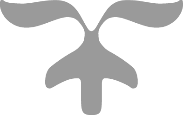


Instituto Tecnológico De Costa Rica



Curso: Bases de datos I

**Proyecto 2**

**CRM**

Profesor:

Ing. Allan Rodríguez Dávila

Estudiantes:

Adjany Gard Alpizar Carné: 2021117164

Kevin Salazar Valles Carné: 2021138547

Segundo semestre, 2022

Fecha de entrega: 02 Noviembre

Tabla de contenidos

[**Manual de usuario**](#_heading=h.gjdgxs) **2**

[Requerimientos:](#_heading=h.1ivlfmf12ef4) 2

[Modo de uso del prototipo:](#_heading=h.paz8j3ytwnwj) 2

[**Pruebas de funcionalidad**](#_heading=h.wm7kcypenal1) **6**

[Funcionalidades asociadas al gestor de productos](#_heading=h.jhn3w4i5dpui) 6

[Funcionalidades asociadas al gestor de clientes](#_heading=h.8hxn5jxwatfd) 7

[Funcionalidades asociadas al gestor de contactos](#_heading=h.jt17e4rwsq9q) 9

[Funcionalidades asociadas al gestor de cotizaciones](#_heading=h.mk0id7erxrq9) 11

[**Descripción del problema**](#_heading=h.md10yheicgxm) **13**

[**Diseño del programa**](#_heading=h.1vpsvtgw3df6) **14**

[Diagrama Base de datos.](#_heading=h.tyjcwt) 15

[**Librerías usadas**](#_heading=h.wcyg2rwhope7) **16**

[**Análisis de resultados**](#_heading=h.kufr6zg0jlvs) **16**

[Objetivos alcanzados](#_heading=h.emhm4lrgatv2) 16

[Objetivos no alcanzados](#_heading=h.4d34og8) 16

[**Bitácora.**](#_heading=h.d5ec2cu3zxcy) **16**

# Manual de usuario

## Requerimientos:

Para utilizar el prototipo es necesario que el equipo cuente con Visual Studio y SQL manager. Debemos tener la base de datos del CRM con todas las tablas y los procedimientos almacenados necesarios para el correcto funcionamiento de la aplicación.. Para la conexión de la base de datos y el programa es necesario tener la librería entity framework para cargar los modelos de la base al código fuente y la herramienta Entity framework tools para establecer la conexión con la base de datos mediante la consola de Visual Studio. Adicionalmente la librería data SQL client para poder ejecutar los procedimientos almacenados necesarios para el uso de la aplicación.

## Modo de uso del prototipo:

Primero debemos de descomprimir el archivo ZIP en donde viene el programa y después abrir el archivo de extensión SQL llamado crm para ejecutar el scrip que contiene la creación de la base de datos.

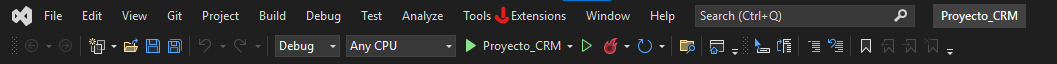


Para ejecutar el scrip debemos de darle al botón execute como se muestra en la siguiente imagen:



Una vez hecho esto se debe de abrir el programa ProyectoCRM en Visual Studio code, de ser necesario debemos de cambiar nuestro conecction string a la base de datos, este por defecto utiliza el server localhost y se especifica el nombre de la base por lo que en la mayoría de los casos no es necesario cambiarlo.

Una vez abierto el programa en Visual Studio le daremos run al botón verde que tiene como nombre el nombre del proyecto.

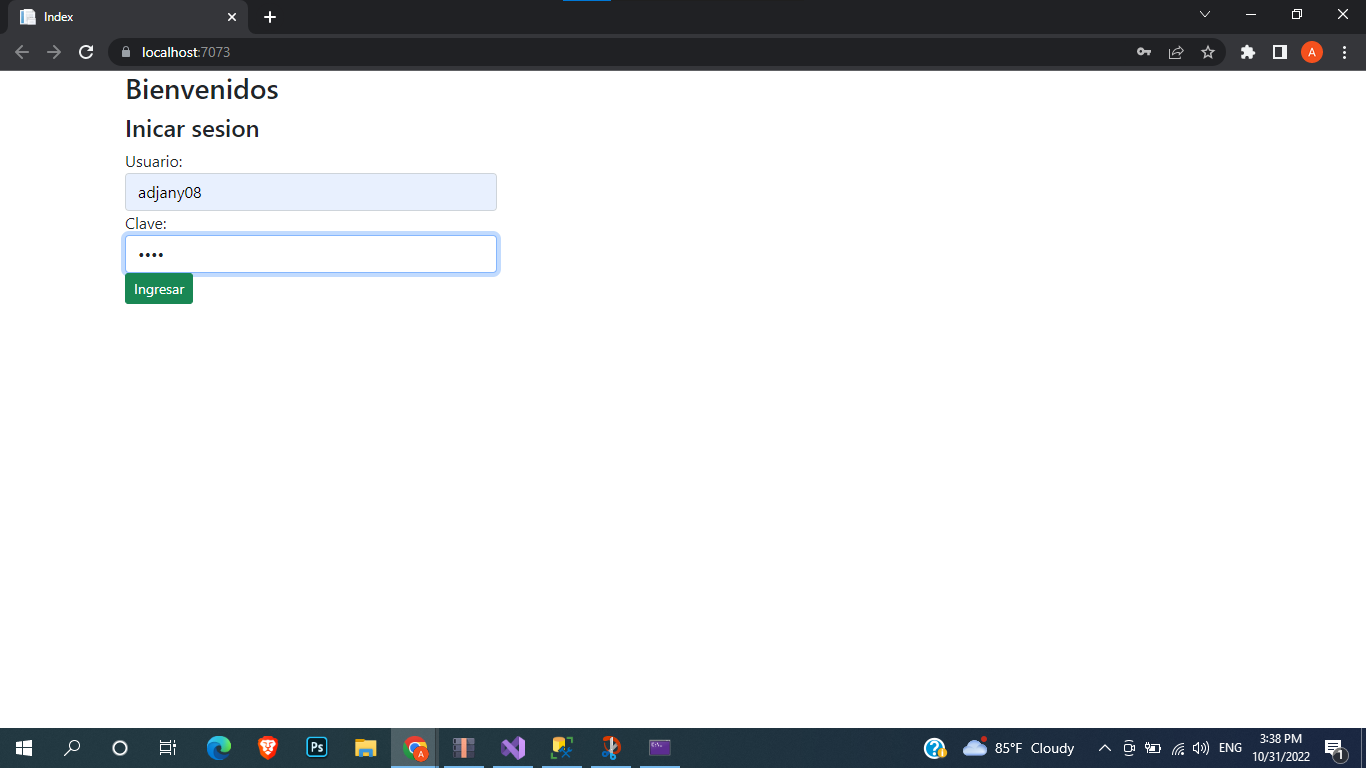


Una vez hecho esto se nos abrirá en nuestro navegador de preferencia la página de login en donde se autentica al usuario que está utilizando el programa.

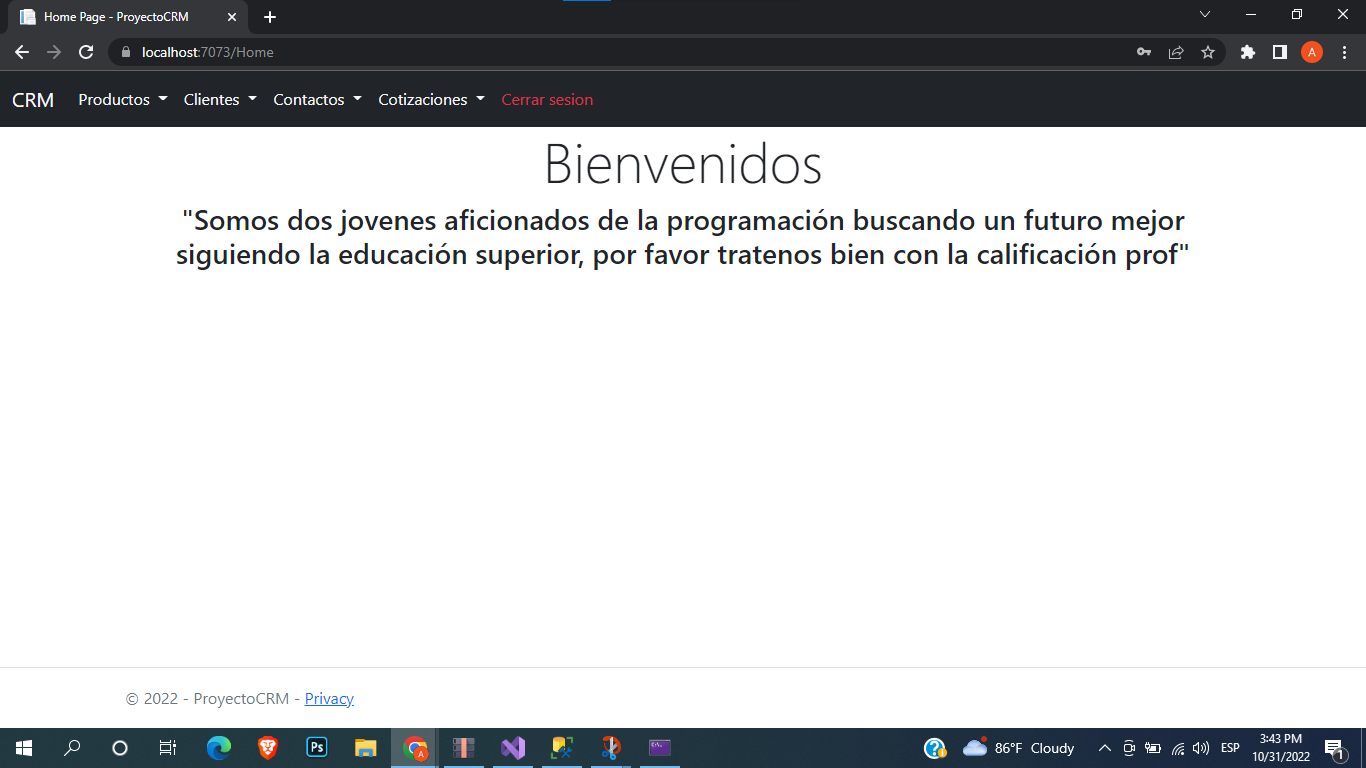
Para ingresar al sistema podemos utilizar las siguientes cuentas:

adjany08 / 1234

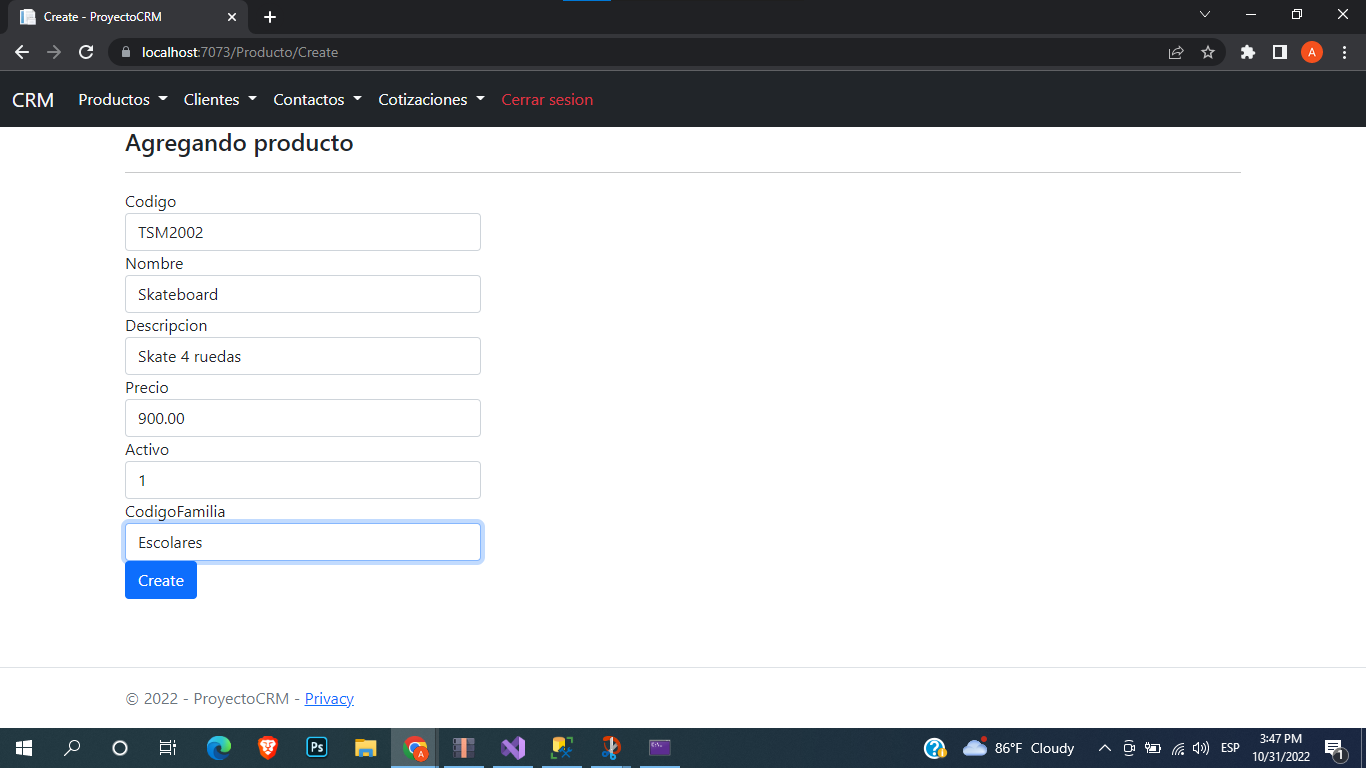
kevin / 1234



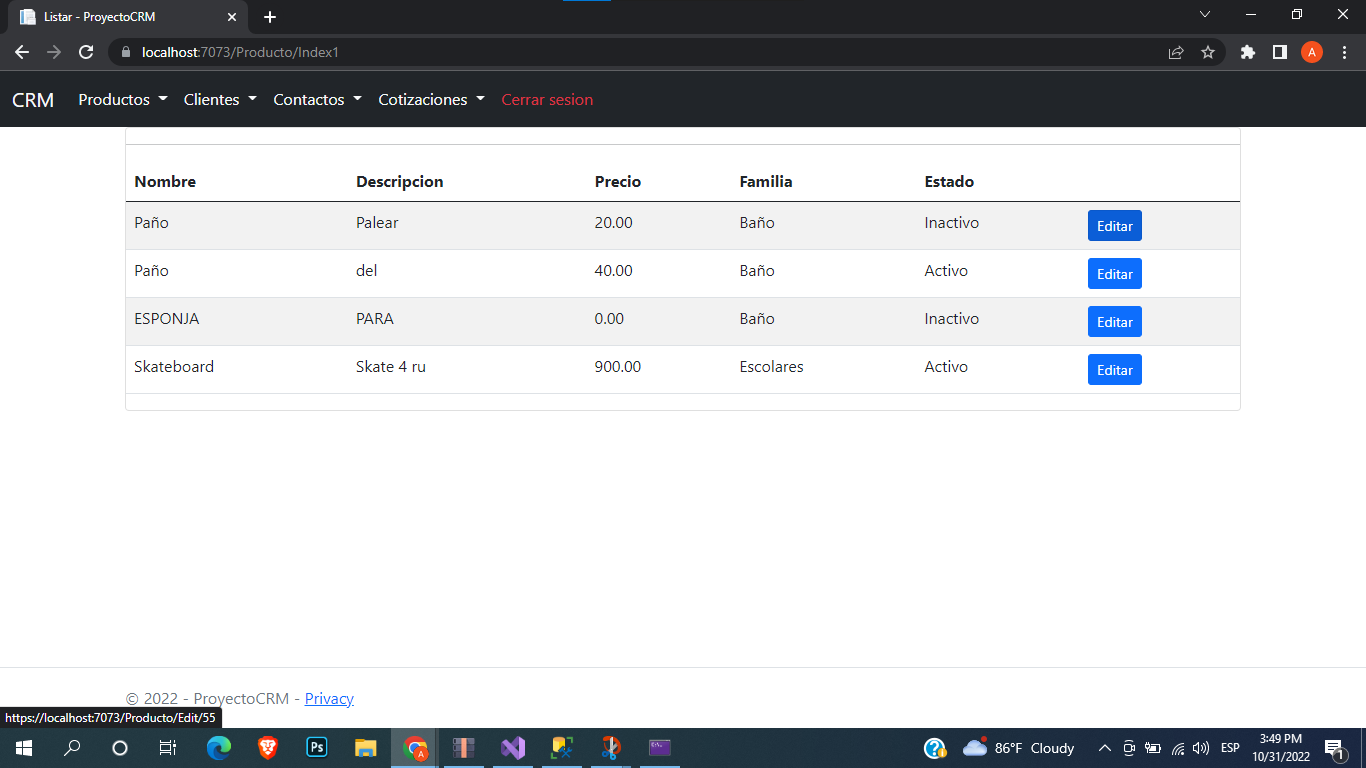
Si el usuario y la contraseña son correctos se nos va a redirigir a la página principal del programa en donde podemos realizar diferentes opciones.



Si deseamos utilizar las funciones de agregar solo debemos tocar el navbar en la sección en donde queremos agregar algo y se nos abrirá un menú en donde solo debemos llenar los campos necesarios. Para todos las funciones de agregar el proceso es el mismo.



Una vez hecho esto podemos observar utilizando las funciones de listar en el navbar los registros que han sido agregados. Si un registro no es agregado y la aplicación redirige al formulario significa que tiene el campo de primary key duplicado lo cual hace imposible insertar el registro en la base de datos ya que estos campos son únicos.



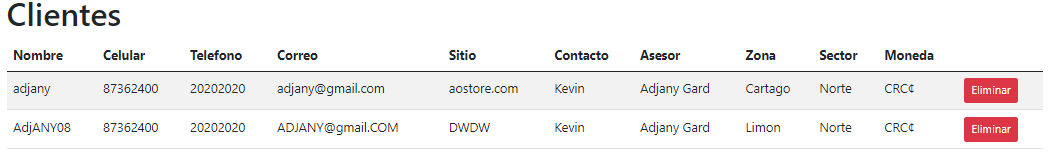
Y si queremos utilizar las funciones de editar le damos al botón azul. Aquí haremos los cambios que deseemos y podemos y se verán reflejados, para este ejemplo cambiaremos el estado del registro que acabamos de crear. Una vez editado se refleja el cambio.



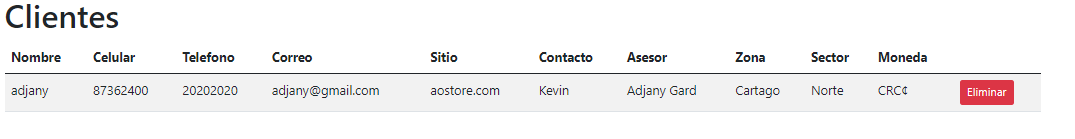
Para la funcionalidad de eliminar debemos seleccionar de la lista el registro que deseamos eliminar y después darle al botón rojo, esto nos redirigirá a una página en donde debemos darle eliminar definitivamente, si no existe alguna restricción el registro será eliminado placenteramente.

Eliminación de un cliente desde la aplicación.

Antes:



Después:



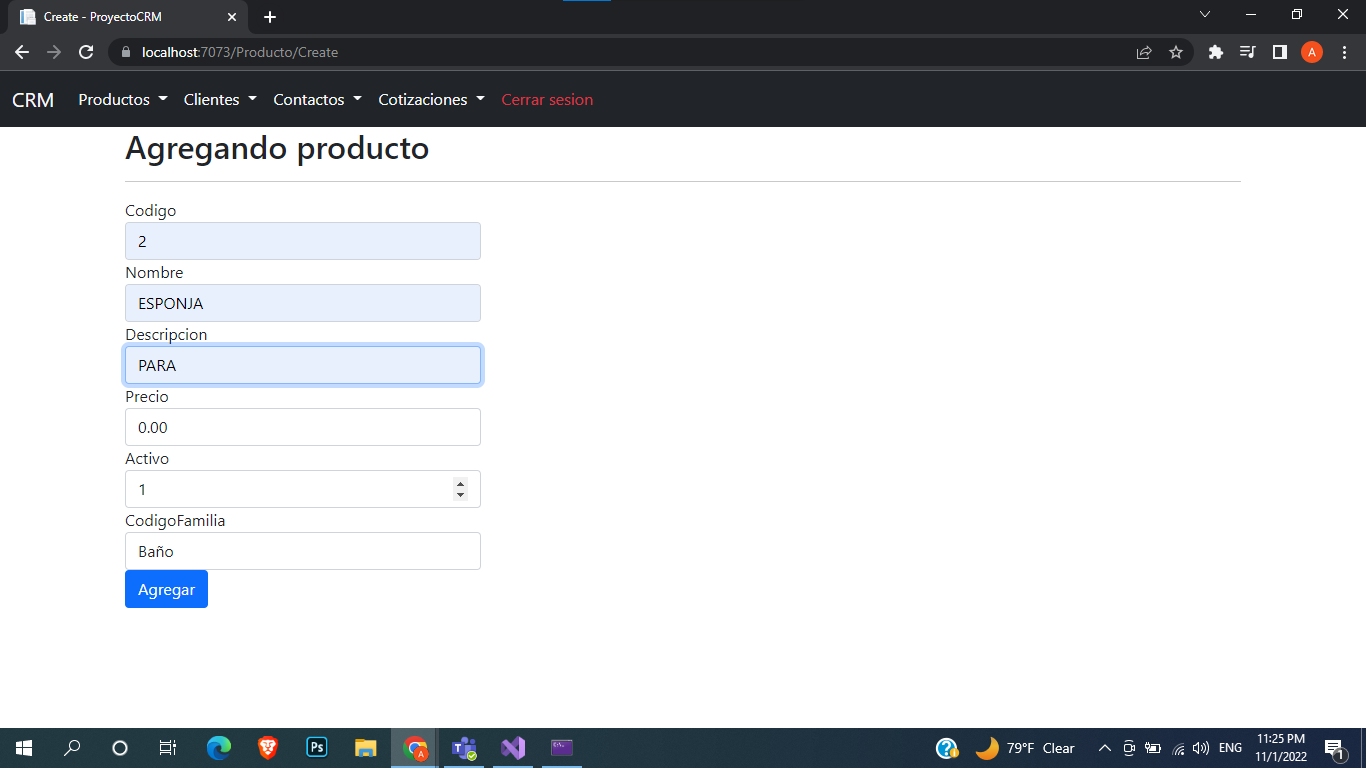
Si el producto no fue eliminado esto significa que estaba asociado con otro registro lo que hace imposible eliminarlo.

## 

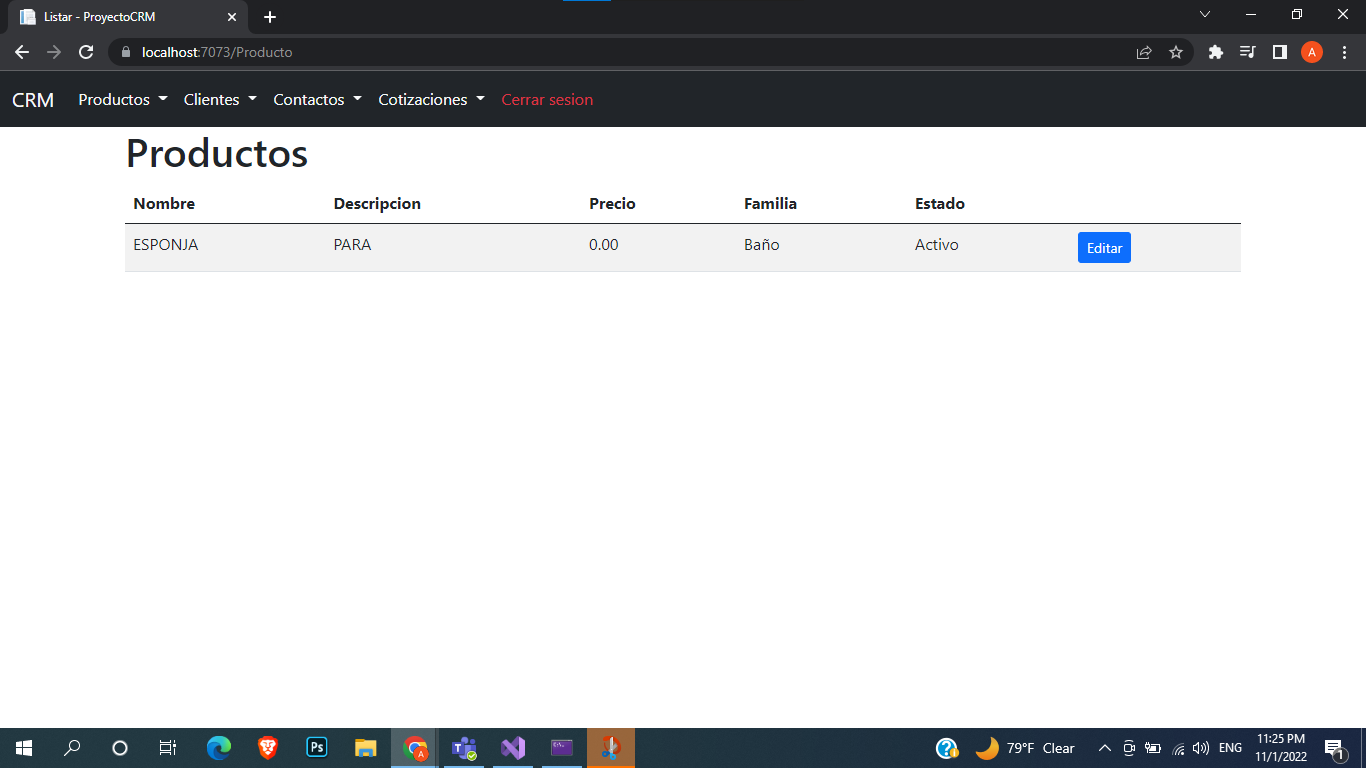
# Pruebas de funcionalidad

## Funcionalidades asociadas al gestor de productos

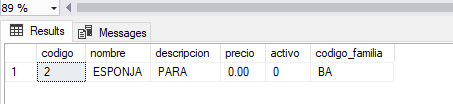
Formulario para agregar productos.



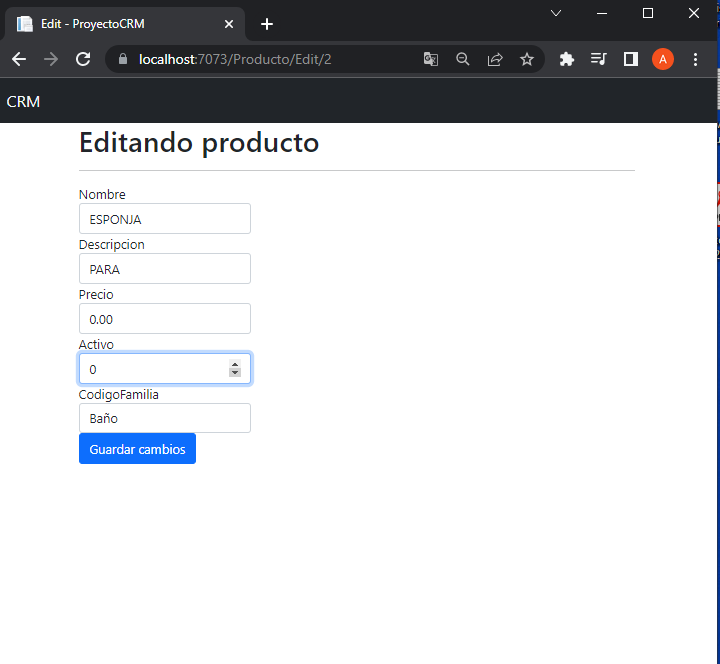
Visualización del producto recientemente agregado desde la aplicación.



Visualización desde la base de datos.



Formulario para la edición del producto.

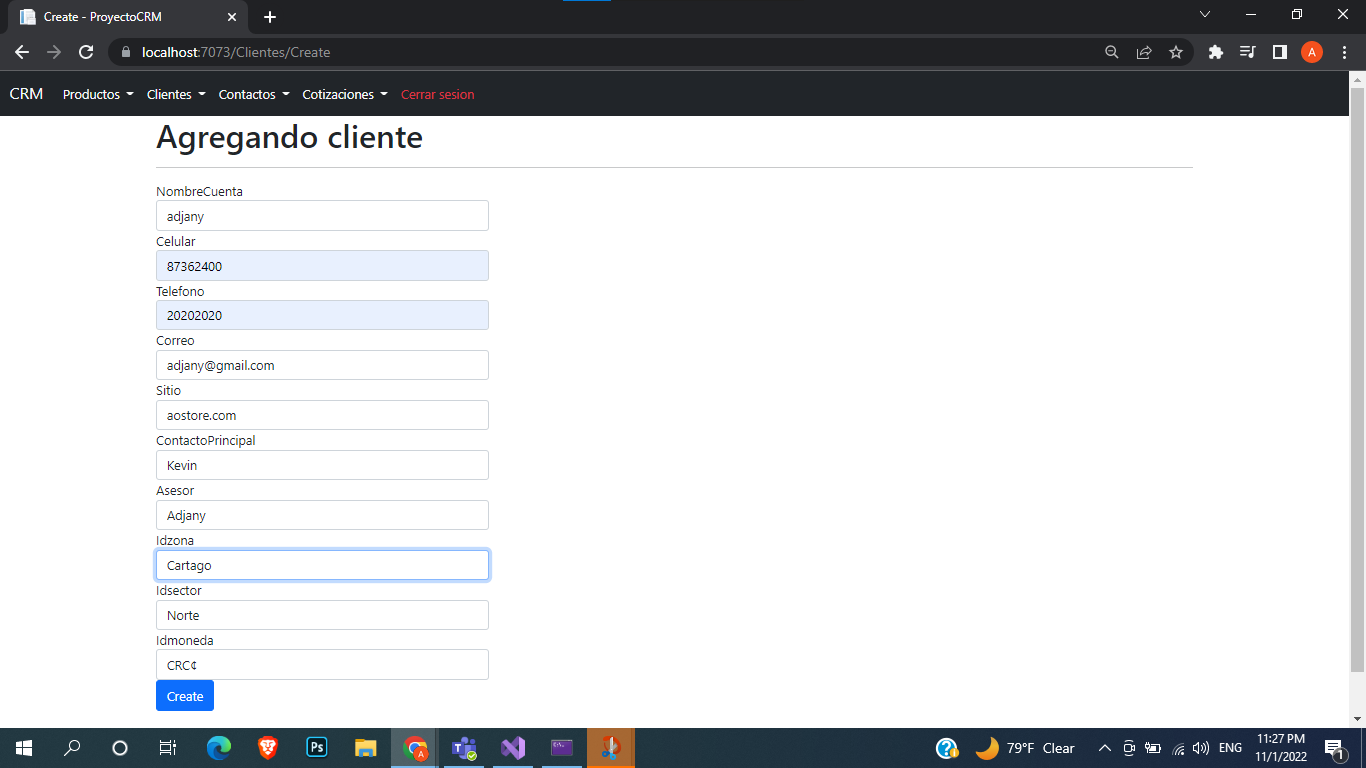


En el ejemplo anterior cambiamos el estado del producto y podemos ver este cambio al guardar.

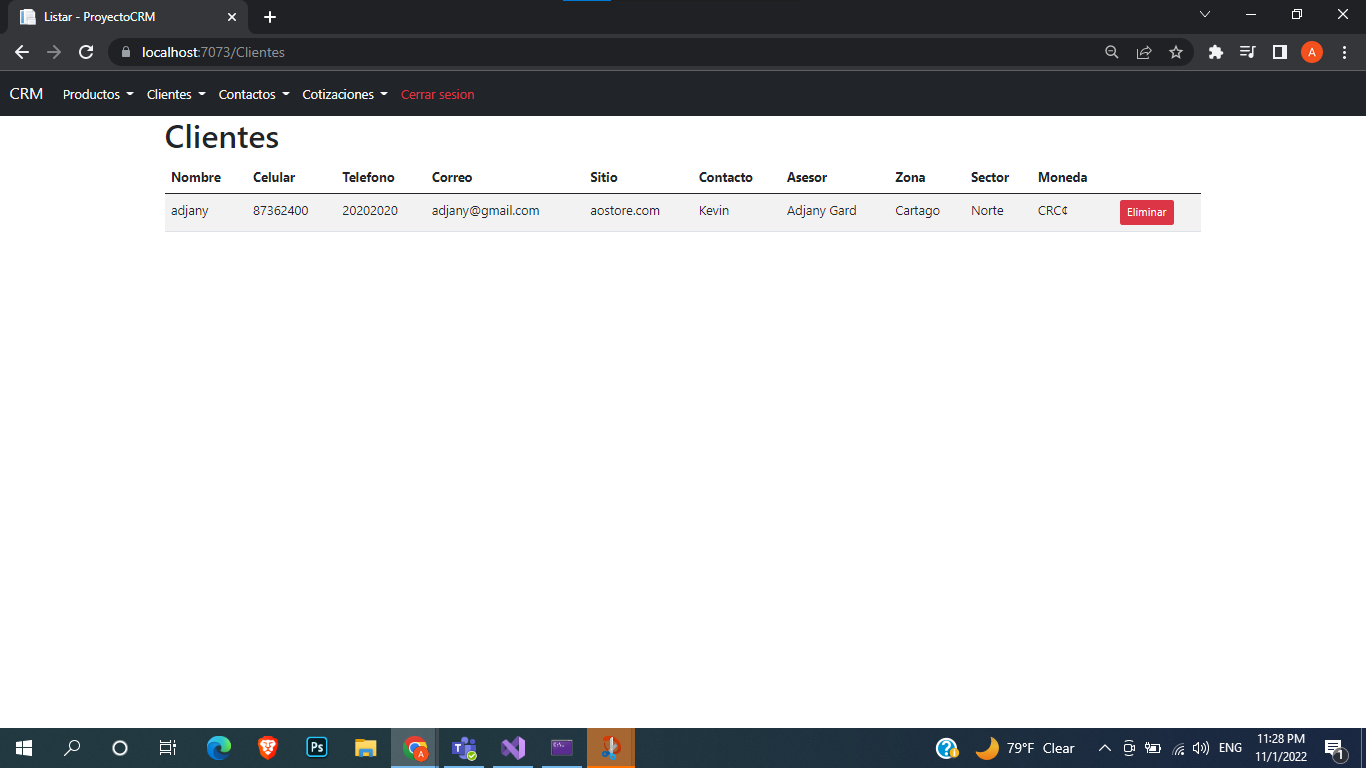


## Funcionalidades asociadas al gestor de clientes

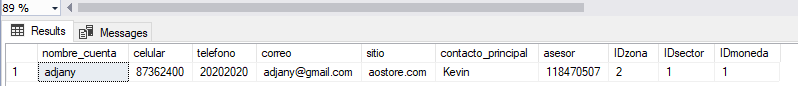
Formulario para agregar clientes.



Visualización del cliente previamente agregado desde la aplicación.

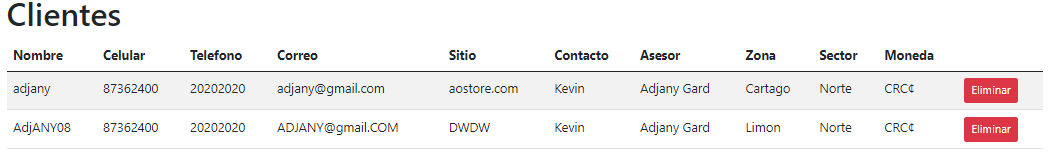


Visualización desde la base de datos.

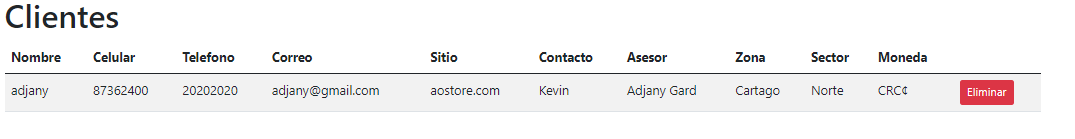


Eliminación de un cliente desde la aplicación.

Antes:



Después:



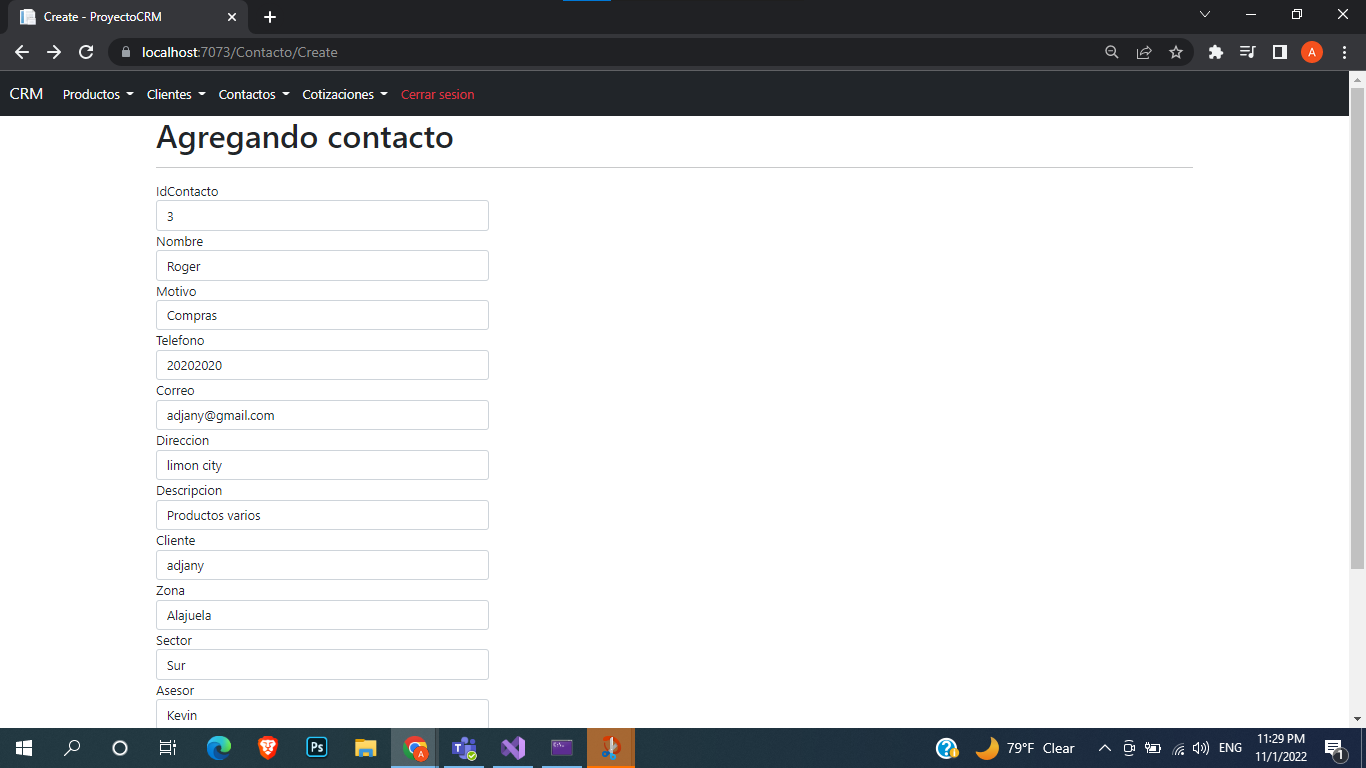
## 

## 

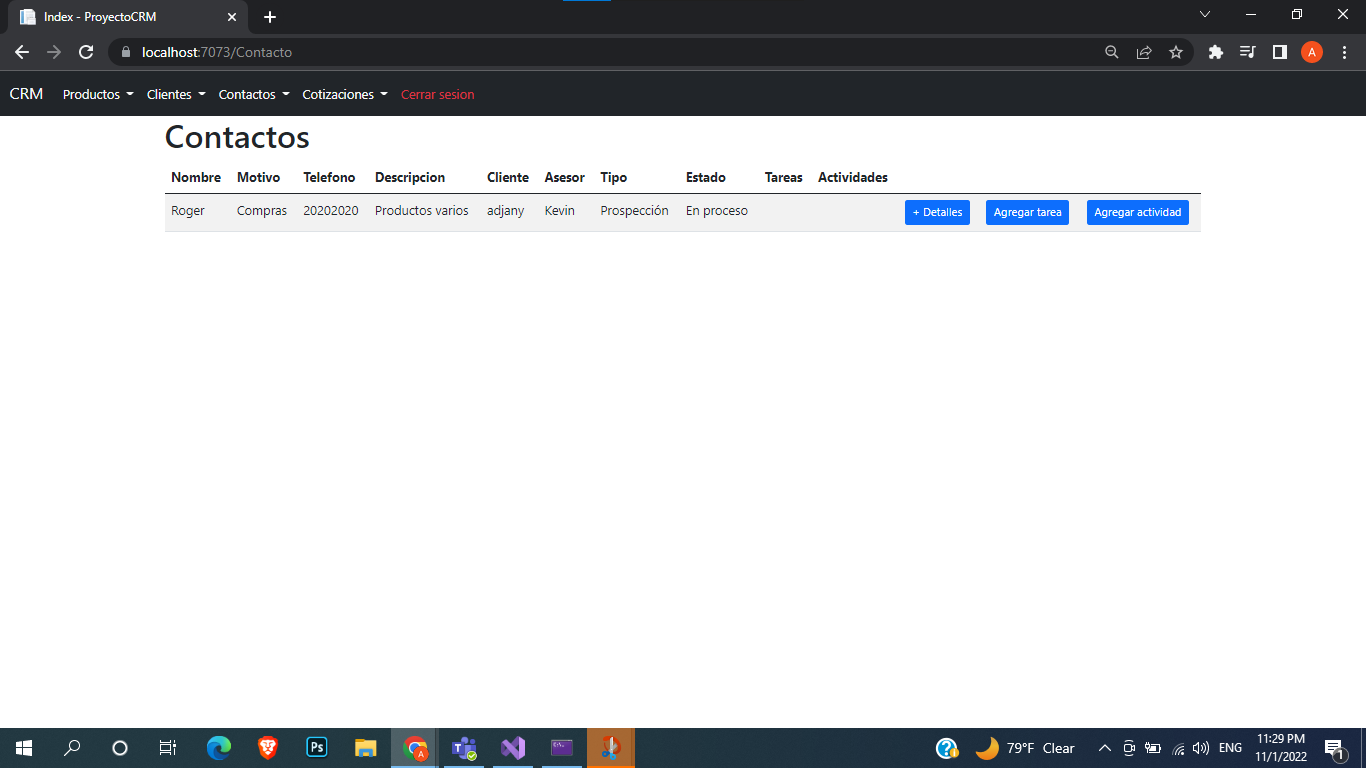
## 

## Funcionalidades asociadas al gestor de contactos

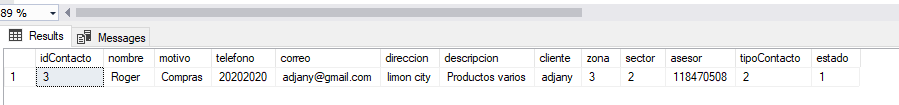
Formulario para agregar contactos desde la aplicación



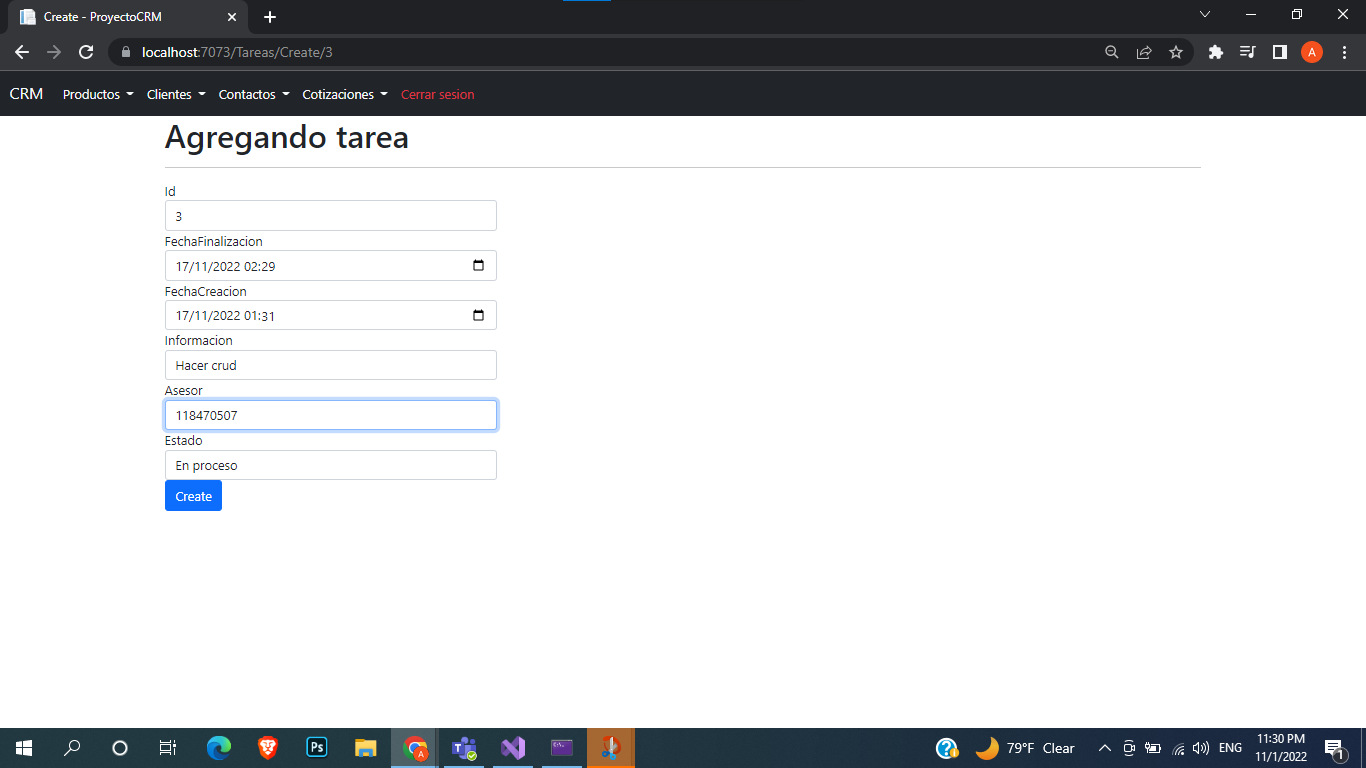
Visualización del contacto previamente agregado desde la aplicación.



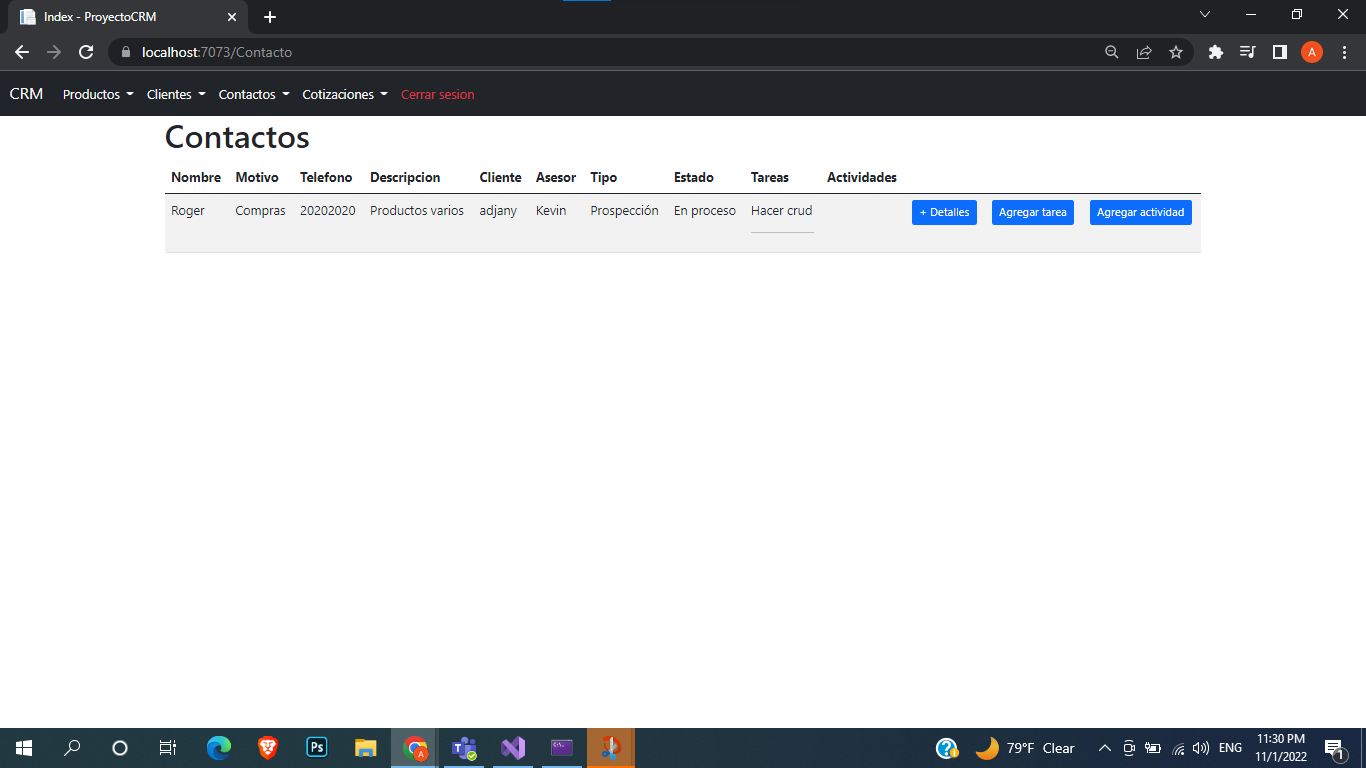
Visualización desde la base de datos.



Formulario para agregar tareas a contactos.



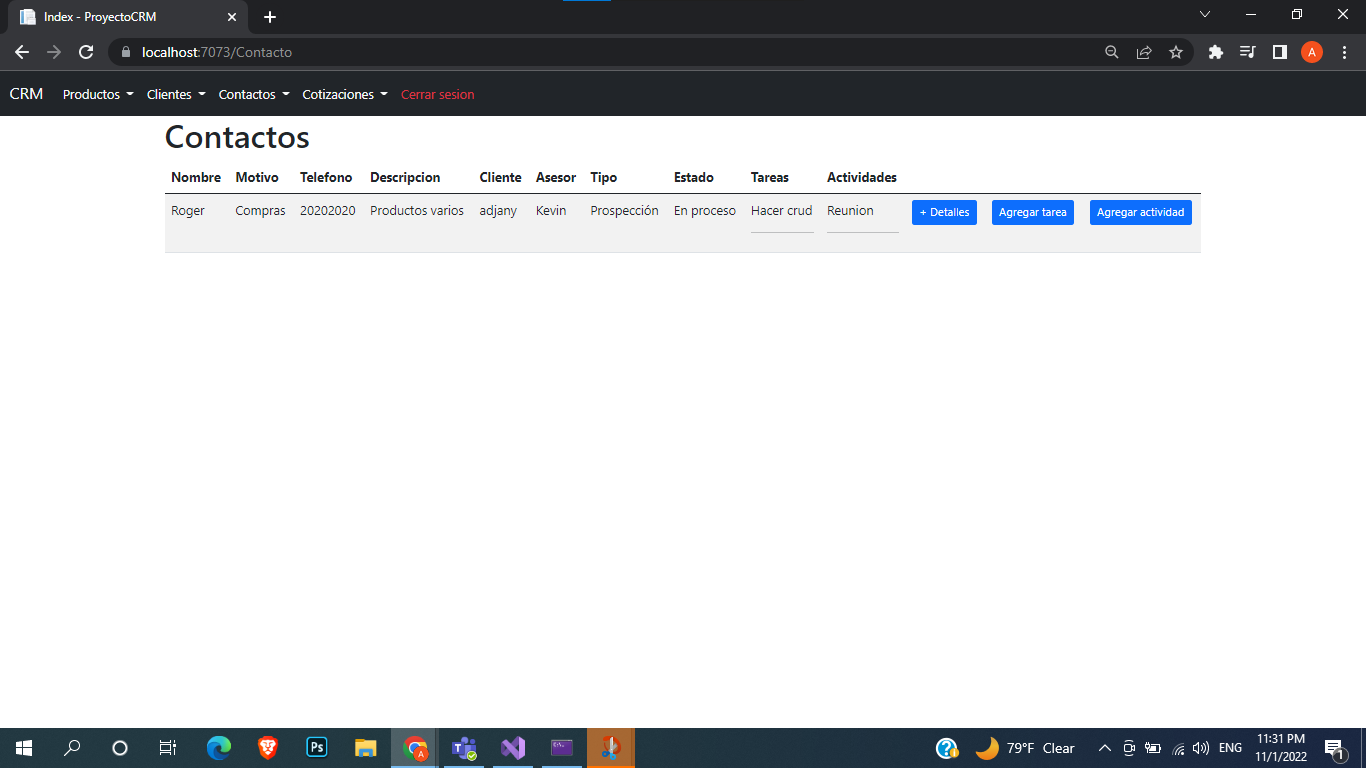
Visualización desde la aplicación.



Formulario para agregar actividades.

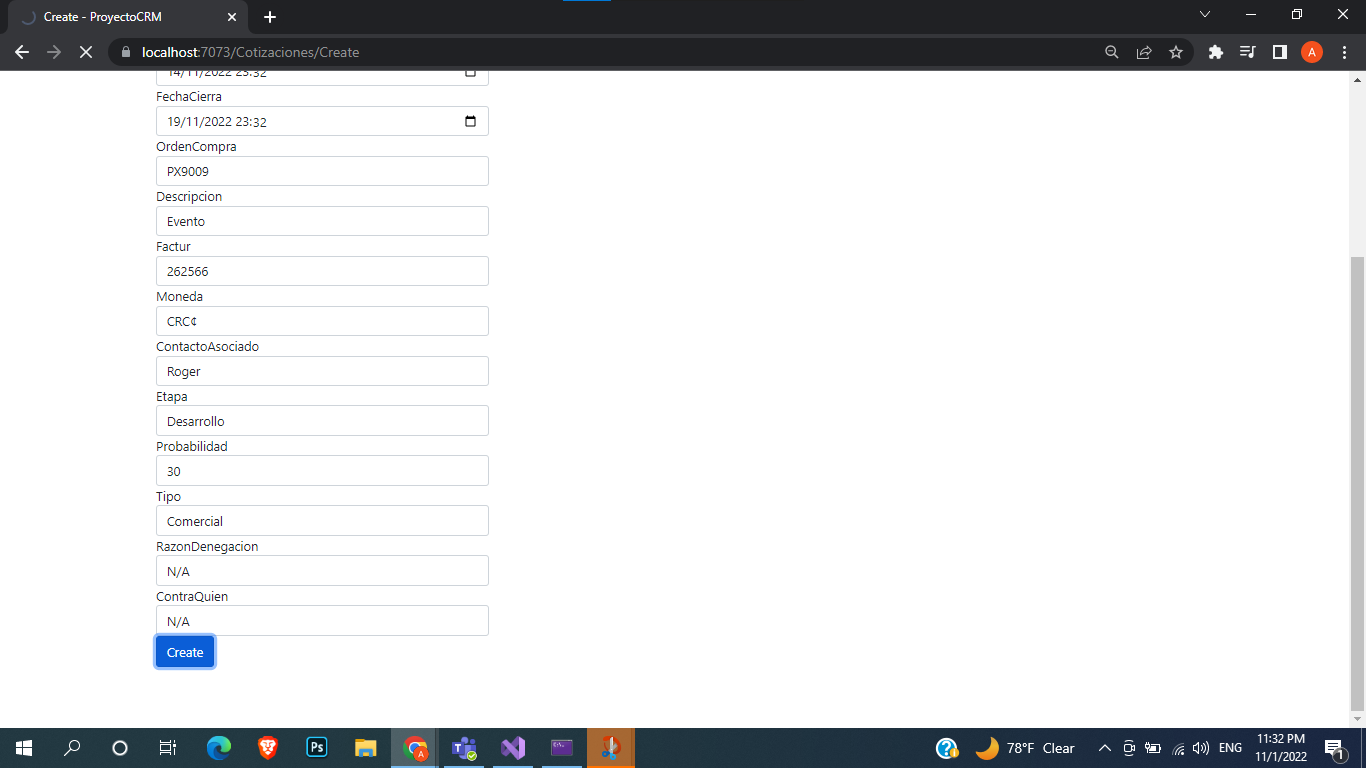


Visualización desde la aplicación.

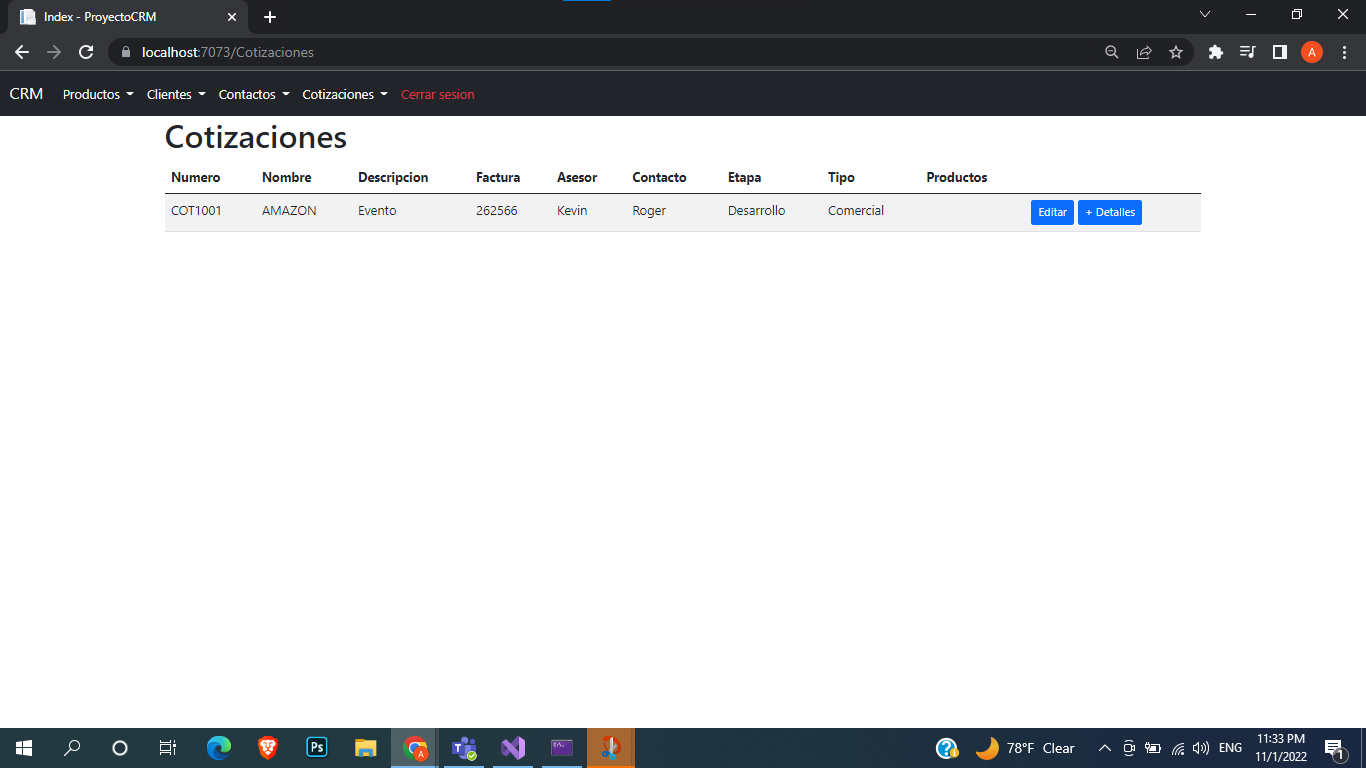


## Funcionalidades asociadas al gestor de cotizaciones

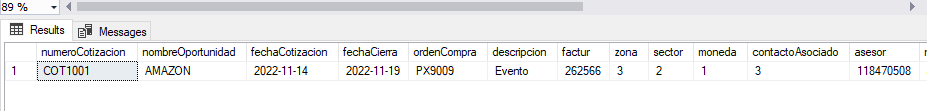
Formulario para agregar cotizaciones desde la aplicación.



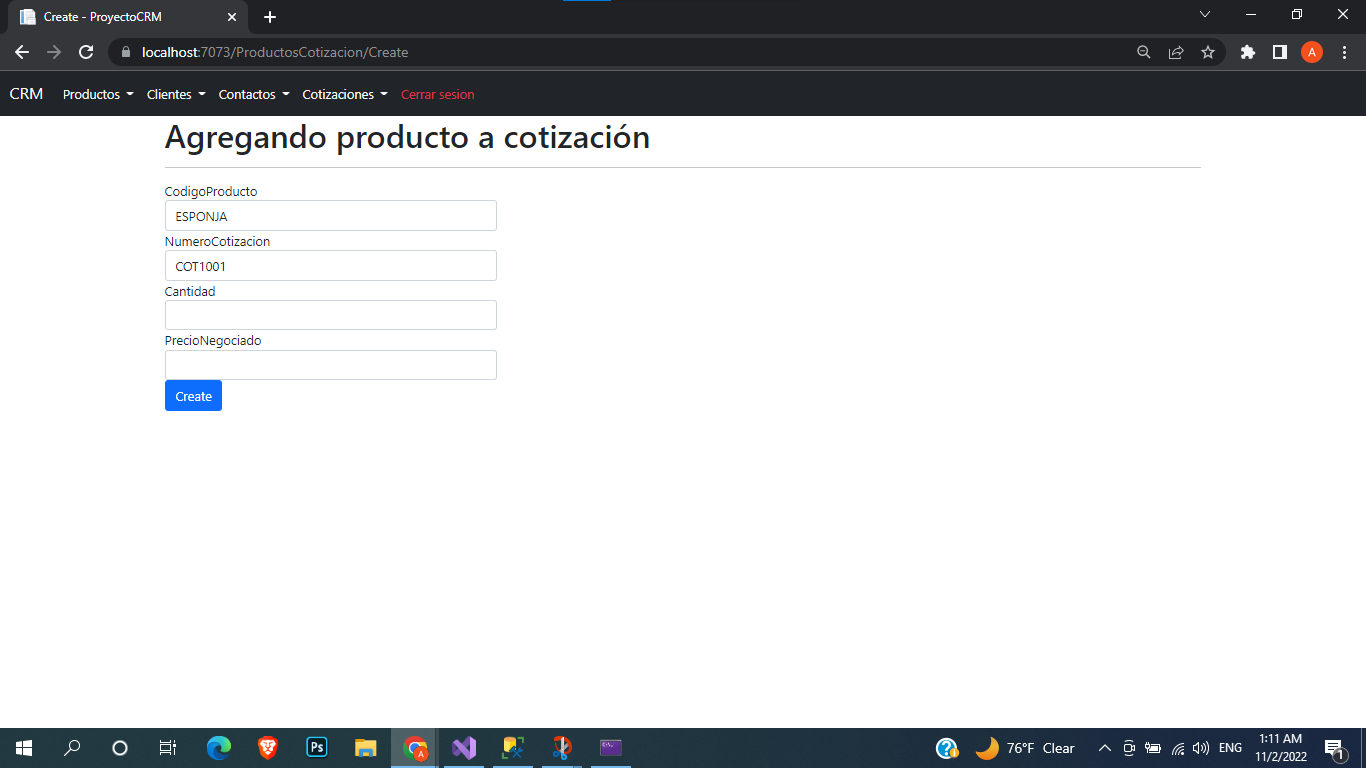
Visualización de la cotización previamente agregada desde la aplicación.



Visualización desde la base de datos.



Formulario para agregar productos.



# Descripción del problema

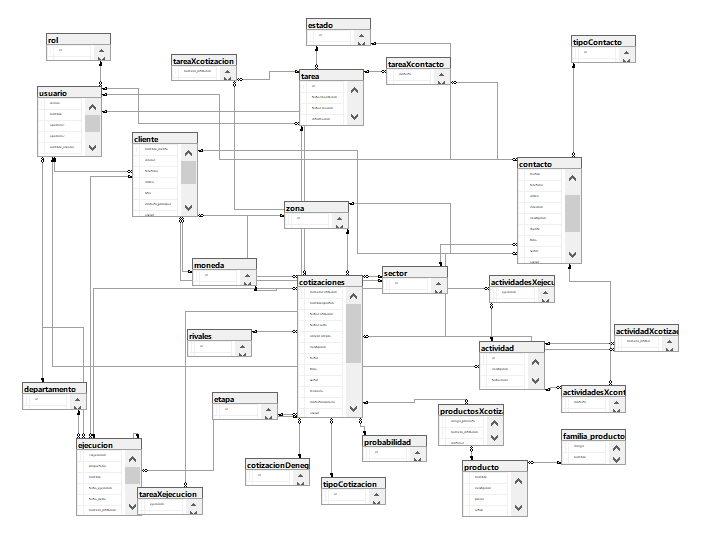
La Administración de Relación con el Cliente (CRM) es una combinación de gente, procesos y tecnología que intenta entender a los clientes de una compañía. Es un acercamiento integrado a las relaciones de manejo del cliente centrándose en la retención del cliente y el desarrollo de las relaciones.

El sistema deberá disponer de un módulo para manejo de usuarios, donde se podrá crear usuarios, modificar la información de estos, crear roles de usuarios, asignación de roles a usuarios, definir la visibilidad y acceso de las opciones y funcionalidades del sistema por roles de usuarios. La información de usuario será nombre de usuario, cédula, apellidos, departamento y clave. El sistema debe permitir controlar el acceso de las funcionalidades a partir de los permisos asignados a los usuarios. El sistema deberá disponer de un módulo de productos para registrar, editar, eliminar, visualizar y generar reportes de la información de los productos (ya cotizados). El módulo de productos debe disponer de la funcionalidad de dar mantenimiento a las familias de productos (tipo producto). El módulo de productos debe disponer de la funcionalidad de reportes de productos, como por ejemplo: productos más cotizados, productos más vendidos, familias de productos más cotizados, familias de productos más cotizados. La información de las familias sería: Nombre de familia de producto, código de familia de producto, activo o no, descripción de familia de producto, entre otros. La información de los productos sería: nombre del producto, código de producto, activo o no, descripción del producto, familia de producto, precio estándar, entre otros. El sistema deberá disponer de un módulo de clientes para registrar, editar, eliminar, visualizar y generar reportes de la información de los clientes. La información de los clientes sería: Nombre de la cuenta, Correo electrónico, Contacto principal, Moneda de la cuenta, Teléfono, Celular, Sitio web, Información adicional, Asesor (usuario), Zona, Sector, entre otros. El módulo de clientes debe disponer de la funcionalidad de reportes de clientes, como por ejemplo: clientes por sector, clientes por zona, cotizaciones por clientes, montos cotizados, entre otros. El sistema deberá contener un módulo de contacto de clientes que permita registrar un contacto, editarlo, consultarlo, asociar actividades, asociar tareas a los registros de contacto de clientes, visualizar y generar reportes de la información de contacto. La información del contacto sería: Cliente, Tipo contacto, Motivo del contacto, Nombre, Teléfono, Correo electrónico, Estado de contacto, Dirección, Sector, Zona, Asesor, Descripción, entre otros. El módulo de contacto de clientes debe disponer en la pantalla de edición la funcionalidad de asignar y dar mantenimiento de tareas a usuarios, relacionada a un contacto de cliente, solicitando la información de la tarea, fecha de finalización, debería tener un estado de tarea y un usuario asignado. El módulo de contacto de clientes debe disponer en la pantalla de edición la funcionalidad de registrar una actividad relacionada al seguimiento de cliente con una descripción asociada a un usuario. El módulo de seguimiento de clientes debe disponer de la funcionalidad para crear reportes que permitan seleccionar los campos almacenados de seguimientos de clientes a desplegar y los campos de filtros. El sistema deberá contener un módulo de cotizaciones que permita registrar una cotización, editarla, asociar actividades, asociar tareas a los registros de cotización, visualizar y generar reportes de la información de cotizaciones. La información de la cotización sería: # Cotización, nombre de la oportunidad, fecha de la cotización, nombre de la cuenta, Mes-Año Proyectado de Cierre, asesor para la cotización, fecha de cierre, etapa, moneda de la oportunidad, probabilidad (%), orden de compra, tipo de cotización, descripción, zona, sector, # contacto asociada, factura #, ¿por qué se denegó?, ¿contra quién?, entre otros, además de asignar productos.

# Diseño del programa

Para el diseño del programa se utiliza el lenguaje de programación c# y para el manejo de la base de datos SQL server. Para la creación del sitio web y la conexión a la base de datos se utiliza .NET 6 usado el patrón de diseño MVC además de la librería llamada Entity framework para la conexión con SQL server y agilizar el trabajo de crear el CRUD necesario para el prototipo del CRM.

## Diagrama Base de datos.



# Librerías usadas

Entity framework para la conexión de la aplicación web con la base de datos.

Data SQLclient para ejecutar los procedimientos almacenados.

Entity framework tools para el scaffold y poder mapear los modelos de la base de datos.

Boostrap para la creación de componentes gráficos.

# Análisis de resultados

## Objetivos alcanzados

1. Crear el login y darle permisos a los usuarios según sus roles.

2. Creación de la base de datos completa para el correcto funcionamiento de la aplicación.

3. Creación de todos los procedimientos almacenados para realizar las funciones necesarias dentro del programa.

4. Todas las funcionalidades que deben implementarse para la gestión de productos.

5. Todas las funcionalidades que deben implementarse para la gestión de clientes.

6. Todas las funcionalidades que deben implementarse para la gestión de contactos.

6. Todas las funcionalidades que deben implementarse para la gestión de cotizaciones.

## Objetivos no alcanzados

# Bitácora.

Link:<https://github.com/Kevin-Salazar-itcr/Proyecto2-BD>