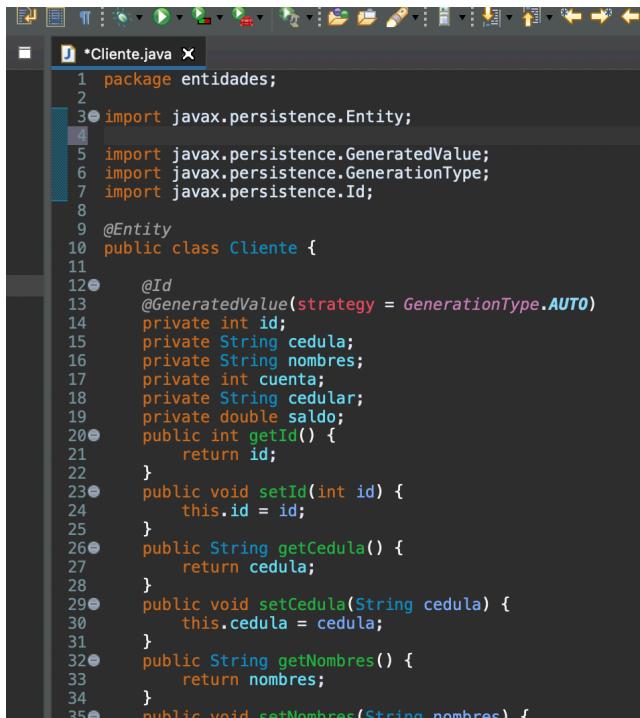
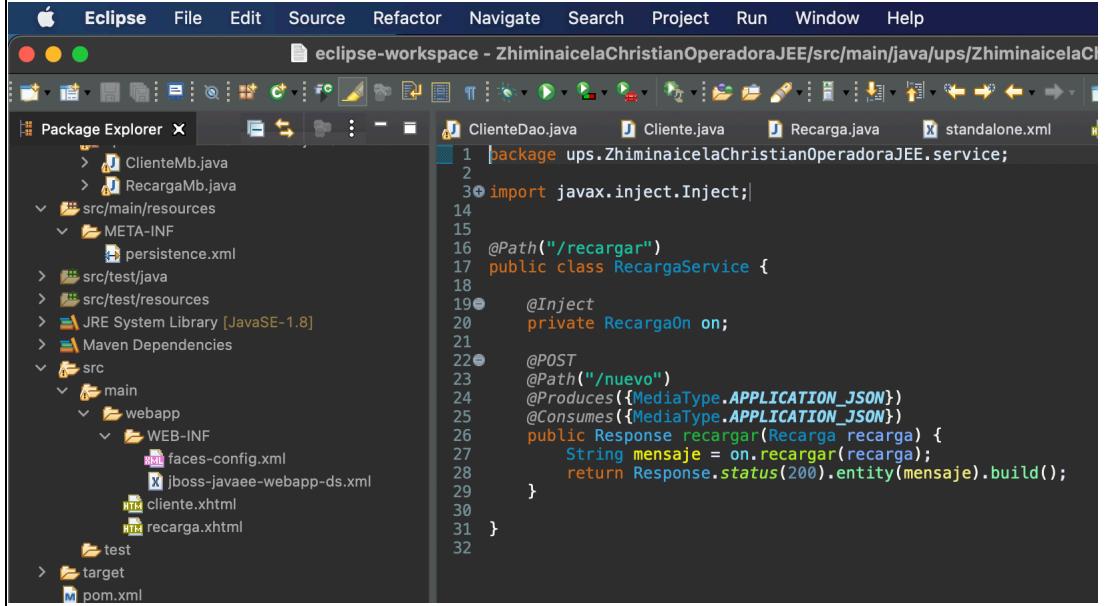


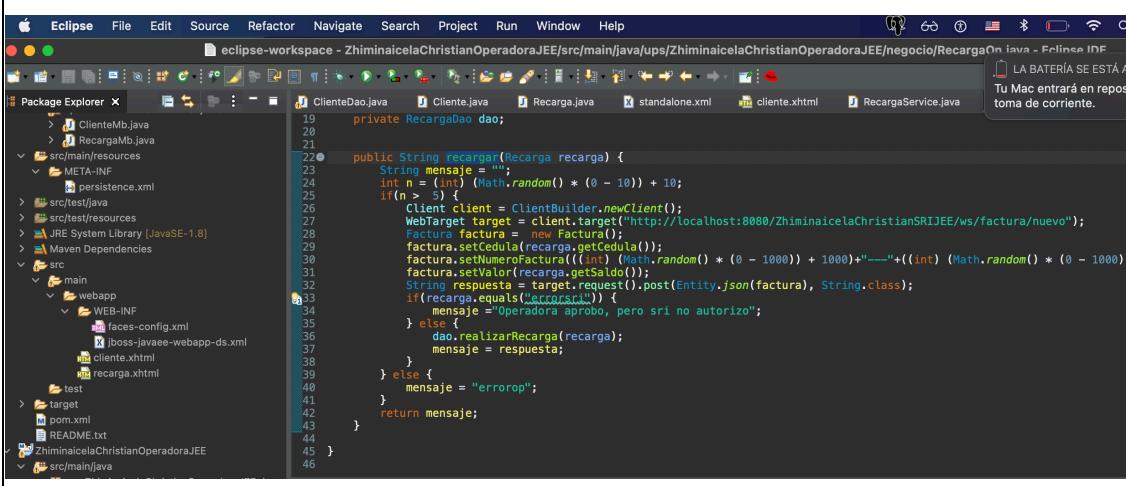
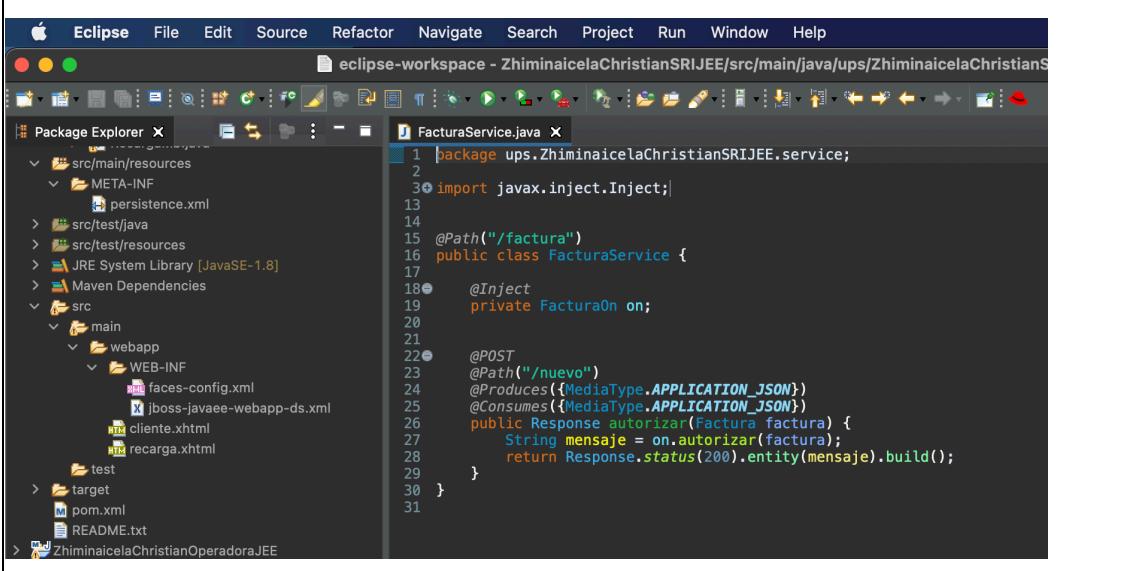
Examende App Distribuidas

1. Rubrica de calificación:

Parametro Descripción/detalles Valoración

Arquitectura JEE	Creación de proyectos bajo la arquitectura JEE , incluye creación correcta de las entidades
	<ul style="list-style-type: none">- Para la creacion de Iso proyectos usamos “proyectos maven” de jakarta.- Entidades: 

Interfaz JSF para crear cuentas	<p>1 > Se crean las cuentas directamente en la base de datos 2 > Se crean las cuentas desde web y se informa que el registro fue satisfactorio</p>
Servicio de recarga en operadora	<p>1 > WS Declarado correctamente en relación a parametros. 2 > WS declarado correctamente junto con lógica para rechazo o aceptación 3 > WS declarado correctamente junto con lógica y con almacenamiento de recarga en BD 4 > Respuesta devuelta de acuerdo a lo solicitado y es también receptada por Cliente</p>
	 <pre> package ups.ZhiminaicelaChristianOperadoraJEE.service; import javax.inject.Inject; @Path("/recargar") public class RecargaService { @Inject private RecargaOn on; @POST @Path("/nuevo") @Produces({MediaType.APPLICATION_JSON}) @Consumes({MediaType.APPLICATION_JSON}) public Response recarga(Recarga recarga) { String mensaje = on.recargar(recarga); return Response.status(200).entity(mensaje).build(); } } </pre>

	 <pre> 19 private RecargaDao dao; 20 21 public String recargar(Recarga recarga) { 22 String mensaje = ""; 23 int n = (int) (Math.random() * (0 - 10)) + 10; 24 if(n > 5) { 25 Client client = ClientBuilder.newClient(); 26 WebTarget target = client.target("http://localhost:8080/ZhiminaicelaChristianSRIJEE/ws/factura/nuevo"); 27 Factura factura = new Factura(); 28 factura.setCedula(recarga.getCedula()); 29 factura.setNumeroFactura((int) (Math.random() * (0 - 1000)) + 1000)+"-"+((int) (Math.random() * (0 - 1000)) + 30 factura.setValor(recarga.getSaldo()); 31 String respuesta = target.request().post(Entity.json(factura), String.class); 32 if(recarga.equals("errorrop")) { 33 mensaje = "Operador aprobó, pero SRI no autorizó"; 34 } else { 35 dao.realizarRecarga(recarga); 36 mensaje = respuesta; 37 } 38 } else { 39 mensaje = "errorrop"; 40 } 41 } 42 return mensaje; 43 } 44 45 } 46 </pre>
Servicio de autorización de factura en SRI	<p>1 > WS declarado correctamente en relación a parámetros.</p> <p>2 > WS declarado correctamente junto con lógica de rechazo o aceptación y almacenada en BD</p> <p>3 > Respuesta devuelta de acuerdo a lo solicitado y es también receptada por Cliente</p>
	 <pre> 1 package ups.ZhiminaicelaChristianSRIJEE.service; 2 3 import javax.inject.Inject; 4 5 @Path("/factura") 6 public class FacturaService { 7 8 @Inject 9 private FacturaOn on; 10 11 12 @POST 13 @Path("/nuevo") 14 @Produces({MediaType.APPLICATION_JSON}) 15 @Consumes({MediaType.APPLICATION_JSON}) 16 public Response autorizar(Factura factura) { 17 String mensaje = on.autorizar(factura); 18 return Response.status(200).entity(mensaje).build(); 19 } 20 21 } 22 23 24 25 26 27 28 29 30 31 </pre>

	<pre> 1 package ups.ZhiminaicelaChristianSRIJEE.negocio; 2 3 import java.util.UUID; 4 5 6 @Stateless 7 public class FacturaOn { 8 9 @Inject 10 private FacturaDao dao; 11 12 public String autorizar(Factura factura) { 13 String mensaje = ""; 14 int n = (int) (Math.random() * (0 - 10)) + 10; 15 16 if(n > 5) { 17 dao.guardarFactura(factura); 18 UUID uuid = UUID.randomUUID(); 19 mensaje = uuid.toString(); 20 } else { 21 mensaje = "errorsri"; 22 } 23 return mensaje; 24 } 25 26 } 27 28 29 30 31 </pre>
Servicio de solicitud de recarga	<p>1 > interfaz web (JSF correcto) para solicitud de servicio 2 > Solicitud (cliente WS) para solicitud de servicios 3 > Interfaz + solicitud + generación debito en la cuenta de cliente 5 > Respuesta a cliente de acuerdo a lo solicitado</p>
	<pre> 29 public List<Cliente> listarClientes() { 30 return dao.listarClientes(); 31 } 32 33 34 public String buscarCliente(String cedula, double saldo) { 35 Cliente cliente = dao.buscarCliente(cedula); 36 double saldoActual = 0; 37 if (cliente != null) { 38 Client client = ClientBuilder.newClient(); 39 WebTarget target = client.target("http://localhost:8080/ZhiminaicelaChristianOperadoraJEE/ws/recargar/nuevo"); 40 Recarga recarga = new Recarga(); 41 recarga.setNumero(cliente.getcedula()); 42 recarga.setCuenta(cliente.getcuenta()); 43 recarga.setSaldo(saldo); 44 recarga.setSaldo(saldo); 45 saldoActual = cliente.getSaldo(); 46 String respuesta = target.request().post(Entity.json(recarga), String.class); 47 if (respuesta.equals("Error")) { 48 respuesta = "Error en la operadora"; 49 } else if (respuesta.equals("Error")) { 50 respuesta = "Error en la autorizacion del SRI"; 51 } else { 52 cliente.setSaldo(saldoActual-saldo); 53 dao.actualizar(cliente); 54 return "Recargar exitosa, verifica tu saldo, numero de autorizacion del SRI: "+respuesta; 55 } 56 } 57 return "no existe el cliente"; 58 } 59 60 61 62 63 64 </pre>