



UNIVERSIDAD DEL ISTMO
INGENIERÍA EN SISTEMAS
ANÁLISIS DISEÑO Y FABRICACIÓN DE SISTEMAS

DOCUMENTACIÓN

Evelyn Dubon
Gabriel Chavarria
Leonel Castañeda

GUATEMALA, ABRIL 2021

Índice

Sistema de Ventas	3
Stacks de tecnologías	3
Diagrama entidad relación	4
Funcion	4
Trigger	5
Procedimiento almacenado	5
Vista 1	6
Vista 2	6
Vista 3	7
Evento	7
Sistema de Fábrica	7
Stacks de tecnologías	7
Colecciones	9
Aggregations	11
Aggregation #1	11
Aggregation #2	12
Herramientas extras para el desarrollo	13

Sistema de Ventas

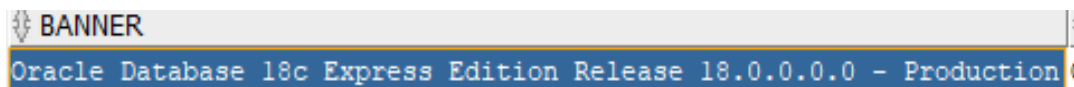
Stacks de tecnologías

Para el sistema de ventas se llegó a utilizar la base de datos oracle xe es una base de datos relacional, es la versión gratuita de oracle que tiene ciertas funcionalidades reducidas pero llega a funcionar de manera correcta.

También se llegó a utilizar Springboot el cual es un framework para el desarrollo de aplicaciones y contenedor de inversión de control, de código abierto para la plataforma Java. Para el front end se llevó a cabo mediante HTML 5, Bootstrap y CSS debido a que consideramos que es una manera sencilla de estructurar el diseño que queremos darle a nuestra página web.

Maven se utiliza en la gestión y construcción de software. Posee la capacidad de realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado

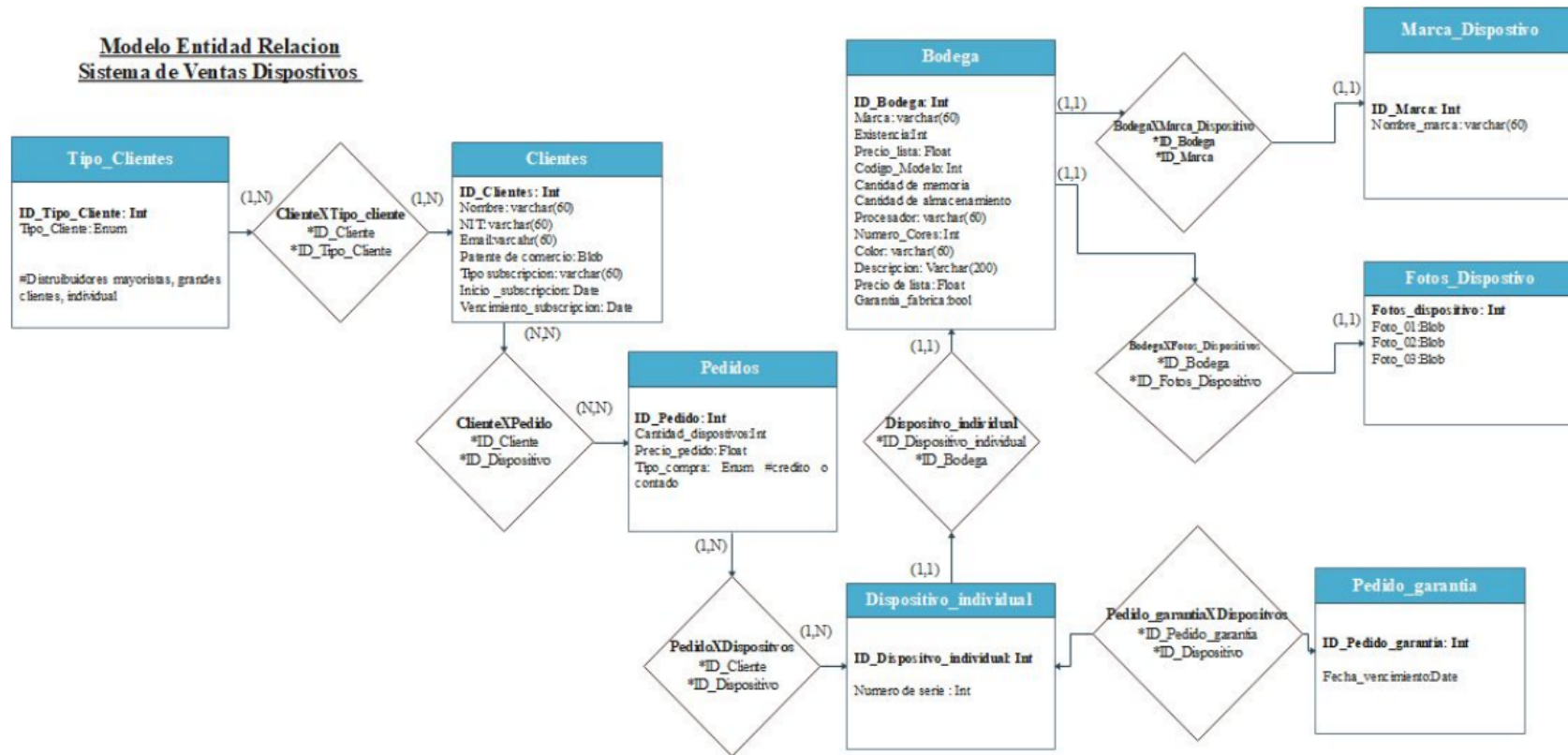
- Oracle xe Version 18c



- HTML 5
- CSS
- Maven
- Spring Boot
- Bootstrap 4

Diagrama entidad relación

Modelo Entidad Relacion
Sistema de Ventas Dispositivos



Funcion

```
create or replace NONEDITIONABLE function precio(tipo_c number, idT number)
    return float

is

costot float;
total float;
begin

    select precio_lista into costot from bodega where id_bodega = idT;
    total := costot + costot*0.15 +costot*0.30+costot*0.05+costot*0.40;
    if tipo_c = 1 then
        total := total - total*0.15;
    elsif tipo_c = 2 then
        total := total - total*0.05;
    else
        total :=total;
    end if;

    return total;
end;
```

Trigger

El trigger llama al procedimiento almacenado cada vez que se confirme un pedido.

```
create or replace NONEDITIONABLE TRIGGER DESCUENTO
after UPDATE of id_estado on PEDIDO

BEGIN

    descu();

END;
```

Procedimiento almacenado

Este procedimiento lo que hace es en un cursor hacer el descuento de los terminales vendidos.

```
create or replace NONEDITIONABLE procedure descu is

begin
  declare
    disponibilidad int;
    descuento int;
    idbn int;
    ide int;
    idc int;
    idp int;
    cursor cur1 is select max(existencia),sum(CANTIDADP_A + CANTIDADP)as descuento ,
max(id_bodega), max(id_estado),max(id_cliente),max(id_pedido) from (
  select *from bodega inner join pedido using(id_bodega))a WHERE id_estado = 2 and
  compra =1 and descontado = 0 group by id_bodega, id_cliente,id_pedido;
  begin
    open cur1;

    loop
      fetch cur1 into disponibilidad,descuento,idbn,ide,idc,idp;
      exit when cur1%notfound;

      update bodega set existencia = (disponibilidad - descuento) where id_bodega = idbn;

      dbms_output.put_line(idbn);
      dbms_output.put_line(descuento);
      dbms_output.put_line(disponibilidad);
    end loop;

    close cur1;

  end;
end;
```

Vista 1

```
SELECT
```

```

ID_PEDIDO AS ID,
ID_CLIENTE,
compra,
nombre,
apellido,
correo,
nombret ,
crédito,
APELLIDO AS apellidoC,
cantidadp + cantidadp_a AS cantidad,
precio(id_tipo_cliente,id_bodega) AS precio_total,
(cantidadp + cantidadp_a)* precio(id_tipo_cliente,id_bodega) AS totales

```

```

FROM bodega inner join(select *from pedido inner join clientes using(id_cliente))a
using(id_bodega)

```

Vista 2

```

SELECT
ID_DISPOSITIVO,
NOMBRET,
NUMERO_SERIE,
CODIGO_MODELO,
NOMBREF,
ID_BODEGA

FROM
ventas.dispositivo_individual inner join(
select *from VENTAS.bodega inner join FABRICA on origen = id_fabrica )a using(id_bodega)

```

Vista 3

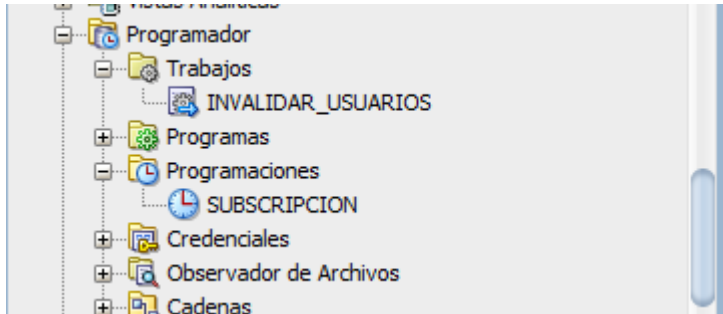
```

select nombref,
id_pedido,
id_cliente,
compra,
cantidadp + cantidadp_a as cantidad,
precio(id_tipo_cliente,id_bodega) as total from clientes inner join(
select *from pedido inner join( select *from fabrica inner join bodega on origen = id_fabrica)a
using (id_bodega))b using(id_cliente)

```

Evento

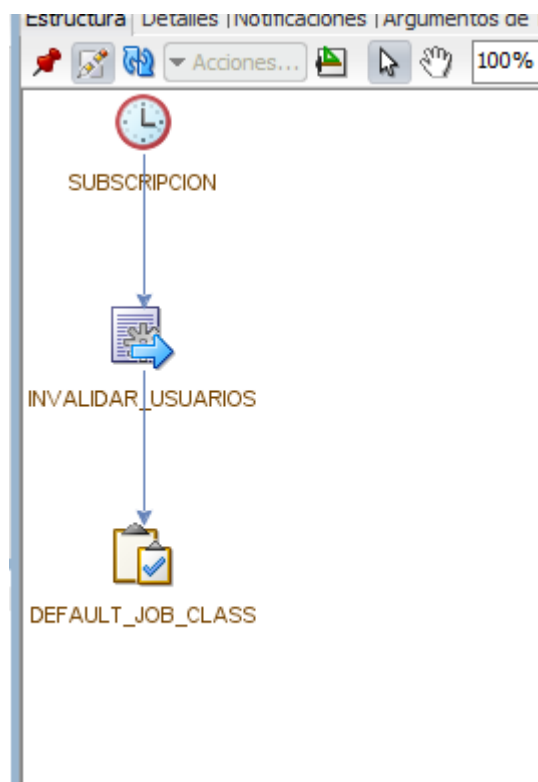
El evento calendarizado es utilizado para quitarle la suscripción a los clientes y volverlos del tipo Individual.



Para esto se crearon dos objetos:

- Primero un trabajo llamado Invalidar usuarios que es la operación lógica para invalidar.
 - La programación es cada cuanto se tiene que realizar.
 - Se puede
- trabajo:

visualizar el flujo del



Sistema de Fábrica

Stacks de tecnologías

El stack de tecnologías que se llegó a utilizar es MEAN Stack para el sistema de fábrica debido a que consideramos que era un stack fácil de manejar y con bastante documentación online. Como base de datos se utiliza MongoDB es un sistema de base de datos no relacional, que almacena los datos en estructuras o documentos, los cuales están definidos con la notación JSON, lo que permite una rápida manipulación y transferencia de los datos.

Express es un método de Node Js el cual nos permite realizar peticiones HTTP(Post, Put, Delete, Get) lo cual hace una manera más sencilla para manipular el API.

Angular lo utilizamos para el desarrollo del Front End, debido que es una single page application y esto da una experiencia más fluida a los usuarios y consideramos que utilizando componentes es una estructura bastante útil y más ordenada.

NodeJs Es la plataforma encargada del funcionamiento del servidor

Las versiones de las tecnologías que se llegaron a utilizar para el sistema de ventas fueron las siguientes:

- Base de datos no relacional MongoDB Version 4.4.3

```
MongoDB shell version v4.4.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("0eec310a-8510-4e86-ab6f-828ad68fa7d6") }
MongoDB server version: 4.4.3
```

- Express Version 4.17.1

```
C:\Users\leone\Documents\Proyecto BD2\Server>npm list express
server@1.0.0 C:\Users\leone\Documents\Proyecto BD2\Server
`-- express@4.17.1
```

- Angular Version 11.1.1

```
Angular CLI: 11.1.2
Node: 14.15.1
OS: win32 x64

Angular: 11.1.1
... animations, common, compiler, compiler-cli, core, forms
... platform-browser, platform-browser-dynamic, router
Ivy Workspace: Yes
```

- Node Js Version 14.15.1
- Bootstrap 4
- HTML 5

- CSS
- ODM: Mongoose

Colecciones

CLIENTES

```
{
  "_id": "604fef2a8cc82f21e0447582",
  "Nombre": "Tigo",
  "Patente": " ",
  "Usuario": "Tigo@gmail.com",
  "Contraseña": "123",
  "IP": "25.250.122.34",
  "Puerto": "5000",
  "Dias de entrega": "7",
}
```

DISPOSITIVOS

```
{
  "_id": "604fef2a8cc82f21e0447582",
  "Color": "Rojo",
  "Descripcion": " El mejor telefono",
  "Modelo": "iPhone 12",
  "Codigo_Modelo": "iPhone12qewqe",
  "Existencia": "10000",
  "Precio": "9000",
  "Ram": "8",
  "Almacenamiento": "128",
  "Procesador": "8",
  "numero_cores": "4",
  "Marca": "Apple",
}
```

HISTORIAL

```
{
  "_id": "604fef2a8cc82f21e0447582",
  "User": "Gio@gmail.com",
  "Accion": "Elimino pedido ",
  "Fecha": "2021-04-24T20:42:48.562Z",
}
```

PEDIDOS

```
{
  "_id": "604fef2a8cc82f21e0447582",  "user":
  {
    "60497ae46ee94f65f8ddfc24"
  }
  "cantidad": 12,
  "cliente": "Tigo",
  "total": 1000,
  "Numero_Serie": [
    {"num":
      "F-608451241287d93b2891628a/9010922416610"},
    {"num":
      "F-608451241287d93b2891628a/9010922416610"}
  ],
  "fecha_p": "2020-02-02T00:00:00.000Z",
  "fecha_e": "2020-02-02T00:00:00.000Z",
  "dispositivo": {
    "604a50f90e6cb52584437746",
  }
}
```

USERS

```
{  
  "_id": "604fef2a8cc82f21e0447582",  
  "Nombres": "Leonel"  
  "Email": " Leonel@gmail.com",  
  "Password": "q23er",  
  "Roles": "Admin",  
}
```

Aggregations

Aggregation #1

El aggregation #1 lo utilizamos para que el administrador pueda llegar a ver la cantidad de dispositivos que se han vendido en el día, teniendo en cuenta únicamente los que estén en estado empacado ya que consideramos que a la hora de estar empacados el dispositivo ya está vendido y listo para ser entregado.

AGGREGATION 1

```
{
  "$lookup": {
    "from": "dispositivos",
    "localField": "dispositivo",
    "foreignField": "_id",
    "as": "dispositivoCol"
  }
},
{
  "$match": {
    "estado": "Empacando"
  }
},
{
  "$group": {
    "_id": "$fecha_p",
    "Dispositivo": {
      "$push": "$dispositivoCol.modelo"
    },
    "cantidad": {
      "$sum": "$cantidad"
    }
  }
}
```

Aggregation #2

El aggregation #2 lo utilizamos para que el administrador pueda llegar a visualizar si el pedido fue entregado, también llegar a observar qué cliente fue el que realizó el pedido y si el pedido fue entregado en los días establecidos

AGGREGATION 2

```
{
  "$lookup": {
    "from": "clientes",
    "localField": "cliente",
    "foreignField": "_id",
    "as": "clienteCol"
  },
  {
    "$replaceRoot": {
      "newRoot": {
        "$mergeObjects": [
          {
            "$arrayElemAt": [
              "$clienteCol",
              0.0
            ]
          },
          "$$ROOT"
        ]
      }
    }
  },
  {
    "$match": {
      "estado": "Entregado"
    }
  }
}
```

Herramientas extras para el desarrollo

- Visual studio code
- Oracle SQL Developer
- Studio 3t
- Github
- GitLens
- Oracle VM Virtual Box
- Postman