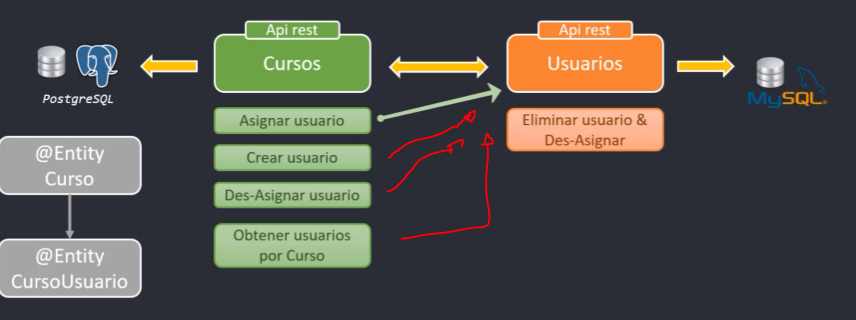
5.HTTP\_Feign\_Comunicación\_entre\_microservicios)

The Feign es un servicio web declarativo (cliente HTTP) desarrollado por **Netflix** . Su objetivo es simplificar los clientes API HTTP. Es un enlazador de cliente Java a HTTP. Si desea utilizar Fingir, cree una interfaz y anótela. Proporciona compatibilidad con anotaciones conectables, incluidas anotaciones de simulación y anotaciones JAX-RS.

Es una biblioteca para crear clientes API REST. Hace que los clientes de servicios web sean más fáciles. Los desarrolladores pueden usar anotaciones declarativas para llamar a los servicios REST en lugar de escribir un código repetitivo representativo.

Introducción conectando microservicios

Relacionando microservicios, por medio de peticiones REST, entonces se deben relación tanto curso coimo cursoUsuario



Creando JPA entity intermedia CursoUsuario

La idea es crear un identificador único para el userId, con el fin de que un usuario no se pueda repetir en un curso por ende el model CursoUsuario. Por buena practica agregamos la tabla CursoUsuario para mitigar el de oneToMany bidireccional y quede de uno a muchos y de muchos a uno

(OneToMany) – (ManyToOne)

Curso -> CursoUsuario -< Usuario

* Configurando el nuevo entity para cursoUsuario – msvc-curso
* Configurando Curso para que realice la relación de OneTMany

Añadiendo JoinColumn curso\_id como foreignkey y la clase POJO Usuario

Procedo a crear un pojo en en msvc-cursos de el servicio Uusuario para recibir el objeto con todos sus atributo, es similar a una clase entity en este caso sera antes des entity Model

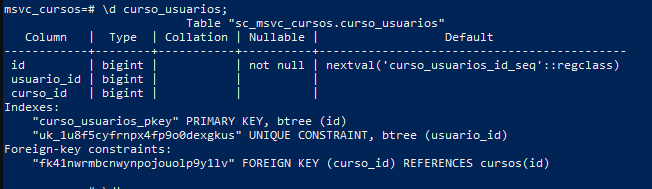
* Crear clase Usuario(Respsentara la estructura del Json) que contiene el entity Usuario de msvc.usuario
* Agrego el objeto list de tipo Usuario a la clase Usuario con la condición de que esta tipo de list no es persistente (Que no esta en la DB), se le distinge con la anotación @Transient
* Inicialiar el pojo en el constructor Curso

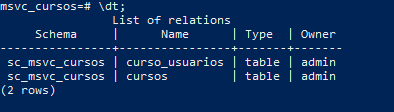
 Revisando tablas de la BBDD y agregando métodos de comunicación HTTP

Ya lista la estructura ejecutamos para validar como quedan las tablas por detrás con todas las configuraciones.

Valido en la DB que se hayan creado la tabla curso\_usuarios y las configuraciones

la tabla curso\_usuario en específico para ver toda la configuración que tiene \d curso\_usuarios;





* Agregando los métodos(Afectan al otro api rest pero no ajusta en DB) que traerá los datos del otro servicio en el CursoService del microservicio Curso

Escribiendo el Cliente HTTP con Spring Cloud Feign

En la creación del proyecto se agregaro lña dependencia feign por ende solo es configurarlo en el microservicio de msvc-cursos

* Agregando el feing a la clase main del servicii
  + Agrear la @EnableFeignClients
* Crear el packege clients
  + Crear la interface usuarioClienteRest
    - Agregar la anotación @FeignClient
      * Agregar el microservicio que va aconsumir (el spring.app.name en este caso msvc-usuarios)
      * Agregar la URL base del microservicio a consumir (localhost:8000)
  + Procedo a consumir los métodos que necesito del otro microservicio
    - Método Detalle este devuelve un usuario, la idea es que los métodos queden lo mas similares posibles
* Inyecto el UsuarioClientRest en la clase CursoServiceImpl

Añadiendo e implementando métodos de comunicación HTTP en el Service

* Configrurando los métodos que mapean la data del microservicio al pojo UsuarioServiceRest – mscv-curso
  + asginarUsuario
  + Crear
  + Eliminar

Añadiendo métodos de comunicación en el Controlador Rest

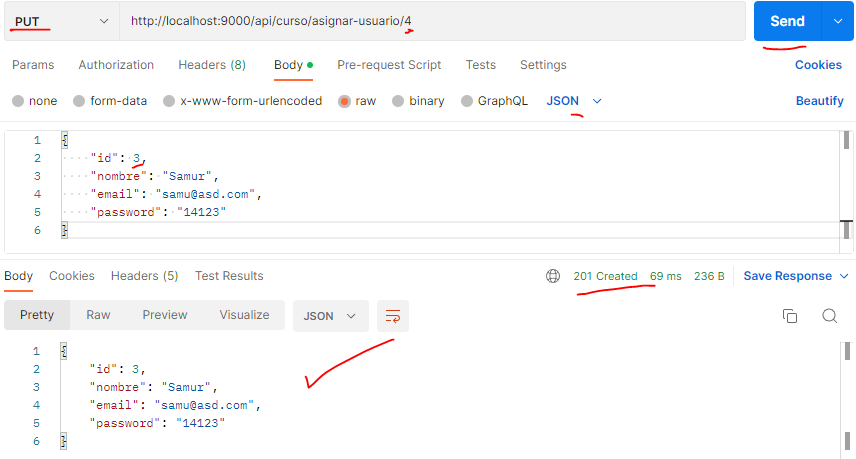
Configurando el controller de curso para los nuevos métodos

Probando comunicaciones HTTP entre microservicios en Postman

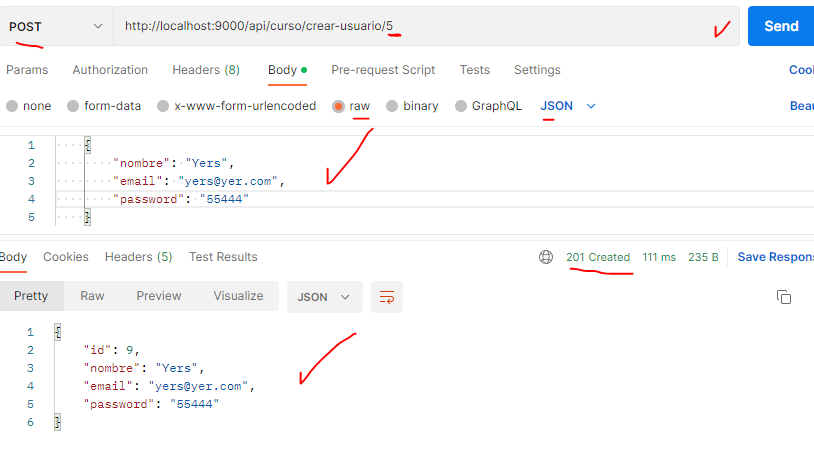
Pruebas

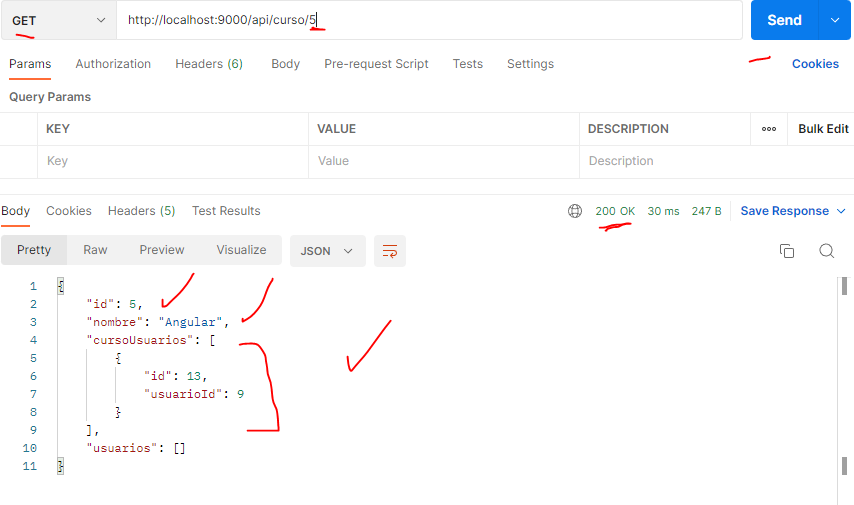
* Asignando el user id=3 al curso(id=4) de Java

OJO IMPORTANTE HACE FALTA MANEJAR LA EXEPCION cuando el curso NO SE ENCUENTRA REGISTRADO

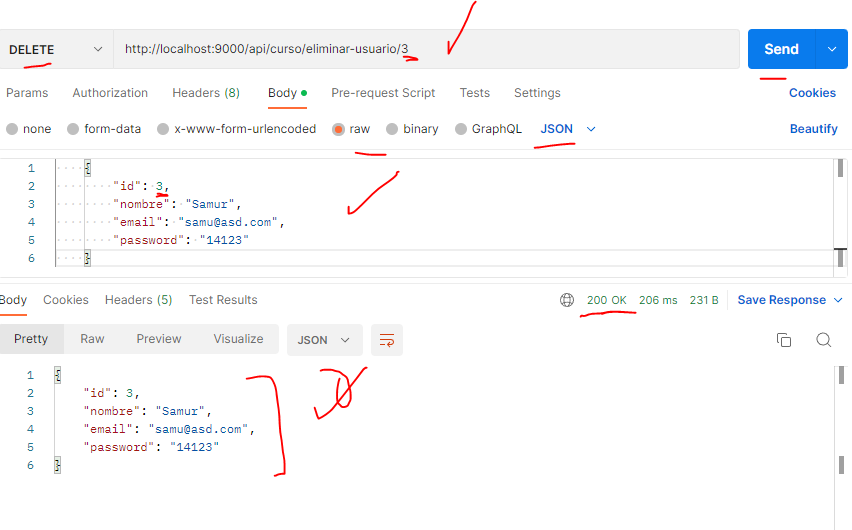


* Creando un usuario nuevo y registrándolo en un Curso(id=5 = Angular)





* Eliminando un USUARIO(id=3) de un CURSO(ID=3 microservices)

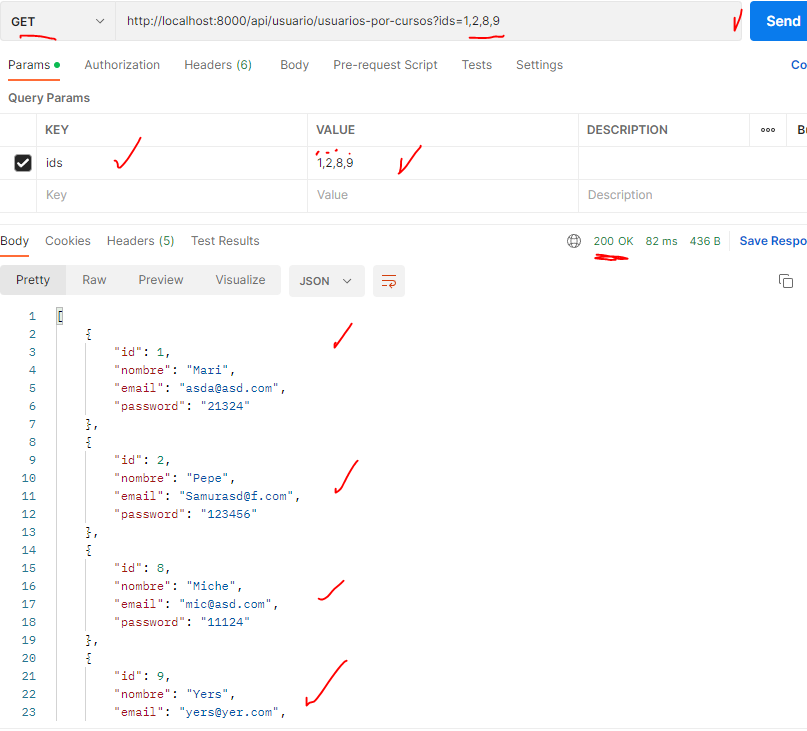


 Msvc usuarios obtener alumnos por ids

Por cada curso se va buscar la lista completa de los usuarios con el detalle en un solo REQUEST desde usuario a curso

Configurando el micrsoservicio msvc-usuario

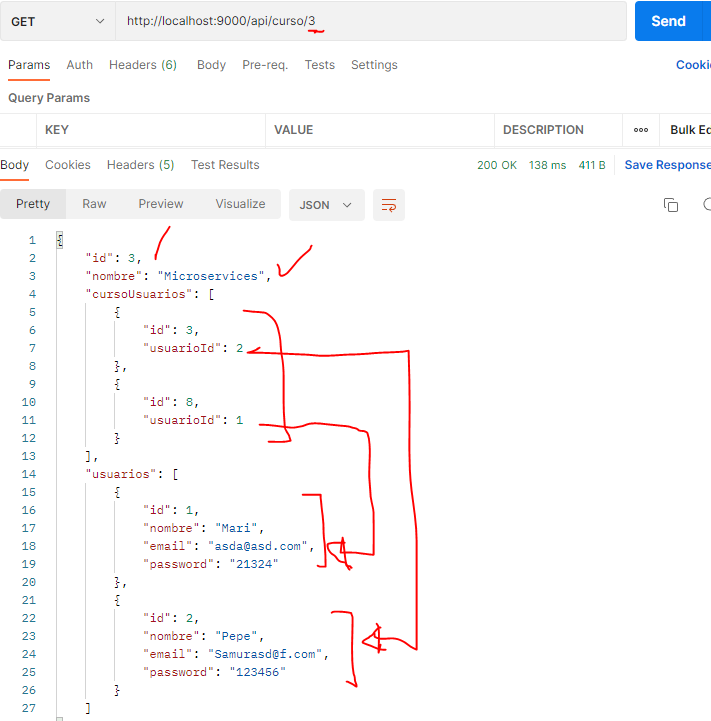
* Como vemos el repositorio ya me trae la lista por id y solo es implmentar el método findAllById
* Configurando Services
  + UsuarioService - UsuarioServiceImpl
* Implementar el controller
* Prueba



Msvc cursos detalle del curso con los alumnos asignados

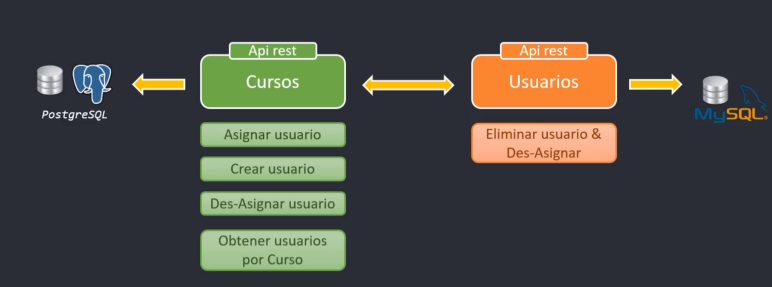
Se procede a configurar el método anterior de finadAllByIds en el Feign de cursos para que mapee en el request la data de este método

* Configurando UsuarioClientRest (CUIDADO CONCEPTO CLAVE UNA LISTA COMO PARAMETRO DE ENTRADA EN EL FEIGN DA PROBLEMAS LO MEJOR ES CAMBIARLA POR UN ITERABLE)
* Configurando el service y serviceImpl para la función del nuevo metodo
* Configurando el controller
* Metodo GETMAPPING(“/{id}”)
* Prueba



Des-asignar un alumno del curso al ser eliminado en el Msvc usuarios

Clase Op lo que valida es que cuando se elimine un usuario de la DB de usuario y este se encontraba registrado en algún curso lo va a desasignar automáticamente



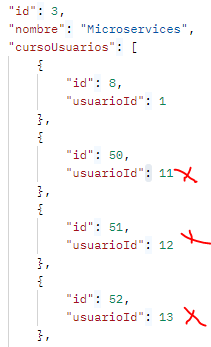
* Modificando CursoRepository – agregar método para eliminar por id, se realiza por medio de un query yagregar ka anotación opara que afecte en la DB
* Configurar el service para implementar el método creado
* Configurar el controller para el método Delete

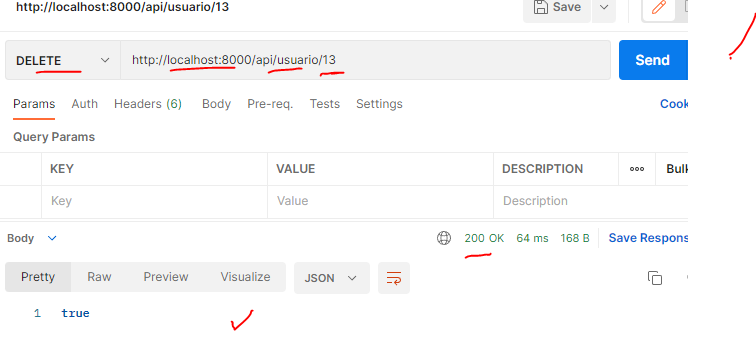
Agregando en cliente Http de Msvc-usuarios para eliminar alumno de Msvc-cursos

Se configurar la operación FEIGN para conectar de usuarios a cursos

* Creando el paquete client
  + Crear interface y configurar cursoClienteRest
* Configurar el service para que llame a la interface del feign y pueda desasignar al usuario del curso por si tenia alguno asignado
  + Inyectar la dependencia
  + Configurar el método eliminar
* Configurar el main para habilitar el feign
* Pruebas

Para este caso se eliminara el usuario con id=11,12,13 desde msvc-usuario y vemos que en automatico se desvincula del curso microservices en el que se encuentra asignado





EN automatico la se designan los usuarios elimnados del curso que tenían asignado para este caso el 3 microservicios

