



Challenge FullStack Backend

Introducción

Recibe un cordial saludo de quienes formamos el equipo de ELDAR, por ese motivo queremos conocer un poco sobre tus habilidades técnicas en el desarrollo de diversos escenarios.

En cada escenario el objetivo es que demuestres de qué manera podrías resolverlo, usando las herramientas que consideres necesarias.

Si algún tópico/escenario no se entiende, nos podés contactar mediante mail a las casillas:

luciano.sosa@eldars.com.ar

mariano.avancini@eldars.com.ar

Si algún tópico/escenario está fuera de tu conocimiento está bien.

Desde ELDAR creemos en el trabajo en equipo, en el acompañamiento y mentoring de las personas, es por eso que te estamos buscando.

Creemos en vos y sabemos que vas a dar lo mejor.

¡Te deseamos éxitos!

Requerimiento

Se ha solicitado el diseño de un sistema para procesar operaciones con tarjetas de crédito para una organización. Dicho sistema debe disponer de un módulo que lo permita con las siguientes consideraciones, desarrollar un aplicativo:

- Una tarjeta se identifica de acuerdo a la marca, número de tarjeta, cardholder (nombre, apellido, dni, fecha de nacimiento y email) y fecha de vencimiento. Además, se asocia un código de seguridad (CVV) privado de 3 dígitos, el cual va a necesitar ingresar el usuario a la hora de operar con la tarjeta.
- Una operación es válida en el sistema si la persona que opera en el mismo consume menos de 10.000 pesos.
- Una tarjeta es válida para operar si su fecha de vencimiento es mayor al presente día.

NOTA: Tener en cuenta que el CVV es un dato sensible.

Hoy en día, existen tres marcas de tarjeta de crédito, a saber: "VISA", "NARA", "AMEX" y es posible que en los siguientes meses existan nuevas marcas. Cada marca tiene un modo de calcular una tasa por el servicio que es desde 0.3% hasta 5%, a saber:

- Tasa VISA = año / mes (Ejemplo: 20/12)
- Tasa NARA = día del mes * 0.5 (Ejemplo: 27 * 0.5)
- Tasa AMEX = mes * 0.1 (Ejemplo: 9 * 0.1)



BACKEND

verde = hecho | rojo = incompleto

Ejercicio 1

Desarrollar un menú interactivo por consola que permita a un usuario operar con las siguientes funciones:

- Registrar una persona indicando nombre, apellido, dni, fecha de nacimiento y email
- Registrar una tarjeta indicando marca, número, fecha de vencimiento y nombre completo del titular
- Retornar información de las tarjetas asociadas de un usuario por dni
- Consultar las tasas de todas las marcas por fecha (Si no se envía una fecha, se toma la fecha actual, formato dd-MM-yyyy) --> No llegue por cuestiones de tiempo pero sería implementar otro endpoint en CreditCardController.
Notar que para el cálculo de la tasa de las TC use patro Strategy

IMPORTANTE: Realizar el correspondiente manejo de excepciones en casos de error.

Ejercicio 2

Implementar una API REST considerando las siguientes funcionalidades:

- Alta de Usuarios (Baja y modificación opcional).
- Alta de Tarjetas (Baja y modificación opcional). Al crear la tarjeta los datos sensibles como el CVV y el PAN deben enviarse al email del usuario/tarjetahabiente.

Nota--> No llegue por cuestiones de tiempo pero cvv pudo haberse "hasheado"



- Consultar la tasa de una operación, informando marca de la tarjeta e importe.
- Realizar una compra indicando monto, detalle, tarjeta asociada y CVV. Una vez finalizada la compra exitosamente se debe notificar al usuario sobre la misma a través de un email.

Hostear solución en algún cloud computing libre e indicar URL.

EXTRA: Desarrollar dos tests unitarios (JUnit- Mockito). No llegue.

IMPORTANTE: Compartir código en GIT con la solución de ambos ejercicios.

Ejercicio 3

Analizar la siguiente situación y responder:

Tu equipo de trabajo está desarrollando un sistema de monitoreo de cámaras de seguridad. El sistema debe mostrar la imagen de hasta 10 cámaras. Cada cámara se muestra en un marco/monitor. Se requiere disminuir al mínimo el ancho de banda utilizado, de tal manera que, si la cámara no modifica su imagen, el sistema siga mostrando la misma captura hasta que haya un cambio en la imagen que la cámara toma.

¿Qué patrón de diseño elegirías para implementar este problema en una solución de software? ¿Por qué?

Ejercicio 4



National Bank tiene una política sencilla para advertir a los clientes sobre posibles actividades fraudulentas en sus cuentas. Si el monto gastado por un cliente en un día en particular es mayor o igual que $2 \times$ el gasto medio del cliente durante un número de días anteriores, le envían una notificación sobre un posible fraude. El banco no envía ninguna notificación al cliente hasta que tenga al menos ese número de días anteriores de datos de transacciones.

Dada d cantidad de días anteriores y los gastos diarios totales de un cliente durante un período de n días, determine la cantidad de veces que el cliente recibirá una notificación durante todos los n días.

Ejemplo

$gastos = [10, 20, 30, 40, 50]$

$d = 3$

En los primeros tres días, solo recopilan datos de gastos. En el día 4, los gastos anteriores son $[10, 20, 30]$. La mediana es 20 y el gasto del día es 40. Ya que $40 \geq 2 \times 20$, habrá un aviso. Al día siguiente, los gastos anteriores son $[20, 30, 40]$ y los gastos son 50. Esto es menor que 2×30 , por lo que no se enviará ningún aviso. Durante el período, se envió un aviso.

Nota: La mediana de una lista de números se puede encontrar ordenando primero los números en orden ascendente. Si hay un número impar de valores, se elige el del medio. Si hay un número par de valores, la mediana se define como el promedio de los dos valores del medio.



FRONTEND

-->Incompleto todo lo de front, hace mucho que no toco front me tomaria mucho tiempo.

Ejercicio 1

Teniendo en cuenta el punto dos del backend, desarrollar un frontend con las siguientes funcionalidades:

Funcionalidades del Frontend

Gestión de Usuarios

- **Alta de usuarios:**

Formulario para registrar un usuario con los siguientes campos:

- Nombre
- Apellido
- DNI
- Fecha de nacimiento
- Email

Gestión de Tarjetas

- **Alta de tarjetas:**

Formulario para asociar una tarjeta a un usuario, con los siguientes campos:

- Marca de la tarjeta (VISA, NARA, AMEX).
- Número de tarjeta.
- Fecha de vencimiento.
- Nombre completo del titular.
- Validar que los datos sensibles (CVV y PAN) se envíen por email al usuario.

- **Consulta de tarjetas asociadas:**

Listar todas las tarjetas registradas para un usuario específico, ingresando su DNI.

Consulta de Tasas

Mostrar las tasas calculadas para todas las marcas de tarjetas, según:

- Fecha ingresada manualmente.
- Fecha actual (si no se ingresa una fecha).

Opcional: Incluir una representación visual (tabla o gráfica) para mostrar las tasas.

Realización de Compras

Simular una compra con los siguientes pasos:

1. Ingresar:
 - Monto de la compra.
 - Detalle de la compra.
 - Tarjeta seleccionada de una lista desplegable (basada en las tarjetas registradas para el usuario).
 - Código CVV de la tarjeta.
 2. Confirmar la compra con un mensaje visual (ejemplo: modal o alerta).
 3. Enviar una notificación por email al usuario tras una compra exitosa (si se implementa esta funcionalidad).
-

Requisitos No Funcionales

- **Crear notificaciones dinámicas:**
 - Si hay muchas notificaciones, deben agruparse.
 - Usar las siguientes categorías: **informativas**, **advertencia**, y **críticas**.
- **Acceso por SSO:** Implementar Single Sign-On (preferentemente con Google).
- **Manejo de errores.**
- **Agregar las validaciones necesarias.**

Notas: Puede incluir otras consideraciones no funcionales que considere relevantes, describirlas en un archivo README.