

Seminario de Ciencias de la Computación A 2023-1

Desarrollo web back-end

Proyecto

Universidad Nacional Autónoma de México
Facultad de Ciencias - Ciencias de la Computación

Profesor: Jesús Iván Saavedra Martínez
Ayudante: Fernando Antonio Sánchez Montoya
Ayudante: Carlos López Rodríguez
Ayudante Lab: Carlos López Rodríguez
Ayudante Lab: Ilse Gisela Suárez Peña

7 de noviembre de 2022

Descripción general

Durante el curso se han desarrollado un par de APIs para gestionar la información de clientes y productos. Asimismo, se ha implementado la estructura de la arquitectura de microservicios. En este proyecto se implementarán funcionalidades faltantes de una API que administra la información de facturas de clientes cuando compran productos (Proyecto 1).

Adicionalmente, se dará una opción alternativa de proyecto para que los alumnos aprovechen el trabajo que están realizando en otras materias del actual semestre 2023-1 (Proyecto 2).

Desarrollo del Proyecto 1

Desarrolla la siguiente lista de requerimientos:

1. Crea un proyecto de Spring con lo siguiente:
 - Maven Project, Spring Boot 2.7.5*, Group: com, Artifact: invoice, Packaging: Jar, Java: 11
 - Agrega las dependencias necesarias de acuerdo con el requerimiento 2
 - Agrega a tu proyecto las clases del archivo "Proyecto.InvoiceModule.zip" en los paquetes correspondientes (Controller, Service, etc.)
2. Integra la API Product y la API generada en el requerimiento 1 a la arquitectura de microservicios desarrollada a partir de la semana 10 del curso, la cual tiene integrada la API Customer.
 - la configuración de los servicios debe ser gestionada por el config-service
 - deben ser registrados en el servidor de eureka
 - deben ser monitorizados por el admin-service
 - cualquier petición a las APIs debe realizarse por el gateway-service
3. Actualmente el método addToCart del Servicio Cart valida con la API Customer si el RFC del cliente es válido. Sin embargo, no tiene implementada la validación con la API Product si el GTIN del producto es válido y si se cuenta con el stock suficiente para agregar el producto al carrito.
 - actualiza el DtoProduct agregando los atributos de clase necesarios para las validaciones
 - actualiza el método getProduct de la interfaz ProductClient (en el paquete config.client) para que recupere la información necesaria de la API Product
 - actualiza el SvcCartImp para que valide que el GTIN del producto exista. Si existe, asignar el stock del producto a una variable llamada product_stock. Si el producto no existe enviar una excepción con el mensaje "product does not exist"

4. ¿Qué pasa si se desea agregar un producto que ya existe en el carrito? En lugar de registrarlo nuevamente, solo se debería actualizar la cantidad del producto en el carrito. Actualiza el SvcCartImp para que valide si el producto ya había sido agregado en el carrito
 - si ya había sido agregado, validar que las cantidades (quantity nueva + quantity ya agregada) no superen el stock disponible del producto
 - si la cantidad total no supera el stock entonces se debe actualizar la cantidad previamente registrada, terminar la ejecución del método y enviar el ApiResponse "quantity updated"
 - si la cantidad total supera el stock se debe enviar una excepción con el mensaje "invalid quantity"
 - si no había sido agregado, continuar con el flujo normal
5. Implementa el método generateInvoice de SvcInvoiceImp considerando lo siguiente:
 - el método recibe un RFC, con este se debe consultar el carrito de compras del cliente, si no tiene productos en el carrito se debe enviar un NOT FOUND con el mensaje "cart has no items"
 - por cada producto en el carrito, se debe generar un artículo de la factura, donde:
 - unit_price = precio del producto
 - total = precio del producto por la cantidad de productos comprados
 - taxes = 16% del total
 - subtotal = total menos impuestos
 - para la factura:
 - total = suma de los totales de todos los productos
 - taxes = suma de los impuestos de todos los productos
 - subtotal = suma de los subtotales de todos los productos
 - created_at = fecha y hora en la que se generó la factura
 - por cada artículo comprado, se debe actualizar el stock del producto utilizando el método updateProductStock de la interfaz ProductClient (paquete config.client).
 - vaciar el carrito de compras del cliente
 - toda la información de la factura y sus artículos deben quedar registrados en la base de datos

Desarrollo del Proyecto 2

Si en el actual semestre 2023-1, en una materia distinta, estás desarrollando un proyecto de desarrollo web back-end y/o front-end, puedes entregar ese proyecto en lugar de desarrollar los requerimientos del proyecto 1 definido en la sección anterior.

Para contar como valido tu proyecto se deben seguir los siguientes lineamientos:

- estar inscrito en la materia de la cual se pretende entregar el proyecto, no se tomarán en cuenta alumnos que estén como oyentes
- enviar un correo a más tardar el 20 de noviembre de 2022 con asunto "dwb2023-1 proyecto alternativo" al profesor y a Carlos López incluyendo lo siguiente:
 - nombre del/los alumno(s) que entregarán el proyecto en esta modalidad
 - nombre de la materia del proyecto
 - documento(s) o evidencia que describa los requerimientos a desarrollar en el proyecto
- apearse a la fecha de entrega del proyecto de Desarrollo web back-end 2023-1 a pesar de que en la otra materia se tenga programada una fecha posterior
- entregar el código desarrollado del proyecto de acuerdo con lo definido en la sección "Entrega", se puede llegar a solicitar al alumno información adicional para la revisión y calificación del proyecto.

Nota: el profesor y los ayudantes podrían llegar a consultar con las otras materias y profesores la veracidad de la información.

Entrega

La entrega del proyecto se debe realizar bajo los siguientes lineamientos:

- El proyecto 1 se puede realizar individual o en parejas. El proyecto 2 se puede entregar en equipos si así está definido en la otra materia.
- La entrega fecha de entrega del proyecto será el día 5 de diciembre de 2022.
- Cada requerimiento del proyecto 1 tiene un valor de 2 puntos.
- El código del proyecto desarrollado se subirá a un repositorio de git, el link se subirá a la sección correspondiente de classroom indicando si se entregará el proyecto 1 o 2 y si se trabajará en equipo o individual.
- Las dudas del proyecto se pueden realizar por slack mediante el canal "proyecto".