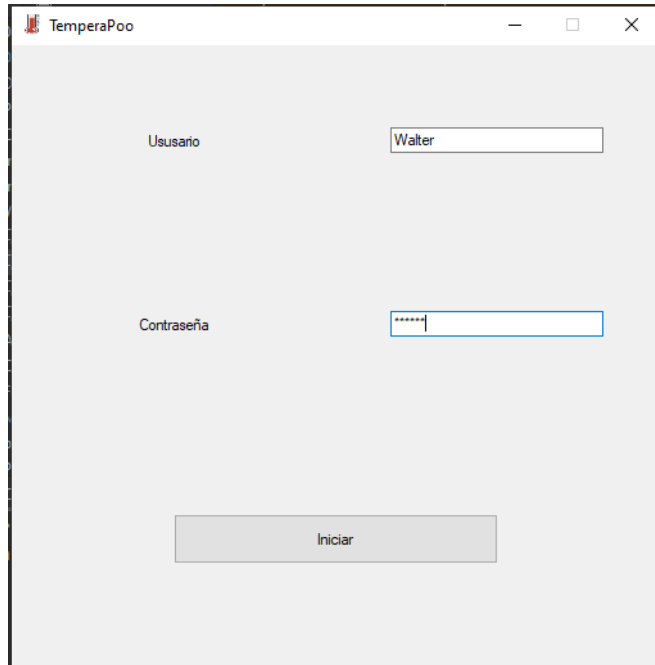
Rigoberto Campos Vega, 00081511

Contenido.

1. **Capturas de Pantalla**
2. **UML**
3. **Patrones de Diseño**
4. **Base de datos**

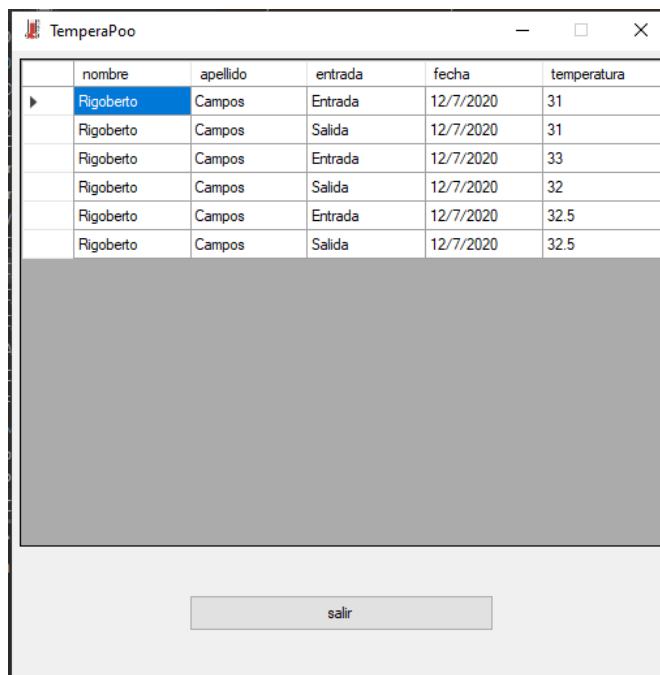
1. Capturas de Pantalla.

Login.



The screenshot shows a login window titled "TemperaPoo". It contains two input fields: "Usuario" with the text "Walter" and "Contraseña" with masked characters "*****". Below the fields is a button labeled "Iniciar".

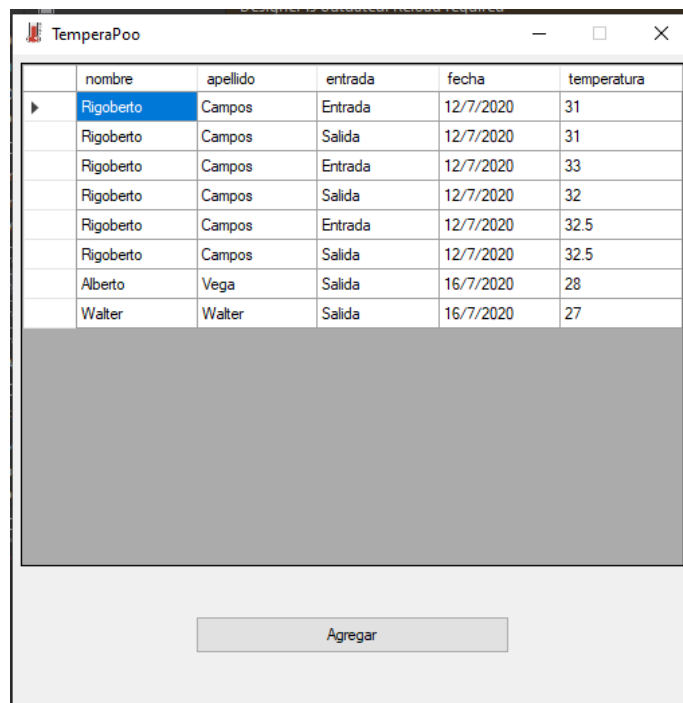
Vista Empleado.



The screenshot shows an employee view window titled "TemperaPoo". It displays a table with employee data. The first row is highlighted in blue. Below the table is a large gray rectangular area and a button labeled "salir".

	nombre	apellido	entrada	fecha	temperatura
▶	Rigoberto	Campos	Entrada	12/7/2020	31
	Rigoberto	Campos	Salida	12/7/2020	31
	Rigoberto	Campos	Entrada	12/7/2020	33
	Rigoberto	Campos	Salida	12/7/2020	32
	Rigoberto	Campos	Entrada	12/7/2020	32.5
	Rigoberto	Campos	Salida	12/7/2020	32.5

Vista Seguridad.

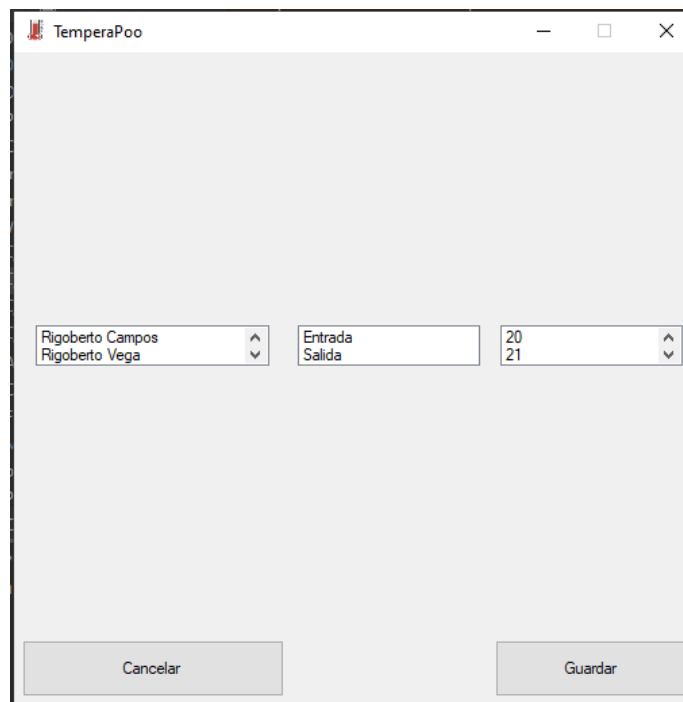


The screenshot shows a window titled 'TemperaPoo' with a standard Windows title bar (minimize, maximize, close buttons). The main content area contains a table with the following columns: 'nombre', 'apellido', 'entrada', 'fecha', and 'temperatura'. The first row is highlighted in blue. Below the table is a large gray rectangular area, and at the bottom center is a button labeled 'Agregar'.

	nombre	apellido	entrada	fecha	temperatura
▶	Rigoberto	Campos	Entrada	12/7/2020	31
	Rigoberto	Campos	Salida	12/7/2020	31
	Rigoberto	Campos	Entrada	12/7/2020	33
	Rigoberto	Campos	Salida	12/7/2020	32
	Rigoberto	Campos	Entrada	12/7/2020	32.5
	Rigoberto	Campos	Salida	12/7/2020	32.5
	Alberto	Vega	Salida	16/7/2020	28
	Walter	Walter	Salida	16/7/2020	27

Agregar

Vista agregar registro.



The screenshot shows a window titled 'TemperaPoo' with a standard Windows title bar. The main content area is mostly empty. At the bottom, there are three input fields: a dropdown menu with 'Rigoberto Campos' and 'Rigoberto Vega' as options, a text box containing 'Entrada' and 'Salida', and a spinner box with values '20' and '21'. Below these fields are two buttons: 'Cancelar' and 'Guardar'.

Rigoberto Campos
Rigoberto Vega

Entrada
Salida

20
21

Cancelar Guardar

Vista Admin.

TemperaPoo					
	nombre	apellido	entrada	fecha	temperatura
▶	Rigoberto	Campos	Entrada	12/7/2020	31
	Rigoberto	Campos	Salida	12/7/2020	31
	Rigoberto	Campos	Entrada	12/7/2020	33
	Rigoberto	Campos	Salida	12/7/2020	32
	Rigoberto	Campos	Entrada	12/7/2020	32.5
	Rigoberto	Campos	Salida	12/7/2020	32.5
	Alberto	Vega	Salida	16/7/2020	28
	Walter	Walter	Salida	16/7/2020	27
	Walter	Walter	Entrada	16/7/2020	29
Administrar					

Vista Información (Datos)

Usuarios Adentro			Top Temperatura	
	entrada	nombre	id_usuario	nombre
▶	12/7/2020	Rigoberto Campos	1	Rigoberto Campos
*			4	Alberto Vega
			5	Walter Walter
			*	
Top Departamentos			Personas Mayores	
	count	nombre	nombre	years
▶	1	Planta	*	
*				
Regresar			Regresar	

2. UML

3. Patrones de Diseño

- **Singleton.**

Se usa este patrón de diseño en la clase Manage, esta clase es la encargada de los movimientos de pantalla y de llamar a la base de datos, así como de lógica de consola. Ya que esta clase se encarga de administrar el programa no es necesario que tenga un objeto como tal y solamente se quiere una durante la ejecución del programa, por eso se optó por este patrón de diseño.

- **Interfaces.**

Se usa una interface de usuario, de esta heredan las tres clases de usuario que tiene el programa, ya que, aunque todos sean usuarios poseen distintos niveles de acceso en la aplicación y distintas funcionalidades, por eso se optó usar la interface.

- **Delegados.**

Se usan delegados en la aplicación para hacer la interacción entre la pantalla de registros y las otras pantallas, ya que cada usuario tiene niveles distintos de acceso el programa asigna la función para mostrar la pantalla de agregar registro o mostrar los datos resumen, según el perfil de usuario que haya ingresado.

4. Base de datos.

Creación.

```
CREATE TABLE public.departamento
(
    id_departamento integer NOT NULL DEFAULT
nextval('departamento_id_departamento_seq'::regclass),
    nombre character varying COLLATE pg_catalog."default",
    ubicacion character varying COLLATE pg_catalog."default",
    CONSTRAINT departamento_pkey PRIMARY KEY (id_departamento)
);

CREATE TABLE public.usuario
(
    id_usuario integer NOT NULL DEFAULT nextval('usuario_id_usuario_seq'::regclass),
    id_departamento integer,
    nombre character varying COLLATE pg_catalog."default",
    apellido character varying COLLATE pg_catalog."default",
    dui character varying(16) COLLATE pg_catalog."default",
    fecha_nacimiento date,
    CONSTRAINT usuario_pkey PRIMARY KEY (id_usuario),
    CONSTRAINT usuario_id_departamento_fkey FOREIGN KEY (id_departamento)
REFERENCES public.departamento (id_departamento) MATCH SIMPLE
ON UPDATE NO ACTION
ON DELETE NO ACTION
)
```



```

CREATE TABLE public.registro
(
    id_registro integer NOT NULL DEFAULT nextval('registro_id_registro_seq'::regclass),
    id_usuario integer,
    entrada integer,
    fecha date,
    temperatura numeric,
    CONSTRAINT registro_pkey PRIMARY KEY (id_registro),
    CONSTRAINT registro_id_usuario_fkey FOREIGN KEY (id_usuario)
        REFERENCES public.usuario (id_usuario) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION
)

```

Querys usadas en la aplicación:

Usuarios Adentro.

```

select a.fecha as entrada, b.nombre || ' ' || b.apellido "
" as nombre from ("+
    " select distinct on (id_usuario)
    id_usuario, fecha, entrada"+
    " from registro"+
    " order by id_usuario, fecha desc) a
inner join usuario b "+
    " on a.id_usuario = b.id_usuario
where entrada = 1

```

Top Departamentos

```

select count(b.id_usuario), c.nombre from ("+
    " select distinct on (id_usuario)
    id_usuario, fecha, entrada"+
    " from registro"+
    " order by id_usuario, fecha desc) a
inner join usuario b "+
    " on a.id_usuario = b.id_usuario
inner join departamento c on "+
    " b.id_departamento =
    c.id_departamento"+
    " where entrada = 1 group by c.nombre

```

Top Temperaturas

```
select distinct on (a.id_usuario) a.id_usuario,"+
                                " b.nombre || ' ' ||b.apellido as
nombre, temperatura"+
                                " from registro a inner join usuario
b"+
                                " on a.id_usuario = b.id_usuario"+
                                " order by id_usuario, temperatura
desc
```

Personas Mayores a 60

```
select nombre || ' ' || apellido as nombre,"+
                                "DATE_PART('year', AGE(current_date,
fecha_nacimiento)) AS years "+
                                "from usuario where DATE_PART('year',
AGE(current_date, fecha_nacimiento)) > 59
```

Guardar Registro nuevo

```
INSERT INTO REGISTRO (ID_USUARIO, FECHA, ENTRADA, TEMPERATURA)" +
                                $" SELECT ID_USUARIO,
CURRENT_DATE,'{entrada}', " +
                                $"'{temperatura}' from usuario " +
                                $"where nombre || ' ' || apellido
='{nombre}'
```

Datos en las tablas.

Select * from departamento		
1	Planta	Planta
2	Seguridad	Entrada
3	Gerente	Oficina

Select * from usuario					
1	1	Rigoberto	Campos	81511	12/7/2020
2	1	Rigoberto	Vega	81512	12/7/2020
3	1	Alberto	Campos	81513	12/7/2020
4	1	Alberto	Vega	81514	12/7/2020
5	2	Walter	Walter	81515	12/7/2020
6	3	Rigoberto	Rigoberto	345081515	12/7/2020

Select * from registro				
1	1	1	12/7/2020	31
2	1	0	12/7/2020	31
3	1	1	12/7/2020	33
4	1	0	12/7/2020	32
5	1	1	12/7/2020	32.5
6	1	0	12/7/2020	32.5
7	4	0	16/7/2020	28
8	5	0	16/7/2020	27