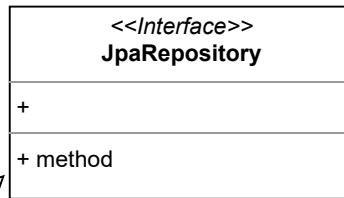
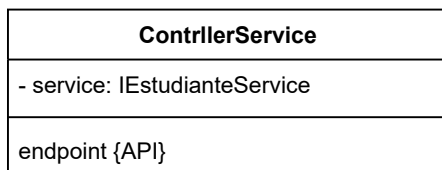
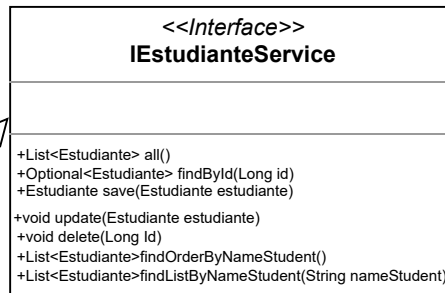
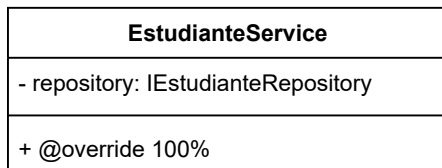
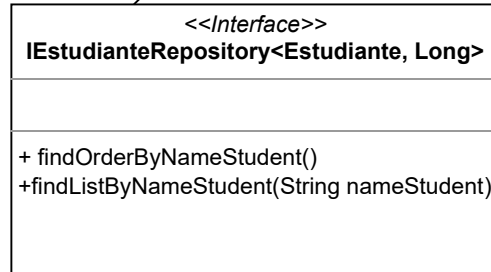
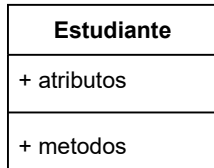


1. Entity
 2. Repository
 3. IService
 4. Service
 5. Controller
- Proyectado DTO



Extends



API: interfaz de programación de aplicaciones
ruta:
Protocolo: http/https
Server: ip - localhost
Port: 8080 8000 8081 9090 9000 9001 //Verificar la disponibilidad
EndPoint: nombre del controller -> tenga el nombre del entity
EndPoints Service: nombres especificos generados.

<http://localhost:9000/Estudiante>

A continuación, te menciono algunos puertos recomendados para usar en la construcción de una aplicación Spring Boot:

Puerto 8080: Es el puerto más comúnmente utilizado para aplicaciones web en Spring Boot. Si no se especifica ningún puerto en la configuración, Spring Boot utilizará el puerto 8080 de forma predeterminada.

Puerto 8000: También es un puerto popular para aplicaciones web y es comúnmente utilizado como alternativa al 8080.

Puerto 9090: Otro puerto comúnmente utilizado para aplicaciones web, especialmente si el 8080 ya está en uso por otra aplicación.

Puertos en el rango 9000-9999: En ocasiones, los desarrolladores eligen puertos en este rango si los puertos anteriores ya están ocupados o si desean organizar los puertos de sus aplicaciones de manera más estructurada.

API:

Una API responde a peticiones de los verbos HTTP

GET => Obtener o consultar

POST => Agregar nuevo registro

PUT => Actualizar registro

DELETE => Eliminar registro

STATUS DE LOS VERBOS HTTP:

Indicar que ha sucedido con la petición HTTP

100 => Informativos

200 => Ejecución con éxito

300 => Redireccionamiento

400 => Error de acceso por del cliente

500 => Error de servidor