**Universidad Tecnológica de Aguascalientes**



SMARTACCESS

**Reporte de Estadía Profesional**

que para obtener el título de

TECNICO SUPERIOR UNIVERSITARIO en

Presenta:

**Nombre del egresado (a)**

**Oscar Renato García Reséndiz**

Aguascalientes, Ags. Mayo, 2021

Proyecto realizado en (COYTECH)

Asesor (a)

Nombre del(a) Asesor(a)

Gonzalo Campos Argueta

Tutor (a)

Nombre del(a) Tutor(a)

Luis Ariel Victoria Chávez

Agradecimientos ó Dedicatorias

Optativo

Contenido

[Introducción 6](#_Toc77897560)

[Capítulo l Generalidades de la empresa 7](#_Toc77897561)

[Datos generales 7](#_Toc77897562)

[Antecedentes Históricos 7](#_Toc77897563)

[Misión 7](#_Toc77897564)

[Visión 7](#_Toc77897565)

[Valores 7](#_Toc77897566)

[Productos o servicios que ofrece 8](#_Toc77897567)

[Organigrama 8](#_Toc77897568)

[Descripción del departamento donde se realizó la estadía 8](#_Toc77897569)

[Capítulo II: Planteamiento de la Estadía Profesional 9](#_Toc77897570)

[Descripción de la Problemática: 9](#_Toc77897571)

[Objetivos: 9](#_Toc77897572)

[Justificación: 9](#_Toc77897573)

[Viabilidad: 10](#_Toc77897574)

[1. Alcance del proyecto: 10](#_Toc77897575)

[2. Análisis de situación: 10](#_Toc77897576)

[3. Definición de requisitos 10](#_Toc77897577)

[4. Determinación del enfoque 11](#_Toc77897578)

[Resultados esperados: Mejorar la seguridad y el acceso a los fraccionamientos 11](#_Toc77897579)

[Metodología a aplicar: 11](#_Toc77897580)

[¿Qué es el modelo en cascada? 11](#_Toc77897581)

[¿Cómo funciona el modelo en cascada? 11](#_Toc77897582)

[Capítulo III. Desarrollo de la estadía profesional. 13](#_Toc77897583)

[Análisis de Requerimientos 13](#_Toc77897584)

[Por Fraccionamiento. 13](#_Toc77897585)

[Para todos los fraccionamientos. 13](#_Toc77897586)

[(REQUERIMIENTOS FUNCIONALES): 14](#_Toc77897587)

[Requisitos de sistema e instalación 14](#_Toc77897588)

[EQUIPO PARA RED (e internet) 14](#_Toc77897589)

[EQUIPAMIENTO PARA SISTEMA DE CONTROL DE ACCESO 14](#_Toc77897590)

[(REQUERIMIENTOS NO FUNCIONALES): 15](#_Toc77897591)

[DISEÑO 16](#_Toc77897592)

[Documento Oficial Smart Access 16](#_Toc77897593)

[SEGURIDAD Y CONFIDENCIALIDAD 17](#_Toc77897594)

[ADMINISTRACION 17](#_Toc77897595)

[PLAN DE PAGOS 18](#_Toc77897596)

[DESARROLLO O PROGRAMACION 19](#_Toc77897597)

[Login 19](#_Toc77897598)

[Menú 24](#_Toc77897599)

[Administrador Fraccionamientos 31](#_Toc77897600)

[Cámaras 34](#_Toc77897601)

[Visitas Programadas y No Programadas 37](#_Toc77897602)

[Salidas Visitas No Programadas 42](#_Toc77897603)

[Usuarios 48](#_Toc77897604)

[Usuarios Editar 55](#_Toc77897605)

[Usuarios Agregar 59](#_Toc77897606)

[Acceso a Vehículos 63](#_Toc77897607)

[Capítulo IV. Resultados y conclusiones 66](#_Toc77897608)

[Descripción de los resultados obtenidos conforme a los objetivos planteados en el proyecto: 66](#_Toc77897609)

[Conclusiones. 66](#_Toc77897610)

[Recomendaciones. 66](#_Toc77897611)

[Anexos 67](#_Toc77897612)

[Cronograma de actividades. 67](#_Toc77897613)

[Formatos varios 67](#_Toc77897614)

[Referencias Bibliográficas 68](#_Toc77897615)

# 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? (Describa de manera breve los apartados que contiene el reporte). Esta organizada en 3 pequeñas aplicaciones, usuario, admín, vigilante y Administrador de la aplicación la cuales tendrán diferentes funcionalidades

# Capítulo l 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.

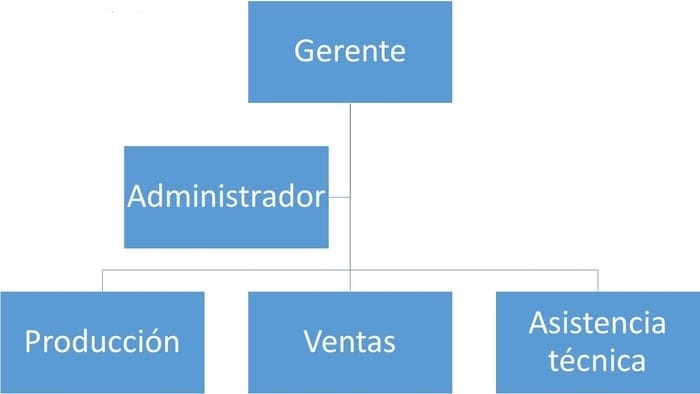
## Valores

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:

1. 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
2. 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
3. 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).

1. 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. Muy al contrario, en su ensayo de 1970 titulado Managing the Development of Large Software Systems, el teórico presenta una reflexión crítica acerca de los procedimientos lineales. A modo de alternativa, Royce presenta un modelo iterativo incremental en el que cada una de las fases se basa en la anterior y verifica los resultados de esta.

Royce propone un modelo compuesto por siete fases que se ha de ejecutar en diversas vueltas (iteraciones):

Requisitos de sistema

Requisitos de software

Análisis

Diseño

Implementación

Prueba

Servicio

# Capítulo III. Desarrollo de la estadía profesional.

Documentación específica de actividades e investigación realizada, pruebas de campo, etc.

Incluir la documentación respectiva.

# Análisis de Requerimientos

## 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 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):

## Requisitos de sistema e instalación

## EQUIPO PARA RED (e internet)

## 

Rack, 3 charolas

Organizadores verticales para rack

Organizadores horizontales para rack

Barra eléctrica de 12 contactos

Regulador de 4 contactos 1kva

Ups de 40 minutos

Gabinete de Pared 6U, hasta 60KGs, Negro

Multicontacto de 8 Ptos

Swtich poe 1G de 24 Ptos

Switch poe 1G de 16 Ptos (sub-site)

## EQUIPAMIENTO PARA SISTEMA DE CONTROL DE ACCESO

Swith Dahua poe 1Gb de 8 Ptos.

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

PC para control de acceso, visitas programadas, no programadas, etc.

Cámara Dahua IP 3Mp

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

ZKTeco Lector de Códigos QR y Tarjeta de Proximidad QR500, RS-485

Lector de huellas digitales ZKTeco para pc USB

## (REQUERIMIENTOS NO FUNCIONALES):

Gabinete para ext. En acero inoxidable para control de acceso

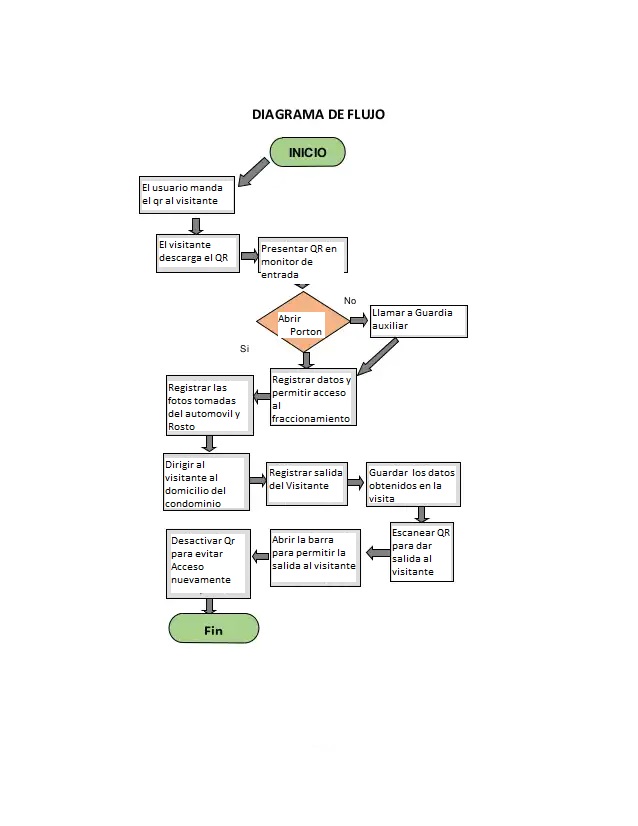
Lector de huellas digitales ZKTeco para pc USB

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



## 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 accesando.

•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 accesando 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 practico 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 hibrido 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 SENCIBLE (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 halla servicio de internet, el sistema NO SE DETIENE, y las notificaciones pendientes, se reanudaran en cuanto vuelva a reanudarse la conexión a internet, esto ayuda a no dejar el fraccionamiento sin control de acceso. (Ser 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 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

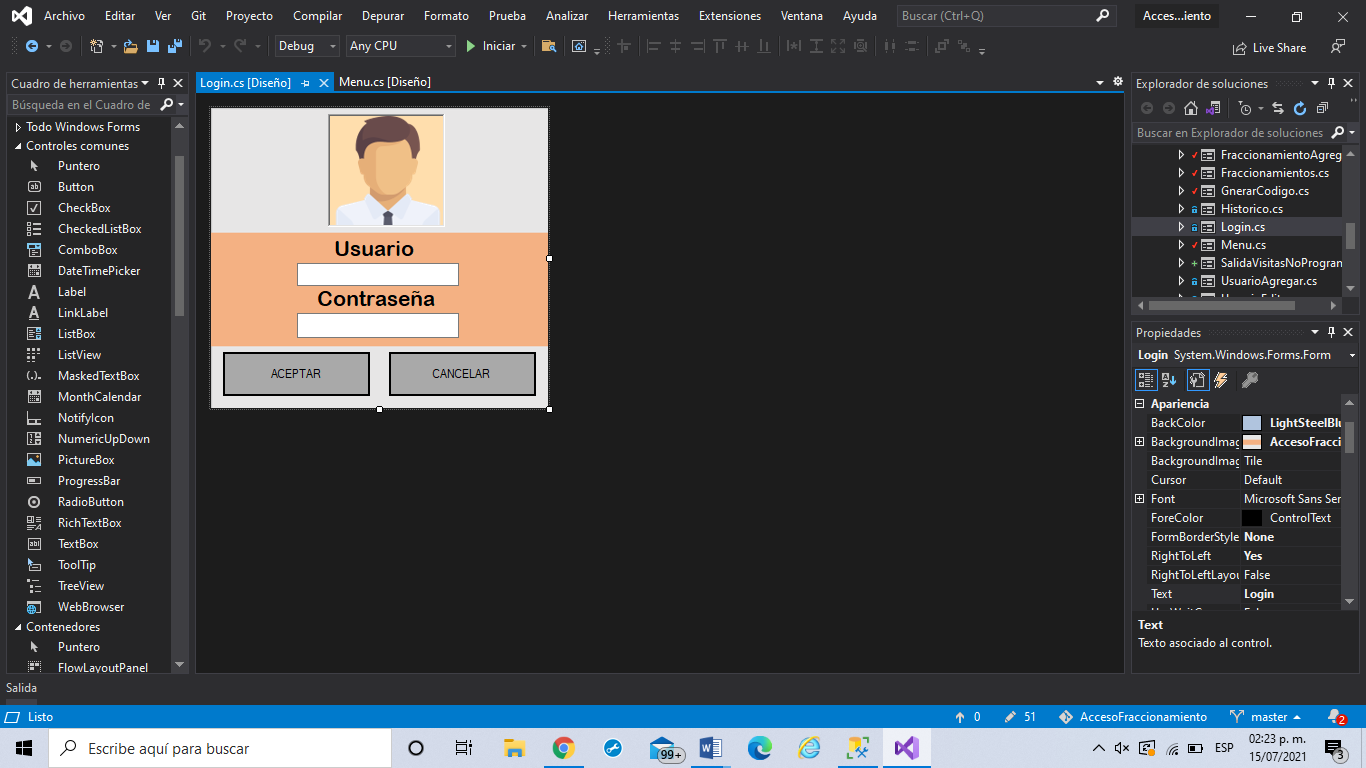
El fraccionamiento Se encargará de reunir el pago de los colonos registrados y tendrá un periodo de 30 días de crédito para realizar el pago en el esquema que elija, tiempo suficiente para la recolección del pago de los licenciamientos.

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();

}

private void Login\_Load(object sender, EventArgs e)

{

OpenLogin();

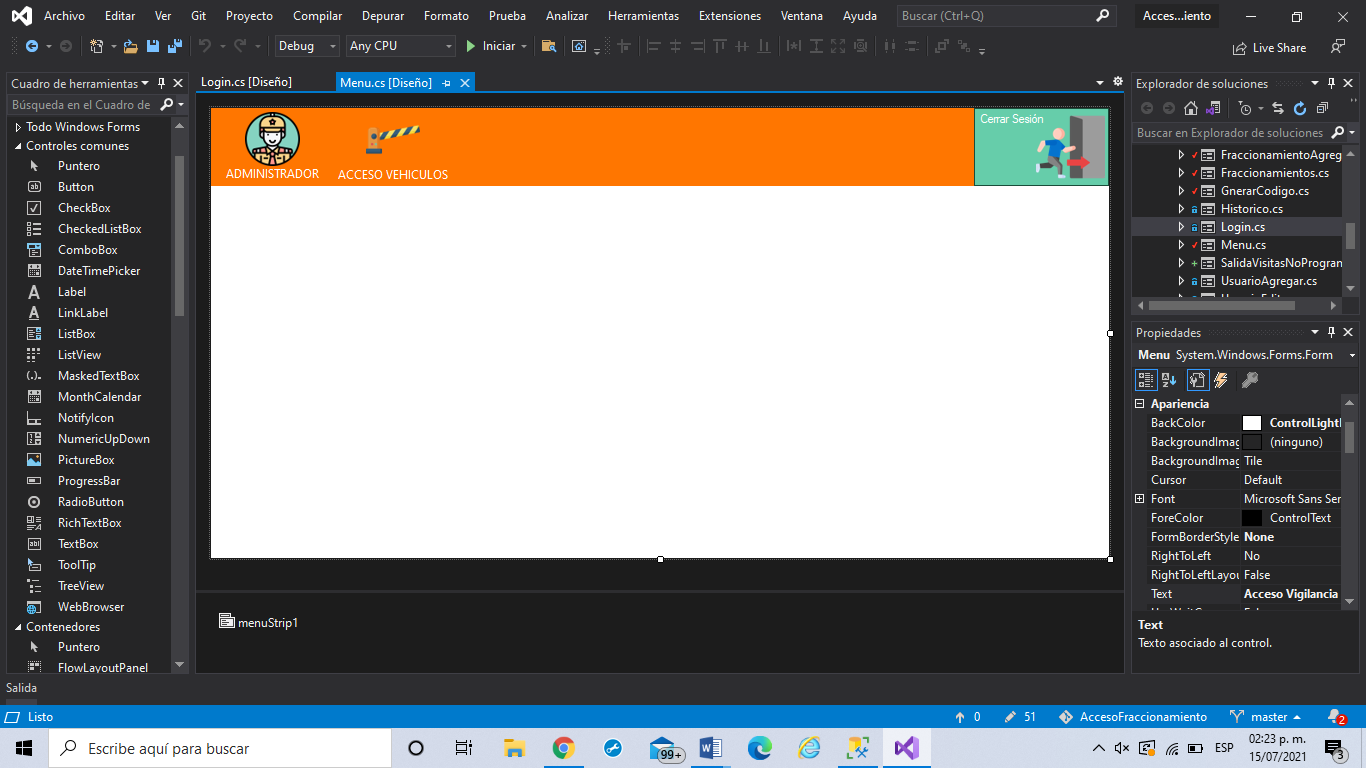
}

}

}

## 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 fechay hora de inicio de sesion

db.SaveChanges();

}

}

catch (Exception ex)

{

MessageBox.Show("" + ex);

}

if (labelaccess.Text.ToLower() == "1")//INGRESA CORRECTAMENTE Y SE DEFINEN LOSROLES

{

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)

{

SalidaVisitasNoProgramadas snp = new SalidaVisitasNoProgramadas();

snp.Show();

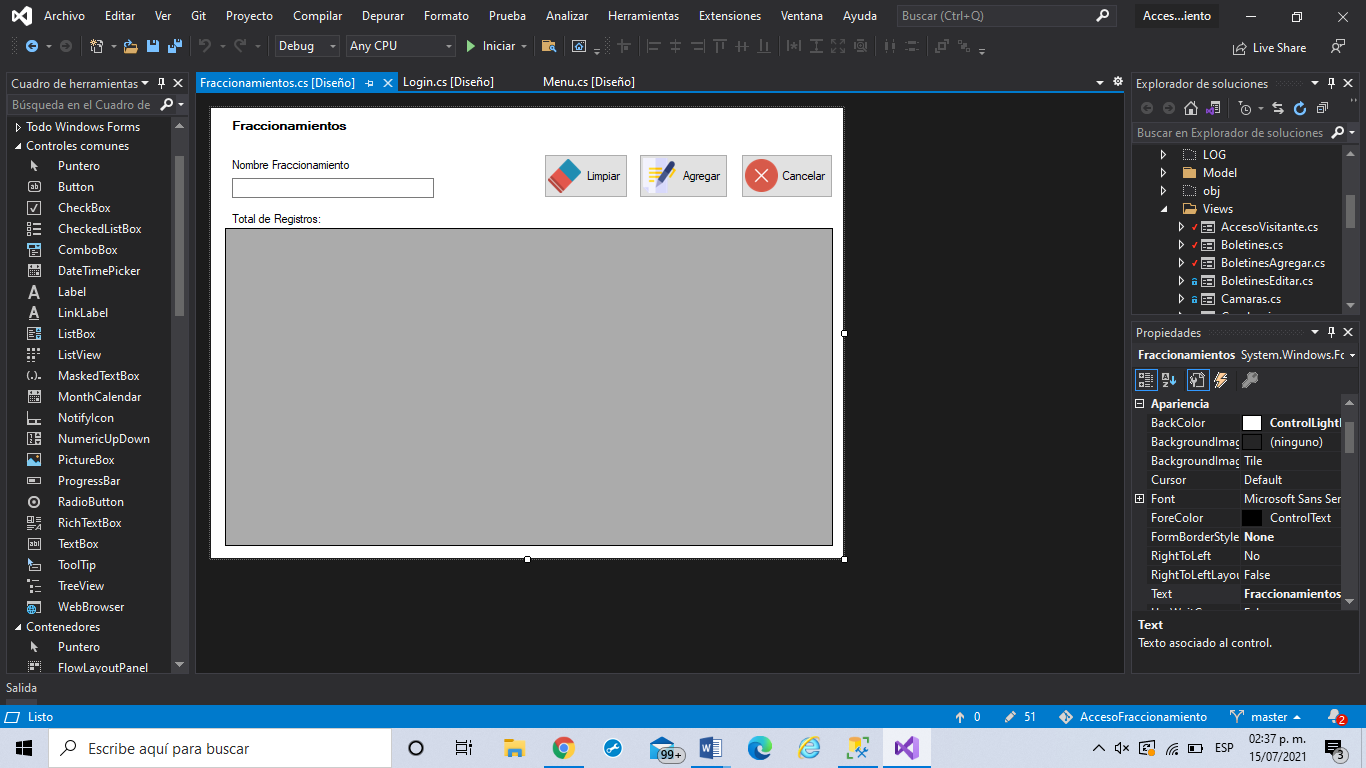
}

}

}

## 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 hace 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.EntityFrameworks;

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 Contrasena2 { 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();

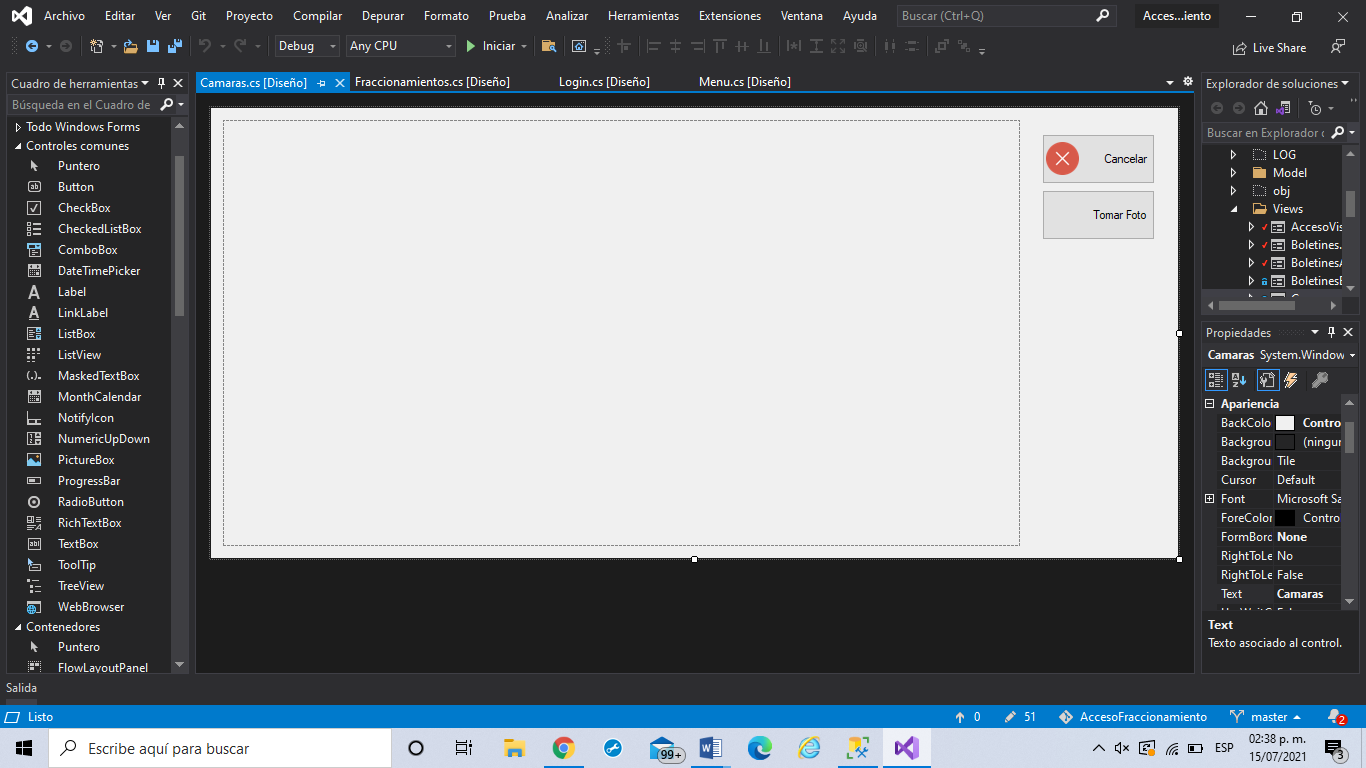
}

}

}

## 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.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 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";

var snapShotImage = \_snapshot.TakeSnapshot().ToImage() as System.Drawing.Image;

snapShotImage.Save(ruta);

}

private void btnCancelar\_Click(object sender, EventArgs e)

{

this.Close();

}

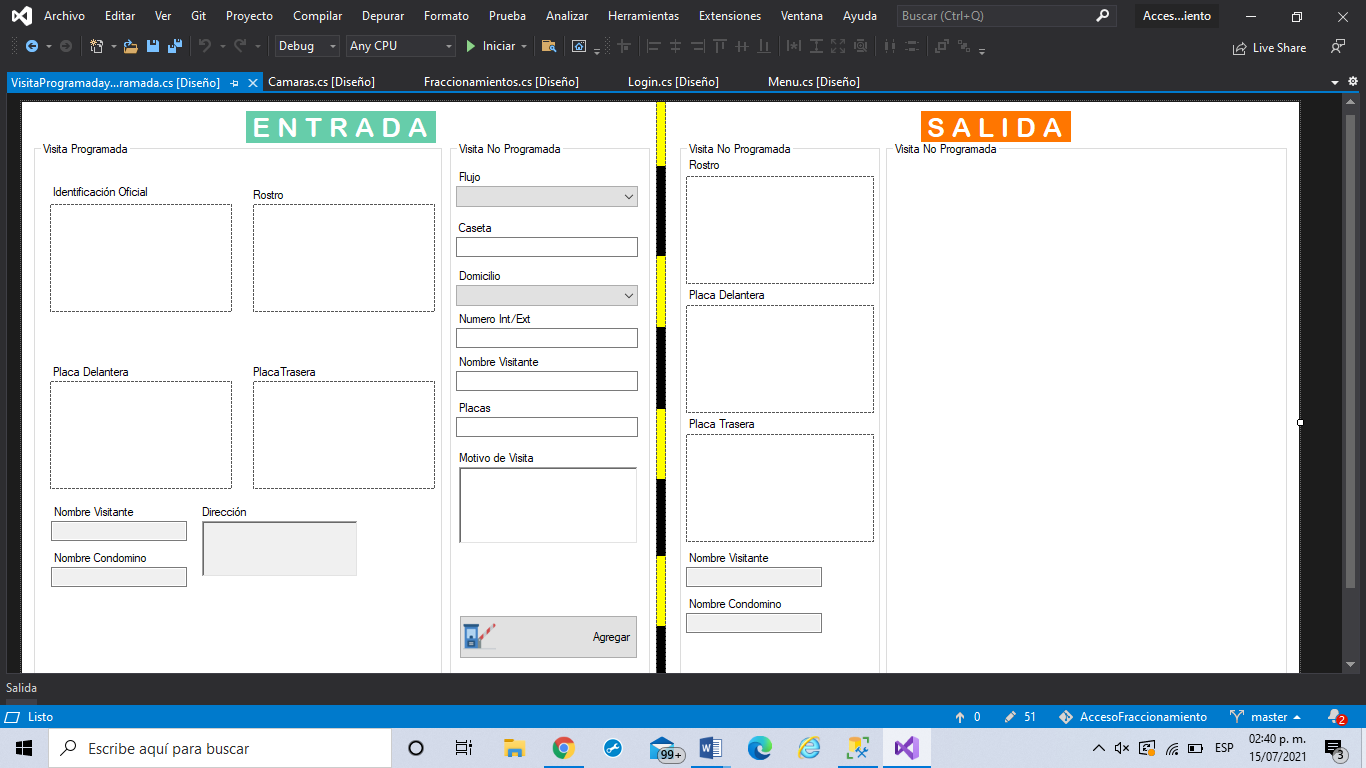
}

}

## 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 dirije a " + vnp.DireccionVisita.Trim() + " " + vnp.Numero.Trim() + " Agregado.", "Información", MessageBoxButtons.OK, MessageBoxIcon.Information);

bi.DescripcionBitacora = "Se agregó visitsa 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)

{

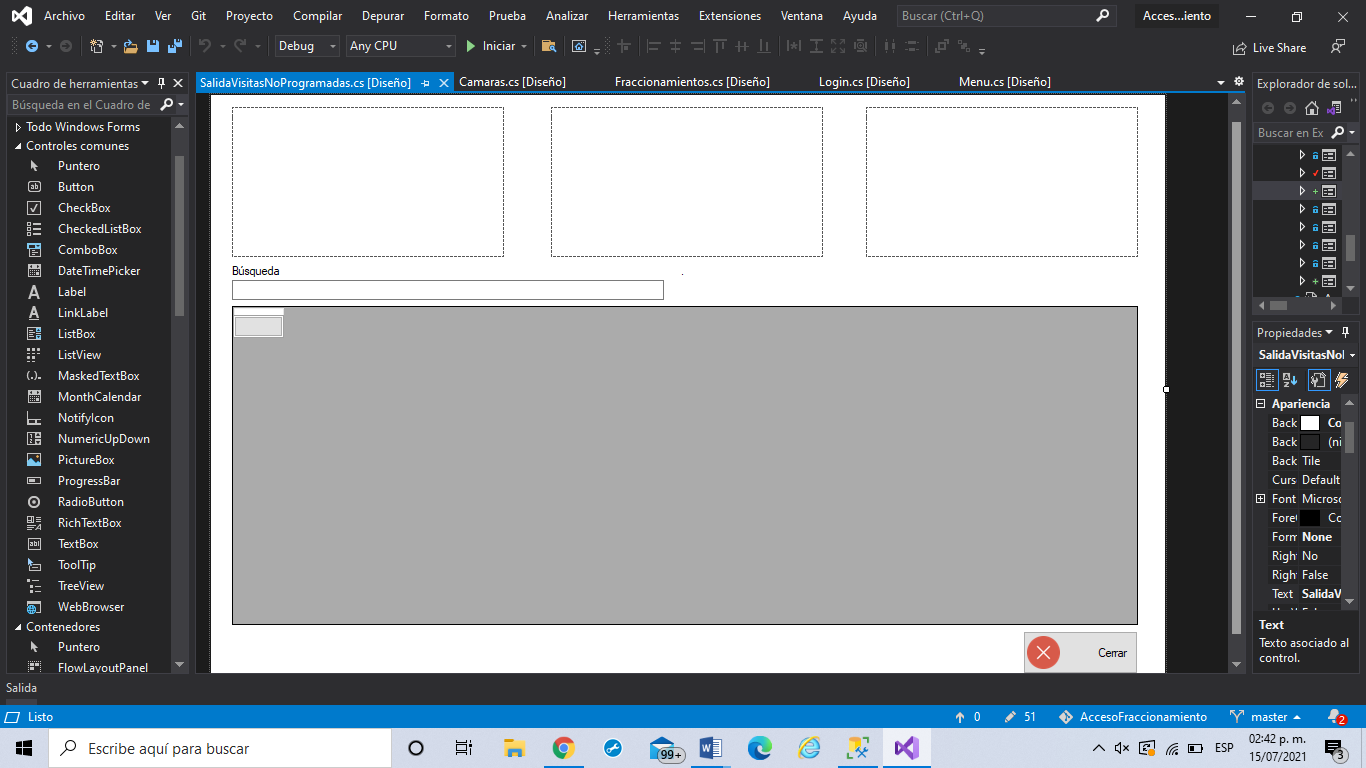
}

}

}

## 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, new 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)

e.CellStyle.BackColor = Color.LightGreen;

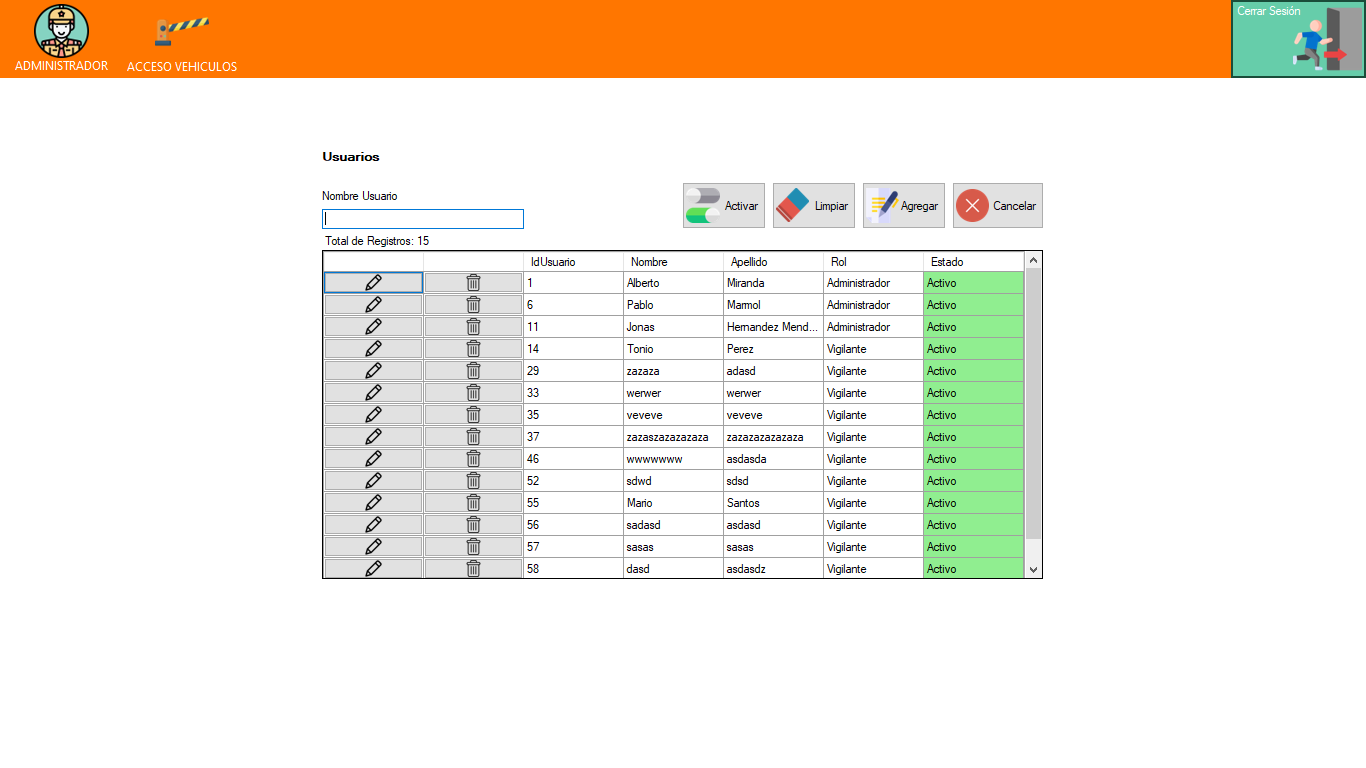
}

}

}

## 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



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 Contrasena2 { 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:

iID = (dgvUsuarios.Rows[e.RowIndex].Cells["IdUsuario"].FormattedValue).ToString();

ID = int.Parse(iID);

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")

{

if (e.Value != null)

{

if (e.Value.GetType() != typeof(System.DBNull))

{

if (e.Value.ToString() == "True")

{

e.Value = "Activo";

e.CellStyle.BackColor = Color.LightGreen;

}

else

{

e.Value = "Inactivo";

e.CellStyle.BackColor = Color.Salmon;

}

}

}

}

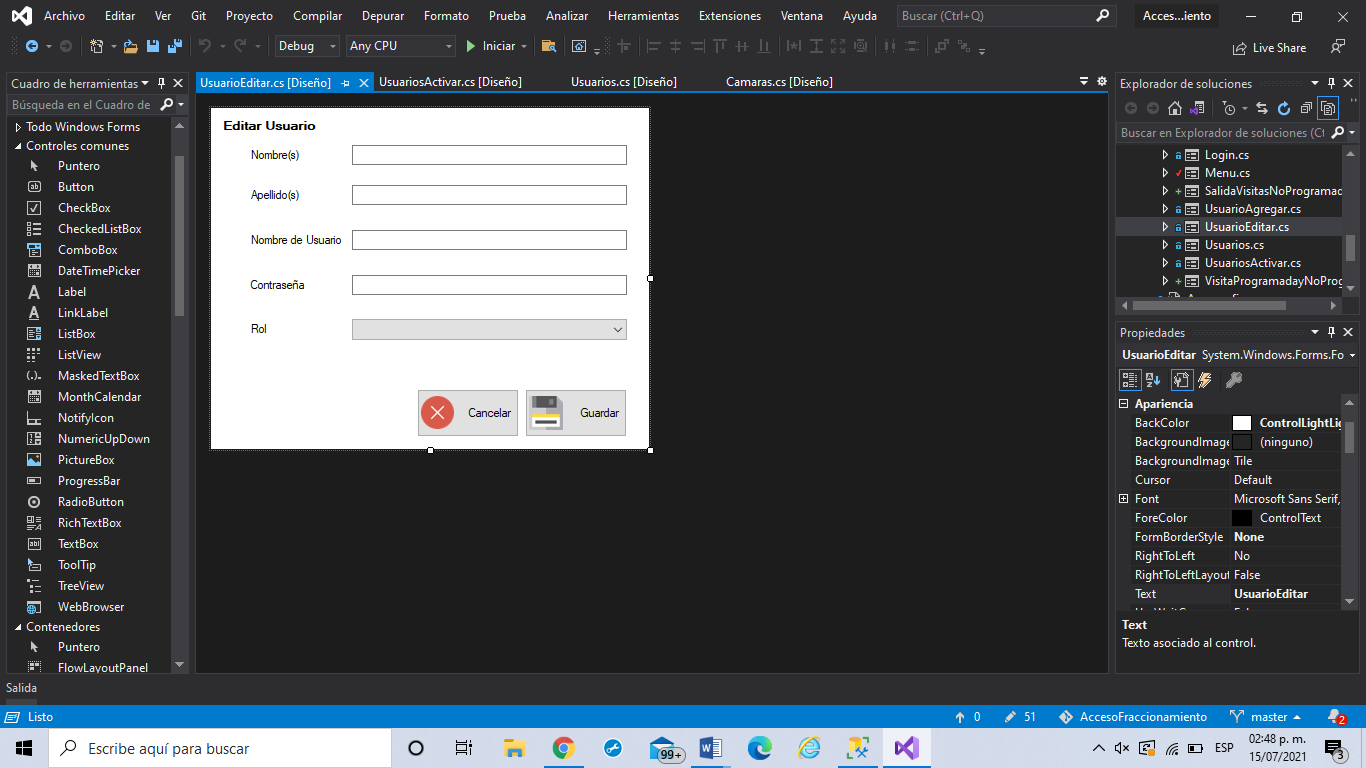
}

}

}

## 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 UsuarioEditar : 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 Contrasena2 { 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();

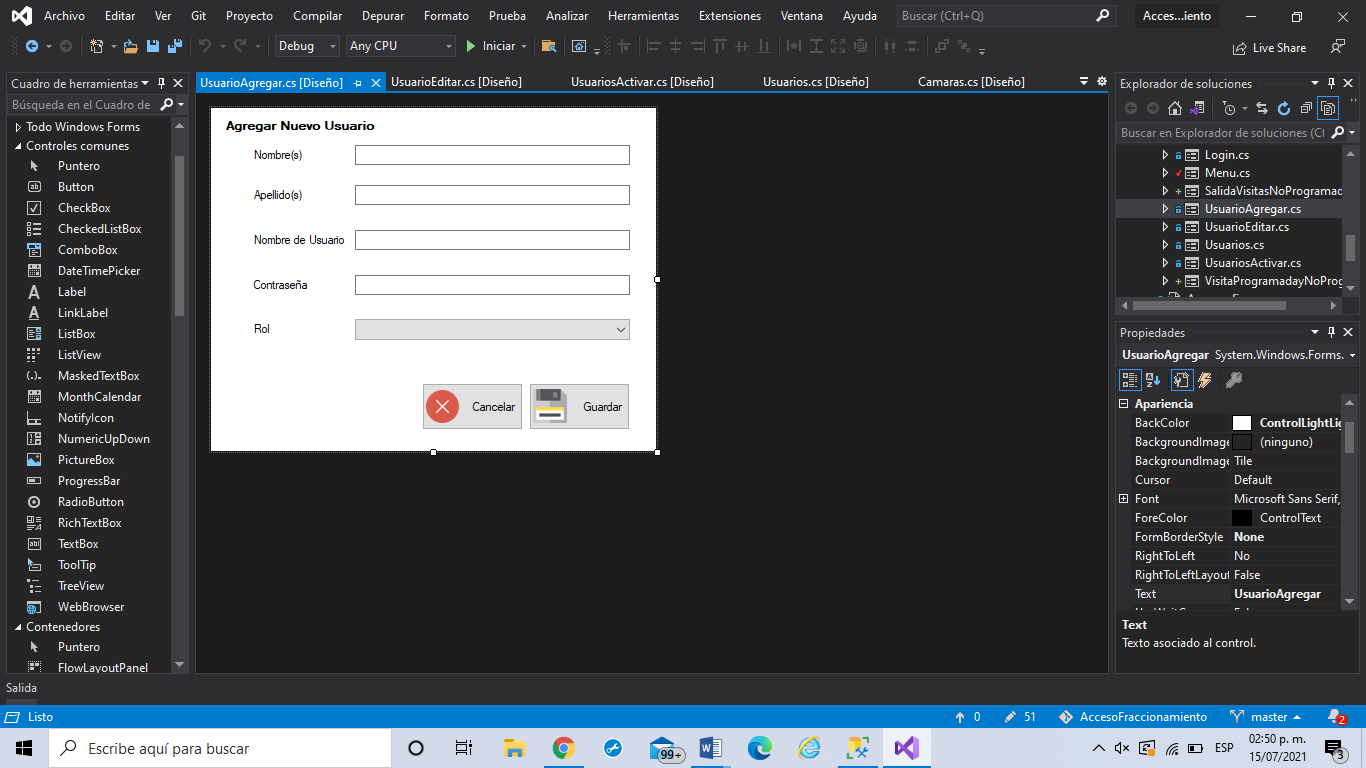
cboxRol.SelectedValue = idrol;

}

}

## 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)

{

var lrol = from r in db.AccFracc\_Roles

select r;

cboxRol.DataSource = lrol.ToList();

cboxRol.DisplayMember = "Rol";

cboxRol.ValueMember = "IdRol";

}

private void btnCancelar\_Click(object sender, EventArgs e)

{

this.Close();

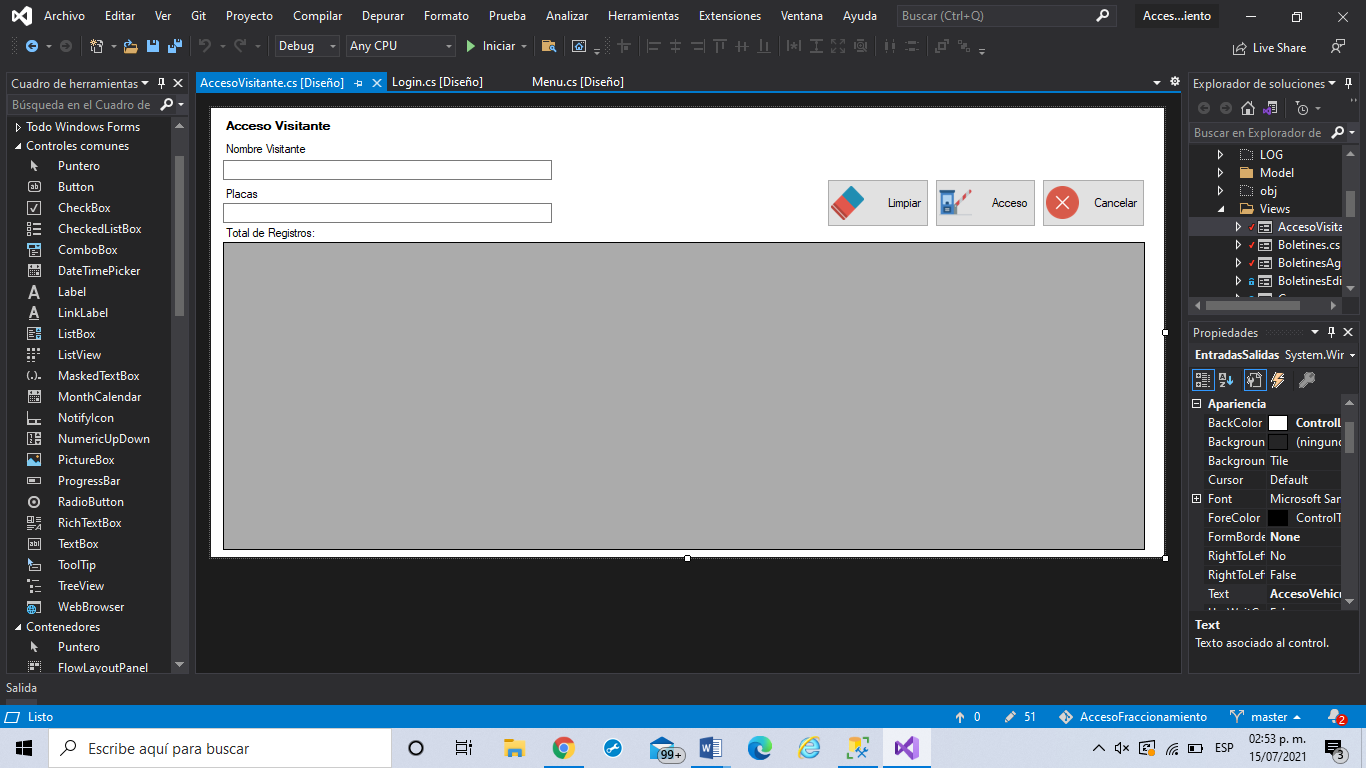
}

}

}

## 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.AccFracc\_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.AccFracc\_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.AccFracc\_Visitante

where u.Nombre.StartsWith(txtPlacas.Text.Trim())

select new { u.Nombre, u.Placas };

dgvAcceso.DataSource = res.ToList();

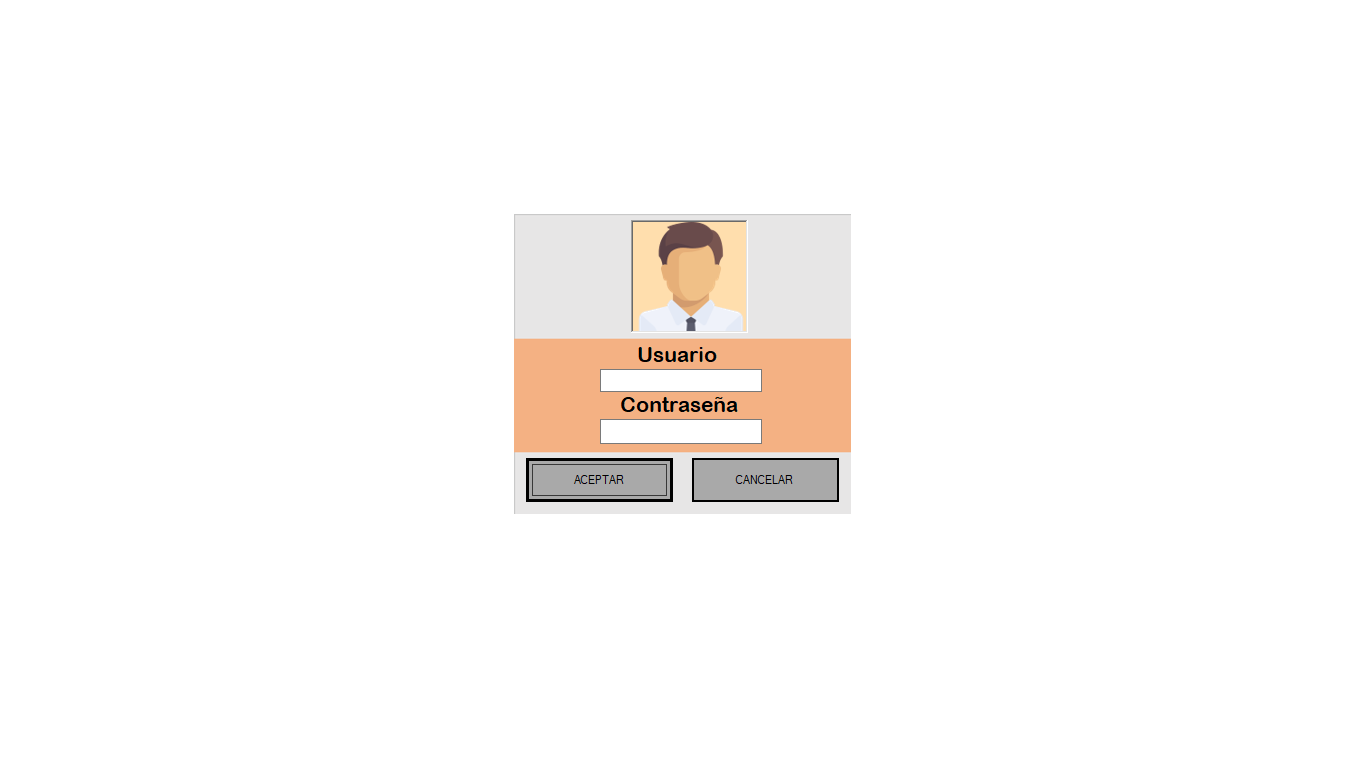
labelInfo.Text = "Total de Registros: " + res.Count();

}

# PRUEBAS E IMPLEMENTACION

## Login

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



## Menu

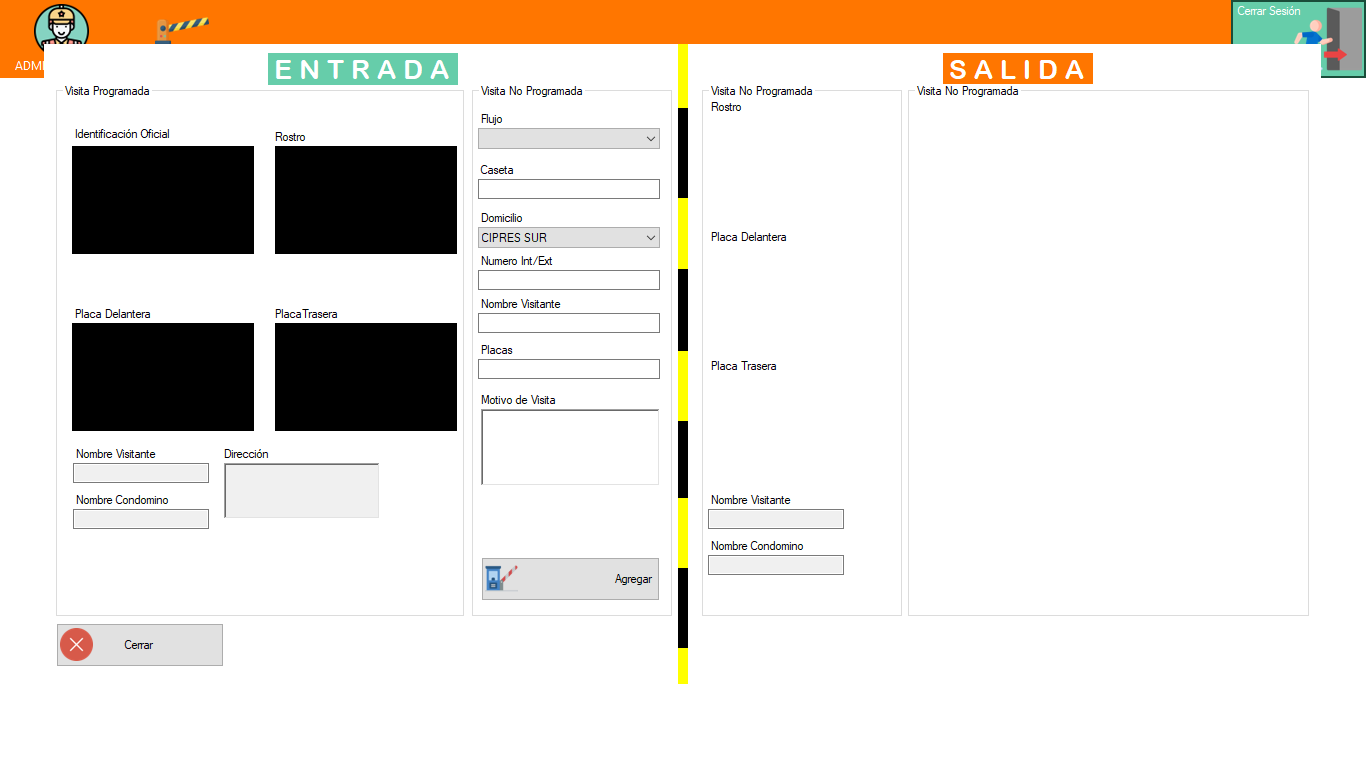
Esta es una pantalla básica donde se muestra en la parte superior tres opciones la de Administrador, Acceso Vehículos y la opción de Cerrar Sesión.



## Visitas Programadas y No Programadas

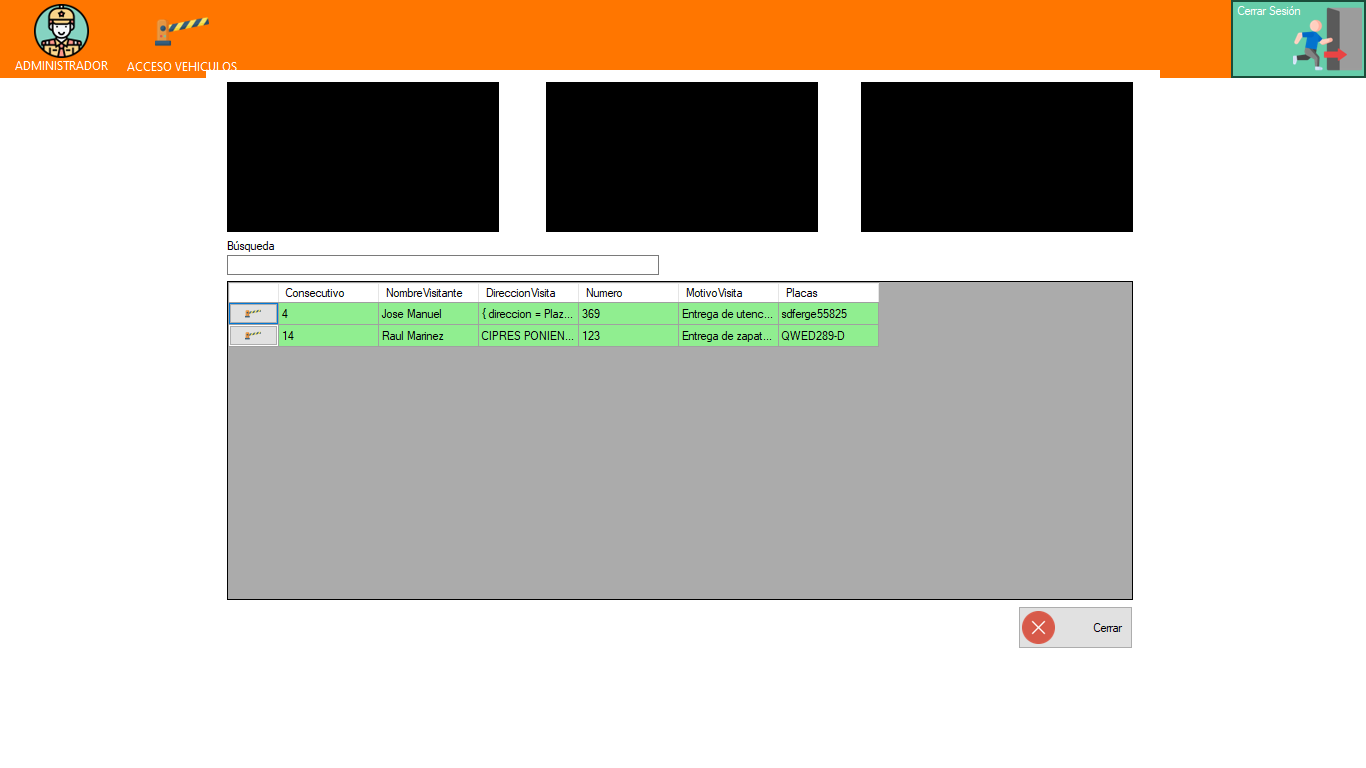
En esta opción de nuestro menú se muestra esta pantalla dividida en dos donde se muestran diferentes opciones, en la pantalla se muestran diferentes cámaras que se encuentran en nuestro monitor de entrada, cámara de Identificación, Cámara de Rostro, Cámara de Placas Delanteras y Traseras, opciones para 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



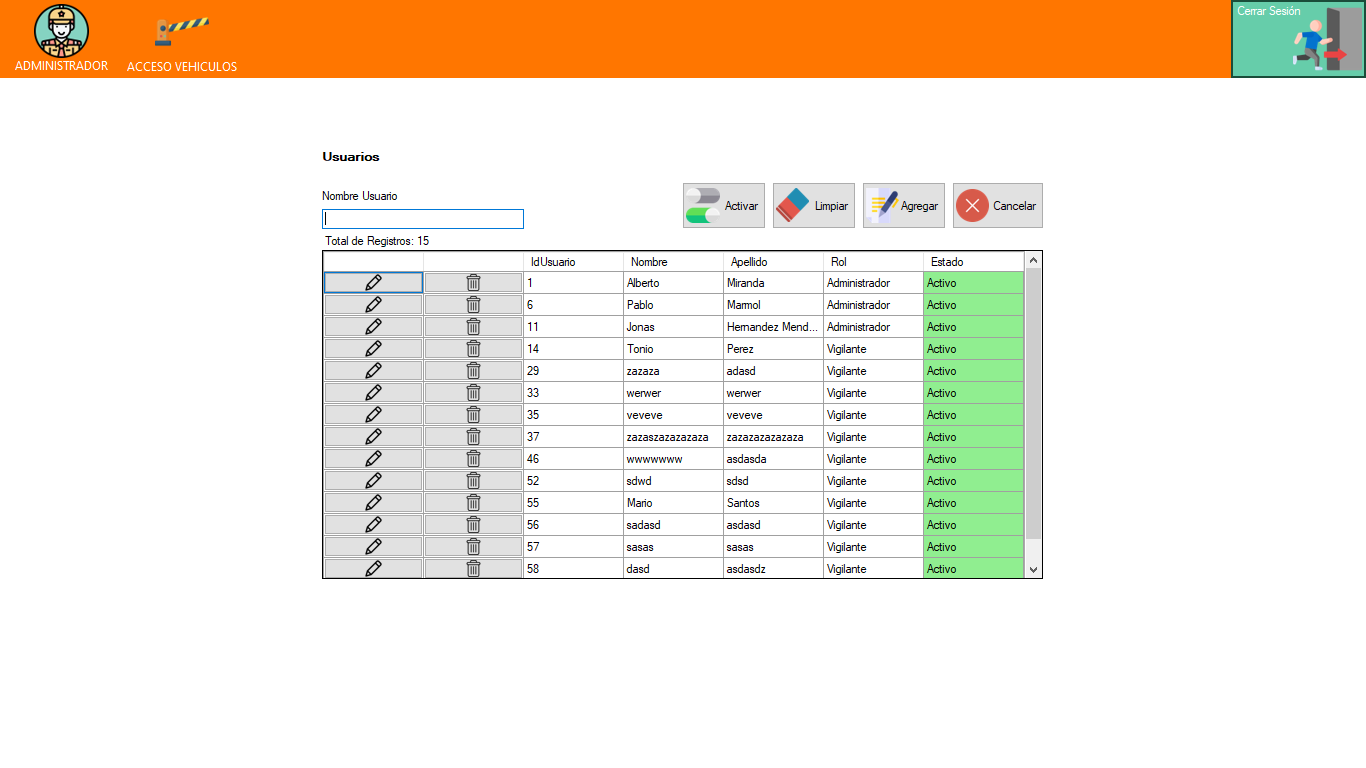
## Salidas Visitas No Programadas

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



## 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



# Capítulo IV. Resultados y conclusiones

Descripción de los resultados obtenidos conforme a los objetivos planteados en el proyecto: Validar los resultados contra el objetivo del empresario.

## 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 enseño 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, esto me funciono para demostrarme lo que puedo hacer, me gusta mucho como se desarrolló esta etapa de mi carrera, llena de nuevos retos y grandes aprendizajes.

## Recomendaciones.

## Anexos

## Cronograma de actividades.

## Formatos varios

## Referencias Bibliográficas

En estilo APA

Forma general

Autor, A.A. (1999). Título del trabajo: Subtítulo del trabajo. (Se menciona la Edición, a partir de la segunda registrando la abreviatura "ed."). Lugar de edición: Editorial.

El listado se debe ordenar alfabéticamente respecto al autor o al título en ausencia de autor; si las obras consultadas fueran de un mismo autor, el orden será cronológico.

Ejemplo 1:

Ronco, E. (2000). Aprender a gestionar el cambio. Barcelona: Paidós.

Ejemplo 2:

Nagel, P. C. (1992). The lees of Virginia: Seven generations of an American family. New York: Oxford University Press.

Ejemplo 3:

Real Academia Española. (1992). Diccionario de la lengua española. (21a ed.). Madrid: Espasa Calpe.

**Aspectos de Edición:**

El documento se escribirá en tamaño carta (fondo blanco), en Word, con márgenes de 2.5 cm., excepto el lado izquierdo que será de 3.5 cm., se prefiere interlineado de 1.5 (para separar títulos o citas textuales puede utilizarse doble interlineado), el número de líneas por página no debe exceder a 30 (incluidas notas al pie de página o referencia). No se contabilizan los encabezados ni pies de página. Al comienzo de cada párrafo es conveniente colocar sangría con 5 o 7 espacios (puede utilizarse el tabulador). Los títulos deben empezar una página nueva si quedaran como línea única al final de la hoja; pueden ser centrados con letras mayúsculas y/o subrayadas, en negritas.

**Formato B**

Se debe usar un solo tipo de letra para todo el documento, a elegir:

Arial,

Courier,

CG Times,

Times New Roman,

Microsoft Sans Serif,

Se pueden utilizar letras negritas, cursivas, mayúsculas y versalitas. Las letras en negrita se utilizan para resaltar una palabra en el texto, las mayúsculas pueden emplearse en los tipos principales del documento, así como en alguna palabra o palabras que resalten de las demás. La letra cursiva o subrayada será utilizada en los siguientes casos:

a) Títulos de obras de referencia científica

b) Títulos de periódicos, revistas, anuarios, almanaques y otras publicaciones similares

c) Expresiones latinas y palabras exóticas

d) Sobrenombres, apodos y palabras escritas en idiomas extranjeros

e) Palabras o términos clave dentro del texto, tecnicismos que se quieran resaltar.

Todas las páginas deben ser numeradas a partir de la Introducción con el número 1.

Tamaño de la letra: Texto 11 puntos. ; Subtítulos 12 puntos. ; Títulos14 puntos.

Siempre que se utilicen abreviaturas deben ser explicadas la primera vez que aparezcan a excepción de las unidades del sistema métrico decimal, abreviaturas latinas, estadísticos y abreviaturas aceptadas como palabras.

Las citas textuales que no excedan de 40 palabras deben ir entrecomillas; cuando excedan, se separan en un párrafo aparte, no olvidando señalar el autor, año y página.

**NOTA: El documento que se entregará a la biblioteca deberá convertirse a formato con extensión .pdf.**

**Referencias bibliográficas consultadas:**

Bernal T., C. A. (2000). Metodología de la investigación para administración y economía. Colombia: Pearson Educación.

Hernández, R. (1991). Metodología de la investigación (2ª. Ed.). México: McGraw-Hill.

Mercado, S. (2008). ¿Cómo hacer una tesis?: Licenciatura, maestría, doctorado. (4ª. Ed.). México: Limusa.

Muñoz, C. (1998). Cómo elaborar y asesorar una investigación de tesis. México: Prentice Hall.

Schmelkes, C. (1998). Manual para la presentación de anteproyectos e informes de investigación (tesis). (2ª. Ed.). México: Oxford University Press.