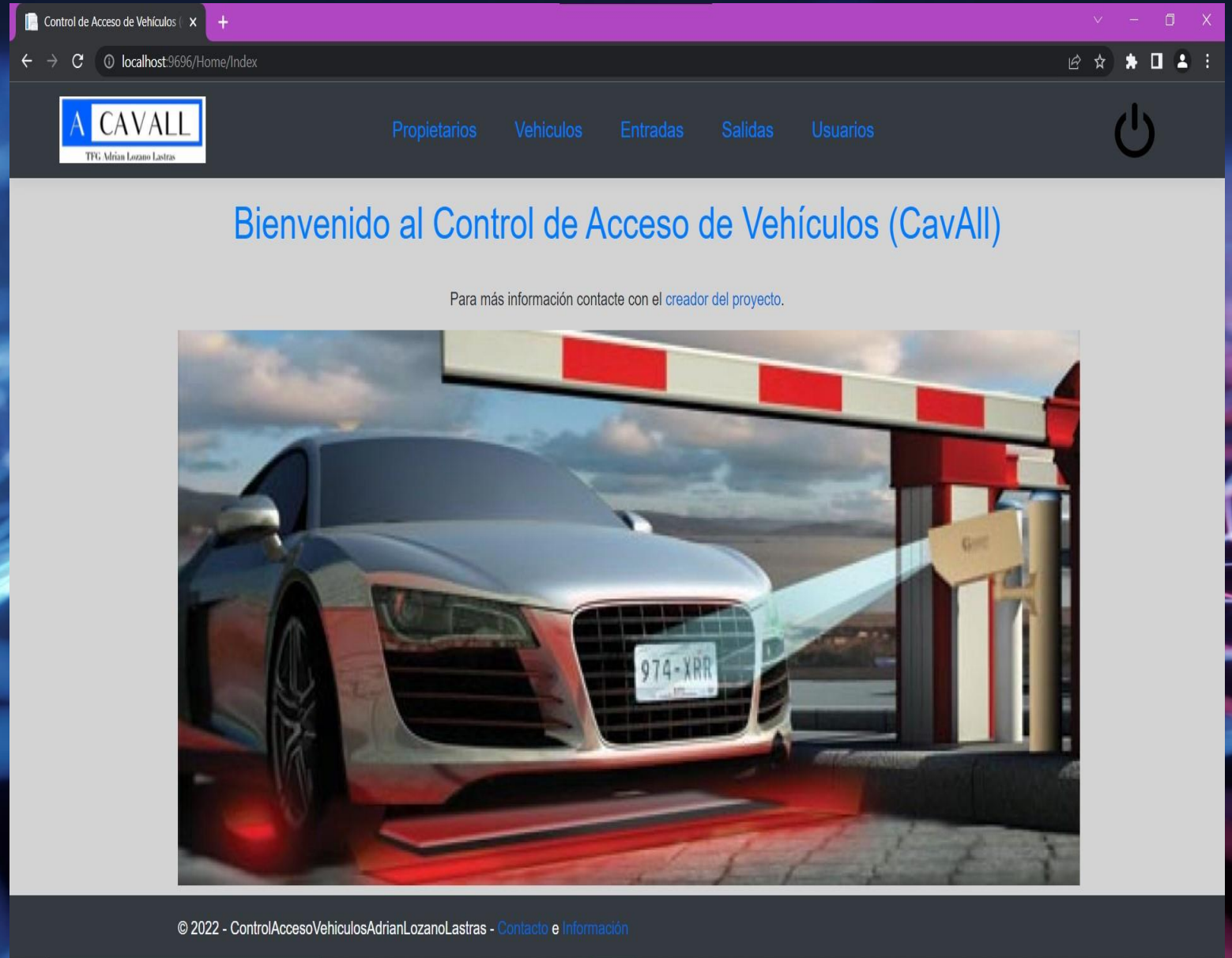


# CONTROL DE ACCESO DE VEHÍCULOS

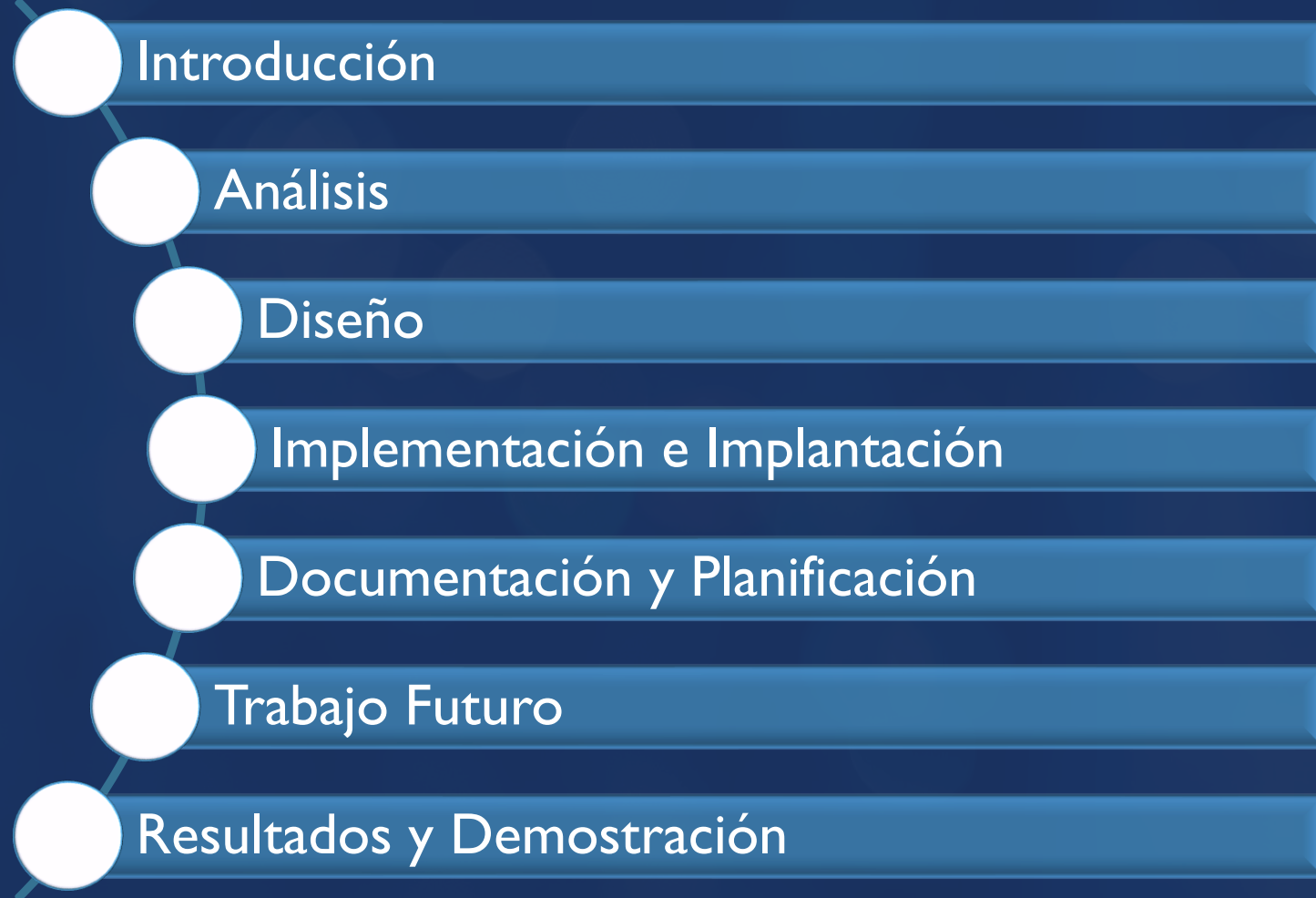
ADRIAN LOZANO LASTRAS

PROYECTO FINAL DEL CICLO  
FORMATIVO DE GRADO SUPERIOR

EN DESARROLLO DE APLICACIONES  
MULTIPLATAFORMA



# CONTROL DE ACCESO DE VEHÍCULOS

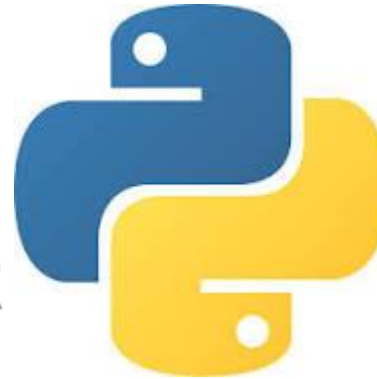


# INTRODUCCIÓN

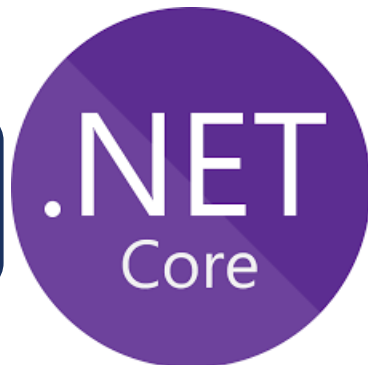
- El presente proyecto pretende desarrollar un sistema para el reconocimiento de matrículas, utilizando visión artificial, que permita detectar automáticamente las matrículas para poder controlar el acceso a una instalación de los diferentes vehículos que deseen acceder, verificando si estos están autorizados o no para poder acceder a un aparcamiento, mejorando de esta forma la seguridad en una empresa, centro, edificio, institución ,etc.
- Consta de dos partes, una parte de cliente y otra parte de servidor.
- El cliente será el encargado de realizar el reconocimiento de las matriculas, validándolas y realizando peticiones al servidor para saber si dichas matrículas tienen o no acceso a la instalación, mostrando una notificación con la matrícula del vehículo indicando si tiene o no permiso. Para ello se ha realizado un programa ejecutable que conectado a una cámara permita el reconocimiento de matrículas.
- El servidor se encargará de obtener las matriculas enviadas por el cliente para cotejarlas con la base de datos y verificarlas, una vez hecho le devuelve una respuesta al cliente y en caso de que la matricula este cotejada se registrará su entrada o salida correspondiente. Para ello se ha realizado una aplicación web que permita realizar las diferentes consultas de las entradas y salidas realizadas, y los vehículos y propietarios registrados.

# ANÁLISIS DEL PROYECTO

CLIENTE



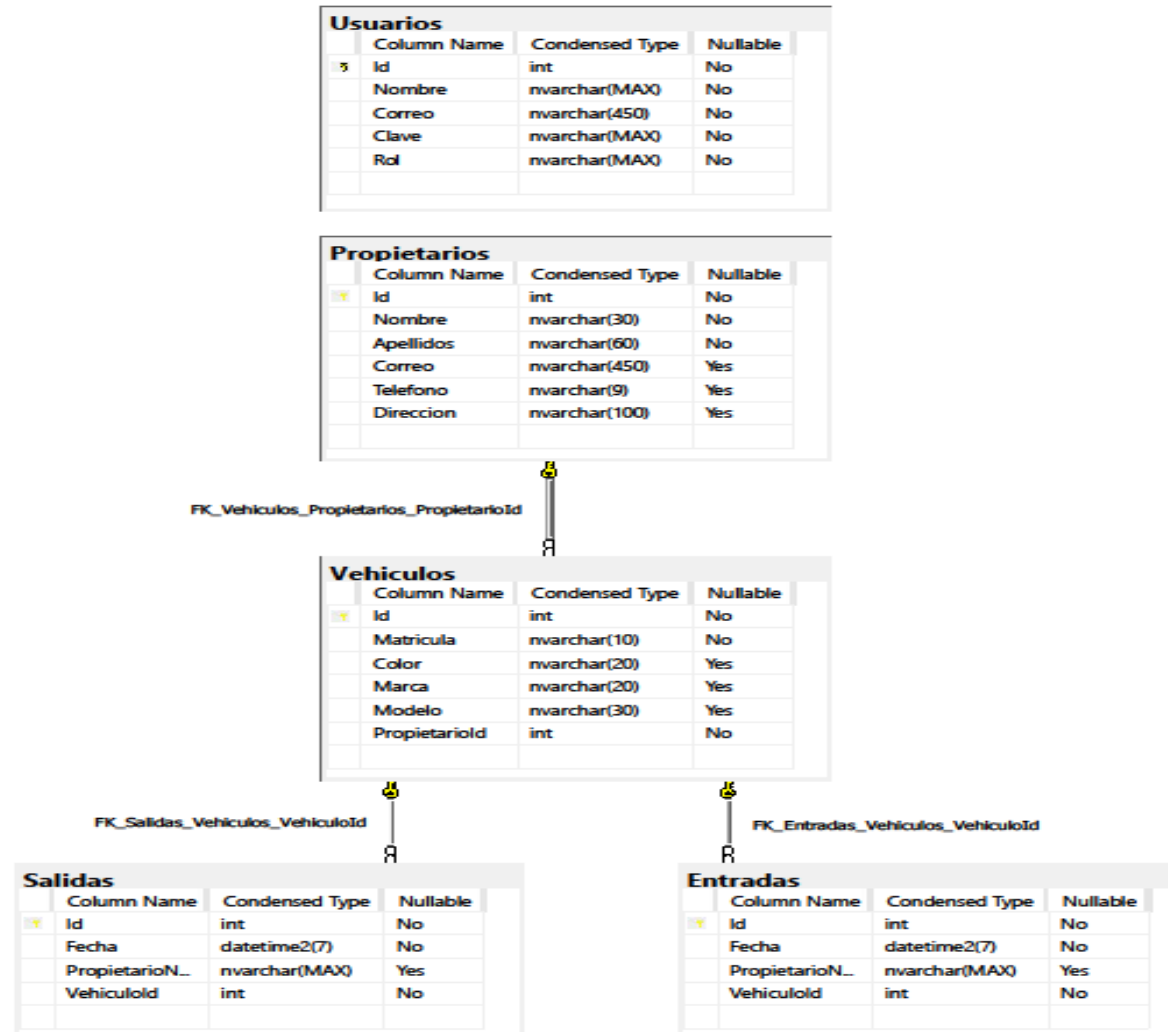
SERVIDOR



Visual Studio 2022




## Diagrama de la Base de Datos del Control de Acceso de Vehículos





# DISEÑO










**EntradaController** 


Clase  
→ Controller

Campos

-  \_context
-  log



Métodos

-  Create (+ 1 sobrecarga)
-  Delete
-  Details
-  EntradaController
-  Entrar
-  Exportar\_Excel
-  Index








**SalidaController** 


Clase  
→ Controller

Campos

-  \_context
-  log


Métodos

-  Create (+ 1 sobrecarga)
-  Delete
-  Details
-  Exportar\_Excel
-  Index
-  SalidaController
-  Salir




**HomeController** 


Clase  
→ Controller

Campos

-  \_logger


Métodos

-  Error
-  HomeController
-  Index









**PropietarioController** 


Clase  
→ Controller

Campos

-  \_context


Métodos

-  Create (+ 1 sobrecarga)
-  DeleteConfirmed
-  Details
-  Edit (+ 1 sobrecarga)
-  Exportar\_Excel
-  Index
-  PropietarioController
-  PropietarioExists









**VehiculoController** 


Clase  
→ Controller

Campos

-  \_context


Métodos

-  Create (+ 1 sobrecarga)
-  DeleteConfirmed
-  Details
-  Edit (+ 1 sobrecarga)
-  Exportar\_Excel
-  Index
-  VehiculoController
-  VehiculoExists











**UsuarioController** 

Clase  
→ Controller

Campos

-  \_context

Métodos

-  ConvertirSha256
-  Create
-  CreateAsync
-  DeleteConfirmed
-  Details
-  Index
-  Login (+ 1 sobrecarga)
-  Salir
-  UsuarioController
-  ValidarUsuario

# DISEÑO

CONTEXTO DE LA BASE DE DATOS

ApplicationDbContext

Clase

DbContext

Propiedades

Entradas

Propietarios

Salidas

Usuarios

Vehiculos

Métodos

ApplicationDbContext

LOG Y PAGINACIÓN

Log

Clase

Métodos

EntradaLog

Log

SalidaLog

PROGRAM Y STARTUP

Program

Clase

Métodos

CreateHostBuilder

Main

Startup

Clase

Propiedades

Configuration

Métodos

Configure

ConfigureServices

Startup

Pager

Clase

Propiedades

CurrentPage

EndPage

PageSize

StartPage

TotalItems

TotalPages

Métodos

Pager (+ 1 sobrecarga)

DISEÑO

**Entrada**  
Clase

▲ Propiedades

- 🔑 Fecha
- 🔑 Id
- 🔑 PropietarioNombre
- 🔑 Vehiculo
- 🔑 Vehiculold

**Salida**  
Clase

▲ Propiedades

- 🔑 Fecha
- 🔑 Id
- 🔑 PropietarioNombre
- 🔑 Vehiculo
- 🔑 Vehiculold

**ErrorViewModel**  
Clase

▲ Propiedades

- 🔑 RequestId
- 🔑 ShowRequestId

**Propietario**  
Clase

▲ Propiedades

- 🔑 Apellidos
- 🔑 Correo
- 🔑 Direccion
- 🔑 Id
- 🔑 Nombre
- 🔑 NombreCompleto
- 🔑 Telefono

**Vehiculo**  
Clase

▲ Propiedades

- 🔑 Color
- 🔑 Id
- 🔑 Marca
- 🔑 Matricula
- 🔑 Modelo
- 🔑 Propietario
- 🔑 Propietariold

**Usuario**  
Clase

▲ Propiedades

- 🔑 Clave
- 🔑 Correo
- 🔑 Id
- 🔑 Nombre
- 🔑 Rol

# DISEÑO



# IMPLEMENTACIÓN DEL CLIENTE

- El cliente será el encargado de obtener el vídeo de una cámara para procesar cada fotograma y así obtener la matrícula, validarla y enviársela al servidor para comprobarla con la base de datos.
- Para ello lo primero será conectar al equipo en el cual se encuentre el cliente y desde el cual se vaya a ejecutar, una [cámara](#) con la cual poder visualizar los diferentes vehículos que deseen acceder, obteniendo una captura en vídeo con la cual el programa cliente se encargará de almacenar cada uno de los fotogramas captados por la cámara para poder [procesarlos y realizar los diferentes tipos de filtrado o emplear en su lugar un clasificador en cascada](#).
- Una vez procesada y tratada la imagen se procede a calcular tanto el contorno como el área donde se encuentra la matrícula, pudiendo de esta forma calcular la relación de aspecto obtenido así en la imagen el rectángulo que forma la matrícula en caso de que se encuentre en la imagen.
- Tras haber obtenido el rectángulo que conforma la matrícula se procede a obtener el texto de la matrícula mediante el reconocimiento óptico de caracteres, intentando sacar los caracteres de la matrícula.
- En cuanto el texto obtenido de la matrícula tenga más de siete caracteres se procede a validar dicho texto para comprobar que la matrícula sea válida, para ello se comprueba que empiece por cuatro números y termine por tres letras o que empiece por una letra con cuatro números en el medio y termine por dos letras, estas matrículas corresponden a España, en otros países la validación será diferente.
- A continuación se realiza una petición al servidor pasándole la matrícula validada y según la respuesta recibida por el servidor el cliente mostrará una notificación con el estado de la matrícula notificando si el acceso a sido permitido o denegado según la respuesta recibida por el servidor. Solo en caso de que la matrícula tenga acceso permitido el programa se pausará durante un minuto para dejar de buscar matrículas hasta que el vehículo pase el control, en caso contrario, seguirá buscando matrículas.
- Todo este proceso lo realiza el cliente continuamente hasta que el usuario pulse sobre la tecla de escape(ESC) o desconecte el equipo en el que se encuentre el cliente, finalizando así el programa.

# IMPLEMENTACIÓN DEL SERVIDOR

- El servidor será el encargado de comprobar las matriculas en la base de datos, realizar el registro tanto de las entradas como de las salidas y gestionar los vehículos con acceso permitido y sus propietarios.
- El servidor esta desarrollado con una aplicación web de ASP.NET Core (Modelo-Vista-Controlador), en donde la capa modelo corresponde con las clases asociadas a las tablas de la base de datos con su información, la capa de vista corresponde con las vistas que se mostraran en la aplicación web y la capa de controlador será la encargada de comunicar los modelos con las vistas obteniendo los datos de los modelos desde la base de datos para tratarlos y pasarlos a la vista y al contrario obtener los datos ingresados en las vistas para insertarlos en las tablas correspondientes de la base de datos.
- **Modelos:** En esta capa se encuentran las clases correspondientes con las tablas de la base de datos.
- **Controladores:** Es la capa más importante y en la que se encuentra la lógica del programa. Se ha creado un controlador por cada entidad, estando asociado cada uno de los controladores con su modelo correspondiente.
- **Vistas:** Es la parte web donde se muestra toda la información y en la cual nada más iniciar la página web se pedirá un inicio de sesión, una vez iniciada la sesión se mostrará la ventana principal con una barra superior donde se encontrarán los diferente apartados para gestionar.

# Control De Entradas

Día

Mes

Año

Nombre del Propietario

Buscar

Mostrar Todas Las Entradas

Crear Entrada

Eliminar todas las Entradas

Exportar Excel

Fecha	Propietario	Vehículo	
28/05/2022 15:19:39	Wanda Alvarez	8915 DWB	Información
28/05/2022 15:22:55	Suzanne Stanley	2340 LWM	Información
01/06/2022 19:05:31	Wanda Alvarez	8915 DWB	Información
01/06/2022 19:22:59	Paco Ramirez Segovia	0735 GPT	Información
01/06/2022 19:25:57	Paco Ramirez Segovia	2759 GPR	Información
01/06/2022 19:28:48	Paco Ramirez Segovia	6363 DDK	Información
01/06/2022 19:30:25	Wanda Alvarez	4324 BTY	Información
03/06/2022 17:44:18	Eli Morgan	3480 HYM	Información

## Propietarios De Los Vehiculos

Busqueda por Nombre ...

Buscar

Mostrar Todos Los Propietarios

Crear Nuevo Propietario

Exportar Excel

Nombre	Apellidos	Correo	Teléfono	Dirección	
Wanda	Alvarez	wanda.alvarez@example.com	687985847	6785 Stevens Creek Blvd	<div>EditarInformaciónEliminar</div>
Suzanne	Stanley	suzanne.stanley@example.com	676786474	5283 College St	<div>EditarInformaciónEliminar</div>
Eli	Morgan	eli.morgan@example.com	685476343	8536 Elgin St	<div>EditarInformaciónEliminar</div>
Paco	Ramirez Segovia	paco@ramirez.com	696758644	Calle de la Piruleta	<div>EditarInformaciónEliminar</div>

## Vehículos Registrados

Busqueda por Matrícula o Propietario ...

Buscar

Mostrar Todos Los Vehículos

Crear nuevo Vehículo

Exportar Excel

Matricula	Color	Marca	Modelo	Propietario	
8915 DWB	Blanco	Citroen	C5	Wanda Alvarez	<div>EditarInformaciónEliminar</div>
2340 LWM	Negro	Seat	Arona	Suzanne Stanley	<div>EditarInformaciónEliminar</div>
0735 GPT	Rojo	Citroen	C7	Paco Ramirez Segovia	<div>EditarInformaciónEliminar</div>
2759 GPR	Blanco	Volkswagen		Paco Ramirez Segovia	<div>EditarInformaciónEliminar</div>
6363 DDK	Negro	Audi	A6	Paco Ramirez Segovia	<div>EditarInformaciónEliminar</div>
4324 BTY	Gris	Citroen		Wanda Alvarez	<div>EditarInformaciónEliminar</div>
7943 KBH	Negro	Toyota		Suzanne Stanley	<div>EditarInformaciónEliminar</div>
3480 HYM	Blanco	Renault		Eli Morgan	<div>EditarInformaciónEliminar</div>
8806 KZS	Azul			Wanda Alvarez	<div>EditarInformaciónEliminar</div>
1234 BCD	Blanco			Suzanne Stanley	<div>EditarInformaciónEliminar</div>

First

Back

1

2

Next

Last

## Iniciar Sesión

adrianll@admin.com

.....

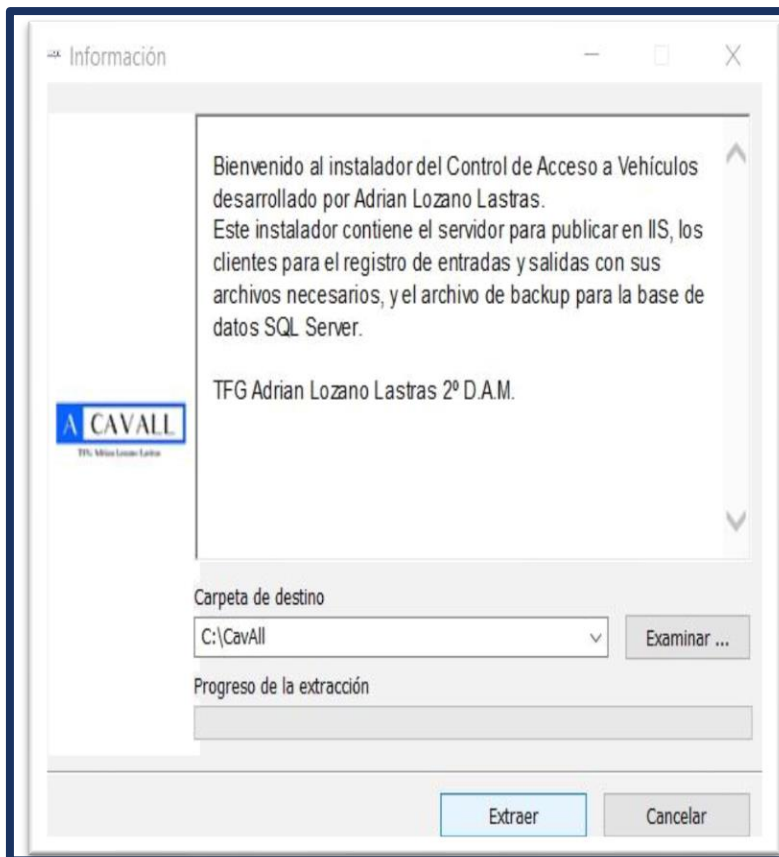
Acceder

Contacto

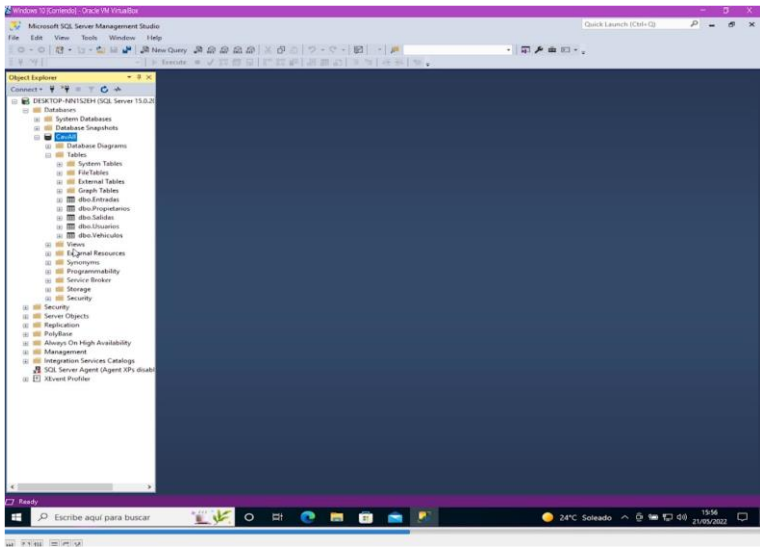
Información

# IMPLEMENTACIÓN DEL SERVIDOR

# IMPLANTACIÓN



- Para realizar el despliegue de la aplicación se han empaquetado todos los archivos necesarios para el correcto funcionamiento del proyecto tanto del cliente como del servidor, generando con ellos un instalador, el cual al ejecutarlo como administrador procederá a realizar la instalación del proyecto.
- **Cliente:** En cuanto al cliente se refiere, una vez instalado el programa CavAll.exe se crearán dos accesos directos en el escritorio uno para la realización de los registros de las entradas y otro para realizar el registro de las salidas. En la carpeta CavAll situada en C:\ se habrán generado el ejecutable de las entradas y el ejecutable de las salidas, junto con el archivo del clasificador necesario para el reconocimiento de las matrículas y las imágenes necesarias del icono y logo del proyecto.
- **Servidor:** En lo que a la parte del servidor se refiere, tras realizar la instalación del programa CavAll.exe se crearán los archivos de backup de la base de datos para restaurarlo en SQL Server y una carpeta con el servidor publicado y listo para usarse en Internet (IIS) Information Service publicándolo en la carpeta correspondiente para su uso y seleccionando el puerto a usar el cual deberá tener permisos del firewall.
- También al realizar las entradas como las salidas se generarán en la carpeta de CavAll los logs de las entradas y salidas tanto permitidas como denegadas según sea el caso.



## CavAllEntrada (version 10.1)

Documentación del programa de Control de Acceso de Vehículos  
Este programa consiste en un script que permite realizar la obtención de la matrícula de un vehículo mediante la captura de video de una cámara, para posteriormente validarla obteniendo los caracteres de la matrícula y verificarla enviandola a un servidor obteniendo una respuesta para saber si dicha matrícula tiene permiso o no para acceder mostrando una notificación con los caracteres de la matrícula indicando el permiso de esta

## Modules

[cv2](#) [requests](#)

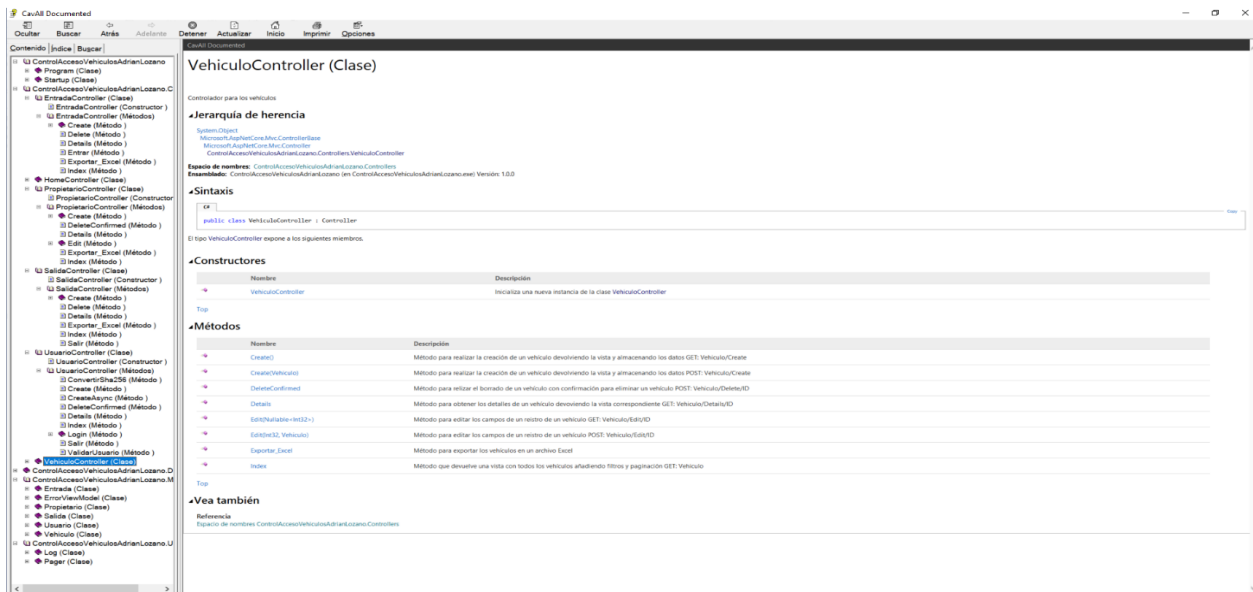
## Data

```
__copyright__ = 'Copyright 2022, AdrianLozanoLastras'
__email__ = 'adrian.loz@hotmail.com'
canny = array([[0, 0, 0, ..., 0, 0], [0, 0, 0, ..., 0, 0]], dtype=uint8)
capture = <VideoCapture 000002A3113B9630>
frame = array([[ [ [ 9, 10, 8], [ 9, 10, 8], ..., 13, 8, 9], [14, 9, 10]], dtype=uint8)
gray = array([[ [ 9, 9, 9, ..., 10, 11, 11], [ 9, ..., [10, 9, 9, ..., 9, 10]], dtype=uint8)
matricula = []
matriculaClasif = <CascadeClassifier 000002A3115844B0>
matriculas = []
mi_matricula = ""
ret = True
salir = 27
```

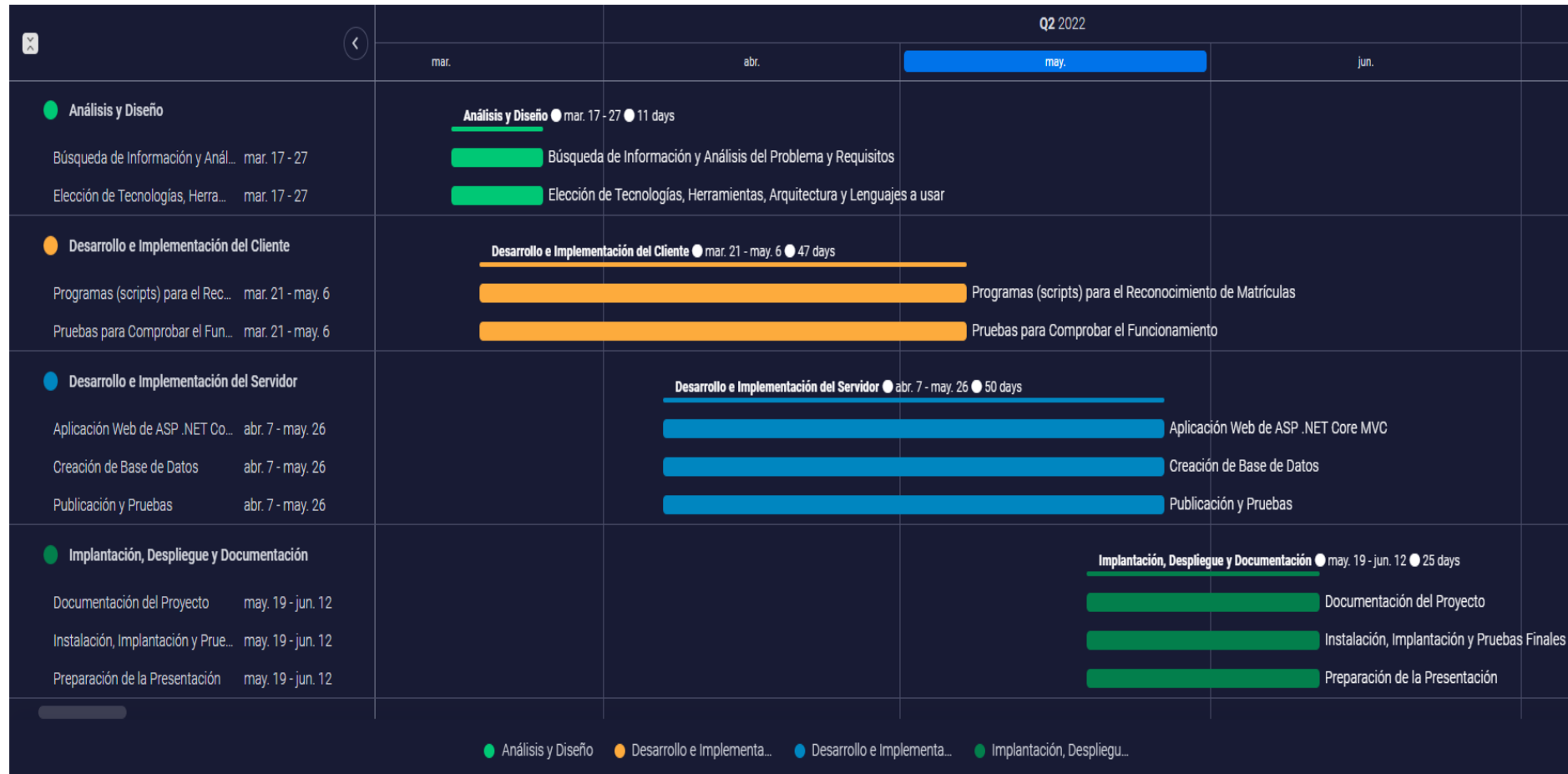
## Author

Adrian Lozano Lastras

# DOCUMENTACIÓN



# PLANIFICACIÓN





# TRABAJO FUTURO

- Como trabajo futuro, se propone realizar el cliente implementándolo en un sistema hardware portátil para la realización de la identificación de las matrículas de los vehículos, como pueda ser una Raspberry Pi que es una placa desarrollada por la organización Raspberry Pi Foundation fundada en Inglaterra en el año 2012 con el objetivo de promover y facilitar el aprendizaje de técnicas de programación en la población.
- La Raspberry Pi es un hardware de bajo coste que suele tener bajos recursos y capacidades, ya que se suele utilizar para el desarrollo de funciones parciales, para ejecutar sistemas embebidos y adaptados para cada plataforma.
- Para ello se han realizado pruebas con una Raspberry Pi Modelo B de 2Gb de RAM, probando el reconocimiento de matrículas en ella pero se han encontrado algunas limitaciones, como pueden ser algunos problemas con el script de Python como ha sido el rendimiento con expresiones regulares a la hora de realizar el análisis de las matrículas en tiempo real y se ha observado un rendimiento inferior al esperado, tardando más en realizar el reconocimiento y procesamiento de las matrículas y obteniendo una tasa mayor de errores que el cliente en Windows. Estos problemas podrían solventarse con una placa con un mayor rendimiento como pueda ser 4 o incluso 8Gb de RAM para que de esta forma funcione más fluido el reconocimiento de las matrículas.
- Por ello se ha decidido no implementar la parte del cliente en un hardware portátil como puede ser Raspberry Pi y se ha dejado como trabajo futuro.



# RESULTADOS

- Con el presente proyecto se ha conseguido desarrollar un sistema de control de acceso completo para vehículos detectando las matrículas de estos para verificarlas y validarlas, empleando para ello tecnologías actuales como puede ser el framework de .NET, el uso de visión artificial junto con reconocimiento de caracteres y lenguajes como son Python y C#.

Control de Acceso de Vehículos por Adrian Lozano Lastras

Usuarios Vehículos Entradas Salidas Usuarios

Nombre del Propietario  Buscar

Eliminar todas las Entradas Exportar Excel

	Propietario	Vehículo	
26/05/2022 15:19:39	Wanda Alvarez	8915 DWB	Información
28/05/2022 15:22:55	Suzanne Stanley	2340 LWM	Información
28/05/2022 15:24:44	Eli Morgan	8316 GXY	Información
01/06/2022 19:05:31	Wanda Alvarez	8915 DWB	Información
01/06/2022 19:22:59	Paco Ramirez Segovia	0735 GPT	Información
01/06/2022 19:25:57	Paco Ramirez Segovia	2759 GPR	Información

Python Application (notify.py)

**Acceso Permitido**  
Acaba de acceder el vehículo con matrícula 6363 DDK

# DEMOSTRACIÓN







# GRACIAS

ADRIAN LOZANO LASTRAS

[HTTPS://GITHUB.COM/ADRIAN  
LOZANO96](https://github.com/AdrianLozano96)