Taller JAVA 21 & SpringBoot 3

URL-Repo: <https://github.com/EYLatamSouth/java-21-onboarding>

**Pre-requisitos:**

* Utilizar JDK 21
* Utilizar SpringBoot 3
* Se puede utilizar los módulos que consideren necesarios de la Suit de SpringBoot.
* Utilizar Intlj IDEA
* Intentar utilizar GIthub Copilot
* Utilizar Docker.
* Unit Testing

**El objetivo primordial es utilizar, las nuevas feactures de java posterior a la versión 11(Selad Class, VirtualThreds, Records(Clases inmutables), Var etc.)**

**Aclaraciones:**

* Todo el proyecto esta construido sobre Docker-compose, por ende si necesita agregar algún otro componente como podría ser un Broker de mensajería, redis u otro microservicios estos deben agregarse al mismo para poder levantarse en conjunto y que Docker los administre y compartirán la misma intranet que gestiona Docker.
* La imagen base que se encuentra en el repositorio ya trae configurado, 3 bases de datos sql con mariadb, y 1 colección de mongodb con transacciones.
* Adicionalmente también trae un api en NodeJS que permite emular el servicio de veraz, wordlsys y renaper que se utilizaran para la matriz.
* Se contara con 3 microservicios:
  + **MS-Personas**: administrara la base de datos de personas la cual cuenta con las siguientes tablas: **domicilio estado\_usuario tipousuarios**
  + **MS-Cuentas**: administrara la base de datos de Cuentas la cual cuenta con las siguientes tablas: código\_moneda cuentas estado\_cuenta
    - Adicionalmente este mismo tendrá accesos a una colección de mongodb la cual le permitirá obtener las transacciones de Debito.
  + **MS-Tarjetas**: administrar a la base de datos de Tarjetas la cual cuenta con las siguientes tablas: estado\_tarjeta tarjeta y rumen\_emitidos
    - Adicionalmente este mismo tendrá accesos a una colección de mongodb la cual le permitirá obtener las transacciones de crédito.

**Objetivo de Taller:**

Se pretende emular el Onboarding de un banco digital, donde el alta estará conformada por varios pasos, llamadas a servicios validación y construcción de una matriz de productos entre otras.

**El alta del nuevo cliente contara con tres pasos:**

1. Validación de datos internos, donde se validara si la persona ya es cliente, si esta dado de baja o se su cuenta está bloqueada.
2. Validación con servicios de terceros y matriz de productos.
3. Alta de usuario y productos.

**Paso 1:**

* En caso de que el cliente ya este dado de alta no permitir el alta.
* En caso de que el cliente este dado de baja, volver a generar un alta nueva
* En caso de que el cliente este bloqueado indicarlo en la respuesta de error.

**Paso 2:**

* Cruzar la información en la matriz y esto devolverá los productos a dar de alta al cliente.

**Matriz:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Productos | Renaper | WorldSys | Veraz | Sueldo Bruto |
| Cuenta PESOS $ | FALSE | - | - | =>0 |
| Cuenta Dólar y pesos | TRUE | FALSE | =< 1.5 | =>200.000,00 |
| Cuenta Dólar y pesos+ Tarjeta basic | TRUE | FALSE | =< 1.0 | =>446.000,00 |
| Cuenta Dólar y pesos+ Tarjeta Gold | TRUE | FALSE | =< 0.5 | =>827.000,00 |
| Cuenta Dólar y pesos+ Tarjeta Black | TRUE | FALSE | =< 0.1 | =>999.000,00 |

**Paso 3:**

* Dar de alta de forma asincrónica cada producto. No se permiten llamadas http bloqueantes entre los 3 micro servicios (intentemos aplicar eventos)

Implemente los patrones de diseño que crea necesarios.

**IMPORTANTE**: Seria ideal que puedan intentar lograr implementar mantener el estado de la aplicación si la misma se reinicia (opcional.)

Entregable, crear un nuevo Branch en el repositorio publico y con el código final incluir un Excel con el detalle de cada evento que emite y escucha cada componente.