

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA:		ASIGNATURA: Programación Aplicada	
NRO. PRÁCTICA:		TÍTULO PRÁCTICA: Prueba_EnunciadoJPA	
OBJETIVO ALCANZADO: <ul style="list-style-type: none"> Consolidar los conocimientos adquiridos en clase sobre JPA. 			
ACTIVIDADES DESARROLLADAS			
https://github.com/ddutan2000/Prueba_EnunciadoJPA.git			
1. Se pidió desarrollar un programa que simule los estimados de hipoteca Se ha creado diversos paquetes para poder manejar el programa con base de datos. <ul style="list-style-type: none"> Ec.edu.ups.controlador Ec.edu.ups.modelo Ec.edu.ups.vista Ec.edu.ups.utils Ec.edu.ups.excepciones META-INF 			
2. Ec.edu.ups.excepciones En este paquete se creo la clase Utils la cual no sirve para inicializar nuestro entityManager			
3. META-INF En este paquete se creo un objeto “persistence.xml” la cual no sirve para conectar nuestros modelos a la base de datos que en este caso se llama “prueba”.			
4. Ec.edu.ups.modelo Dentro de este paquete se crearon todos los modelos para poder conectarlos a nuestra case de datos. de la misma manera cada clase en este paquete es un Entity class. <ul style="list-style-type: none"> Casa Hipoteca Persona Usuario 			
5. Casa Esta clase implementa la interfaz serializable. Esta clase utiliza el lenguaje JPQL para crear las columnas y las relaciones de algunos objetos con otros.			

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Atributos:

- @Id
 - @Column
- private int codigoCasa;
- @Column
- private String direccionCalle1;
- @Column
- private String direccionCalle2;
- @Column
- private double valorDeCasa;
- @ManyToOne
 - @JoinColumn(name="fk_persona")
- private Persona persona;


Cada atributo tiene instanciado su getter y setter.

6. Hipoteca

Esta clase implementa la interfaz serializable. Esta clase utiliza el lenguaje JPQL para crear las columnas y las relaciones de algunos objetos con otros.

Atributos:

- private static final long serialVersionUID = 1L;
- @Id
 - @GeneratedValue(strategy = GenerationType.AUTO)
- private Long id;
- @Column
- private String CedulaCliente;
- @Column
- private String CedulaGarante;
- @Column
- private double interes;
- @Column

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

- private double prestamo ;
@Column
- private String LapzoDeTiempo;

Cada atributo tiene instanciado su getter y setter. El atributo id tiene instanciado su hashCode y Equals para generar un código único automáticamente.

7. Persona

Esta clase implementa la interfaz serializable. Esta clase utiliza el lenguaje JPQL para crear las columnas y las relaciones de algunos objetos con otros.

Atributos:


- @Id
@Column
- private String cedula;

@Column
- private String nombre;
@Column
- private String apellido;
@Column
- private String telefono;
@Column
@Temporal(TemporalType.DATE)
- private Date fechaDeNacimiento;
@Column
- private double salarioMensual;
- @OneToMany(mappedBy = "persona", cascade = CascadeType.ALL)
- private List<Casa> casas;

Cada atributo tiene instanciado su getter y setter. El atributo casa tiene una relación con un person de tipo “oneToMany”.

8. Usuario

Esta clase implementa la interfaz serializable. Esta clase utiliza el lenguaje JPQL para crear las columnas y las relaciones de algunos objetos con otros.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

Atributos:

- @Id
- @Column
- private String cedula;
- @Column
- private String correo;
- @Column
- private String contrasenia;

Cada atributo tiene instanciado su getter y setter.

9. Ec.edu.ups.excepciones

En este paquete se crearon excepciones personales en las cuales se utilizarán en caso de que ocurra un error en el programa.

- ExcepcionBuscar
- ExcepcionActualizar

10. Ec.edu.ups.controlador

En este paquete se crearon todos los controladores para que interactúen con los modelos y la base de datos.

Controladores:


- ControladorGenerico
- ControladorCasa
- ControladorHipoteca
- ControladorPersona
- ControladorRegex
- ControladorUsuario

11. Controladro Generico

En clase es una clase abstracta genérica que se crearon todos los métodos principales solamente para poder utilizar la clase y ahorrar mas códigos.

Atributos:

- private Class<T> clase;
- private EntityManager em;

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

En esta clase se crearon dos controladores en las cuales se inicializa los atributos.

Métodos:

- **Créate:** Este método recibe en su parámetro un objeto. Después se integra una excepción en caso de alguno error, después se inicializa la base de datos y se crea el objeto y al final se hace un commit.
- **Update:** Este método recibe en su parámetro un objeto. Después se integra una excepción en caso de alguno error, después se inicializa la base de datos y se actualiza el objeto y al final se hace un commit.
- **Read:** Este método recibe un id en su parámetro. Después se integra una excepción en caso de alguna falla al momento de lectura. Luego se utiliza el entityManeger para poder encontrar el objeto con el id que se recibió en los parámetros.
- **Delete:** Este método recibe en su parámetro un objeto. Después se integra una excepción en caso de alguno error, después se inicializa la base de datos y se elimina el objeto y al final se hace un commit.
- **findAll:** este método utiliza el lenguaje de JPQL para que devuelva una lista de objetos dentro de la base de datos. también se utiliza un query para estos tipos de consultas por ultimo se retorna la lista o un null.
- **validarActualizacion:** Este es un método abstracto que se utiliza para poder desarrollar las excepciones.
- **validarExistente:** Este es un método abstracto que se utiliza para poder desarrollar las excepciones.

Por ultimo se crearon los getters y setters de los atributos.

12. ControladorCasa

Este método es un extends del controladorGenerico.

Se reutiliza todos los métodos de la clase genérica y por ultimo se especifica los métodos abstractos para poder implementar las excepciones con mas éxito.

13. ControladorHipoteca


Este método es un extends del controladorGenerico.

Se reutiliza todos los métodos de la clase genérica y por ultimo se especifica los métodos abstractos para poder implementar las excepciones con mas éxito.

14. ControladorPersona

Este método es un extends del controladorGenerico.

Se reutiliza todos los métodos de la clase genérica y por ultimo se especifica los métodos abstractos para poder implementar las excepciones con mas éxito.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

15. ControladorUsuario

Este método es un extends del controladorGenerico.

Se reutiliza todos los métodos de la clase genérica y por ultimo se especifica los métodos abstractos para poder implementar las excepciones con mas éxito.

16. ControladorRegex

Este controlador se utiliza para poder crear validación al poder crear patrones personalizados.

Atributos:

- private Pattern patron;
- private Matcher corpus;

métodos:

- ingresarRegex: Este método recibe un objeto de tipo String en su parámetro. Este método realiza la tarea de reasignar el patrón.
- validar: Este método recibe un objeto de tipo String en su parámetro. Este metodo valida si el String cumple con el patrón asignado. Si cumple retorna true, caso contrario retorna un false.

17. Ec.edu.ups.vista

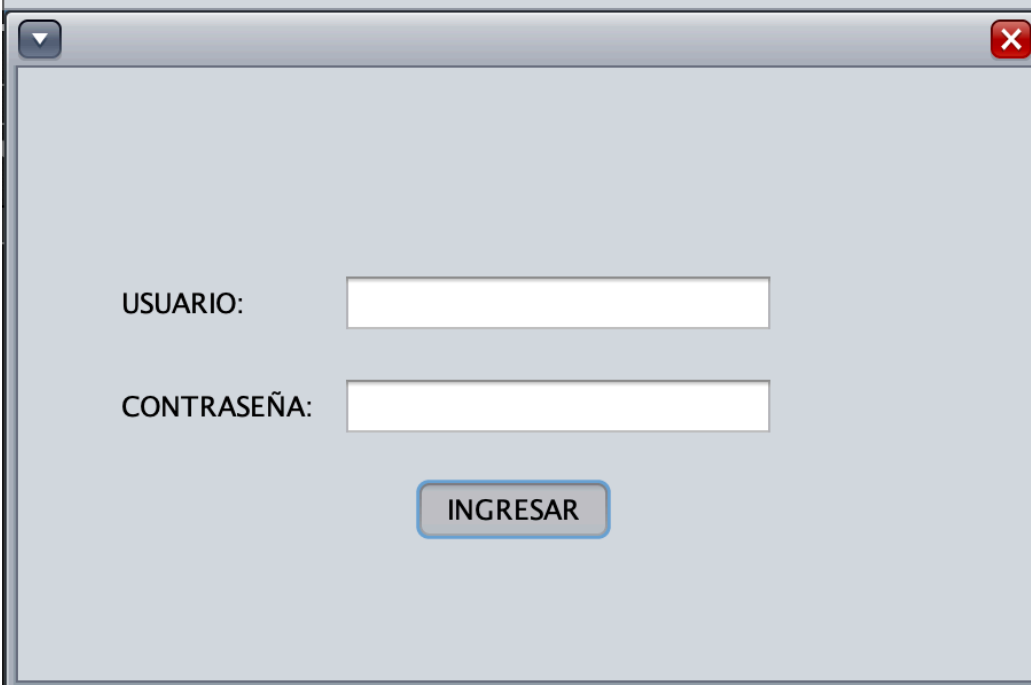
En este paquete se crearon todas las interfaces con las cuales el usuario podrá interactuar con el programa.

- IniciarSession
- RegistrarCasa
- RegistrarPersona
- RegistrarUsuario
- RegistrarHipoteca
- VentanaPrincipal

18. IniciarSession

Esta interfaz se utiliza para poder acceder a la aplicación y poder realizar hipotecas.

MENU



USUARIO:

CONTRASEÑA:

INGRESAR

19. RegistrarCasa

Esta interfaz se utiliza para poder registrar una cas de una persona.

MENU BANCO

DIRECCION

CALLE 1:

CALLE 2:

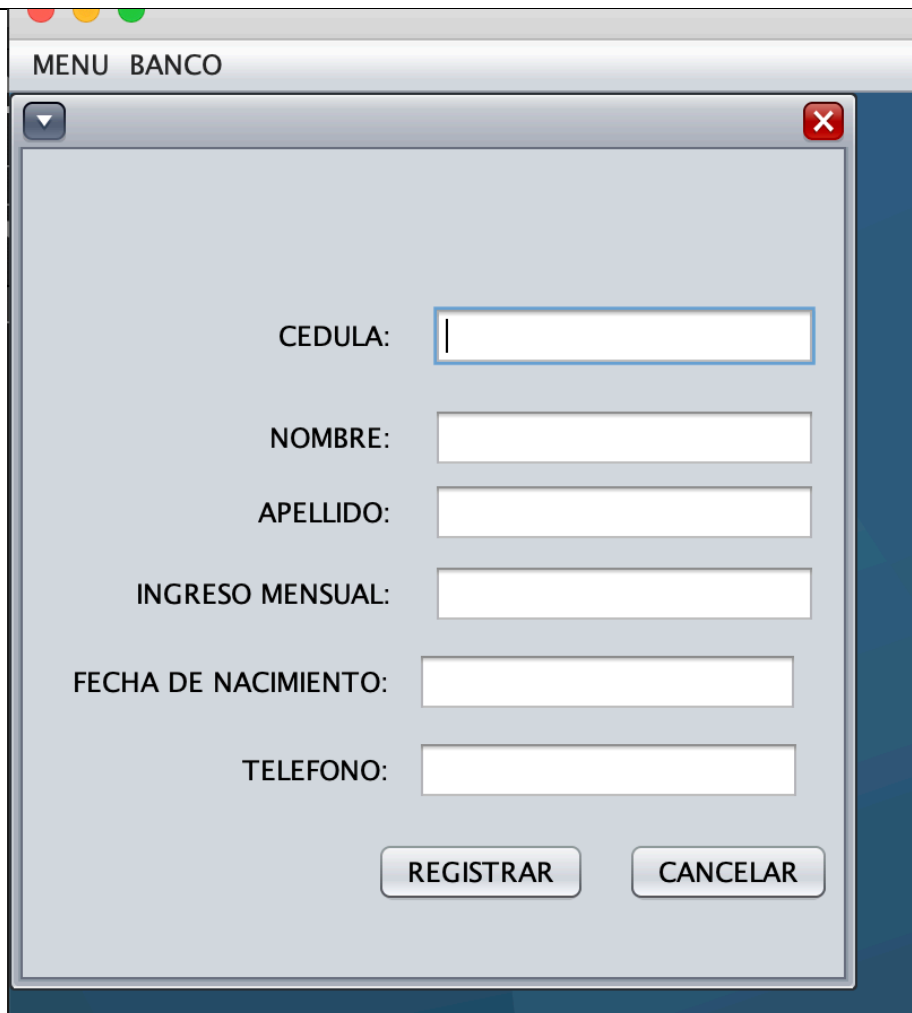
VALOR:

CEDULA:

REGISTRAR CANCELAR

20. RegistrarPersona

Esta interfaz se utiliza para poder crear una persona y registrarla dentro del sistema.



MENU BANCO

▼

CEDULA:

NOMBRE:

APELLIDO:

INGRESO MENSUAL:


FECHA DE NACIMIENTO:

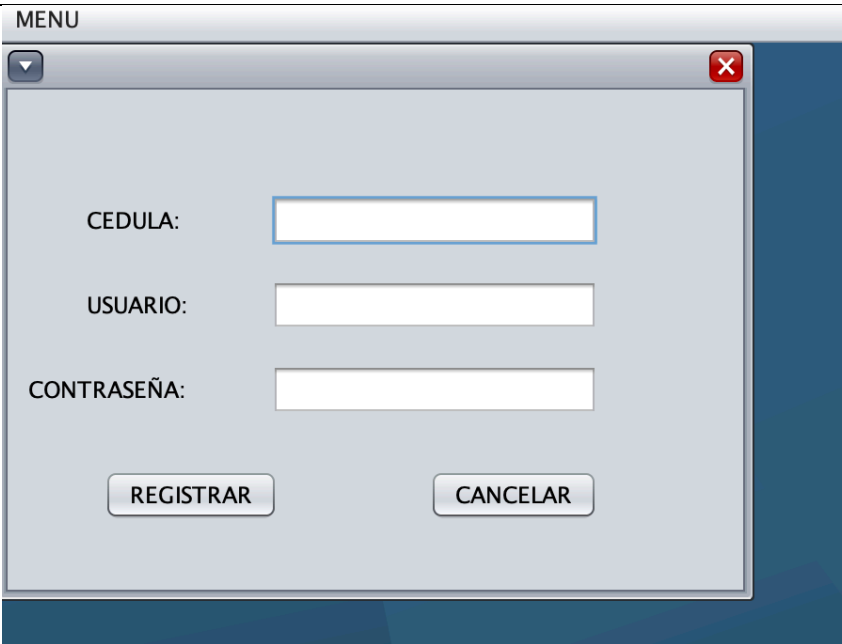
TELEFONO:

REGISTRAR CANCELAR

21. RegistrarUsuario

Esta interfaz crea un usuario para poder acceder a la aplicación.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



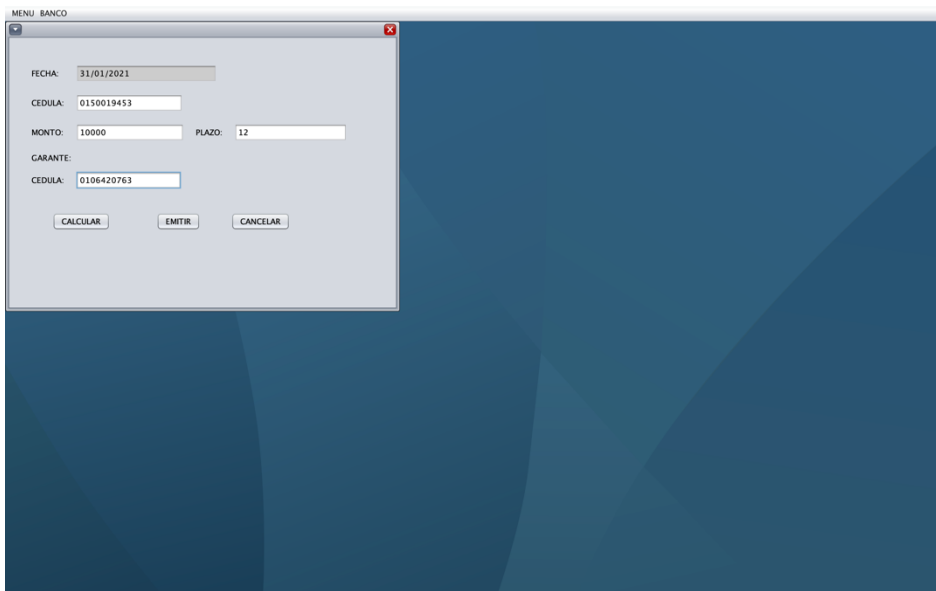
MENU

CEDULA:
 USUARIO:
 CONTRASEÑA:

REGISTRAR CANCELAR

22. RegistrarHipoteca

Esta interfaz calcula la hipoteca del usuario y realiza un pdf con los pagos y el interés a pagar por cada mes.



MENU BANCO

FECHA: 31/01/2021
 CEDULA: 0150019453
 MONTO: 10000 PLAZO: 12
 GARANTE:
 CEDULA: 0106420763

CALCULAR EMITIR CANCELAR

BANCO PAUTE

SOLICITANTE

CEDULA	NOMBRE	APELLIDO	TELEFONO
0150019453	Denys	Dutan	7703552458


GARANTE

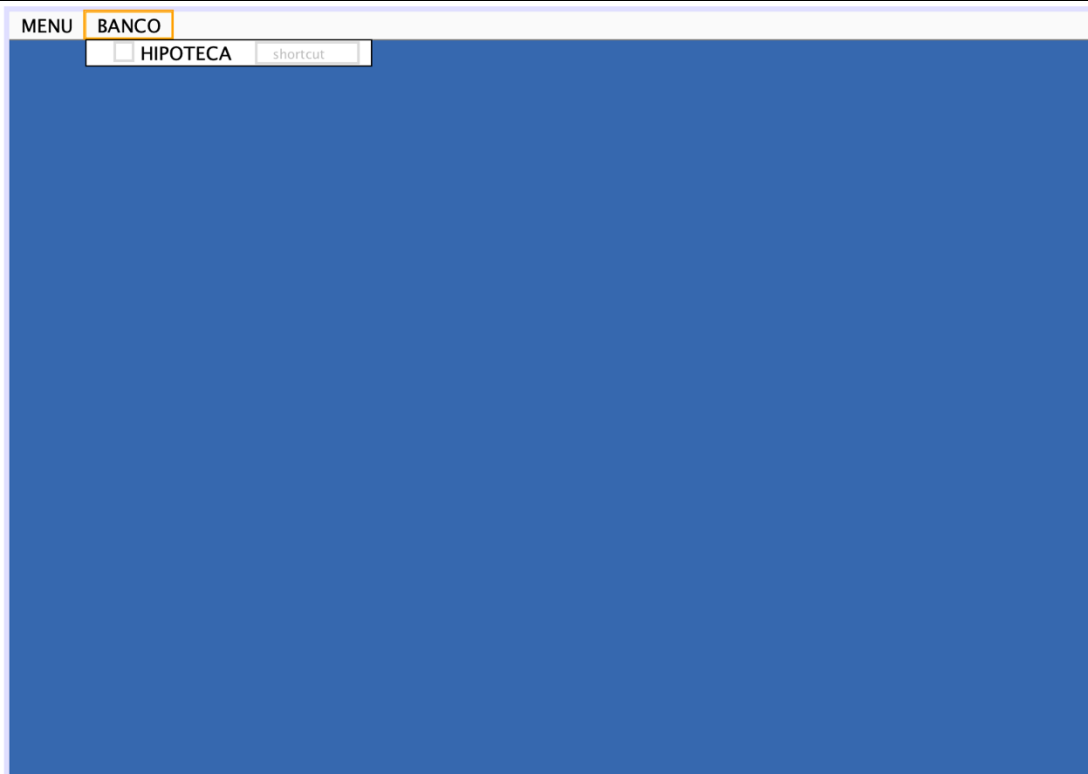
CEDULA	NOMBRE	APELLIDO	TELEFONO
0106420763	Paul	Idrovo	0987654321

MES	CAPITAL	SALDO	INTERES	TOTAL
1	833.33	10,000.00	141.67	975.00
2	833.33	9,166.67	129.86	963.19
3	833.33	8,333.33	118.06	951.39
4	833.33	7,500.00	106.25	939.58
5	833.33	6,666.67	94.44	927.78
6	833.33	5,833.33	82.64	915.97
7	833.33	5,000.00	70.83	904.17
8	833.33	4,166.67	59.03	892.36
9	833.33	3,333.33	47.22	880.56
10	833.33	2,500.00	35.42	868.75
11	833.33	1,666.67	23.61	856.94
12	833.33	833.33	11.81	845.14

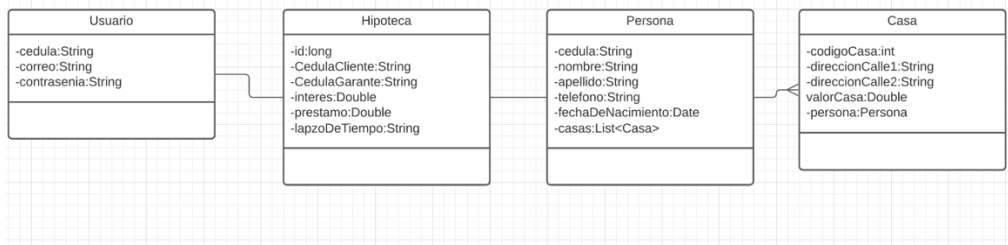
23. VentanaPrincipal

En esta interfaz se inicializan todas las demás interfaces para poder interactuar con ellas.

	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		



24.



25.

RESULTADO(S) OBTENIDO(S):

BANCO PAUTE

SOLICITANTE

CEDULA	NOMBRE	APELLIDO	TELEFONO
0150019453	Denys	Dutan	7703552458

GARANTE

CEDULA	NOMBRE	APELLIDO	TELEFONO
0106420763	Paul	Idrovo	0987654321

MES	CAPITAL	SALDO	INTERES	TOTAL
1	833.33	10,000.00	141.67	975.00
2	833.33	9,166.67	129.86	963.19
3	833.33	8,333.33	118.06	951.39
4	833.33	7,500.00	106.25	939.58
5	833.33	6,666.67	94.44	927.78
6	833.33	5,833.33	82.64	915.97
7	833.33	5,000.00	70.83	904.17
8	833.33	4,166.67	59.03	892.36
9	833.33	3,333.33	47.22	880.56
10	833.33	2,500.00	35.42	868.75
11	833.33	1,666.67	23.61	856.94
12	833.33	833.33	11.81	845.14

CONCLUSIONES:

En conclusión, se debe obtener un entendimiento de como utilizar clases genéricas y base de datos. Un detalle importante que también se debe tener en cuenta es que se debe tener un entendimiento de como codificar utilizando lenguaje JPQL. Por ultimo se debe comprender como utilizar el lenguaje JPQL para conectarse a la base de datos y ahorrarse tiempo codificando.

RECOMENDACIONES:

Nombre de estudiante: Denys Dutan

Firma de estudiante: 