

# Universidad central Ecuador



## Programación Distribuida

Ian Hendry Paucar Montes

31/01/2023

### APP-AUTHORS

#### Realizado con Quarkus

- **Clase Author:** Define las variables, en este caso el identificador, primer nombre y apellido
- **Clase AuthorRepository:** Es la capa repository. Esta interfaz implementa métodos de consulta y de guardado.
- **Clase AuthorRest:** Aloja las diferentes funciones de buscar, eliminar, crear y actualizar.
- **Clase ImplemetacionServicioAutor:** Clase de transacciones conectada al repositorio. Comprende las anteriores funciones, pero implementadas al repositorio. Lógica del método.
- **V1, V2, V3 y V4 de SQL.** Archivos encargados de la migración, crea la base de datos, en este caso Books, inserta registros, después crea una tabla Author si es que no existe y por ultimo añade registros a la ultima tabla. Añade la Foreign Key.
- **Application.propieties:** Configuraciones de Quarkus y de la conexión a la base de datos. Detalla el puerto

### APP-BOOKS

#### Realizado con Helidon

- **DbConfig:** Conexión y configuración de la base
- **Clase Book:** Define las variables, en este caso el identificador, isbn, titulo, autor y precio
- **Clase BookRest:** Aloja las diferentes funciones de buscar, eliminar, crear y actualizar.
- **Clase ImplementaciónServicioLibros:** Clase de transacciones conectada al repositorio. Comprende las anteriores funciones, pero implementadas al repositorio. Lógica del método.
- **Servidor:** Inicia la app book con helidon.
- **Application.yaml:** Configuración de la URL de la base de datos, las credenciales y el puerto a usar

## APP-WEB

### Realizado con Spark

- **Settings:** Representa la URI específica que queremos invocar para acceder a un recurso REST particular. Usa a `BASE_URL=`<http://traefik>
- **WebServer:** Clase encargada de redirigir a las rutas correspondientes dado los strings entregados. Redirige a los diferentes servicios.
- **AuthService:** Tiene los métodos para gestionar Author, el CRUD
- **AuthServiceImpl:** Es la primera parte del proceso, donde la parte cliente hace una petición o request que llegará al servidor web, después, da solución que brinda el servidor web después de hacer lo que necesita para darle respuesta.
- **BookService:** Tiene los métodos para gestionar Author, el CRUD
- **BookServiceImpl:** Es la primera parte del proceso, donde la parte cliente hace una petición o request que llegará al servidor web, después, da solución que brinda el servidor web después de hacer lo que necesita para darle respuesta.
- **Clase Author:** Define las variables, en este caso el identificador, primer nombre y apellido
- **Clase Book:** Define las variables, en este caso el identificador, isbn, titulo, autor y precio
- **Application:** Provee un contenedor responsive y el servidor
- **Templates:** Contiene los archivos html encargados de la vista o pagina web para dar un frontend a la aplicación.

**Compile.bat:** Encargado de generar las imágenes a partir del directorio GitHub, lo realiza en el Docker.

**Docker-compose.yml:** Tiene las configuraciones necesarias como la conexión a la base de datos, los puertos a usar por Traefik, número de réplicas. Inicializa cada una de las aplicaciones: Author, Books y Web

**Comandos:** Conjunto de comandos necesarios en la ejecución local de la aplicación

**Link GitHub:** <https://github.com/Ian-YangMax/Trabajo01>