**Nombre:** Jonathan Joel Anchundia Navas.

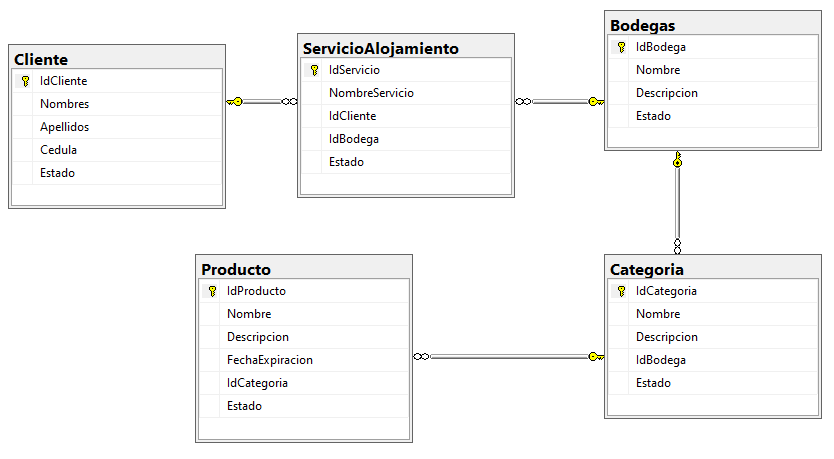
**Empresa:** Corporación Kruger.

**Diseño de Clases:**

Se crea Base de datos Bodega, la cual tiene las siguientes tablas:

1. Bodegas.
2. Categoría.
3. Productos.
4. Clientes.
5. ServicioAlojamiento.

**Diagrama de Base de Datos**



**DICCIONARIO DE DATOS.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de tabla:** Bodegas  **Descripción:** Datos de Bodega | | | |
| **Campo** | **Tipo** | **Longitud** | **Descripción** |
| IdBodega | Int | 11 | Clave Primaria |
| Nombre | Varchar | 50 | Nombre de la Bodega |
| Descripción | Varchar | 50 | Descripción de la Bodega. |
| Estado | Int |  | Estado 1 o 0 para eliminar Bodega. |
| **Nombre de tabla:** Categoría  **Descripción:** Datos de Categoría | | | |
| **Campo** | **Tipo** | **Longitud** | **Descripción** |
| IdCategoria | Int | 11 | Clave Primaria |
| IdBodega | Int | 11 | Clave Foranea |
| Nombre | Varchar | 50 | Nombre de la Categoría. |
| Descripción | Varchar | 50 | Descripción de la Categoría. |
| Estado | Int |  | Estado 1 o 0 para eliminar Categoría. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de tabla:** Cliente  **Descripción:** Datos del Cliente | | | |
| **Campo** | **Tipo** | **Longitud** | **Descripción** |
| IdCliente | Int | 11 | Clave Primaria |
| Nombre | Varchar | 50 | Nombre de la Bodega |
| Apellidos | Varchar | 50 | Descripción de la Bodega. |
| Cedula | Varchar | 50 | Cedula del cliente |
| Estado | Int |  | Estado 1 o 0 para eliminar Bodega. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de tabla:** Producto  **Descripción:** Datos del Producto | | | |
| **Campo** | **Tipo** | **Longitud** | **Descripción** |
| IdProducto | Int | 11 | Clave Primaria. |
| IdCategoria | Int | 11 | Clave foránea para relacionar con Tbl Categoría |
| Nombre | Varchar | 50 | Nombre del producto. |
| Descripción | Varchar | 50 | Descripción del producto. |
| FechaExpiracion | DateTime |  | Fecha de Expiración del Producto |
| Estado | Int |  | Estado 1 o 0 para eliminar Producto. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de tabla:** ServicioAlojamiento  **Descripción:** Datos de servicios de bodegas | | | |
| **Campo** | **Tipo** | **Longitud** | **Descripción** |
| IdServicio | Int | 11 | Clave Primaria. |
| IdCliente | Int | 11 | Clave foránea para relacionar con Tbl Cliente. |
| IdBodega | Int | 11 | Clave foránea para relacionar con Tbl Bodega. |
| NombreServicio | Varchar | 50 | Nombre del Servicio. |
| Descripción | Varchar | 50 | Descripción del producto. |
| FechaExpiracion | DateTime |  | Fecha de Expiración del Producto |
| Estado | Int |  | Estado 1 o 0 para eliminar Producto. |

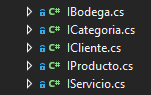
**Proyecto Api**

Se creó el proyecto en net Core 3.1, la cual tiene las siguientes bibliotecas de clases:

**DAO:** Contiene 2 carpetas que son:

**Datos:** Es nuestro Contexto de base de datos, y es la encargada de hacer la interacción con la misma (Crud). Tiene el Contexto de la base de datos:

**Services:** En esta carpeta se va a encontrar todas nuestras interfaces para inyectarle a cada clase de la carpeta DAO, las interfaces son:



**DAO:** Esta carpeta seria nuestro repositorio para manejar los Crud e interacción a la base de datos, por medio de la inyección de nuestro contexto a cada clase.

**Clases que contiene esta carpeta:**

**BodegaDAO:** Se compone de los siguientes métodos:

void GuardarBodega(Entidades.Bodega model); - Este método es el encargado de Registrar una Bodega a la Base de Datos.

void ActualizarBodega(Bodega model); - Método encargado de Actualizar una Bodega en Especifico.

void EliminarBodega(int idBodega); - Método encargado de Eliminar una Bodega en Especifico.

List<Bodega> ConsultarBodegas(); - Método encargado de consultar todas las Bodegas.

List<SelectListItem> ListaBodega(); - Método que va a servir para desplegar una lista de bodegas en un drowdoplist.

**CategoriaDAO:** Se compone de los siguientes métodos:

void GuardarCategoria(Entidades.Bodega model); - Este método es el encargado de Registrar una Categoría a la Base de Datos.

void ActualizarCategoria(Categorias model); - Método encargado de Actualizar una Bodega en Especifico.

void EliminarBodega(int idCategoria); - Método encargado de Eliminar una Categoría en Especifica.

List<Categorias> ConsultarBodegas(); - Método encargado de consultar todas las Categorias.

List<SelectListItem> ListaCategoria(); - Método que va a servir para desplegar una lista de Categorias en un drowdoplist.

**ClienteDAO:** Se compone de los siguientes métodos:

void GuardarCliente(Entidades.Clientes model); - Este método es el encargado de Registrar un Cliente a la Base de Datos.

void ActualizarCliente(Clientes model); - Método encargado de Actualizar un Cliente en Especifico.

void EliminarCliente(int idCliente); - Método encargado de Eliminar un Cliente en Especifico.

List<Categorias> ConsultarClientes(); - Método encargado de consultar todos los Clientes.

List<SelectListItem> ListaCliente(); - Método que va a servir para desplegar una lista de Clientes en un drowdoplist.

**ProductoDAO:** Se compone de los siguientes métodos:

void GuardarProducto(Entidades.Productos model); - Este método es el encargado de Registrar un Producto a la Base de Datos.

void ActualizarProducto(Productos model); - Método encargado de Actualizar un Producto en Especifico.

void EliminarProducto(int idProducto); - Método encargado de Eliminar un Producto en Especifico.

List<Productos>ConsultarProductos(); - Método encargado de consultar todos los Producto.

List<SelectListItem> ListaProductos(); - Método que va a servir para desplegar una lista de Productos en un drowdoplist.

**ServicioDAO:** Se compone de los siguientes métodos:

void GuardarServicio(Entidades.Servicio model); - Este método es el encargado de Registrar un Servicio a la Base de Datos.

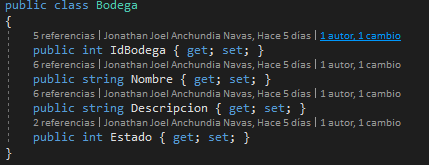
List<Servicio>ConsultarServicios(); - Método encargado de consultar todos los Servicios.

Servicio ConsultarReporte(int idCliente, DateTime fecha); - Método que va a servir para consultar datos de productos de los clientes.

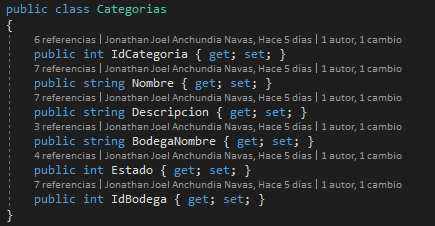
**Biblioteca Entidades:**

Esta Biblioteca contiene las clases que va a permitir mapear el Context de la base de datos con cada clase, contiene los mismos atributos (Variables-Columnas) de cada tabla de la base de datos, y son las siguientes:

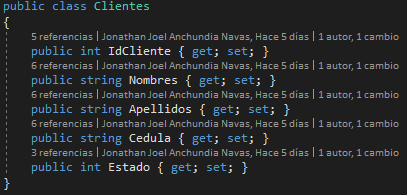
**Bodega:**



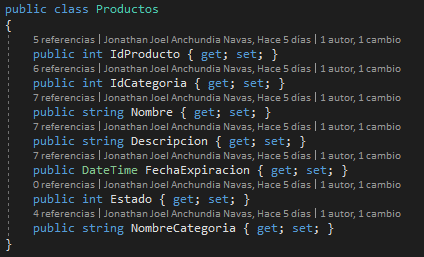
**Categorias:**



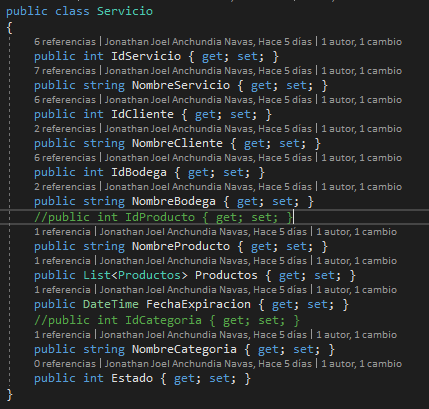
**Cliente:**



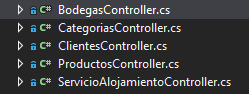
**Productos:**



**Servicio:**

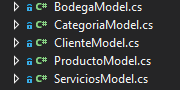


Y en el proyecto principal Api tenemos los controladores que son los siguientes:

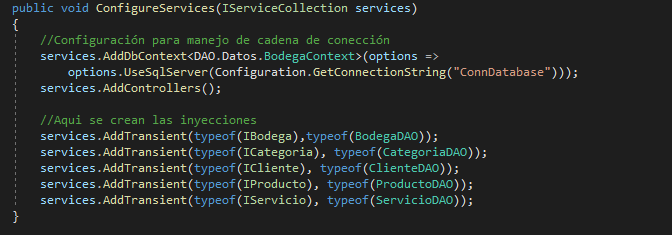


Cada controlador tiene las funcionalidades del proyecto DAO, de la carpeta DAO que es nuestro repositorio, las implementa mediante inyección de dependencia según sea el controlador tendrá inyectada la interfaz y la clase Context que es para manipular la base de datos.

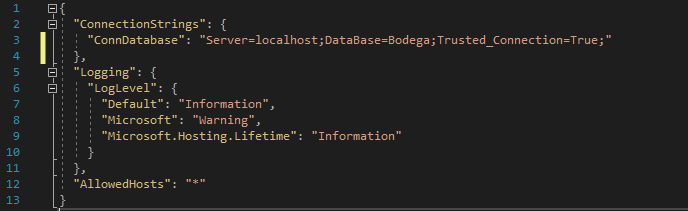
También disponemos de la carpeta Models, que básicamente son nuestros modelos:



Por último, tenemos la clase startup, en esta clase se modifica el método que es la que inyecta las dependencias a los controladores, le pasamos la interfaz y los métodos DAO que son los repositorios.



Se modifico el archivo de configuración de nuestra base de datos:



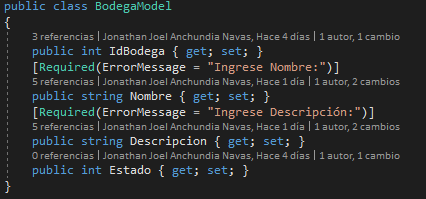
**Proyecto WebBodega**

Este es un proyecto realizado en Net Core con Razor Page, está compuesto por la carpeta Models, Pages.

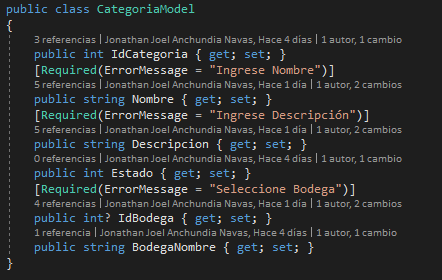
**Models:**

Esta carpeta contiene las clases modelos de nuestra aplicación, nos va a permitir enviar datos de nuestra aplicación a nuestra Api, contiene los mismos atributos (Variables-Columnas) de cada tabla de la base de datos, y son las siguientes:

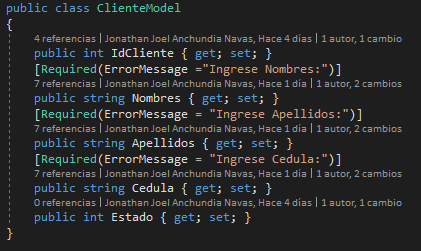
**BodegaModel:**



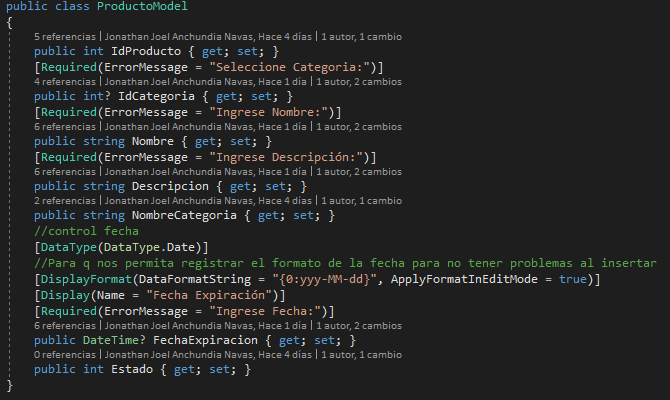
**CategoriaModel:**



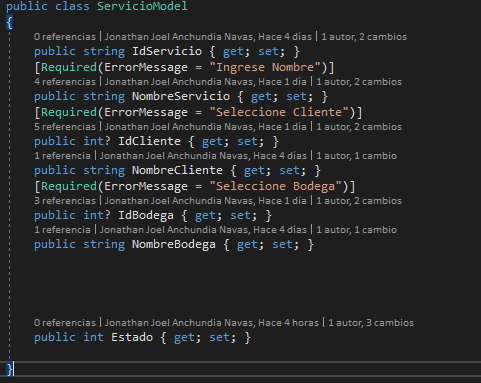
**ClienteModel:**



**ProductoModel:**



**ServicioModel:**



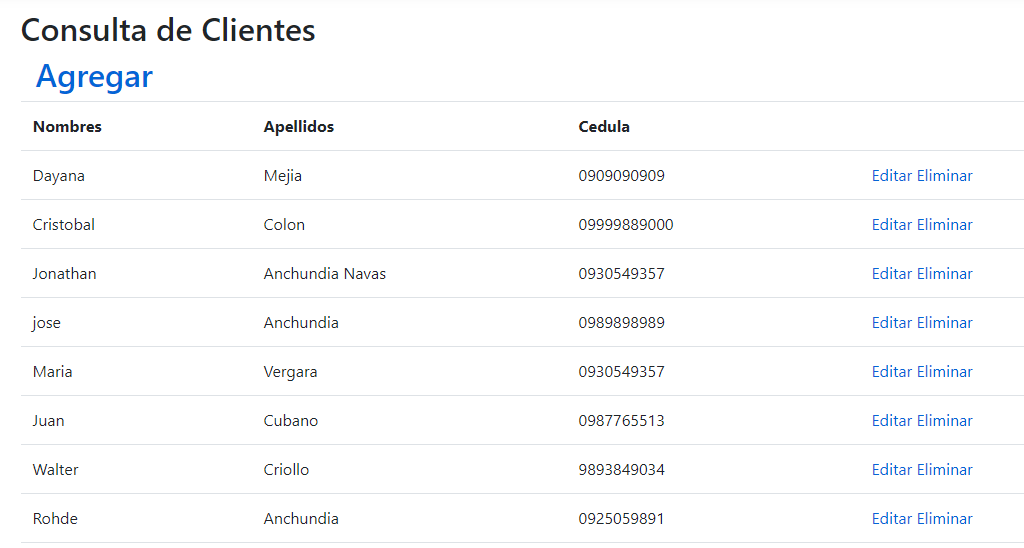
**Pages:**

Carpeta que contiene las vistas de nuestro programa y la interacción con ellas, las carpetas que tnemos son las siguientes:

**Clientes:**

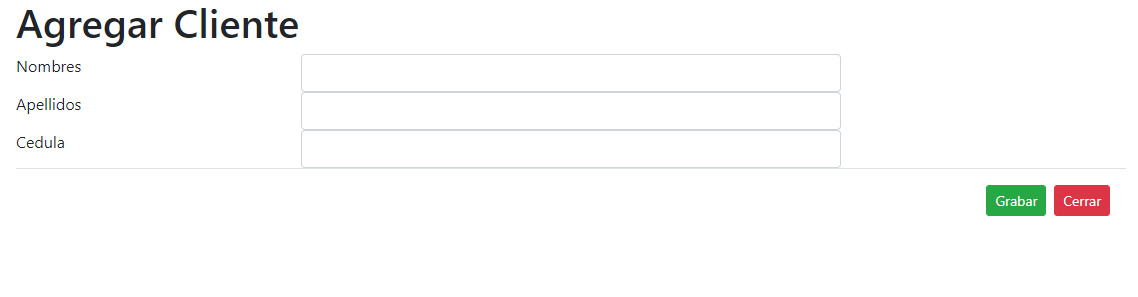


En nuestra aplicación se verá de la siguiente manera:

**IndexCliente:**

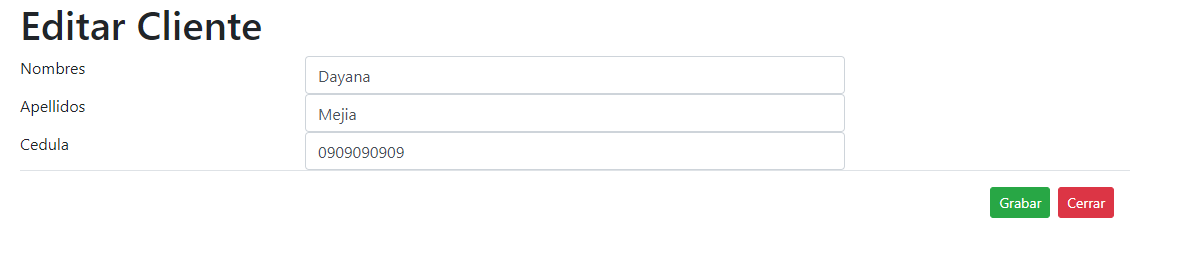
**AgregarCliente:**

Al dar clic en el botón **Agregar** nos mostrará una pantalla donde podemos agegar un cliente.



**EditarCliente:**

Al dar clic en editar nos mostrará una pantalla donde al seleccionar un cliente podemos editar su informacion:



Posteriormente al dar clic en **eliminar** se eliminará el cliente seleccionado.

**Bodegas:**

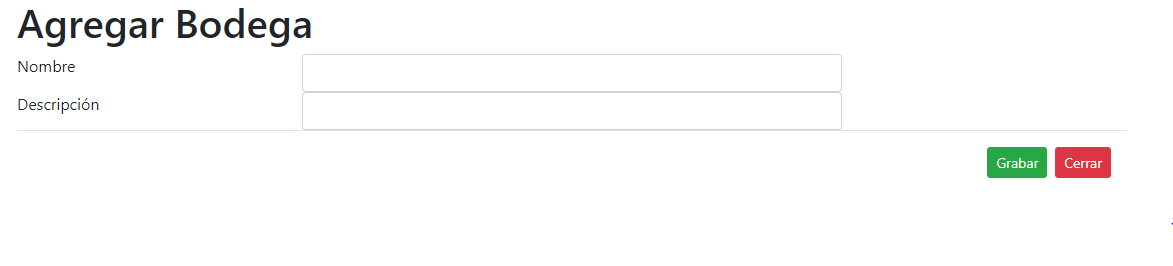


En nuestra aplicación se verá de la siguiente manera:

**IndexBodega:** 

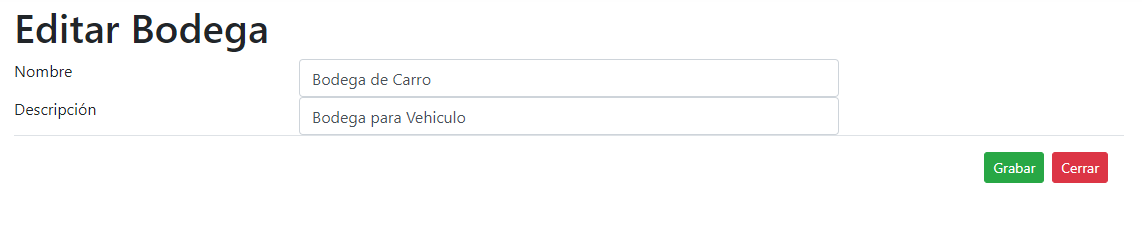
**AgregarBodega:**

Al dar clic en el botón **Agregar** nos mostrará una pantalla donde podemos agegar una nueva Bodega.



**EditarBodega:**

Al dar clic en editar nos mostrará una pantalla donde al seleccionar un cliente podemos editar su informacion:

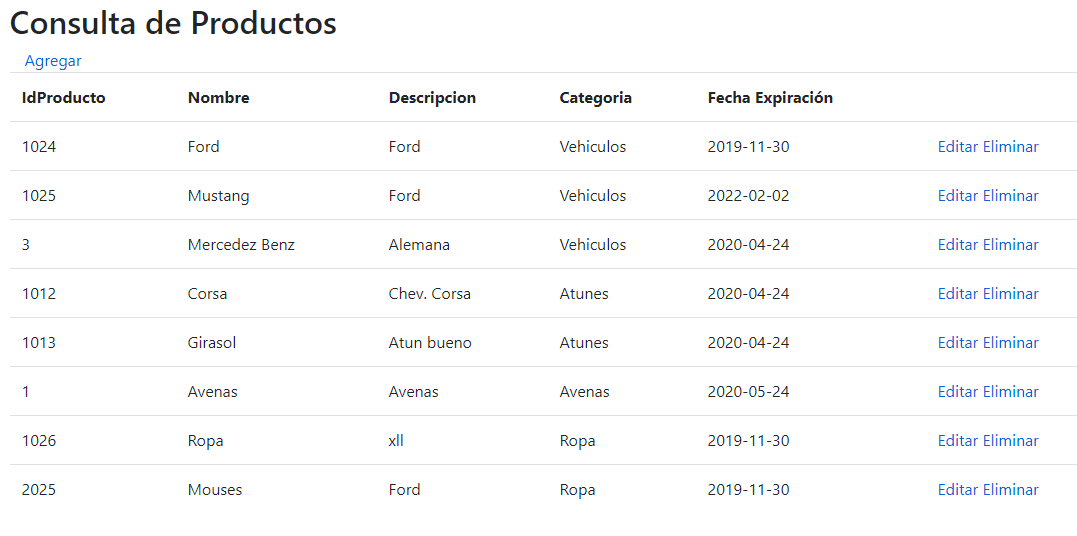


Posteriormente al dar clic en **eliminar** se eliminará la Bodega seleccionada.

**Productos:**

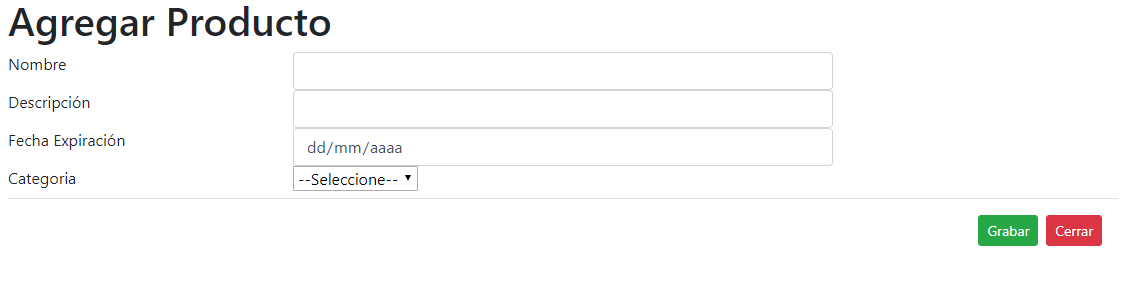


En nuestra aplicación se verá de la siguiente manera:

**IndexProducto:** 

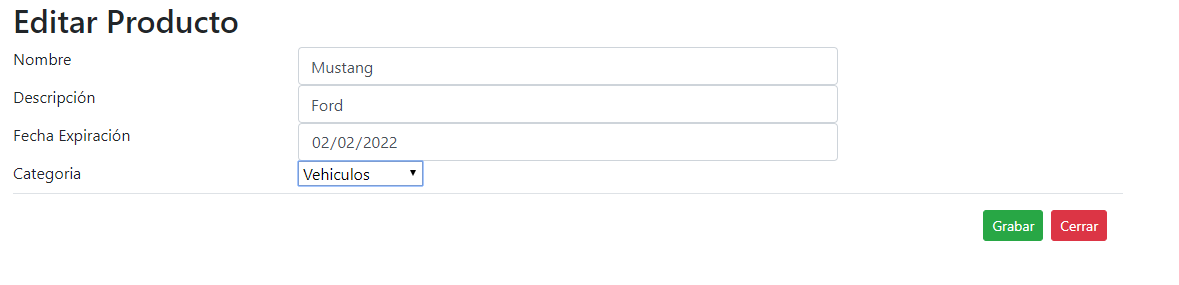
**AgregarProducto:**

Al dar clic en el botón **Agregar** nos mostrará una pantalla donde podemos agegar un nuevo Producto.



**EditarProducto:**

Al dar clic en editar nos mostrará una pantalla donde al seleccionar un Producto podemos editar su informacion:



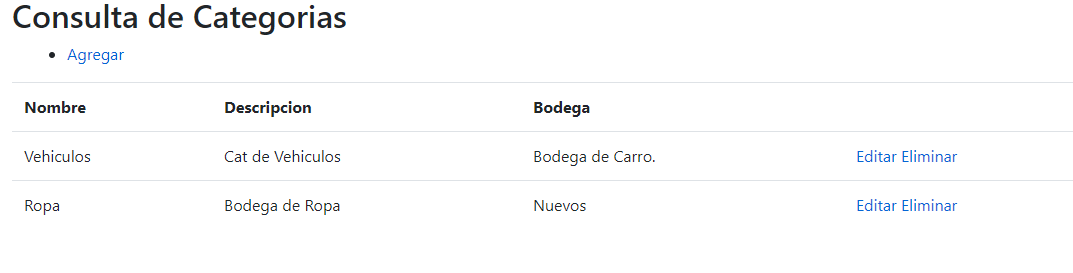
Posteriormente al dar clic en **eliminar** se eliminará el Producto seleccionado.

**Categoria:**



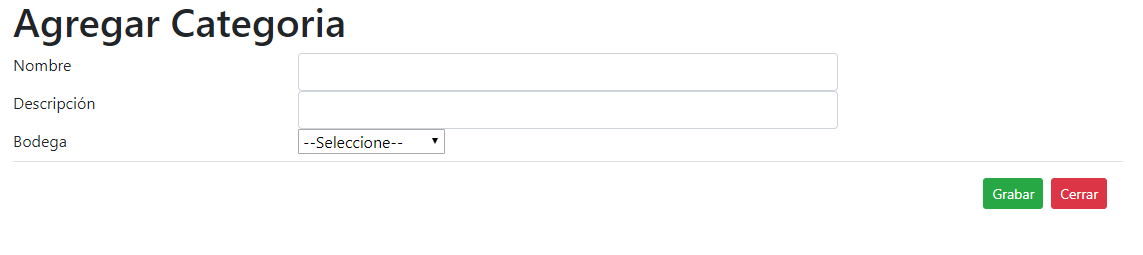
En nuestra aplicación se verá de la siguiente manera:

**IndexCategoria:**



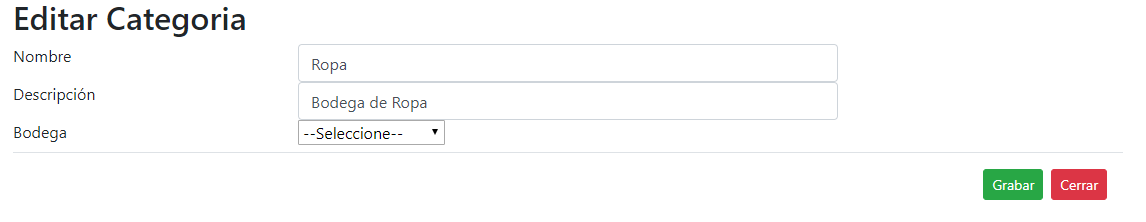
**AgregarCategoria:**

Al dar clic en el botón **Agregar** nos mostrará una pantalla donde podemos agegar una nueva Categoria.



**EditarCategoria:**

Al dar clic en **Editar** nos mostrará una pantalla donde al seleccionar una Categoria podemos editar su informacion:



Posteriormente al dar clic en **eliminar** se eliminará la Categoria seleccionada.

**ServicioAlojamiento:**



En nuestra aplicación se verá de la siguiente manera:

**IndexServicio:**



**AgregarServicio:**

Al dar clic en el botón **Agregar** nos mostrará una pantalla donde podemos agegar un nuevo Servicio a un Cliente.

