Informe TPO BDII

Benjamin Delasoie 61231

Franco Panighini 61258

Santiago Rivas Betancourt 61007

Introducción

Este proyecto tiene como objetivo desarrollar un sistema de Facturación que gestiona la información de clientes, productos y facturas. La implementación se basa en dos bases de datos: PostgreSQL y MongoDB. En este informe, se detallarán las consultas y vistas implementadas, así como las decisiones de diseño tomadas.

Decisiones de Diseño

Modelo de Datos

1. **Clientes:**
   - Se optó por mantener la información del cliente en una tabla llamada `E01_CLIENTE`. Los campos incluyen `nro_cliente`, `nombre`, `apellido`, `direccion`, y `activo`. La clave primaria es `nro_cliente`.
2. **Productos:**
   - La información de los productos se almacena en una tabla `E01_PRODUCTO`. Campos como `codigo_producto`, `marca`, `nombre`, `descripcion`, `precio`, y `stock` están presentes. La clave primaria es `codigo_producto`.
3. **Facturas:**
   - Se implementa una tabla `E01_FACTURA` que registra información sobre las transacciones. Los campos incluyen `nro_factura`, `fecha`, `total_sin_iva`, `iva`, `total_con_iva`, y `nro_cliente`. La clave primaria es `nro_factura`.
4. **Vistas y Consultas:**
   - Se diseñaron vistas para simplificar consultas frecuentes, como la ordenación de facturas por fecha.

Restricciones y Claves Extranjeras

1. **Claves Extranjeras:**
   - Se establecieron relaciones mediante claves foráneas entre las tablas `E01_DETALLE_FACTURA` y `E01_PRODUCTO`, `E01_DETALLE_FACTURA` y `E01_FACTURA`, `E01_FACTURA` y `E01_CLIENTE`, así como entre `E01_TELEFONO` y `E01_CLIENTE`.
2. **Actualización de Precios:**

- Se creó un procedimiento almacenado llamado `calcular_precios` para actualizar los precios en las facturas, considerando descuentos según la cantidad de productos comprados.

MongoDB

A la hora de diseñar los modelos en Mongo, se tomaron decisiones para aprovechar los beneficios del enfoque schemaless de esta tecnología y se encontraron también limitaciones.

En particular se optó por embeber el detalle de la factura, que en el diseño se encontraba normalizado en su propia tabla, dentro de las facturas ya que se consideró pertinente, pensando que el patrón de acceso a los datos de la factura normalmente desean traer todo este modelo completo.

Por otro lado, se decidió mantener normalizados los teléfonos de los clientes, por ejemplo, debido a que encontramos la limitación de que si lo manteníamos embebido dentro del cliente, no se podía asegurar la unicidad de los números de los campos del teléfono, lo cual consideramos un requerimiento.

API: Node y ExpressJS

Para el diseño de la API se decidió utilizar ExpressJS. Express es un framework ligero especialmente eficiente para implementar aplicaciones webs y APIs. Su uso es muy simple y nos permitió definir las rutas y acceder programáticamente a las instancias de PSQL y Mongo.

Consultas y Vistas Implementadas

SQL (PostgreSQL)

```sql
-- 1. Obtener el teléfono y el número de cliente del cliente con nombre
-- "Wanda" y apellido "Baker".

SELECT
E01_TELEFONO.NRO_TELEFONO,
E01_CLIENTE.NRO_CLIENTE
FROM E01_CLIENTE
NATURAL JOIN E01_TELEFONO
WHERE
E01_CLIENTE.NOMBRE =  'Wanda'
AND E01_CLIENTE.APELLIDO =  'Baker';

-- 2. Seleccionar todos los clientes que tengan registrada al menos una factura.

SELECT  *  FROM E01_CLIENTE
WHERE NRO_CLIENTE IN (SELECT DISTINCT NRO_CLIENTEFROM E01_FACTURA);
```

```sql
-- 3. Seleccionar todos los clientes que no tengan registrada una factura.


SELECT  * FROM E01_CLIENTE
WHERE NRO_CLIENTE NOT  IN (SELECT DISTINCT NRO_CLIENTE FROM E01_FACTURA);

-- 4. Seleccionar los productos que han sido facturados al menos 1 vez.


SELECT  *  FROM E01_PRODUCTO
WHERE codigo_producto IN (SELECT DISTINCT codigo_producto
    FROM E01_DETALLE_FACTURA);

-- 5. Seleccionar los datos de los clientes junto con sus teléfonos.
SELECT
E01_CLIENTE.NRO_CLIENTE,
E01_CLIENTE.NOMBRE,
E01_CLIENTE.APELLIDO,
E01_CLIENTE.DIRECCION,
E01_CLIENTE.ACTIVO,
E01_TELEFONO.NRO_TELEFONO
FROM E01_CLIENTE
NATURAL JOIN E01_TELEFONO;

-- 6. Devolver todos los clientes,
-- con la cantidad de facturas que tienen registradas
-- (admitir nulos en valores de Clientes).

SELECT
    E01_FACTURA.nro_cliente,
    COUNT(nro_factura)
FROM
    E01_CLIENTE RIGHT
    JOIN E01_FACTURA
    ON E01_CLIENTE.nro_cliente = E01_FACTURA.nro_cliente
GROUP BY
    E01_FACTURA.nro_cliente;

-- 7. Listar todas las Facturas que hayan sido compradas por
-- el cliente de nombre "Pandora" y apellido "Tate".

SELECT  *  FROM E01_FACTURA
WHERE NRO_CLIENTE IN (
SELECT NRO_CLIENTE
```

```sql
    FROM E01_CLIENTE
    WHERE NOMBRE = 'Pandora' AND APELLIDO = 'Tate'
    );

-- 8. Listar todas las Facturas que contengan productos
-- de la marca "In Faucibus Inc."

SELECT * FROM E01_FACTURA
WHERE nro_factura IN (
SELECT DISTINCT nro_factura
    FROM E01_DETALLE_FACTURA NATURAL JOIN e01_producto
    WHERE marca = 'In Faucibus Inc.'
);

-- 9. Mostrar cada teléfono junto con los datos del cliente.

SELECT
E01_TELEFONO.NRO_TELEFONO,
E01_CLIENTE.NRO_CLIENTE,
E01_CLIENTE.NOMBRE,
E01_CLIENTE.APELLIDO,
E01_CLIENTE.DIRECCION,
E01_CLIENTE.ACTIVO
FROM E01_CLIENTE
NATURAL JOIN E01_TELEFONO;

-- 10. Mostrar nombre y apellido de cada cliente junto con lo que gastó en total
-- (con IVA incluido).

SELECT nombre, apellido, SUM(total_con_iva)
FROM E01_CLIENTE NATURAL JOIN E01_FACTURA
GROUP BY nombre, apellido;

## Vistas
```sql
-- 1. Se debe realizar una vista que devuelva las facturas ordenadas por fecha.
CREATE VIEW facturas_por_fecha AS
SELECT * FROM E01_FACTURA ORDER BY fecha;

-- 2. Se necesita una vista que de devuelva todos los productos que
-- aún no han sido facturados.

CREATE VIEW productos_no_facturados AS
SELECT * FROM E01_PRODUCTO
WHERE codigo_producto NOT IN (
    SELECT codigo_producto FROM E01_DETALLE_FACTURA
```
```

```
);
```

MONGO

Queries

Query 1

```
db.e01_cliente.aggregate([
    {
        $match: {
            "nombre": "Wanda",
            "apellido": "Baker"
        }
    },
    {
        $lookup: {
            from: "e01_telefono",
            localField: "nro_cliente",
            foreignField: "nro_cliente",
            as: "telefono_info"
        }
    },
    {
        $project: {
            "nombre": 1,
            "apellido": 1,
            "telefono_info.nro_telefono": 1
        }
    }
])
```

Query 2

```
db.e01_cliente.aggregate([
    {
        $lookup: {
            from: "e01_factura",
            localField: "nro_cliente",
            foreignField: "nro_cliente",
            as: "facturas"
        }
    },
    {
        $match: {
            "facturas": { $exists: true, $ne: [] }
        }
```

```
    }
])
```

Query 3

```
db.e01_cliente.find({
    "nro_cliente": {
        $nin: db.e01_factura.distinct("nro_cliente")
    }
})
```

Query 4

```
db.e01_factura.aggregate([
    {
        $unwind: "$detalle_facturas"
    },
    {
        $group: {
            _id: "$detalle_facturas.codigo_producto",
            count: { $sum: 1 }
        }
    },
    {
        $match: {
            count: { $gte: 1 }
        }
    }
])
```

Query 5

```
db.e01_cliente.aggregate([
    {
        $lookup: {
            from: "e01_telefono",
            localField: "nro_cliente",
            foreignField: "nro_cliente",
            as: "telefonos"
        }
    }
])
```

Query 6

```
db.e01_cliente.aggregate([
    {
```

```
        $lookup: {
            from: "e01_factura",
            localField: "nro_cliente",
            foreignField: "nro_cliente",
            as: "facturas"
        }
    },
    {
        $project: {
            "_id": 1,
            "nro_cliente": 1,
            "nombre": 1,
            "apellido": 1,
            "direccion": 1,
            "activo": 1,
            "cantidad_facturas": { $size: "$facturas" }
        }
    }
])
```

Query 7

```
db.e01_factura.aggregate([
    {
        $lookup: {
            from: "e01_cliente",
            localField: "nro_cliente",
            foreignField: "nro_cliente",
            as: "cliente"
        }
    },
    {
        $match: {
            "cliente.nombre": "Pandora",
            "cliente.apellido": "Tate"
        }
    }
])
```

Query 8

```
db.e01_factura.aggregate([
    {
        $unwind: "$detalle_facturas"
    },
    {
```

```
        $lookup: {
            from: "e01_producto",
            localField: "detalle_facturas.codigo_producto",
            foreignField: "codigo_producto",
            as: "productos"
        }
    },
    {
        $match: {
            "productos.marca": "In Faucibus Inc."
        }
    },
    {
        $group: {
            _id: "$_id",
            nro_factura: { $first: "$nro_factura" },
            fecha: { $first: "$fecha" },
            total_sin_iva: { $first: "$total_sin_iva" },
            iva: { $first: "$iva" },
            total_con_iva: { $first: "$total_con_iva" },
            nro_cliente: { $first: "$nro_cliente" }
        }
    }
])
```

Query 9

```
db.e01_telefono.aggregate([
  {
    $lookup: {
      from: "e01_cliente",
      localField: "nro_cliente",
      foreignField: "nro_cliente",
      as: "cliente_info"
    }
  },
  {
    $unwind: "$cliente_info"
  },
  {
    $project: {
      _id: 0,
      "Código de Área": "$codigo_area",
      "Número de Teléfono": "$nro_telefono",
      "Tipo de Teléfono": "$tipo",
      "Nombre del Cliente": "$cliente_info.nombre",
```

```
      "Apellido del Cliente": "$cliente_info.apellido",
      "Dirección del Cliente": "$cliente_info.direccion",
      "Activo del Cliente": "$cliente_info.activo"
    }
  }
])
```

Query 10

```
db.e01_factura.aggregate([
  {
    $group: {
      _id: "$nro_cliente",
      totalGasto: { $sum: "$total_con_iva" }
    }
  },
  {
    $lookup: {
      from: "e01_cliente",
      localField: "_id",
      foreignField: "nro_cliente",
      as: "cliente_info"
    }
  },
  {
    $unwind: "$cliente_info"
  },
  {
    $project: {
      _id: 0,
      "Nombre del Cliente": "$cliente_info.nombre",
      "Apellido del Cliente": "$cliente_info.apellido",
      "Gasto Total (con IVA)": "$totalGasto"
    }
  }
])
```

Vistas

Vista 1

```
db.createView("facturas_sorted_by_fecha", "e01_factura", [
  {
    $sort: { fecha: 1 }
  }
])
```

Vista 2

```
db.createView("productos_not_facturados", "e01_producto", [
  {
    $lookup: {
      from: "e01_factura",
      localField: "codigo_producto",
      foreignField: "detalle_facturas.codigo_producto",
      as: "facturas"
    }
  },
  {
    $match: {
      facturas: { $size: 0 }
    }
  }
]);
```