



# Backend de Informe de Servicios por Clientes que mayor descuento generó

Sprint deadline	29/11/2023
Tarjeta	SCRUM-46
Responsable	Martín Paez

*Tabla 1 Detalle de la tarjeta correspondiente en Jira*

## 1. Objetivos. Contenido de la tarjeta

- Implementar en la base de datos un store procedure y una vista para generar un informe del servicio que mayor descuento le generó a cada cliente.
- Agregar funcionalidad a la API para que el frontend pueda obtener la información de la base de datos:
  - Implementar Controller
  - Implementar Service
  - Implementar Repositorio



## 2. Dependencias

Existe dependencia con la tarjeta “SCRUM-47 Frontend de Informe de Servicios por Clientes que mayor descuento generó”. Se consultó al encargado de dicha tarjeta que datos necesitaba obtener para mostrar la información en función de aquello que pudiera implementar. Luego, se acordó como debía ser el endpoint de la API antes de comenzar el desarrollo de ambas partes.

## 3. Entregable

- Store Procedure
- Vista
- Nueva funcionalidad para que la API entregue la totalidad de la información que necesita la implementación de la tarjeta SCRUM-52.

## 1. Procedimientos

### 1.1. Metodología

Una vez decidido como debía ser la API se implementó el Controlador. En paralelo, aprovechando a los tutores, y, al hecho de que todas las tarjetas de corriente Sprint necesitaban resolver estas mismas cuestiones técnicas, se fue investigando eh intercambiando información con el resto del equipo de un modo coordinado para afrontar el reto.

Para cuando pudimos comunicar la base de datos con el backend ya estaba listo el Controlador, la cual es una pieza fundamental ya que establece el contrato para la comunicación entre el frontend y el backend, o sea, la dependencia con la tarjeta SCRUM52.

La primera solución para comunicar el backend con la base de datos recibía una Entity, la cual era suficiente para que parte del equipo completara sus responsabilidades. Sin embargo,



se continuó profundizando, en solitario, la investigación sobre como enviar parámetros y recibir información en otro formato.

Una vez resueltas todas las cuestiones relativas a la comunicación con la base de datos ser terminó la tarjeta para dejar la funcionalidad terminada a disposición, tanto, de quien estaba a cargo de la tarjeta SCRUM-52, como, de quien la utilizaría como plantilla para terminar su correspondiente implementación del backend, respectiva a otra tarjeta.

## 1.2. View y Store Procedure

La dificultad de esta tarjeta radicaba en el hecho de que no conocíamos el mecanismo mediante el cual es posible ejecutar un Store Procedure utilizando Spring Data JPA. Se partió de la información provista por los tutores, pero se terminó empleando un mecanismo distinto para lograr mapear la respuesta de la base de datos a un DTO Response. Este cambio en el método se debió a las características de la información que retorna el procedimiento, esta no puede ser mapeada a una Entity en particular.

Además, se investigó sobre el manejo de fechas desde SQL, por ejemplo, la operación de suma y cuestiones relativas a usos horarios.

Por último, hay que mencionar la dificultad asociada al hecho de que nuestro sistema reconoce dos tipos de clientes: las empresas y las personas físicas. El informe requería que se tratara a ambos por igual, de modo tal que en la misma columna de clientes apareciera el nombre y apellido de la persona física, o bien, el nombre de fantasía de la empresa. La solución se encontró del lado de la base de datos, dentro de la Vista, donde se evalúa que tipo de cliente es y se completa el registro de la nueva tabla con el nombre correspondiente.

## 2. Conclusión

Se trabajó en equipo para encontrar diversas formas de ejecutar y obtener la información de los Store Procedure de la base de datos usando Spring Data JPA. Además se completo la funcionalidad del backend que permite obtener los servicios que mayor descuento generaron a cada cliente.