**Universidad Tecnológica Nacional**

**Ingeniería en Sistemas de Información**

**Cátedra: Tecnologías de Desarrollo de Software IDE**

**TRABAJO PRÁCTICO 2**

**Sistema de Autogestión Academia (SAG)**



**Informe**

**Año de Cursado: 2014**

**Grupo Nº: 07**

**Profesores:**

**- Porta, Ezequiel**

**- Joaquín, Andrés**

**Integrantes:**

**- Alvarez, Facundo (40149)**

**- Gentiletti, Matías (38579)**

# Índice

**Pág.**

**Presentación 2**

**Descripción 2**

**Funcionalidades 2**

**Descripción de las capas 5**

**Otras funcionalidades 7**

**Capturas de pantalla 13**

# Presentación

El presente **Trabajo Práctico (TP2)** tiene como finalidad generar un sistema simple en donde se apliquen los conocimientos aprendidos en las diferentes unidades temáticas que componen la asignatura.

Utilizando las herramientas brindadas por la cátedra y los conocimientos que fuimos adquiriendo en cada una de las clases, desarrollamos una aplicación que consiste en una versión simplificada de un **Sistema de Gestión Académica (SGA).**

La aplicación fue desarrollada utilizando la herramienta **Visual Studio 2010** en un marco de desarrollo **Framework .Net 4**, en el cual el código se compila en el **lenguaje C#.** La misma tiene acceso y respaldo de datos con una base de datos (**tp2\_net**) implementada en **MS SQL Server 2008 R2** y manejadas a través de las clases que ofrece el IDE (**ADO.net**)

# Descripción

La solución de nuestra aplicación está dividida en **varios proyectos (capas)** en donde cada una ofrece distintas funcionalidades para lograr un sistema más flexible y modular, permitiendo ser modificado fácilmente en cuanto a nuevos requisitos que se vayan presentando a lo largo del desarrollo.

# 

# Funcionalidades del Sistema

El sistema desarrollado permite registrar y administrar las actividades académicas de la universidad en torno a las entidades principales de ***Personas (Docentes y Alumnos)*** *y sus respectivos* ***Usuarios, Materias, Cursos, Comisiones, Especialidades y Planes***. Estas actividades se identifican como: ***Inscripciones*** (Alumno que se inscribe a un Curso) y ***Dictados*** (Profesores que “dictan” un Curso).

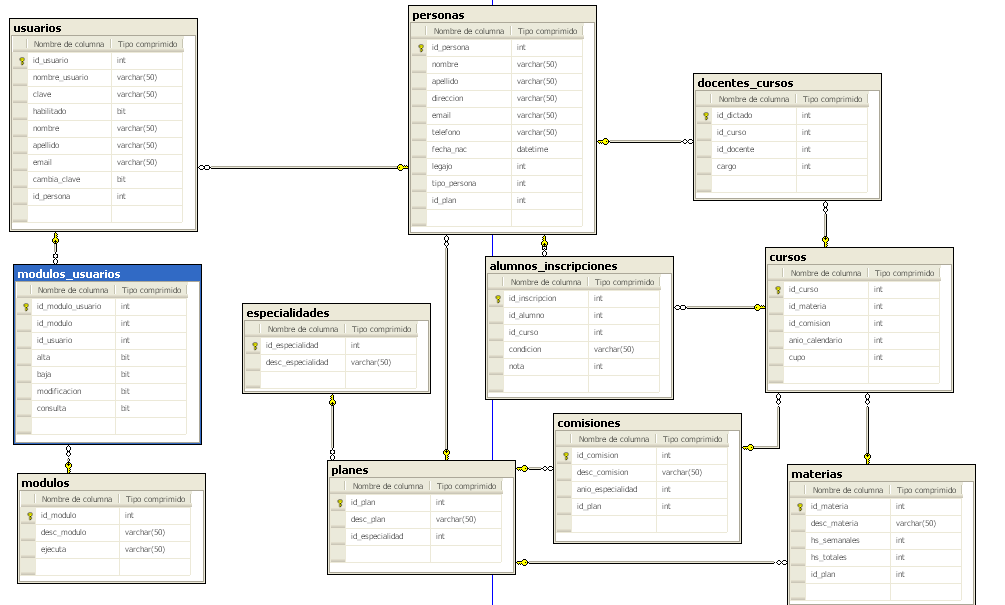
Teniendo en cuenta este modelo los **Requerimientos Funcionales** a implementar son los siguientes:

1. ABM de Usuarios y Permisos por Modulo
2. ABM de Alumnos y Docentes (Personas)
3. ABM de Especialidades
4. ABM de Profesores
5. ABM de Planes y Materias
6. ABM de Comisiones
7. ABM de Cursos
8. Inscripciones de Alumnos a Cursos
9. Registro de Notas
10. Reporte de Cursos
11. Reporte de Planes

**Modelo de Objetos**

## https://lh4.googleusercontent.com/mW1Be89e1_FS7G7lnlkaG73xmUENUakBRQIsC76vN4Oi_GeYk7rqstLVG6JC_vM_rtevMWT5vfbdCpE4Q4lSAs9TgI08v6yIAHUb8QhS3xlsgUpXCJIKwo0COWIaf48k

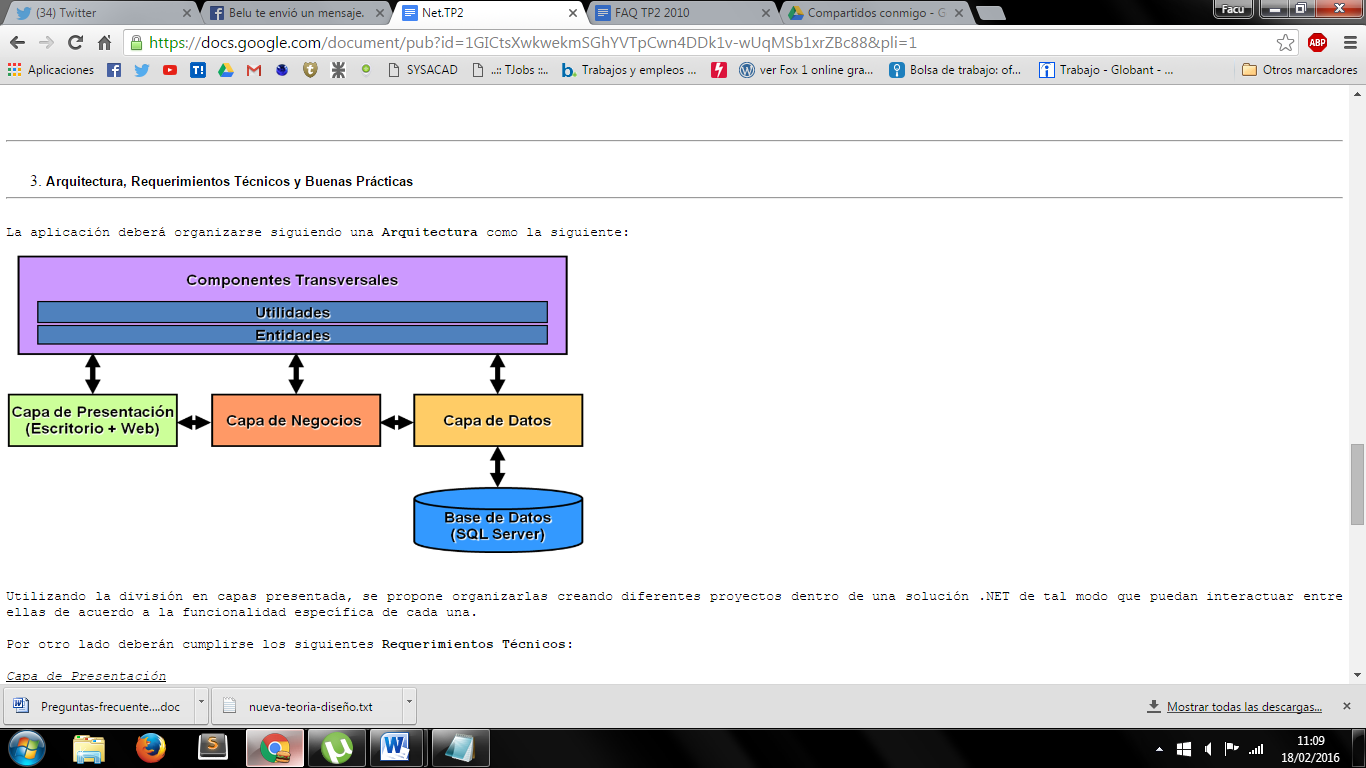
**Modelo de Datos**

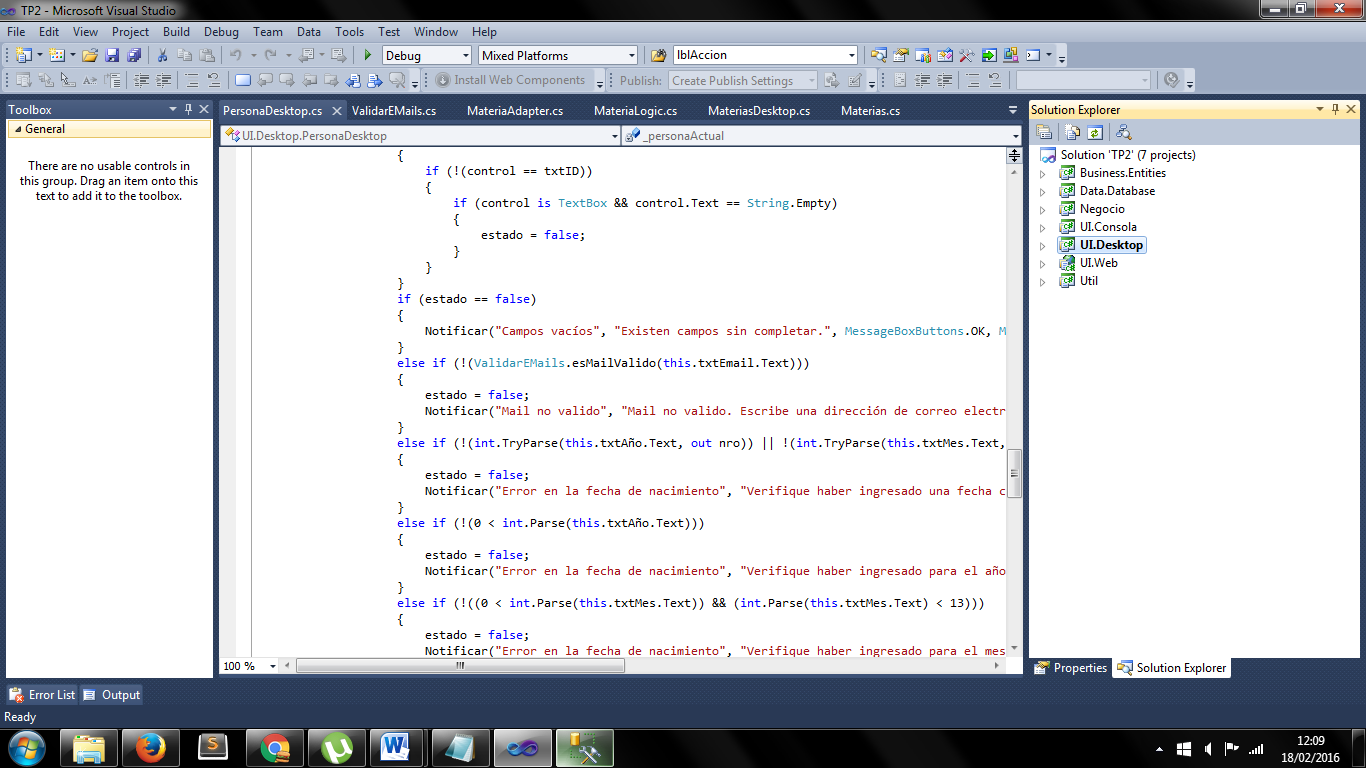


# Descripción de las capas

El SAG se conforma de distintas capas, entre ellas, la capa de **Presentación** (Escritorio + Web), la capa de **Negocio**, la de **Datos** (con su respectivo acceso a la ***Base de Datos SQL Server***), y una capa transversal de **Utilidades** y **Entidades** que tiene comunicación con las demás.

La aplicación se organiza siguiendo una **Arquitectura** como la siguiente:





***Vista Jerárquica de la Solución***

En la capa de **Presentación** (*UI.Desktop y UI.Web*) se encuentran los formularios, tanto de Escritorio (***WinForm***) como de web (***WebForm***), que permiten realizar las operaciones de Alta, Baja, Modificación y Consulta a cada una de las entidades que forman la aplicación, incluyendo la inscripción de un alumno a un curso y la asignación de los docentes a los cursos.

La capa de **Negocio** (Negocio) tiene como función ser un medio de enlace y comunicación entre la capa de Presentación y la de Datos, conteniendo las distintas validaciones de negocio que se fueron identificando a lo largo del desarrollo. Algunas de ellas son:

* *Validar que un Alumno no pueda volver a inscribirse al mismo Curso.*
* *Validar que un Docente no se vuelva a asignar a un Curso al cual ya fue asignado.*
* *Validar que haya cupo en un Curso.*
* *A la hora de dar de alta una materia, validar que las hs semanales no sean mayores a las hs totales.*
* *Validar que no se ingresen dos Personas distintas con el mismo legajo.*
* *Validar que un Usuario que se haya identificado en el sistema, esté habilitado o no (con esta opción una Persona puede tener más de un Usuario dentro del sistema).*

La capa de **Datos** (*Data.Database*) contiene métodos específicos para recuperar y actualizar (***selección, insertar, editar y borrar***) registros almacenados en una base de datos (**tp2\_net**) con su respectivo control de excepciones en caso de que alguno falle durante la ejecución de la aplicación.

La capa de **Entidades** (*Bussines.Entities*) contiene todas las clases que conforman el modelo de objeto, con sus respectivas variables, métodos y propiedades.

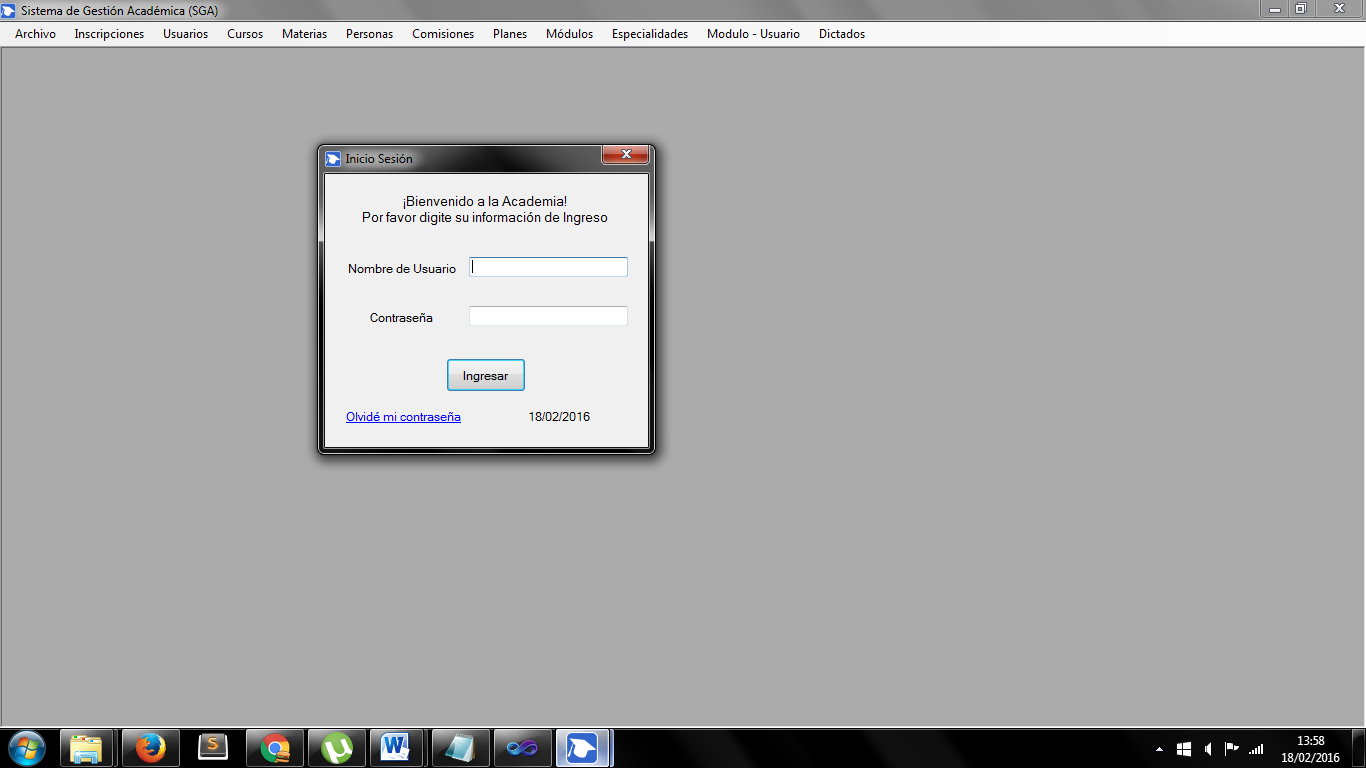
Existe una capa extra, **Utilidades** (Util), que se compone de utilidades representadas en clases que realizan algún tipo de validación/verificación:

* ***ErrorEliminar***: Es una clase que hereda de Exception que maneja las excepciones que ocurren cuando se intentan eliminar registros de una tabla de la base de datos, que poseen dependencias con otros registros en otras tablas.
* ***ValidarEMails***: Es una clase que posee un solo método que valida, mediante una expresión regular, que el email ingresado (en la ejecución en Escritorio) sea un email válido.

# Otras funcionalidades

* Las capas que conforman la Presentación (tanto Escritorio como Web) poseen un **menú** que solo puede ser accedido mediante una **autentificación del usuario**.

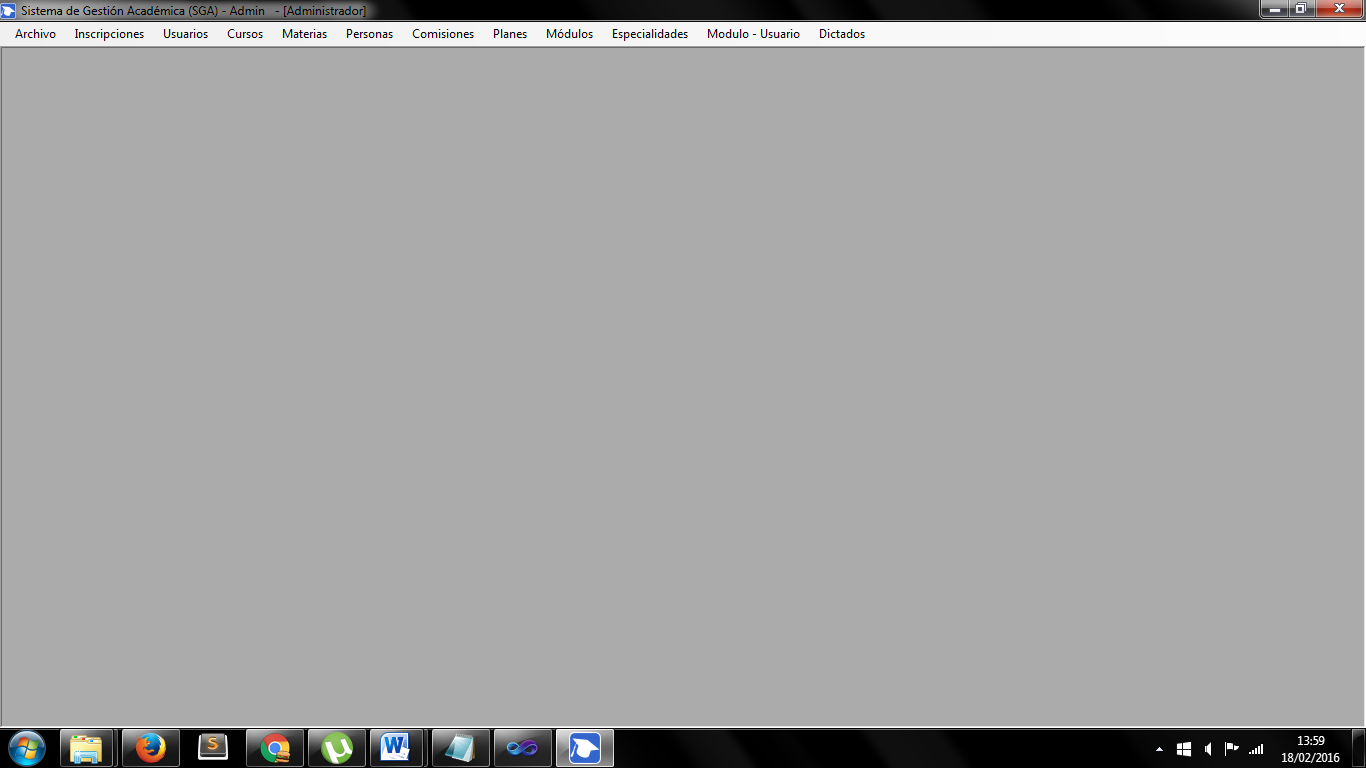
**Escritorio:**



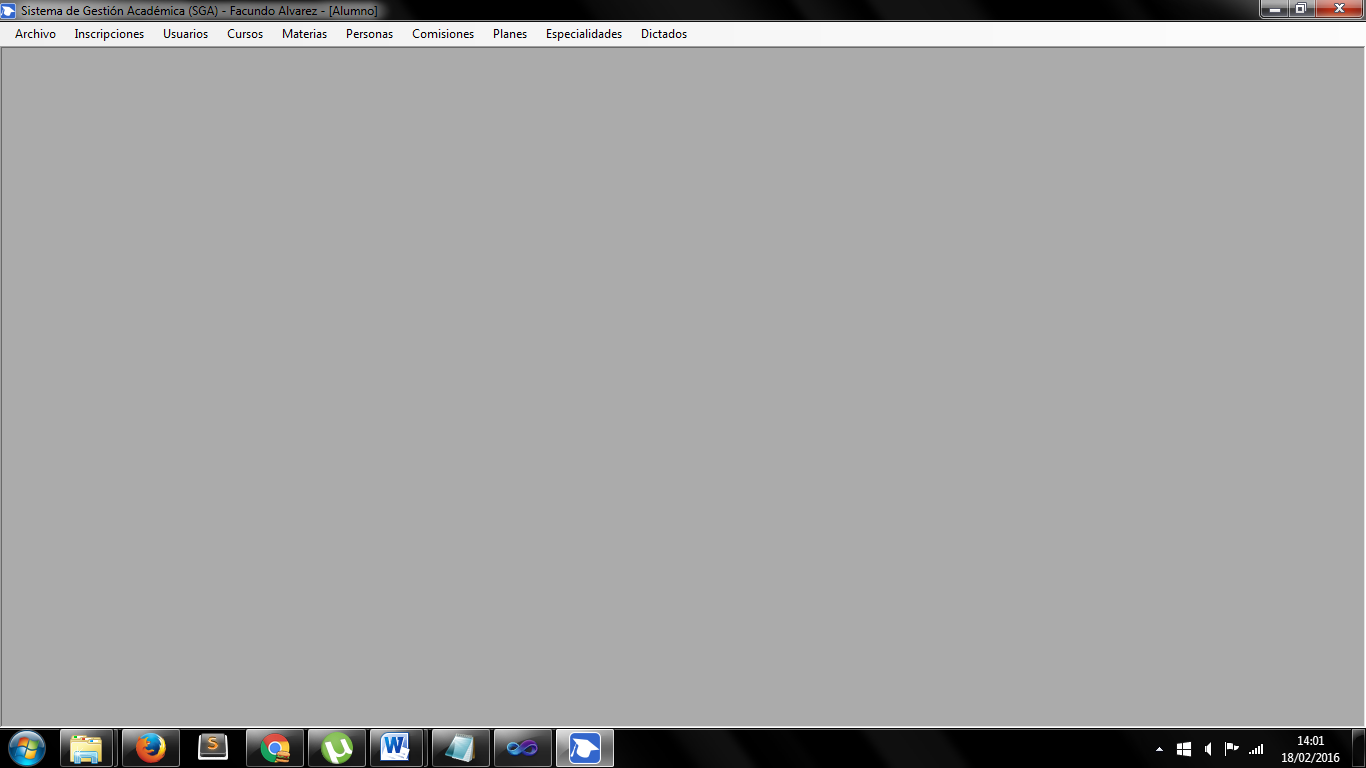
***Login***

Menú según el rol:

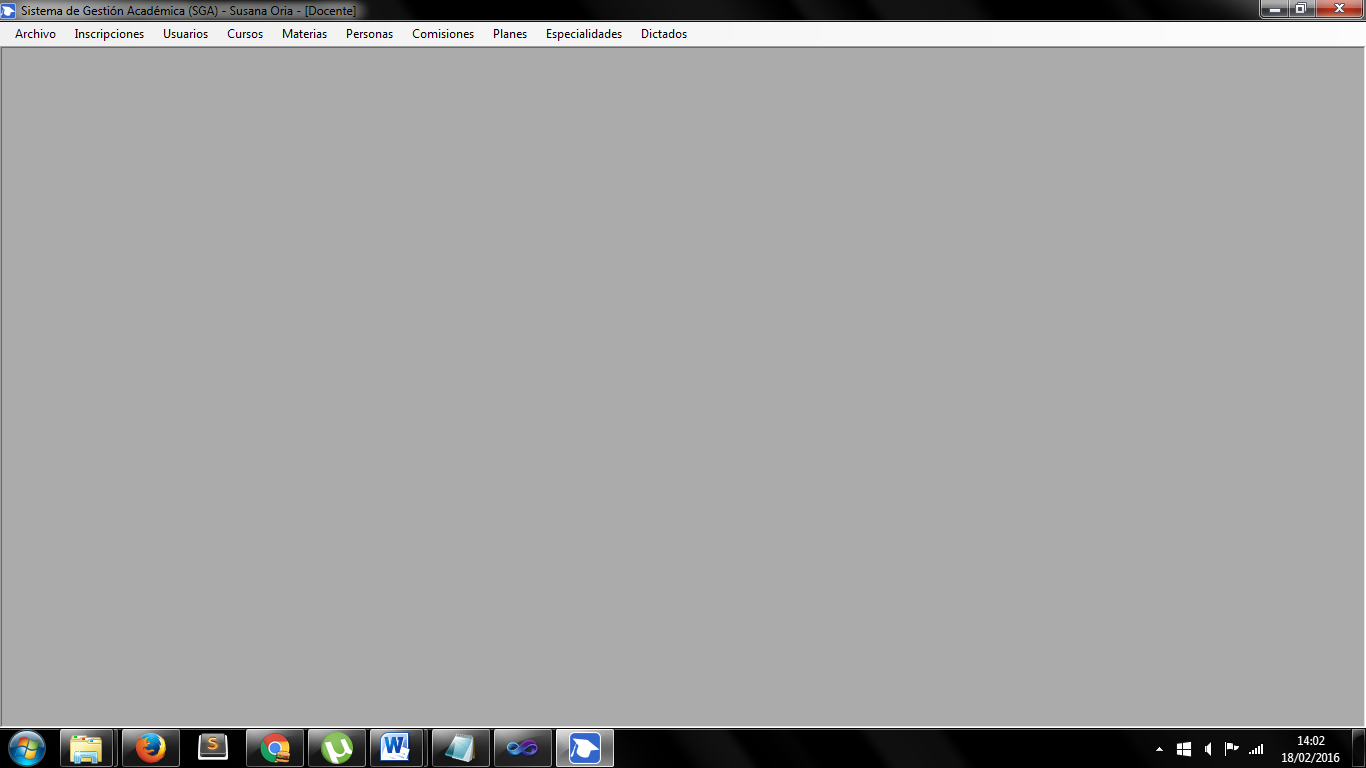
- Administrador



- Alumno



- Docente



**Web:**



***Login***

El formulario de logueo se compone de un User Control (Control Usuario), un control vacío que se puede usar para crear otros controles, es un objeto ASP.Net que comienza con este tipo de etiqueta:

<%@ Control Inherits="UI.Web.Login1" … … … … %>

Y se instancia dentro de un WebForm de la siguiente forma:

<uc1:Ingreso ID=Ingreso runat=server />

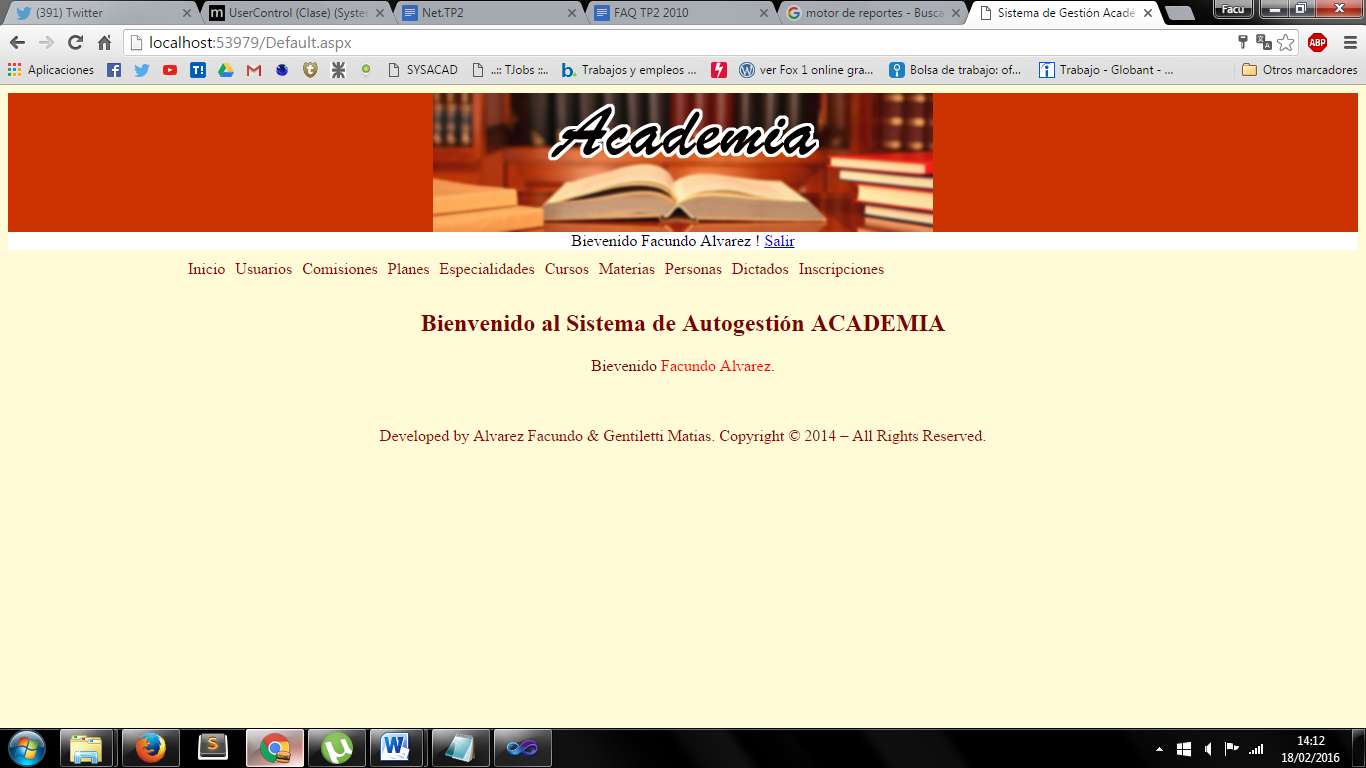
*Cuando se crea el objeto User Control se le da también un nombre, en este caso “Ingreso”.*

Menú según el rol:

- Administrador



- Alumno

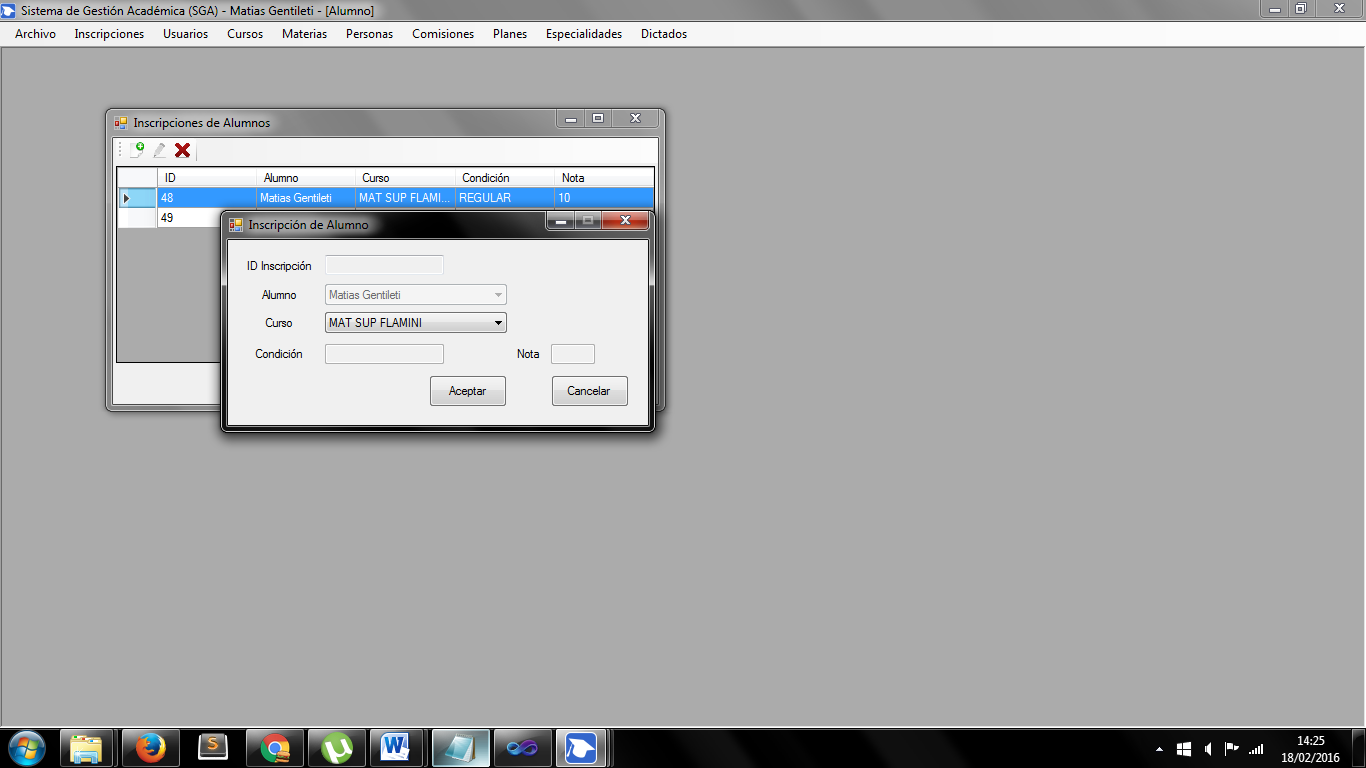


- Docente



En ambos ámbitos de la Presentación, si bien las opciones poseen el mismo nombre para todos los usuarios, **las funcionalidades son distintas según el rol**:

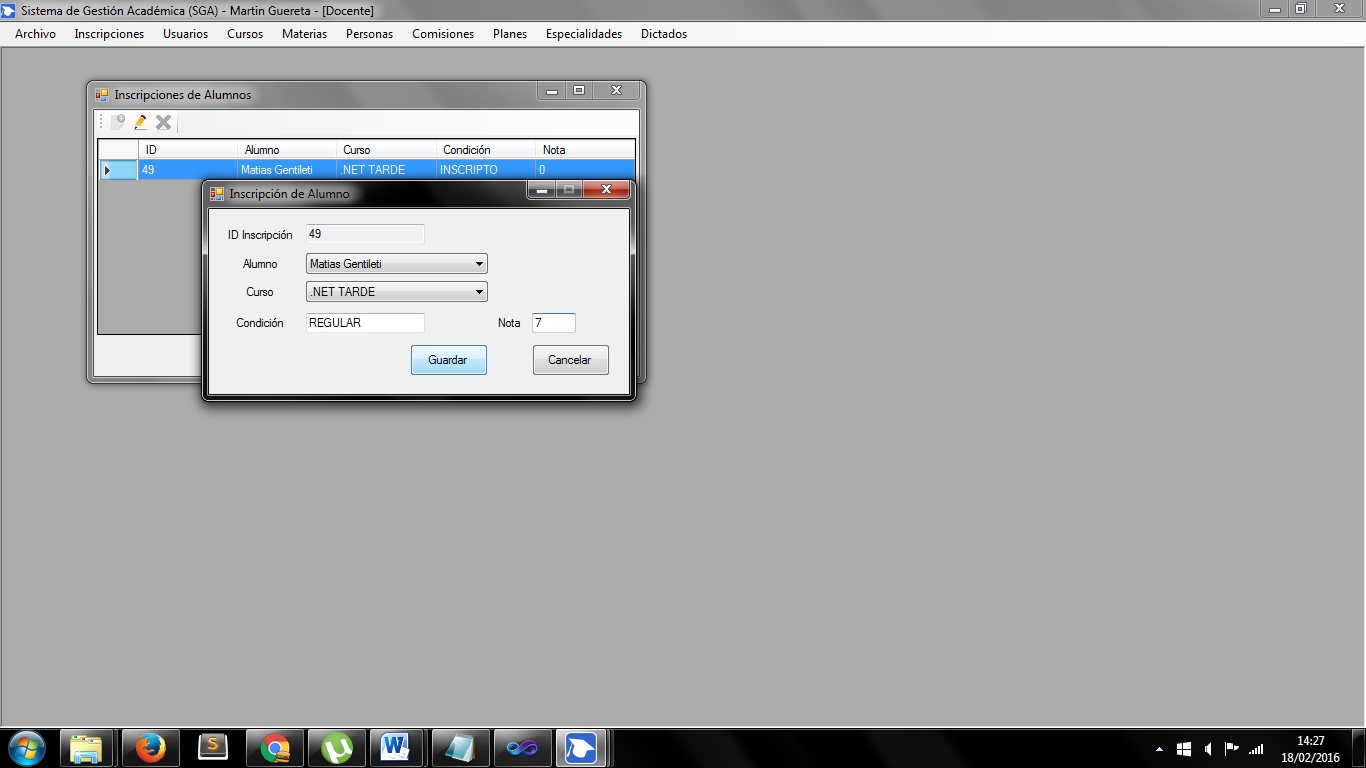
* + Tanto Alumno como Docente pueden editar sus datos personales y sus datos de logueo.
  + Según el Plan de la Persona/Usuario, se listan las Materias, Cursos, Comisiones y Dictados.
  + Un Alumno puede inscribirse solo a materias de su Plan.
  + Un Docente puede dictar los Cursos que pertenezcan a su Plan.
  + Un Docente solo puede editar las inscripciones (actualizar) de los Alumnos inscriptos a sus Cursos.
  + El Administrador tiene acceso a todas las opciones de selección, edición, modificación y eliminación en todas las opciones. Y además, se agregan las opciones de Módulos y Módulos Usuario.



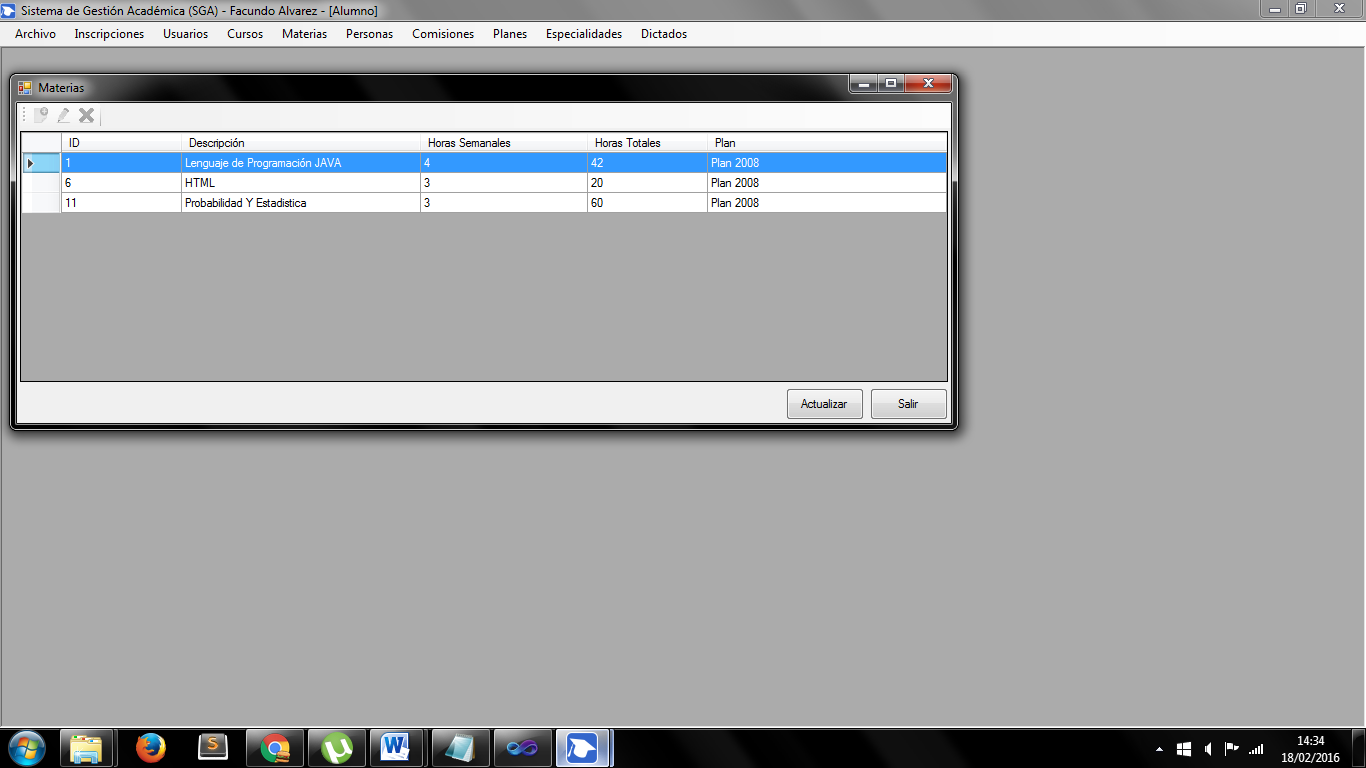
***Escritorio: Un Alumno se inscribe a un Curso***

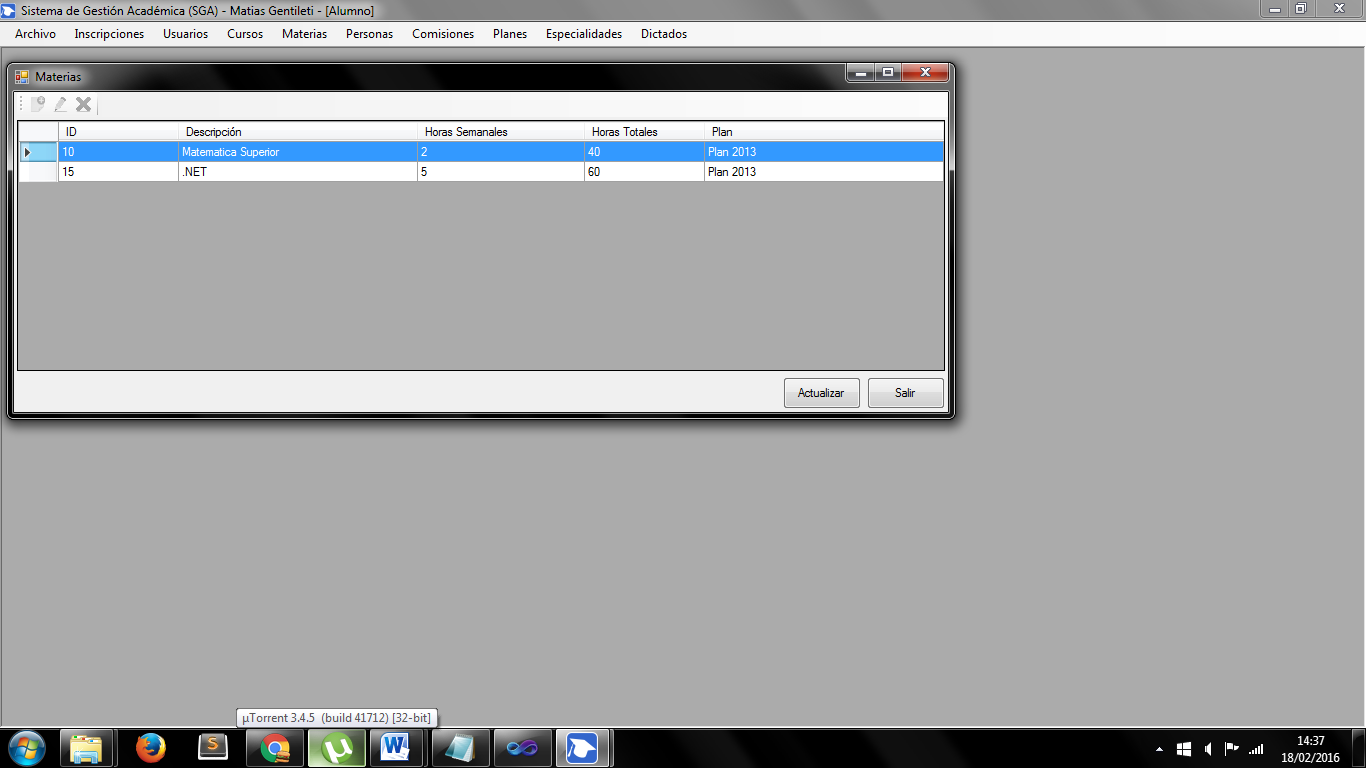


***Web: El Administrador asociando un Docente a un Curso***



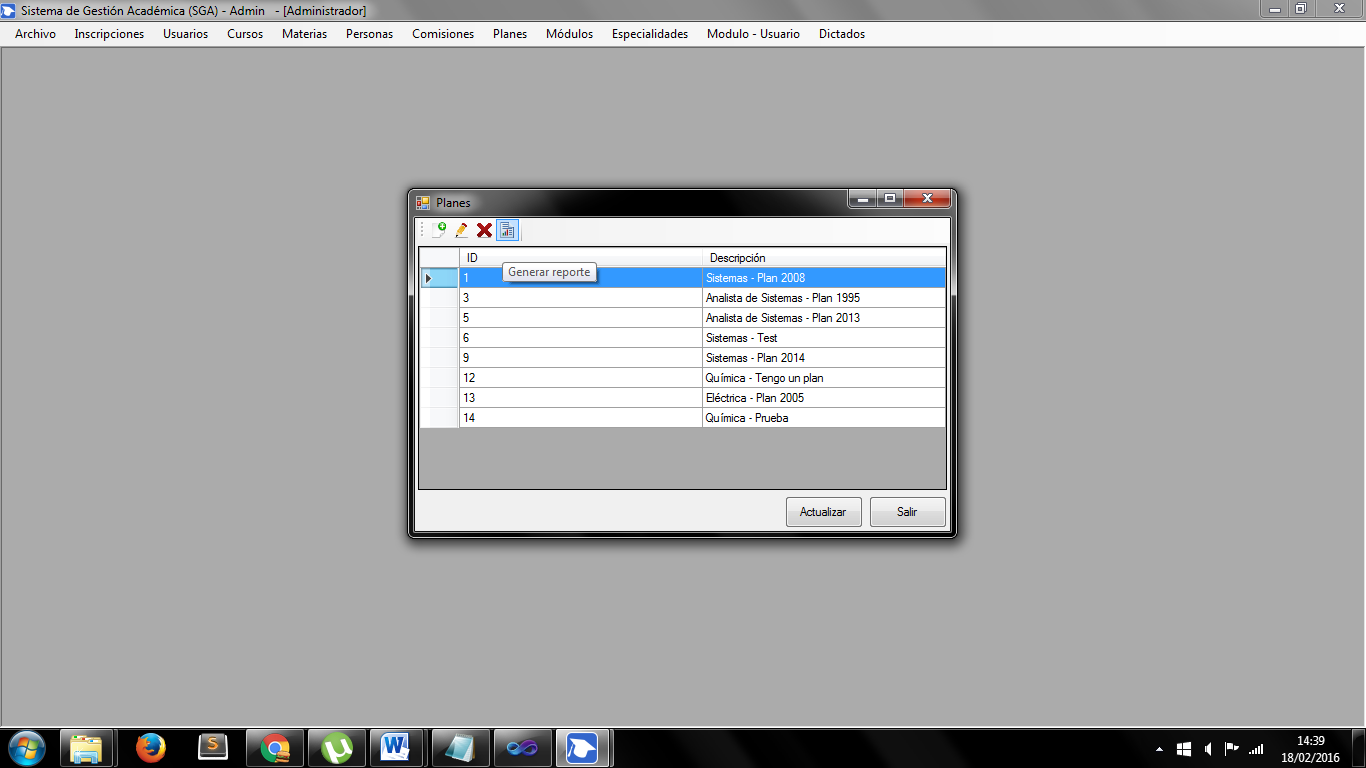
***Escritorio: El Docente actualiza una Inscripción de uno de sus Cursos***

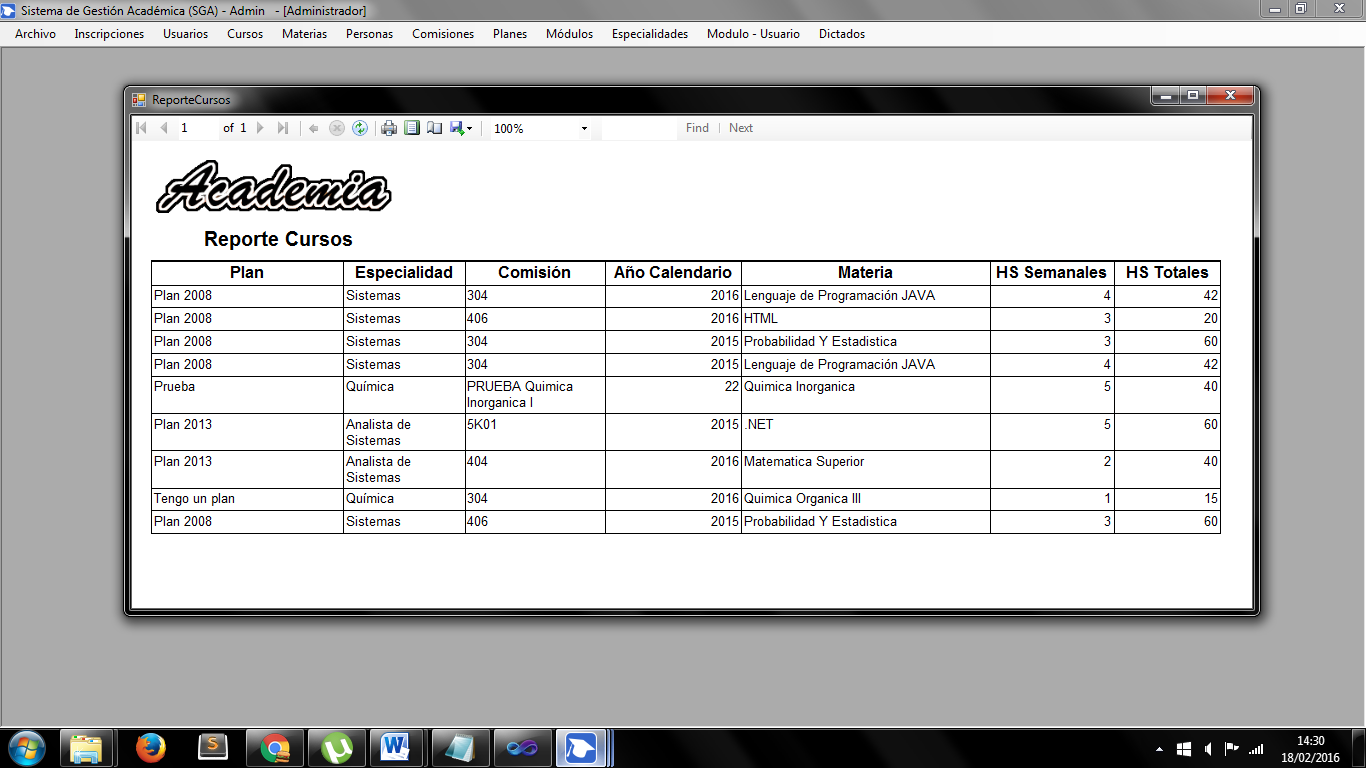




***Dos Alumnos que poseen distintos Planes pueden ver sólo las materias correspondientes a dicho Plan***

* El SAG ofrece la opción de **generar Reportes**: de Cursos y de Planes.



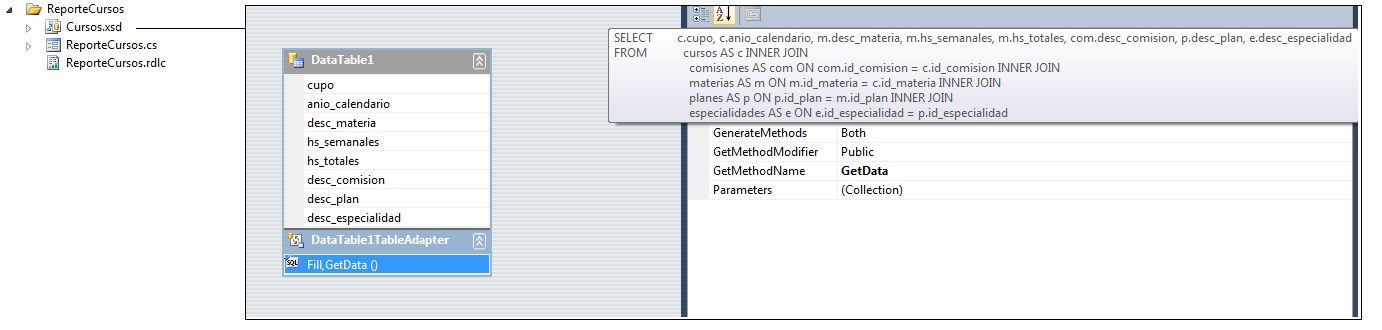


***Escritorio: Reporte de Cursos***



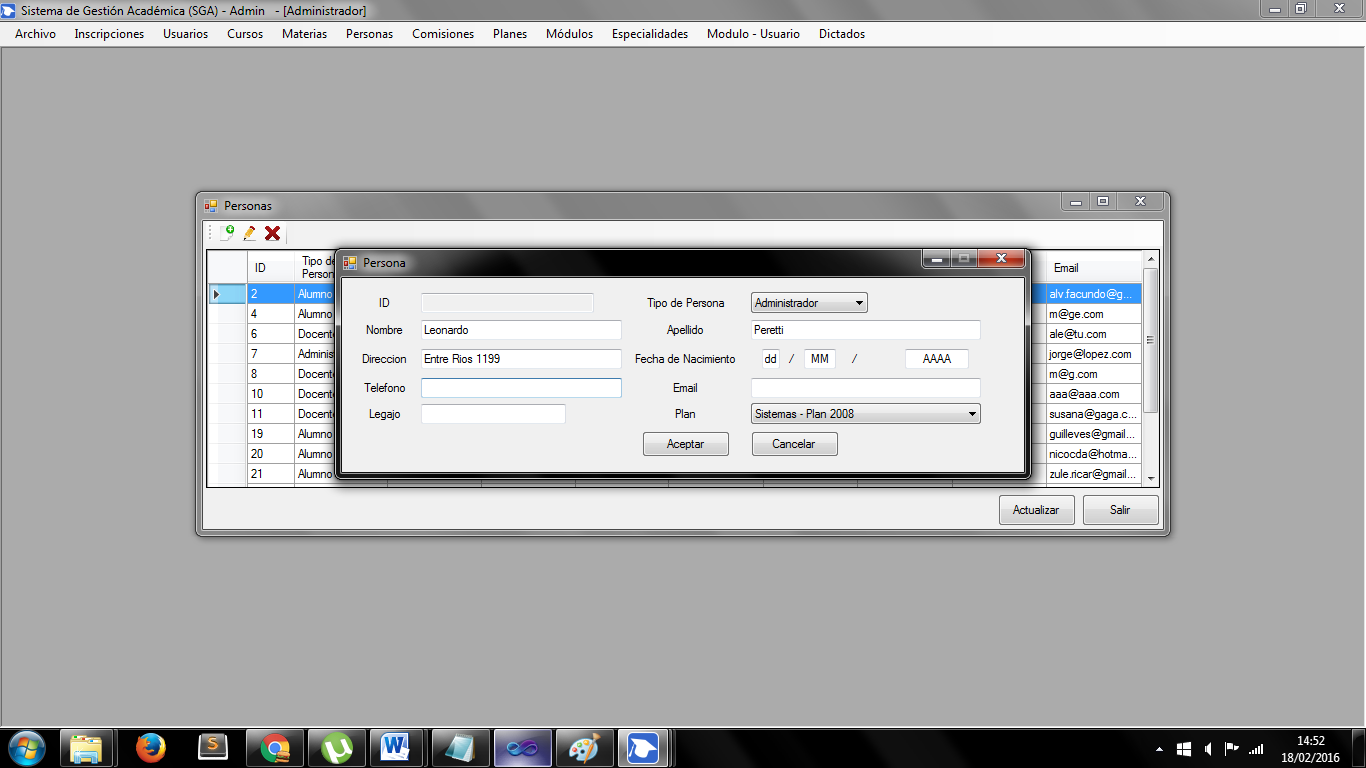
***Web: Reporte de Planes***

La fuente de los reportes se establece a través de un **DataSet Tipado** que contiene una sentencia SQL, y son vistos a través de un **ReportViewer**.

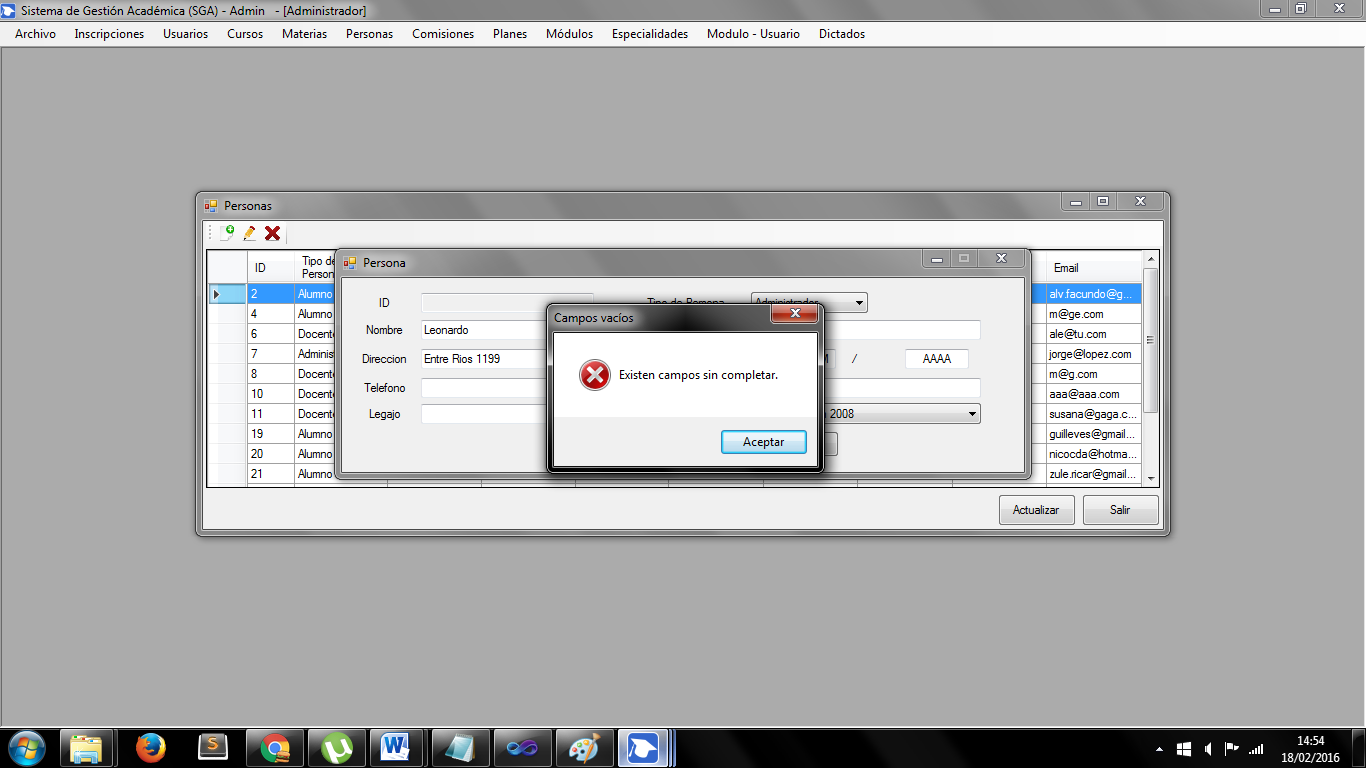


***DataSet Tipado + Sentencia SQL***

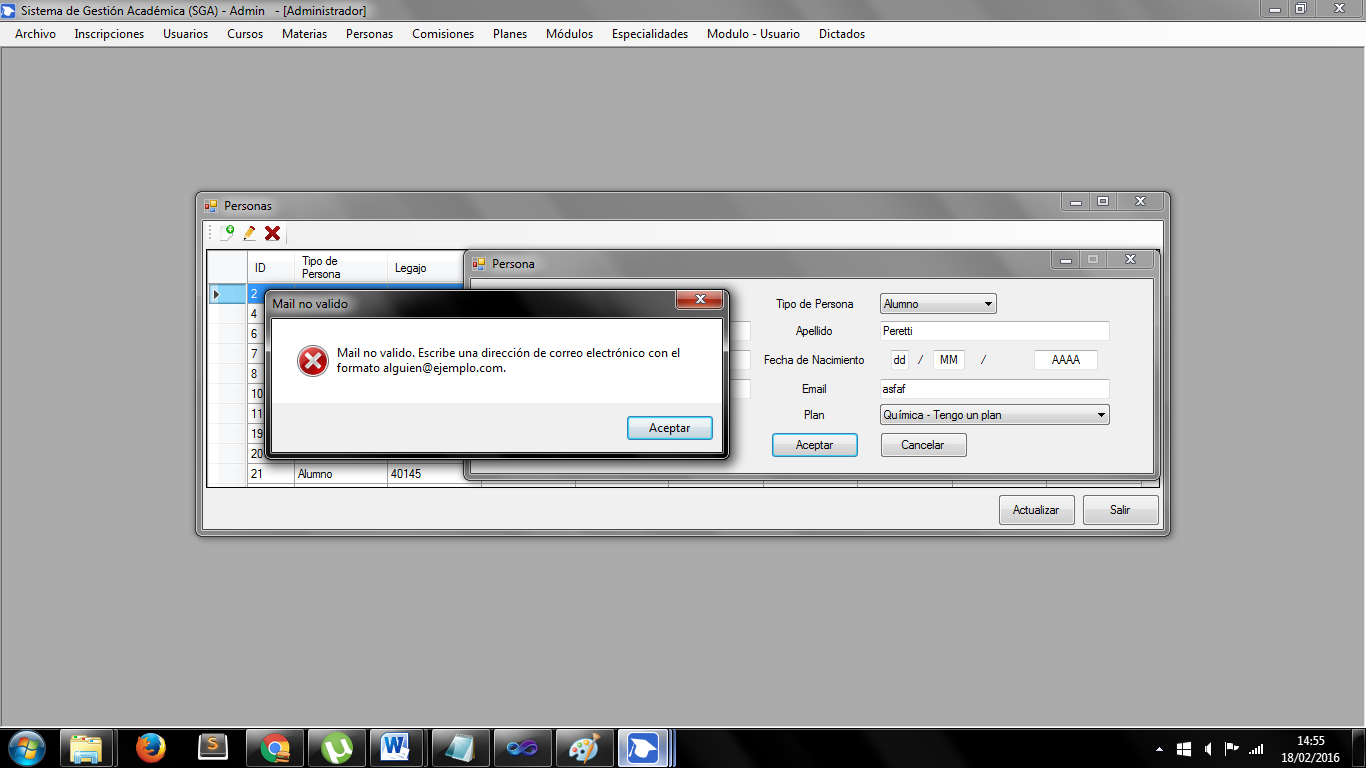
# Capturas de pantalla



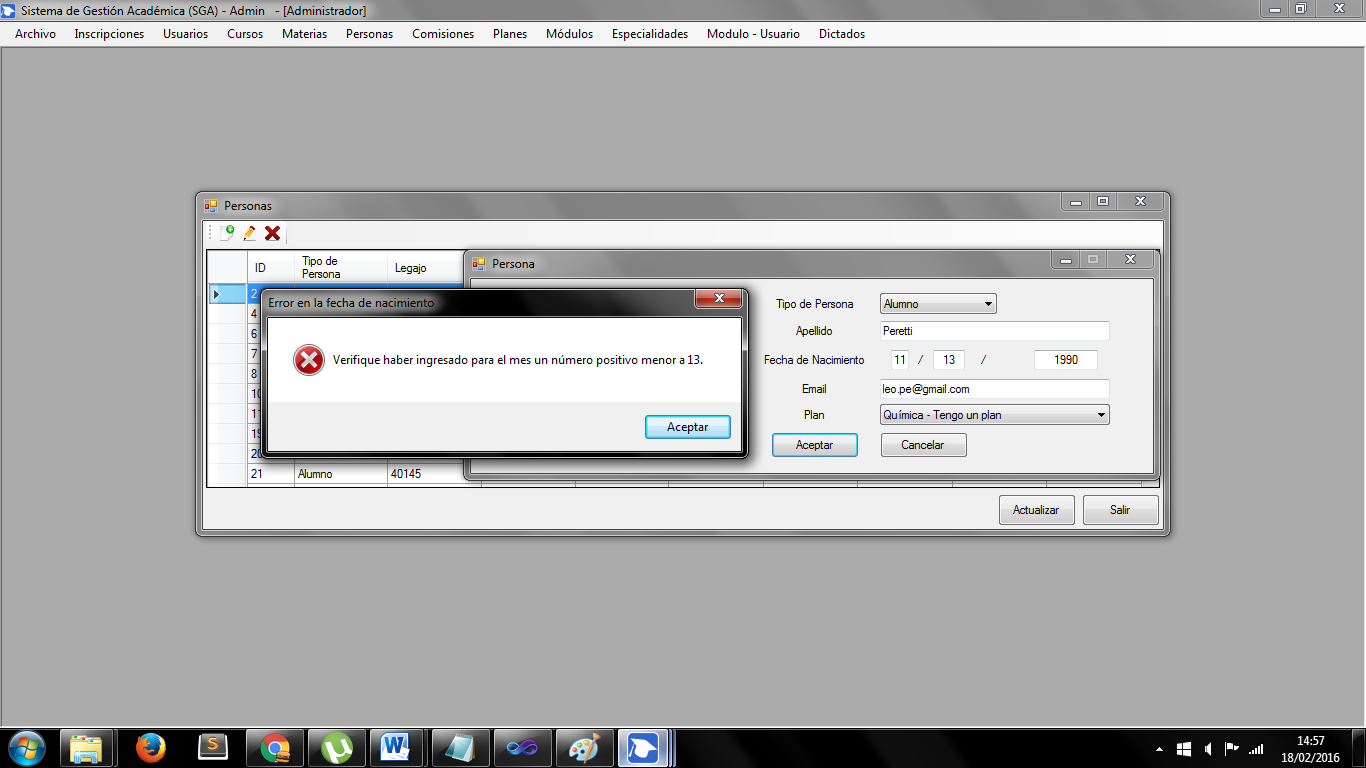
***Escritorio: ABM Persona***



***Escritorio: Validación de campos vacíos***



***Escritorio: Validación del email (a través de una expresion regular)***



***Escritorio: Validación de la fecha de nacimiento***



***Web: ABM Usuario***



***Web: Validaciones del lado del Cliente (campos vacíos)***

******

***Web: Validación del lado del Servidor (legajo repetido)***

Las validaciones del lado del cliente se realizan a través de elementos ASP.Net, con ellos verificamos que *no existan campos vacíos* y que el *tipo de dato ingresado sea el esperado* (a través de expresiones regulares). Mediante estos elementos, no sobrecargamos al servidor y se ahorra más tiempo en el proceso.

Por otro lado, las validaciones del lado del servidor, son validaciones que involucran las reglas de negocio, es decir, que “atrapa” la capa de Negocios: *existencia de cupos, existencia de usuario, inscripciones repetidas, legajos repetidos*, etc. Este tipo de validación si se envía al servidor, porque es él quien las puede tratar y responder.