

UNIVERSIDAD TECNOLÓGICA DE AGUASCALIENTES



SMARTACCESS

REPORTE DE ESTADÍA PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE

TECNICO SUPERIOR UNIVERSITARIO EN

TECNOLOGIAS DE LA INFORMACION AREA DESARROLLO DE SOFTWARE

PRESENTA:

NOMBRE DEL EGRESADO (A)

OSCAR RENATO GARCÍA RESÉNDIZ

AGUASCALIENTES, AGS.

MAYO, 2021

PROYECTO REALIZADO EN (COYTECH)

ASESOR (A)

GONZALO CAMPOS ARGUETA

TUTOR (A)

LUIS ARIEL VICTORIA CHÁVEZ

## **Agradecimientos ó Dediatorias**

Agradecimientos a Ariel, y mi maestro Gonzalo por su infinita paciencia para coordinar mi trabajo, A mi madre Lorena Yasmin y y mi padre Oscar Guillermo siempre dispuestos para mí.

Especialmente dedicado a Oscar Renato, por el efuerzo, desgaste, y tiempo invertido en este trabajo, seguir siempre adelante y por nunca rendirse

## Contenido

Agradecimientos ó Dediatorias .....	3
Introducción .....	6
Capítulo I Generalidades de la empresa .....	7
Datos generales .....	7
Antecedentes Históricos .....	7
Misión.....	7
Visión.....	7
Valores.....	7
Productos o servicios que ofrece .....	8
Organigrama .....	8
Descripción del departamento donde se realizó la estadía .....	8
Capítulo II: Planteamiento de la Estadía Profesional .....	9
Descripción de la Problemática: .....	9
Objetivos: .....	9
Justificación: .....	9
Viabilidad:.....	10
1. Alcance del proyecto: .....	10
2. Análisis de situación: .....	10
3. Definición de requisitos .....	10
4. Determinación del enfoque .....	10
Resultados esperados: Mejorar la seguridad y el acceso a los fraccionamientos .....	11
Metodología a aplicar: .....	11
¿Qué es el modelo en cascada? .....	11
¿Cómo funciona el modelo en cascada? .....	11
Capítulo III. Desarrollo de la estadía profesional.....	12
Análisis de Requerimientos .....	12
Por Fraccionamiento. ....	12
Para todos los fraccionamientos.....	12
(REQUERIMIENTOS FUNCIONALES): .....	13
Requisitos de sistema e instalación .....	15
EQUIPO PARA RED (e internet).....	15
EQUIPAMIENTO PARA SISTEMA DE CONTROL DE ACCESO .....	15
(REQUERIMIENTOS NO FUNCIONALES): .....	15

DISEÑO .....	16
Diagrama Smart Access .....	16
Documento Oficial Smart Access .....	17
SEGURIDAD Y CONFIDENCIALIDAD .....	18
ADMINISTRACION .....	18
PLAN DE PAGOS .....	19
DESARROLLO O PROGRAMACION .....	20
Login.....	20
Menú .....	24
Administrador Fraccionamientos.....	30
Cámaras.....	33
Visitas Programadas y No Programadas.....	35
Salidas Visitas No Programadas .....	40
Usuarios.....	45
Usuarios Editar.....	51
Usuarios Agregar.....	55
Acceso a Vehículos .....	58
PRUEBAS E IMPLEMENTACION.....	60
Login .....	60
Menu .....	60
Visitas Programadas y No Programadas.....	61
Salidas Visitas No Programadas .....	62
Usuarios.....	62
Capítulo IV. Resultados y conclusiones .....	63
Objetivos .....	63
Lo que se logro.....	63
Lo que no se pudo cumplir .....	64
Trabajo en equipo .....	64
Desarrollo personal.....	64
Conclusiones.....	65
Anexos.....	66
Cronograma de actividades.....	66
Formatos varios .....	67
Referencias Bibliográficas.....	68

## **Introducción**

El proyecto sobre el cual estaremos realizando nuestra estadía profesional es una aplicación de sistema de acceso para condominios y visitantes de fraccionamientos muy amplia y con una variedad de cosas que realiza, esta aplicación brevemente sirve para que el condominio acceda de manera instantánea a su fraccionamiento de residencia y también para los visitantes de los condóminos a través de un código QR. Realizo la estadía profesional para ayudarme a mí a conocer sobre cómo es la vida profesional de mi carrera y para acreditar mi formación de Técnico superior Universitario

¿Cómo está organizada? Esta organizada en 3 pequeñas aplicaciones, usuario, admín, vigilante y Administrador de la aplicación la cuales tendrán diferentes funcionalidades

## **Capítulo I Generalidades de la empresa**

### *Datos generales*

Diseñamos soluciones a medida para la necesidad de nuestros clientes, implementando nueva tecnología para la optimización de sus procesos basados en la industria 4.0

Tenemos nuevas ideas, ya que somos un grupo abierto a propuestas.

Trabajamos en un entorno flexible, abierto a la voz de nuestros compañeros.

Como pilar de nuestra empresa, la innovación siempre prevalecerá.

### *Antecedentes Históricos*

Fundada en 2016, Coytech es un proveedor nacional de servicios orientados a procesos de TI con 1 oficina en Aguascalientes, En diciembre de 2017, fue nombrada en México como una pequeña compañía de servicios de TI.

La compañía cuenta con un equipo maduro de ventas, diseño y entrega, y con la dedicación de personal de alta calidad, se compromete a proporcionar soluciones perfectas para las empresas nacionales y extranjeras

### *Misión*

Maximizar el valor del portafolio de aplicaciones de nuestros clientes.

### *Visión*

Trascender como proveedor global líder en Soluciones de TI y Procesos de Negocio, generando relaciones mutuamente benéficas, de largo plazo y cimentadas en una base de confianza ganada. Construiremos nuestro futuro siendo una empresa sólida y socialmente responsable, con un historial rentable. Proporcionamos servicios innovadores y de alta calidad, impulsados por la pasión de nuestra cultura centrada en el elemento humano.

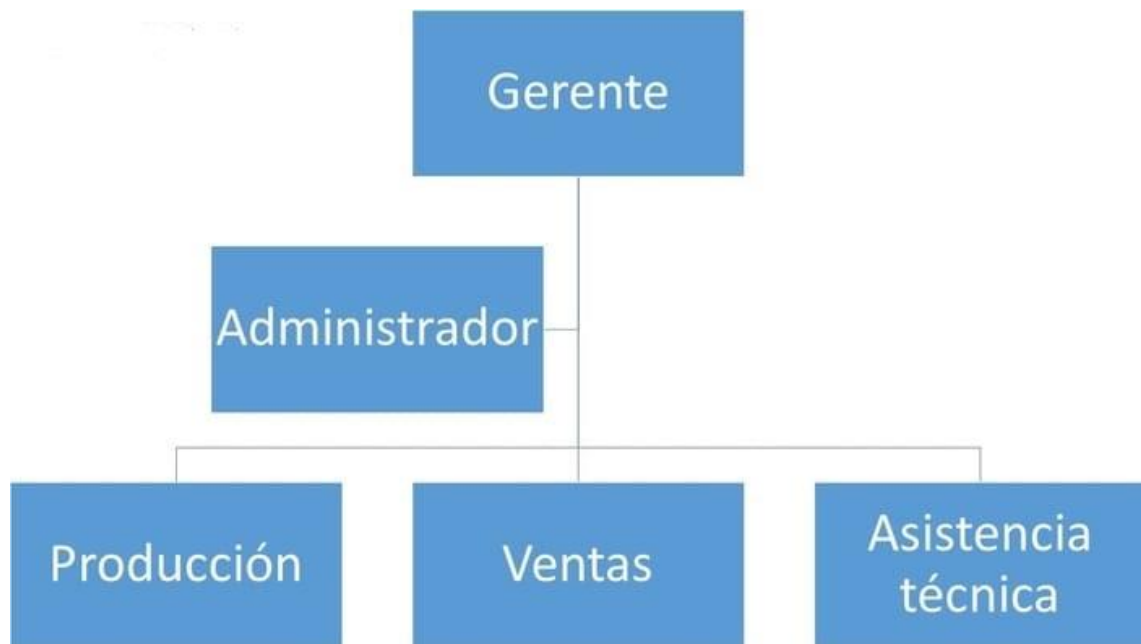
### *Valores*

Apertura, auto-determinación, compromiso, confianza, entusiasmo, flexibilidad, perseverancia, respeto, sinergia, colaboración y visión.

*Productos o servicios que ofrece*

Programación de PLC, Diseño e implementación de centros de datos

*Organigrama*



*Descripción del departamento donde se realizó la estadía*

La estadía se realizó en el departamento de Producción web ya que tuve que trabajar en la aplicación de escritorio, era un lugar de puro desarrollo web móvil, y impresiones 3D



## **Capítulo II: Planteamiento de la Estadía Profesional**

### *Descripción de la Problemática:*

Cuando nosotros vamos a un Fraccionamiento privado para visitar a tu familia, amigos o simplemente si eres el que vive ahí, siempre hay problemas para acceder, o es muy tardado acceder por manera de los guardias o diversas situaciones presentadas en el momento, o a veces las medidas de seguridad para el acceso no son muy buenas y cualquier persona desconocida pueda entrar y no sabemos qué peligro tendríamos, por eso se crea esta aplicación para hacer el acceso y la salida de los condominios más sencilla de manera tecnológica a través de una App.

### *Objetivos:*

El proyecto se basa en la seguridad de los fraccionamientos al momento de permitir el acceso a los visitantes, por medio de un sistema y aplicación los usuarios darán acceso de forma remota desde su celular sin la necesidad de que sea tardado el proceso de entrada ya que con la aplicación validará todos los datos del visitante y a la entrada las cámaras detectaran su rostro, auto y matrícula para mayor seguridad

### *Justificación:*

El sistema brinde mayor seguridad en el acceso de condominios, controle acceso el usuario, el administrador y brinde todos los datos del visitante Por que los condominios no cuentan con mayor seguridad para los usuarios y esta aplicación la brinda

### *Viabilidad:*

*Alcance del proyecto:* Es una aplicación que se encargará de dar correcto funcionamiento a la seguridad de acceso a los fraccionamientos privados será una aplicación web y una aplicación móvil la que serán encargadas de hacer esta función

*Análisis de situación:* Es una aplicación muy funcional con diferentes interacciones hacia el usuario que permite brindar seguridad acceso y sobre todo una experiencia muy sencilla con la aplicación está totalmente se basa en la seguridad del usuario del condominio y de los fraccionamientos haciendo del usuario una muy sencilla utilidad

### *Definición de requisitos:*

#### *Por Fraccionamiento.*

\*Windows Server Estándar (Revisar si se puede usar uno pirata).

\*SQL Server Express (la versión exprés es gratuita y es funcional para fraccionamientos pequeños).

\*Software de escritorio remoto para poder dar mantenimiento y hacer configuraciones remotamente (Puede ser cualquier versión p.e. AnyDesk).

#### *Para todos los fraccionamientos.*

\*Hosting de Windows expuesto a internet que permita alojar Servicio Web Api .Net Framework 4.7 o superior (Es importante que su rendimiento sea óptimo para que se comunique con la App móvil y los servidores de cada fraccionamiento).

\*Cuenta de desarrollador IOS 99 dls. Por año, para desarrollar y publicar en tienda App store.

\*Cuenta de google play pago de una sola vez 25 dls. Para publicar en google play android.

\*Cuenta de correo para enviar emails (ya se tiene uno smart.acces@sisags.com).

*Determinación del enfoque:* Todo este trabajo que se Está realizando solamente con el simple hecho de la determinación de la seguridad que está la aplicación dispuesta a brindarle a los usuarios, condominios y fraccionamientos que utilicen nuestra aplicación esta aplicación dada las pruebas a resultado simplemente 100% seguro efectivo y funcionalmente al 100%

*Resultados esperados:*

*Mejorar la seguridad y el acceso a los fraccionamientos*

*Metodología a aplicar:*

La metodología que se seleccionó para llevar a cabo este proyecto es la de cascada debido a que permite llevar un seguimiento secuencial que ayuda a ordenar cuidadosamente las etapas del proceso para el desarrollo del software, de tal manera que el inicio de cada etapa debe esperar a la finalización de la anterior.

*¿Qué es el modelo en cascada?*

El desarrollo en cascada (en inglés, waterfall model) es un procedimiento lineal que se caracteriza por dividir los procesos de desarrollo en sucesivas fases de proyecto. Al contrario que en los modelos iterativos, cada una de estas fases se ejecuta tan solo una vez. Los resultados de cada una de las fases sirven como hipótesis de partida para la siguiente. El waterfall model se utiliza, especialmente, en el desarrollo de software.

*¿Cómo funciona el modelo en cascada?*

El desarrollo del modelo se atribuye al teórico de la informática Winston W. Royce. Sin embargo, Royce no es el inventor de este modelo.

Requisitos de sistema

Requisitos de software

Análisis

Diseño

Implementación

Prueba

Servicio

## **Capítulo III. Desarrollo de la estadía profesional.**

### **Análisis de Requerimientos**

*Por Fraccionamiento.*

\*Windows Server Estándar (Revisar si se puede usar uno pirata).

\*Sql Server Express (la versión expés es gratuita y es funcional para fraccionamientos pequeños).

\*Software de escritorio remoto para poder dar mantenimiento y hacer configuraciones remotamente (Puede ser cualquier versión p.e. AnyDesk).

*Para todos los fraccionamientos.*

\*Hosting de windows expuesto a internet que permita alojar Servicio Web Api .Net Framework 4.7 o superior (Es importante que su rendimiento sea optimo para que se comunique con la App móvil y los servidores de cada fraccionamiento).

\*Cuenta de desarrollador IOS 99 dls. Por año, para desarrollar y publicar en tienda App store.

\*Cuenta de google play pago de una sola vez 25 dls. Para publicar en google play android.

\*Cuenta de correo para enviar emails (ya se tiene uno [smart.acces@sisags.com](mailto:smart.acces@sisags.com)).

(REQUERIMIENTOS FUNCIONALES):

<b>Identificación del requerimiento:</b>	RF01
<b>Nombre del Requerimiento:</b>	Autenticación de Usuario.
<b>Características:</b>	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
<b>Descripción del requerimiento:</b>	El sistema podrá ser consultado por cualquier usuario dependiendo del módulo en el cual se encuentre y su nivel de accesibilidad.
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"><li>• RNF01</li><li>• RNF02</li><li>• RNF05</li><li>• RNF08</li></ul>
<b>Prioridad del requerimiento:</b> Alta	

<b>Identificación del requerimiento:</b>	RFO2
<b>Nombre del Requerimiento:</b>	Lector de código QR
<b>Características:</b>	Funciona para dar acceso al faccionamiento desde el OR
<b>Descripción del requerimiento:</b>	Es el principal método de entrada para Visitas y condominios
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"><li>• RNF01</li><li>• RNF02</li><li>• RNF05</li><li>• RNF08</li></ul>
<b>Prioridad del requerimiento:</b> Alta	

<b>Identificación del requerimiento:</b>	RFO3
<b>Nombre del Requerimiento:</b>	Camara de Rostro
<b>Características:</b>	Los usuarios tendrán que verificarse con la cámara de rosotro
<b>Descripción del requerimiento:</b>	Sirve para de manera constante verificar por cámaras la persona que ingresa
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"><li>• RNF01</li><li>• RNF02</li><li>• RNF05</li><li>• RNF08</li></ul>
<b>Prioridad del requerimiento:</b> Alta	

<b>Identificación del requerimiento:</b>	RFO4
<b>Nombre del Requerimiento:</b>	Camara de Identificacion
<b>Características:</b>	Camara pegada a monitor de QR para monitorear
<b>Descripción del requerimiento:</b>	Esta cámara moitorea las ID de los visitantes al ser escaneadas
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> <li>• RNF05</li> <li>• RNF08</li> </ul>
<b>Prioridad del requerimiento:</b> Alta	

<b>Identificación del requerimiento:</b>	RFO5
<b>Nombre del Requerimiento:</b>	Lector de tags
<b>Características:</b>	Los usuarios deberán identificarse a través del tag para abrir
<b>Descripción del requerimiento:</b>	Se validan los tags para dar de alta condominios y permitir acceso
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> <li>• RNF05</li> <li>• RNF08</li> </ul>
<b>Prioridad del requerimiento:</b> Alta	

<b>Identificación del requerimiento:</b>	RFO6
<b>Nombre del Requerimiento:</b>	Autenticacion de código OR
<b>Características:</b>	Los usuarios deben autenticar el tag al momento de entrega
<b>Descripción del requerimiento:</b>	Para el funcionamiento es necesario activar los tags antes de su uso
<b>Requerimiento NO funcional:</b>	<ul style="list-style-type: none"> <li>• RNF01</li> <li>• RNF02</li> <li>• RNF05</li> <li>• RNF08</li> </ul>
<b>Prioridad del requerimiento:</b> Alta	

### *Requisitos de sistema e instalación*

#### *EQUIPO PARA RED (e internet)*

Rack, 3 charolas

Organizadores verticales para rack

Barra eléctrica de 12 contactos

Ups de 40 minutos

Gabinete de Pared 6U, hasta 60KGs, Negro

Switch poe 1G de 24 Ptos

#### *EQUIPAMIENTO PARA SISTEMA DE CONTROL DE ACCESO*

Switch Dahua poe 1Gb de 8 Ptos.

Servidor (PC) para base de datos y comunicación con APP

Biométrico ZKTeco K30 lector de huella digital para acceso

Arduino interconector

Tarjeta Control 4 Relevadores Led 10a 250v Azul

W5100 Placa De Expansión Uno Ethernet Shield

ArduinoMXSLP Módulo Adaptador Convertidor De USB A Rs485

#### *(REQUERIMIENTOS NO FUNCIONALES):*

Gabinete para ext. En acero inoxidable para control de acceso

Lector de huellas digitales ZKTeco para pc USB

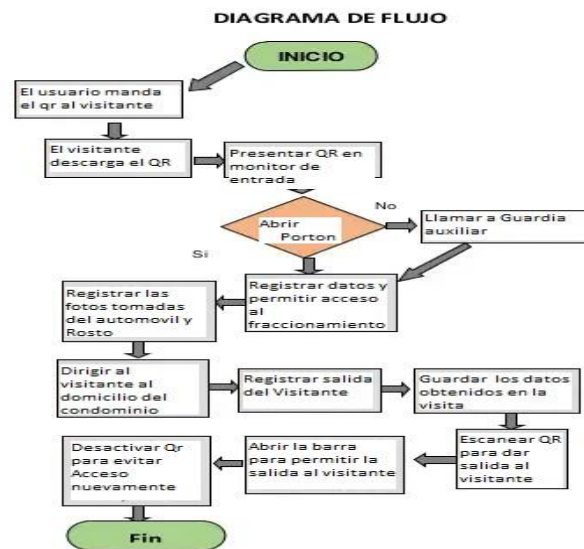
Lector de largo alcance Access Pro, UHF hasta 6 m

Tp-Link EAP110 802.11 b/g/n (2.4 GHz), hasta 300 Mbps, alimentación PoE pasivo, para montaje en techo

## DISEÑO

### Diagrama Smart Access

En estas dos imágenes de diagramas se muestra el funcionamiento interno de como va a trabajar la aplicación.





## *Documento Oficial Smart Access*

Es una plataforma creada para dar la seguridad y confiabilidad necesaria a los colonos de su fraccionamiento.

Actualmente la inseguridad en nuestro país esta incontrolable y los sistemas de control de acceso TRADICIONALES no dan la confiabilidad ni la seguridad necesaria para tu tranquilidad. Esto se debe a que el control del acceso RECAE EN EL VIGILANTE que es quien toma la decisión de quien entra y quién no a tu FRACC.

Es por esto que, en Smart Access, combinamos la tecnología de tu celular con múltiples dispositivos de seguridad para poner el CONTROL DE ACCESO EN TUS MANOS y no en la del vigilante.

- Toda visita será programada desde tu celular ya sea visita única, frecuente o permanente, dándote a ti el control de a quien dejas entrar a tu fraccionamiento al crear un código QR único de acceso en la plataforma.

- Mediante la tecnología del código QR tu visita podrá acusar al fraccionamiento SIN QUE EL VIGILANTE INTERVENGA y el sistema validará el registro y te enviará a tu celular las fotografías de rostro, placas e identificación de tu visita, avisándote que está accedando.

- El código QR queda bloqueado mientras tu visitante se encuentra dentro del fraccionamiento y queda deshabilitado al salir.

- En el caso de visitas NO PROGRAMADAS el sistema cuenta con un módulo para que el vigilante proporcione el acceso quedando un registro de que vigilante y por qué motivo dio el acceso, además que te enviara las fotos de quien está accedando a tu domicilio. (estafeta, correos, CFE, CAPAMA, etc.)

- Para el control de tus vehículos, TU MISMO podrás activar tu tarjeta TAG, adicionando una fotografía de tu vehículo(s) para que en caseta el vigilante pueda corroborar que eres tú quien está entrando y no otro vehículo desconocido.

- De la misma manera podrás bloquear y desbloquear tus TAG's cada vez que tu así lo requieras.

- Otra de las características de control interno y externo, son los biométricos de acceso peatonal, con los cuales podrás controlar a tu personal doméstico, así como el acceso a áreas recreativas.
- La APP es intuitiva y con diseño práctico y el licenciamiento es por DOMICILIO por lo que podrás registrar a toda tu familia por el mismo costo.
- En fraccionamiento que cuentan con el acceso híbrido de portón y barrera, el sistema cuenta con una configuración de apertura por horario, la cual es configurada por el administrador del sistema y de esta manera queda automatizado el control de portón y barrera alternativo.

## *SEGURIDAD Y CONFIDENCIALIDAD*

Nuestra plataforma NO SE INSTALA EN LA NUVE, con lo cual TODA TU INFORMACION SENSIBLE (fotos, datos, horarios, registros de domicilios, etc.) se guarda en un SERVIDOR instalado físicamente en el fraccionamiento al cual nadie tiene acceso.

En caso de que por alguna razón ajena no haya servicio de internet, el sistema NO SE DETIENE, y las notificaciones pendientes, se reanudarán en cuanto vuelva a reanudarse la conexión a internet, esto ayuda a no dejar el fraccionamiento sin control de acceso. (Se recomienda tener dos proveedores de internet)

## *ADMINISTRACION*

A la administración de condóminos se les proporciona un control web, mediante el cual podrán ver en todos los momentos estadísticos de accesos, visitas y permanencias, así como dar de alta domicilios nuevos o bloquear a MOROSOS.

También podrán consultar y revisar historial de visitas incluyendo sus fotografías almacenadas, por domicilio.

## PLAN DE PAGOS

### CONCEPTO

### PRECIO

Plataforma administradora de eventos Smart Access	\$ 25,000.00
Cortesía por contratación con SiS (Soporte Integral de Servicios)	-\$ 25,000.00
Licenciamiento anual por colono (App Smart Access)	\$ 1,200.00
Licenciamiento mensual por colono (App Smart Access)	\$ 110.00

Todos los precios son más i.v.a.

La implementación del sistema incluye:

Capacitación a administradores

Capacitación a personal de vigilancia

Capacitación a condóminos

Asesoría presencial durante el periodo de arranque (un mes aprox.)

Revisión mensual del equipamiento y su buen funcionamiento

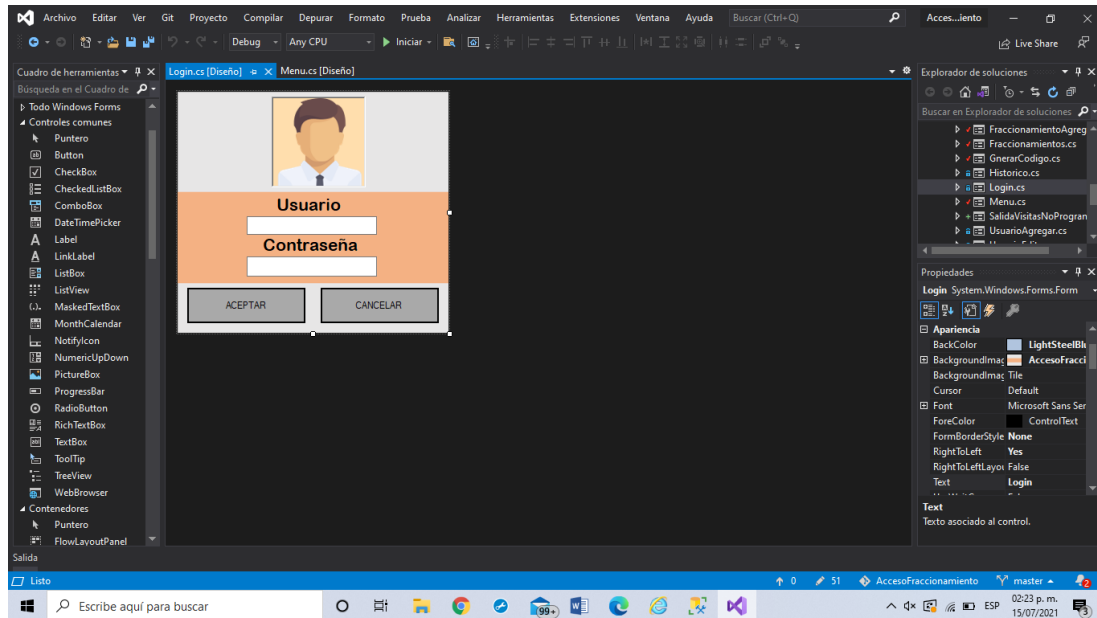
El fraccionamiento Se encargará de reunir el pago de los colonos registrados y tendrá un periodo de 30 días de crédito para realizar el pago en el esquema que elija, tiempo suficiente para la recolección del pago de los licenciamientos.

Soporte Integral de Servicios

## DESARROLLO O PROGRAMACION

### *Login*

En esta pantalla se muestra el inicio de sesión de nuestra aplicación, pidiendo Usuario y Contraseña para dar acceso.



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using AccesoFraccionamiento.EntityFrameworks;
```

```
namespace AccesoFraccionamiento.Views
```

```
{
```

```
public partial class Login : Form
```

```
{
```

```
public static int? Nombre2 { get; set; }
```

```
public static bool entrar2 { set; get; }
```

```
public static string NombreU2 { set; get; }
```

```
public static string Apellido2 { set; get; }
```

```

AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
public static string Nombre { set; get; }
public static string Apellido { set; get; }
public Login()
{
    InitializeComponent();
}

private void OpenLogin()
{
    try
    {
        Form ex = checkStateForm("Login");
        if (ex == null)
            VisualizarFormas(new Login());
        else
        {
            ex.WindowState = FormWindowState.Normal;
            ex.Activate();
        }
    }
    catch
    {
    }
}

private static Form checkStateForm(string formName)
{
    return Application.OpenForms.OfType<Form>().Where(pre => pre.Name ==
formName).SingleOrDefault<Form>();
}

private void VisualizarFormas(Form form)
{

```

```

form.Show();
}

private void btnAceptar_Click(object sender, EventArgs e)
{
    Cursor.Current = Cursors.WaitCursor;
    AccFracc_UsuarioSistema ac = new AccFracc_UsuarioSistema();
    //UsuariosModel appUser;
    ac.NombreUsuario = txtbxUsername.Text.Trim();
    ac.Contrasena = txtbxPassword.Text.Trim();
    using (ACCFRACCEntities db = new ACCFRACCEntities())
    {
        var usu = from u in db.AccFracc_UsuarioSistema
        where u.NombreUsuario == txtbxUsername.Text.Trim() && u.Contrasena ==
        txtbxPassword.Text.Trim()
        select u;

        int id = 0;
        foreach (var i in usu)
        {
            id = i.IdUsuario;
        }
        if (usu.Count() > 0)
        {
            try
            {
                {
                    int? idrol = 0;
                    string Name = "";
                    string Ape = "";
                    foreach (var i in usu)
                    {
                        idrol = i.IdRol;
                        Name = i.Nombre;
                        Ape = i.Apellido;
                    }
                    Nombre2 = idrol;
                }
            }
        }
    }
}

```

```

entrar2 = true;
NombreU2 = Name;
Apellido2 = Ape;

bi.DescripcionBitacora = "El usuario " + Name.Trim() + " inició sesión";
bi.FechaBitacora = DateTime.Now;
bi.IdUsuario = id;
db.AccFracc_Bitacora.Add(bi);
db.SaveChanges();
this.Close();

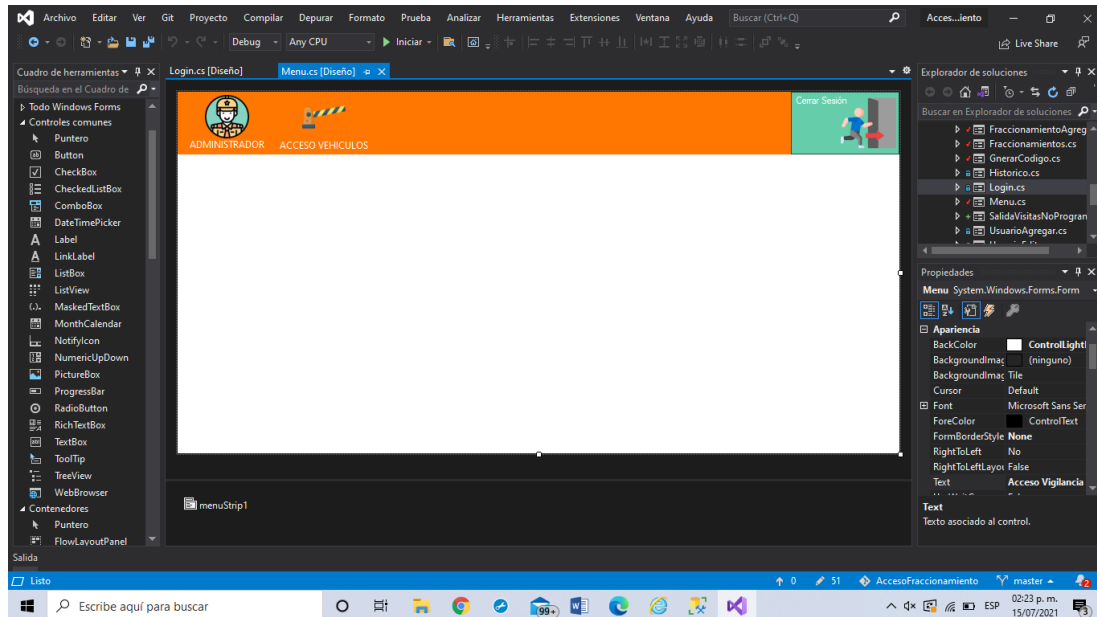
}
catch(Exception ex)
{
    el.Error = "" + ex;
    el.Modulo = "Login";
    el.FechaErrorLog = DateTime.Now;
    db.AccFracc_ErrorLogs.Add(el);
    db.SaveChanges();
    MessageBox.Show("No se pudo acceder al sistema " + ex);
}
}
else
{
    entrar2 = false;
    MessageBox.Show("No se encontró ningun Usuario con el nombre " +
    txtbxUsername.Text.Trim());
}
}
Cursor.Current = Cursors.Default;
}

private void btnCancelar_Click(object sender, EventArgs e)
{
    Application.Exit();
}

```

## Menú

Esta es una pantalla básica donde se muestra en la parte superior tres opciones la de Administrador, Acceso Vehículos y la opción de Cerrar Sesión.



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using AccesoFraccionamiento.EntityFrameworks;
```

```
namespace AccesoFraccionamiento.Views
```

```
{  
    public partial class Menu : Form  
    {  
        ACCFRACCEntities db = new ACCFRACCEntities();  
        AccFracc_UsuarioSistema usu = new AccFracc_UsuarioSistema();  
        public static int Nombre2 { get; set; }  
        public static bool entrar2 { set; get; }  
        public static string NombreU2 { set; get; }
```



```

public static string Apellido2 { set; get; }
AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
public static string Nombre { set; get; }
public static string Apellido { set; get; }
public Menu()
{
    InitializeComponent();
}

private void Menu_Load(object sender, EventArgs e)
{
    menuStrip1.Hide();
    btnCerrarSesion.Hide();
    Login l = new Login();
    l.BringToFront();
    l.StartPosition = FormStartPosition.CenterScreen;
    l.ShowDialog(this);
    entrar2 = Views.Login.entrar2;
    if (entrar2 == true)
    {
        menuStrip1.Show();
        labelaccess.Text = Views.Login.Nombre2.ToString().Trim();
        lblNombreApe.Text = Login.NombreU2 + " " + Login.Apellido2;
        var usuario = from u in db.AccFracc_UsuarioSistema
        where u.Nombre == Login.NombreU2 && u.Apellido == Login.Apellido2
        select u;
        int id = 0;
        foreach (var i in usuario)
        {
            id = i.IdUsuario;
        }
        try
        {
            {
                using (db = new ACCFRACCEntities())
                {

```

```

var updateUsu = db.AccFracc_UsuarioSistema.Find(id);
updateUsu.FechaEntrada = DateTime.Now;

db.Entry(updateUsu).State = System.Data.Entity.EntityState.Modified; // actualiza la
fecha y hora de inicio de sesion
db.SaveChanges();
}
}
catch (Exception ex)
{
    MessageBox.Show("" + ex);
}

if (labelaccess.Text.ToLower() == "1")//INGRESA CORRECTAMENTE Y SE
DEFINEN LOS ROLES
{
    btnCerrarSesion.Show();
    ADMIN.Visible = true;
    AccesoVehiculos.Visible = true;

}
else
{
    btnCerrarSesion.Show();
    ADMIN.Visible = false;
    AccesoVehiculos.Visible = true;
}
}
else
{
    menuStrip1.Hide();
    btnCerrarSesion.Hide();
}

}

```

```
private void AdministrarVigilantes_Click(object sender, EventArgs e)
{
    Usuarios av = new Usuarios();
    av.Show();
}
```

```
private void Boletines_Click(object sender, EventArgs e)
{
    Boletines bo = new Boletines();
    bo.Show();
}
```

```
private void Condominos_Click(object sender, EventArgs e)
{
    Condominos c = new Condominos();
    c.Show();
}
```

```
private void Historico_Click(object sender, EventArgs e)
{
    Historico h = new Historico();
    h.Show();
}
```

```
private void AccesoVehiculos_Click(object sender, EventArgs e)
{
    EntradasSalidas ave = new EntradasSalidas();
    ave.Show();
}
```

```
private void btnCerrarSesion_Click(object sender, EventArgs e) //cierra sesion del
usuario modificando la fecha y hora desalida del usuario
{
    using (db = new ACCFRACCEntities())
    {
```

```

DialogResult res = MessageBox.Show("¿" + lblNombreApe.Text + ", Estás seguro de salir?", "Advertencia", MessageBoxButtons.YesNo, MessageBoxIcon.Question, MessageBoxDefaultButton.Button2);
if (res == System.Windows.Forms.DialogResult.Yes)
{
    try
    {
        var usuario = from u in db.AccFracc_UsuarioSistema
        where u.Nombre == Login.NombreU2 && u.Apellido == Login.Apellido2
        select u;
        int id = 0;
        string name = "";
        foreach (var i in usuario)
        {
            id = i.IdUsuario;
            name = i.Nombre;
        }
        var updateUsu = db.AccFracc_UsuarioSistema.Find(id);
        updateUsu.FechaSalida = DateTime.Now;

        db.Entry(updateUsu).State = System.Data.Entity.EntityState.Modified;
        db.SaveChanges();

        bi.DescripcionBitacora = "El usuario " + name.Trim() + " cerró sesión";
        bi.FechaBitacora = DateTime.Now;
        bi.IdUsuario = id;
        db.AccFracc_Bitacora.Add(bi);
        db.SaveChanges();
    }
    catch (Exception ex)
    {
        el.Error = "" + ex;
        el.Modulo = "Menu Cerrar sesión";
        el.FechaErrorLog = DateTime.Now;
        db.AccFracc_ErrorLogs.Add(el);
        db.SaveChanges();
        MessageBox.Show("No se pudo cerrar sesión " + ex);
    }
}

```

```
Application.Exit();
```

```
private void entradasSalidastb_Click(object sender, EventArgs e)
{
    EntradasSalidas es = new EntradasSalidas();
    es.Show();
}
```

```
private void generarCódigoToolStripMenuItem_Click(object sender, EventArgs e)
{
    GnerarCodigo gc = new GnerarCodigo();
    gc.Show();
}
```

```
private void administrarFraccionamientosToolStripMenuItem_Click(object sender,
    EventArgs e)
{
    Fraccionamientos f = new Fraccionamientos();
    f.Show();
}
```

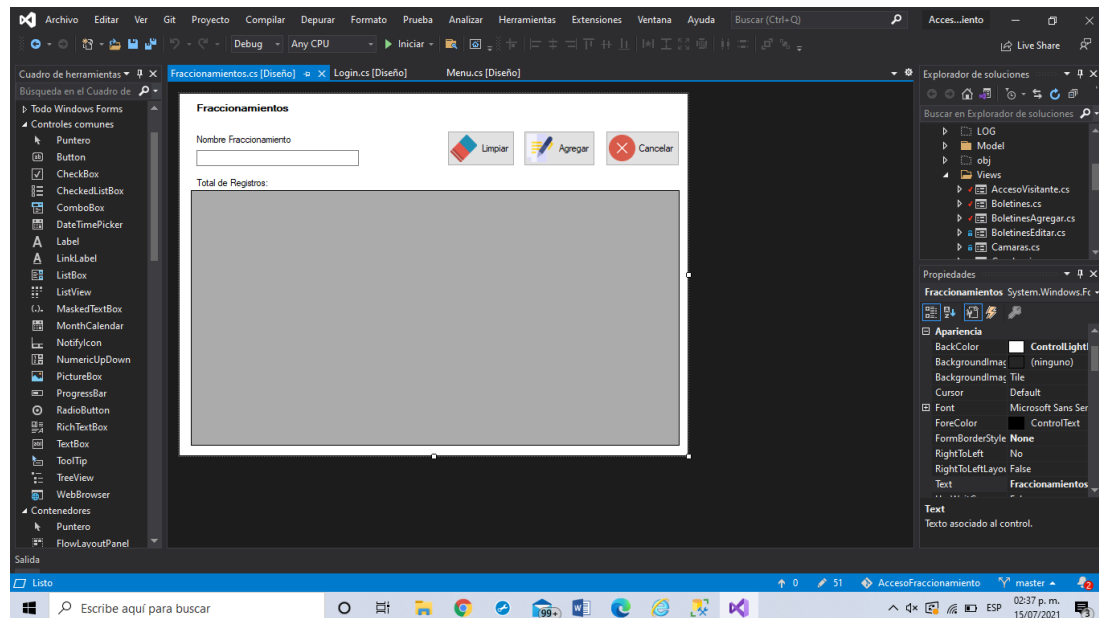
```
private void camarasToolStripMenuItem_Click(object sender, EventArgs e)
{
    Camaras cam = new Camaras();
    cam.Show();
}
```

```
private void visitasProgramadasYNoProgramadasToolStripMenuItem_Click(object
sender, EventArgs e)
{
    VisitaProgramadayNoProgramada vpynp = new VisitaProgramadayNoProgramada();
    vpynp.Show();
}

private void salidaVisitaNoProgramadaToolStripMenuItem_Click(object sender,
    EventArgs e)
{
}
```

## Administrador Fraccionamientos

La opción de Administrador nos desglosa un menú con lo siguiente, en esta se muestra una pantalla donde se muestra una lista con todos los fraccionamientos que ya están previamente guardados en nuestra base de datos, se puede hacer una búsqueda, Agregar y Borrar datos.



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using AccesoFraccionamiento.EntityFramework;
```

```
namespace AccesoFraccionamiento.Views
```

```
{
```

```
public partial class Fraccionamientos : Form
```

```
{
```

```
public static int IdUsuario2 { get; set; }
```

```
public static string Apellido2 { get; set; }
```

```
public static string Nombre2 { get; set; }
```

```

public static string Username2 { get; set; }
public static string Contraseña2 { set; get; }
public static int? IdRol2 { get; set; }
public static string Rol2 { get; set; }
public static bool? Estado2 { get; set; }
int ID = 0;
string iID = "";
ACCFRACCEntities db = new ACCFRACCEntities();
AccFracc_UsuarioSistema usu = new AccFracc_UsuarioSistema();
AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
public static string Nombre { set; get; }
public static string Apellido { set; get; }
public Fraccionamientos()
{
InitializeComponent();
}

private void btnAgregar_Click(object sender, EventArgs e)
{
//FraccionamientoAgregar fa = new FraccionamientoAgregar();
//fa.ShowDialog();
//var lusu = from u in db.AccFracc_Fraccionamientos

//      select new { u.Nombre, u.Estado };
//dgvFraccionamientos.DataSource = lusu.ToList();
//dgvFraccionamientos.AutoSizeColumnsMode =
DataGridViewAutoSizeColumnsMode.AllCells;
//labelInfo.Text = "Total de Registros: " + lusu.Count();
}

private void btnCancelar_Click(object sender, EventArgs e)
{
this.Close();
}

private void btnLimpiar_Click(object sender, EventArgs e)

```

```

{
//var lusu = from u in db.AccFracc_Fraccionamientos

//      select new { u.Nombre, u.Estado };
//dgvFraccionamientos.DataSource = lusu.ToList();
//labelInfo.Text = "Total de Registros: " + lusu.Count();
}

private void Fraccionamientos_Load(object sender, EventArgs e)
{
//var lusu = from u in db.AccFracc_Fraccionamientos

//      select new { u.Nombre, u.Estado };
//dgvFraccionamientos.DataSource = lusu.ToList();
//labelInfo.Text = "Total de Registros: " + lusu.Count();
}

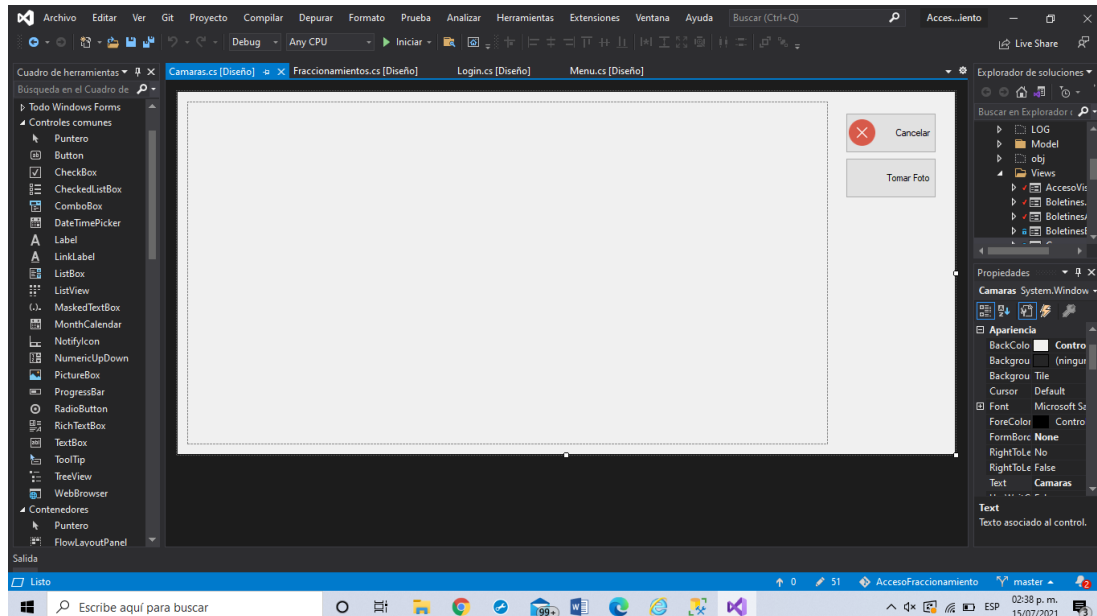
private void txtBuscarFrc_TextChanged(object sender, EventArgs e)
{
//var res = from u in db.AccFracc_Fraccionamientos
//      where u.Nombre.StartsWith(txtBuscarFrc.Text.Trim())
//      select new { u.Nombre, u.Estado }; ;
//dgvFraccionamientos.DataSource = res.ToList();
//labelInfo.Text = "Total de Registros: " + res.Count();
}
}
}

```



## Cámaras

En esta pantalla una vez funcionando nuestra aplicación se muestran las cámaras que están conectadas a nuestra aplicación, se muestra un recuadro con las tomas de las 5 cámaras conectadas a nuestro sistema con la opción de tomar foto.



```
using System;
using System.Windows.Forms;
using Ozeki.Media.Video.Controls;
using Ozeki.Media.IPCamera;
using Ozeki.Camera;
using Ozeki.Media;
using Ozeki.Media.MediaHandlers;

namespace AccesoFraccionamiento.Views
{
    public partial class Camaras : Form
    {
        private Ozeki.Camera.IIPCCamera _camera;
        private Ozeki.Media.DrawingImageProvider _imageProvider = new
            Ozeki.Media.DrawingImageProvider();
        private Ozeki.Media.MediaConnector _connector = new
            Ozeki.Media.MediaConnector();
        private Ozeki.Media.VideoViewerWF _videoViewerWF1;
        private SnapshotHandler _snapshot;
```

```

public Camaras()
{
    InitializeComponent();
    // Create video viewer UI control
    _videoViewerWF1 = new Ozeki.Media.VideoViewerWF();
    _videoViewerWF1.Size = panel1.Size;
    panel1.Controls.Add(_videoViewerWF1);
    // Bind the camera image to the UI control
    _videoViewerWF1.SetImageProvider(_imageProvider);
    _snapshot = new SnapshotHandler();
}

private void Camaras_Load(object sender, EventArgs e)
{
    try
    {
        _camera = Ozeki.Camera.IPCameraFactory.GetCamera("192.168.1.108", "admin",
"C0yt3c125");
        _connector.Connect(_camera.VideoChannel, _imageProvider);
        _connector.Connect(_camera.VideoChannel, _snapshot);
        _camera.Start();
        _videoViewerWF1.Start();
    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex);
    }
}

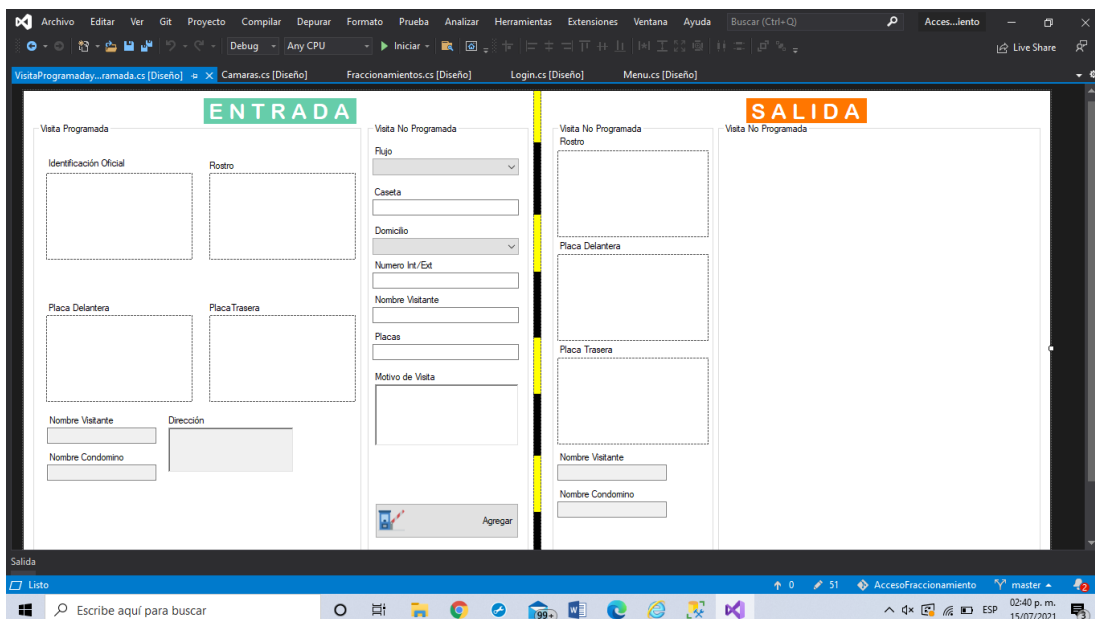
private void connectBtn_Click(object sender, EventArgs e)
{
    var date = DateTime.Now.Year + "-" + DateTime.Now.Month + "-" +
DateTime.Now.Day + "-" +
DateTime.Now.Hour + "H" + DateTime.Now.Minute + "m" + DateTime.Now.Second +
"s";
    string ruta = @"C:\Users\AlbertoMirandaSantos\Pictures\Capture_" + date + ".jpg";
}

```

## Visitas Programadas y No Programadas

En esta opción de nuestro menú se muestra esta pantalla dividida en dos donde se muestran diferentes opciones, en la pantalla se muestran diferentes cámaras que se encuentran en nuestro monitor de entrada, cámara de Identificación, Cámara de Rostro, Cámara de Placas Delanteras y Traseras, opciones para ingresar el Nombre de Visitante, Nombre de Condominio, Dirección y Motivo de la Visita.

En la pantalla de Salida Se muestran las Cámaras de Rostro, Placa Delantera, Placa Trasera, Nombre de Visitante y De Condominio



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Linq;  
using System.Windows.Forms;  
using Ozeki.Media.Video.Controls;  
using Ozeki.Media.IPCamera;  
using Ozeki.Camera;  
using Ozeki.Media;  
using Ozeki.Media.MediaHandlers;  
using System.Drawing;  
using AccesoFraccionamiento.EntityFrameworks;
```

```
namespace AccesoFraccionamiento.Views
```

```

{
public partial class VisitaProgramadayNoProgramada : Form
{
private Ozeki.Camera.IIPCamera _camera;
private Ozeki.Media.DrawingImageProvider _imageProvider = new
Ozeki.Media.DrawingImageProvider();
private Ozeki.Media.MediaConnector _connector = new
Ozeki.Media.MediaConnector();
private Ozeki.Media.VideoViewerWF _videoViewerWF1;
private Ozeki.Media.VideoViewerWF _videoViewerWF2;
private Ozeki.Media.VideoViewerWF _videoViewerWF3;
private Ozeki.Media.VideoViewerWF _videoViewerWF4;
private SnapshotHandler _snapshot;
AccFRACCEntities db = new AccFRACCEntities();
AccFracc_VisitaNoProgramada vnp = new AccFracc_VisitaNoProgramada();
AccFracc_Calle calle = new AccFracc_Calle();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
public VisitaProgramadayNoProgramada()
{
InitializeComponent();
//setear los estados de los radiobuttons
// Create video viewer UI control
_videoViewerWF1 = new Ozeki.Media.VideoViewerWF();
_videoViewerWF2 = new Ozeki.Media.VideoViewerWF();
_videoViewerWF3 = new Ozeki.Media.VideoViewerWF();
_videoViewerWF4 = new Ozeki.Media.VideoViewerWF();
_videoViewerWF1.Size = pnlIdentificacion.Size;
_videoViewerWF2.Size = pnlRostro.Size;
_videoViewerWF3.Size = pnlCarroceria.Size;
_videoViewerWF4.Size = pnlPlacas.Size;
pnlCarroceria.Controls.Add(_videoViewerWF3);
pnlIdentificacion.Controls.Add(_videoViewerWF1);
pnlPlacas.Controls.Add(_videoViewerWF4);
pnlRostro.Controls.Add(_videoViewerWF2);
// Bind the camera image to the UI control
_videoViewerWF1.SetImageProvider(_imageProvider);

```

```

_videoViewerWF2.SetImageProvider(_imageProvider);
_videoViewerWF3.SetImageProvider(_imageProvider);
_videoViewerWF4.SetImageProvider(_imageProvider);
_snapshot = new SnapshotHandler();

}

private void VisitaProgramadayNoProgramada_Load(object sender, EventArgs e)
{
    var lcalles = from c in db.AccFracc_Calle
    select new { c.Nombre };
    cmbxDomicilioNP.DataSource = lcalles.ToList();
    cmbxDomicilioNP.DisplayMember = "Nombre";
    try
    {
        _camera = Ozeki.Camera.IPCameraFactory.GetCamera("192.168.1.108", "admin",
"C0yt3c125");
        _connector.Connect(_camera.VideoChannel, _imageProvider);
        _connector.Connect(_camera.VideoChannel, _snapshot);
        _camera.Start();
        _videoViewerWF1.Start();
        _videoViewerWF2.Start();
        _videoViewerWF3.Start();
        _videoViewerWF4.Start();
    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex);
    }
}

private void btnAgregar_Click(object sender, EventArgs e)
{
    string Nombre = Login.NombreU2;
    string Apellido = Login.Apellido2;
    var usr = from u in db.AccFracc_UsuarioSistema
    where u.Nombre == Nombre.Trim() && u.Apellido == Apellido.Trim()

```

```

select u;
int idu = 0;
foreach (var i in usr)
{
    idu = i.IdUsuario;
}
try
{
    var date = DateTime.Now.Year + "-" + DateTime.Now.Month + "-" +
    DateTime.Now.Day + "-" +
    DateTime.Now.Hour + "H" + DateTime.Now.Minute + "m" + DateTime.Now.Second +
    "s";
    string ruta = @"C:\Users\AlbertoMirandaSantos\Pictures\Capture_" + date + ".jpg";
    vnp.Flujo =
    cmbxFlujoNP.GetItemText(cmbxFlujoNP.SelectedItem.ToString()).Trim();
    vnp.Caseta = txtCasetaNP.Text.Trim();
    vnp.DireccionVisita = cmbxDomicilioNP.Text;
    vnp.Numero = textNumero.Text.Trim();
    vnp.NombreVisitante = txtNombreVisiNP.Text.Trim();
    vnp.Placas = txtPlacasNP.Text.Trim();
    vnp.MotivoVisita = txtMotivoNP.Text.Trim();
    vnp.Vigilante = (string)(Login.NombreU2 + " " + Login.Apellido2).Trim();
    vnp.PathFotoCarroseria = ruta.Trim();
    vnp.PathFotoIne = ruta.Trim();
    vnp.PathFotoPlacas = ruta.Trim();
    vnp.PathFotoRostro = ruta.Trim();
    vnp.FechaEntrada = DateTime.Now;
    db.AccFracc_VisitaNoProgramada.Add(vnp);
    db.SaveChanges();
    MessageBox.Show(vnp.NombreVisitante.Trim() + " ha entrado por " +
    vnp.Caseta.Trim() + " y se dirige a " + vnp.DireccionVisita.Trim() + " " +
    vnp.Numero.Trim() + " Agregado.", "Información", MessageBoxButtons.OK,
    MessageBoxIcon.Information);
    bi.DescripcionBitacora = "Se agregó visita no programada exitosamente";
    bi.FechaBitacora = DateTime.Now;
    bi.IdUsuario = idu;
    db.AccFracc_Bitacora.Add(bi);
}

```

```

db.SaveChanges();
var    snapShotImage    =    _snapshot.TakeSnapshot().ToImage()    as
System.Drawing.Image;
snapShotImage.Save(ruta);
//agregar elmetodode insercion alabd
}
catch(Exception ex)
{
el.Error = "" + ex;
el.Modulo = "Usuarios Add";
el.FechaErrorLog = DateTime.Now;
db.AccFracc_ErrorLogs.Add(el);
db.SaveChanges();
}

}

private void btnCerrar_Click(object sender, EventArgs e)
{
this.Close();
}

private void groupBox1_Enter(object sender, EventArgs e)
{

}

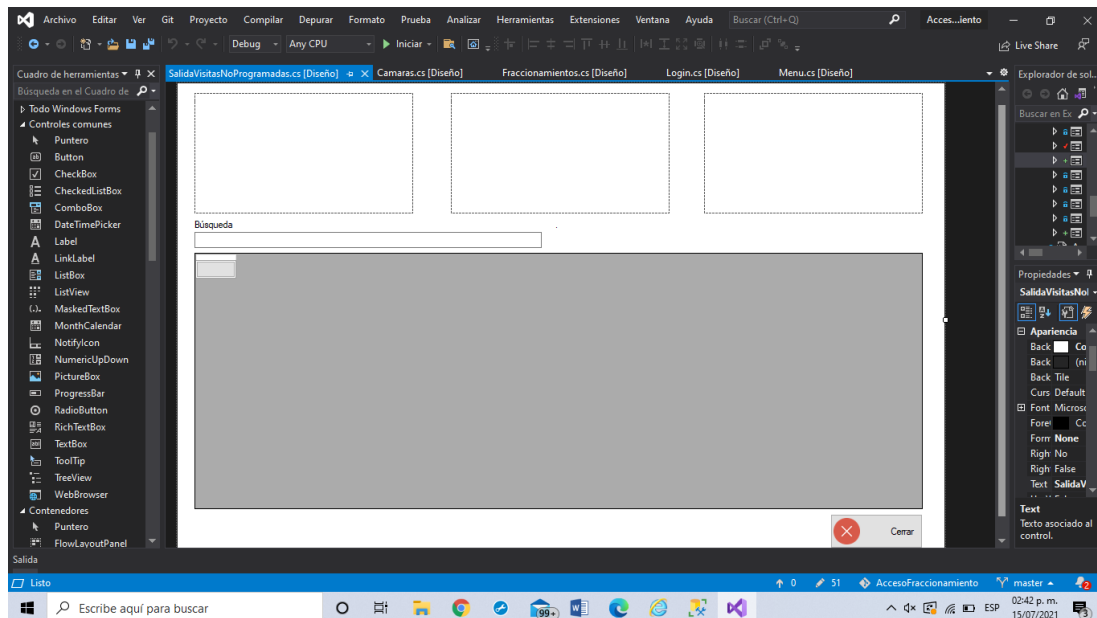
}

}

```

## Salidas Visitas No Programadas

En esta pantalla una vez en funcionamiento la Aplicación muestra las Cámaras de Rostro y Placas, Donde se encuentra el Modulo en la aparte de abajo se muestra una lista conectada a nuestra base de datos donde muestra una lista de los visitantes que se encuentran dentro del condominio en ese momento



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Linq;  
using System.Windows.Forms;  
using Ozeki.Media.Video.Controls;  
using Ozeki.Media.IPCamera;  
using Ozeki.Camera;  
using Ozeki.Media;  
using Ozeki.Media.MediaHandlers;  
using System.Drawing;  
using AccesoFraccionamiento.EntityFrameworks;  
using System.Threading;
```

```
namespace AccesoFraccionamiento.Views
```

```
{
```

```
public partial class SalidaVisitasNoProgramadas : Form
```



```

{
string iID = "";
int ID = 0;
private Ozeki.Camera.IIPCamera _camera;
private Ozeki.Media.DrawingImageProvider _imageProvider = new
Ozeki.Media.DrawingImageProvider();
private Ozeki.Media.MediaConnector _connector = new
Ozeki.Media.MediaConnector();
private Ozeki.Media.VideoViewerWF _videoViewerWF2;
private Ozeki.Media.VideoViewerWF _videoViewerWF3;
private Ozeki.Media.VideoViewerWF _videoViewerWF4;
private SnapshotHandler _snapshot;
ACCFRACCEntities db = new ACCFRACCEntities();
AccFracc_VisitaNoProgramada vnp = new AccFracc_VisitaNoProgramada();
AccFracc_Calle calle = new AccFracc_Calle();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
public SalidaVisitasNoProgramadas()
{
InitializeComponent();
//setear los estados de los radiobuttons
// Create video viewer UI control
_videoViewerWF2 = new Ozeki.Media.VideoViewerWF();
_videoViewerWF3 = new Ozeki.Media.VideoViewerWF();
_videoViewerWF4 = new Ozeki.Media.VideoViewerWF();
_videoViewerWF2.Size = pnlRostro.Size;
_videoViewerWF3.Size = pnlCarroceria.Size;
_videoViewerWF4.Size = pnlPlacas.Size;
pnlCarroceria.Controls.Add(_videoViewerWF3);
pnlPlacas.Controls.Add(_videoViewerWF4);
pnlRostro.Controls.Add(_videoViewerWF2);
// Bind the camera image to the UI control
_videoViewerWF2.SetImageProvider(_imageProvider);
_videoViewerWF3.SetImageProvider(_imageProvider);
_videoViewerWF4.SetImageProvider(_imageProvider);
_snapshot = new SnapshotHandler();
lblVigi.Text = Login.NombreU2 + " " + Login.Apellido2;

```

```
}
```

```
private void SalidaVisitasNoProgramadas_Load(object sender, EventArgs e)
{
    var lcalles = from c in db.AccFracc_VisitaNoProgramada
    where c.FechaSalida == null
    select new { Consecutivo = c.IdVisitaNoProgramada , c.NombreVisitante,
    c.DireccionVisita, c.Numero, c.MotivoVisita, c.Placas };
    dgvSalida.DataSource = lcalles.ToList();
    try
    {
        _camera = Ozeki.Camera.IPCameraFactory.GetCamera("192.168.1.108", "admin",
        "C0yt3c125");
        _connector.Connect(_camera.VideoChannel, _imageProvider);
        _connector.Connect(_camera.VideoChannel, _snapshot);
        _camera.Start();
        _videoViewerWF4.Start();
        _videoViewerWF2.Start();
        _videoViewerWF3.Start();
    }
    catch (Exception ex)
    {
        MessageBox.Show("" + ex);
    }
}
```

```
private void dgvSalida_CellPainting(object sender,
DataGridViewCellPaintingEventArgs e)
{
    if (e.RowIndex < 0) return;
    if (e.ColumnIndex == 0)
    {
        e.Paint(e.CellBounds, DataGridViewPaintParts.All);
        var w = Properties.Resources.barra.Width;
        var h = Properties.Resources.barra.Height;
        var x = e.CellBounds.Left + (e.CellBounds.Width - w) / 2;
        var y = e.CellBounds.Top + (e.CellBounds.Height - h) / 2;
```

```

e.Graphics.DrawImage(Properties.Resources.barra,
System.Drawing.Rectangle(x, y, w, h)); e.Handled = true;
}
}

private void dgvSalida_CellContentClick(object sender, DataGridViewCellEventArgs
e)
{
try
{
if (e.RowIndex < 0 || e.ColumnIndex < 0)
return;
switch (e.ColumnIndex)
{
case 0:
iID
=
(dgvSalida.Rows[e.RowIndex].Cells["Consecutivo"].FormattedValue).ToString();
ID = int.Parse(iID);
var updateVisi = db.AccFracc_VisitaNoProgramada.Find(ID);
DateTime salida = DateTime.Now;
updateVisi.FechaSalida = salida;
db.Entry(updateVisi).State = System.Data.Entity.EntityState.Modified;
db.SaveChanges();
MessageBox.Show(updateVisi.NombreVisitante.Trim() + " ha salido de " +
updateVisi.DireccionVisita.Trim() + " " + updateVisi.Numero.Trim() + ".",
"Información", MessageBoxButtons.OK, MessageBoxIcon.Information);

break;
}
var lcalles = from c in db.AccFracc_VisitaNoProgramada
where c.FechaSalida == null
select new { Consecutivo = c.IdVisitaNoProgramada, c.NombreVisitante,
c.DireccionVisita, c.Numero, c.MotivoVisita, c.Placas };
dgvSalida.DataSource = lcalles.ToList();
db = new ACCFRACCEntities();
}
catch (Exception ex)

```

```

{
    MessageBox.Show(ex.Message);
}
}

private void txtBusqueda_TextChanged(object sender, EventArgs e)
{
    if (txtBusqueda.Text == "")
    {
        var lcalles = from c in db.AccFracc_VisitaNoProgramada
        where
            c.FechaSalida == null
        select new { Consecutivo = c.IdVisitaNoProgramada, c.NombreVisitante,
            c.DireccionVisita, c.Numero, c.MotivoVisita, c.Placas };
        dgvSalida.DataSource = lcalles.ToList();
    }
    else
    {
        var lcalles = from c in db.AccFracc_VisitaNoProgramada
        where
            (c.NombreVisitante.Contains(txtBusqueda.Text.Trim()) ||
            c.Placas.Contains(txtBusqueda.Text.Trim()))
            && c.FechaSalida == null
        select new { Consecutivo = c.IdVisitaNoProgramada, c.NombreVisitante,
            c.DireccionVisita, c.Numero, c.MotivoVisita, c.Placas };
        dgvSalida.DataSource = lcalles.ToList();
        private void button1_Click(object sender, EventArgs e)
        {
            Thread.Sleep(2000);
            this.Close();
        }
        private void dgvSalida_CellFormatting(object sender,
            DataGridViewCellFormattingEventArgs e)
        {
            if (e.Value != null)
            }
        }
    }
}

```

## Usuarios

En esta pantalla se muestra una tabla de todos los Condominios Registrados en nuestra Aplicación, se muestran los Usuarios, Nombre, Apellido, Rol, y Estado, así mismo muestra las opciones de Activar, Limpiar tabla, Agregar Usuario y Eliminar



**Usuarios**

Nombre Usuario:

Activar Limpiar Agregar Cancelar

Total de Registros: 15

		IdUsuario	Nombre	Apellido	Rol	Estado
		1	Alberto	Miranda	Administrador	Activo
		6	Pablo	Mamoli	Administrador	Activo
		11	Jonas	Hernandez Mend.	Administrador	Activo
		14	Tonio	Perez	Vigilante	Activo
		29	zazaza	asdad	Vigilante	Activo
		33	wenwer	wenwer	Vigilante	Activo
		35	veveve	veveve	Vigilante	Activo
		37	zazazazazazaza	zazazazazaza	Vigilante	Activo
		46	wwwwww	asdasda	Vigilante	Activo
		52	sdwd	sdad	Vigilante	Activo
		55	Mario	Santos	Vigilante	Activo
		56	sadasd	asdasd	Vigilante	Activo
		57	sasas	sasas	Vigilante	Activo
		58	daad	asdasdz	Vigilante	Activo

```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using AccesoFraccionamiento.EntityFrameworks;
```

```
namespace AccesoFraccionamiento.Views  
{  
    public partial class Usuarios : Form  
    {  
        public static int IdUsuario2 { get; set; }  
        public static string Apellido2 { get; set; }  
        public static string Nombre2 { get; set; }  
        public static string Username2 { get; set; }
```

```

public static string Contraseña2 { set; get; }
public static int? IdRol2 { get; set; }
public static string Rol2 { get; set; }
public static bool? Estado2 { get; set; }
int ID = 0;
string iID = "";
ACCFRACCEntities db = new ACCFRACCEntities();
AccFracc_UsuarioSistema usu = new AccFracc_UsuarioSistema();
AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
public static string Nombre { set; get; }
public static string Apellido { set; get; }
public Usuarios()
{
    InitializeComponent();
}

private void btnCancelar_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnAgregar_Click(object sender, EventArgs e)
{
    UsuarioAgregar ua = new UsuarioAgregar();
    ua.ShowDialog();
    var lusu = from u in db.AccFracc_UsuarioSistema
    where u.Estado == true
    select new { u.IdUsuario, u.Nombre, u.Apellido, u.Rol, u.Estado };
    dgvUsuarios.DataSource = lusu.ToList();
    labelInfo.Text = "Total de Registros: " + lusu.Count();
}

private void Usuarios_Load(object sender, EventArgs e)
{
    var lusu = from u in db.AccFracc_UsuarioSistema
    where u.Estado == true

```

```

select new { u.IdUsuario, u.Nombre, u.Apellido, u.Rol, u.Estado };
dgvUsuarios.DataSource = lusu.ToList();
labelInfo.Text = "Total de Registros: " + lusu.Count();
}

```

```

private void dgvUsuarios_CellContentClick(object sender,
DataGridViewCellEventArgs e)
{
db = new ACCFRACCEntities();
Nombre = Login.NombreU2;
Apellido = Login.Apellido2;
var usr = from u in db.AccFracc_UsuarioSistema
where u.Nombre == Nombre.Trim() && u.Apellido == Apellido.Trim()
select u;
int idu = 0;
foreach (var i in usr)
{
idu = i.IdUsuario;
}
db = new ACCFRACCEntities();
try
{
if (e.RowIndex < 0 || e.ColumnIndex < 0)
return;
switch (e.ColumnIndex)
{
case 0:
ilID = (dgvUsuarios.Rows[e.RowIndex].Cells["IdUsuario"].FormattedValue).ToString();
ID = int.Parse(ilID);
var updateUsu = db.AccFracc_UsuarioSistema.Find(ID);
IdUsuario2 = ID;
Apellido2 = updateUsu.Apellido;
Nombre2 = updateUsu.Nombre;
Username2 = updateUsu.NombreUsuario;
Contrasena2 = updateUsu.Contrasena;
IdRol2 = updateUsu.IdRol;

```

```

Rol2 = updateUsu.Rol;
Estado2 = updateUsu.Estado;
UsuarioEditar tse = new UsuarioEditar();
tse.ShowDialog();
break;
case 1:
    iID = (dgvUsuarios.Rows[e.RowIndex].Cells["IdUsuario"].FormattedValue).ToString();
    ID = int.Parse(iID);
    var deleteu = db.AccFracc_UsuarioSistema.Find(ID);
    DialogResult res = MessageBox.Show("¿Seguro de eliminar " +
        deleteu.Nombre.Trim() + " " + deleteu.Apellido + "?", "Advertencia",
        MessageBoxButtons.YesNo, MessageBoxIcon.Question,
        MessageBoxDefaultButton.Button2);
    if (res == System.Windows.Forms.DialogResult.Yes)
    {
        try
        {
            deleteu.Estado = false;
            db.Entry(deleteu).State = System.Data.Entity.EntityState.Modified;
            db.SaveChanges();
            MessageBox.Show("Registro Eliminado Exitosamente", "Advertencia",
                MessageBoxButtons.OK, MessageBoxIcon.Exclamation);

            bi.DescripcionBitacora = "Se eliminó el usuario " + deleteu.Nombre.Trim();
            bi.FechaBitacora = DateTime.Now;
            bi.IdUsuario = idu;
            db.AccFracc_Bitacora.Add(bi);
            db.SaveChanges();
        }
        catch (Exception ex)
        {
            el.Error = "" + ex;
            el.Modulo = "Usuarios Eliminar";
            el.FechaErrorLog = DateTime.Now;
            db.AccFracc_ErrorLogs.Add(el);
            db.SaveChanges();
        }
    }

```



```

MessageBox.Show("No se pudo eliminar el registro intentalo de nuevo " + ex);
}

}
break;
}

var lusu = from u in db.AccFracc_UsuarioSistema
where u.Estado == true
select new { u.IdUsuario, u.Nombre, u.Apellido, u.Rol, u.Estado };
dgvUsuarios.DataSource = lusu.ToList();
labelInfo.Text = "Total de Registros: " + lusu.Count();
db = new ACCFRACCEntities();
}
catch (Exception ex)
{
    MessageBox.Show(ex.Message);
}
}

private void txtBuscarUsuario_TextChanged(object sender, EventArgs e)
{
    var res = from u in db.AccFracc_UsuarioSistema
where u.Nombre.StartsWith(txtBuscarUsuario.Text.Trim()) && u.Estado == true
select new { u.IdUsuario, u.Nombre, u.Apellido, u.Rol, u.Estado }; ;
dgvUsuarios.DataSource = res.ToList();
labelInfo.Text = "Total de Registros: " + res.Count();
}

private void btnLimpiar_Click(object sender, EventArgs e)
{
    txtBuscarUsuario.Text = "";
    var lusu = from u in db.AccFracc_UsuarioSistema
where u.Estado == true
select new { u.IdUsuario, u.Nombre, u.Apellido, u.Rol, u.Estado };
dgvUsuarios.DataSource = lusu.ToList();
labelInfo.Text = "Total de Registros: " + lusu.Count();
}

```

```

private void dgvUsuarios_CellPainting(object sender,
DataGridViewCellPaintingEventArgs e) //agregar una imagen al boton del grid
{
    if (e.RowIndex < 0) return;
    if (e.ColumnIndex == 0)
    {
        e.Paint(e.CellBounds, DataGridViewPaintParts.All);
        var w = Properties.Resources.lapiz.Width;
        var h = Properties.Resources.lapiz.Height;
        var x = e.CellBounds.Left + (e.CellBounds.Width - w) / 2;
        var y = e.CellBounds.Top + (e.CellBounds.Height - h) / 2;
        e.Graphics.DrawImage(Properties.Resources.lapiz, new Rectangle(x, y, w, h));
        e.Handled = true;
    }
    if (e.ColumnIndex == 1)
    {
        e.Paint(e.CellBounds, DataGridViewPaintParts.All);
        var w = Properties.Resources.trash.Width;
        var h = Properties.Resources.trash.Height;
        var x = e.CellBounds.Left + (e.CellBounds.Width - w) / 2;
        var y = e.CellBounds.Top + (e.CellBounds.Height - h) / 2;
        e.Graphics.DrawImage(Properties.Resources.trash, new Rectangle(x, y, w, h));
        e.Handled = true;
    }
}

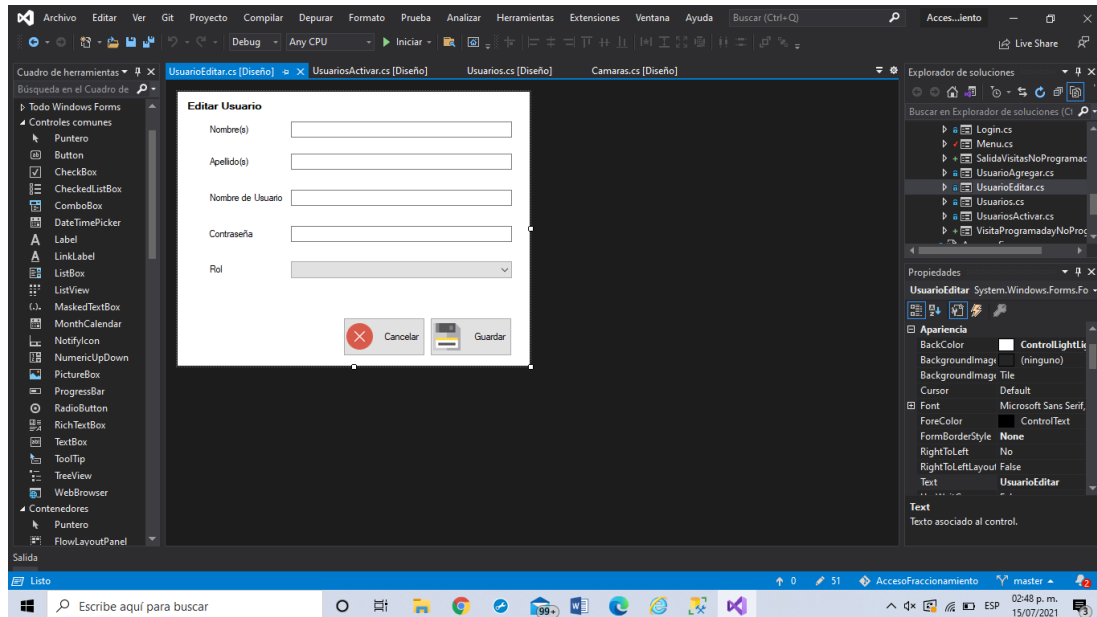
private void btnActivar_Click(object sender, EventArgs e)
{
    this.Close();
    UsuariosActivar ua = new UsuariosActivar();
    ua.ShowDialog();
}

private void dgvUsuarios_CellFormatting(object sender,
DataGridViewCellFormattingEventArgs e)
{
    if (this.dgvUsuarios.Columns[e.ColumnIndex].Name == "Estado")

```

## Usuarios Editar

Esta es la pantalla donde se Editan los campos de los usuarios, El Nombre, Apellido, Nombre de Usuario, y contraseña, con la opción de cancelar o guardar los datos.



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using AccesoFraccionamiento.EntityFrameworks;
```

```
namespace AccesoFraccionamiento.Views  
{  
    public partial class UsuarioEditor : Form  
    {  
        public static int IdUsuario2 { get; set; }  
        public static string Apellido2 { get; set; }  
        public static string Nombre2 { get; set; }  
        public static string Username2 { get; set; }  
        public static string Contraseña2 { set; get; }  
    }  
}
```

```

public static int? IdRol2 { get; set; }
public static string Rol2 { get; set; }
public static bool? Estado2 { get; set; }
int id = Usuarios.IdUsuario2;
ACCFRACCEntities db = new ACCFRACCEntities();
AccFracc_UsuarioSistema usu = new AccFracc_UsuarioSistema();
AccFracc_Roles rol = new AccFracc_Roles();
AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
public static string Nombre { set; get; }
public static string Apellido { set; get; }
public UsuarioEditar()
{
    InitializeComponent();
}

private void btnCancelar_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnAgregar_Click(object sender, EventArgs e)
{
    if (txtApellidos.Text == "" ||
        txtNombre.Text == "" ||
        txtPassword.Text == "" ||
        txtUsuarioName.Text == "")
    {
        MessageBox.Show("Debes llenar todos los campos");
    }
    else
    {
        using (db = new ACCFRACCEntities())
        {
            Nombre = Login.NombreU2;
            Apellido = Login.Apellido2;
            var usr = from u in db.AccFracc_UsuarioSistema

```

```

where u.Nombre == Nombre.Trim() && u.Apellido == Apellido.Trim()
select u;
int idu = 0;
foreach (var i in usr)
{
    idu = i.IdUsuario;
}
try
{
    var updateUsu = db.AccFracc_UsuarioSistema.Find(id);
    updateUsu.Nombre = txtNombre.Text.Trim();
    updateUsu.Apellido = txtApellidos.Text.Trim();
    updateUsu.NombreUsuario = txtUsuarioName.Text.Trim();
    updateUsu.Contrasena = txtPassword.Text.Trim();
    updateUsu.IdRol = (int)cboxRol.SelectedValue;
    updateUsu.Rol = cboxRol.GetItemText(cboxRol.SelectedItem);
    updateUsu.Estado = true;

    // Mark as Changed
    db.Entry(updateUsu).State = System.Data.Entity.EntityState.Modified;
    db.SaveChanges();
    MessageBox.Show("Registro Editado Exitosamente", "Información",
    MessageBoxButtons.OK, MessageBoxIcon.Information);

    bi.DescripcionBitacora = "Se edito el usuario " + updateUsu.Nombre.Trim();
    bi.FechaBitacora = DateTime.Now;
    bi.IdUsuario = idu;
    db.AccFracc_Bitacora.Add(bi);
    db.SaveChanges();
}
catch (Exception ex)
{
    el.Error = "" + ex;
    el.Modulo = "Usuarios Edit";
    el.FechaErrorLog = DateTime.Now;
    db.AccFracc_ErrorLogs.Add(el);
    db.SaveChanges();
}

```

```
MessageBox.Show("No se pudo editar el registro intentalo de nuevo " + ex);  
}
```

```
this.Close();  
}  
}
```

```
}
```

```
private void UsuarioEditar_Load(object sender, EventArgs e)
```

```
{
```

```
var lrol = from r in db.AccFracc_Roles
```

```
select r;
```

```
cboxRol.DataSource = lrol.ToList();
```

```
cboxRol.DisplayMember = "Rol";
```

```
cboxRol.ValueMember = "IdRol";
```

```
id = Usuarios.IdUsuario2;
```

```
string nom = Usuarios.Nombre2.Trim();
```

```
string ape = Usuarios.Apellido2.Trim();
```

```
string uname = Usuarios.Username2.Trim();
```

```
string contra = Usuarios.Contrasena2.Trim();
```

```
int? idrol = Usuarios.IdRol2==null?1: Usuarios.IdRol2;
```

```
string rol = Usuarios.Rol2;
```

```
bool? estado = Usuarios.Estado2;
```

```
txtApellidos.Text = ape.Trim();
```

```
txtNombre.Text = nom.Trim();
```

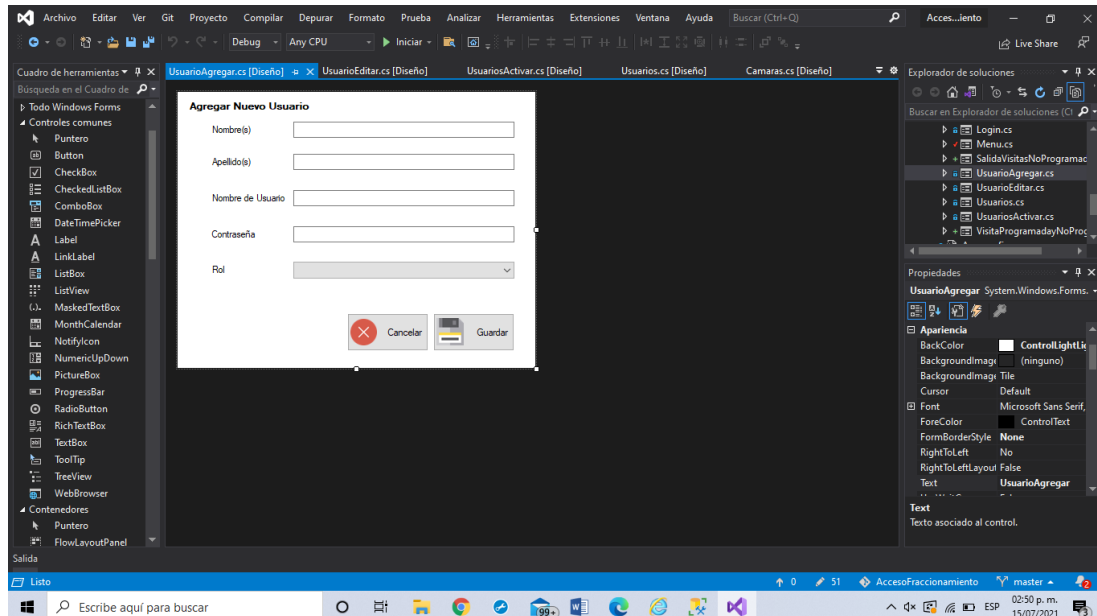
```
txtPassword.Text = contra.Trim();
```

```
txtUsuarioName.Text = uname.Trim();
```

```
cboxRol.SelectedValue = idrol;
```

## Usuarios Agregar

En esta pantalla se muestra una lista con campos para Agregar los Usuarios, se encuentra Nombre, Apellido, Nombre de Usuario y Contraseña, así mismo se encuentran los botones de Cancelar y Guardar.



```
using System;  
using System.Collections.Generic;  
using System.ComponentModel;  
using System.Data;  
using System.Drawing;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using System.Windows.Forms;  
using AccesoFraccionamiento.EntityFrameworks;
```

```
namespace AccesoFraccionamiento.Views  
{
```

```
public partial class UsuarioAgregar : Form  
{  
    ACCFRACCEntities db = new ACCFRACCEntities();  
    AccFracc_UsuarioSistema usu = new AccFracc_UsuarioSistema();  
    AccFracc_Roles rol = new AccFracc_Roles();
```

```

AccFracc_ErrorLogs el = new AccFracc_ErrorLogs();
AccFracc_Bitacora bi = new AccFracc_Bitacora();
public static string Nombre { set; get; }
public static string Apellido { set; get; }
public UsuarioAgregar()
{
InitializeComponent();
}

private void btnAgregar_Click(object sender, EventArgs e)
{
if (txtApellidos.Text == "" ||
txtNombre.Text == "" ||
txtPassword.Text == "" ||
txtUsuarioName.Text == "")
{
MessageBox.Show("Debes llenar todos los campos");
}
else
{
using (db = new ACCFRACCEntities())
{
Nombre = Login.NombreU2;
Apellido = Login.Apellido2;
var usr = from u in db.AccFracc_UsuarioSistema
where u.Nombre == Nombre.Trim() && u.Apellido == Apellido.Trim()
select u;
int idu = 0;
foreach (var i in usr)
{
idu = i.IdUsuario;
}
try
{
usu.Nombre = txtNombre.Text.Trim();
usu.Apellido = txtApellidos.Text.Trim();
usu.NombreUsuario = txtUsuarioName.Text.Trim();

```



```

usu.Contrasena = txtPassword.Text.Trim();
usu.IdRol = (int)cboxRol.SelectedValue;
usu.Rol = cboxRol.GetItemText(cboxRol.SelectedItem);
usu.Estado = true;
db.AccFracc_UsuarioSistema.Add(usu);
db.SaveChanges();
txtApellidos.Text = "";
txtNombre.Text = "";
txtPassword.Text = "";
txtUsuarioName.Text = "";
cboxRol.SelectedValue = 1;
MessageBox.Show(usu.Nombre.Trim() + " " + usu.Apellido.Trim() + " Agregado.",
"Información", MessageBoxButtons.OK, MessageBoxIcon.Information);

bi.DescripcionBitacora = "Se agregó " + usu.Nombre.Trim() + " exitosamente";
bi.FechaBitacora = DateTime.Now;
bi.IdUsuario = idu;
db.AccFracc_Bitacora.Add(bi);
db.SaveChanges();
}
catch (Exception ex)
{
    el.Error = "" + ex;
    el.Modulo = "Usuarios Add";
    el.FechaErrorLog = DateTime.Now;
    db.AccFracc_ErrorLogs.Add(el);
    db.SaveChanges();
}

}
}

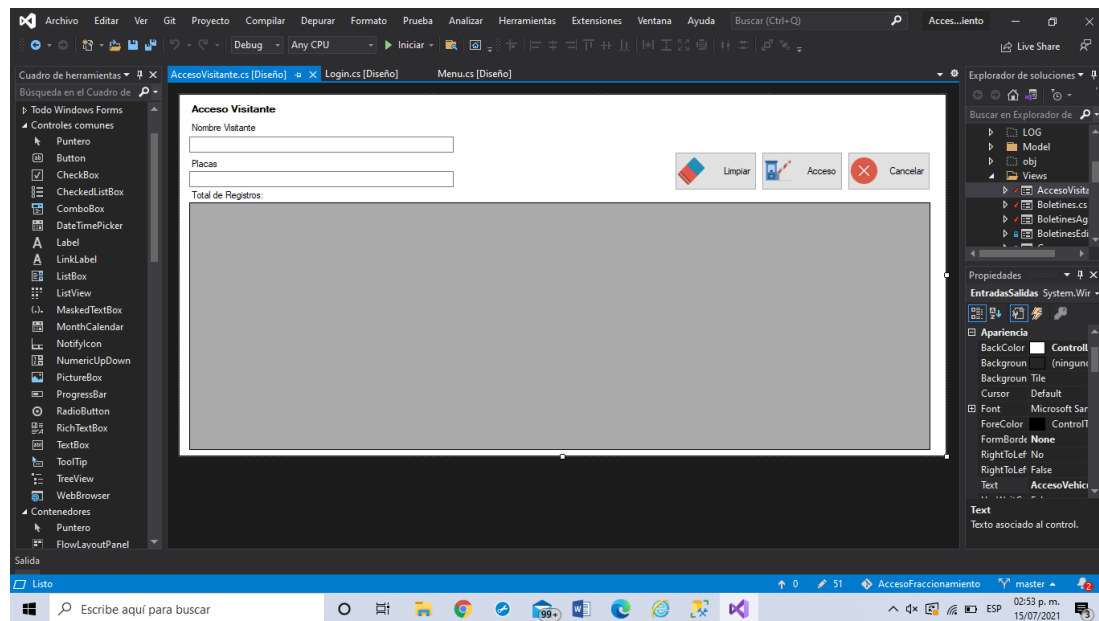
db = new ACCFRACCEntities();
}

private void UsuarioAgregar_Load(object sender, EventArgs e)
{

```

## Acceso a Vehículos

La funcionalidad de la siguiente pantalla es para dar acceso manualmente a los vehículos visitantes sin registro previo, Se encuentran la opciones para agregar el Nombre del Visitante y la Matrícula del vehículo, De igual forma en panel Gris de la parte inferior se muestra una lista de los Vehículos visitantes están dentro del Fraccionamiento, Muestra un botón de Limpiar pantalla, dar Acceso y Cancelar.



```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using AccesoFraccionamiento.EntityFrameworks;

namespace AccesoFraccionamiento.Views
{
    public partial class EntradasSalidas : Form
    {
        ACCFRACCEntities db = new ACCFRACCEntities();
        AccFracc_Visitante visita = new AccFracc_Visitante();
    }
}
```

```

public EntradasSalidas()
{
    InitializeComponent();
}

private void btnCancelar_Click(object sender, EventArgs e)
{
    this.Close();
}

private void btnLimpiar_Click(object sender, EventArgs e)
{
    txtPlacas.Text = "";
    txtVisita.Text = "";
    var lusu = from u in db.AccFrac_Visitante
    select new { u.Nombre, u.Placas };
    dgvAcceso.DataSource = lusu.ToList();
    labelInfo.Text = "Total de Registros: " + lusu.Count();
}

private void txtVisita_TextChanged(object sender, EventArgs e)
{
    var res = from u in db.AccFrac_Visitante
    where u.Nombre.StartsWith(txtVisita.Text.Trim())
    select new { u.Nombre, u.Placas };
    dgvAcceso.DataSource = res.ToList();
    labelInfo.Text = "Total de Registros: " + res.Count();
}

private void txtPlacas_TextChanged(object sender, EventArgs e)
{
    var res = from u in db.AccFrac_Visitante
    where u.Nombre.StartsWith(txtPlacas.Text.Trim())
    select new { u.Nombre, u.Placas };
    dgvAcceso.DataSource = res.ToList();
    labelInfo.Text = "Total de Registros: " + res.Count();
}

```

## **PRUEBAS E IMPLEMENTACION**

### *Login*

En esta pantalla se muestra el inicio de sesión de nuestra aplicación, pidiendo Usuario y Contraseña para dar acceso.

A login form with a light gray background. At the top center is a placeholder icon for a user profile. Below it, the word "Usuario" is displayed above a white text input field. Underneath that, the word "Contraseña" is displayed above another white text input field. At the bottom, there are two gray buttons: "ACEPTAR" on the left and "CANCELAR" on the right.

### *Menu*

Esta es una pantalla básica donde se muestra en la parte superior tres opciones la de Administrador, Acceso Vehículos y la opción de Cerrar Sesión.



### Visitas Programadas y No Programadas

En esta opción de nuestro menú se muestra esta pantalla dividida en dos donde se muestran diferentes opciones, en la pantalla se muestran diferentes cámaras que se encuentran en nuestro monitor de entrada, cámara de Identificación, Cámara de Rostro, Cámara de Placas Delanteras y Traseras, opciones para ingresar el Nombre de Visitante, Nombre de Condominio, Dirección y Motivo de la Visita.

En la pantalla de Salida Se muestran las Cámaras de Rostro, Placa Delantera, Placa Trasera, Nombre de Visitante y De Condominio

The screenshot displays a security management interface with an orange header bar. On the left, the 'ENTRADA' (Entry) section is active, showing a 'Visita Programada' (Scheduled Visit) form. This form includes four camera feeds: 'Identificación Oficial' (Official Identification), 'Rostro' (Face), 'Placa Delantera' (Front Plate), and 'Placa Trasera' (Back Plate). Below the feeds are input fields for 'Nombre Visitante' (Visitor Name), 'Dirección' (Address), and 'Nombre Condominio' (Condominium Name). A 'Cerrar' (Close) button is at the bottom left. To the right of the feeds is a 'Visita No Programada' (Unscheduled Visit) form with fields for 'Flujo' (Flow), 'Caseta' (Booth), 'Domicilio' (Residence) with a dropdown menu showing 'CIPRES SUR', 'Numero Int/Ext' (Internal/External Number), 'Nombre Visitante' (Visitor Name), 'Placas' (Plates), and 'Motivo de Visita' (Reason for Visit). An 'Agregar' (Add) button is at the bottom right of this form. On the far right, the 'SALIDA' (Exit) section is visible, showing a 'Visita No Programada' form with fields for 'Rostro' (Face), 'Placa Delantera' (Front Plate), 'Placa Trasera' (Back Plate), 'Nombre Visitante' (Visitor Name), and 'Nombre Condominio' (Condominium Name). A 'Cerrar Sesión' (Log Out) button is in the top right corner of the header bar.

## Salidas Visitas No Programadas

En esta pantalla una vez en funcionamiento la Aplicación muestra las Cámaras de Rostro y Placas, Donde se encuentra el Modulo en la aparte de abajo se muestra una lista conectada a nuestra base de datos donde muestra una lista de los visitantes que se encuentran dentro del condominio en ese momento

Consecutivo	Nombre Visitante	Direccion Visita	Numero	Motivo Visita	Placas
4	Jose Manuel	(direccion = Plaz...	369	Entrega de utenc...	sdferge55825
14	Raul Marin	CIPRES PONEN...	123	Entrega de zapat...	QWED289-D

## Usuarios

En esta pantalla se muestra una tabla de todos los Condominios Registrados en nuestra Aplicación, se muestran los Usuarios, Nombre, Apellido, Rol, y Estado, así mismo muestra las opciones de Activar, Limpiar tabla, Agregar Usuario y Eliminar

ID Usuario	Nombre	Apellido	Rol	Estado
1	Alberto	Miranda	Administrador	Activo
6	Pablo	Mamoli	Administrador	Activo
11	Jonas	Hernandez Mend...	Administrador	Activo
14	Tonio	Perez	Vigilante	Activo
29	zazaza	adad	Vigilante	Activo
33	werwer	werwer	Vigilante	Activo
35	veveve	veveve	Vigilante	Activo
37	zazazazazazazazaz	zazazazazazazazaz	Vigilante	Activo
46	wwwwww	adadad	Vigilante	Activo
52	sdad	adad	Vigilante	Activo
55	Mato	Santos	Vigilante	Activo
56	sadad	adad	Vigilante	Activo
57	sasas	sasas	Vigilante	Activo
58	dad	adadad	Vigilante	Activo

## **Capítulo IV. Resultados y conclusiones**

Se esperaba un funcionamiento de la app brindando así una mejor accesibilidad a sus componentes y funcionalidades, integrando así diferentes herramientas y acceso a las diversas funciones de la aplicación junto con la posibilidad de acceder de forma mas fácil y rápida a cualquier condominio.

### *Objetivos*

El proyecto se basa en la seguridad de los fraccionamientos al momento de permitir el acceso a los visitantes, por medio de un sistema y aplicación los usuarios darán acceso de forma remota desde su celular sin la necesidad de que sea tardado el proceso de entrada ya que con la aplicación validará todos los datos del visitante y a la entrada las cámaras detectaran su rostro, auto y matrícula para mayor seguridad

### *Lo que se logro*

Uno de los principales objetivos de mayor prioridad fue dar seguridad, implementación e automaticacion de nuestra aplicación, eso fue en lo que mas empeño le puse, trate de hacer la aplicación lo mas segura para los usuarios y también para los administradores ya que sea fácil de usar pero con una seguridad impenetrable, también trabaje demasiado en el diseño para que sea fácil de usar y no cause problemas al usuario, y la interfaz sea fácil de entender, al igual se logro que la aplicación hiciera sus funciones, al llamar imagen de las cámaras y guardar datos al momento de dar de alta algún usuario, mi tutor queda satisfecho con la aplicación ya que hace totalmente lo que se requería.

Como segunda tarea teníamos que incorporar un chat propio de la aplicación para así evitar usar una aplicación de mensajería externa a la propia plataforma, el objetivo fue cumplido, aunque no se pueden enviar archivos logramos crear un chat bidireccional instantáneo entre dos usuarios propios de la misma aplicación, así como saber quien te manda mensajes mediante notificaciones.

### *Lo que no se pudo cumplir*

Una de las tareas secundarias fue agregar métodos de pagos externos, como el uso de tarjetas bancarias para así hacer transferencias desde la misma aplicación, fue una tarea que no pudo ser cumplida por falta de conocimientos de lo necesario para poder programar ese tipo de métodos de pago, pero eso no interfiere con el funcionamiento de la aplicación

### *Trabajo en equipo*

En este proyecto trabaje de manera individual ya que solo estaba apoyado por los diferentes compañeros trabajadores de la misma empresa donde me encontraba realizando mis estadías, ellos fueron mi apoyo, los que me ayudaban cuando se me dificultaba algo, cuando necesitaba alguna idea ellos estaban ahí para apoyarme, me gusto mucho esta etapa en mi educación universitaria ya que me ayudo a comprender mi carrera en el mundo real y saber como funciona, el proyecto lo realice individualmente recibiendo apoyo de mis compañeros.

### *Desarrollo personal*

Personalmente me siento con un gran desempeño en este tiempo de trabajo, mi estadía me ayudo a creer personalmente en las cosas que puedo hacer y a creer en que el trabajo en equipo es de gran ayuda, pues ahorramos mucho tiempo de trabajo así como también compartimos diferentes métodos para hacer las cosas, además de conocer un poco el mundo laboral de mi carrera en la cual me estoy desempeñando, son de las pocas experiencias obtenidas este tiempo pero creo que son de las mas valiosas pues pude poner en practica esos conocimientos adquiridos a lo largo de mis estudios en la universidad.



### *Conclusiones.*

Personalmente elaborar y desarrollar este trabajo de investigación y desarrollo para la documentación de mi trabajo de estadía que estuve desarrollando durante estos 4 meses que tengo realizando una aplicación de escritorio que permita el fácil control y acceso de condominios, elaborar esta aplicación me impuso nuevos retos y desafíos personalmente ya que necesité de mucha investigación para poder elaborarla ya que fue hecha en una aplicación donde no tenía conocimiento de cómo funcionaba esa aplicación de desarrollador, me enseñó a cómo funcionaban las empresas en el mundo real, fuera del salón de clases, mientras estas desarrollando una aplicación muy importante y de alto nivel.

## Anexos

### *Cronograma de actividades.*

 <b>UNIVERSIDAD TECNOLÓGICA DE AGUASCALIENTES</b> <b>Cronograma de Actividades</b>		
Programa Educativo: Estadías Profesionales		Periodo: Mayo - Agosto 2021
Nombre del Estudiante: OSCAR RENATO GARCIA RESENDIZ		
Nombre de la Empresa o Institución: COYTECH		
Título de la Estadía Profesional: Smart Access		Fecha de elaboración: 12/05/2021
Objetivo de la Estadía Profesional: Rediseño de un conjunto de aplicaciones que permitan la administración de fraccionamientos y cotos para optimizar las reservaciones, visitas, pagos y avisos		

No.	P/R	Actividad	Semanas															
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	P	Capacitación en Windows Forms y SQL server mangnament																
	R																	
2	P	Trabajos de investigacion y ejemplos de Windows Forms																
	R																	
3	P	Documentacion y Diseño de app																
	R																	
4	P	Investigacion de la base de datos usada																
	R																	
5	P	Programación en Visual Studio																
	R																	
6	P	Desarrollo de app para altas y baja de Tags																
	R																	
7	P	Realización de prototipo																
	R																	
8	P	Conexión con la base de datos																
	R																	
9	P	Desarrollo frontend																
	R																	
11	P	Conexión Backend-Frontend																
	R																	
12	P	Correcciones y Cambios																
	R																	
13	P	Pruebas Finales																
	R																	
15	P	Aplicación web funcional																
	R																	
16	P	Programación de la funcionalidad de la App																
	R																	
17	P	Documentación del proyecto																
	R																	

Firma del estudiante

P= Programado  
R= Real

Firma del asesor

Firma del tutor

F-SA-06  
Revisión No. 1  
Fecha de Revisión: 22/03/17

## Formatos varios



Aguascalientes, Ags, a 13 del mes de agosto, 2021.

A quien corresponda:

Por medio de la presente, se hace constar que el estudiante Oscar Renato Garcia Resendiz, de la carrera de TSU. en Tecnologías de la Información Y comunicación en Desarrollo de Software realizó el proyecto de estadía denominado Smart Access, teniendo como tutor por parte de la empresa a Luis Ariel Victoria Chávez, y como asesor por parte de la Universidad Tecnológica de Aguascalientes Gonzalo Campos Argueta, cumpliendo con un total de 525 horas de trabajo, y con la elaboración y entrega a la empresa del reporte del trabajo de estadía digital, con lo cual se libera de cualquier responsabilidad al estudiante a partir de la fecha señalada, en el entendido de que cumplió satisfactoriamente con los requisitos y objetivos del mismo.

Se extiende la presente como evidencia de la conclusión del proyecto, y para los fines que al interesado convengan.

ATENTAMENTE:

Ing. Luis Ariel Victoria Chávez  
Director del área de sistemas

Escaneado con CamScanner

## *Referencias Bibliográficas*

Técnicas de programación. México: McGraw Hill. Joyanes, L. (2008).

Fundamentos de programación. España: McGraw Hill. Joyanes, L. y Zahonero, I. (2011).

Rajsbaum, S., Bribiesca, E., Galavis, J., Solsona, F. (2009). Conocimientos fundamentales de computación. México: UNAM.

Joyanes, L. y Zahonero, I. (2010). Programación en C, C++, Java y UML. México: McGraw Hill.