



**EBRI CENTER**  
Encryption, BackUp and Recovery Information



GESTION DE TURNOS

GRUPO 1

# Manual Técnico



## Integrantes

Cristian Ricardo Portillo

Jaime Iván Muñoz Enríquez

Jorge Leonel Lam Pazos

Cristian Omar Aguirre

Jessica Marisol Castellanos Martínez

**E b r i C e n t e r**

**Guatemala Agosto 2014**

## Índice

Introducción.....pág. 1

Objetivo .....pág. 2

Contenido .....pág. 2

## Introducción

La finalidad de este manual técnico es proporcionar al lector la lógica con la que se ha desarrollado la aplicación, ya que esta varía para cada programador, por lo que se considera necesario ser documentada.

**Objetivo:** Proporcionar una guía al lector, del desarrollo de la interfaz y de la instalación de la aplicación gestión de usuarios.

## Contenido:

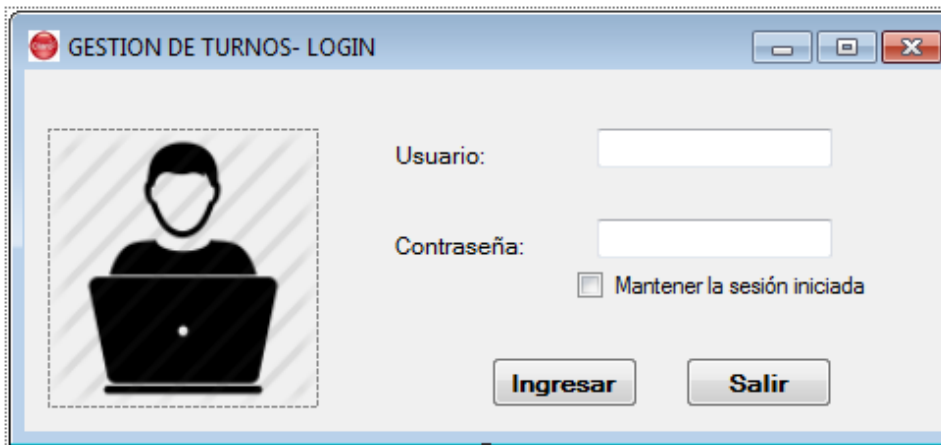
### Desarrollo de la interfaz

Desarrollo de páginas maestras e interfaces de usuarios:

Software utilizado: Visual C3 2010 Express Edition

Desde el menú de inicio de sesión

En el inicio de sesión contamos con nuestra ventana que está conectada a la base de datos por medio del servidor.



Este posee un pequeño string de conexión a la base de datos el cual verifica que el usuario ingresado sea correcto

```
private void button1_Click(object sender, EventArgs e)
{
    wfInicioSesion frmSesion = new wfInicioSesion();
    wfPantallaTurnos frmPantalla = new wfPantallaTurnos();

    try
    {
        String S_cadena = "Server=192.168.1.100;User ID= root; Database= bd_turnos; Password=GrupoSGT014;";
        SQL_Conexion.ConnectionString = S_cadena;
        SQL_Conexion.Open();
        SQL_cmd = new MySqlCommand("Select id_usuario from tabm_usuario WHERE usur_pass='" + txt_pass.Text + "'", SQL_Conexion);
        Object obj = SQL_cmd.ExecuteScalar();

        if (obj == null)
        {
            MessageBox.Show("Error", "El usuario y contraseña son invalidos validos, intente de nuevo");
            contadorInicioSesion += 1;
            txt_usuario.Text = "";
            txt_pass.Text = "";
            SQL_Conexion.Close();
        }
        else
        {
            IPHostEntry IP_host;
            string S_localIP = "";
        }
    }
}
```

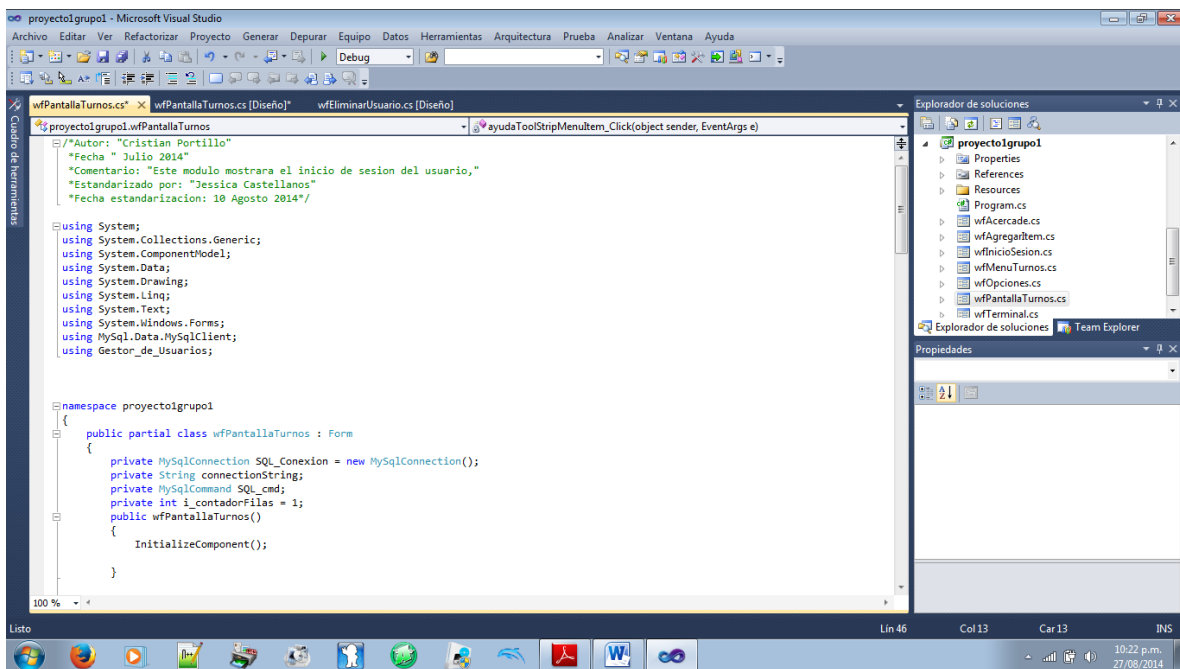
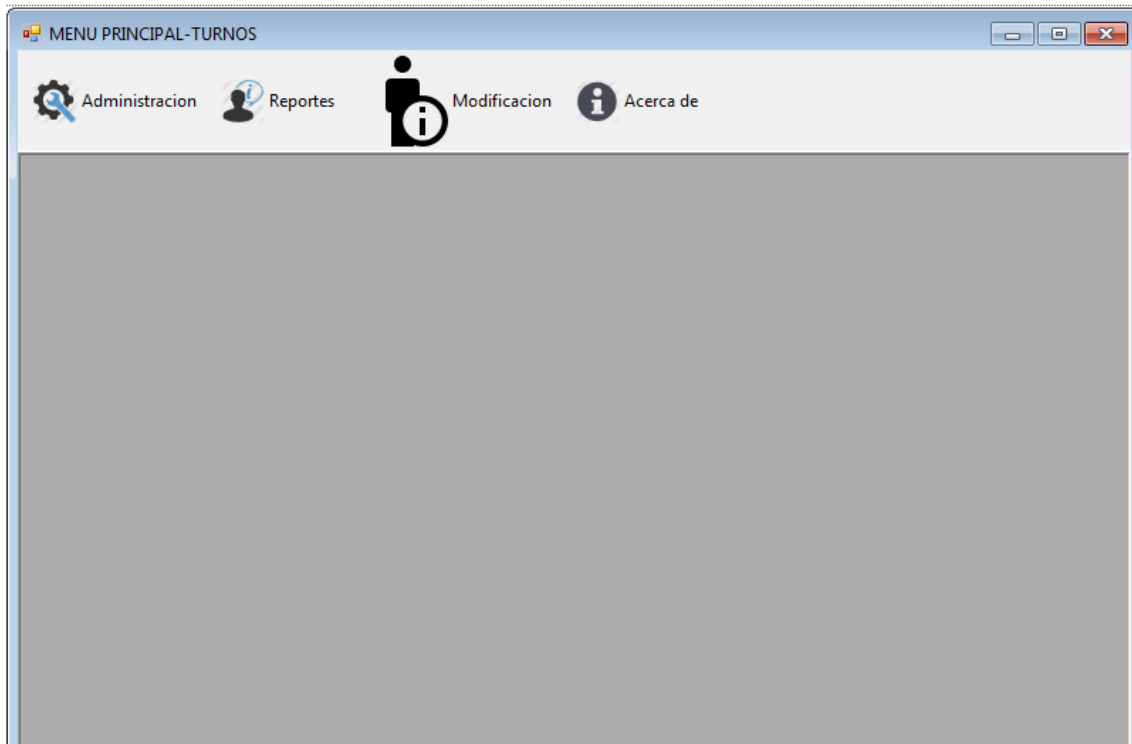
## El menú principal

El menú principal de la gestión de turnos es un formulario con MDI ya que es necesario contener los demás formularios para tener un mejor orden dentro de la aplicación encargado de cada área de turnos tenemos el área de administración que es en donde se agregan nuevos usuarios, se eliminan usuarios y se pueden modificar, en esta área de administración también se encuentra el buscar a usuarios ya ingresados y actualizar a los usuarios.

También se encuentra lo que es el área de reportes esta área genera reportes de los clientes que inician sesión, contiene reportes de los clientes que inician semanal y mensualmente, también los gráficos de que es lo que los clientes solicitan más.

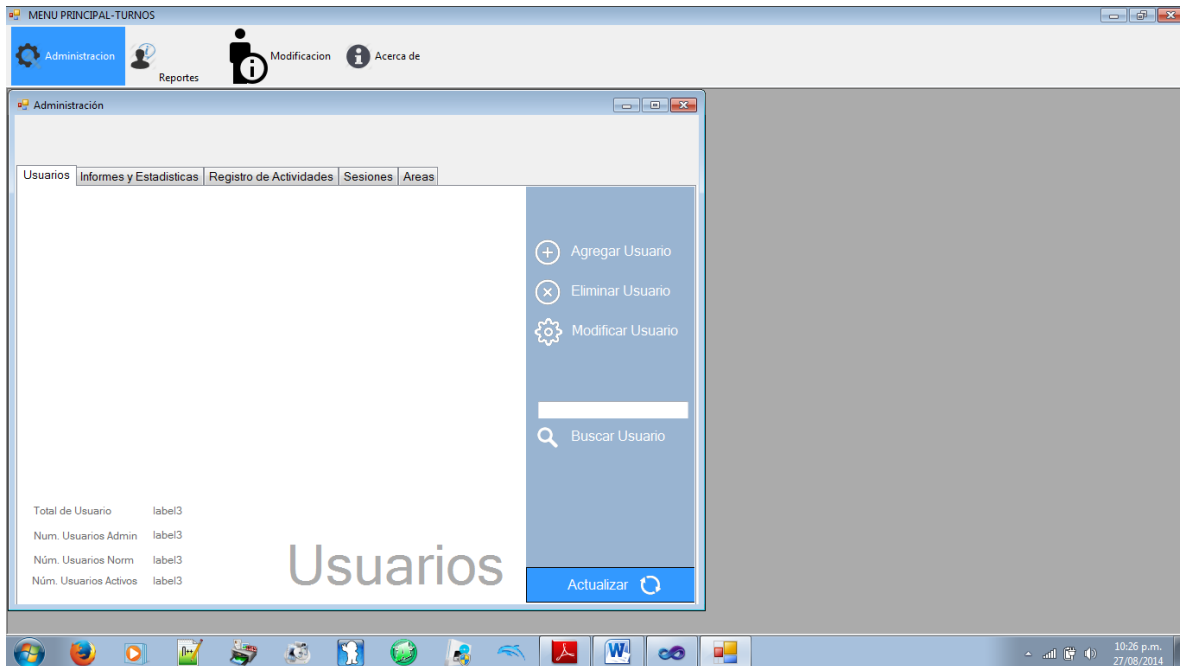
Tenemos también la pestaña de acerca de contiene la mención de cada uno de los integrantes del grupo.

Poseemos otra pestaña de modificaciones, esta pestaña se utiliza para ver qué clase de modificación a nivel de usuario se requiere hacer.



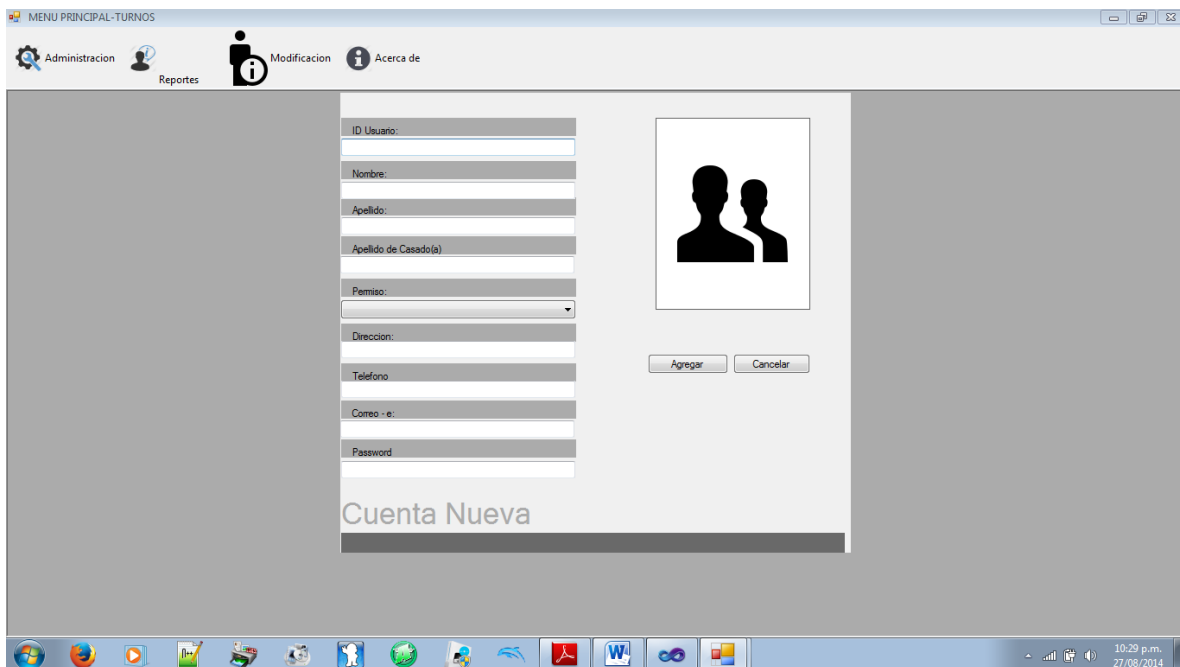
Como podemos ver el código de este formulario contiene simplemente llamadas a los demás formularios hijos y las referencias necesarias, cuenta con propiedad es como `IsMDIContainer` para poder contener a los formularios hijos.

Dentro de la pantalla del menú principal encontramos lo que es el administrador



Como podemos ver es otro formulario contenido dentro del MDI del menú principal, este nuevo menú posee las opciones de agregar, eliminar, modificar, buscar un usuario.

Estos servicios además contienen las estadísticas los registros de actividad las sesiones, y las áreas de nuestro sistema.



En la imagen anterior podemos ver lo que es agregar un nuevo usuario que esta contenido dentro del primer formulario que es el menú principal de turnos. Y así tenemos contenidas las demás opciones eliminar, modificar, actualizar, modificar.

```
private void ayudaToolStripMenuItem_Click(object sender, EventArgs e)
{
    wfAdministracion Childform = new wfAdministracion();
    Childform.MdiParent = this;
    Childform.Show();
}
```

En esta pantalla podemos ver como volvemos formulario padre al formulario de administración así los demás formularios sabrán que deben acoplarse a él como sus hijos.

```
private void btnAgregarUser_Click(object sender, EventArgs e)
{
    |

    wfAgregarUsuarios FrmNuevoUsuario = new wfAgregarUsuarios();
    FrmNuevoUsuario.Show();
    this.Hide();
}
```

Acá podemos observar cómo se llama al formulario para agregar un nuevo usuario a continuación vemos al formulario que llama:



Como vemos este formulario está hecho para insertar los datos del usuario nuevo.

Este formulario hace la conexión con la base de datos, establece la conexión con la tabla indicada y llena los campos que se le solicitan a continuación podemos ver el código que contiene esta ventanilla:





```

;CreacionPDF.Form1
+ button1_Click(object sender, EventArgs e)
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;

namespace CreacionPDF
{
    public partial class Form1 : Form
    {
        private PDF pdf;
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
        }

        private void button1_Click(object sender, EventArgs e)
        {
            pdf = new PDF();
            pdf.DocPDF("Turno VS-6", "Ticket VS-6");
        }
    }
}

```

Este código es el que genera el PDF.

En la siguiente ventanilla podemos observar la creación de una DLL para la creación del PDF la cual indica en que área se encontrara el PDF

```

CreacionPDF.cs Form1.cs Form1.cs [Diseño]
+ PDF()
using iTextSharp.text;
using System.IO;

namespace CreacionPDF
{
    class PDF
    {
        public PDF()
        {
        }

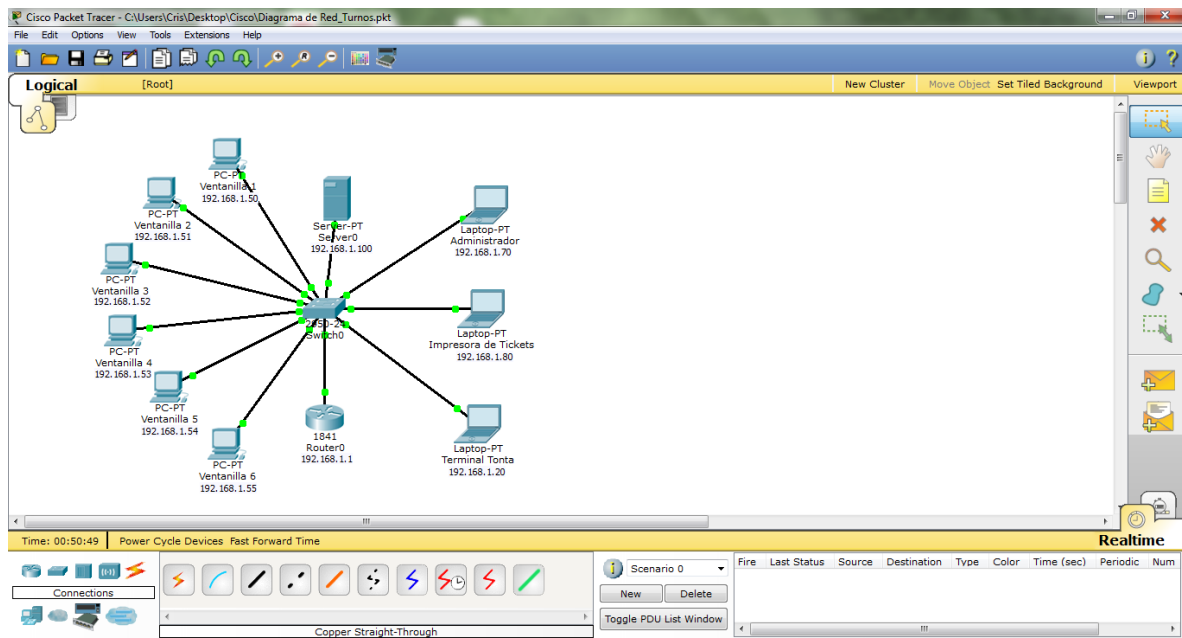
        public void DocPDF(String S_Cadena, String S_NombrePDF)
        {
            try
            {
                if (!Directory.Exists(@"C:/Users/Orochi/Desktop/Ticket/"))
                {
                    Directory.CreateDirectory(@"C:/Users/Orochi/Desktop/Ticket/");
                }
                Document documento = new Document();
                PdfWriter.GetInstance(documento, new FileStream(@"C:/Users/Orochi/Desktop/Ticket/" + S_NombrePDF + ".pdf", FileMode.OpenOrCreate));
                documento.Open();
                documento.Add(new Paragraph(S_Cadena));
                documento.Close();
            }
            catch (Exception)
            {
                System.Windows.Forms.MessageBox.Show("Error en la generacion", "Error", System.Windows.Forms.MessageBoxButtons.OK, System.Windows.Forms.MessageBoxIcon.Error);
            }
        }
    }
}

```

Las referencias utilizadas

Diagram

a de red



En el diagrama de red podemos observar la manera en que el servidor y la capa de datos fueron distribuidos.

Podemos observar que las computadoras fueron relacionadas para hacer pruebas, cada usuario puede iniciar sesión sin problemas.

## Creación del servidor

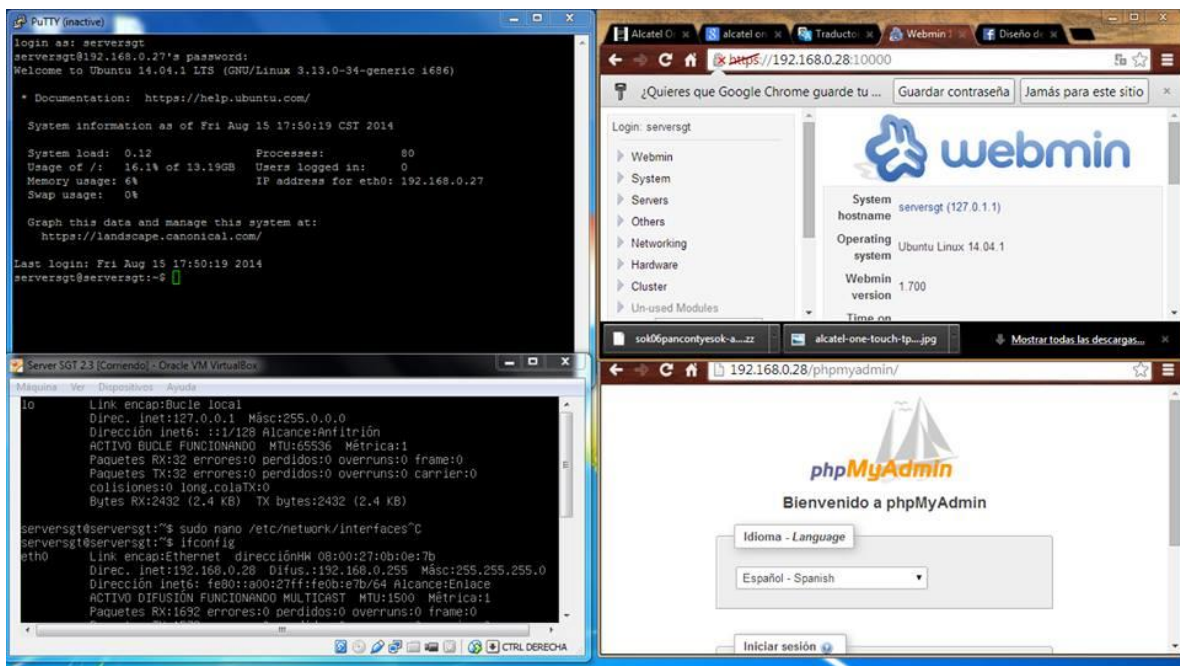
Nuestro servidor fue creado de la siguiente manera:

Versión ServerSGT 2.4.

- SO Ubuntu Server 14.04 LTS.
- Webmin 1.700
- SSH Server para conexión remota con el servidor por medio del cliente

Putty para Windows

- IP Estática
- LAMP server con Conexión remota



## Creación de la terminal tonta

Este módulo es el que se muestra al cliente en el cual el selecciona el servicio que desea y se genera su ticket para pasar, este módulo es uno de los más modificados ya que debía quedar perfecto para el cliente a continuación mostraremos la serie de mejoras por las que paso la terminal tonta:

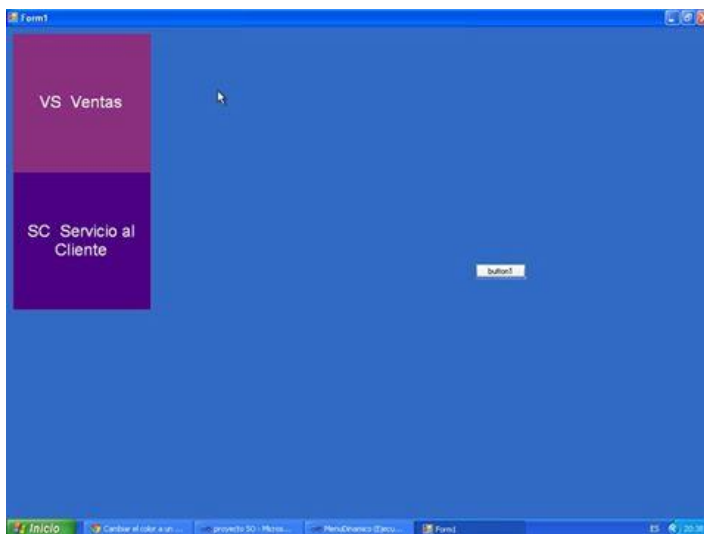
Primera versión:

Consta de Botones Virtuales Generados en Conjunto con la BD esto quiere decir que la Base de datos contiene la imagen del botón.



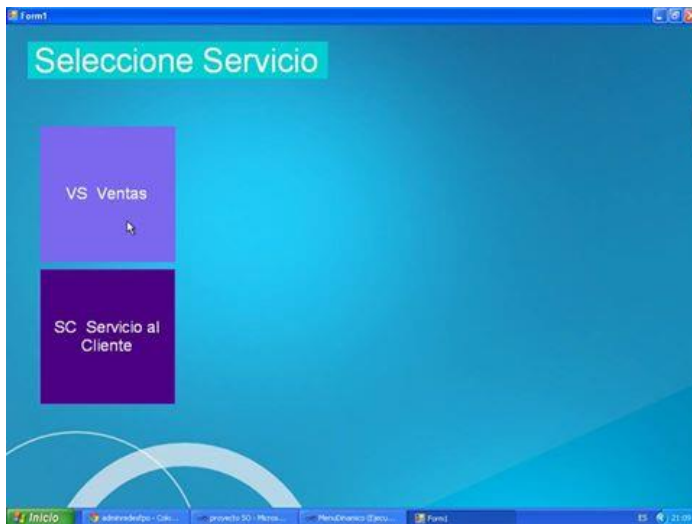
Segunda versión:

Mejora de la interfaz de botones



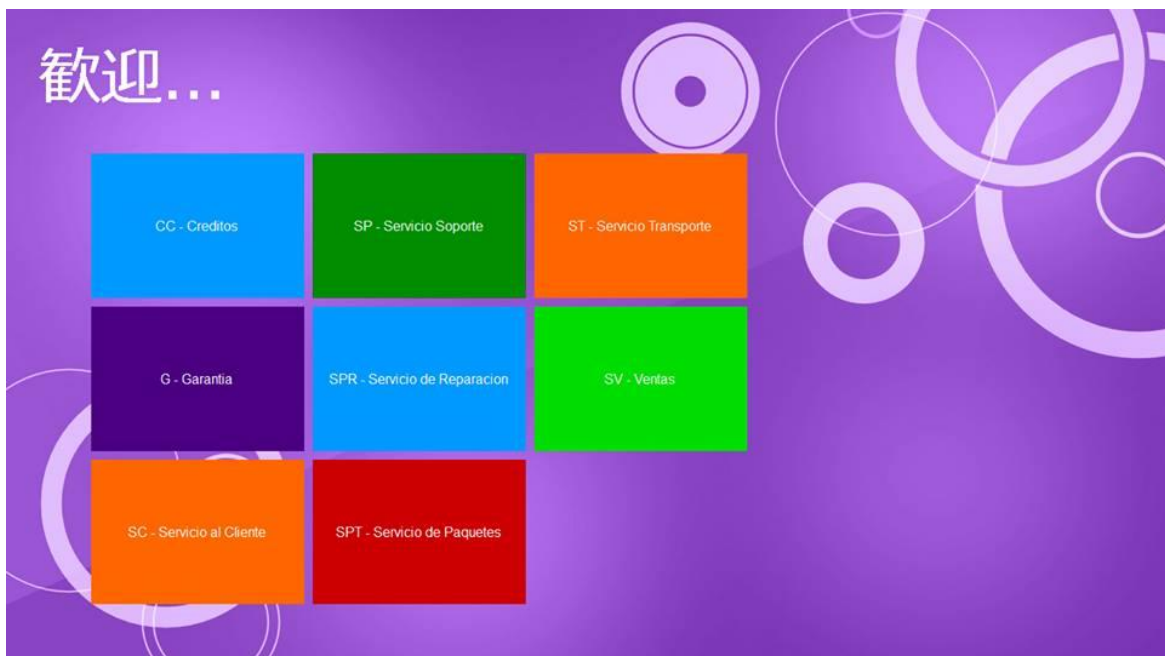
Tercera versión:

Combinación de colores



Cuarta versión:

Nuevo formato de clientes, para interfaz más amigable



Quinta versión:

Diferentes idiomas, mejoras por el cliente agregando fechas, hora y opción de imprimir su ticket que es la simulación por medio de PDF descrito anteriormente en este manual técnico.

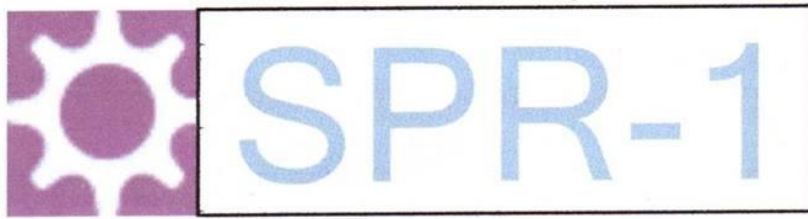


Versión final:

Versión finalizada con todas las áreas de gestión de turnos



La versión final lleva incluida la generación del ticket que es simulada por medio de la creación del PDF



Turno Impreso ☺

En esta ventana podemos ver los turnos que van en cola:



Es una simple ventana que muestra que turnos están en proceso y a que ventanilla le toca pasar.



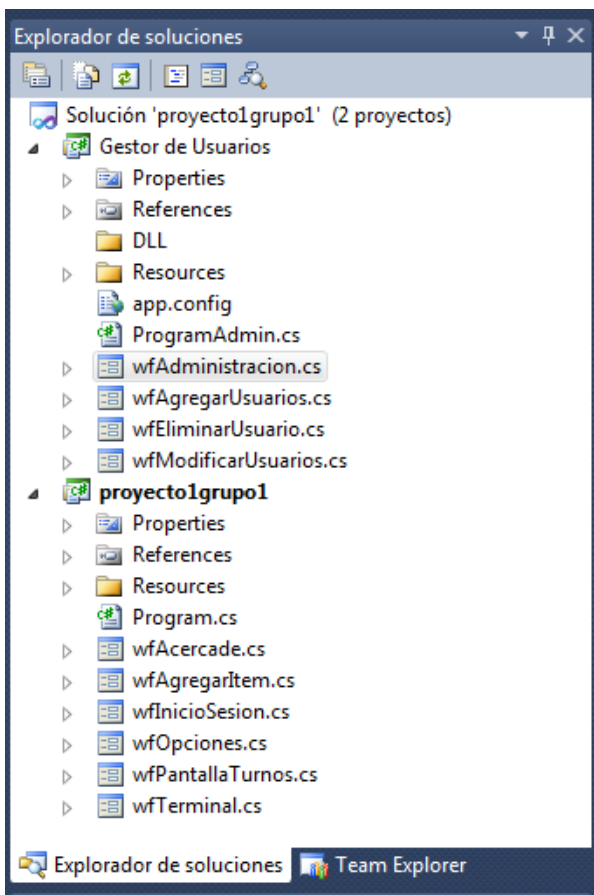
Después de realizar todas las ventanillas de todas las áreas fue necesaria la estandarización

```
oyecto1grupo1.wfPantallaTurnos
]/*Autor: "Cristian Portillo"
 *Fecha " Julio 2014"
 *Comentario: "Este modulo mostrara el inicio de sesion del usuario,"
 *Estandarizado por: "Jessica Castellanos"
 *Fecha estandarizacion: 10 Agosto 2014*/

]using System;
using System.Collections.Generic;
using System.ComponentModel;
```

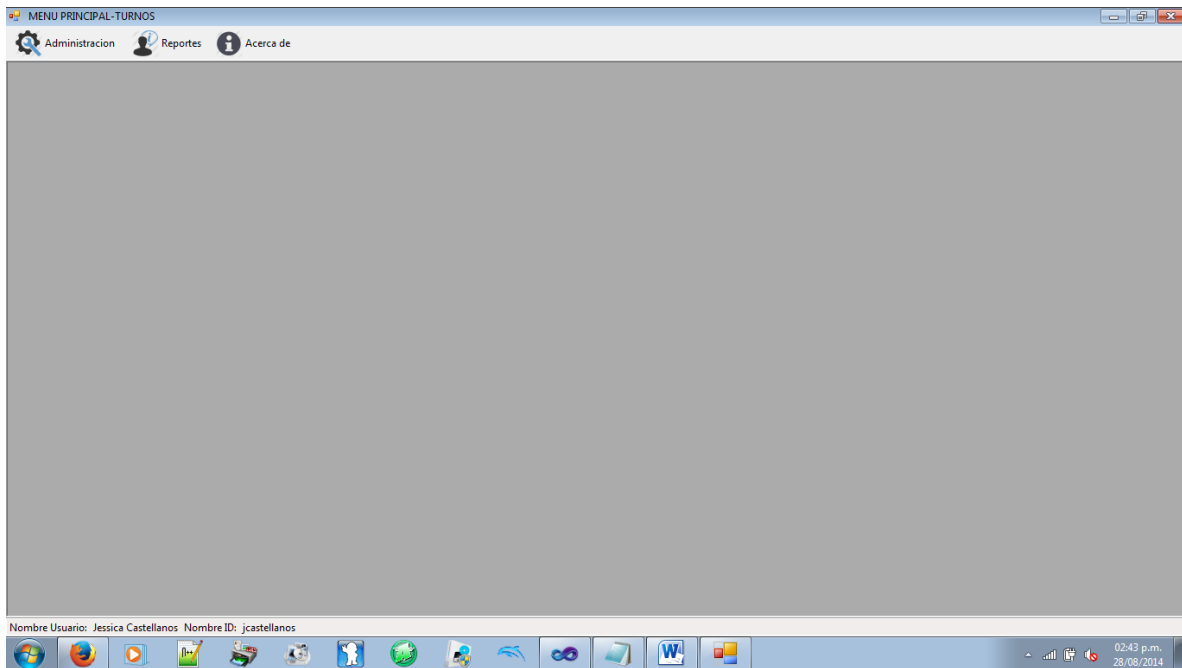
Que va desde colocarle un encabezado de autor a cada módulo y quien lo estandarizo.

También lleva estandarización en nombres de formularios, botones y demás herramientas que se utilizaron



Como podemos ver en esta ventana se encuentran los nombres de los formularios con su debida estandarización por ejemplo a los formularios se les antepone wf que quiere decir Windows form.

También se estandarizo lo que es el tamaño de pantalla de cada módulo. .

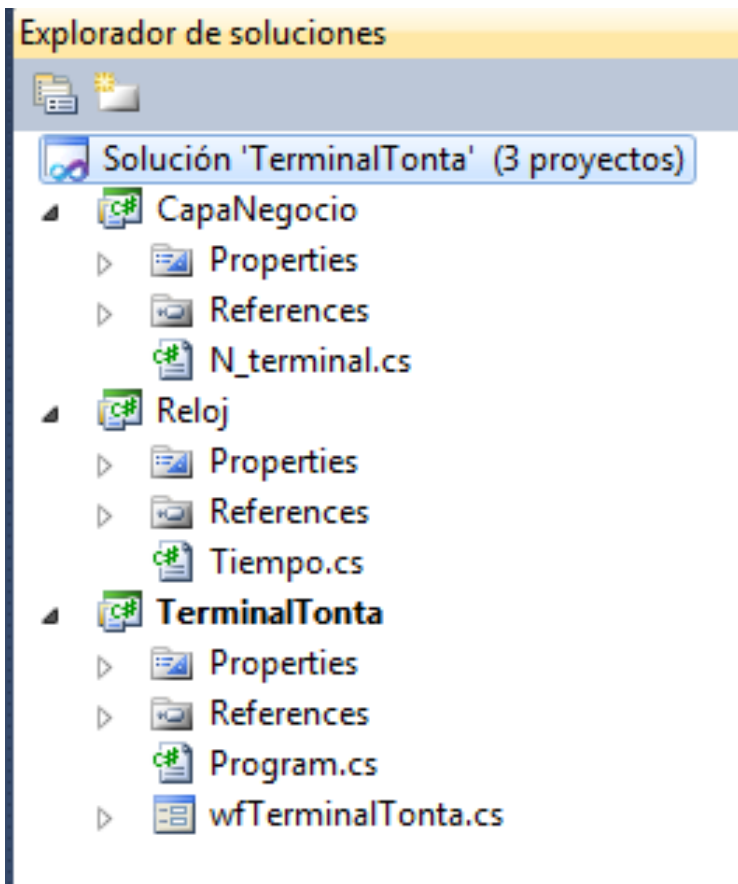


El color del fondo, el tipo de letra, variables etc.

Para el formulario MDI contenedor se utilizó el siguiente comando para que no se salga del contenedor:

```
|  
  
wfAdministracion frmadmin = new wfAdministracion();  
frmadmin.MdiParent = this;  
frmadmin.Show();
```

La terminal tonta está integrada por una programación por capas



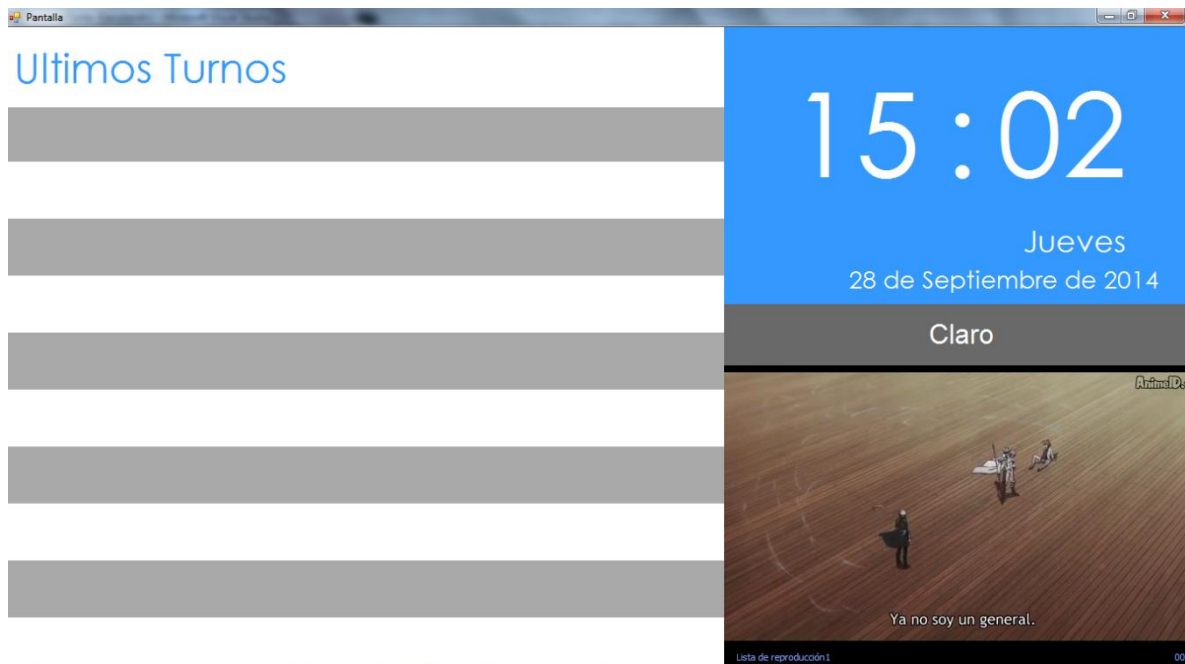
Como se puede ver tenemos la capa de negocio, la capa de datos y la capa de presentación.

Tenemos un contador para los meses y los días de la pantalla de la terminal tonta:

```
oj.Tiempo
sDiaSemana = String.Empty;
switch (DateTime.Now.DayOfWeek)
{
    case DayOfWeek.Monday: sDiaSemana = "Lunes"; break;
    case DayOfWeek.Tuesday: sDiaSemana = "Martes"; break;
    case DayOfWeek.Wednesday: sDiaSemana = "Miercoles"; break;
    case DayOfWeek.Thursday: sDiaSemana = "Jueves"; break;
    case DayOfWeek.Friday: sDiaSemana = "Viernes"; break;
    case DayOfWeek.Saturday: sDiaSemana = "Sabado"; break;
    case DayOfWeek.Sunday: sDiaSemana = "Domingo"; break;
}
return sDiaSemana;
}

protected String getMonth(int Mes)
{
    String sMes = "None";
    switch (Mes)
    {
        case 0: sMes = "Enero"; break;
        case 1: sMes = "Febrero"; break;
        case 2: sMes = "Marzo"; break;
        case 3: sMes = "Abril"; break;
        case 4: sMes = "Mayo"; break;
        case 5: sMes = "Junio"; break;
        case 6: sMes = "Julio"; break;
        case 7: sMes = "Agosto"; break;
        case 8: sMes = "Septiembre"; break;
        case 9: sMes = "Octubre"; break;
        case 10: sMes = "Noviembre"; break;
        case 11: sMes = "Diciembre"; break;
    }
    return sMes;
}
```

Vemos lo que es la terminal y los turnos en cola, con una pantalla de entretenimiento en lo que los clientes esperan su turno:



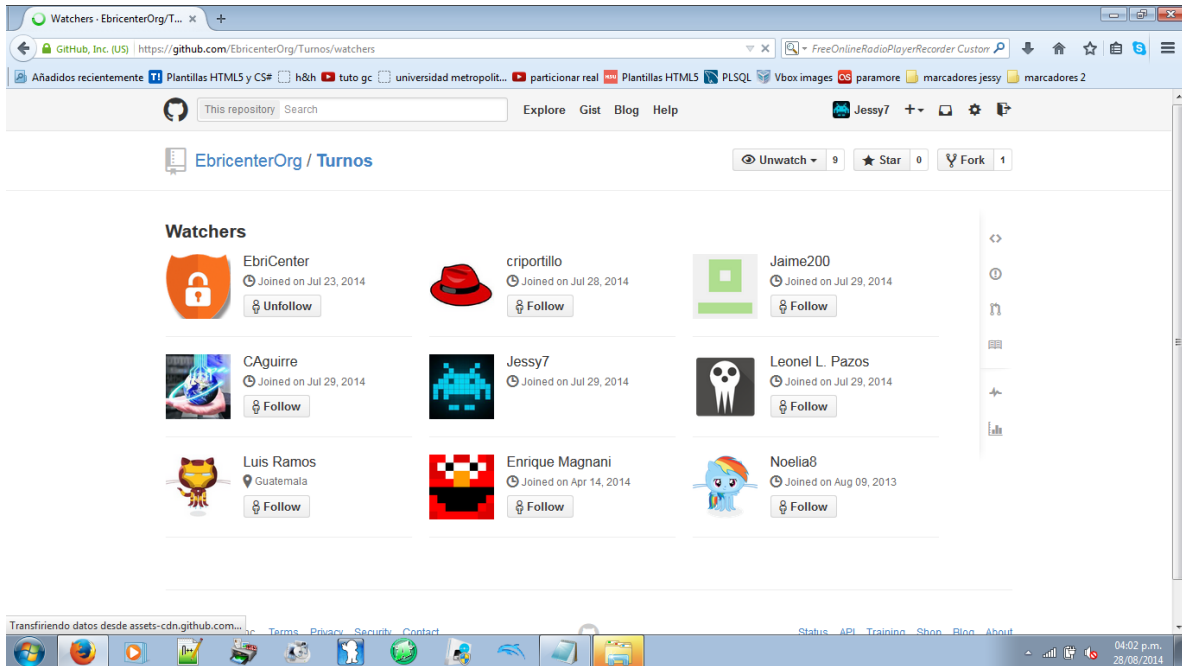
Nuestro sistema cuenta con un control sobre inicio de sesión de usuarios, donde guarda en la base de datos el id del usuario, el usuario, la fecha de inicio y la hora, la fecha y la hora del fin de la sesión, la ip, y el nombre de la pc.

id_sesion	usur_id	ses_fech_inic	ses_fech_fin	ses_hora_inic	ses_hora_fin	ses_dir_ip	ses_host
18	jmunoz	2014-08-27	0000-00-00	03:01:25	05:10:02	192.168.1.54	Orochi-PCOrochi
19	jmunoz	2014-08-27	0000-00-00	03:34:12	05:10:02	192.168.1.54	Orochi-PCOrochi
20	jmunoz	2014-08-27	0000-00-00	03:35:55	05:10:02	192.168.1.54	Orochi-PCOrochi
21	jmunoz	2014-08-27	0000-00-00	03:53:55	05:10:02	192.168.1.54	Orochi-PCOrochi
22	jmunoz	2014-08-27	0000-00-00	03:54:37	05:10:02	192.168.1.54	Orochi-PCOrochi
23	jmunoz	2014-08-27	0000-00-00	03:55:36	05:10:02	192.168.1.54	Orochi-PCOrochi
24	cportillo	2014-08-27	2014-08-27	04:42:38	05:11:53	192.168.1.54	Orochi-PCOrochi
25	jmunoz	2014-08-27	0000-00-00	04:45:08	05:10:02	192.168.1.54	Orochi-PCOrochi
26	jmunoz	2014-08-27	0000-00-00	04:47:05	05:10:02	192.168.1.54	Orochi-PCOrochi
28	jmunoz	2014-08-27	0000-00-00	05:07:40	05:10:02	192.168.1.54	Orochi-PCOrochi
29	jmunoz	2014-08-27	0000-00-00	05:09:59	05:10:02	192.168.1.54	Orochi-PCOrochi
30	cportillo	2014-08-27	2014-08-27	05:10:58	05:11:53	192.168.1.54	Orochi-PCOrochi
31	jcastellanos	2014-08-27	2014-08-27	05:16:54	08:46:17	192.168.1.51	JessyJessica
32	jcastellanos	2014-08-27	2014-08-27	05:17:57	08:46:17	192.168.1.51	JessyJessica
33	jcastellanos	2014-08-27	2014-08-27	05:19:50	08:46:17	192.168.1.51	JessyJessica
34	jcastellanos	2014-08-27	2014-08-27	08:00:26	08:46:17	192.168.1.51	JessyJessica
35	jcastellanos	2014-08-27	2014-08-27	08:20:51	08:46:17	192.168.1.51	JessyJessica
36	jcastellanos	2014-08-27	2014-08-27	08:30:01	08:46:17	192.168.1.51	JessyJessica
37	jcastellanos	2014-08-27	2014-08-27	08:31:26	08:46:17	192.168.1.51	JessyJessica
38	jcastellanos	2014-08-27	2014-08-27	08:32:17	08:46:17	192.168.1.51	JessyJessica
39	jcastellanos	2014-08-27	2014-08-27	08:33:35	08:46:17	192.168.1.51	JessyJessica
40	jcastellanos	2014-08-27	2014-08-27	08:46:12	08:46:17	192.168.1.51	JessyJessica

Aquí tenemos los usuarios que inician sesión para cualquier gestión

id_usuario	usur_nom	usur_ape	usur_ape_cas	usur_permiso	usur_dir	usur_tel	usur_mail	usur_pass
5	Cristian	Portillo	54654	0	2 Ciudad de Guatemala, Guatemala	41548785	cportillo@ebricenter.com	cportillo2014
	Jessica	Castellanos			2 Ciudad Quetzal	5798	jcastellanos@ebricenter.com	jcastellanos2
	Jorge	Lam			1 San Francisco, Zona 6, Mixco	3254	jlam@ebricenter.com	jlam2014
	Jaime	Munoz			2 Los Olivos, Zona 18	4158	jmunoz@ebricenter.com	jmunoz2014

Se cuenta con un repositorio para trabajar versiones del proyecto



Dentro de este repositorio se encuentran las versiones del proyecto las mejoras, las actualizaciones, las estandarizaciones, aquí se encuentran las minutas la documentación, los manuales etc.

