
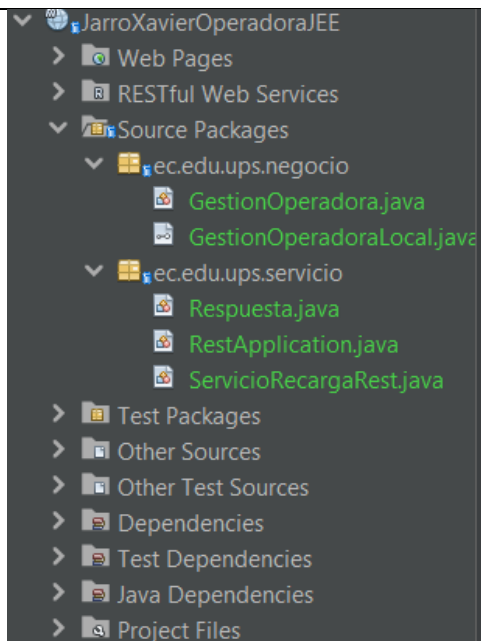
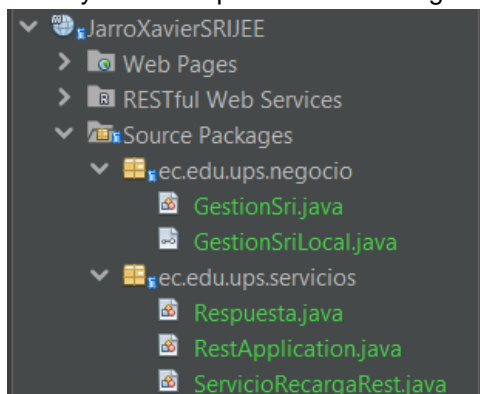
	VICERRECTORADO DOCENTE	Código: GUIA-PRL-001
	CONSEJO ACADÉMICO	Aprobación: 2016/04/06
Formato: Guía de Práctica de Laboratorio / Talleres / Centros de Simulación		

		FORMATO DE INFORME DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA ESTUDIANTES	
CARRERA: Ingeniería de Sistemas		ASIGNATURA: Aplicaciones Distribuidas	
NRO. PRÁCTICA:	1	TÍTULO PRÁCTICA: Examen Final	
ACTIVIDADES DESARROLLADAS			
<p>Arquitectura JEE.</p> <p>El Proyecto en la parte del cajero tiene dao,gestion de negocio, modelo de clases y servicios.</p>  <p>El Proyecto en la parte de la operadora tiene gestión de negocio y servicios.</p>			

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



El Proyecto en la parte del sri tiene gestión de negocio y servicios.




Las entidades se crearon conforme a los requerimientos de jee. Por ejemplo la clase recarga.

```
@Entity
public class Recarga implements Serializable {

    @Id
    private int codigo;
    @OneToOne
    @JoinColumn(name = "numero_cuenta")
    private Cuenta cuenta;
    private String numerCelular;
    private double valorRecarga;
    private String estado;
```

Creación de cuentas y usuarios desde la web.

Se tiene la siguiente interfaz donde se ingresa el numero de cedula, nombre y apellido del cliente, adicional se crea en la misma ventana la cuenta que estará enlazada con el cliente.

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

Inicio Cajero

Recarga
Crear Cuenta

Crear Cliente

Cedula

Nombre

Apellido

Cuenta de Ahorros

Numero de Cuenta

Saldo

Guardar

Al ingresar los datos se puede ver en la consola del servidor como hacer los queries respectivos para crear el cliente y la cuenta a la vez


```

Output x
WildFly Application Server x Run (Recarga) x
11:18:49,790 INFO [stdout] (default task-10) Hibernate:
11:18:49,790 INFO [stdout] (default task-10) insert
11:18:49,790 INFO [stdout] (default task-10) into
11:18:49,790 INFO [stdout] (default task-10) Cliente
11:18:49,790 INFO [stdout] (default task-10) (apellido, nombre, cedula_cliente)
11:18:49,790 INFO [stdout] (default task-10) values
11:18:49,790 INFO [stdout] (default task-10) (?, ?, ?)
11:18:49,807 INFO [stdout] (default task-10) Hibernate:
11:18:49,807 INFO [stdout] (default task-10) insert
11:18:49,807 INFO [stdout] (default task-10) into
11:18:49,807 INFO [stdout] (default task-10) Cuenta
11:18:49,807 INFO [stdout] (default task-10) (cedula_cliente, saldo, numero_cuenta)
11:18:49,807 INFO [stdout] (default task-10) values
11:18:49,807 INFO [stdout] (default task-10) (?, ?, ?)

```

También se puede verificar en la base de datos.

Tabla Clientes:

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

Data Output Explain Messages Notifications

	cedula_cliente [PK] character varying (255)	apellido character varying (255)	nombre character varying (255)
1	0104796230	jarro	xavier
2	0104796233	priena	priebe
3	9999999999	UPS	UPS

Tabla cuenta:

	numero_cuenta [PK] character varying (255)	saldo double precision	cedula_cliente character varying (255)
1	12345678	1500	0104796230
2	12345677	150	0104796233
3	123456789	4444	9999999999

Servicio de recarga en operadora.

Para rechazar o aprobar la recarga de parte de la operadora se implementó el siguiente método.


```
@Stateless
public class GestionOperadora implements GestionOperadoraLocal {

    public Respuesta aceptarRechazarRecarga() {
        Respuesta res = new Respuesta();

        int numero = (int) (Math.random() * 10 + 1);
        if (numero == 0) {
            res.setCodigo(1);
            res.setEstado("aprobado");
        } else {
            res.setCodigo(2);
            res.setEstado("rechazado");
        }

        return res;
    }
}
```

En la parte del web service se declaró de la siguiente manera:

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

```

@Path("/bo")
public class ServicioRecargaRest {

    @Inject
    private GestionOperadoraLocal on;

    @POST
    @Path("/transferenciaOperadora")
    @Produces("application/json")
    @Consumes("application/json")
    public Respuesta transferenciaOperadora() {
        return on.aceptarRechazarRecarga();
    }

}

```

Servicio de autorización de factura en SRI.

Para rechazar o autorizar la factura en sri se implementó el siguiente método.

```

@Stateless
public class GestionSri implements GestionSriLocal {

    public Respuesta aceptarRechazarFactura() {
        Respuesta res = new Respuesta();


        int numero = (int) (Math.random() * 10 + 1);
        if (numero == 0) {
            res.setCodigo(1);
            res.setEstado("aprobado");
        } else {
            res.setCodigo(2);
            res.setEstado("rechazado");
        }

        return res;
    }

}

```

En la parte del web service se declaró de la siguiente manera:

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

```
@Path("/bs")
public class ServicioRecargaRest {

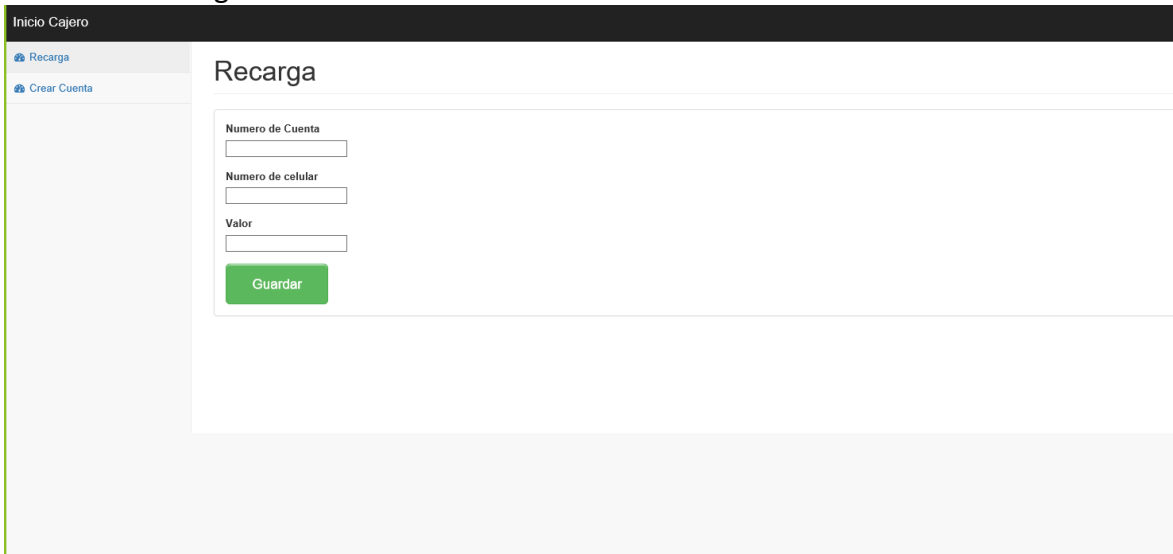
    @Inject
    private GestionSriLocal on;

    @POST
    @Path("/transferenciaSri")
    @Produces("application/json")
    @Consumes("application/json")
    public Respuesta transferenciaSri() {
        return on.aceptarRechazarFactura();
    }


}
```

Servicio de solicitud de recarga.

La interfaz web para realizar la recarga pide como parámetros número de cuenta, número de teléfono y valor de la recarga.



Para realizar la solicitud de recarga se crea los siguientes métodos donde el primero llama al servicio de la operadora donde rechazara o aprobara la recarga.

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

```

public class OperadoraCli {

    private String WS = "http://localhost:8083/JarroXavierOperadoraJEE/operadora/bo/transferenciaOperadora";

    public String operadora() {
        Client client = ClientBuilder.newClient();
        WebTarget target = client.target(WS);
        String respuesta = target.request().post(Entity.json(null), String.class);
        return respuesta;
    }

}

```

Como segundo método para realizar la recarga se tiene el siguiente método donde se llama al sri y se obtiene la respuesta si rechaza o aprueba la factura.

```

public class SriCli {

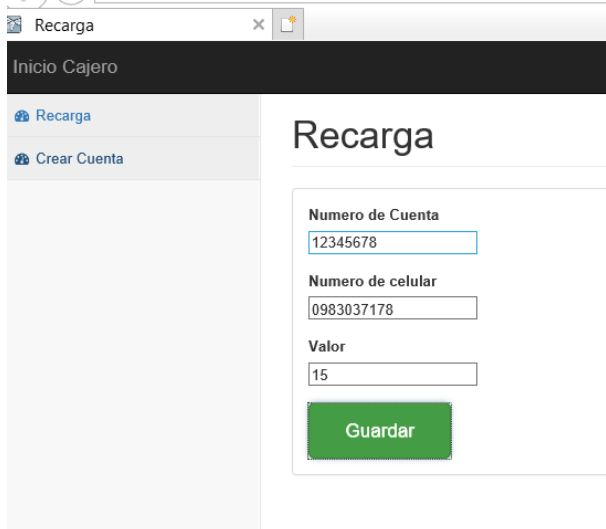
    private String WS = "http://localhost:8083/JarroXavierSRIJEE/sri/bs/transferenciaSri";

    public String sri() {
        Client client = ClientBuilder.newClient();
        WebTarget target = client.target(WS);
        String respuesta = target.request().post(Entity.json(null), String.class);
        return respuesta;
    }


}

```

Se realiza la recarga y se guarda en la base de datos.



Como se ve en la consola se realiza la recarga

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

```

WildFly Application Server × Run (Recarga) ×
11:39:18,430 INFO [stdout] (default task-21) Hibernate:
11:39:18,430 INFO [stdout] (default task-21)      select
11:39:18,430 INFO [stdout] (default task-21)          recarga0_.codigo as codigo1_2_,
11:39:18,430 INFO [stdout] (default task-21)          recarga0_.numero_cuenta as numero_c5_2_,
11:39:18,430 INFO [stdout] (default task-21)          recarga0_.estado as estado2_2_,
11:39:18,430 INFO [stdout] (default task-21)          recarga0_.numerCelular as numerCel3_2_,
11:39:18,430 INFO [stdout] (default task-21)          recarga0_.valorRecarga as valorRec4_2_
11:39:18,430 INFO [stdout] (default task-21)      from
11:39:18,430 INFO [stdout] (default task-21)      Recarga recarga0_

```

Y también se puede ver en la base de datos.

	codigo [PK] integer	numercelular character varying (255)	valorrecarga double precision	numero_cuenta character varying (255)
1	0	0983037178	150	12345678

URL GIT HUB.

<https://github.com/XavierJarro/JarroXavierExamenFinal.git>

Nombre del estudiante: Xavier Jarro

Firma del estudiante: 