

Universidad Autónoma Metropolitana
División de Ciencias Básica e Ingeniería

Licenciatura en Ingeniería en Computación

Modalidad: Experiencia Profesional

Sistema de Gestión de Promoción
“SIGPRO”

Alumno: Ángel Grajales Pérez

Matricula: 207201617

Empresa: Editorial Esfinge S. de R.L.

Departamento: Sistemas y Tecnología

Jefe Inmediato: Gabriel Torres Messina

Puesto: Director General

Trimestre 17-I

Declaratoria

Yo, Gabriel Torres Messina, declaro que aprobé el contenido del presente Reporte de Proyecto de Integración y doy mi autorización para su publicación en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Yo, Ángel Grajales Pérez, doy mi autorización a la Coordinación de Servicios de Información de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco, para publicar el presente documento en la Biblioteca Digital, así como en el Repositorio Institucional de UAM Azcapotzalco.



Resumen del Proyecto

El siguiente proyecto se realizó en la empresa Esfinge S. de R.L. en el departamento de Sistemas y Tecnología. El cual consistió en el desarrollo de un software para gestionar la promoción de libros de la empresa.

Este software fue un requerimiento para llevar un control más amplio y detallado de donde se entregan los libros, ya se a colegios o distribuidores de libros, lleva varios módulos en conjuntos que trabajan conjuntamente con diferentes tipos de roles en su operación desde el área de servicio a clientes hasta el gerente de ventas que evalúa el desempeño de cada uno de sus representantes de venta.

El sistema de información llamado SIGPRO está hecho en un lenguaje de programación de Microsoft llamado C#(C-sharp), debido a que es un acoplamiento del ERP (Planificación de Recursos Empresariales) principal llamado "SAP Bussines One" el cual funciona bajo el sistema operativo Windows.

Por tal motivo se decidió hacerlo en el lenguaje c# que es compatible con SQL Server ambos para Windows, también se incluye un Web Service (Servicio de datos) Fig. 1.2 el cual lleva y trae los datos SigproMovil a la base de datos, todos los módulos se desarrollaron con esta tecnología de Windows a excepción del móvil que es con Android.

El sistema se encuentra ya trabajando en producción con los representantes de ventas y los involucrados dentro de la empresa en sus roles asignados, podemos encontrarlo en un servidor con Windows Server 2008 R2 ubicado en las instalaciones de la empresa.

Actualmente el soporte de dicho servidor corre a cargo del área, ya que tiene enlace directo a la base de datos de "SAP Bussines One" que se encuentra en otro servidor y hace la conexión internamente para el intercambio de datos necesarios para la operación.

La parte del software que está hecha en Android está instalada en los celulares de los representantes de ventas, los cuales hacen el envío de datos a través del Web Service directamente a nuestro servidor de la empresa por medio de una IP Publica que enlaza al servidor donde está la aplicación Web.

Contenido

Empresa	6
Departamento	6
Sistemas y Tecnología	6
Actividades principales que desarrollo	7
Introducción	8
¿Qué es SIGPRO?.....	8
Objetivo del Sistema	8
Diagrama General	8
Diseño del sistema	9
SIGPRO	10
Sistema de Gestión de Promoción	10
Control de Versiones	11
Duración del proyecto.....	11
Roles de usuario	11
Pre-condiciones de uso para SIGPRO.....	11
Diagrama de Clases	12
Modelo Entidad Relación – Diccionario de datos	13
Ejemplo Código Fuente	14
Codigo de Store Procedure Login.....	14
Codigo de Store Procedure de BuscarEscuela.....	14
Codigo de Log	16
Codigo de Controlador Del Login	17
Codigo De La Vista Del Login	18
Codigo Del Modelo Del Login	19
Codigo del Controlador Escuela	20
Código Del Modelo De Escuela	30
Codigo De La Vista De Buscaescuelaxrepresentante	44
Diagrama de Flujo SIGPRO	49
Manuales.....	50
SigproMovil	50
Flujo de operación.....	51
Modulos	54
Asignación De Presupuesto.....	54
Pedidos de promoción	55
Promociones Asignadas	56
Móvil.....	57
Validación de Gerencia.....	57

Pruebas	59
Plan.....	59
Pruebas de Carga.....	59
Otros Proyectos.....	60
App Precios Esfinge	60
Plataforma educativa para Primaria (SIMPLE-Conacyt).....	60
Apéndice A. Glosario de Términos.....	61
BIBLIOGRAFIA	62

Empresa

Editorial Esfinge es una empresa dedicada al desarrollo de contenidos educativos para la educación de México con más de 50 años en el ramo. Continúa fiel al compromiso de apoyar a la educación mexicana. Este compromiso se ha manifestado en la gran calidad editorial, de la que dan fe los títulos clásicos que siguen siendo indispensables para la educación mexicana pero también la propuesta de las nuevas generaciones de autores mexicanos que orgullosamente forman parte del catálogo actual de Editorial Esfinge.

Editorial Esfinge tiene como misión apoyar a maestros y alumnos con herramientas pedagógicas de calidad, creadas bajo un código ético para alcanzar el desarrollo pleno de la educación en México.

La empresa editorial esfinge genera materiales educativos impresos y digitales para nivel: preescolar, primaria, secundaria y bachillerato.

Departamento

Sistemas y Tecnología

El departamento en donde se desarrolló el proyecto se llama Sistemas y Tecnología, se encarga de dos áreas principales en el ámbito laboral que son el soporte y el desarrollo de software para garantizar la continuidad operativa de los servicios tecnológicos y de infraestructura de la empresa.

Los ingenieros de desarrollo llevan a cabo todos los requerimientos de software de la editorial, para llevar a cabo una automatización de procesos que tenga un beneficio para la empresa.

Los ingenieros de soporte hacen la instalación, desde el circuito cerrado hasta los servidores, reparación de los equipos de cómputo, impresoras, cámaras CCTV, Red, Servidores, Telefonía, así como la creación y administración de Software de tipo administrativo como de venta (plataforma educativa digital).

Actividades principales que desarrollo

- Coordinar las áreas de soporte y desarrollo de software en la empresa.
- Desarrollar software interactivo educativo de apoyo para los diferentes títulos de libros en educación básica y media.
- Elaborar planes para el desarrollo, mejora y optimización continua de los sistemas de información y estándares de calidad de los procesos internos del área.
- Definir políticas y estándares de seguridad de la información.
- Planificar, diseñar, crear y monitorear los sistemas de información de la empresa.
- Elaborar planes y ejecutar acciones de capacitación orientadas al mejoramiento continuo del personal de la organización para el aprovechamiento de la tecnología.
- Proponer, crear y mantener la óptima infraestructura de red.
- Mantener el hardware y software en óptimas condiciones para atender las necesidades y el cumplimiento de los procesos de tecnología incorporados a la empresa.
- Propiciar la investigación, desarrollo y aplicación de nuevas tecnologías asociadas a la mejora de competencias para la institución.
- Crear planes de desarrollo de nuevos productos digitales tecnológicos de la organización.

Introducción

¿Qué es SIGPRO?

SIGLAS: Sistema de Gestion de Promocion y es un software para administrar la gestion de promocion de la empresa Editorial Esfinge.

Objetivo del Sistema

Gestionar la promoción de libros, informa en que institucion se quedó la muestra del libro, que representante lo dejo y el histórico.

Diagrama General

En la figura 1.0 se muestra la integración de los módulos principales que conforman la aplicación llamada SIGPRO y su relación en el orden consecutivo.

La figura 1.0 en color azul se muestra quien opera cada módulo y en blanco el nombre de los módulos, va un proceso de modulo a modulo iniciando con la asignación de presupuesto, pasando por el módulo de promoción que tiene relación con SIGPROMOVIL y finalizando con el módulo de liberación en SAP.



Fig. 1.0 Diagrama General SIGPRO

Diseño del sistema

SIGPRO es un acoplamiento al ERP (en inglés de Enterprise Resource Planning, , Planificación de Recursos Empresariales) principal llamado "SAP Bussines One" del cual se obtiene información de los representantes.

Se divide en 5 módulos principales y uno de administracion.

Los módulos son:

- Asignación de presupuesto
- Pedidos
- Promociones asignadas
- SigproMóvil
- Validación
- Administracion(únicamente para uso de Sistemas y Tecnologia)

Podemos ver en la siguiente URL el video del flujo que lleva SIGPRO.

<https://www.youtube.com/watch?v=z8XJ4PIPRvI>

Se muestran la relación de los módulos en la figura 1.1. En color azul está el nombre del módulo, mientras que en color blanco las acciones que se realiza en dicho modulo, así las flechas llevan el proceso de modulo a modulo.

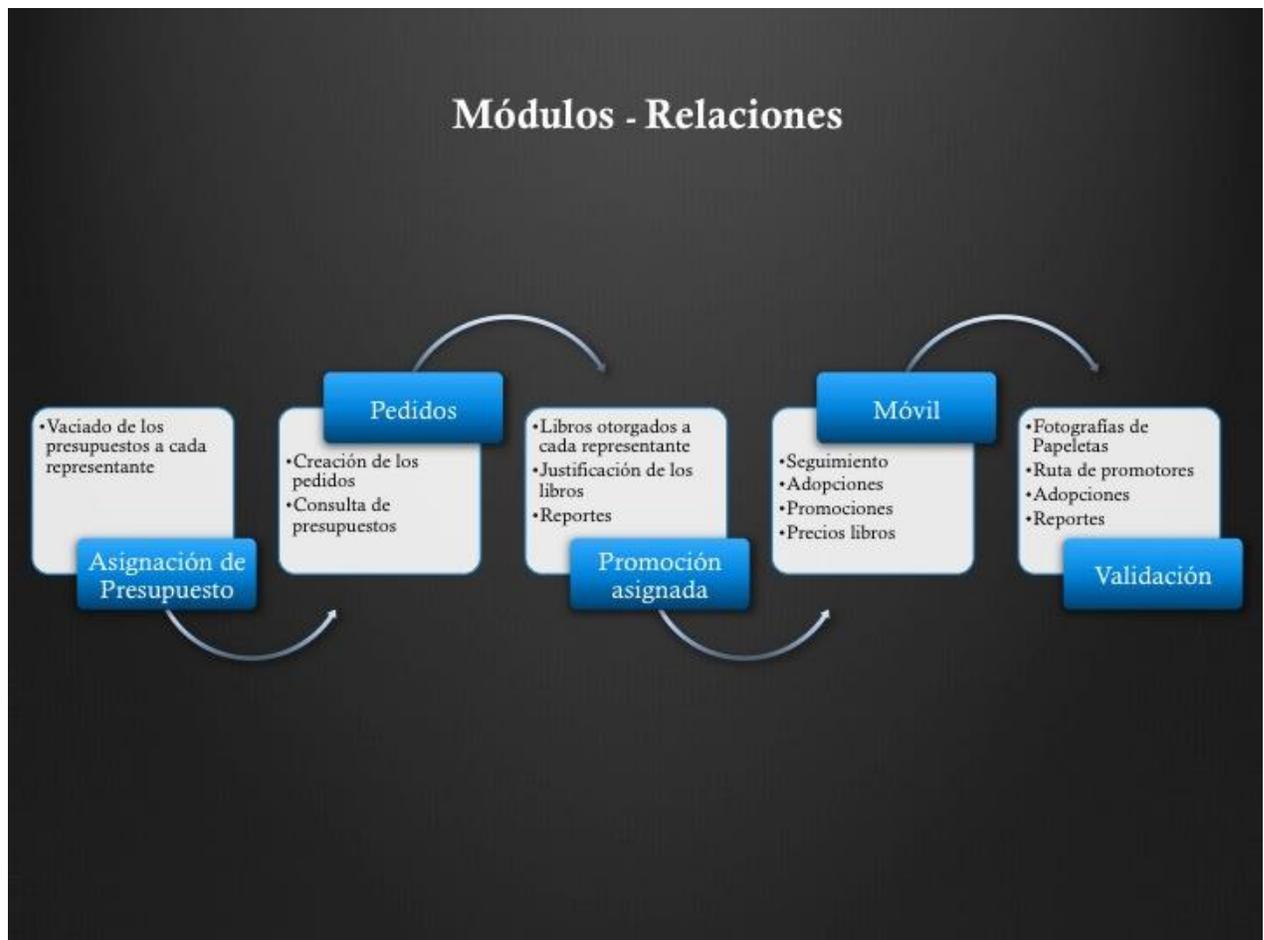


Fig. 1.1 Relación de Módulos

SIGPRO

Sistema de Gestión de Promoción

El sistema de información llamado SIGPRO está hecho en un lenguaje de programación de Microsoft llamado C#(C-sharp), debido a que es un acoplamiento del ERP (Planificación de Recursos Empresariales) principal llamado "SAP Bussines One" el cual funciona bajo el sistema operativo Windows.

Por tal motivo se decidió hacerlo en el lenguaje c# que es compatible con SQL Server ambos para windows, también se incluye un Web Service (Servicio de datos) Fig. 1.2 el cual lleva y trae los datos SigproMovil a la base de datos, todos los módulos se desarrollaron con esta tecnología de Windows a excepción del móvil que es con Android.

Un Web Service es un medio de comunicación entre un servidor en un lenguaje universal con el cual se ofrecen datos para cualquier cliente que lo requiera.

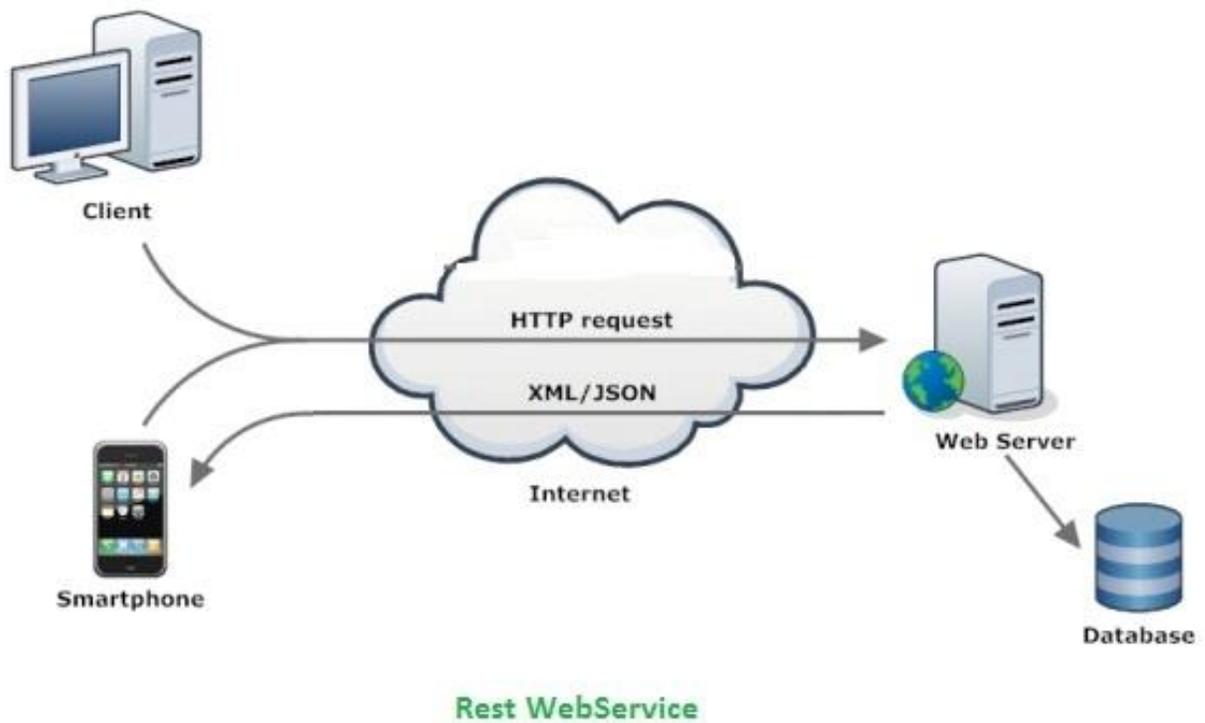


Fig. 1.2 Diagrama de Web Service

Control de Versiones

El control de versiones es para futuros cambios en el software, así que se debe contemplar un repositorio con alguna herramienta para que puedan hacerse modificaciones.

Esta es la primera versión de software inicial 1.0, si se requiere un nuevo requerimiento se tendrá que hacer la versión 2.0. si se trata de un cambio muy drástico, de lo contrario será decimales.

Se usará el plug-in Tortoise como programa de control de versiones en el software Visual Studio, este programa se encargara de gestionar las versiones en código fuente para futuros cambios.

Duración del proyecto

SIGPRO se realizó en un promedio de 8 meses con 2 desarrolladores, con las siguientes fases:

1. Problemática
2. Requerimientos
3. Análisis
4. Diseño
5. Desarrollo
6. Pruebas
7. Implementación

SIGPRO cuenta con cinco roles especificados en la tabla 1.0

Roles de usuario

En la tabla 1.0 se muestra están los diferentes roles que tiene el sistema con sus respectivos módulos asignados.

Rol	Módulos			
	Presupuesto	Pedidos	Promoción	Validación
Administrador	X	X	X	X
Gerente	X		X	X
Representante			X	
Servicio al cliente		X		
Asistente	X			

Tabla 1.0 Tabla de Roles

Pre-condiciones de uso para SIGPRO

1. El representante debe tener un presupuesto de venta asignado por él asistente.
2. El representante debe tener escuelas de promocion asignadas por el gerente comercial.
3. Al representante se le creara una cuenta en SAP por Sistemas y Tecnología.
4. Servicio al Cliente de Capturar los datos de los representantes en SAP

Diagrama de Clases

Los diagramas de clases muestran como esta internamente distribuidos y particionados los diferentes elementos de programación llamadas clases en el software.

Siguiendo el modelo de programación MVC (Modelo Vista Controlador) tenemos las clases en las figuras 1.3 y 1.4

En la figura 1.3 encontramos todas las clases que son controladores (son los que llevan la lógica del software).



Fig. 1.3 Diagrama de Clases de Controladores

En la figura 1.4 encontramos todas las clases que son modelos (son los que llevan la información de la base de datos).

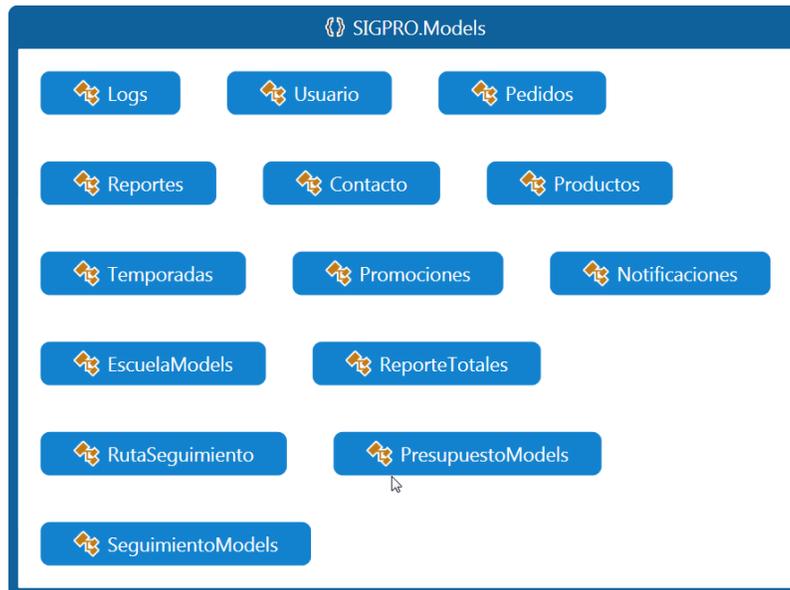


Fig. 1.4 Diagrama de Clases de Modelos

Modelo Entidad Relación – Diccionario de datos

El modelo de datos que permite representar la información de un sistema formado por un conjunto de objetos (tablas de información), vemos que en la Fig. 1.5 se cuenta con dieciocho tablas donde se almacenara toda la información de SIGPRO.



Fig. 1.5 Diagrama de la Base de datos

Ejemplo Código Fuente

Codigo de Store Procedure Login

```
USE [SIGPRO]
GO
/***** Object:  StoredProcedure [dbo].[ListaUsuarios]    Script Date:
03/28/2017 14:44:21 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Angel
-- Create date:
-- Description:
-- =====
ALTER PROCEDURE [dbo].[ListaUsuarios]
    @Rol    NVarChar(max)= 0,
    @Nivel  NVarChar(max) = ''
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;
    -- Insert statements for procedure here
    if(@Rol='gerente')
    begin
        SELECT Usuarios.Nombre,Usuarios.IDNivel,Niveles.Nombre from
        Usuarios inner join Niveles on
        Niveles.IDNivel=Usuarios.IDNivel where ClaveAlmacen like 'RE-%' and
        Usuarios.IDNivel in (select IDNivel from Niveles where Nombre like
        '%'+@Nivel+'%')
    end
    else
    begin
        SELECT Usuarios.Nombre from
        Usuarios where ClaveAlmacen like 'RE-%'
    end
END
```

Codigo de Store Procedure de BuscarEscuela

```
USE [SIGPRO]
GO
/***** Object:  StoredProcedure [dbo].[BuscarEscuela]    Script Date: 03/28/2017
14:55:47 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          Angel
-- Create date:
-- Description:
-- =====
ALTER PROCEDURE [dbo].[BuscarEscuela]
    -- Add the parameters for the stored procedure here
    @idEscuela uniqueidentifier = null,
    @idUserario uniqueidentifier = null,
    @rol nchar(18) = ''
```

```

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    if @rol='administrador'
    begin
        SELECT
            c1.Clave
            , c1.[IDEscuela]
        ,c1.[Nombre]
        ,c1.[Estado]
        ,c1.[Municipio]
        ,c1.[Colonia]
        ,c1.[Direccion]
        ,Niveles.Nombre as Nivel
        ,c1.[Tipo]
        ,c1.[Telefono]
        ,c1.[Email]
        ,c1.[NumAlumnos]
        ,c1.[Longitud]
        ,c1.[Latitud]
        ,c1.[FechaMod]
        ,c1.[Clave],
        c4.Nombre as Zona
        ,c1.Nivel as subnivel
        FROM [dbo].[Usuarios_Escuelas] c2
        right join [dbo].[Escuelas] c1
        on c1.IDEscuela=c2.IDEscuela
        inner join [dbo].[Zonas_Estados] c3
        on c3.Estado=c1.Estado
        inner join [dbo].[Zonas] c4
        on c4.IDZona=c3.IDZona
        inner join Niveles
        on Niveles.IDNivel=c1.IDNivel
        where c1.[IDEscuela]= @idEscuela
    end
    else if @rol='gerente'
    begin
        SELECT c1.Clave,
            c1.[IDEscuela]
        ,c1.[Nombre]
        ,c1.[Estado]
        ,c1.[Municipio]
        ,c1.[Colonia]
        ,c1.[Direccion]
        ,Niveles.Nombre as Nivel
        ,c1.[Tipo]
        ,c1.[Telefono]
        ,c1.[Email]
        ,c1.[NumAlumnos]
        ,c1.[Longitud]
        ,c1.[Latitud]
        ,c1.[FechaMod]
        ,c1.[Clave],
        c4.Nombre as Zona
        ,c1.Nivel as subnivel FROM [dbo].[Usuarios_Escuelas] c2
        right join [dbo].[Escuelas] c1
        on c1.IDEscuela=c2.IDEscuela
        inner join [dbo].[Zonas_Estados] c3
        on c3.Estado=c1.Estado
        inner join [dbo].[Zonas] c4
        on c4.IDZona=c3.IDZona
        inner join Niveles
        on Niveles.IDNivel=c1.IDNivel
        where c1.[IDEscuela]= @idEscuela
    end
    else
    begin

```

```

        SELECT c1.Clave,
               c1.[IDEscuela]
,c1.[Nombre]
,c1.[Estado]
,c1.[Municipio]
,c1.[Colonia]
,c1.[Direccion]
,Niveles.Nombre as Nivel
,c1.[Tipo]
,c1.[Telefono]
,c1.[Email]
,c1.[NumAlumnos]
,c1.[Longitud]
,c1.[Latitud]
,c1.[FechaMod]
,c1.[Clave],
c4.Nombre as Zona
,c1.Nivel as subnivel FROM [dbo].[Usuarios_Escuelas] c2
    inner join [dbo].[Escuelas] c1
    on c1.IDEscuela=c2.IDEscuela
    inner join [dbo].[Zonas_Estados] c3
    on c3.Estado=c1.Estado
    inner join [dbo].[Zonas] c4
    on c4.IDZona=c3.IDZona
    inner join Niveles
    on Niveles.IDNivel=c1.IDNivel
    where c1.[IDEscuela]= @idEscuela
    and c2.[IDUsuario] = @idUsuario
end
end

```

Codigo de Log

```

USE [SIGPRO]
GO
/***** Object:  StoredProcedure [dbo].[CreaLog]      Script Date:
03/28/2017 15:01:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      Angel
-- Create date:
-- Description:
-- =====
ALTER PROCEDURE [dbo].[CreaLog]
    -- Add the parameters for the stored procedure here
    @idUsuario nvarchar(max) = null,
    @query nvarchar(max)=null
AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Insert statements for procedure here
    insert into Logs(Fecha, IDUsuario,Query)
values (GETDATE(),@idUsuario,@query);
END

```

Codigo de Controlador Del Login

```
using Helpers;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;

namespace SIGPRO.Controllers
{
    public class LoginController : Controller
    {
        Datos Usuario = null;
        Configuracion url=null;
        Models.Logs log = new Models.Logs();
        public ActionResult Index()
        {
            //Muestra la sección de login del sitio
            try
            {
                if (Session["Usuario"] != null) {
                    Usuario = Session["Usuario"] as Datos;
                    return redirecciona();
                }
                ViewData["nombre"] = "";
                return View();
            }
            catch (SystemException e)
            {
                ViewData["nombre"] = "";
                return View();
            }
        }

        protected void creaSesion(){
            Session["Usuario"] = Usuario;
            url=new Configuracion();
            Session["url"] = url.url;
        }
        public ActionResult destruyeSesion()
        {
            //Se destruyen los datos de sesión del usuario y se finaliza la sesión
            try
            {
                Usuario = Session["Usuario"] as Datos;
                log.creaLog(Usuario.idUsuario, "Logout");
                Usuario.estado = false;
                Usuario.nombre = "";
                Session["Usuario"] = Usuario;
                ViewData["nombre"] = "";
                Session.Abandon();
                return RedirectToAction("Index", "Login");
            }
            catch (SystemException e) {
                return RedirectToAction("Index", "Login");
            }
        }
        public bool sesionIniciada(){
            Usuario = Session["Usuario"] as Datos;
            return Usuario.estado;
        }
    }
}
```



```

        @Html.ValidationSummary(true, "Error al iniciar sesión. Por favor revise sus
datos.");
        <div class="row">
            <fieldset>
                <legend>Login</legend>
                <div class="campo col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">
                    @Html.LabelFor(u => u.NombreUsuario)
                </div>
                <div class="campo col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">
                    @Html.TextBoxFor(u => u.NombreUsuario)
                    @Html.ValidationMessageFor(u => u.NombreUsuario)
                </div>
                <div class="campo col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">
                    @Html.LabelFor(u => u.ClaveUsuario)
                </div>
                <div class="campo col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">
                    @Html.PasswordFor(u => u.ClaveUsuario)
                    @Html.ValidationMessageFor(u => u.ClaveUsuario)
                </div>
                <input class="campo col-xs-6 col-xs-offset-3 col-sm-4 col-sm-offset-4
col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-5" type="submit" value="Iniciar sesión"
/>
            </fieldset>
        </div>
    }
</div>

```

Codigo Del Modelo Del Login

```

using Helpers;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
namespace SIGPRO.Models
{
    public class Usuario
    {
        [Required]
        [Display(Name = "Nombre de usuario")]
        public string NombreUsuario { get; set; }

        [Required]
        [DataType(DataType.Password)]
        [Display(Name = "Contraseña de usuario")]
        public string ClaveUsuario { get; set; }

        Configuracion db = new Configuracion();
        public Datos esValido(string usuario, string clave)
        {
            Datos nombre = new Datos();
            using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
            {
                var consulta = new SqlCommand();
                consulta.CommandText = "ValidarUsuario";
                consulta.CommandType = CommandType.StoredProcedure;
                consulta.Connection = conexion;
            }
        }
    }
}

```

```

consulta.Parameters
    .Add(new SqlParameter("@usuario", SqlDbType.NVarChar))
    .Value = usuario;
consulta.Parameters
    .Add(new SqlParameter("@clave", SqlDbType.NVarChar))
    .Value = clave;
try
{
    conexion.Open();
    var resultado = consulta.ExecuteReader();
    if (resultado.HasRows)
    {
        resultado.Read();
        nombre.idUsuario = resultado["IDUsuario"].ToString().Trim();
        nombre.nombre = resultado["Nombre"].ToString().Trim();
        nombre.Usuario = resultado["Usuario"].ToString().Trim();
        nombre.rol = resultado["rol"].ToString().Trim();
        nombre.idZona = resultado["IDZona"].ToString().Trim();
        nombre.nombreZona = resultado["NombreZona"].ToString().Trim();
        nombre.claveAlmacen =
resultado["ClaveAlmacen"].ToString().Trim();
        nombre.nivel = resultado["nivel"].ToString().Trim();
        nombre.estado = true;
        resultado.Dispose();
        conexion.Dispose();
        return nombre;
    }
    else
    {
        resultado.Dispose();
        conexion.Dispose();
        nombre.estado = false;
        return nombre;
    }
}
catch (SystemException e)
{
    Console.WriteLine("Error al conectar con servidor");
    nombre.estado = false;
    return nombre;
}
};
}

```

Codigo del Controlador Escuela

```

using Helpers;
using System;
using System.Collections;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace SIGPRO.Controllers
{
    public class EscuelasController : Controller
    {
        Models.EscuelaModels escuela=null;
        Datos usuario = null;
        Roles autorizacion = new Roles();
        Models.Logs log = new Models.Logs();
        public ActionResult Index()
        {
            return RedirectToAction("Index", "Login");
        }
    }
}

```

```

}
public ActionResult buscaEscuelaXRepresentante()
{
    //Una vez logeado el usuario se le da acceso a la sección de escuelas
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            ViewData["nombre"] = usuario.nombre;
            ViewData["almacen"] = usuario.claveAlmacen;
            escuela = new Models.EscuelaModels();
            //Inicialmente se le muestran al usuario una lista con sus primeras
10 escuelas
            ArrayList escuelas =
escuela.getEscuelasXRepresentante(usuario.Usuario);
            ViewData["escuelas"] = escuelas;
            ViewData["rol"] = usuario.rol;
            return View();
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para buscar escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        return RedirectToAction("Index", "Login");
    }
}
/**Ya tiene control de rol***/
public ActionResult buscaEscuela(string Escuela, string Estado, string Municipio,
string Colonia, string Direccion, string Tipo, string Nivel,int alumnos=100000)
{
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            ViewData["nombre"] = usuario.nombre;
            ViewData["almacen"] = usuario.claveAlmacen;
            ViewData["rol"] = usuario.rol;
            escuela = new Models.EscuelaModels();
            ArrayList escuelas =
escuela.getEscuelasXRepresentanteFiltrado(usuario.idUsuario, Escuela, Estado, Municipio,
Colonia, Direccion, Tipo, Nivel, usuario.rol,alumnos);
            return Json(escuelas);
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para buscar escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        ArrayList escuelas = new ArrayList();
        var a = new { Nombre = "Escuela", Direccion = "Estado" };
    }
}

```

```

        escuelas.Add(a);
        return Json(escuelas);
    }
}
/****Ya tiene control de rol****/
public ActionResult creaEscuela() {
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_aniadir"))
        {
            ViewData["nombre"] = usuario.nombre;
            ViewData["almacen"] = usuario.claveAlmacen;
            ViewData["idEscuela"] = usuario.idEscuela;
            ViewData["nombreZona"] = usuario.nombreZona;
            ViewData["rol"] = usuario.rol;
            return View();
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para añadir escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        return RedirectToAction("Index", "Login");
    }
}
/****Ya tiene control de rol****/
public ActionResult actualizaEscuela(string idEscuela) {
    //Una vez logeado el usuario se le da acceso a la sección de escuelas
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_actualizar"))
        {
            if (idEscuela == null && usuario.idEscuela == null)
            {
                mens.estado = false;
                mens.mensaje = "Elija una escuela para actualizar";
                Session["mensaje"] = mens;
                return RedirectToAction("buscaEscuelaXRepresentante",
"Escuelas");
            }
            else if (idEscuela == null && usuario.idEscuela != null)
            {
                idEscuela = usuario.idEscuela;
            }
            usuario.idEscuela = idEscuela;
            ViewData["nombre"] = usuario.nombre;
            ViewData["almacen"] = usuario.claveAlmacen;
            ViewData["idEscuela"] = usuario.idEscuela;
            ViewData["nombreZona"] = usuario.nombreZona;
            ViewData["rol"] = usuario.rol;
            escuela = new Models.EscuelaModels();
            ViewData["escuela"] = escuela.buscaEscuela(usuario.idUsuario,
idEscuela,usuario.rol);
            return View();
        }
        else
        {
            mens.estado = false;

```

```

        mens.mensaje = "No está autorizado para actualizar escuelas";
        Session["mensaje"] = mens;
        ViewData["rol"] = usuario.rol;
        return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
    }
}
else
{
    return RedirectToAction("Index", "Login");
}
}
/**Ya tiene control de rol**/
public ActionResult buscaNombreEscuela(string nombreEscuela){
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            escuela = new Models.EscuelaModels();
            ArrayList Escuelas;
            if (usuario.rol == "administrador")
            {
                Escuelas = escuela.buscaNombreEscuela("-1", nombreEscuela);
            }
            else
            {
                Escuelas = escuela.buscaNombreEscuela(usuario.idUsuario,
nombreEscuela);
            }
            return Json(Escuelas);
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para buscar escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        return null;
    }
}
/**Ya tiene control de rol**/
public ActionResult buscaDireccionEscuela(string direccion)
{
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            escuela = new Models.EscuelaModels();
            ArrayList Escuelas;
            if (usuario.rol == "administrador")
            {
                Escuelas = escuela.buscaDireccionEscuela("-1", direccion);
            }
            else
            {
                Escuelas = escuela.buscaDireccionEscuela(usuario.idUsuario,
direccion);
            }
            return Json(Escuelas);
        }
    }
}

```

```

    }
    else
    {
        mens.estado = false;
        mens.mensaje = "No está autorizado para buscar escuelas";
        Session["mensaje"] = mens;
        ViewData["rol"] = usuario.rol;
        return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
    }
}
else
{
    return null;
}
}
/****Ya tiene control de rol**/
public ActionResult buscaColoniaEscuela(string colonia)
{
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            escuela = new Models.EscuelaModels();
            ArrayList Escuelas;
            if (usuario.rol == "administrador")
            {
                Escuelas = escuela.buscaColoniaEscuela("-1", colonia);
            }
            else
            {
                Escuelas = escuela.buscaColoniaEscuela(usuario.idUsuario,
colonia);
            }
            return Json(Escuelas);
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para buscar escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        return null;
    }
}
/****Ya tiene control de rol**/
public ActionResult buscaMunicipioEscuela(string municipio)
{
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            escuela = new Models.EscuelaModels();
            ArrayList Escuelas;
            if (usuario.rol == "administrador")
            {
                Escuelas = escuela.buscaMunicipioEscuela("-1", municipio);
            }
            else
            {

```



```

        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para buscar escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        return null;
    }
}
/****Ya tiene control de rol****/
public ActionResult buscaTipoEscuela()
{
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            escuela = new Models.EscuelaModels();
            ArrayList Escuelas = escuela.buscaTipoEscuela(usuario.idUsuario);
            return Json(Escuelas);
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para buscar escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        return null;
    }
}
/****Ya tiene control de rol****/
public ActionResult buscaZonaUsuario()
{
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_buscar"))
        {
            ArrayList Escuelas = new ArrayList();
            var reg = new
            {
                Zona = usuario.nombreZona
            };
            Escuelas.Add(reg);
            return Json(Escuelas);
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No está autorizado para buscar escuelas";
            Session["mensaje"] = mens;
            ViewData["rol"] = usuario.rol;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {

```

```

        return null;
    }
}
/****Ya tiene control de rol****/
public ActionResult borraEscuela(string idEscuela)
{
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_borrar"))
        {
            escuela = new Models.EscuelaModels();
            bool estado = escuela.borraEscuela(idEscuela);
            mens.estado = estado;
            if (estado)
            {
                log.creaLog(usuario.idUsuario, "borraEscuela");
                mens.mensaje = "La escuela se borró correctamente";
            }
            else
            {
                mens.mensaje = "La escuela no se borró porque tiene promociones
asignadas";
            }
            Session["mensaje"] = mens;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
        else
        {
            mens.estado = false;
            mens.mensaje = "No tienes suficiente privilegios para borrar
escuelas";
            Session["mensaje"] = mens;
            return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
        }
    }
    else
    {
        return RedirectToAction("Index", "Login");
    }
}
[HttpPost]
/****Ya tiene control de rol**/
public ActionResult guardaEscuela() {
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_aniadir"))
        {
            var claveEscuela = (Request["claveEscuela"].ToString() != "---" ?
Request["claveEscuela"].ToString() : "");
            var idEscuela = (Request["idEscuela"].ToString() != "---" ?
Request["idEscuela"].ToString() : null);
            var nomEscuela = (Request["nomEscuela"].ToString() != "---" ?
Request["nomEscuela"].ToString() : null);
            var nomDireccion = (Request["direccion"].ToString() != "---" ?
Request["direccion"].ToString() : null);
            var nomColonia = (Request["colonia"].ToString() != "---" ?
Request["colonia"].ToString() : null);
            var nomMunicipio = (Request["municipio"].ToString() != "---" ?
Request["municipio"].ToString() : null);
            var nomEstado = (Request["estado"].ToString() != "---" ?
Request["estado"].ToString() : null);
            var nomTelefono = (Request["telefono"].ToString() != "---" ?
Request["telefono"].ToString() : null);

```

```

        var nomCorreo = (Request["correo"].ToString() != "---" ?
Request["correo"].ToString() : null);
        var nomTipo = (Request["tipo"].ToString() != "---" ?
Request["tipo"].ToString() : null);
        var nomNivel = (Request["nivel"].ToString() != "---" ?
Request["nivel"].ToString() : null);
        var subnivel = (Request["subnivel"].ToString() != "---" ?
Request["subnivel"].ToString() : null);
        var numAlumnos = (Request["numAlumnos"].ToString() != "---" ?
Request["numAlumnos"].ToString() : "");
        if (nomEscuela == "" && nomEstado == "")
            return RedirectToAction("creaEscuela", "Escuelas");
        escuela = new Models.EscuelaModels();
        bool estado = escuela.updateEscuela(claveEscuela, idEscuela,
nomEscuela, nomEstado, nomMunicipio, nomColonia, nomDireccion, nomNivel, nomTipo,
nomTelefono, nomCorreo, numAlumnos, subnivel);
        if (!estado)
        {
            mens.estado = false;
            mens.mensaje = "No ha sido posible actualizar la escuela, por
favor verifique sus datos";
            Session["mensaje"] = mens;
            return RedirectToAction("actualizaEscuela", "Escuelas", new {
idEscuela = idEscuela });
        }
        else
        {
            mens.estado = true;
            mens.mensaje = "La escuela se actualizó correctamente";
            Session["mensaje"] = mens;
            log.crealog(usuario.idUsuario, "updateEscuela");
            return RedirectToAction("buscaEscuelaXRepresentante",
"Escuelas");
        }
    }
    else
    {
        mens.estado = false;
        mens.mensaje = "No tienes suficiente privilegios para borrar
escuelas";
        Session["mensaje"] = mens;
        return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
    }
}
else {
    return RedirectToAction("Index", "Login");
}
}
/****Ya tiene control de rol****/
public ActionResult guardaNuevaEscuela()
{
    //Una vez logeado el usuario se le da acceso a la sección de escuelas
    if (Session["Usuario"] != null)
    {
        Mensajes mens = new Mensajes();
        usuario = Session["Usuario"] as Datos;
        if (autorizacion.tienePrivilegios(usuario.rol, "escuela_aniadir"))
        {
            var claveEscuela = (Request["claveEscuela"].ToString() != "---" ?
Request["claveEscuela"].ToString() : "");
            var nomEscuela = (Request["nomEscuela"].ToString() != "---" ?
Request["nomEscuela"].ToString() : "");
            var nomDireccion = (Request["direccion"].ToString() != "---" ?
Request["direccion"].ToString() : "");
            var nomColonia = (Request["colonia"].ToString() != "---" ?
Request["colonia"].ToString() : "");
            var nomMunicipio = (Request["municipio"].ToString() != "---" ?
Request["municipio"].ToString() : "");

```

```

        var nomEstado = (Request["estado"].ToString() != "---" ?
Request["estado"].ToString() : "");
        var nomTelefono = (Request["telefono"].ToString() != "---" ?
Request["telefono"].ToString() : "");
        var nomCorreo = (Request["correo"].ToString() != "---" ?
Request["correo"].ToString() : "");
        var nomTipo = (Request["tipo"].ToString() != "---" ?
Request["tipo"].ToString() : "");
        var nomNivel = (Request["nivel"].ToString() != "---" ?
Request["nivel"].ToString() : "");
        var subNivel = (Request["subnivel"].ToString() != "---" ?
Request["subnivel"].ToString() : "");
        var numAlumnos = (Request["numAlumnos"].ToString() != "---" ?
Request["numAlumnos"].ToString() : "");

        if (nomEscuela == "" && nomColonia == "" && nomMunicipio == "" &&
nomEstado == "")
            return RedirectToAction("creaEscuela", "Escuelas");

        escuela = new Models.EscuelaModels();
        bool estado = escuela.creaEscuela(claveEscuela, usuario.idUsuario,
nomEscuela, nomEstado, nomMunicipio, nomColonia, nomDireccion, nomNivel, nomTipo,
nomTelefono, nomCorreo, numAlumnos, subNivel);
        if (!estado)
        {
            mens.estado = false;
            mens.mensaje = "La escuela no se pudo crear";
            Session["mensaje"] = mens;
            return RedirectToAction("creaEscuela", "Escuelas");
        }
        else
        {
            mens.estado = true;
            mens.mensaje = "La escuela se creó correctamente";
            Session["mensaje"] = mens;
            log.creaLog(usuario.idUsuario, "creaEscuela");
            return RedirectToAction("creaEscuela", "Escuelas");
        }
    }
    else
    {
        mens.estado = false;
        mens.mensaje = "No tienes suficiente privilegios para anadir
escuelas";

        Session["mensaje"] = mens;
        return RedirectToAction("buscaEscuelaXRepresentante", "Escuelas");
    }
}
else
{
    return RedirectToAction("Index", "Login");
}
}
public ActionResult buscaEstados() {
    escuela = new Models.EscuelaModels();
    var resultado=escuela.listaEstados();
    return Json(resultado);
}
public ActionResult buscaMunicipios()
{
    escuela = new Models.EscuelaModels();
    var resultado = escuela.listaMunicipios();
    return Json(resultado);
}
public ActionResult buscaNiveles()
{
    escuela = new Models.EscuelaModels();
    var resultado = escuela.listaNiveles();
}

```



```

        .Value = usuario.Trim();
    try
    {
        conexion.Open();
        var resultado = consulta.ExecuteReader();
        if (resultado.HasRows)
        {
            while (resultado.Read()) {
                ArrayList reglon = new ArrayList();
                reglon.Add(resultado["IDEscuela"].ToString().Trim());
                reglon.Add(resultado["Nombre"].ToString().Trim());
                reglon.Add(resultado["Clave"].ToString().Trim());
                reglon.Add(resultado["Direccion"].ToString().Trim());
                reglon.Add(resultado["Colonia"].ToString().Trim());
                reglon.Add(resultado["Municipio"].ToString().Trim());
                reglon.Add(resultado["Estado"].ToString().Trim());
                reglon.Add(resultado["Nivel"].ToString().Trim());
                reglon.Add(resultado["NumAlumnos"].ToString().Trim());
                Escuelas.Add(reglon);
            }

            resultado.Dispose();
            conexion.Dispose();
            return Escuelas;
        }
        else
        {
            resultado.Dispose();
            conexion.Dispose();
            return Escuelas;
        }
    }
    catch (SystemException e)
    {
        Console.WriteLine("Error al conectar con servidor");
        return Escuelas;
    }
};
}
public ArrayList getEscuelasXRepresentanteFiltrado(string idUsuario, string
Escuela, string Estado, string Municipio, string Colonia, string Direccion, string Tipo,
string Nivel, string rol, int alumnos)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText="buscarEscuelaFiltrado";
        consulta.CommandType=CommandType.StoredProcedure;
        consulta.Parameters
            .Add(new SqlParameter("@idUsuario", SqlDbType.NChar))
            .Value = idUsuario;
        consulta.Parameters
            .Add(new SqlParameter("@escuela", SqlDbType.NChar))
            .Value = Escuela.Trim() ;
        consulta.Parameters
            .Add(new SqlParameter("@estado", SqlDbType.NChar))
            .Value = Estado.Trim() ;
        consulta.Parameters
            .Add(new SqlParameter("@municipio", SqlDbType.NChar))
            .Value = Municipio.Trim() ;
        consulta.Parameters
            .Add(new SqlParameter("@direccion", SqlDbType.NChar))
            .Value = Direccion.Trim() ;
        consulta.Parameters
            .Add(new SqlParameter("@colonia", SqlDbType.NChar))

```

```

        .Value = Colonia.Trim() ;
consulta.Parameters
    .Add(new SqlParameter("@nivel", SqlDbType.NChar))
    .Value = Nivel.Trim() ;
consulta.Parameters
    .Add(new SqlParameter("@tipo", SqlDbType.NChar))
    .Value = Tipo.Trim();
consulta.Parameters
    .Add(new SqlParameter("@rol", SqlDbType.NChar))
    .Value = rol.Trim() ;
consulta.Parameters
    .Add(new SqlParameter("@alumnos", SqlDbType.Int))
    .Value = alumnos;
consulta.Connection = conexion;
try
{
    conexion.Open();
    var resultado = consulta.ExecuteReader();
    if (resultado.HasRows)
    {
        while (resultado.Read())
        {
            var reglones = new
            {
                IDEscuela = resultado["IDEscuela"].ToString().Trim() !=
"" ? resultado["IDEscuela"].ToString().Trim() : "",
                Escuela = resultado["Nombre"].ToString().Trim() != "" ?
resultado["Nombre"].ToString().Trim() : "",
                Direccion = resultado["Direccion"].ToString().Trim() !=
"" ? resultado["Direccion"].ToString().Trim() : "",
                Colonia = resultado["Colonia"].ToString().Trim() != "" ?
resultado["Colonia"].ToString().Trim() : "",
                Municipio = resultado["Municipio"].ToString().Trim() !=
"" ? resultado["Municipio"].ToString().Trim() : "",
                Estado = resultado["Estado"].ToString().Trim() != "" ?
resultado["Estado"].ToString().Trim() : "",
                Nivel = resultado["Nivel"].ToString().Trim() != "" ?
resultado["Nivel"].ToString().Trim() : "",
                Tipo = resultado["Tipo"].ToString().Trim() != "" ?
resultado["Tipo"].ToString().Trim() : "",
                Alumnos = resultado["NumAlumnos"].ToString().Trim() != ""
? resultado["NumAlumnos"].ToString().Trim() : "",
                Clave=(resultado["Clave"].ToString().Trim()),
                Rol = rol
            };
            Escuelas.Add(reglones);
        }

        resultado.Dispose();
        conexion.Dispose();
        return Escuelas;
    }
    else
    {
        resultado.Dispose();
        conexion.Dispose();
        return Escuelas;
    }
}
catch (SystemException e)
{
    Console.WriteLine("Error al conectar con servidor");
    return Escuelas;
}
};
}
public ArrayList buscaNombreEscuela(string idUsuario, string Escuela)
{

```

```

        ArrayList Escuelas = new ArrayList();
        using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
        {
            var consulta = new SqlCommand();
            consulta.CommandText = "BuscaNombreEscuela";
            consulta.CommandType = CommandType.StoredProcedure;
            consulta.Connection = conexion;
            consulta.Parameters
                .Add(new SqlParameter("@idUsuario", SqlDbType.NChar))
                .Value = idUsuario;
            consulta.Parameters
                .Add(new SqlParameter("@escuela", SqlDbType.NChar))
                .Value = "%" + Escuela.Trim() + "%";
            try
            {
                conexion.Open();
                var resultado = consulta.ExecuteReader();
                if (resultado.HasRows)
                {
                    while (resultado.Read())
                    {
                        var reglones = new
                        {
                            Escuela = resultado[0].ToString().Trim() != "" ?
resultado[0].ToString().Trim() : "---",
                        };
                        Escuelas.Add(reglones);
                    }

                    resultado.Dispose();
                    conexion.Dispose();
                    return Escuelas;
                }
                else
                {
                    resultado.Dispose();
                    conexion.Dispose();
                    return Escuelas;
                }
            }
            catch (SystemException e)
            {
                Console.WriteLine("Error al conectar con servidor");
                return Escuelas;
            }
        }
    };
}
public ArrayList buscaDireccionEscuela(string idUsuario, string Direccion)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "BuscaDireccionEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        consulta.Parameters
            .Add(new SqlParameter("@idUsuario", SqlDbType.NChar))
            .Value = idUsuario;
        consulta.Parameters
            .Add(new SqlParameter("@direccion", SqlDbType.NChar))
            .Value = "%" + Direccion.Trim() + "%";
        try
        {

```

```

        conexion.Open();
        var resultado = consulta.ExecuteReader();
        if (resultado.HasRows)
        {
            while (resultado.Read())
            {
                var reglones = new
                {
                    Direccion = resultado[0].ToString().Trim() != "" ?
resultado[0].ToString().Trim() : "---",
                };
                Escuelas.Add(reglones);
            }

            resultado.Dispose();
            conexion.Dispose();
            return Escuelas;
        }
        else
        {
            resultado.Dispose();
            conexion.Dispose();
            return Escuelas;
        }
    }
    catch (SystemException e)
    {
        Console.WriteLine("Error al conectar con servidor");
        return Escuelas;
    }
};
}
public ArrayList buscaColoniaEscuela(string idUsuario, string colonia)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText="BuscaColoniaEscuela";
        consulta.CommandType=CommandType.StoredProcedure;
        consulta.Connection=conexion;
        consulta.Parameters
            .Add(new SqlParameter("@idUsuario", SqlDbType.NChar))
            .Value = idUsuario;

        consulta.Parameters
            .Add(new SqlParameter("@Colonia", SqlDbType.NChar))
            .Value = "%" + colonia + "%";
        try
        {
            {
                conexion.Open();
                var resultado = consulta.ExecuteReader();
                if (resultado.HasRows)
                {
                    while (resultado.Read())
                    {
                        var reglones = new
                        {
                            Colonia = resultado[0].ToString().Trim() != "" ?
resultado[0].ToString().Trim() : "---",
                        };
                        Escuelas.Add(reglones);
                    }
                }
            }
        }
    }
}

```

```

        resultado.Dispose();
        conexion.Dispose();
        return Escuelas;
    }
    else
    {
        resultado.Dispose();
        conexion.Dispose();
        return Escuelas;
    }
}
catch (SystemException e)
{
    Console.WriteLine("Error al conectar con servidor");
    return Escuelas;
}
};
}
public ArrayList buscaMunicipioEscuela(string idUsuario, string Municipio)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "BuscaMunicipioEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        if(idUsuario!="-1"){
            consulta.Parameters
                .Add(new SqlParameter("@idUsuario", SqlDbType.NChar))
                .Value = idUsuario;
        }
        consulta.Parameters
            .Add(new SqlParameter("@municipio", SqlDbType.NChar))
            .Value = "%" + Municipio.Trim() + "%";
        try
        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    var reglones = new
                    {
                        Municipio = resultado[0].ToString().Trim() != "" ?
resultado[0].ToString().Trim() : "---",
                    };
                    Escuelas.Add(reglones);
                }

                resultado.Dispose();
                conexion.Dispose();
                return Escuelas;
            }
            else
            {
                resultado.Dispose();
                conexion.Dispose();
                return Escuelas;
            }
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");

```

```

        return Escuelas;
    }
};
}
public ArrayList buscaEstadoEscuela(string idUsuario, string Estado)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "BuscaEstadoEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        consulta.Parameters.Add(new SqlParameter("@idUsuario", SqlDbType.NChar))
            .Value = idUsuario;
        consulta.Parameters.Add(new SqlParameter("@estado", SqlDbType.NChar))
            .Value = "%" + Estado.Trim() + "%";
        try
        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    var reglones = new
                    {
                        Estado = resultado[0].ToString().Trim() != "" ?
resultado[0].ToString().Trim() : "---",
                    };
                    Escuelas.Add(reglones);
                }

                resultado.Dispose();
                conexion.Dispose();
                return Escuelas;
            }
            else
            {
                resultado.Dispose();
                conexion.Dispose();
                return Escuelas;
            }
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return Escuelas;
        }
    }
};
}
public ArrayList buscaNivelEscuela(string idUsuario)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "BuscaNivelEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        consulta.Parameters.Add(new SqlParameter("@idUsuario", SqlDbType.NChar))
            .Value = idUsuario;
        try

```

```

    {
        conexion.Open();
        var resultado = consulta.ExecuteReader();
        if (resultado.HasRows)
        {
            while (resultado.Read())
            {
                var reglones = new
                {
                    Nivel = resultado[0].ToString().Trim()
                };
                Escuelas.Add(reglones);
            }

            resultado.Dispose();
            conexion.Dispose();
            return Escuelas;
        }
        else
        {
            resultado.Dispose();
            conexion.Dispose();
            return Escuelas;
        }
    }
}
catch (SystemException e)
{
    Console.WriteLine("Error al conectar con servidor");
    return Escuelas;
}
};
}
public ArrayList buscaTipoEscuela(string idUsuario)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"]
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "BuscaTipoEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        consulta.Parameters.Add(new SqlParameter("@idUsuario",
SqlDbType.NVarChar))
        .Value = idUsuario;
        try
        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    var reglones = new
                    {
                        Tipo = resultado[0].ToString().Trim()
                    };
                    Escuelas.Add(reglones);
                }

                resultado.Dispose();
                conexion.Dispose();
                return Escuelas;
            }
            else
            {
                resultado.Dispose();
            }
        }
    }
}

```

```

        conexion.Dispose();
        return Escuelas;
    }
}
catch (SystemException e)
{
    Console.WriteLine("Error al conectar con servidor");
    return Escuelas;
}
};
}
public ArrayList buscaEscuela(string idUsuario, string idEscuela, string rol)
{
    ArrayList Escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"]
.ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText="BuscarEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Parameters
            .Add(new SqlParameter("@idUsuario", SqlDbType.NVarChar))
            .Value = idUsuario;
        consulta.Parameters
            .Add(new SqlParameter("@idEscuela", SqlDbType.NVarChar))
            .Value = idEscuela;
        consulta.Parameters
            .Add(new SqlParameter("@rol", SqlDbType.NVarChar))
            .Value = rol;
        consulta.Connection = conexion;
        try
        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    ArrayList reglones = new ArrayList();
                    reglones.Add(resultado["IDEscuela"].ToString().Trim() != "" ?
resultado["IDEscuela"].ToString().Trim() : "");
                    reglones.Add(resultado["Nombre"].ToString().Trim() != "" ?
resultado["Nombre"].ToString().Trim() : "");
                    reglones.Add(resultado["Direccion"].ToString().Trim() != "" ?
resultado["Direccion"].ToString().Trim() : "");
                    reglones.Add(resultado["Colonia"].ToString().Trim() != "" ?
resultado["Colonia"].ToString().Trim() : "");
                    reglones.Add(resultado["Municipio"].ToString().Trim() != "" ?
resultado["Municipio"].ToString().Trim() : "");
                    reglones.Add(resultado["Estado"].ToString().Trim() != "" ?
resultado["Estado"].ToString().Trim() : "");
                    reglones.Add(resultado["Nivel"].ToString().Trim() != "" ?
resultado["Nivel"].ToString().Trim() : "");
                    reglones.Add(resultado["Tipo"].ToString().Trim() != "" ?
resultado["Tipo"].ToString().Trim() : "");
                    reglones.Add(resultado["Zona"].ToString().Trim() != "" ?
resultado["Zona"].ToString().Trim() : "");
                    reglones.Add(resultado["Telefono"].ToString().Trim() != "" ?
resultado["Telefono"].ToString().Trim() : "");
                    reglones.Add(resultado["Email"].ToString().Trim() != "" ?
resultado["Email"].ToString().Trim() : "");
                    reglones.Add(resultado["NumAlumnos"].ToString().Trim() != ""
? resultado["NumAlumnos"].ToString().Trim() : "");
                    reglones.Add(resultado["Clave"].ToString().Trim() != "" ?
resultado["Clave"].ToString().Trim() : "");
                    reglones.Add(resultado["subnivel"].ToString().Trim() != "" ?
resultado["subnivel"].ToString().Trim() : "");
                }
            }
        }
    }
}

```

```

        Escuelas.Add(reglones);
    }
    resultado.Dispose();
    conexion.Dispose();
    return Escuelas;
}
else
{
    resultado.Dispose();
    conexion.Dispose();
    return Escuelas;
}
}
catch (SystemException e)
{
    Console.WriteLine("Error al conectar con servidor");
    return Escuelas;
}
};
}
public bool updateEscuela(string claveEscuela, string IDEscuela, string nombre,
string estado, string municipio, string colonia, string direccion, string nivel, string
tipo, string telefono, string email, string numAlumnos, string subnivel) {
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText="ActualizarEscuela";
        consulta.CommandType=CommandType.StoredProcedure;
        consulta.Parameters
            .Add(new SqlParameter("@claveEscuela", SqlDbType.NChar))
            .Value = claveEscuela;
        consulta.Parameters
            .Add(new SqlParameter("@idEscuela", SqlDbType.NChar))
            .Value = IDEscuela;
        consulta.Parameters
            .Add(new SqlParameter("@nombre", SqlDbType.NChar))
            .Value = nombre;
        consulta.Parameters
            .Add(new SqlParameter("@estado", SqlDbType.NChar))
            .Value = estado;
        consulta.Parameters
            .Add(new SqlParameter("@municipio", SqlDbType.NChar))
            .Value = municipio;
        consulta.Parameters
            .Add(new SqlParameter("@colonia", SqlDbType.NChar))
            .Value = colonia;
        consulta.Parameters
            .Add(new SqlParameter("@direccion", SqlDbType.NChar))
            .Value = direccion;
        consulta.Parameters
            .Add(new SqlParameter("@nivel", SqlDbType.NChar))
            .Value = nivel;
        consulta.Parameters
            .Add(new SqlParameter("@subnivel", SqlDbType.NChar))
            .Value = subnivel;
        consulta.Parameters
            .Add(new SqlParameter("@tipo", SqlDbType.NChar))
            .Value = tipo;
        consulta.Parameters
            .Add(new SqlParameter("@telefono", SqlDbType.NChar))
            .Value = telefono;
        consulta.Parameters
            .Add(new SqlParameter("@email", SqlDbType.NChar))
            .Value = email;
        consulta.Parameters
            .Add(new SqlParameter("@numAlumnos", SqlDbType.Real))

```

```

        .Value = numAlumnos;
        consulta.Connection = conexion;
        try
        {
            conexion.Open();
            int resultado = consulta.ExecuteNonQuery();
            if (resultado==0)
            {
                conexion.Dispose();
                return false;
            }
            else
            {
                conexion.Dispose();
                return true;
            }
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return false;
        }
    };
}
public bool borraEscuela(string idEscuela){
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "BorrarEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        consulta.Parameters.Add(new SqlParameter("@idEscuela", SqlDbType.NChar))
            .Value = idEscuela;
        try
        {
            conexion.Open();
            int resultado = consulta.ExecuteNonQuery();
            if (resultado == 0)
            {
                conexion.Dispose();
                return false;
            }
            else
            {
                return true;
            }
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return false;
        }
    };
}
}
public bool creaEscuela(string claveEscuela, string IDUsuario, string nombre,
string estado, string municipio, string colonia, string direccion, string nivel, string
tipo, string telefono, string email, string numAlumnos, string subnivel)
{
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "CreaEscuela";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;

```

```

        consulta.Parameters
            .Add(new SqlParameter("@claveEscuela", SqlDbType.NChar))
            .Value = claveEscuela;
        consulta.Parameters
            .Add(new SqlParameter("@idUsuario", SqlDbType.NVarChar))
            .Value = IDUsuario;
        consulta.Parameters
            .Add(new SqlParameter("@nombre", SqlDbType.NChar))
            .Value = nombre;
        consulta.Parameters
            .Add(new SqlParameter("@estado", SqlDbType.NChar))
            .Value = estado;
        consulta.Parameters
            .Add(new SqlParameter("@municipio", SqlDbType.NChar))
            .Value = municipio;
        consulta.Parameters
            .Add(new SqlParameter("@colonia", SqlDbType.NChar))
            .Value = colonia;
        consulta.Parameters
            .Add(new SqlParameter("@direccion", SqlDbType.NChar))
            .Value = direccion;
        consulta.Parameters
            .Add(new SqlParameter("@nivel", SqlDbType.NChar))
            .Value = nivel;
        consulta.Parameters
            .Add(new SqlParameter("@subnivel", SqlDbType.NChar))
            .Value = subnivel;
        consulta.Parameters
            .Add(new SqlParameter("@tipo", SqlDbType.NChar))
            .Value = tipo;
        consulta.Parameters
            .Add(new SqlParameter("@telefono", SqlDbType.NChar))
            .Value = telefono;
        consulta.Parameters
            .Add(new SqlParameter("@email", SqlDbType.NChar))
            .Value = email;
        consulta.Parameters
            .Add(new SqlParameter("@numAlumnos", SqlDbType.Real))
            .Value = numAlumnos;
        try
        {
            conexion.Open();
            int resultado = consulta.ExecuteNonQuery();
            if (resultado == 0)
            {
                conexion.Dispose();
                return false;
            }
            return true;
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return false;
        }
    };
}
public ArrayList listaEstados() {
    ArrayList escuelas =new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "listarEstados";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        try

```

```

        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    escuelas.Add(resultado[0].ToString());
                }
                conexion.Dispose();
            }
            else {
                conexion.Dispose();
            }
            return escuelas;
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return escuelas;
        }
    }
};
}
public ArrayList listaMunicipios()
{
    ArrayList escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "listarMunicipios";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        try
        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    escuelas.Add(resultado[0].ToString());
                }
                conexion.Dispose();
            }
            else
            {
                conexion.Dispose();
            }
            return escuelas;
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return escuelas;
        }
    }
};
}
public ArrayList listaNiveles()
{
    ArrayList escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "listarNiveles";
    }
}

```

```

        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        try
        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    escuelas.Add(resultado[0].ToString());
                }
                conexion.Dispose();
            }
            else
            {
                conexion.Dispose();
            }
            return escuelas;
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return escuelas;
        }
    };
}
public ArrayList listaTipos()
{
    ArrayList escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"]
].ConnectionString))
    {
        var consulta = new SqlCommand();
        consulta.CommandText = "listarTipos";
        consulta.CommandType = CommandType.StoredProcedure;
        consulta.Connection = conexion;
        try
        {
            conexion.Open();
            var resultado = consulta.ExecuteReader();
            if (resultado.HasRows)
            {
                while (resultado.Read())
                {
                    escuelas.Add(resultado[0].ToString());
                }
                conexion.Dispose();
            }
            else
            {
                conexion.Dispose();
            }
            return escuelas;
        }
        catch (SystemException e)
        {
            Console.WriteLine("Error al conectar con servidor");
            return escuelas;
        }
    }
};
}
public ArrayList listaZonasEstadosMunicipios(string idZona,string idEstado){
    ArrayList escuelas = new ArrayList();
    using (var conexion = new
SqlConnection(System.Web.Configuration.WebConfigurationManager.ConnectionStrings["SIGPRO"]
].ConnectionString))

```



```

    };
}
@section scripts
{
    @Scripts.Render("~/Scripts/jquery-2.1.4.min.js")
    @Scripts.Render("~/bundles/bootstrapjs")
    @Scripts.Render("~/bundles/generaljs")
    @Scripts.Render("~/bundles/consultaEscuelasjs")
}
@section css
{
}
<div id="contenedor_todo" class="container-fluid">
    <!--h4>@Session["ipCliente"]</!--h4-->
    <h3>Escuelas</h3>
    <hr />

    @try
    {
        if (HttpContext.Current.Session["mensaje"] != null)
        {
            Mensajes mensaje = HttpContext.Current.Session["mensaje"] as Mensajes;
            if (mensaje.estado)
            {
                <div class="alert alert-success" id="mensaje"><strong>Éxito!
</strong>@mensaje.mensaje</div>
            }
            else
            {
                <div class="alert alert-danger" id="mensaje"><strong>Error!
</strong>@mensaje.mensaje</div>
            }
            HttpContext.Current.Session["mensaje"] = null;
        }
    }
    catch (Exception e)
    {
    }
    <div class="panel-group">
        <div class="panel panel-default">
            <div class="panel-heading">
                <h4 class="panel-title">
                    <a data-toggle="collapse" data-parent="#accordion"
href="#collapse1">Buscar escuela</a>
                </h4>
            </div>
            <div id="collapse1" class="panel-collapse collapse in">
                <div class="panel-body">
                    <div class="row">
                        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Escuela:</div>
                        <input id="nomEscuela" type="text" class="col-xs-12 col-sm-4 col-
sm-offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0">
                        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Estado:</div>
                        <input id="nomEstado" type="text" class="col-xs-12 col-sm-4 col-
sm-offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0" list="Estados">
                        <datalist id="Estados"></datalist>
                        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Del. /Mun.:</div>
                        <input id="nomMunicipio" type="text" class="col-xs-12 col-sm-4
col-sm-offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0" list="Municipios">
                        <datalist id="Municipios"></datalist>
                        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Colonia:</div>
                        <input id="nomColonia" type="text" class="col-xs-12 col-sm-4 col-
sm-offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0">

```

```

        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Direccion:</div>
        <input id="nomDireccion" type="text" class="col-xs-12 col-sm-4
col-sm-offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0">
        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Nivel:</div>
        <input id="nomNivel" type="text" class="col-xs-12 col-sm-4 col-
sm-offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0" list="Niveles">
        <datalist id="Niveles"></datalist>
        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Tipo:</div>
        <input id="nomTipo" type="text" class="col-xs-12 col-sm-4 col-sm-
offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0" list="Tipos">
        <datalist id="Tipos"></datalist>
        <div class="col-xs-12 col-sm-4 col-sm-offset-4 col-md-3 col-md-
offset-4 col-lg-2 col-lg-offset-5">Número de alumnos:</div>
        <input id="numAlumnos" type="text" class="col-xs-12 col-sm-4 col-
sm-offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-0">
        <!--input class="campo col-xs-6 col-xs-offset-3 col-sm-4 col-sm-
offset-4 col-md-3 col-md-offset-4 col-lg-2 col-lg-offset-5" type="submit" value="Buscar"
/-->
    </div>
</div>
</div>
</div>
<div class="panel panel-default">
    <div class="panel-heading">
        <h4 class="panel-title">
            <a id="Escuelas" data-toggle="collapse" data-parent="#accordion"
href="#collapse2">Escuelas</a>
        </h4>
    </div>
    <div id="collapse_2" class="panel-collapse collapse in">
        <div id="listadoEscuelas" class="panel-body">
            @f
                Helpers.Roles rol = new Helpers.Roles();
                int cont = 0;
                string tipo;
            }

            <div class="row encabezado hidden-xs hidden-sm hidden-md">
                <div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">Escuela</div>
                <div class="col-xs-12 col-sm-2 col-md-1 col-lg-1">Clave</div>
                <div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">Dirección</div>
                <div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">Colonia</div>
                <div class="col-xs-12 col-sm-4 col-md-3 col-lg-2">Municipio</div>
                <div class="col-xs-12 col-sm-4 col-md-3 col-lg-1">Estado</div>
                <div class="col-xs-12 col-sm-2 col-md-2 col-lg-1">Nivel</div>
                <div class="col-xs-12 col-sm-2 col-md-1 col-lg-1">Número de
alumnos</div>
            </div>

            @foreach (var reg in (@ViewData["escuelas"] as ArrayList))
            {
                int contador = 0;
                string id = null;
                if (cont % 2 == 0)
                {
                    tipo = "par";
                }
                else
                {
                    tipo = "non";
                }
                <div class="row @tipo">
                    @foreach (var col in (reg as ArrayList))
                    {
                        if (contador == 0)

```

```

    {
        id = (col.ToString());
    }
    else if (contador == 1)
    {
        <div class="cnon col-lg-2">
            <div class="encabezado hidden-lg col-xs-
5">Escuela</div>
            <div class="col-xs-7 col-lg-12">
                <div class="dropdown">
                    <a href="@id" class="dropdown-toggle" data-
toggle="dropdown">
                        <div class="col-xs-11">@(col)</div>
                        <span class="caret"></span>
                    </a>
                    <div class="dropdown-menu pull-right">
                        <div role="presentation"><a
role="menuitem"
href="@HttpContext.Current.Session["url"]/Escuelas/actualizaEscuela?IDEscuela=@id">Actual
izar escuela</a></div>
                            <div class="divider"></div>
                            @if
(rol.tienePrivilegios((HttpContext.Current.Session["Usuario"] as Datos).rol,
"escuela_borrar"))
                                {
                                    <div role="presentation"><a
role="menuitem"
href="@HttpContext.Current.Session["url"]/Escuelas/borraEscuela?IDEscuela=@id">Borrar
escuela</a></div>
                                        <div class="divider"></div>
                                    }
                                <div role="presentation"><a
role="menuitem"
href="@HttpContext.Current.Session["url"]/Contactos/creaContacto?IDEscuela=@id">Nuevo
contacto</a></div>
                                    <div class="divider"></div>
                                    <div role="presentation"><a
role="menuitem"
href="@HttpContext.Current.Session["url"]/Contactos/buscaContacto?IDEscuela=@id">Mostrar
Contactos</a></div>
                                    <div class="divider"></div>
                                    <div role="presentation"><a
role="menuitem"
href="@HttpContext.Current.Session["url"]/Pedidos/asignaPromocion?IDEscuela=@id">Asignar
promociones</a></div>
                                    @if
(rol.tienePrivilegios((HttpContext.Current.Session["Usuario"] as Datos).rol,
"usuario_asociar_escuelas"))
                                        {
                                            <div class="divider"></div>
                                            <div role="presentation"><a
role="menuitem"
href="@HttpContext.Current.Session["url"]/Usuarios/asignaEscuela?IDEscuela=@id">Asignar
representante</a></div>
                                                }
                                            </div>
                                        </div>
                                    </div>
                                <!--@Html.RouteLink((col.ToString() != "" ?
col.ToString() : "---"), new { action = "Oportunidad", controller = "Login", escuela =
(col.ToString() != "" ? col.ToString() : "---") })-->
                                </div>
                            }
                        else if (contador == 2)
                        {
                            <div class="cnon col-lg-1">

```


Manuales

Se cuenta con 6 manuales:

1. Promotor
2. Gerente
3. Asistente
4. Administración
5. Atención al Cliente
6. Móvil

Se tienen manuales para cada módulo en la siguiente URL, los cuales se pueden encontrar en el siguiente sitio de internet:

<http://sigpro.esfinge.mx/>

SigproMóvil

Es una aplicación móvil para dar información de campo a los gerentes de tienda, está diseñado para el sistema operativo Android versión 4.0 y posterior, usando las propiedades del teléfono como son:

Datos,
WIFI,
SQLite,
Cámara de video,
Almacenamiento interno

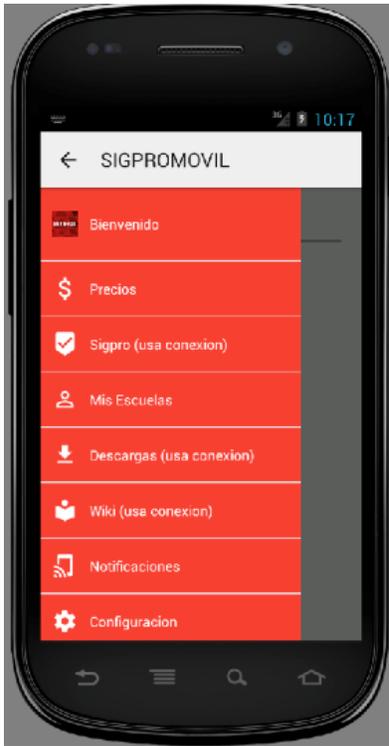
La comunicación con el dispositivo se realiza mediante un Web Service (Fig. 1.2) usando tecnología C# y REST. La Base de datos esta en SQL SERVER en el servidor y en el teléfono en SQLite ya que el dispositivo almacena los datos para usarse sin internet.

Contiene un total de 15 clases con 10 vistas diferentes, así como el uso de notificaciones con el API de Google, en el diagrama de la figura 3.0 se da una muestra de cómo está relacionado como una herramienta de datos para campo.



Fig. 3.0 Diagrama General de Operación (Sigpro Móvil)

Fig. 4.0 SigproMovil



En la Figura 4.0 podemos ver el menú que tiene la aplicación de Sigpro Movil.

1. Precios – Nos muestra los precios, títulos, stock de los libros activos de la editorial
2. SIGPRO – Versión Desktop
3. Mis escuelas – Contiene todas las escuelas asociadas al promotor para dar información de campo una vez estando en ellas
4. Descargas – Pueden descargar archivos importantes para cada grupo de promotores
5. Wiki – Contiene los procedimientos de la empresa para que los promotores puedan consultarlos
6. Notificaciones - Contiene un historial de cada una de las notificaciones que les ha llegado
7. Configuraciones – Es el panel donde pueden actualizar.

Flujo de operación

A continuación, podemos ver el flujo en las interacciones que se hace entre el usuario y el sistema.

PRESUPUESTO	
USUARIO	SISTEMA
Ingresa: sigpro.esfinge.mx	Abre sitio web
Ingresa: usuario y password	Abre Modulo de presupuesto
Teclea título de libro	Muestra todos los títulos disponibles
Asigna cantidades de títulos a representantes.	Inserta títulos a su presupuesto y se le manda notificación a su celular de cuantos títulos se le asignaron.
consultar el presupuesto asignado por representante	Busca en la base de datos y muestra los títulos insertados

PEDIDOS	
Ingresar: sigpro.esfinge.mx	Abre sitio web
Ingresar: usuario y password	Abre Modulo de pedidos
Busca RE-(numero de empleado) para ver si tiene presupuesto	Le devuelve los títulos y cantidades de títulos disponibles de su presupuesto
Clic en crear pedido	Muestra formulario
Ingresar: temporada, títulos, crea pedido	Descuenta los títulos del pedido de su saldo de promoción
	Se crea PDF y Excel del pedido con folio
Da clic para descargarlos y llevarlos a almacén	Le llega notificación al promotor de que su pedido se realizó.
PROMOCION	
Ingresar: sigpro.esfinge.mx	Abre sitio web
Ingresar: usuario y password	Abre Modulo de promoción
Modifica, contacto, escuela.	Se actualiza los registros de la escuela
Clic en una escuela asignar promoción	Muestra todos sus pedidos
Selecciona los libros y cantidades junto con el folio de la papeleta y asigna	Le quita sus promociones, y le genera un reporte de sus promociones asignadas.
MOVIL	
Descarga el APP de sigpro.esfinge.mx	Instala en celular
Abre el menú configuración clic actualizar precios	Importa la última lista de precios de SAP con el stock y vigentes
Escribe el número de representante Clic en registrar y en escuelas	Registra el celular para él envió de notificaciones, así como también le importa los datos de todas sus escuelas asignadas
Clic en el menú de escuelas	Mostrará todas sus escuelas
Clic en una escuela	Abrirá un menú con 3 opciones: Visita Promoción Adopción
Selecciona visita	Abre formulario para llenar
Selecciona Promoción	Abre formulario para llenar y fotografía
Selecciona Adopción	Abre formulario para llenar y fotografía
Abre menú Configuración	
Clic en actualizar Fotografía y Datos	Sube los datos almacenados en el celular para su validación en oficinas

GERENTE (Validación)	
Ingresar: sigpro.esfinge.mx	Abre sitio web
Ingresar: usuario y password	Abre Modulo de Gerente (Validación)
Clic en ver escuelas asignadas a sus representantes	Ve las escuela asignadas a los representantes
Clic en consultar presupuesto	Presupuesto por representante
Clic en buscar representante	Ve que municipios tiene asignado
Clic en mapa	Traza el recorrido del representante
Clic en visitas	Muestras las visitas que ha tenido el representante
Clic en enviar notificaciones	Ventana donde se pueden enviar notificaciones a los representantes
Clic en ventas	Total de ventas por representante

Modulos

Asignación De Presupuesto

La asignación de presupuesto se lleva a cabo por los gerentes de divisiones, tanto de educación básica como de media superior y presupuestan cierta cantidad de libros, los cuales sus asistentes lo distribuyen en los representantes autorizados.

Tienen un módulo para hacer esta actividad como en la figura 2.0, donde pueden consultar el presupuesto y asignarlo.

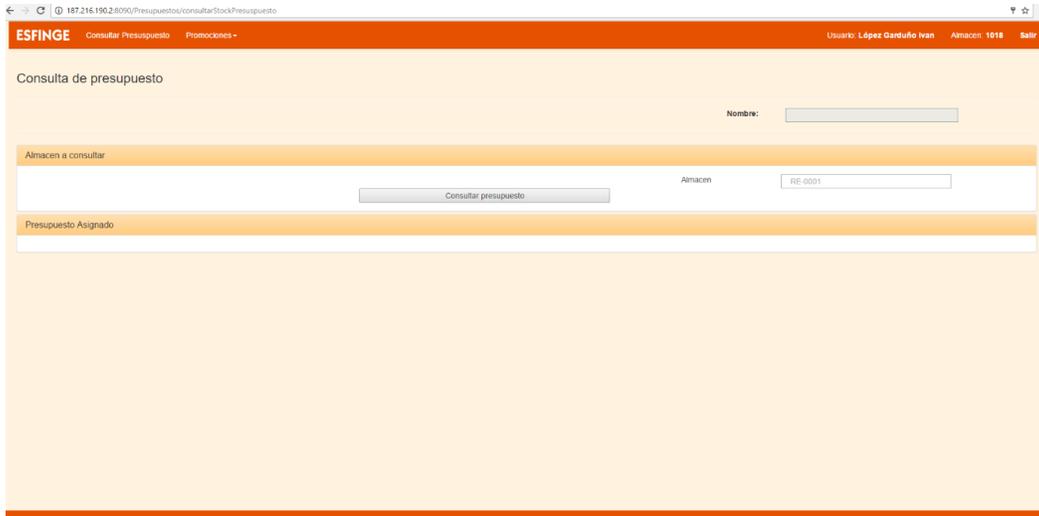


Fig. 2.0 Modulo de Presupuesto

Las acciones son que el asistente ingrese al módulo, ponga el número de representante al cual se le va a asignar la promoción, lo registra y le llega la notificación al representante de que ya tiene su presupuesto asignado con esto se proseguirá a que el representante pida sus libros que requiera.

Podremos verlo en un diagrama como el presentado en la figura 2.10, donde el actor va haciendo sus acciones paso a paso.

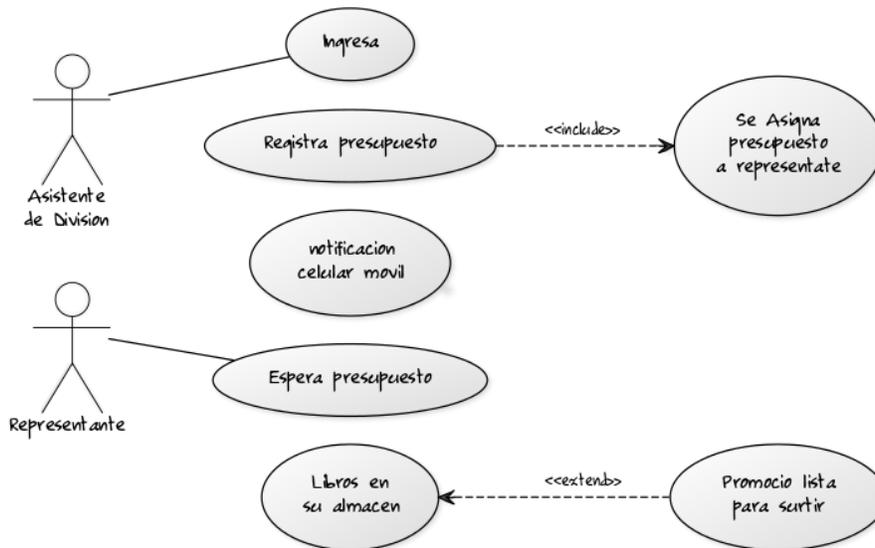


Fig. 2.10 Diagrama de Asignación de Presupuesto

Pedidos de promoción

Los pedidos son creados por servicio al cliente, ante las peticiones de los representantes de venta y con el presupuesto previamente autorizado por los asistentes de ventas.

El módulo de servicio al cliente es como el de la figura 2.20, donde se pone la temporada, el nombre del producto que pidió el representante y las cantidades solicitadas.

The screenshot shows the 'Registro de pedidos de promoción' interface. At the top, there's a navigation bar with 'ESFINGE', 'Consultar Presupuesto', 'Promociones', and user information: 'Usuario: López Garduño Ivan', 'Almacén: 1018', and 'Salir'. Below the title, there are input fields for 'Nombre:' and 'Almacén:'. A section titled 'Datos del representante' contains three dropdown menus: 'Almacén' (value: RE-0001), 'Temporada' (value: A), and 'Año-Temporada' (value: 2017). Below this is a 'Buscar libro' section with two sub-sections: 'Presupuesto del representante' and 'Sin Presupuesto'. Each sub-section has input fields for 'Código producto' and 'Título'. An 'Agregar' button is positioned between the two sub-sections. At the bottom of the form is a 'Crear pedido de promoción' button. The footer of the page reads 'Desarrollado por ©ESFINGE'.

Fig. 2.20 Módulo de Pedidos

Las acciones a realizar en la creación del pedido de promoción son que el promotor que previamente ya tiene su presupuesto, puede llamar al personal de servicio al cliente para hacer el pedido de los libros que desea, con lo cual:

- 1.- Ingresa a la plataforma
- 2.- Consulta que tenga el presupuesto
- 3.- Crea el pedido con los números de libros y títulos que requiere el promotor
- 4.- Le llega una notificación al promotor de que su pedido ya está en trámite con el número de folio
- 5.- El promotor pasa al almacén a recoger su pedido

En la figura 2.21 se puede notar como van las relaciones anteriormente mencionadas, donde el actor va haciendo sus acciones paso a paso y las flechas nos van diciendo que acción pasa a resultado

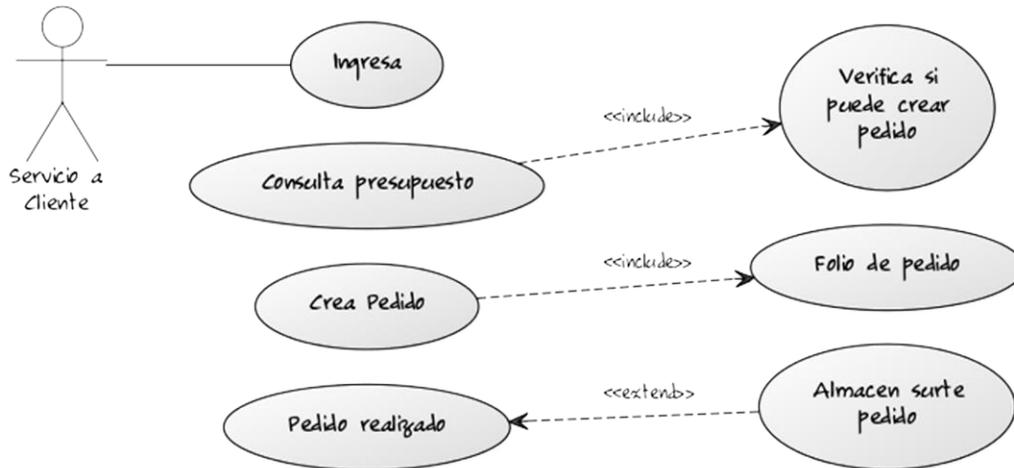


Fig. 2.21 Diagrama de Pedidos

Promociones Asignadas

Los libros que son asignados a los promotores deben justificarse, por lo cual tienen un módulo para poder asignar esos libros a los colegios que visitaron e ingresar el folio de la papeleta que les firmo el profesor.

En la figura 2.30 veremos cómo es el módulo de asignación de promoción, donde aparece la lista de escuelas que tiene asignadas el promotor y va a poder asignar sus libros.

Escuela	Clave	Dirección	Colonia	Municipio	Estado	Nivel	Número de alumnos
EDAYO ZINACANTEPEC	15EIC0008E	PASEO ADOLFO LOPEZ MATEOS KILOMETRO 4.5	SAN MIGUEL ZINACANTEPEC	ZINACANTEPEC	MEXICO	BÁSICA	4766
EDAYO METEPEC	15EIC0030G	PASEO DE LA ASUNCION LOTE 63	METEPEC	METEPEC	MEXICO	BÁSICA	3067
EDAYO IXTAPAN DE LA SAL	15EIC0015O	AVENIDA MORELOS	IXTAPAN DE LA SAL	IXTAPAN DE LA SAL	MEXICO	BÁSICA	2697
EDAYO TEJUPILCO	15EIC0001L	DONIGILIO CONOCIDO OJO DE AGUA	ZACATEPEC	TEJUPILCO	MEXICO	BÁSICA	2634
EDAYO VALLE DE BRAVO	15EIC0016N	AVENIDA CARLOS GOMEZ ALMAZAN	VALLE DE BRAVO	VALLE DE BRAVO	MEXICO	BÁSICA	2145
CENTRO DE CAPACITACION PARA EL TRABAJO INDUSTRIAL NO. 37	15DBT0037Y	IGNACIO LOPEZ RAYON	SAN MIGUEL ZINACANTEPEC	ZINACANTEPEC	MEXICO	BÁSICA	2030
EDAYO ATLACOMULCO	15EIC0019K	AVENIDA ISIDRO FABELA	ATLACOMULCO DE FABELA	ATLACOMULCO	MEXICO	BÁSICA	2008
UNIDAD EDAYO LERMA	15EIC0007F	AVENIDA PERALTA CARR AMOMOLULCO-ATARASQUILLO	LERMA DE VILLADA	LERMA	MEXICO	BÁSICA	1839
UNIDAD EDAYO TEMOAYA	15EIC0005H	CONOCIDO TLALTENANGO ARRIBA	TEMOAYA	TEMOAYA	MEXICO	BÁSICA	1792
LEONA VICARIO	15EPR0032N	CALZADA CHIMALTECATL	OCOYOACAC	OCOYOACAC	MEXICO	BÁSICA	1706
FRANCISCO LOPEZ RAYON	15EFR0187S	AVENIDA FRANCISCO LOPEZ RAYON	IXTLAHUACA DE RAYÓN	IXTLAHUACA	MEXICO	BÁSICA	1635
CENTRO DE CAPACITACION PARA EL TRABAJO INDUSTRIAL NO. 115	15DBT0005F	EX RANCHO LOS URIBE	EJIDO SANTA CRUZ ATZCAPOZALTONGO	TOLUCA	MEXICO	BÁSICA	1618
EDAYO IXTLAHUACA	15EIC0033D	LOTE NUM. 3 MANZANA 8 RANCHO	IXTLAHUACA DE RAYÓN	IXTLAHUACA	MEXICO	BÁSICA	1592

Fig. 2.30 Modulo de asignación de promoción

Las acciones del representante son que ingrese a su módulo, busque la escuela y asigne la escuela los libros que dejen así como el folio una vez realizado esto, ya justifica la promoción.

En la figura 2.31 muestra como son las acciones del representante que hace cuando asigna su promoción.

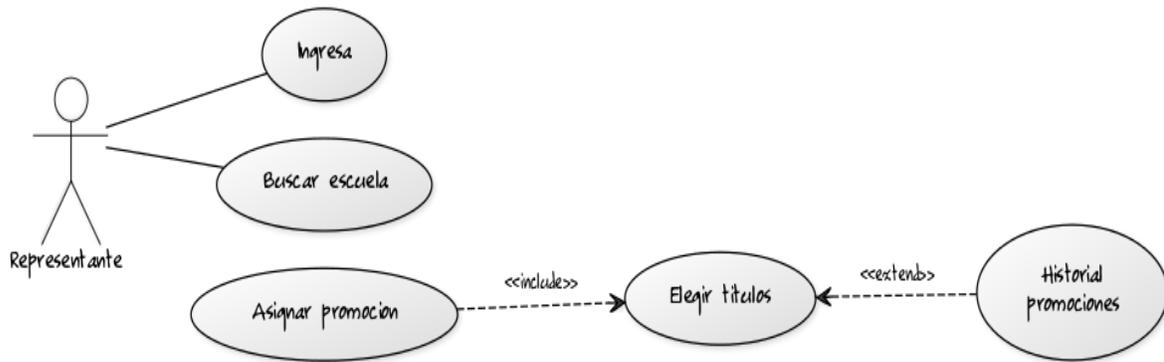


Fig. 2.31 Diagrama de Asignación de promoción.

Móvil

El mismo representante tiene una herramienta móvil con la cual puede hacer diferentes acciones, como son la visita, la promoción y la adopción, así como la lista de precios con stock actualizado en tiempo real, mismo que puede consultarse en modo offline sin necesidad de internet.

En el Diagrama 2.40 veremos las diferentes acciones sobre la aplicación celular.

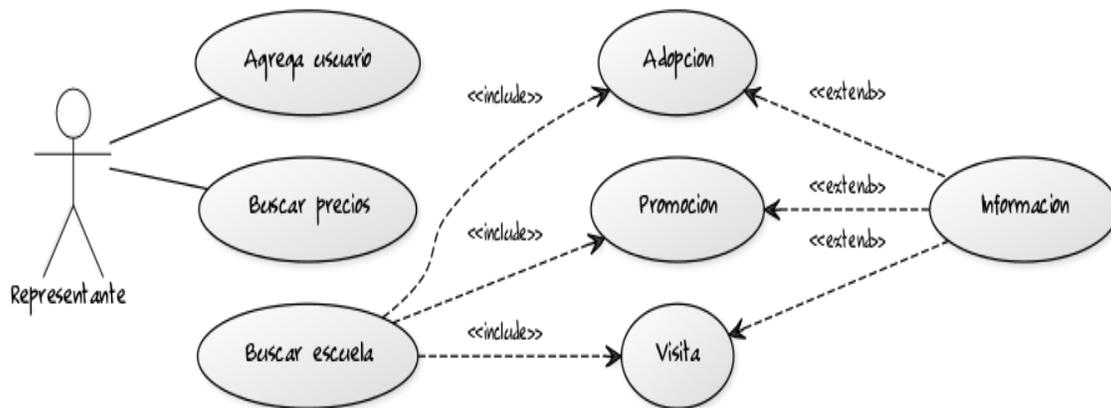


Fig. 2.40 Diagrama de Acciones de aplicación móvil.

Validación de Gerencia

El último módulo es el de validación donde los gerentes de ventas ven las promociones que asignaron sus representantes, también contiene reportes y tiene una vista como la mostrada en la figura 2.50, donde se aparecen las visitas que realizaron los representantes.

ESFINX Escuelas Contactos Promociones Usuarios Presupuestos Reportes Usuario: Grajales Pérez Angel Almacén: Salir

Buscar Comentarios de Seguimiento

Criterios de búsqueda

Temporada: 2017
 División: BASICA
 Nombre del libro:
 Nombre de la escuela:
 Nombre del promotor:

Buscar

Resultados

Promotor	Escuela	Temporada	Fecha	Próxima visita	Libro	Comentario
Tovar Monroy Juan Carlos	COLEGIO CULTURAL JUAN ALVAREZ	2017	31/01/2017 09:54:30 a.m.	02/01/2017 09:00:00 a.m.	Alfa , Letrados y el Arca	Presente catálogo y hay inter?ns en adoptar.
Tovar Monroy Juan Carlos	AMERICANA	2017	01/02/2017 03:38:48 p.m.	02/09/2017 02:00:00 p.m.	ortograf?a	Directora quiere revisar ortograf?a
Escamilla Velázquez Silvia	COLEGIO BELFORTT	2017	07/02/2017 11:58:47 a.m.	02/07/2017 12:30:00 p.m.	vive, ortograf?a, rutas, y textos.	Si le interesa revisar nuestro material.
Escamilla Velázquez Silvia	INSTITUTO METEPEC	2017	07/02/2017 12:04:35 p.m.	02/07/2017 09:39:00 a.m.	foundations, Criss Cross, Kells secundaria, y tools.	Solicito material de kells
Escamilla Velázquez Silvia	COLEGIO BICULTURAL SAN PATRICIO	2017	09/02/2017 10:57:41 a.m.	02/09/2017 10:58:00 a.m.		Van a continuar con lo mismo del año pasado, Trillas y Fern?ndez.
Escamilla Velázquez Silvia	OFIC NO 0246 "15 DE SEPTIEMBRE"	2017	23/02/2017 12:19:37 p.m.	07/03/2017 12:00:00 p.m.		visitar en julio
Robles Hernandez Braulio Ernesto	COLEGIO ALMA DE MEXICO	2017	27/02/2017 09:55:53 a.m.	03/06/2017 10:00:00 a.m.	cuadernos Alfa	llevan sistema ASC, pero le interesa mucho los cuadernos Alfa se dejaron segundo y sexto grado para
Robles Hernandez Braulio Ernesto	COLEGIO LA MONARCA	2017	27/02/2017 11:36:52 a.m.	03/06/2017 08:00:00 a.m.	literatura para primaria	No llevan materiales adicionales en espa?ol trabajan con los libros de SEP, ?nicamente llevan plan I
García Vivero María Guadalupe	JARDIN DE NIÑOS CERVANTINO	2017	28/02/2017 11:54:17 a.m.	03/07/2017 10:00:00 a.m.		a?n est?n revisando
García Vivero María Guadalupe	INSTITUTO PEDAGOGICO CERVANTINO	2017	28/02/2017 11:55:56 a.m.	03/07/2017 10:00:00 a.m.		a?n est?n revisando materiales
Robles Hernandez Braulio Ernesto	COLEGIO LOMAS HILL	2017	28/02/2017 08:25:07 a.m.	03/08/2017 09:00:00 a.m.	cuadernos Alfa y vive los n?meros	solo llevan material extra en matem?ticas con la serie fundamental de editorial castillo, y le inter
Robles Hernandez Braulio Ernesto	MANUEL ACOSTA	2017	28/02/2017 10:35:41 a.m.	05/02/2017 09:00:00 a.m.		El colegio lleva sistema de savia con plataforma de SM y tiene un contrato por 3 años y llevan por a

Fig. 2.50 Modulo de Gerentes

Las acciones que realiza el gerente para ver a sus usuarios son:

1. Ver escuelas asignadas
2. Presupuestos
3. Representante asignados
4. Reportes de sus promotores

En el diagrama 2.60 podremos ver sus acciones

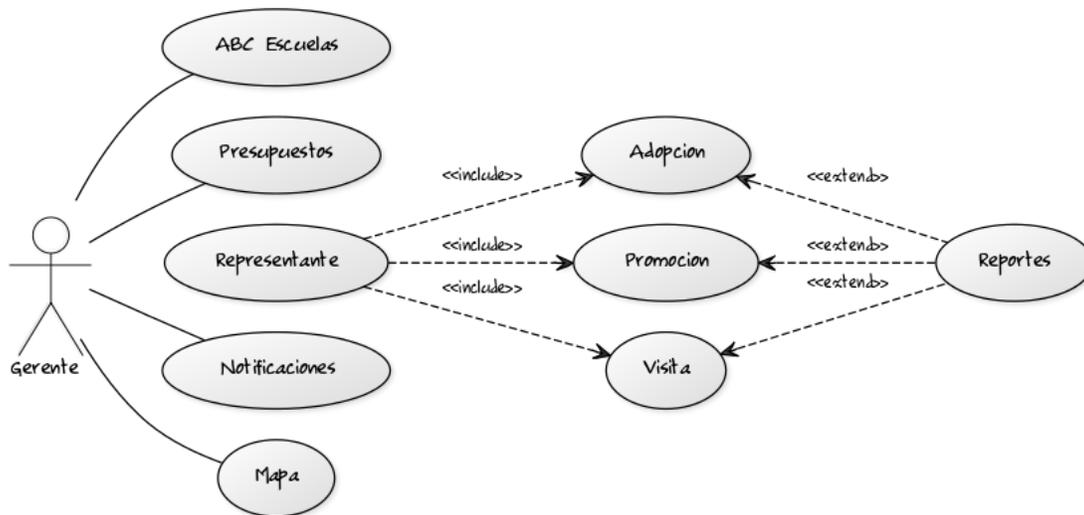


Fig. 2.60 Diagrama de Acciones de Gerentes

Pruebas

Plan

Los siguientes puntos son la estrategia para solventar las pruebas de la aplicación.

- 1.- Capacitación
- 2.- Muestreo con algunos usuarios
- 3.- Informes de Problemas
- 4.- Restauración y corrección
- 5.- Liberación a Producción

Pruebas de Carga

Las pruebas de carga son para identificar que tanto soporta el sitio las llamadas de los usuarios a la aplicación, pudiendo medir si podrá soportar el ingreso de una cantidad considerable de usuarios y no pierde fluidez.

Se midieron con el sitio <http://www.webpagetest.org/>

RUM First Paint	domInteractive	domContentLoaded	loadEvent
2.047s	1.941s	1.941s - 2.020s (0.079s)	2.026s - 2.034s (0.008s)

http://187.216.190.2:8090/	Keep-Alive 100%	GZip 29%	Compress Ing 100%	Progressive N/A	Cache Static 62%	CDN Detected 0%
1: 187.216.190.2:8090 - /	✓	✗				✗
2: 187.216.190.2:8090 - bootstrapcss	✓	✗			✓	✗
3: 187.216.190.2:8090 - css	✓	✗			✓	✗
4: 187.216.190.2:8090 - cssLogin	✓	✗			✓	✗
5: 187.216.190.2:80...jquery-2.1.4.min.js	✓	✓			✗	✗
6: 187.216.190.2:8090 - bootstrapjs	✓	✗			✓	✗
7: 187.216.190.2:8090 - generaljs	✓	✗			✓	✗
8: 187.216.190.2:8090 - Logo_esfinge.png	✓	✗	✓		✗	✗
9: 187.216.190.2:8090 - Favicon.ico	✓	✗			✗	✗
	Keep-Alive	GZip	Compress Ing	Progressive	Cache Static	CDN Detected

Dominio	Respuesta
187.216.190.2	8

Dominio	Bytes
187.216.190.2	229.506

Otros Proyectos

App Precios Esfinge

Aplicación móvil realizada en Android para los Ejecutivos de Venta de la Editorial, está hecha con tecnología .Net MVC en C# y Base de datos SQL Server, Montado sobre nuestro servidor dedicado Con Windows Server 2008 R2. Se realiza la comunicación del servidor al celular a través de un Web Service con tecnología REST donde la información es proporcionada por nuestro ERP de SAP Bussines One. El desarrollo fue 100% de un servidor (yo como desarrollador), la aplicación se distribuye en nuestro centro de descargas en el sitio web de la empresa.

Plataforma educativa para Primaria (SIMPLE-Conacyt)

Plataforma de contenido digital para niños de primaria, contiene actividades interactivas realizadas con el estándar SCORM 2004, alrededor de 560, así como actividades hechas con HTML5, Jquary y CSS3, alrededor de 550. Todas las actividades están integradas dentro de la plataforma MOODLE. Esta hospedado en un servidor dedicado Apache, la cual se tendrán diferentes instancias de acuerdo a la compra de las escuelas interesadas. Se realizó en trabajo colaborativo de equipo de desarrollo. Utilice metodología SCRUM para el proyecto de 6 meses, como líder de proyecto se llevó con éxito el proyecto a la entrega con las autoridades de Conacyt. La documentación del proyecto fue 100% realizada por un servidor, participación en 30 actividades, instalación y administración de Moodle, así como SCRUM master.

Apéndice A. Glosario de Términos

CentOs : (Community ENTERprise Operating System) es una bifurcación a nivel binario de la distribución Linux Red Hat Enterprise Linux RHEL, compilado por voluntarios a partir del código fuente publicado por Red Hat.

jQuery: es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX.

Xeon: es una familia de microprocesadores Intel para servidores PC y Macintosh. El primer procesador Xeon apareció en 1998 con el nombre Pentium II Xeon.

PHP: es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico

C#: (pronunciado si sharp en inglés) es un lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET

MySQL: es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo y una de las más populares en general junto a Oracle y Microsoft SQL Server.

Moodle: es una aplicación web de tipo Ambiente Educativo Virtual, un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conoce como Learning Content Management System (LCMS).

HTML5: es la quinta revisión importante del lenguaje básico de la World Wide Web.

SCRUM: es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado posible de un proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

RAID: el acrónimo RAID (del inglés Redundant Array of Independent Disks), traducido como conjunto redundante de discos Independientes, hace referencia a un sistema de almacenamiento de datos en tiempo real que utiliza múltiples unidades de almacenamiento de datos.

TB: es una unidad de almacenamiento de información cuyo símbolo es el TB y equivale a 1012 bytes. Mientras tanto, el prefijo “tera” proviene del griego que refiere monstruo o bestia.

BIBLIOGRAFIA

- Rick Anderson , Scott Hanselman, Microsoft Press, Intro to ASP.NET MVC 4 with Visual Studio
- Jess Chadwick, O REILLY,ASP.NET MVC
- Ross Mistry, SQL Server 2008 Management and Administration
- John Sharp, Microsoft Visual C# Step by Step
- Andrew Troelsen, Pro C# 5.0 and the .NET 4.5 Framework
- Belén Cruz Zapata, Android Studio Essentials
- Karli Watson, Professional C#
- Herbert Schildt, C# 4.0 The Complete Reference
- Project Management Institute,Guía de los fundamentos de gestión de proyectos
- Venkat Subramaniam, Practices Of An Agile Developer
- <http://stackoverflow.com/questions/21993758/asp-net-mvc-5-error-handling>
- <http://stackoverflow.com/questions/27720064/hax-kernel-module-is-not-installed>
- <http://stackoverflow.com/questions/41376309/what-exactly-is-a-rest-api>