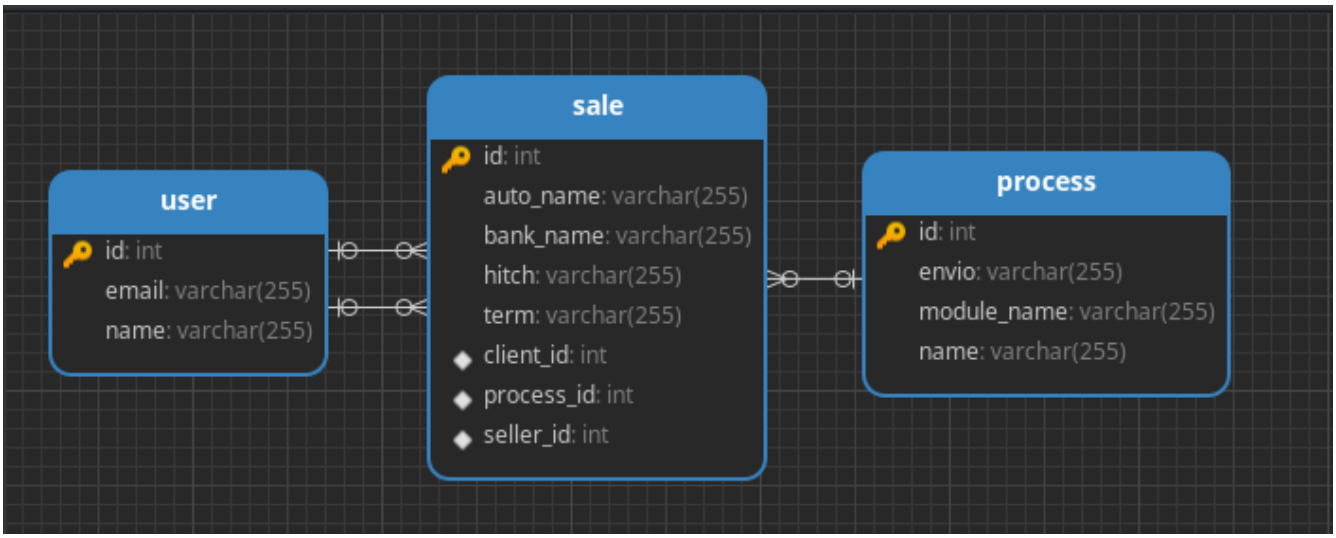
Technical Laboratory

Agustín Rosales Corona

Modelo ER:



Para resolver el reto se tomo en cuenta tener los datos de las ventas como parte central, a partir de este punto se determina que cliente se encuentra en que proceso, teniendo un vendedor asignado, en un análisis muy general de los datos se conoce que el usuario (cliente y vendedor) debe contener como mínimo el nombre y la dirección de correo electrónico, de la misma manera las otras entidades cuentan con los campos que se pueden apreciar en la imagen, se podrían agregar mas campos, pero en el caso de este reto se opto por solo considerar los necesarios.

Según el requerimiento "*Por vendedor el proceso debe tener la lista de las personas como destinatario*" , para esto se toma a partir de sacar la relación en sale, de todos los registros de sale donde seller_id y process_id sean nuestros vendedor y proceso de interés, esto nos regresa todos los registros de sale y de estos mismos obtenemos los clientes relacionados con client_id, una vez obtenidos los clientes tomamos sus direcciones de correo y mandamos el mensaje pertinente.

También se especifico un ejemplo de lo que recibe el endpoint:

Como ejemplo: el microservicio esperaría recibir:

```
Módulo
Proceso
Map de key-value (placeholder)
correo de cliente
```

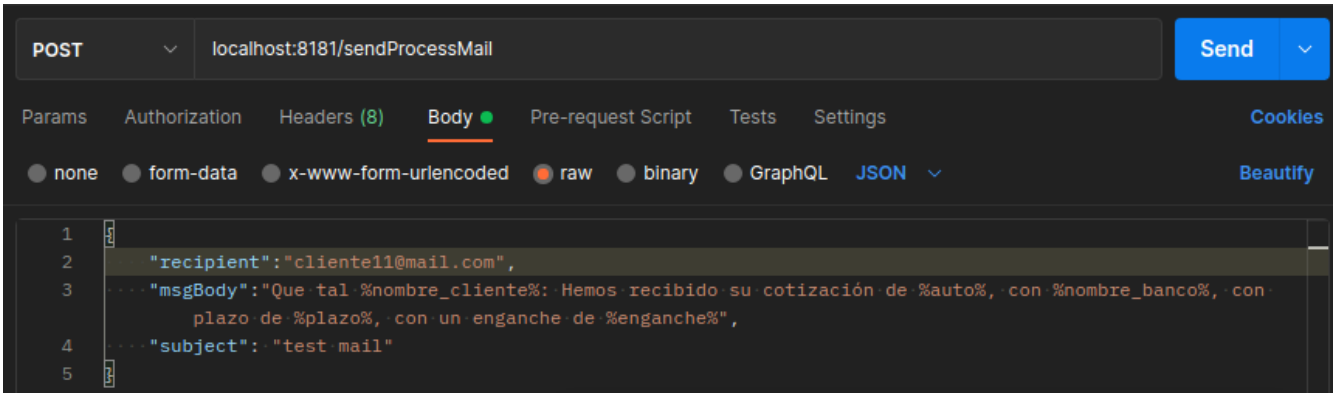
Al final se puede observar que se solicita el correo electrónico del cliente, aquí me surgieron algunas dudas, ya que anteriormente se especifico que según el proceso se mandarían correos a los clientes en ese proceso, sin embargo opte por realizar dos endpoints, que cumpliera cada uno con estos dos requisitos.

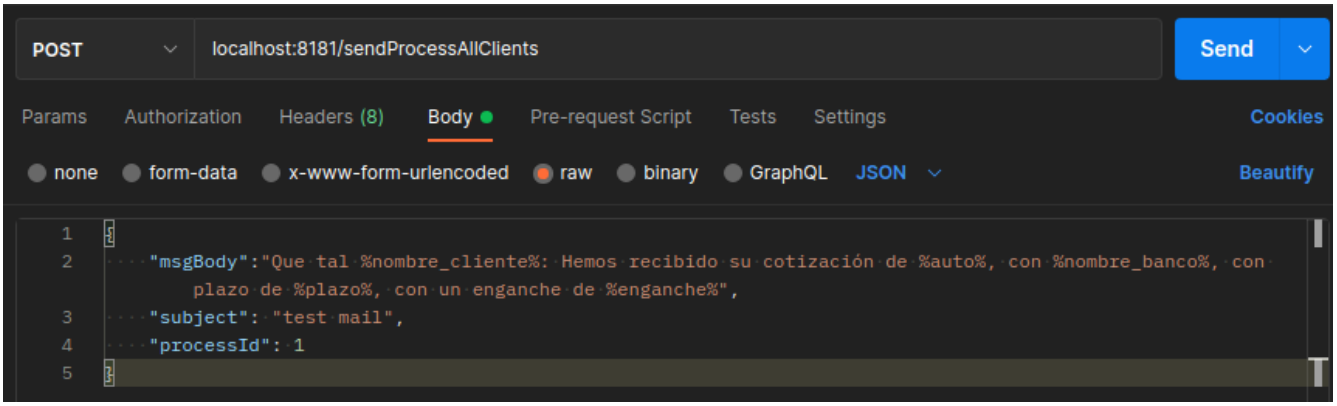
Los endpoints son los siguientes:

- POST: localhost:8181/sendProcessMail
este endpoint cumple el requisito de recibir una direccion de correo electronico, esta se busca en BD en la tabla Users para comprobar que dicho correo pertenece a un cliente

- POST: localhost:8181/sendProcessAllClients
Este endpoint cumple el requerimiento original, se manda el id del proceso, este se corrobora en la base de datos, como cada proceso tiene solo un vendedor este se toma directamente, se buscan todos los usuarios de ese proceso, se itera, se obtienen los correos electrónicos y se manda el mensaje.

Ambos endpoints reciben un Body en JSON





Probando funcionamiento:

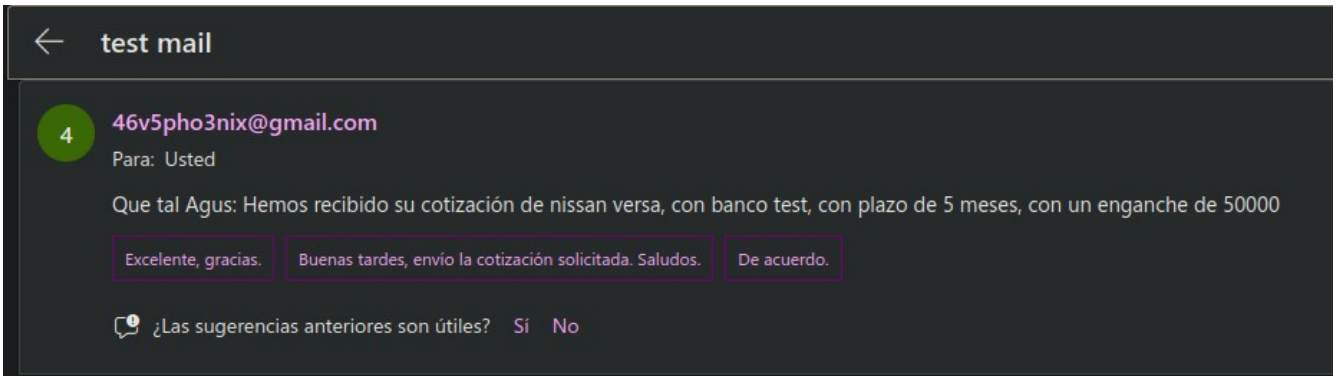
Para probar el envió y recepción de correos registre provisionalmente un usuario cliente con un correo personal en la base de datos:

7	vendedorContacto@mail.c...	Vendedor Contacto
8	agusrc2020@outlook.com	Agus
9	cliente12@mail.com	cliente cita 2

Podemos apreciar que dicho usuario se encuentra en el proceso 1:

id	auto_name	bank_name	hitch	term	client_id	process_id	seller_id
1	nissan versa	banco test	50000	5 meses	8	1	1
2	auto test	banco test	60000	6 meses	9	1	1

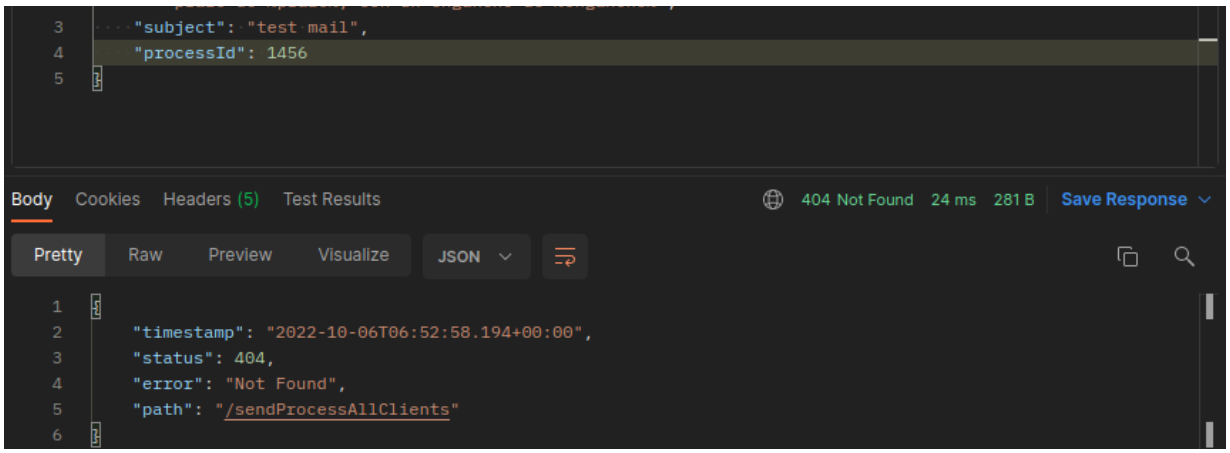
mensaje recibido:



Mensajes enviados impresos en consola:

```
2022-10-06 01:37:06.629  INFO 119740 --- [nio-8181-exec-2] o.s.web.servlet.DispatcherServlet        : Completed initialization in 2 ms
Que tal Agus: Hemos recibido su cotización de nissan versa, con banco test, con plazo de 5 meses, con un enganche de 50000
Que tal cliente cita 2: Hemos recibido su cotización de auto test, con banco test, con plazo de 6 meses, con un enganche de 60000
```

excepcion 404 cuando se manda un processId no valido:



Conclusión:

A pesar de que personalmente no tenia experiencia en Spring Boot me pareció muy ameno y parecido a lo que ya había manejado en NodeJs, el uso de controllers, services y repositorios, el ORM, librerías externas y conexiones con base de datos es muy parecido, solo era cuestión de adaptarme a la sintaxis y a las cuestiones mas especificas del framework, por cuestiones de tiempo no logre empezar la parte no relacional, aunque también he de mencionar que su contraparte en NodeJS la conozco, utilizando mongoose para el servicio MongoDB.

Algunas consideraciones que por cuestiones de tiempo no pude implementar pero que serian prudentes realizar en un proyecto real serian, pedir especificaciones mas concretas de ciertos requerimientos, despejar dudas respecto a estos, como definir los placeholders, para así agregar excepciones si el usuario agrega valores no validos, agregar mas validaciones a base de datos, uniques, not nulls, deafult values, tipados mas estrictos, optimizar querys.

En cuanto al código algo que puedo decir que puedo mejorar es el orden de los archivos de código, controllers, services y repositories en package separados, algo que intente hacer ya avanzado el proyecto, pero Spring boot lanzo un error extraño que no considere prudente invertirle tiempo, a pesar de que no utilice un Lint intente ser lo mas limpio posible en el indexado del código, aunque también hubiese sido bueno implementar uno, el uso de GitFlow en el repositorio de GIT es algo que también suelo usar, en este caso no lo vi necesario ya que se trato de un proyecto de una sola persona, lo cual no lo hace muy útil.