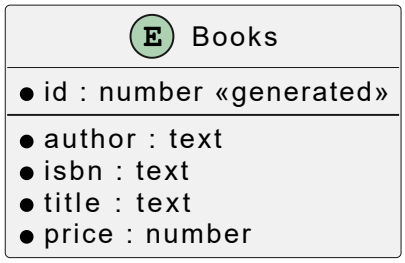
Trabajo 01

Fecha de entrega: 2023-01-31, 07:00 horas

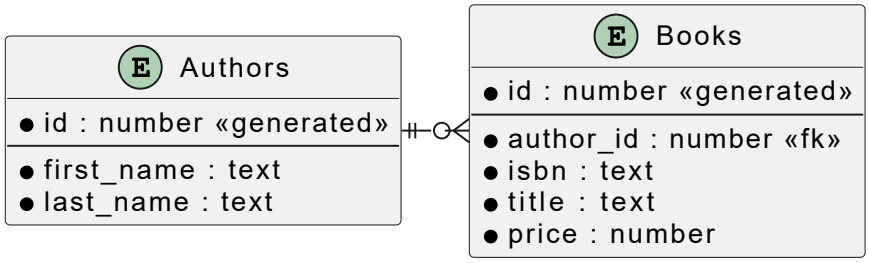
Realizar una aplicación que permita realizar la administración de la información asociada a libros (books) y autores (authors).

Base de datos

La base de datos consta de las tablas `Books` y `Authors`, a continuación se muestra el modelo ER:



Arquitectura de la aplicación



Descripción de los módulos

La aplicaciónm consta de los siguientes módulos

app-books

Este módulo permite realizar operaciones CRUD sobre la tabla `Books`. La tecnología a utilizar en este módulo corresponde a:

- Helidon 3.1
- Acceso a la base de datos a través de DBClient

app-authors

Este módulo permite realizar operaciones CRUD sobre la tabla `Authors`. La tecnología a utilizar en este módulo corresponde a:

- Qurkus 2.16

- Acceso a la base de datos a través de Panache-Quarkus Data Repository

app-web

Este módulo contiene la aplicación web accesible por el cliente. La tecnología a utilizar en este módulo corresponde a:

- SparkJava
- Páginas dinámicas mediante el uso de Thymeleaf

gateway

Este módulo realiza el balanceo de carga entre las diferentes instacias de los módulos. Se lo debe implementar utilizando Traefik con soporte para balanceo de carga de forma automática.

Despliegue de la aplicación

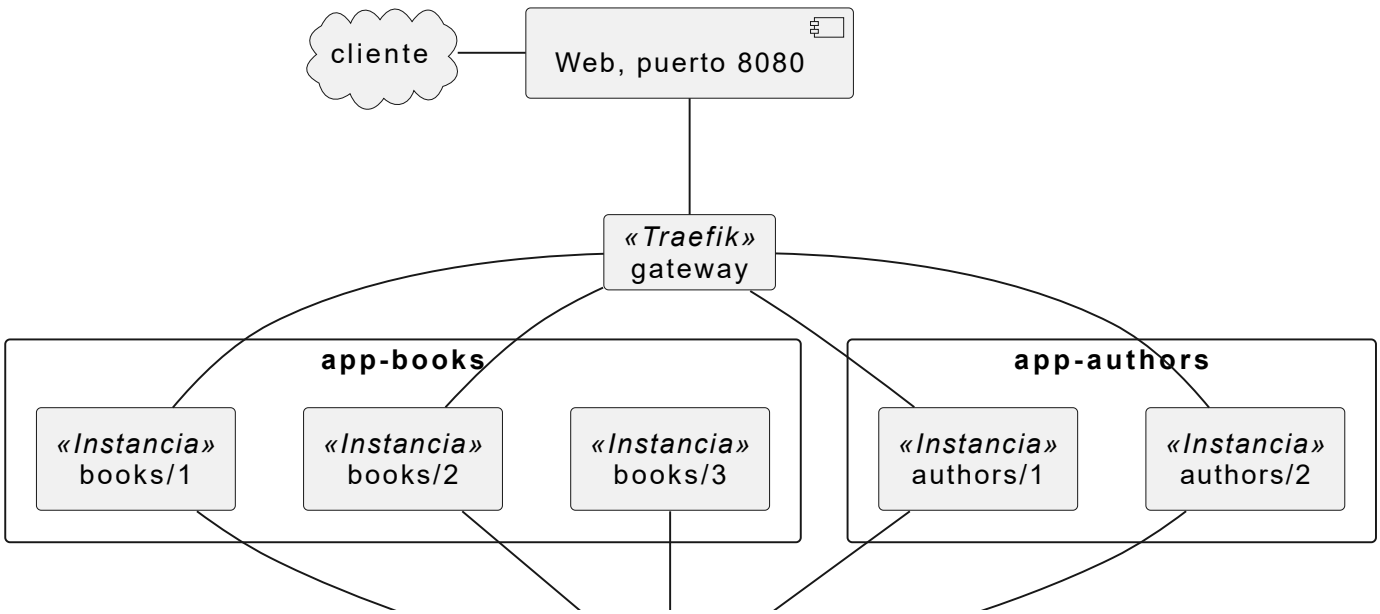
La aplicación debe ser desplegada utilizando insfraestructura Docker. Para cada aplicación se debe generar un contenedor docker.

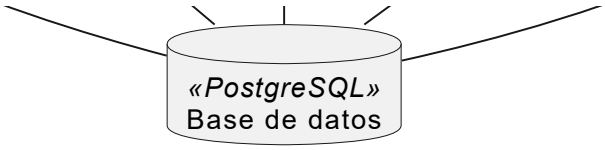
El despliegue debe contener 5 contenedores:

- app-books
- app-authors
- app-web
- Base de datos (postgresql)
- Gateway (traefik)

Para las móduoos *app-books*, *app-authors*, *app-web* se debe generar el respectivo archivo `Dockerfile` que permita generar la imagen de la aplicación.

La base de datos debe permitir actualizar el esquema y los datos en base a migraciones automáticas considerando que el esquema inicial correspomde a la siguiente tabla `Books`:





y a su vez, la tabla mencionada anteriormente contiene la siguiente información:

```
insert into books(isbn,title,author,price) values ('11-11','title1',
'author1',20);
insert into books(isbn,title,author,price) values ('22-22','title2',
'author2',20);
insert into books(isbn,title,author,price) values ('33-33','title3',
'author3',20);
insert into books(isbn,title,author,price) values ('44-44','title4',
'author4',20);
```

Para la aplicación completa, se debe generar el archivo de despliegue para `Docker Compose`

Entregables

Se debe registrar los siguientes archivos:

1. Archivo de despliegue `docker compose` que permita desplegar la aplicación completa

```
c:\>mi-app\docker-compose up
```

2. Archivo `compilar.bat` que contenga el script para:

- clonar el/los repositorios
- compilar el código fuente y generar los archivos JARs necesarios
- crear la imagen docker para cada aplicación
- registrar la imagen docker en el repositorio docker-bub
- desplegar la aplicación sobre docker

3. Documento PDF con la descripción de la configuración de cada uno de los módulos

NOTA: para el registro de la imagen en docker-hub utilziar la cuenta `jaimesalvador`, es decir, las imágenes contendran el nombre `jaimesalvador/imagen:version`.

Consideraciones generales

1. Todos los móulos deben utilizar *Gradle* como gestor de dependecias
2. Los componentes de negocio se deben implementar con CDI
3. Los componentes de acceso remoto se debe implementar mediante servicios REST utilizando JAX-RS
4. Aplicar todas las buenas prácticas mencionadas en la metodología de los *12-factores*

5. Luego de enviar la tarea **no se debe modificar** ningún repositorio de código fuente o imagen en docker-hub. En el caso que se detecte modificaciones luego de la fecha y hora de entrega, se invalidará el trabajo y recibirá la nota de 0 (cero).
   .