**DOCUMENTO DE DISEÑO TECNICO**

**Talle final semillero .NET**

**Presentado por: Santiago Bedoya Ossa**

**Armenia, Quindío.**

**2019**

Contenido

[**1.** **Introducción.** 3](#_Toc2982465)

[**2.** **Arquitectura de diseño.** 3](#_Toc2982466)

[**3.** **Arquitectura de Información.** 3](#_Toc2982467)

[**4.** **Arquitectura de servidor.** 3](#_Toc2982468)

[**5.** **Arquitectura de entrega.** 3](#_Toc2982469)

1. **Introducción.**

El presente documente va a contener información referida a todo el contenido realizado en el taller final del semillero .NET que ofreció la empresa de software Heinsohn Business Technology. Aquí se detallara toda la información trabajada que va desde el tipo de tecnologías y lenguajes utilizados, hasta la entrega del software el cual va a contener todo los requerimientos expresados en el documento del Taller final semillero .NET.

1. **Arquitectura de diseño.**

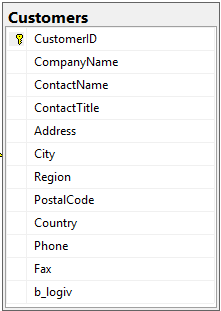
Para el desarrollo del Taller final se utilizó el framework de desarrollo .NET, el cual provee un extenso conjunto de soluciones para el desarrollar de soluciones. Para este caso concreto se utilizó una solución web Api la cual está orientada a la creación de servicios web y paginas web bajo un entorno de desarrollo de aplicaciones web llamado ASP.NET y está regida por protocolo de servicios REST.

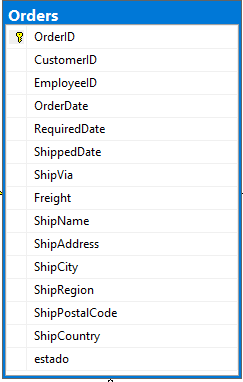
La solución web Api cuenta con una arquitectura similar al MVC (Modelo, Vista, Controlador) la cual es llamada code-behind en donde permite la separación de componentes los cuales serán explicados a continuación:

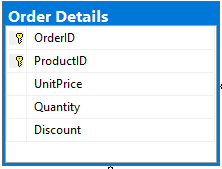
* **Modelo:** Hace referencia a los datos que se van a manejar en la aplicación y que se encuentran alojados en un base de datos desarrollada en SQL server.
* **Vista:** Hace referencia a toda la parte visual la cual se encuentra desarrollada en un framework Front-End llamado Angular el cual maneja un lenguaje de marcado HTML, un lenguaje de estilos CSS y un lenguaje de programación TypeScript el cual es un supe conjunto del lenguaje de programación JavaScript y que es soportado por Microsoft.
* **Controlador:** Hace referencia a la parte lógica en donde se maneja todo lo relacionado con la trasferencia de información entre el modelo y la vista se encuentra creada con el lenguaje de programación C#.

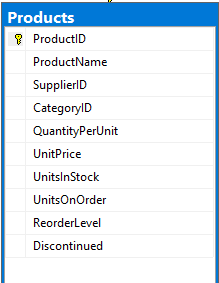
1. **Arquitectura de Información.**

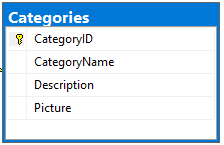
La aplicación web cuenta con una base de datos llamada Northwind que es ofrecida por Microsoft y la cual provee los datos utilizado en esta aplicación. De esta base de datos solo se van a utilizar las tablas Customers, Orders, Orders Details, Products y Categories, las cuales contienen información relevante para el desarrollo del Taller final semillero .net, a continuación se expone la información de cada una de las tablas.



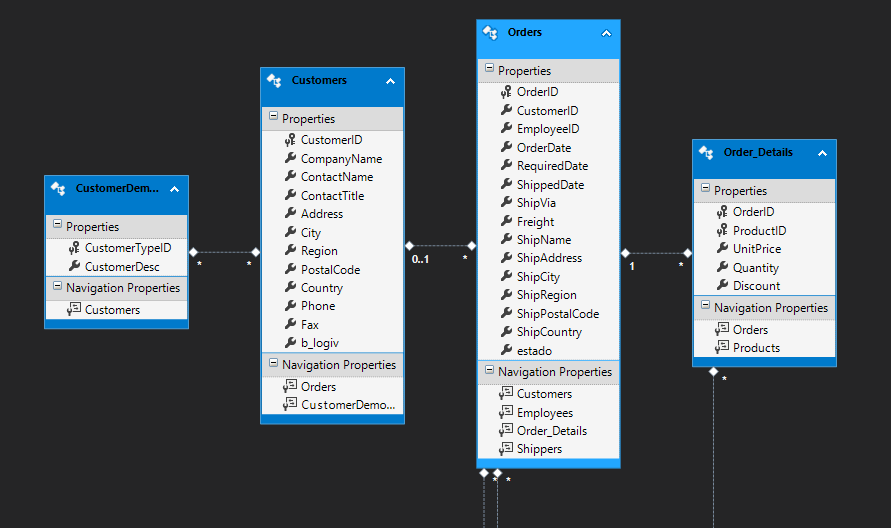




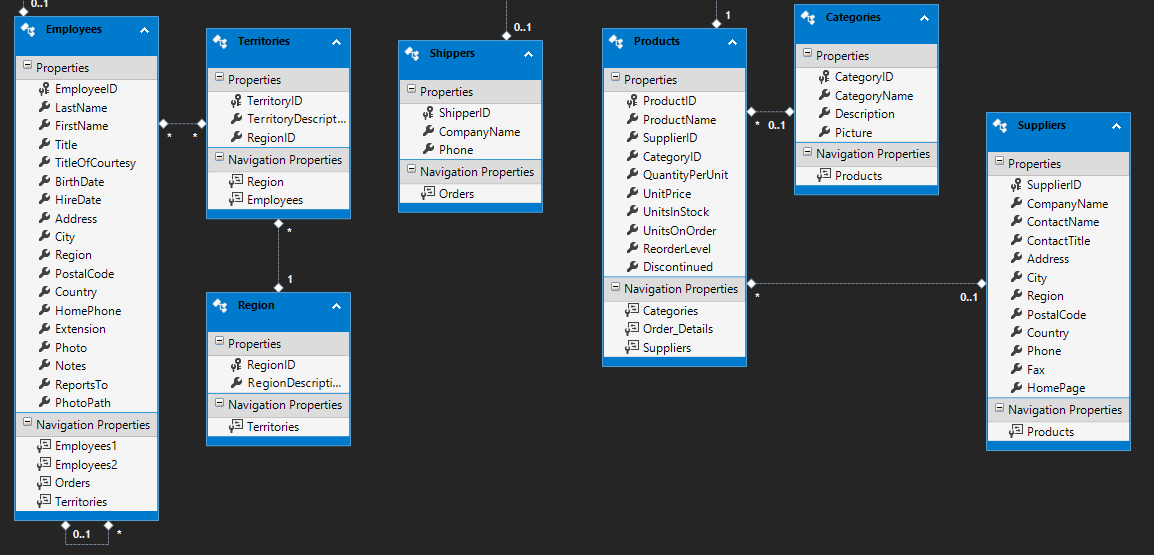




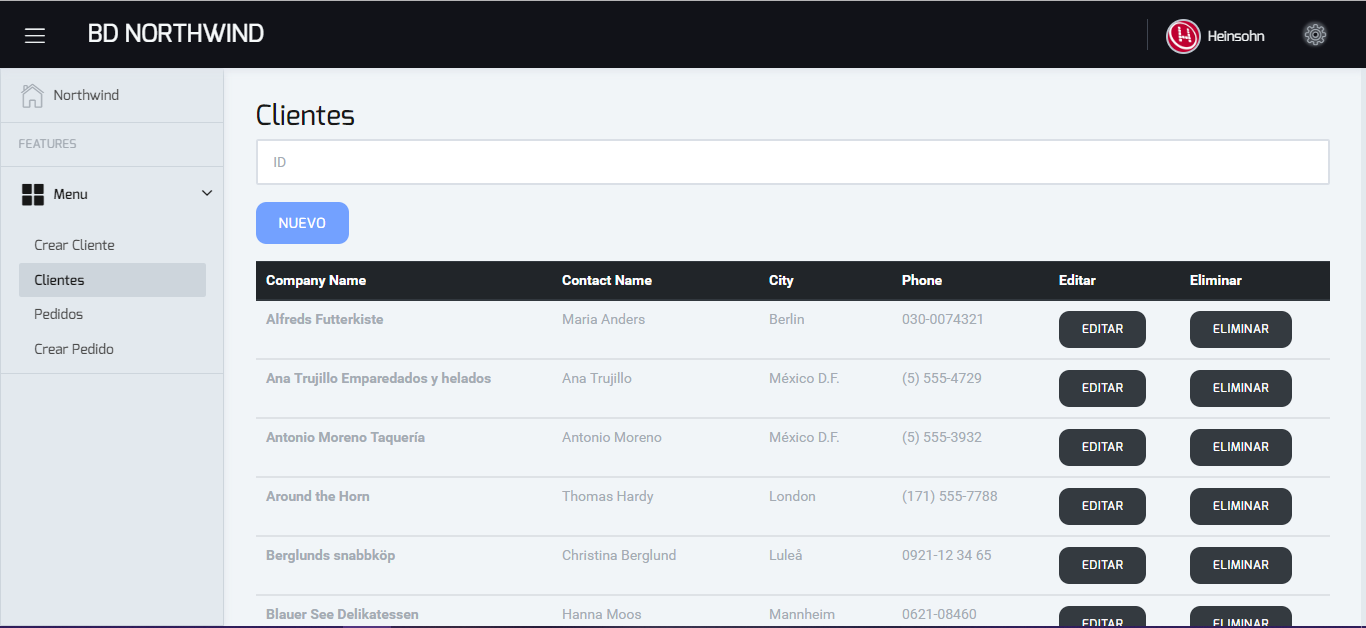
A continuación se mostrara las relaciones de estas tablas

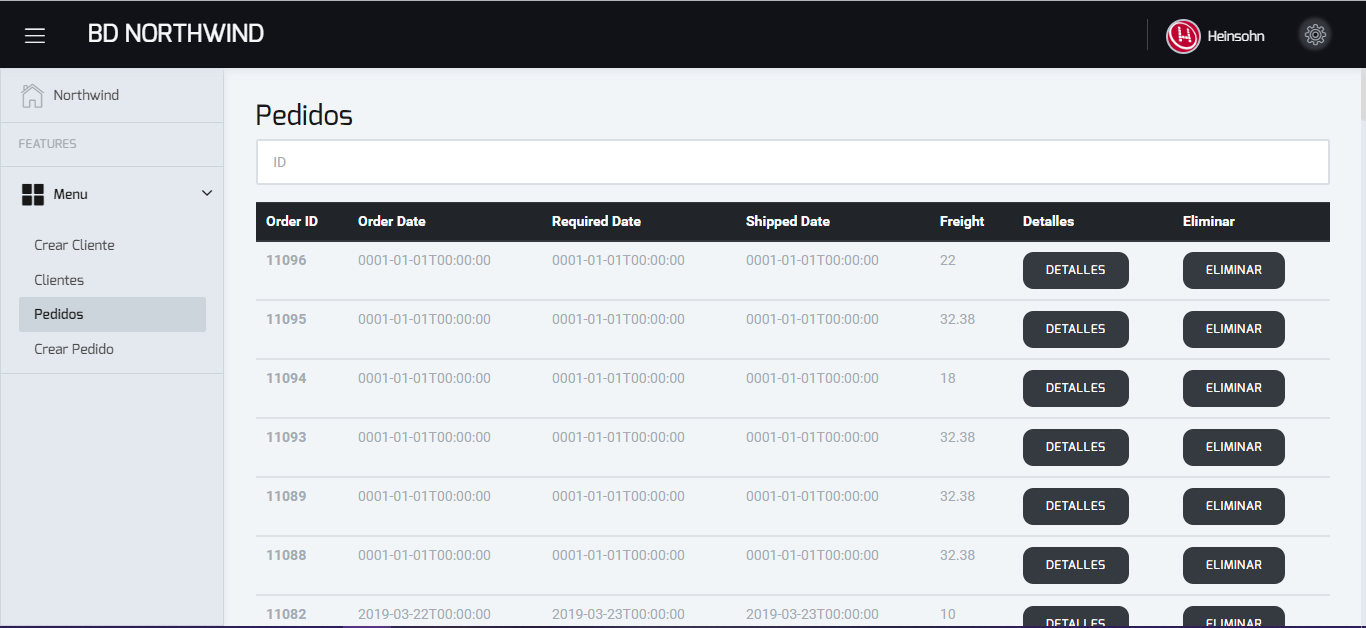
Parte 1

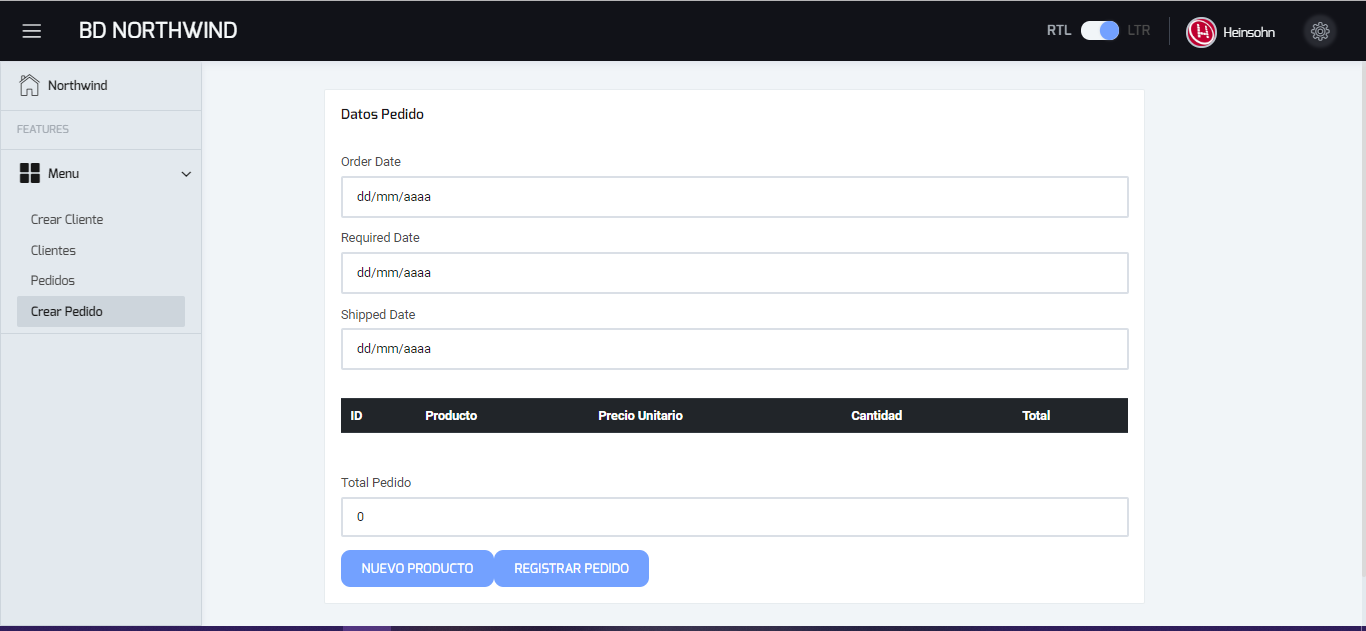
Parte2

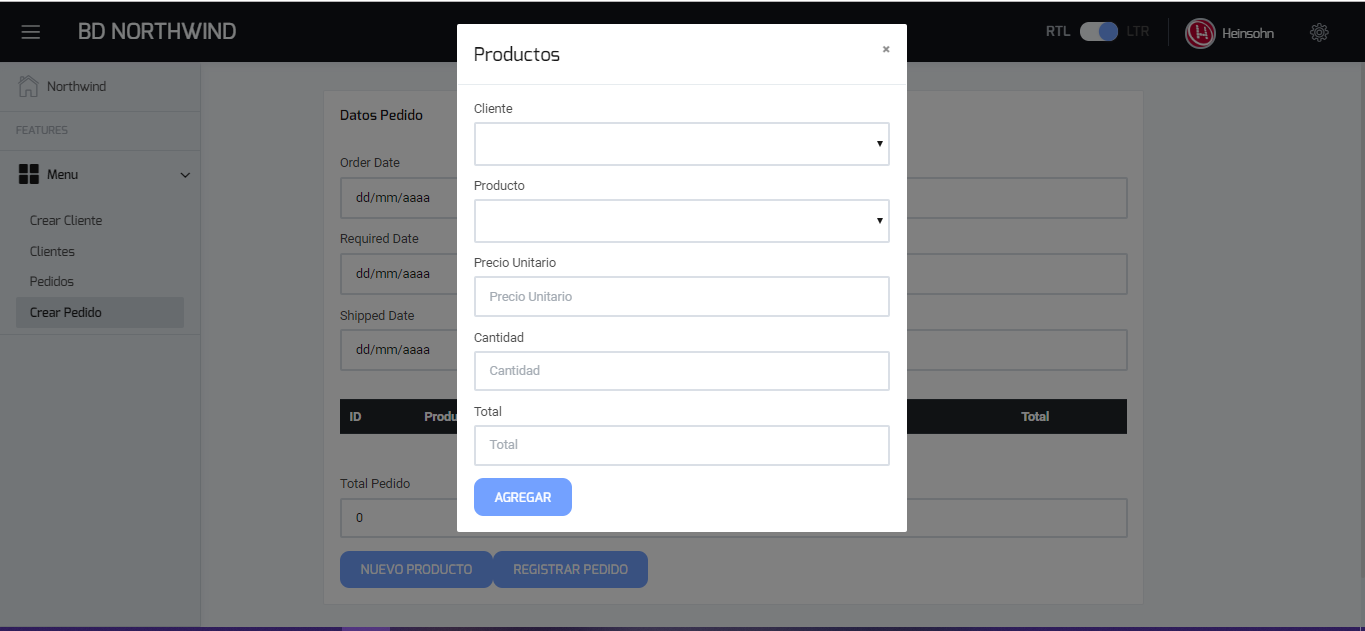


La aplicación va a contar con las pantallas de ver Clientes, ver Productos, ver Categorías y la creación de un pedido. A continuación se exponen cada una de las pantallas mencionadas anteriormente.









1. **Arquitectura de servidor.**

La aplicación desarrollado en el Taller final semillero .NET va a contar con un servidor local, el cual es proveído por el ordenador perteneciente a la persona que está presentando este documento.

El servidor expondrá unos servicios que va ser consumidos bajo el protocolo REST los cuales va a tener las siguientes rutas con su respectivo verbo Http que consumirá el servicio:

* Category

GET

* Customer

GET

GET/ID

POST

PUT

DELETE

* Order

GET

GET/ID

POST

DELETE

* Product

GET

1. **Arquitectura de entrega.**

La aplicación contara con una serie de métodos los cuales van a ser el puente para transmitir y recibir la información que se manejara. A continuación se expondrá los controladores en donde se encuentra la lógica para hacer este tipo de puente de informa y la explicación de cada uno de los métodos pertenecientes a cada uno de estos controladores:

* Category

consultarCategorias()->Consulta la base de datos y devuelve las categorías obtenidas

* Customer

consultarClientes()->Consulta la base de datos y devuelve todos los clientes obtenidos

obtenerClientePorID(string id)->Consulta a la base de datos y me devuelva un cliente en especifico

eliminarCliente(string id)-> Consulta a la base de datos y elimina (cambio de estado en una columna en la base de datos) un cliente en específico.

* Order

consultarPedidos()->Consulta la base de datos y devuelve los pedidos obtenidas.

obtenerPedidoPorID(int id)-> Consulta a la base de datos y me devuelva un pedido en específico.

eliminarPedido(int id)-> )-> Consulta a la base de datos y elimina (cambio de estado en una columna en la base de datos) un pedido en específico.

insertarPedido(Orders order)-> Se genera un pedido obtenido los datos de la vista y creando un consulta para insertarlo en la base de datos.

* Product

getProductos()->Consulta la base de datos y devuelve los productos obtenidos.