Backend de Informe de Historial de Pedidos discriminado por cliente, servicio y producto

Sprint deadline	01/12/2023
Tarjeta	SCRUM-50
Responsable	Malleret, Luciano Joaquín

Tabla 1 Detalle de la tarjeta correspondiente en Jira

1. Objetivos. Contenido de la tarjeta:

- Crear store procedure, generar informe de servicios por clientes que mayor descuento generó.
- Controller del store procedure
- Service del store procedure
- Repositorio del store procedure

2. Dependencias

Se necesitaba la entidad de pedidos completa en la base de datos con todas las columnas. En este caso faltaba implementar el patrón state se tuvo que esperar a que finalice la implementación para poder cerrar esta tarea.

3. Procedimientos

3.1. Creación del Stored Procedure

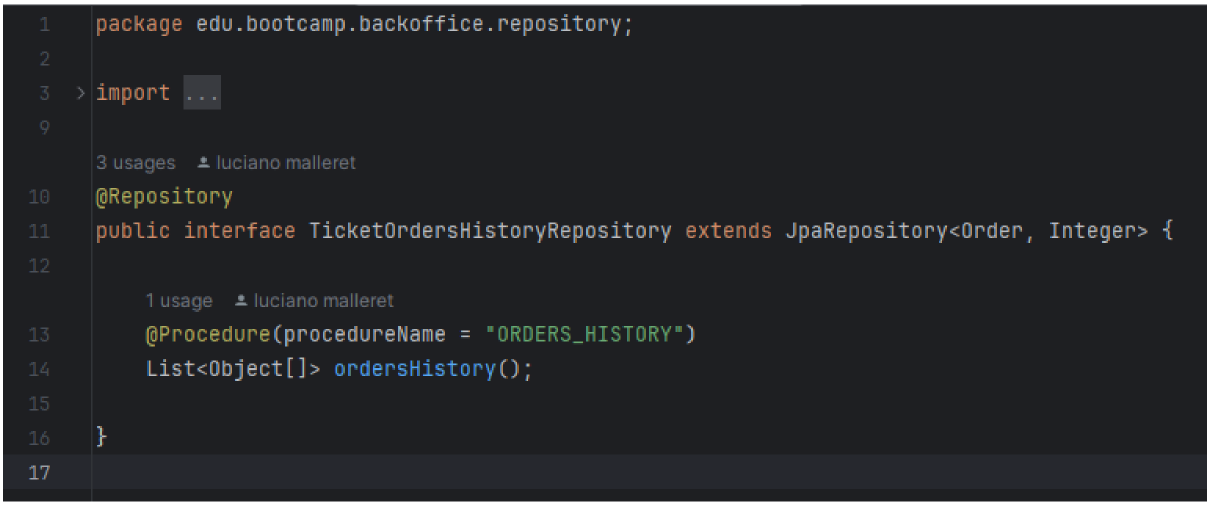
Se creó un stored procedure con los requerimientos necesarios solicitados para generar la tabla.

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[ORDERS_HISTORY]
AS
BEGIN
WITH CombinedData AS (
SELECT
        "cli"."name",
        "cli".last_name,
        "cli".is_bussiness,
        "cli".bussiness_name,
        "order".id AS order_id,
        "order".order_state,
        "order".[date],
        "prod"."name" AS product_name,
        "prod_detail".quantity,
        "prod_detail".base_price,
        "prod_detail".sub_total,
        SUM("tax_by_order".amount) AS total_impuestos,
        "order".total_discount,
        "order"."total"
    FROM dbo.order_table AS "order"
    INNER JOIN dbo.client AS "cli" ON
"order".client_id = "cli".id
    INNER JOIN dbo.product_detail AS "prod_detail" ON
"prod_detail"."order_id" = "order".id
    INNER JOIN dbo.product_table AS "prod" ON
"prod_detail".product_id = "prod".id
    LEFT JOIN dbo.tax_by_order AS "tax_by_order" ON
"tax_by_order"."order_id" = "order"."id"
    GROUP BY "order"."id",
            "cli"."name",
            "cli".last_name,
            "cli".is_bussiness,
            "cli".bussiness_name,
            "order"."order_state",
            "order"."date",
            "prod"."name",
            "prod_detail"."base_price",
            "order".total_discount,
            "prod_detail".sub_total,
            "order".total,
            "prod_detail"."quantity"
    UNION ALL
    SELECT
        "cli"."name",
        "cli".last_name,
        "cli".is_bussiness,
        "cli".bussiness_name,
        "order".id AS order_id,
        "order".order_state,
        "order".[date],
        "serv"."name" AS service_name,
        1 AS quantity,
        "serv_detail".base_price,
        "serv_detail".sub_total,
        SUM("tax_by_order".amount) AS total_impuestos,
        "order".total_discount,
        "order"."total"
    FROM dbo.order_table AS "order"
    INNER JOIN dbo.client AS "cli" ON
"order".client_id = "cli".id
    INNER JOIN dbo.service_detail AS "serv_detail" ON
"serv_detail"."order_id" = "order".id
    INNER JOIN dbo.service_table AS "serv" ON
"serv_detail".service_id = "serv".id
    LEFT JOIN dbo.tax_by_order AS "tax_by_order" ON
"tax_by_order"."order_id" = "order"."id"
    GROUP BY "order"."id",
            "cli"."name",
            "cli".last_name,
            "cli".is_bussiness,
            "cli".bussiness_name,
            "order"."order_state",
            "order"."date",
            "serv"."name",
            "serv_detail"."base_price",
            "order".total_discount,
            "serv_detail".sub_total,
            "order".total
)

SELECT
    ROW_NUMBER() OVER (ORDER BY CAST(GETDATE() AS
TIMESTAMP)) AS ticket_id, *
FROM CombinedData
```

```
ORDER BY name, last_name, bussiness_name, order_id,
product_name, date
END
GO
```

3.2. Conexión con el repositorio

Se realizó la conexión con el repositorio para poder llamar al stored procedure para su posterior utilización.



```
package edu.bootcamp.backoffice.repository;

import ...

3 usages   luciano malleret
@Repository
public interface TicketOrdersHistoryRepository extends JpaRepository<Order, Integer> {

    1 usage   luciano malleret
    @Procedure(procedureName = "ORDERS_HISTORY")
    List<Object[]> ordersHistory();

}
```

3.3. Construcción del dto

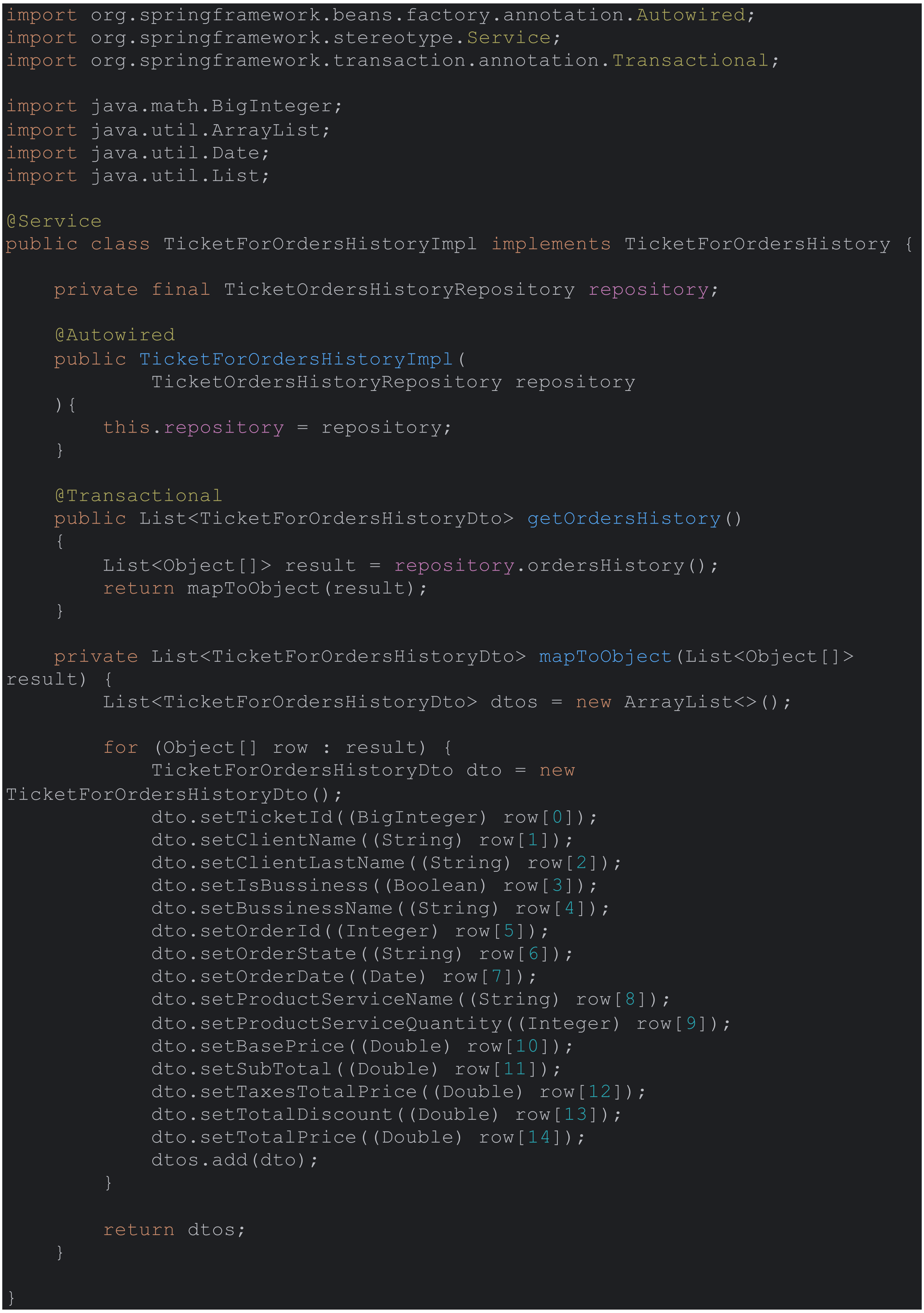
Construcción del dto para ser pasado por la API al front.

```
package edu.bootcamp.backoffice.model.ticket;

import com.fasterxml.jackson.annotation.JsonProperty;
import lombok.*;

import java.math.BigInteger;
import java.util.Date;

@Getter
@Setter
@AllArgsConstructor
@NoArgsConstructor
@Builder
public class TicketForOrdersHistoryDto {

    @JsonProperty("ticket_id")
    private BigInteger ticketId;

    @JsonProperty ("client_name")
    private String clientName;

    @JsonProperty ("client_lastname")
    private String clientLastName;

    @JsonProperty ("is_bussiness")
    private Boolean isBussiness;

    @JsonProperty ("bussiness_name")
    private String bussinessName;

    @JsonProperty ("order_id")
    private Integer orderId;

    @JsonProperty ("order_state")
    private String orderState;

    @JsonProperty ("order_date")
    private Date orderDate;

    @JsonProperty ("product_service_name")
    private String productServiceName;

    @JsonProperty ("product_service_quantity")
    private Integer productServiceQuantity;

    @JsonProperty ("base_price")
    private Double basePrice;

    @JsonProperty ("sub_total")
    private Double subTotal;

    @JsonProperty ("taxes_total_price")
    private Double taxesTotalPrice;

    @JsonProperty ("total_discount")
    private Double totalDiscount;

    @JsonProperty ("total_price")
    private Double totalPrice;

}
```

3.4. Construcción del service

Construcción del service para llamar al stored procedure y generar el dto para pasarlo al controller.

```
package edu.bootcamp.backoffice.service;

import edu.bootcamp.backoffice.model.ticket.TicketForOrdersHistoryDto;
import edu.bootcamp.backoffice.repository.TicketOrdersHistoryRepository;
import edu.bootcamp.backoffice.service.Interface.TicketForOrdersHistory;
```



```
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Service;
import org.springframework.transaction.annotation.Transactional;

import java.math.BigInteger;
import java.util.ArrayList;
import java.util.Date;
import java.util.List;

@Service
public class TicketForOrdersHistoryImpl implements TicketForOrdersHistory {

    private final TicketOrdersHistoryRepository repository;

    @Autowired
    public TicketForOrdersHistoryImpl(
            TicketOrdersHistoryRepository repository
    ){
        this.repository = repository;
    }

    @Transactional
    public List<TicketForOrdersHistoryDto> getOrdersHistory()
    {
        List<Object[]> result = repository.ordersHistory();
        return mapToObject(result);
    }

    private List<TicketForOrdersHistoryDto> mapToObject(List<Object[]>
result) {
        List<TicketForOrdersHistoryDto> dtos = new ArrayList<>();

        for (Object[] row : result) {
            TicketForOrdersHistoryDto dto = new
TicketForOrdersHistoryDto();
            dto.setTicketId((BigInteger) row[0]);
            dto.setClientName((String) row[1]);
            dto.setClientLastName((String) row[2]);
            dto.setIsBussiness((Boolean) row[3]);
            dto.setBussinessName((String) row[4]);
            dto.setOrderId((Integer) row[5]);
            dto.setOrderState((String) row[6]);
            dto.setOrderDate((Date) row[7]);
            dto.setProductServiceName((String) row[8]);
            dto.setProductServiceQuantity((Integer) row[9]);
            dto.setBasePrice((Double) row[10]);
            dto.setSubTotal((Double) row[11]);
            dto.setTaxesTotalPrice((Double) row[12]);
            dto.setTotalDiscount((Double) row[13]);
            dto.setTotalPrice((Double) row[14]);
            dtos.add(dto);
        }

        return dtos;
    }

}
```

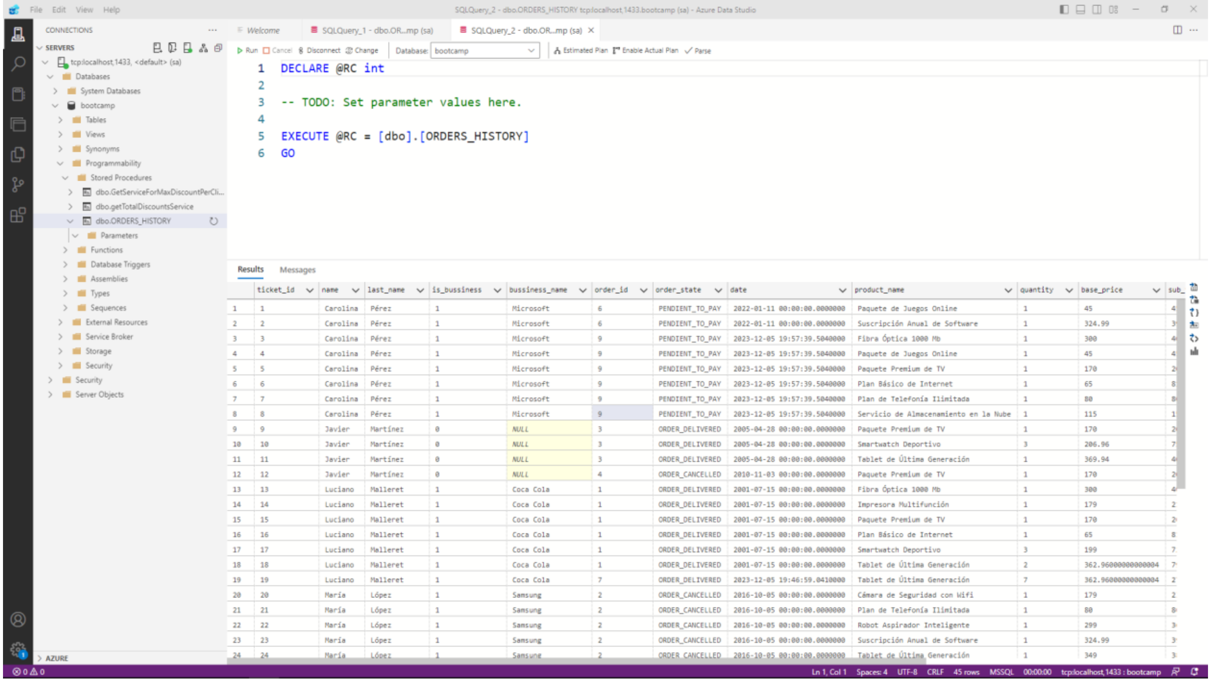
3.5. Construcción del controller

Construcción del controller con el end-point para llamar a través de la API.

```
@Transactional
@GetMapping(
        path = "/orders-history",
        produces = MediaType.APPLICATION_JSON_VALUE
)
public ResponseEntity<List<TicketForOrdersHistoryDto>> getOrdersHistory() {
    List<TicketForOrdersHistoryDto> tickets =
ticketService.getOrdersHistory();
    return ResponseEntity.ok(tickets);
}
```

4. Resultados

4.1. Generación de la tabla



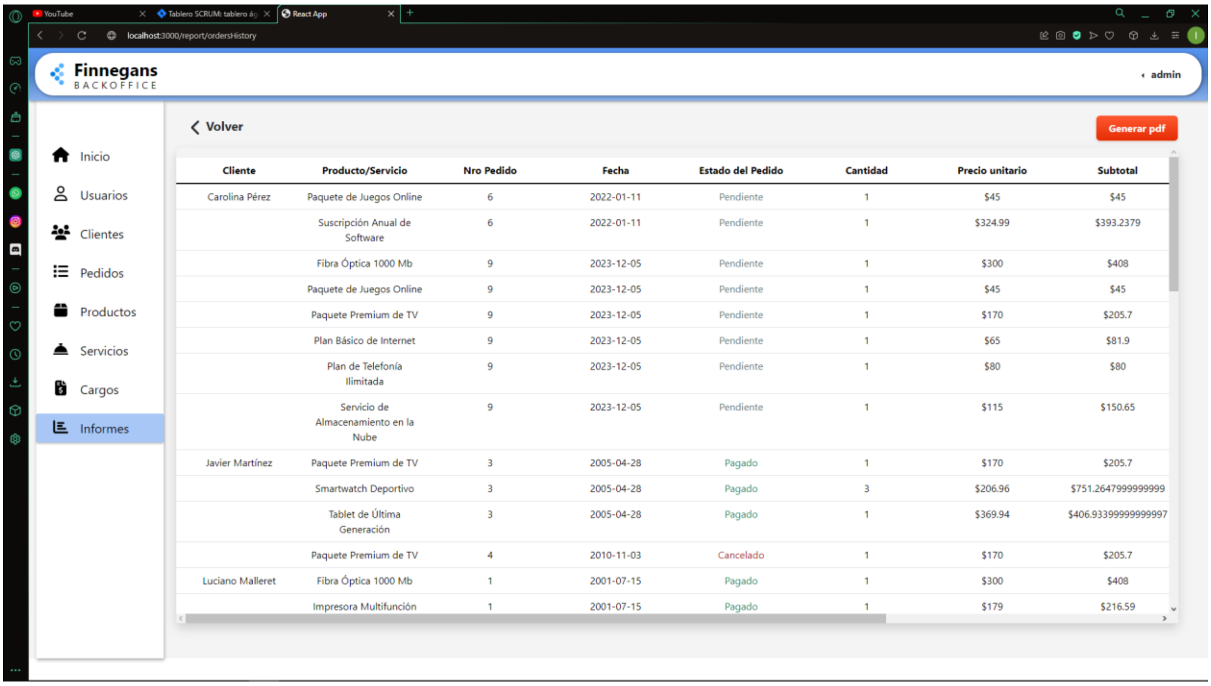
```
DECLARE @RC int

-- TODO: Set parameter values here.

EXECUTE @RC = [dbo].[ORDERS_HISTORY]
GO
```

	ticket_id	name	last_name	is_bussiness	bussiness_name	order_id	order_state	date	product_name	quantity	base_price	sub_
1	1	Carolina	Pérez	1	Microsoft	6	PENDIENT_TO_PAY	2022-01-11 00:00:00.0000000	Paquete de Juegos Online	1	45	4
2	2	Carolina	Pérez	1	Microsoft	6	PENDIENT_TO_PAY	2022-01-11 00:00:00.0000000	Suscripción Anual de Software	1	324.99	3
3	3	Carolina	Pérez	1	Microsoft	9	PENDIENT_TO_PAY	2023-12-05 19:57:39.5040000	Fibra Óptica 1000 Mb	1	300	4
4	4	Carolina	Pérez	1	Microsoft	9	PENDIENT_TO_PAY	2023-12-05 19:57:39.5040000	Paquete de Juegos Online	1	45	4
5	5	Carolina	Pérez	1	Microsoft	9	PENDIENT_TO_PAY	2023-12-05 19:57:39.5040000	Paquete Premium de TV	1	170	2
6	6	Carolina	Pérez	1	Microsoft	9	PENDIENT_TO_PAY	2023-12-05 19:57:39.5040000	Plan Básico de Internet	1	65	8
7	7	Carolina	Pérez	1	Microsoft	9	PENDIENT_TO_PAY	2023-12-05 19:57:39.5040000	Plan de Telefonía Ilimitada	1	80	8
8	8	Carolina	Pérez	1	Microsoft	9	PENDIENT_TO_PAY	2023-12-05 19:57:39.5040000	Servicio de Almacenamiento en la Nube	1	115	1
9	9	Javier	Martínez	0	NULL	3	ORDER_DELIVERED	2005-04-28 00:00:00.0000000	Paquete Premium de TV	1	170	2
10	10	Javier	Martínez	0	NULL	3	ORDER_DELIVERED	2005-04-28 00:00:00.0000000	Smartwatch Deportivo	3	206.96	7
11	11	Javier	Martínez	0	NULL	3	ORDER_DELIVERED	2005-04-28 00:00:00.0000000	Tablet de Última Generación	1	369.94	4
12	12	Javier	Martínez	0	NULL	4	ORDER_CANCELLED	2010-11-03 00:00:00.0000000	Paquete Premium de TV	1	170	2
13	13	Luciano	Malleret	1	Coca Cola	1	ORDER_DELIVERED	2001-07-15 00:00:00.0000000	Fibra Óptica 1000 Mb	1	300	4
14	14	Luciano	Malleret	1	Coca Cola	1	ORDER_DELIVERED	2001-07-15 00:00:00.0000000	Impresora Multifunción	1	179	2
15	15	Luciano	Malleret	1	Coca Cola	1	ORDER_DELIVERED	2001-07-15 00:00:00.0000000	Paquete Premium de TV	1	170	2
16	16	Luciano	Malleret	1	Coca Cola	1	ORDER_DELIVERED	2001-07-15 00:00:00.0000000	Plan Básico de Internet	1	65	8
17	17	Luciano	Malleret	1	Coca Cola	1	ORDER_DELIVERED	2001-07-15 00:00:00.0000000	Smartwatch Deportivo	3	199	7
18	18	Luciano	Malleret	1	Coca Cola	1	ORDER_DELIVERED	2001-07-15 00:00:00.0000000	Tablet de Última Generación	2	362.96000000000004	7
19	19	Luciano	Malleret	1	Coca Cola	7	ORDER_DELIVERED	2023-12-05 19:46:59.0410000	Tablet de Última Generación	7	362.96000000000004	2
20	20	María	López	1	Samsung	2	ORDER_CANCELLED	2016-10-05 00:00:00.0000000	Cámara de Seguridad con Wifi	1	179	2
21	21	María	López	1	Samsung	2	ORDER_CANCELLED	2016-10-05 00:00:00.0000000	Plan de Telefonía Ilimitada	1	80	8
22	22	María	López	1	Samsung	2	ORDER_CANCELLED	2016-10-05 00:00:00.0000000	Robot Aspirador Inteligente	1	299	3
23	23	María	López	1	Samsung	2	ORDER_CANCELLED	2016-10-05 00:00:00.0000000	Suscripción Anual de Software	1	324.99	3
24	24	María	López	1	Samsung	2	ORDER_CANCELLED	2016-10-05 00:00:00.0000000	Tablet de Última Generación	1	349	3

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF 45 rows MSSQL 00:00:00 tcp:localhost,1433 : bootcamp

Ilustración 1: Captura de pantalla ejecución del stored procedure

4.2. Visualización de la tabla en el front



Cliente	Producto/Servicio	Nro Pedido	Fecha	Estado del Pedido	Cantidad	Precio unitario	Subtotal
Carolina Pérez	Paquete de Juegos Online	6	2022-01-11	Pendiente	1	$45	$45
	Suscripción Anual de Software	6	2022-01-11	Pendiente	1	$324.99	$393.2379
	Fibra Óptica 1000 Mb	9	2023-12-05	Pendiente	1	$300	$408
	Paquete de Juegos Online	9	2023-12-05	Pendiente	1	$45	$45
	Paquete Premium de TV	9	2023-12-05	Pendiente	1	$170	$205.7
	Plan Básico de Internet	9	2023-12-05	Pendiente	1	$65	$81.9
	Plan de Telefonía Ilimitada	9	2023-12-05	Pendiente	1	$80	$80
	Servicio de Almacenamiento en la Nube	9	2023-12-05	Pendiente	1	$115	$150.65
Javier Martínez	Paquete Premium de TV	3	2005-04-28	Pagado	1	$170	$205.7
	Smartwatch Deportivo	3	2005-04-28	Pagado	3	$206.96	$751.2647999999999
	Tablet de Última Generación	3	2005-04-28	Pagado	1	$369.94	$406.93399999999997
	Paquete Premium de TV	4	2010-11-03	Cancelado	1	$170	$205.7
Luciano Malleret	Fibra Óptica 1000 Mb	1	2001-07-15	Pagado	1	$300	$408
	Impresora Multifunción	1	2001-07-15	Pagado	1	$179	$216.59

Ilustración 2: Captura de pantalla visualización de la información devuelta