FACTOR IT

MOVE FORWARD

Examen Técnico: Back-End

Bienvenido!!!
Si estás leyendo este texto es que estás por un muy buen camino para formar parte de nuestro equipo de desarrollo.

A continuación te plantearemos un desafío para que puedas mostrarnos todo lo que fuiste aprendiendo en tu carrera como desarrollador Back-End.

En cuanto al tiempo para realizarlo, tendrás 1 semana desde el momento en que lo recibas. En caso de que surjan dudas o algún imponderable para realizarlo, no dudes en avisarnos!

Apenas lo recibamos, trataremos de darte feedback lo antes posible, siempre tratamos de que no se extienda más allá de 1 semana.

DESAFÍO

En los últimos tiempos las plataformas de eCommerce han tenido que ir evolucionando para brindar a sus clientes una mejor experiencia de usuario como así también ser más escalables.

Como consecuencia de la pandemia, estos cambios han sido más vertiginosos, por lo que es fundamental que todo proyecto tenga un enfoque ágil y orientado al usuario.

Es por eso que en este desafío vamos a pedirte que armes parte del back-end para un sitio de eCommerce, en donde estará compuesto por una serie de servicios (API RestFull) para brindar funcionalidad al front.

REQUERIMIENTOS

Los servicios a desarrollar estarán basados en 2 grandes puntos del mundo eCommerce, por un lado las compras y por el otro el Checkout (gestión del Carrito de compras).

- **Compras realizadas (GET):** El servicio deberá brindar el detalle de las compras realizadas por un usuario en particular (identificado por dni).
  - **Filtros:** se podrá filtrar por un período (From-To). Si se le envía solo Fecha-From, el servicio deberá devolver todas las compras a partir de la fecha indicada.
  - **Ordenamiento:** El servicio podrá ser solicitado según 2 tipos de orden: fechas y montos.

- **Gestión del Carrito:** Se deberán contemplar las acciones para poder llevar adelante el uso de un carrito de compras, como ser:
  - Crear y eliminar un carrito.
  - Agregar y eliminar productos de un carrito.
  - Consultar el estado de un carrito. Esta acción devolverá el total de productos que contiene.
  - Finalización de un carrito por compra. Esta acción cerrará el carrito, dando el valor final del mismo (con promociones aplicadas si correspondiesen).

REGLAS DE NEGOCIO

Existen dos tipos de carritos, **común y especial**. Este hecho se determinará con un flag sobre el servicio de creación del carrito (isSpecial:true) además de acompañarse la fecha de creación del mismo.

El cliente puede realizar varias compras en el mismo día.

No es necesario desarrollar ningún tipo de ABM de los productos, ni de clientes. Los productos enviados al agregarse al carrito se tomarán como "válidos" y el precio indicado será el que se tome como válido (además no se tiene en cuenta stocks de los mismos, así que no hace falta que realices control de este punto).

Los productos deberán tener precios mayores a $100.

Para calcular el valor del carrito se debe tener en cuenta:

- ✓ Si se compran más de 3 productos se realizará un descuento de $100 para carritos comunes y de $150 para carritos especiales.
- ✓ Si el cliente en un determinado mes calendario, realizó compras por más de $5.000, pasa a ser considerado VIP y por lo tanto, en su próxima compra, se le aplicará un "descuento especial" de $500 pesos, solo si la compra supera los $2000 (acumulable con el descuento del punto 1).
- ✓ Si el cliente "compra" 4 productos iguales (quantity:4), el sistema aplicará un descuento extra, realizando la promoción 4x3 en dicho producto.

Observación: Sería bueno que en la determinación de si un cliente es vip o no, se consuma el servicio de "compras realizadas" para poder iterar y determinar dicha situación (obteniendo el "mes" para el cálculo sobre la fecha de finalización del carrito).

CRITERIOS DE EVALUACION

A la hora de evaluar las pruebas tendremos en cuenta:

- ✓ Cumplimiento del Objetivo.
- ✓ Pruebas unitarias y/o funcionales
- ✓ Buenas prácticas de desarrollo
- ✓ Adaptabilidad del código.
- ✓ Patrones de Diseño.

CONSIDERACIONES

- ➢ Dejamos a libre criterio las definiciones de estructuras a utilizar en cada servicio (Json), pero las mismas deben ser compartidas para poder realizar el consumo correspondiente.
- ➢ En el caso de no poder implementar alguna funcionalidad, no dejes de hacer lo que puedas! Si existe el servicio, pero no filtra, al menos lo podemos probar!
- ➢ Tecnologías: Java 7+ / Spring-SpringBoot / JPA-Hibernate.
- ➢ Exponer servicios REST (agregar proyecto postman. Usar Swagger/OpenAPI).
- ➢ Tests unitarios JUnit/Mockito.
- ➢ Scripts SQL.
- ➢ Se valorará la implementación de algún caso y/o método de testing unitario.
- ➢ Si no conoces algunas de las tecnologías indicadas, podrás reemplazarla indicando por cual y realizando la justificación del reemplazo.
- ➢ Deberás enviar el código mediante algún repositorio GIT.
- ➢ Se permiten agregados o extras a los requerimientos detallados que sumen al desafío propuesto en el tiempo solicitado. Todo esfuerzo se valorará extra!