

Sistema de agendamientos de estudios médicos RIS (Sistema de información radiológica)

Salazar Martha ¹; Tagliacozzo Davide ²

¹ Universidad Tecnológica Israel-Departamento de Ciencias de la Ingeniería –Carrera de Sistemas, Quito, Ecuador, msalazar@uisrael.edu.ec

² Universidad Tecnológica Israel-Departamento de Ciencias de la Ingeniería –Carrera de Sistemas, Quito, Ecuador, dtagliacozzo@uisrael.edu.ec

Resumen: En primer lugar, RIS(sistema de información radiológica) es un sistema el cual se manejan el área de radiología o imagenología de un centro médico, hospital para poder agenda y direccionar la información de las citas médicas y de los estudios de imagen hacia los consultorios y equipos médicos para poder coordinar de mejor manera las citas y estudios a ser realizados dentro del hospital, de esta manera tanto los médicos, tecnólogos, usuarios en general del sistema podrán tener un control más detallado de lo que se realiza en el área de imagen.

Palabras clave: RIS, Agendamientos, Estudios médicos.

System of medical study agendas RIS (Radiological Information System)

Abstract: First of all, RIS (radiological information system) is a system which manages the radiology or imaging area of a medical center, hospital to schedule and direct the information of medical appointments and imaging studies to the offices and medical teams to better coordinate appointments and studies to be performed within the hospital, in this way both doctors, technologists, and users in general of the system will be able to have more detailed control of what is done in the image area .

Keywords: RIS, Schedules, studies.

1. INTRODUCCIÓN

Se utilizó Microsoft Visual Studio C# para realizar el desarrollo de la programación del sistema, ya que es el lenguaje impartido en la materia de Plataformas de Desarrollo, podemos tener más información sobre el Framework en la siguiente dirección para tener un poco más claro la utilización y las opciones que nos ofrece como desarrolladores.

<https://visualstudio.microsoft.com/es/vs/features/net-development/>

Tecnologías de Visual Studio

Desarrolle aplicaciones con .NET

[.NET](#) [Instalador completo](#)

 **Desarrollo con .NET (web y escritorio)**

Cree aplicaciones de escritorio de Windows y web multiplataforma con C#, F# o Visual Basic

Visual Studio con .NET 

Para realizar el diseño de la base de datos que utilizara el sistema, se realizó el modelo lógico, físico y el script de la base de datos en el programa Power Designer. ChartArea Class, <https://docs.microsoft.com/es->

1. Master Universitario en Ingeniería Matemática y Computación, msalazar@uisrael.edu.ec
2. Estudiante de la Carrera de Sistemas de Información, dtagliacozzo@uisrael.edu.ec

es/dotnet/api/system.web.ui.datavisualization.charting.chartarea?view=netframework-4.8 (2018)



Podemos obtener más información en el siguiente link

https://www.powerdesigner.biz/ES/powerdesigner/probar-powerdesigner-source_adw847a.html?gclid=CjwKCAjwx9_4BRAHEiwApAt0znN2ZMrxxwzKRq8XTLnJm332gYcNvIlrsc7yIO_NsOxkafS-LBPeQRoCa5EQAvD_BwE

Nota

El modelo tanto lógico como físico de la base de datos se puede realizar en diferentes tipos de programas, en este caso utilizamos PowerDesigner en su versión 16.5 ya que es de fácil utilización y para los propósitos deseados está muy acorde

El script que desarrollamos fue para el sistema gestor de base de datos Sql Server 2019, en el cual cargamos nuestro script y realizamos la manipulación de datos desde la aplicación desarrollada.



2. METODOLOGÍA

Una vez instalados los sistemas anteriormente mencionados, tenemos que verificar el servicio del SQL Server Management Studio se encuentre iniciado para no tener ningún tipo de problema y poder importar el script creado. Auto-Interval precision in MS Chart <https://stackoverflow.com/es/q/860504> (2010)

SMP de Espacios de almace...	Servicio hos...	Manual	Servicio de red
Solicitante de instantáneas ...	Coordina las...	Manual (dese...	Sistema local
SQL Server (SQLEXPRESS)	Proporciona...	En ejecución	Automático
SQL Server Browser	Proporciona...		Deshabilitado
SQL Server VSS Writer	Proporciona...	En ejecución	Automático
SysMain	Mantiene y ...	En ejecución	Automático
Tarjeta inteligente	Administra ...	En ejecución	Manual (dese...
TeamViewer	TeamViewer ...	En ejecución	Automático

Normalmente el servicio se inicia automáticamente, pero podemos verificar para no tener inconvenientes.

SQL Server <https://www.microsoft.com/es-es/sql-server/sql-server-downloads> (2011)

Cuando ya verificamos podemos ingresar a Sql Server Management Studio y logearnos con la autenticación de Windows sin problemas, después podemos en la parte de new Query colocar el script que hemos creado anteriormente en Power Designer en su versión 16.5 para poder ejecutarlo que nos cree la base de datos con las tablas que hemos creado.

```

Query1 - MS-SQLSERVER (2019)
ALTER DATABASE [SIRSI] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [SIRSI] SET DB_CHECKING OFF
GO
USE [SIRSI]
GO
***** Object: Table [dbo].[ESTUDIO] Script Date: 10/07/2020 22:01:19 *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[ESTUDIO]
(
    [ID_ESTUDIO] [numeric](8, 0) IDENTITY(1,1) NOT NULL,
    [ID_PACIENTE] [numeric](8, 0) NULL,
    [NOM_ESTUDIO] [varchar](50) NOT NULL,
    [TIPO_ESTUDIO] [varchar](50) NOT NULL,
    [HOR_ESTUDIO] [varchar](50) NOT NULL,
    [FEC_ESTUDIO] [varchar](50) NOT NULL,
    [DOC_ESTUDIO] [varchar](50) NOT NULL
) ON [PRIMARY]
GO
***** Object: Table [dbo].[PACIENTE] Script Date: 10/07/2020 22:01:19 *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[PACIENTE]
(
    [ID_PACIENTE] [numeric](8, 0) IDENTITY(1,1) NOT NULL,
    [ID_USUARIO] [numeric](8, 0) NULL,
    [NOM_PACIENTE] [varchar](50) NOT NULL,
    [APN_PACIENTE] [varchar](50) NOT NULL,
    [IMP_PACIENTE] [varchar](13) NOT NULL,
    [CID_PACIENTE] [varchar](13) NOT NULL
) ON [PRIMARY]
GO
***** Object: Table [dbo].[USUARIO] Script Date: 10/07/2020 22:01:19 *****
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[USUARIO]
(
    [ID_USUARIO] [numeric](8, 0) IDENTITY(1,1) NOT NULL,
    [NOM_USUARIO] [varchar](50) NOT NULL,
    [CUI_USUARIO] [varchar](50) NOT NULL,
    [FEI_USUARIO] [varchar](50) NOT NULL
) ON [PRIMARY]
GO
SET IDENTITY_INSERT [dbo].[ESTUDIO] ON
INSERT [dbo].[ESTUDIO] ([ID_ESTUDIO], [ID_PACIENTE], [NOM_ESTUDIO], [TIPO_ESTUDIO], [HOR_ESTUDIO], [FEC_ESTUDIO], [DOC_ESTUDIO]) VALUES (CAST(1 AS numeric(8, 0)), CAST(1 AS numeric(8, 0)), 'ESTUDIO', 'ESTUDIO', 'ESTUDIO', 'ESTUDIO', 'ESTUDIO')
INSERT [dbo].[ESTUDIO] ([ID_ESTUDIO], [ID_PACIENTE], [NOM_ESTUDIO], [TIPO_ESTUDIO], [HOR_ESTUDIO], [FEC_ESTUDIO], [DOC_ESTUDIO]) VALUES (CAST(2 AS numeric(8, 0)), CAST(1 AS numeric(8, 0)), 'ESTUDIO', 'ESTUDIO', 'ESTUDIO', 'ESTUDIO', 'ESTUDIO')
INSERT [dbo].[ESTUDIO] ([ID_ESTUDIO], [ID_PACIENTE], [NOM_ESTUDIO], [TIPO_ESTUDIO], [HOR_ESTUDIO], [FEC_ESTUDIO], [DOC_ESTUDIO]) VALUES (CAST(3 AS numeric(8, 0)), CAST(1 AS numeric(8, 0)), 'ESTUDIO', 'ESTUDIO', 'ESTUDIO', 'ESTUDIO', 'ESTUDIO')

```

Cuando ya hemos realizado la ejecución del script podemos verificar que tenemos las tablas creadas, con esto ya tendríamos todo listo para comenzar con

la programación del sistema de agendamientos RIS sistema de información radiológica.

3. RESULTADOS Y DISCUSIÓN

3.1. C# con MySQL.

Abrimos el proyecto que creamos y verificamos que la conexión con la base de datos es correcto para de esta manera poder tener acceso a toda la información desde el mismo.

```
SqlConnection conexion = new
SqlConnection("server=MSI\\SQLEXPRESS;
database=SISRIS ; integrated security =
true");
conexion.Open();
```

Verificando de esta manera que le nombre tanto del servidor así como de la base de datos es la correcta.

Después procedemos a verificar que esté funcionando el proceso de validación con el usuario y clave que hemos creado en la tabla USUARIO con efecto de pruebas.

```
if (AUX1 == 1 && AUX2 == 1)
{
    this.Hide();
    Form1 frm = new Form1();
    frm.Show();
    string message = "BIENVENIDO
    USUARIO --"+textBox1.Text+"-- ";
    string caption = "Ingreso
    Correcto";
    MessageBoxButtons buttons =
    MessageBoxButtons.OK;
    DialogResult result;
    result =
    MessageBox.Show(message, caption, buttons);
}
```

Podremos verificar que la conexión se ha realizado satisfactoriamente.

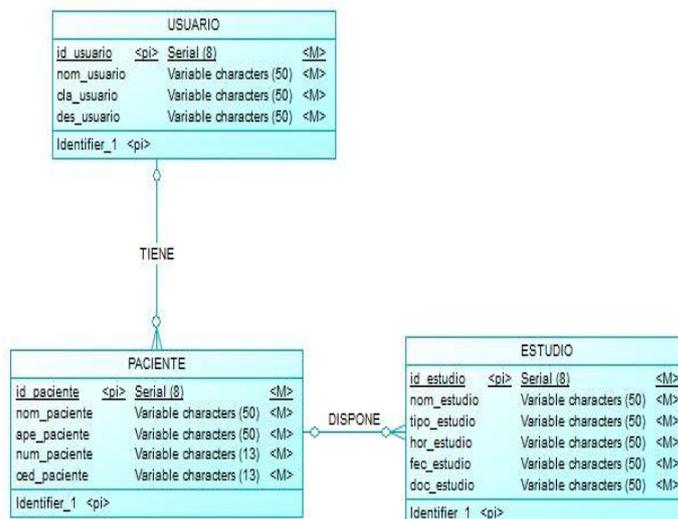
3.2. MODELO LOGICO BDD:

Podemos ver el modelo logico de la base de datos que hemos creado para el sistema RIS, el cual va a disponer de tres tablas.

USUARIO: es la tabla donde se van a registrar todos los usuarios que podrán ingresar al sistema por medio de una contraseña.

PACIENTE: En esta tabla se almacenara toda la información respectiva de los pacientes ingresados al sistema que tendrán la posibilidad de realizarse uno o varios estudios de imagenología dentro del hospital o institución donde se encuentre instalado el sistema.

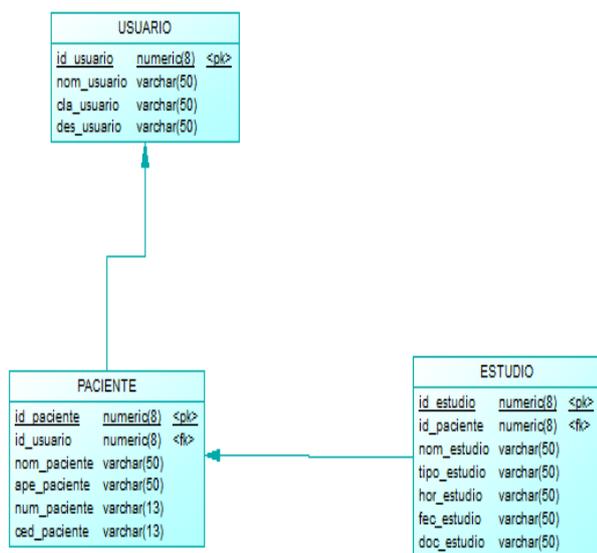
ESTUDIO: Aquí se almacenará la información referente a los estudios agendados para cada paciente en los días seleccionados, así como la hora a realizarse el estudio, también se va a almacenar



En base a las necesidades se dimensionara el tamaño y tipo de cada uno de los campos de las tablas de la base de datos, para disponer de tamaño suficiente para poder almacenar la información necesaria ya que muchos de los estudios realizados en imagenología tienen una longitud de hasta 200 caracteres, por lo cual con un varchar no vamos a tener ningún inconveniente.

3.3. MODELO FISICO BDD:

Después realizamos el modelo físico para poder generar posteriormente el script que vamos a ejecutar en el sistema gestor de base de datos.



3.4. SISTEMA EN C#:

Vamos a tener una pantalla donde el usuario puede logearse para tener acceso a las funciones del sistema.



Nota

En el caso de que la clave no sea correcta se desplegará un mensaje para notificar al usuario que ya sea el nombre de usuario o la clave son incorrectas.

El texto donde se ingresa la clave del usuario va a tener la propiedad PASSWORDCHAR * para que no se pueda ver la clave que se está ingresando.



Vamos a ver el código del proceso de la validación del usuario con la base de datos, donde compara tanto el usuario como la clave del mismo.

```
int AUX1 = 0, AUX2 = 0;
SqlConnection conexion = new SqlConnection("server=MSI\\SQLEXPRESS; database=ISISRIS; integrated security = true");
conexion.Open();
//SELECT PARA PODER BUSCAR EL PACIENTE INGRESADO AL SISTEMA POR MEDIO DEL CAMPO "COD_PAC" QUE ES UNICO PK
string BUSCAR = "select * from USUARIO where NOM_USUARIO ='" + textBox1.Text+"'and CLA_USUARIO ='" + textBox2.Text+"'";

SqlCommand comando = new SqlCommand(BUSCAR, conexion);

comando.ExecuteNonQuery();

SqlDataReader REGISTRO = comando.ExecuteReader();

//ASIGNAMOS LOS DATOS DE LA BASE A LOS TEXTOS EN CASO DE QUE EL REGISTRO SEA ENCONTRADO
while (REGISTRO.Read())
{
    textBox1.Text = REGISTRO["NOM_USUARIO"].ToString();
    AUX1 = 1;
    AUX2 = 1;
}

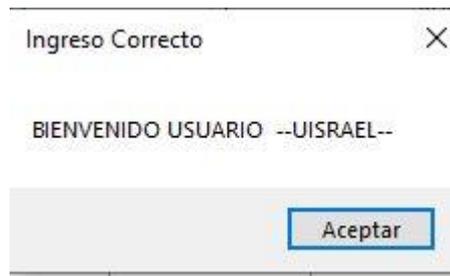
if (AUX1 == 1 && AUX2 == 1)
{
    this.Hide();

    Form1 frm = new Form1();

    frm.Show();

    string message = "BIENVENIDO USUARIO --" + textBox1.Text + "-- ";
    string caption = "Ingreso Correcto";
    MessageBoxButtons buttons = MessageBoxButtons.OK;
    DialogResult result;
    result = MessageBox.Show(message, caption, buttons);
}
else
{
    string message = "Clave o Usuario Incorrecto";
    string caption = "Error de Ingreso";
    MessageBoxButtons buttons = MessageBoxButtons.OK;
    DialogResult result;
    result = MessageBox.Show(message, caption, buttons);
    textBox1.Text = "";
    textBox2.Text = "";
}
}
```

Cuando el usuario haya ingresado satisfactoriamente al sistema, se desplegará un mensaje indicando el nombre del usuario que se logeo correctamente.

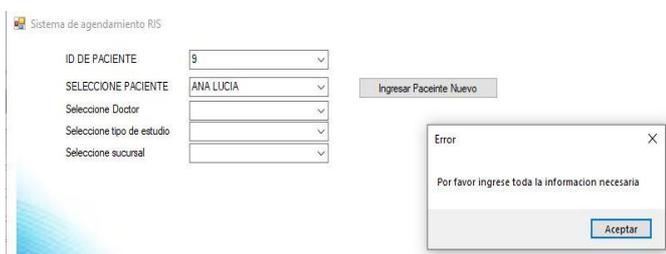


Ya dentro del sistema vamos a tener la pantalla principal donde encontraremos toda la información de los pacientes, las citas, ingresos nuevos y el gráfico estadístico.

En la siguiente pantalla vamos a ver la sección de información de pacientes e ingreso de pacientes nuevos, donde podremos seleccionar una nueva cita escogiendo el tipo de estudio que se va a realizar, el médico que lo va a atender y la sucursal.



En esta parte del sistema también dispone de las validaciones de ingreso de información, es decir la información es mandatoria y validara si falta algún dato a ingresar y desplegara un mensaje notificando que hay que completar toda la información para poder agendar la nueva cita médica.



Podemos ingresar un nuevo paciente en el caso de que no se encuentre en la base de datos.



Tenemos la sección de las citas médicas, en la cual vamos a poder visualizar las citas médicas que se encuentran agendadas.

ID_ESTUDIO	ID_PACIENTE	NOM_ESTUDIO	TIPO_ESTUDIO	HOR_ESTUDIO	FEC_ESTUDIO	DOC_ESTUDIO
1	1	ECOGRAFIA RE...	ECOGRAFIA DE ...	1000	14/07/2020	DRA TEJADA
2	1	ECOGRAFIA DE ...	ECOGRAFIA DE ...	1630	martes, 14 de jul...	DRA TEJADA
3	1	RAYOS X TORAX	RAYOS X TORAX	1000	domingo, 19 de j...	DRA TEJADA
4	9	RAYOS X TORAX	RAYOS X TORAX	1000	lunes, 13 de julio ...	DRA TEJADA
5	9	RAYOS X TORAX	RAYOS X TORAX	1000	martes, 14 de jul...	DRA TEJADA
6	3	RAYOS X TORAX	RAYOS X TORAX	1000	martes, 14 de jul...	DRA TEJADA
7	9	RAYOS X TORAX	RAYOS X TORAX	1000	miércoles, 15 de j...	DRA TEJADA
8	5	RAYOS X TORAX	RAYOS X TORAX	1000	miércoles, 15 de j...	DRA TEJADA
9	7	RAYOS X TORAX	RAYOS X TORAX	1000	jueves, 16 de jul...	DRA TEJADA
10	1	RAYOS X TORAX	RAYOS X TORAX	1000	jueves, 16 de jul...	DRA TEJADA
13	3	RAYOS X TORAX	RAYOS X TORAX	1000	viernes, 17 de jul...	DRA TEJADA
14	4	RAYOS X TORAX	RAYOS X TORAX	1000	viernes, 17 de jul...	DRA TEJADA
15	8	RAYOS X TORAX	RAYOS X TORAX	1000	viernes, 17 de jul...	DRA TEJADA
16	1	ECOGRAFIA OB...	ECOGRAFIA OB...	900	lunes, 13 de julio ...	DR DEL CARMEN
17	9	ECOGRAFIA PA...	ECOGRAFIA PA...	800	lunes, 13 de julio ...	DRA TEJADA
18	9	ECOGRAFIA PA...	ECOGRAFIA PA...	1130	lunes, 13 de julio ...	DRA TEJADA
19	9	RAYOS X PIE	RAYOS X PIE	1200	lunes, 13 de julio ...	DR NAVARRO

En esta parte del sistema vamos a poder visualizar el datagrid view donde va a estar toda la

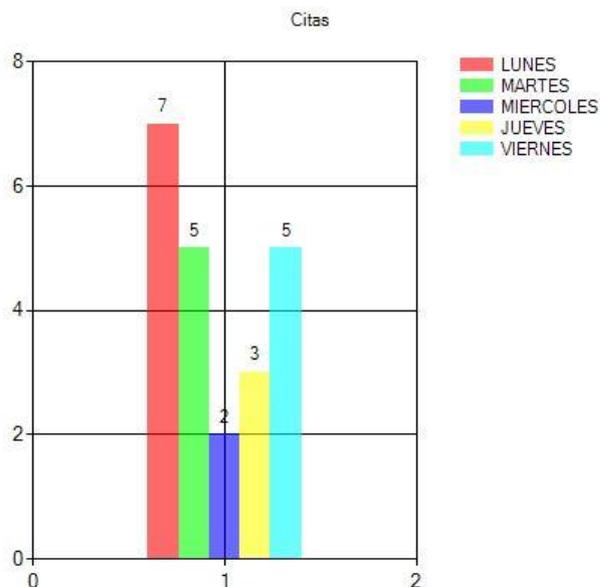
información referente a las citas médicas, en el caso de que se necesite modificar una cita médica lo podemos hacer directamente en el datagrid ya que se encuentra habilitada la opción UPDATE.

Podemos de igual manera ver el medico asignado asi como el horario de la cita, también podemos seleccionar ver solo un dia o el dia actual.

El sistema nos va a preguntar al momento que queremos salir, para verificar que en realidad deseamos salir del sistema y no se nos presionó el botón por error.



Dispone de un gráfico el cual se va a actualizar automáticamente después de realizar cada agendamiento en el sistema, el que nos indicara el número de citas que hay cada día de la semana.



Se realizó una comparación con los días de la semana y en un bucle se procedió a verificar a que

día de la semana pertenece cada cita para poder obtener el total de citas por día y así por darle los valores necesarios al sistema para poder realizar el grafico estadístico.

El medico puede realizar una consulta por día, para poder determinar cuál es el siguiente paciente a atender.

ID_ESTUDIO	ID_PACIENTE	NOM_ESTUDIO	TIPO_ESTUDIO	HOR_ESTUDIO	FEC_ESTUDIO	DOC_ESTUDIO
2	1	ECOGRAFIA DE...	ECOGRAFIA DE...	1630	martes, 14 de juli...	DRA TEJADA
5	9	RAYOS X TORAX	RAYOS X TORAX	1000	martes, 14 de juli...	DRA TEJADA
6	3	RAYOS X TORAX	RAYOS X TORAX	1000	martes, 14 de juli...	DRA TEJADA
20	1	ECOGRAFIA OB...	ECOGRAFIA OB...	1130	martes, 14 de juli...	DR NAVARRO
21	9	ECOGRAFIA OB...	ECOGRAFIA OB...	1130	martes, 14 de juli...	DR NAVARRO

A continuación vamos a ver el proceso de verificación por días de la semana para poder realizar el grafico y tomar los datos correspondientes.

```

conexionL.Open();
SqlDataReader registrol = comandoL.ExecuteReader();

while (registrol.Read())
{
    comboBox5.Items.Add(registrol["fec_estudio"].ToString());
}

conexionL.Close();
string newstring;
int lunes = 0, martes = 0, miercoles = 0, jueves = 0, viernes = 0;
for (int cont = 0; cont < comboBox5.Items.Count; cont++)
{
    newstring = comboBox5.Items[cont].ToString().Substring(0, 5);
    if (newstring == "lunes")
    {
        lunes = lunes + 1;
    }
}
textBox5.Text = lunes.ToString();

for (int cont = 0; cont < comboBox5.Items.Count; cont++)
{
    newstring = comboBox5.Items[cont].ToString().Substring(0, 6);
    if (newstring == "martes")
    {
        martes = martes + 1;
    }
}
textBox6.Text = martes.ToString();
    
```

Después de haber tomados los valores necesarios procedemos a realizar el grafico con la información obtenida.

```

////inicio grafico
string[] series = { "LUNES", "MARTES", "MIERCOLES", "JUEVES", "VIERNES" };
int[] puntos = { int.Parse(textBox5.Text), int.Parse(textBox6.Text), int.Parse(textBox7.Text), int.Parse(textBox8.Text), int.Parse(textBox9.Text) };
chart1.Palette = ChartColorPalette.SemiTransparent;

chart1.Titles.Add("Citas");

for (int i = 0; i < series.Length; i++)
{
    Series serie = chart1.Series.Add(series[i]);

    //cantidades
    serie.Label = puntos[i].ToString();
    serie.Points.Add(puntos[i]);
}
////fin grafico
    
```

4. CONCLUSIONES

- Para poder mantener una gestión adecuada en el flujo de trabajo dentro del área de imagenología de cualquier hospital o centro de imagen es indispensable tener un sistema RIS (Sistema de Información Radiológica), para de esta manera poder tener un mejor control de flujo de pacientes y que se puedan atender de una manera adecuada y rápida.
- Para poder determinar dentro del centro de imagen la afluencia de pacientes es necesario disponer de un gráfico estadístico y así poder evidenciar de una manera ágil y visual la cantidad de estudios agendados por día y poder determinar la carga de trabajo del personal.
- En la actualidad el personal administrativo de los hospitales no tienen muy claro el ahorro que puede representar por ellos un sistema RIS, ya que teniendo un mejor control y orden en la afluencia de pacientes, entrega de resultados y determinar la carga laboral por horarios, se puede atender de mejor manera a los pacientes y organizadamente atender a una mayor cantidad de personas en el mismo tiempo, lo cual represente una ganancia para el hospital ya que se puede tener mayor cantidad de ingresos en el mismo tiempo y una mejor atención a los pacientes.

REFERENCIAS

ChartArea Class, <https://docs.microsoft.com/es-es/dotnet/api/system.web.ui.datavisualization.charting.chartarea?view=netframework-4.8> (2017)

Auto-Interval precision in MS Chart
<https://stackoverflow.com/es/q/860504> (2010)

SQL Server, <https://www.microsoft.com/es-es/sql-server/sql-server-downloads> (2011)

POWERDESIGNER, https://www.powerdesigner.biz/ES/powerdesigner/probar-powerdesigner-source_adw847a.html?gclid=CjwKCAjwx9_4BRAHEiwApAt0zvJUoht2eBkUnAd0YiXPD_DmHot-Hxas8trB2X94a23JrhD0qyNjHhoCORsQAvD_BwE

DML SQL Server Connection, <http://csharp.netinformations.com/data-providers/csharp-sql-server-connection.htm> (2016)

DML SQL C#, <https://docs.microsoft.com/es-es/biztalk/adapters-and-accelerators/adapter-sql/insert-update-delete-and-select-on-tables-and-views-with-the-sql-adapter> (2013)