

Electiva 5: Programación Web - Backend

Trabajo Práctico: Primer Parcial.

Profesores:

Lic. José Rodrigo Benítez Paredes.

Ing. Gustavo Sosa Cataldo

Este trabajo involucra la implementación del Backend de un sistema informático que será especificado para cada grupo. El proyecto desarrollado se validará con peticiones GET, POST, PUT y DELETE que deberán elaborar y facilitar para el día de la revisión de manera a validar el funcionamiento. Como complemento también pueden tener preparadas consultas (select) SQL como apoyo a las peticiones GET para verificar el estado de los datos.

Revisiones y entrega

- Entrega del TP:
 - Viernes 01/04/2022(a la vuelta del primer parcial)

Grupos de alumnos

Los grupos serán individuales o de 3 alumnos como máximo, en donde uno de los cuales estará identificado como el Líder del Proyecto (Project Leader), quien tendrá la obligación de tratar todos los temas concernientes a este trabajo con el cliente (profesor).

Observaciones

- 1) Para la implementación puede utilizar el IDE de su elección (Eclipse, Netbeans, IntelliJ IDEA).
- 2) Como servidor de aplicación deben utilizar Wildfly (versión elección).
- 3) La base de datos debe ser Postgres (versión 9.x para facilitar la compatibilidad).
- 4) El gestor de proyecto DEBE ser MAVEN.
- 5) Tiene que cumplir el stack tecnológico JEE:
 - a. Modelo: JPA (con Hibernate, o algún otro ORM que sea una implementación de JPA)
 - b. Capa de negocios: EJB3
 - c. Capa de exposición: JAX-RS (Restful)
- 6) La estructura del proyecto puede ser una sola (como lo vimos en laboratorio) o pueden investigar más sobre MAVEN y crear diferentes submódulos (proyecto padre, sub proyecto de modelo, sub proyecto de capa de negocios, sub proyecto de capa de exposición)
- 7) Para el día de la entrega y defensa del TP deben estar presentes TODOS los integrantes del grupo, y el que no esté presente lleva ausente.
- 8) Para los procesos que deben ejecutarse planificadamente cada cierto tiempo, pueden investigar acerca de los session bean @Singleton y de los timer en ejb3 (@Schedule).

Enunciado 1: Sistema de fidelización de clientes

Se requiere la implementación de un sistema de fidelización de clientes que pueda hacer seguimiento de los puntos otorgados por operaciones de pago.

Los módulos a desarrollar son los siguientes:

1) Administración de datos del cliente (POST, GET, PUT, DELETE)

Este módulo contempla la administración de datos del cliente, los cuales serán los que acumulen puntos de fidelidad con sus operaciones.

Los datos de clientes a almacenar serán los siguientes: nombre, apellido, número de documento, tipo de documento, nacionalidad, email, teléfono, fecha de nacimiento.

Estructura: id autogenerado, nombre, apellido, número de documento, tipo de documento, nacionalidad, email, teléfono, fecha de nacimiento.

2) Administración de conceptos de uso de puntos (POST, GET, PUT, DELETE)

Este módulo contempla la administración de los diferentes conceptos que especifican a qué fueron destinados los puntos utilizados, con su respectiva cantidad de puntos requerida. Por ejemplo: vale de premio, vale de descuento, vale de consumición, etc.

Estructura: id autogenerado, descripción de concepto, puntos requeridos.

3) Administración de reglas de asignación de puntos (POST, GET, PUT, DELETE)

Este módulo permite definir las reglas que rigen la cantidad de puntos a asignar a un cliente en base al rango de valor de consumo:

- Ejemplo:
 - o 0 a 199.999 Gs.: 1 punto cada 50.000
 - o 200.000 Gs. a 499.999 Gs. 1 punto cada 30.000
 - o 500.000 Gs. para arriba: 1 punto cada 20.000

Observación: los rangos serán opcionales, es decir, se puede tener una sola regla que asigne 1 punto cada X Gs. sin importar en qué rango cae el monto de la operación.

Estructura: id autogenerado, límite inferior, límite superior, monto de equivalencia de 1 punto

4) Parametrización de vencimientos de puntos (POST, GET, PUT, DELETE)

Este módulo permite definir el tiempo de validez de los puntajes asignados a los clientes. Una vez alcanzado el tiempo determinado, los puntos son descontados de la bolsa por vencimiento.

Estructura: id autogenerado, fecha de inicio de validez, fecha fin de validez, días de duración del puntaje.

5) Bolsa de puntos

Estructura: id autogenerado, identificador del cliente, fecha de asignación de puntaje, fecha de caducidad de puntaje, puntaje asignado, puntaje utilizado, saldo de puntos, monto de la operación

6) Uso de puntos

Debe utilizarse en un esquema FIFO (primero se utilizan las bolsas más antiguas). Tiene un detalle debido a que para satisfacer una petición de puntos se puede utilizar más de una bolsa.

Estructura:

- **Cabecera:** id autogenerado, identificador del cliente, puntaje utilizado, fecha, concepto de uso de punto
- **Detalle:** id autogenerado, identificador de la cabecera, puntaje utilizado, identificador de la bolsa de puntos utilizada

7) Consultas (GET)

Este módulo contempla la consulta para el desarrollo de reportes.
Las consultas a proveer son:

- uso de puntos por: concepto de uso, fecha de uso, cliente
- bolsa de puntos por: cliente, rango de puntos
- clientes con puntos a vencer en x días
- consulta de clientes por: nombre (aproximación), apellido (aproximación), cumpleaños

8) Servicios

- **carga de puntos (POST):** se recibe el identificador de cliente y el monto de la operación, y se asigna los puntos (genera datos con la estructura del punto 5)
- **utilizar puntos (POST):** se recibe el identificador del cliente y el identificador del concepto de uso y se descuenta dicho puntaje al cliente registrando el uso de puntos (genera datos con la estructura del punto 6 y actualiza la del punto 5)
 - o además debe enviar un correo electrónico al cliente como comprobante
- **consultar cuantos puntos equivale a un monto X (GET):** es un servicio informativo que devuelve la cantidad de puntos equivalente al monto proporcionado como parámetro utilizando la configuración del punto 3

9) Proceso planificado cada x horas

Proceso que pueda planificarse que corra cada X horas y actualice el estado de las bolsas con puntos vencidos.