



**BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE  
PUEBLA**

---

---

**FACULTAD DE CIENCIAS DE LA  
COMPUTACIÓN**

**DISEÑO Y DESARROLLO DE UN SISTEMA DE  
REGISTRO DE CALIFICACIONES BASADO EN  
SERVICIOS WEB**

**Tesis para obtener el título de:  
LICENCIADO EN CIENCIAS DE LA  
COMPUTACIÓN**

**PRESENTA:  
NAHUM FRANCISCO CORTÉS CERVANTES**

**ASESOR:  
DOCTOR MARIO ANZURES GARCÍA**

**PUEBLA, PUE.**

**SEPTIEMBRE 2013**

## Índice.

1	Introducción.....	1
1.1	Historia.....	1
1.2	Justificación.....	2
1.3	Antecedentes del proyecto.....	3
1.4	Objetivos.....	3
1.5	Aportaciones.....	4
2	Estado del arte.....	4
2.1	Herramientas de desarrollo web.....	4
2.1.1	Herramientas para el desarrollo del sistema.....	5
2.1.2	LINQ.....	9
2.2	Servicios web.....	10
2.2.1	XML.....	11
2.2.2	SOAP.....	12
2.2.3	WSDL.....	14
2.2.4	UDDI.....	15
3	Sistema gestor de calificaciones.....	18
3.1	Justificación.....	18
3.2	Análisis de requisitos.....	19
3.2.1	Ámbito del sistema.....	21
3.2.2	Funciones del sistema.....	23
3.2.3	Características de los actores del sistema.....	24
3.2.4	Casos de uso.....	24
3.3	Diseño.....	34
3.3.1	Diseño de la Base de Datos.....	34
3.3.2	Diagramas de clases.....	37
3.3.3	Diagramas de actividades.....	38
3.3.4	Diagramas de secuencia.....	43
3.4	Implementación.....	44
3.4.1	Desarrollo del sistema mediante Silverlight.....	44
3.4.2	Servicios realizados.....	53
4.	Conclusión y trabajo futuro.....	61
4.1.	Conclusión.....	61
4.2.	Trabajo futuro.....	62
5.	Bibliografía.....	62
6.	Apéndices.....	63
A.	Acrónimos.....	63

B. Definiciones.....	64
C. Manual de usuario. ....	69
C.1. Página principal. ....	69
C.2. Validando datos. ....	69
C.3. Ingreso al sistema. ....	71
C.4. Administrar mi cuenta. ....	72
C.5. Listas de calificaciones.....	75
C.6. Cargar grupo. ....	80
C.7. Cambiar contraseña.....	86
C.8. Administrar usuarios (Administrador).....	89
C.9. Alta de período (Administrador). ....	98
D. Manual técnico. ....	100
D.1. Descripción general. ....	100
D.2. Parte Silverlight “ <i>Proyecto</i> ”.....	101
D.3. Parte web “ <i>Proyecto.Web</i> ”.....	111

# 1 Introducción.

En el presente trabajo de tesis se presenta el análisis, desarrollo e implementación de un sistema para la gestión de calificaciones utilizando un servicio web compuesto que permita a un profesor de la Facultad de Ciencias de la Computación (FCC) crear, registrar y modificar sus listas de calificaciones por materia y periodo en cualquier computadora que tenga accesos a internet o esté dentro de la red de la Benemérita Universidad Autónoma de Puebla (BUAP).

Los servicios web [**Rotem-Gal-Oz 4**] son calificados como una nueva etapa del desarrollo de los sistemas distribuidos, permiten aprovechar todas las ventajas de trabajar en un ambiente web y a su vez contar con una amplia gama de tecnologías basadas en XML (*Extensible Markup Language*) [**XML 8**], facilitando la granularidad e interoperabilidad, pueden ser utilizados para el desarrollo de aplicaciones o sistemas complejos gracias a su composición. Tales sistemas o aplicaciones se consideran como servicios web compuestos, donde se debe considerar la interacción, coordinación e integración para su buen funcionamiento. La importancia de los servicios web es que pueden ser consumidos por cualquier aplicación, sin preocuparse por cómo fueron construidos, donde residen, sobre qué sistema operativo se ejecutan o cómo acceder a ellos, porque permiten crear una interfaz de acceso a las aplicaciones.

## 1.1 Historia.

El desarrollo de aplicaciones distribuidas ha evolucionado en la búsqueda de la interoperabilidad y principalmente, en la integración de los componentes heterogéneos que son parte de dicha aplicación. Tal evolución da inicio con *Sockets*, que presenta una forma primitiva de comunicación entre componentes distribuidos, en la cual el usuario debía tener la información necesaria para que el cliente hiciera peticiones al servidor y viceversa. Después, con el paradigma basado en procedimientos, surge *RPC (Remote Procedure Call)*, que hace la comunicación transparente al usuario, es decir, ni el cliente ni el servidor deben tener la información necesaria para interactuar. Cuando surge el paradigma orientado a objetos se desarrolla *RMI (Remote Method Invocation)*, que es similar a *RPC* pero trabaja con objetos en lugar de funciones o procedimientos, en particular, *RMI* trabaja con el lenguaje *Java*. [**Deitel 23**]

Ni *Sockets*, *RPC* o *RMI* permiten la interoperabilidad y mucho menos la integración de aplicaciones, con la idea de proveer esos dos aspectos no funcionales, se crea *CORBA (Common Object Request Broker Architecture)* [**Henning 12**], que facilita la interoperabilidad entre componentes basados en objetos a través de un bus de comunicación denominado *ORB (Object Request Broker)*. *ORB* proporciona un conjunto de *IDL (Interface Definition Language)* con la finalidad de proporcionar la interoperabilidad entre componentes desarrollados con diferentes lenguajes de programación. Sin embargo, en ocasiones no existe el *IDL* de determinados lenguajes y estos deberían ser desarrollados por el usuario de *CORBA*, lo cual dificulta la integración de aplicaciones heterogéneas.

*Sockets*, *RPC*, *RMI* y *CORBA* utilizan una comunicación síncrona basada en una arquitectura cliente-servidor, por tanto, presentan las desventajas tanto del tipo de comunicación como de la arquitectura presentada. Por ello, es que se desarrolla *MOM (Middleware-Oriented Messages)*, que presenta una comunicación asíncrona aunque sigue basada en una arquitectura cliente-servidor. El aspecto importante de *MOM* es que presenta una cola de mensajes, donde se guardan los mensajes que envían tanto

el cliente como el servidor, de esta manera no es necesario que el cliente o el servidor estén activos cuando alguno de los dos envía un mensaje al otro.

*EAI (Enterprise Application Integration)* se enfoca al ámbito de negocios y busca la integración de las aplicaciones de una empresa u organización, para lo cual propone una metodología basada en un bus de comunicación. Sin embargo, la integración se debe llevar a cabo a nivel de datos, plataforma, procesos e interface, por lo que se utiliza una herramienta para realizar cada una de éstas tareas de integración, haciendo costoso y difícil de implementar la metodología de *EAI*, principalmente en pequeñas empresas. **[Manouvrier 6]**

Posteriormente surge un paradigma llamado *Computo Orientado a Servicios (SOC)*, utiliza a los servicios como su principal elemento para obtener un desarrollo rápido y de bajo costo para aplicaciones distribuidas en sistemas heterogéneos. SOC consiste de un conjunto de servicios que débilmente acoplados crean procesos empresariales dinámicos y aplicaciones ágiles que pueden expandirse en organizaciones y distintas plataformas. **[Georgakopoulos 1]**

El desarrollo de aplicaciones SOC involucra el desarrollo de una *Arquitectura Orientada a Servicios (SOA)* **[Erl 2]**, que corresponde a un *Middleware* que permite el descubrimiento, utilización y combinación de servicios que interactúan virtualmente en cualquier proceso de negocios dentro de un contexto.

La base de este paradigma y de la arquitectura que utiliza son los *servicios web*, los *servicios web* pueden ser utilizados por cualquier aplicación, insertados en un servidor y visibles en una página web, por tanto son accesibles para cualquier aplicación ya que están basados en protocolos *XML*. Los servicios son entidades modulares, que encapsulan la lógica de negocio y que utilizan interfaces débilmente acopladas.

El proceso de componer servicios implica coordinar y determinar la forma en que interactuarán los servicios entre sí, en alguna literatura a tal proceso se le conoce como orquestación de servicios. Existen varias herramientas para llevar a cabo la composición de servicios, la más utilizada es *BPEL (Business Process Execution Language)*, que básicamente consiste en un lenguaje basado en *XML* diseñado para el control centralizado de la invocación de diferentes *servicios web*, con cierta lógica de negocio añadida que ayudan a la programación en gran escala. *BPEL* se desarrolló dentro del marco de la metodología empresarial *BPM (Business Process Management)* **[Cummins 7]**, cuyo objetivo es mejorar la eficiencia a través de la gestión sistemática de los procesos de negocio, que se deben modelar, automatizar, integrar, monitorizar y optimizar de forma continua.

En la actualidad existen tecnologías que facilitan el desarrollo de *servicios web*, su orquestación y generación de los protocolos de comunicación necesarios (*SOAP, UDDI, WSDL*), permitiendo solo realizar la funcionalidad del servicio y haciendo transparente la comunicación, un ejemplo de esta tecnología es *Windows Communication Foundation (WCF)* **[16 Johnson]**. *WCF* es un marco de trabajo para desarrollar aplicaciones orientadas a servicios, donde se puede tener la clásica interacción cliente servidor o también tener la interacción entre servicios, además permite una comunicación asíncrona.

## **1.2 Justificación.**

En la Facultad de Ciencias de la Computación los profesores necesitan un sistema capaz de llevar el control de sus grupos, que pueda ser accesible desde cualquier punto dentro de la Universidad, desde sus hogares o cualquier sitio con conexión a

internet. Necesitan que el sistema sea flexible, es decir que se pueda adecuar al criterio de evaluación de cada profesor, donde puedan generar reportes, obtener promedios y poder supervisar día a día el aprovechamiento de los alumnos en sus calificaciones.

### **1.3 Antecedentes del proyecto.**

Los profesores de la FCC deben contar con una lista de calificaciones por materia que utilizarán para diferentes gestiones administrativas y llevar el control de sus alumnos, anotando tareas, trabajos entregados, calificaciones de parciales, proyectos y presentaciones para obtener un promedio final del período cursado. Comúnmente éste proceso, la mayor parte de los profesores lo realiza de dos formas:

- 1) manualmente, la forma más tradicional, utilizando una hoja de papel y registrando la información en ella,
- 2) electrónicamente, manejando una o varias hojas de cálculo de Excel (generalmente) de acuerdo al número de materias que imparta el profesor.

Sin embargo, ambas tienen inconvenientes, en la primera forma, registrar las calificaciones en hojas de papel es impráctico en muchos sentidos, dificulta las modificaciones y existe la posibilidad de perder dicha hoja. La segunda forma, aunque es más práctica y sencilla, en ocasiones puede ser no portable, es decir, que la versión de Excel de la computadora en que se desee abrir la hoja de cálculo sea diferente y no se pueda abrir dicha hoja y también cabe la posibilidad de perder la memoria USB en que se tiene almacenado el archivo de Excel. Además, se tiene que recurrir a otro u otros programas de software para generar los reportes que le sean útiles al docente, tanto para sus tareas de gestión académicas como administrativas.

Por todo ello, es necesario que el profesor de la FCC cuente con una aplicación para generar y adaptar la lista de calificaciones de cada una de sus materias. Dicha aplicación debe ser:

- accesible vía web (el profesor pueda usarla desde cualquier sitio donde haya Internet),
- flexible (puede adaptar la lista de calificaciones de acuerdo a las necesidades del grupo y de la materia) y
- portable (que se ejecute su sistema sin importar con qué sistema o plataforma de software fue construido y dónde reside la aplicación).

Todos estos requerimientos los satisface el análisis, diseño e implementación de un servicio web compuesto que le permitirá generar y adaptar sus diferentes listas de calificaciones de forma rápida y sencilla.

### **1.4 Objetivos.**

Crear una aplicación para generar listas de calificaciones por materia, esta podrá ser adaptable por el profesor según sus criterios de evaluación.

La aplicación será accesible desde cualquier lugar, para cualquier tipo de computadora sin importar el sistema operativo.

Será una aplicación completa con el potencial de una aplicación de escritorio porque será construida como una aplicación rica de internet (RIA) y la portabilidad en los navegadores web será proporcionada por un Plug-In (Silverlight).

Integridad de los datos de las listas de cada profesor, ya que contará con seguridad para acceso al sistema, además los datos estarán centralizados en un servidor y disponibles para su uso en cualquier momento a través del portal.

## 1.5 Aportaciones.

Análisis y desarrollo de una aplicación basada en el paradigma de SOC para la integración de tecnologías heterogéneas, en específico SOA enfocado en los servicios web y en especial en la pila básica de servicios para la mejora del proceso de llenado de actas y trámites administrativos con las listas de calificaciones.

Un conjunto de servicios web que posteriormente podrán ser consumidos por otras aplicaciones, para hacer crecer la aplicación o crear nuevas aplicaciones.

## 2 Estado del arte.

### 2.1 Herramientas de desarrollo web.

#### Introducción.

En toda actividad industrial, comercial y de servicios desarrollada en las empresas, universidades, centros de investigación, se requiere constantemente de información accesible desde cualquier punto, y debido a que la información no siempre está centralizada y es necesario tener constante interacción con la información y con otros usuarios, entonces surge el desarrollo web aprovechando tecnologías de comunicación como las redes, en este caso el internet o una intranet, se puede contar con aplicaciones portables capaces de ejecutarse en un navegador web, en las cuales a través de un servidor se gestione la información y se proporcionen herramientas para la interacción de usuarios y procedimientos específicos para el uso de la información.

El desarrollo web es un conjunto de tecnologías de software del lado del servidor y del cliente que involucran una combinación de procesos, actividades y bases de datos con el uso de un navegador web a fin de realizar determinada tarea o consulta de información.

Los lenguajes de programación más usados en desarrollo web son: *.NET* y *JSP*, en este caso solo nos centraremos en *.NET*.

Las herramientas para desarrollo *.NET* son *Visual Studio* y *Mono*, en cuanto a *JSP* son *NetBeans*, *Eclipse* y *Jedit* basadas en el lenguaje de programación *Java*.

Para el aprovechamiento de la red, conectividad de aplicaciones y reutilización de software, surgen los servicios web, por la necesidad de estandarizar la comunicación entre diversas plataformas y lenguajes de programación. Los servicios web son funciones publicadas en la web, que gracias a protocolos de comunicación estándares, permiten interactuar con distintas aplicaciones y su reutilización.

Para el almacenamiento de datos y distribución, los gestores de bases de datos más populares en el desarrollo web son *MySQL*, *Oracle*, *SQLServer* y *PostgreSQL* [Beaulieu 21], en este caso se utilizara *SQLServer 2008* por la compatibilidad con *Visual Studio* para desarrollo de aplicaciones *.NET*. [Aitchison 20]

### 2.1.1 Herramientas para el desarrollo del sistema.

En el año 2000 *Microsoft* anuncio *.NET* una nueva visión de internet para el desarrollo, construcción, distribución y uso de software para web, esta herramienta permitió a los desarrolladores no tener solamente un lenguaje de desarrollo, ya que *.NET* es compatible con *Visual Basic*, *C#* y otros lenguajes. [Sharp 19]

Al mismo tiempo con el lanzamiento de *.NET*, *Microsoft* anuncio el lenguaje de programación *C#* (*C-Sharp*) creado específicamente para la plataforma *.NET*, este lenguaje tiene sus raíces en *C*, *C++* y *Java*. El lenguaje *C#* es orientado a objetos, esto permite tener acceso a una *Librería de Clases del .NET Framework* y poder utilizar todo su potencial.

*.NET Framework* es una colección pre construida de componentes que permite realizar aplicaciones completas en un corto tiempo. [Deitel 13]

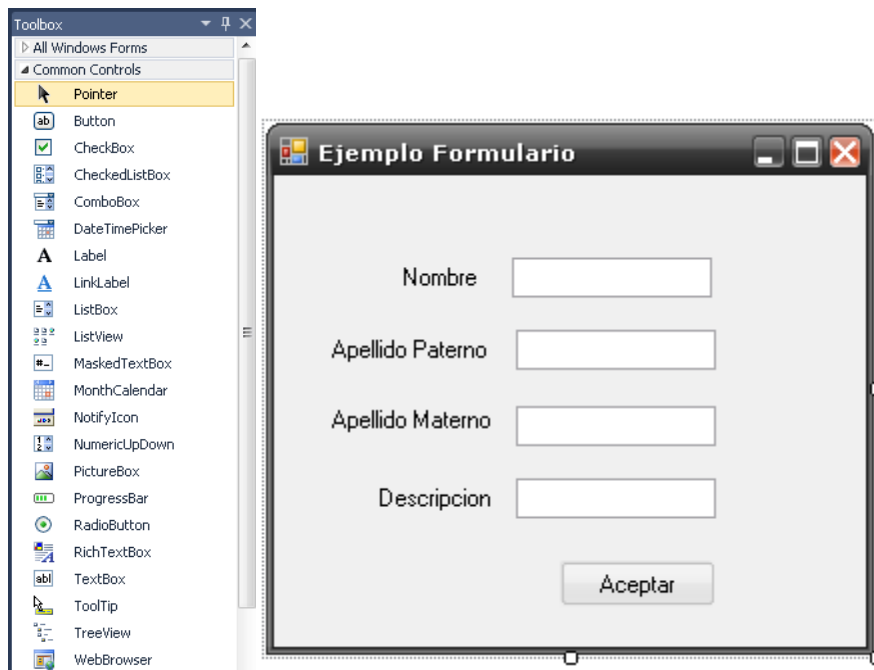
*C#* al igual que *Java*, permite la programación y utilización de eventos iniciados por el usuario como el clic del mouse, eventos lanzados al cargar una página web o cambiar de ventana. Este lenguaje de programación en combinación con el *.NET Framework* provee de una herramienta para el desarrollo de *Interfaces Graficas de Usuario* (*GUI - Graphical User Interface*), ya que permite gracias a su interfaz de desarrollo realizar la creación de *GUI's* arrastrando los controles (botones, cajas de texto, mensajes, matrices, etc.) a un panel de diseño para posteriormente implementar la lógica de la interfaz con el lenguaje.

El *GUI* es un programa informático que ayuda a interactuar a un usuario visualmente con una computadora utilizando un conjunto de imágenes y objetos gráficos para representar información y acciones, permite proveer interfaces intuitivas para poder interactuar de manera sencilla y amigable con una aplicación incrementando la productividad de cierta tarea a desarrollar.

Parte de la iniciativa incluida por el desarrollo de *.NET* y el lenguaje de programación *C#* era *ASP.NET* [Northrup 18], que permitía crear aplicaciones web extendiendo la idea de la reutilización, permitiendo concentrarse en la programación y dejando de lado la creación de componentes como botones, cuadros de dialogo, barras de desplazamiento y ventanas para cada aplicación, para solo enfocarse en la funcionalidad de cada elemento de la interfaz desarrollada.

La creación de *GUI's* es a través de formularios de *Windows* (*Windows Forms*, controles *WPF* o *Web Forms*). Un formulario es un elemento gráfico que aparece en el escritorio de la computadora, puede ser un dialogo, una ventana o un *MDI window* (*Multiple Document Interface Window*). Están conformados por clases, en las cuales se definen comportamientos de los componentes que se implementan, por ejemplo, cuando un componente es cargado.

En la siguiente figura (Figura 2.1) se muestra un ejemplo de un formulario de *Windows*, donde todo el contenido de la ventana es un conjunto de controles predeterminados, que para la creación de la interfaz solo es necesario arrastrar los controles de la barra de herramientas ubicada a la izquierda y dar el formato deseado para la creación del *GUI*.



**Figura 2.1.** Ejemplo de un formulario de Windows (Windows Form).

Para hacer la lógica del programa, se realiza con código C#, por ejemplo en el método *private void button1\_Click* (Figura 2.2.), es un evento lanzando al realizar un clic en el botón “Aceptar”, ahí adentro va la función que se desea realicé el botón al ser presionado, la función *public Form1* es el constructor del formulario que inicializa los componentes de la interfaz que se crearon.

```

public partial class Form1 : Form
{
    public Form1()
    {
        InitializeComponent();
    }

    private void button1_Click(object sender, EventArgs e)
    {

    }

}

```

**Figura 2.2.** Ejemplo de código detrás de un formulario de Windows (Windows Form) en lenguaje C#.

Anteriormente se tenían que utilizar múltiples tecnologías para la construcción de aplicaciones para el cliente. Si un formulario de Windows requería gráficos 3D, video o audio, se necesitaba incorporar tecnología adicional, entonces surge *WPF (Windows Presentation Fundation)*. [Stoecker 14]

*WPF* es un marco de trabajo de interfaz de usuario que crea aplicaciones enriquecidas e interactivas visualmente. La plataforma de desarrollo de *WPF* admite un amplio conjunto de características de desarrollo de aplicaciones que incluye un modelo de aplicaciones, recursos, controles, gráficos, diseño, enlace de datos, documentos, seguridad y es un subconjunto del *.NET Framework*.

*WPF* permite la aceleración de los gráficos de la computadora, incrementando el rendimiento de la aplicación, está basado en gráficos vectoriales y es independiente a la resolución del dispositivo. Una de las ventajas de que sea basado en gráficos vectoriales es que cuando se cambia la resolución del dispositivo no se pierde calidad de los gráficos y los gráficos son portables en una gran cantidad de dispositivos para conservar siempre tamaño y calidad. Construir una aplicación con *WPF* es igual de sencillo que con *ASP.NET* o un *formulario de Windows*, solo se arrastran y pegan de una barra de herramientas los controles que sean necesarios para la interfaz y se manejan eventos similares a los *formularios de Windows*, como el clic del mouse, pero se pueden extender aún más.

*WPF* utiliza el lenguaje *XAML (eXtensible Application Markup Language, Lenguaje Extensible de Formato para Aplicaciones-Zammel)* para proporcionar un modelo declarativo para la programación de aplicaciones. *XAML* es un lenguaje declarativo basado en *XML*, optimizado para describir interfaces de usuarios visualmente atractivas desde el punto de vista gráfico.

Los elementos de *XAML* se interconectan con objetos en tiempo de ejecución para implementar la lógica de la aplicación, los atributos se conectan con propiedades o eventos de esos objetos definidos por el *.NET Framework*, ya que fue diseñado para soportar las clases y métodos de la plataforma de desarrollo *.NET*.

Una parte del marco de desarrollo de aplicaciones *.NET* que engloba el desarrollo de este tipo de aplicaciones es *Silverlight*. *Microsoft Silverlight* es una estructura para aplicaciones web que agrega nuevas funciones multimedia como la reproducción de videos, gráficos vectoriales, animaciones e interactividad, en forma similar a lo que hace *Adobe Flash*. El lenguaje para el desarrollo de este tipo de aplicaciones es *XAML* ya que está basado en *WPF*.

El subconjunto del *.NET Framework* que contiene *Silverlight* incluye *API's* para manejo de colecciones (arreglos, matrices), input/output (lectura y escritura de archivos), multihilo, bases de datos, *XML*, *LINQ*, *API's* para interacción con *JavaScript* [Goodman 24] y *API's* para almacenamiento local de datos para la creación de aplicaciones web robustas.

*Silverlight* es una implementación de la plataforma *.NET* y por lo tanto permite realizar aplicaciones en lenguajes como *C#, Visual Basic, IronRuby* y *Iron-Python*, esto permite a los desarrolladores *.NET* crear aplicaciones para los navegadores web. [Deitel 23]

*Silverlight* provee también soporte a aplicaciones en red, aplicaciones web que invocan servicios web y uso de otras tecnologías en red.

Al igual que *WPF*, *Silverlight* proporciona un potente modelo de enlace de datos que facilita la visualización de datos a objetos, colecciones, bases de datos, *XML* y los controles de interfaz gráfica de usuario.

En concreto, *XAML* puede inicializar objetos y establecer propiedades de los objetos con una estructura de lenguaje que muestra las relaciones jerárquicas entre los diversos objetos y con un sistema de tipos de respaldo que admite la extensión de los tipos.

En las aplicaciones *Silverlight* cuando se genera un formulario o *GUI*, el *IDE Visual Studio* genera sentencias de código para crear y configurar estos controles. El

lenguaje XAML está diseñado para ser interpretado o leído por la computadora o el humano de forma sencilla, también es posible la escritura directa del código para la generación de controles para la creación de GUI's.

XAML entonces entre sus principales ventajas permite a los diseñadores de la interfaz y programadores trabajar juntos eficientemente. Sin escribir el código, el diseñador de la interfaz puede crearla basándose en una herramienta de edición como *Microsoft's Expression Blend*—un programa de diseño gráfico XAML o en el mismo *IDE Visual Studio*.

Un programador puede importar el XAML dentro de su herramienta de desarrollo que es el *IDE Visual Studio* y así sólo enfocarse a la lógica y funcionalidad que la aplicación requiere utilizando patrones de diseño Modelo-Vista-Controlador o Modelo Vista-Vista Modelo.

Al trabajar con una arquitectura orientada a servicios, se puede estar creando toda la funcionalidad de la aplicación a nivel de servicios web, mientras paralelamente se está trabajando con la interfaz de la aplicación.

*Silverlight* se ejecuta en un navegador con un *Plug-in* (un complemento del navegador web). Los proyectos web *Silverlight* se dividen en 2 partes, en el proyecto *Silverlight* y el proyecto de la aplicación web.

- *Proyecto Silverlight*.
- Al crear proyecto *Silverlight* se generan 2 pares de archivos:
  - *MainPage.xaml*, a donde se contendrá la aplicación principal o diseño principal de la página web o formulario, con su respectiva lógica en un archivo *MainPage.xaml.cs* en C#. En *Silverlight* en el archivo XAML el elemento raíz es un *UserControl* (Control de Usuario), en él se contiene todo el diseño de la interfaz, y las cabeceras que se utilizan o referencias a los controles de usuario. Para ir añadiendo ventanas o más vistas se agregan más *UserControl* con sus respectivos archivos XAML (vista) y CS (lógica), y dependiendo del diseño y lógica de la aplicación se irán llamando o consumiendo los *UserControl*.
  - *App.xaml*, en este par de archivos se declaran los recursos compartidos y los GUI's de la aplicación, y el *app.xaml.cs*, se encarga de inicializar la aplicación, y ahí se definen los eventos y excepciones de la aplicación. Estos archivos pueden ser utilizados por todas la páginas de la aplicación o *UserControls*.
- Proyecto de la Aplicación web  
En la parte web del proyecto *Silverlight*, es a donde se pueden implementar los servicios web y crear consultas para bases de datos, además de poder tomar la lógica de una *Base de Datos* y pasarla a clases con el modelado de entidades y poder trabajar con ella como objetos con *LINQ*.

Al compilar una aplicación *Silverlight* el *IDE* empaqueta la información en un archivo *.xap* y sus recursos soportados (como imágenes o archivos utilizados por la aplicación), la página web que tiene la aplicación *Silverlight* hace referencia al archivo *.xap* y el *Plug-in* instalado en el navegador ejecuta la aplicación. La parte del proyecto web es la encargada de realizar todo esto (la conexión entre la página web al archivo *.xap* y el *Plug-In* de *Silverlight*), en una página generada en *aspx* mediante *JavaScript*.

### **Ventajas de uso de Silverlight.**

- Lenguaje potente basado en *WPF* y *.NET*.
- Visualmente atractivo e interactivo.
- Se puede ejecutar en todos los entornos: con exploradores y en múltiples dispositivos y sistemas operativos de escritorio.
- Fácil manejo entre lógica e interfaz gráfica.
- Aceleración por hardware mediante el uso de *DirectX*.
- Soporta llamadas asíncronas.
- Se puede realizar una aplicación completa, compleja y cuenta con la portabilidad gracias al *Plug-in*, además de ser fácil de desarrollar.

### **Desventajas.**

- No soportado en todos los navegadores, ni por la mayoría de los teléfonos inteligentes y Tabletas.
- Dependencia del *Plug-in* para que la aplicación pueda ser mostrada.
- La aplicación web es independiente del navegador soportado, pero es pesada por toda la interfaz gráfica que provee.
- Solo se puede desarrollar en la plataforma de *Windows*.

### **2.1.2 LINQ.**

Para realizar la conexión a bases de datos por medio de *Visual Studio*, se va a realizar por medio de *LINQ* (Language Integrated Query), es un proyecto de *Microsoft* que agrega consultas nativas semejantes a las de *SQL* a los lenguajes de la plataforma *.NET* como *C#*. [Piaorsi 22]

*LINQ* define operadores de consulta estándar que permiten a lenguajes habilitados filtrar, enumerar y crear proyecciones de varios tipos de colecciones usando la misma sintaxis. Tales colecciones pueden incluir vectores, clases *e-numerables*, *XML*, conjuntos de datos desde bases de datos relacionales y orígenes de datos de terceros.

El objetivo de crear *LINQ* es permitir que todo el código hecho en *Visual Studio* sea también orientado a objetos. Antes de *LINQ*, la manipulación de datos externos tenía un concepto más estructurado que orientado a objetos. Además trata de facilitar y estandarizar el acceso a dichos objetos.

*LINQ* permite obtener la información de una amplia variedad de orígenes de datos, no solo bases de datos (*Excel*, *XML*, etc.).

#### ***LINQ Providers***

La sintaxis de *LINQ* está construida en *C#*, pero las consultas *LINQ* pueden ser usadas en diferentes contextos por las librerías de los proveedores de datos. *LINQ Provider* (*Proveedor LINQ*) es un conjunto de clases que implementan operaciones *LINQ* y permiten a los programas interactuar con estas fuentes de datos para realizar tareas como ordenamiento, agrupamiento y filtro de datos o elementos. Estos proveedores están incluidos en *.NET Framework*.

#### ***LINQ a SQL.***

*LINQ a SQL* es un *LINQ Provider* que permite acceder a datos en *SQL Server* usando sentencias parecidas a *SQL*. La interacción se lleva a cabo por clases que son generadas automáticamente del esquema de la *Base de Datos* por el *IDE Visual Studio LINQ a SQL*.

La forma de pasar la estructura de la *Base de Datos* realizada con *SQL Server* al *IDE Visual Studio* se realiza de la siguiente manera: para cada tabla en la *Base de Datos* el IDE crea 2 clases:

- Una clase que representa un renglón en la tabla: Esta clase contiene propiedades para cada columna en la tabla. *LINQ a SQL* crea objetos de esta clase llamados *row objects* (Objetos del renglón) para almacenar los datos de manera individual en la tabla.
- La otra es una clase que representa la tabla: *LINQ a SQL* crea un objeto de esta clase para almacenar una colección de objetos renglón (*row objects*) que corresponden a todos los renglones de la tabla.

Las relaciones entre tablas son tomadas en cuenta en las clases generadas: En una clase objeto renglón, una propiedad adicional es creada para cada llave foránea. Esta propiedad regresa el objeto renglón que corresponde a la llave primaria en otra tabla.

### **Interfaz *IQueryable***

*LINQ a SQL* trabaja a través de una interfaz *IQueryable*, que hereda de la interfaz *IEnumerable* (expone el enumerador, que admite una iteración simple en una colección de un tipo especificado). Cuando una consulta *LINQ a SQL* se ejecuta de nuevo en la *Base de Datos* en un objeto *IQueryable*, el resultado es cargado dentro de objetos que corresponden a las clases para tener un acceso conveniente al código.

### **Clase *DataContext***

Todas las consultas *LINQ a SQL* son a través de la clase *DataContext*, que controla el flujo de datos entre el programa y la *Base de Datos*. Una clase específica *DataContext* derivada que hereda de la clase *System.Data.Linq.DataContext*, es creada cuando *LINQ a SQL* representa cada renglón de la tabla generada por el IDE. Esto deriva propiedades de clases para cada tabla de la *Base de Datos*, que permite utilizar el proveedor de datos en consultas *LINQ*.

Cualquier cambio realizado al *DataContext* puede ser salvado en la *Base de Datos* utilizando el método *DataContext's SubmitChanges*, y con *LINQ a SQL* se puede modificar el contenido de la *Base de Datos*.

## **2.2 Servicios web.**

Un servicio web es una colección de funciones que son empaquetadas como una sola entidad y son publicadas en la red para el uso de otros programas. Los servicios web son componentes que permiten crear sistemas distribuidos abiertos, de tal manera que las compañías e individuos pueden acceder rápidamente a los recursos digitales disponibles en la web.

Un servicio web simple está caracterizado por cuatro estándares: *XML (eXtended Markup Language)*, *SOAP (Simple Object Access Protocol)*, *WSDL (Web Services Definition Language)* y *UDDI (Universal Discovery, Description and Integration)*, los cuales al trabajar juntos proporcionan una funcionalidad básica de tipo "solicitud/respuesta".

## 2.2.1 XML.

Fue desarrollado en 1996 por el *World Wide Web Consortium's* (W3C's). *XML* es un lenguaje robusto, estándar para la descripción de datos, y es comúnmente utilizado para el intercambio de datos entre aplicaciones sobre el internet.

Cantidades masivas de datos son almacenados a través del internet en bases de datos, páginas web, documentos de texto, es por esto la importancia de *XML* ya que al estar almacenada la información en diversos contenedores de datos, codificada con distintos lenguajes de programación, *XML* provee de un formato estándar para la interacción de información en los contenedores. Para los servicios web, se permite el intercambio de datos a través de protocolos estándar basados en *XML*. También la descripción de los servicios es realizada con un vocabulario *XML* llamado *WSDL*. La siguiente generación de Internet y del *World Wide Web* está fundada en *XML*, que permite el desarrollo de sofisticadas aplicaciones basadas en web.

*XML* es una clase de objetos de datos llamados documentos *XML*, que describe parcialmente el comportamiento de los programas que los procesan. Los documentos *XML* están formados por unidades de almacenamiento llamadas entidades, las cuales contienen datos analizados (*parsed*) o sin analizar (*unparsed*). Los documentos analizados están formados por caracteres, algunos son datos y otros son etiquetas.

### [Arboleda 3]

Aunque esta definición es un poco compleja por la cantidad de elementos involucrados, es importante resaltar que *XML* permite representar documentos (documento, se refiere no sólo a textos, sino también gráficos, correos electrónicos, transacciones de comercio electrónico, etc.) y que todo documento *XML* posee una estructura lógica y una física.

*XML* permite a los autores de los documentos crear marcado para hacer cualquier tipo de información virtual. Esto permite crear un nuevo lenguaje de marcado para describir cualquier tipo de dato, como fórmulas matemáticas, instrucciones para configuración de software, estructuras moleculares químicas, música, noticias, recetas y reportes financieros.

*XML* describe datos con la finalidad de ser entendibles por los humanos y las computadoras (Figura 2.3).

```
1 <?xml version = "1.0"?>
2 <!-- Fig. 24.1: player.xml -->
3 <!--Estructura de una persona en XML -->
4
5 <Persona>
6   <Nombre>José</Nombre >
7
8   <Apellido>Pérez</Apellido >
9
10  <Edad>35</Edad>
11 </Persona>
```

**Figura 2.3.** Ejemplo de archivo XML, describe la información básica acerca de una persona: nombre y edad.

Los documentos *XML* contienen texto que representa datos, como en el ejemplo anterior en la línea 6 se representa el nombre de una persona, estos elementos especifican la estructura del documento. Los documentos *XML* delimitan elementos

con etiquetas iniciales y finales. Los elementos contenidos dentro de las etiquetas iniciales y finales representan una pieza de datos, por ejemplo el nombre de la Persona José, esta contenido dentro de <Nombre> y </Nombre>, o Pérez en <Apellido> y </Apellido >.

Todos los documentos XML deben tener un elemento raíz que contenga los otros elementos, como en el ejemplo anterior <Persona> y </Persona> es el elemento raíz.

### **Ver y modificar documentos XML.**

Los documentos XML son portables. Para ver o modificar un documento XML se puede realizar desde un editor de texto con la extensión .xml, no requiere de algún software, existen muchas herramientas, por ejemplo el IDE Visual Studio provee de un editor con IntelliSense, este editor verifica el esquema y sintaxis del documento.

### **Procesamiento de documentos XML.**

Para procesar un documento XML se requiere de un software llamado XML parser (o XML processor), el analizador o parser hace al documento de datos disponible para las aplicaciones, verifica las reglas de sintaxis especificadas por W3C's XML Recommendation ([www.w3.org/XML](http://www.w3.org/XML)).

La sintaxis XML requiere un simple elemento raíz, una etiqueta de inicio y una de final para cada elemento como en el ejemplo mostrado anteriormente.

### **Validación de documentos XML.**

Un documento XML puede opcionalmente referenciar a un Document Type Definition (DTD) o a un esquema XML W3C que define una estructura adecuada al documento XML. Cuando un documento XML referencia a un DTD o a un esquema (esquema XML W3C), algunos parsers pueden utilizar el DTD/schema para validar que tenga una estructura adecuada, si el documento XML está conformado por el DTD/schema entonces el documento XML es válido.

### **Formato y manipulación de los documentos XML**

Un documento XML contiene solo datos, no instrucciones de formato, entonces las aplicaciones que procesan documentos XML deciden como manipular o mostrar cada documento. Por ejemplo un PDA (Personal Digital Assistant) puede representar un documento XML diferente para un teléfono celular o una PC. Se pueden utilizar Extensible Stylesheet Language (XSL) para especificar la representación de instrucciones para las diferentes plataformas.

Los programas que procesan XML pueden también buscar, ordenar y manipular datos XML utilizando tecnologías como XSL, otras tecnologías relacionadas son XPath (XML Path Language—un lenguaje para acceder a partes de un documento XML), XSL-FO (XSL Formatting Objects—Un Vocabulario XML utilizado para describir el formato de los documentos XML) y XSLT (XSL Transformations— Un lenguaje para transformar documentos XML en otros documentos). [Löwy 15]

## **2.2.2 SOAP.**

El protocolo SOAP [SOAP 9] utiliza mensajes XML para invocar métodos remotos. Un servicio web podría interactuar con servicios remotos a través de los métodos Get y Post de http, pero SOAP es mucho más robusto y flexible. Es un protocolo liviano, para el intercambio de información estructurada en un ambiente descentralizado y distribuido. Sin embargo, SOAP no define la aplicación, ni la semántica de implementación. En vez de esto, proporciona un modelo de empaquetamiento modular y los mecanismos para la codificación de los datos dentro de los módulos. Esto

permite que el protocolo se utilice en una amplia variedad de sistemas modulares y que cumpla su propósito primario de facilitar la interoperabilidad entre componentes de software heterogéneos. *SOAP* es una especificación para la invocación de métodos en servidores, servicios, componentes, objetos y codifica la práctica existente de utilizar *XML* y *http* como un mecanismo de invocación de métodos, la estructura del mensaje se muestra a continuación (Figura 2.4).

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>
  ...
  <soap:Fault>
  ...
  </soap:Fault>
  </soap:Body>

</soap:Envelope>
```

**Figura 2.4.** Ejemplo SOAP. Estructura de un mensaje SOAP.

#### **Reglas importantes para un mensaje SOAP.**

- Un mensaje *SOAP* debe estar codificado utilizando *XML*
- Un mensaje *SOAP* debe usar el espacio de nombres *SOAP Envelope*: es la raíz de un mensaje *SOAP* ya que define en el documento *XML* que es un mensaje *SOAP*.
- Un mensaje *SOAP* debe usar el espacio de nombres *SOAP Encoding*
- Un mensaje *SOAP* no debe contener referencias *DTD*
- Un mensaje *SOAP* no debe contener instrucciones de procesamiento *XML*

#### **Partes del mensaje SOAP.**

##### **Header.**

La cabecera *SOAP* o *header*, contiene información específica de la aplicación, por ejemplo la autenticación del mensaje *SOAP*.

Si el encabezado o *header* está presente debe ser el primer hijo del elemento *Envelope*.

##### **Body.**

Contiene el mensaje *SOAP*.

En las siguientes figuras se muestra un ejemplo de la solicitud (Figura 2.5) y respuesta (Figura 2.6).

```

SOAP Request.
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
<m:GetStockPrice>
<m:StockName>IBM</m:StockName>
</m:GetStockPrice>
</soap:Body>

</soap:Envelope>

```

**Figura 2.5.** En este ejemplo (Request o Solicitud), en el mensaje SOAP se quiere obtener al precio del artículo IBM con la función `GetStockPrice` en el Servicio web.

```

SOAP Response.
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

<soap:Body xmlns:m="http://www.example.org/stock">
<m:GetStockPriceResponse>
<m:Price>34.5</m:Price>
</m:GetStockPriceResponse>
</soap:Body>

</soap:Envelope>

```

**Figura 2.6.** En este ejemplo (Response o Respuesta), en el mensaje SOAP se regresa el resultado obtenido del servicio o función `GetStockPriceResponse`.

### 2.2.3 WSDL.

Para consumir un servicio web, el cliente debe determinar la funcionalidad del servicio y cómo usarlo, con este propósito los servicios web contienen una descripción. *WSDL* es el lenguaje común utilizado para la descripción de los servicios web [WSDL 10], está basado en *XML* donde describe totalmente la forma en la cual los clientes externos pueden interactuar con los servicios web existentes en un servidor, los métodos que soportan y la sintaxis de los protocolos de comunicación (*http*, *SOAP*). En términos generales, un documento *WSDL* contiene información acerca de la interfaz, la semántica y los aspectos administrativos involucrados en una solicitud

(llamado) a un servicio web, a continuación se muestra la definición de los tipos de elementos (Figura 2.7) y la estructura del documento (Figura 2.8).

<i>Elemento</i>	<i>Define</i>
<types>	Tipo de datos utilizado por el servicio web.
<message>	Mensaje utilizado por el servicio web, define el tipo de dato utilizado por cada operación. Cada mensaje puede consistir de una o más partes.
<portType>	Al servicio web, las operaciones que puede realizar y los mensajes que involucra.
<binding>	El formato del mensaje y Protocolo de comunicación utilizado por el Servicio web.

**Figura 2.7.** Definición de los tipos de elementos de la estructura del WSDL.

```
<definitions>
<types> Definición de tipos..... </types>
<message> Definición de mensaje..... </message>
<portType> Definición de un puerto..... </portType>
<binding> Definición del protocolo..... </binding>
</definitions>
```

**Figura 2.8.** Estructura del WSDL.

```
<message name="getTermResponse">
<part name="value" type="xs:string"/>
</message>

<portType name="glossaryTerms">
<operation name="getTerm">
<input message="getTermRequest"/>
<output message="getTermResponse"/>
</operation>
</portType>
```

**Figura 2.9.** Ejemplo WSDL.

En el ejemplo anterior (Figura 2.9) el elemento <portType> define "glossaryTerms" como el nombre del puerto, y "getTerm" es el nombre de la operación.

La operación "getTerm" tiene un mensaje de entrada llamado "getTermRequest" y un mensaje de salida "getTermResponse".

El elemento <message> define las partes "part" de cada mensaje y la asociación de cada tipo de dato.

Comparando con la programación tradicional, "glossaryTerms" es una librería, "getTerm" es una función donde "getTermRequest" es el parámetro de entrada, y "getTermResponse" el parámetro devuelto o de salida.

## 2.2.4 UDDI.

UDDI para los servicios web es el mecanismo por medio del cual los objetos de negocio acuerdan lo que van a hablar unos a otros, SOAP indica cómo van a efectuar la comunicación y UDDI les permite conocer con quién se va a comunicar y dónde encontrar otros objetos de negocio [UDDI 11]. En pocas palabras, UDDI define la

forma en la cual se publica y descubre información acerca de los servicios web. El método utilizado por *UDDI* para el descubrimiento de servicios es tener un registro de aquellos servicios que se encuentran distribuidos a través del web. En el registro, los negocios y los servicios se describen utilizando un formato *XML* común. Los datos estructurados en esos documentos *XML* son de fácil búsqueda, análisis y manipulación. Debe mencionarse que *UDDI/SOAP* no son los únicos modelos para el descubrimiento y manejo de mensajes en los servicios web, por ejemplo *EbXML* ha sido desarrollado para un fin similar, así como para proporcionar una interfaz de negocio, seguridad robusta y otras funciones que permiten el manejo de transacciones reales de *e-business*. *EbXML* y *UDDI/SOAP* pueden ser tecnologías complementarias.

*UDDI* incluye un esquema *XML* conformado por:

- businessEntity
- businessService
- bindingTemplate
- tModel
- publisherAssertion

***businessEntity***.

La estructura *businessEntity* representa la información del proveedor del servicio web, información acerca de la compañía, contacto, etc., (Figura 2.10).

```

<businessEntity businessKey="uuid:C0E6D5A8-C446-4f01-99DA-70E212685A40"
  operator="http://www.ibm.com"
  authorizedName="Nahum">
<name>BUAP Company</name>
<description>
  Nosotros creamos servicios web escolares
</description>
<contacts>
<contact useType="general info">
<description>General Information</description>
<personName>Nahum</personName>
<phone>(123) 123-1234</phone>
<email>nahumfcc@info.com</email>
</contact>
</contacts>
<businessServices>
...
</businessServices>
<identifierBag>
<keyedReference
  tModelKey="UUID:8609C81E-EE1F-4D5A-B202-3EB13AD01823"
  name="D-U-N-S"
  value="123456789" />
</identifierBag>
<categoryBag>
<keyedReference
  tModelKey="UUID:C0B9FE13-179F-413D-8A5B-5004DB8E5BB2"
  name="NAICS"
  value="111336" />
</categoryBag>
</businessEntity>

```

**Figura 2.10.** Ejemplo de *businessEntity*.

### **businessService.**

La estructura representa individualmente los servicios web que son proveídos por *businessEntity*. Esta descripción incluye información de cómo se puede consumir el servicio web, de que tipo es (Figura 2.11).

```
<businessService serviceKey="uuid:D6F1B765-BDB3-4837-828D-8284301E5A2A"
businessKey="uuid:C0E6D5A8-C446-4f01-99DA-70E212685A40">
  <name>Hello World Web Service</name>
  <description>Un web service de ejemplo</description>
  <bindingTemplates>
    ...
  </bindingTemplates>
  <categoryBag />
</businessService>
```

**Figura 2.11.** Ejemplo de *businessService*.

### **bindingTemplate.**

Es la técnica de descripción del servicio web representada por la estructura *businessService*. Un *businessService* puede tener múltiples plantillas (Figura 2.12).

```
<bindingTemplate serviceKey="uuid:D6F1B765-BDB3-4837-828D-8284301E5A2A"
bindingKey="uuid:C0E6D5A8-C446-4f01-99DA-70E212685A40">
  <description>Hola Mundo SOAP Binding</description>
  <accessPoint URLType="http">
    http://localhost:8080
  </accessPoint>
  <tModelInstanceDetails>
    <tModelInstanceInfo
      tModelKey="uuid:EB1B645F-CF2F-491f-811A-4868705F5904">
      <instanceDetails>
        <overviewDoc>
          <description>
            Referencia a la descripción del
            WSDL
          </description>
          <overviewURL>
            http://localhost/helloworld.wsdl
          </overviewURL>
        </overviewDoc>
      </instanceDetails>
    </tModelInstanceInfo>
  </tModelInstanceDetails>
</bindingTemplate>
```

**Figura 2.12.** Ejemplo de *bindingTemplate*.

### **tModel.**

*tModel* es el último núcleo de tipo de datos, es la opción para describir varios negocios, servicios y estructuras de datos guardadas sin un registro *UDDI*.

Cualquier concepto abstracto puede ser registrado sin un *UDDI* como un *tModel*. Si se define un nuevo tipo de puerto *WSDL*, se puede definir un modelo *tModel* que represente el tipo de puerto sin *UDDI*. Entonces, se puede especificar cómo se va a

implementar el servicio de negocio que tipo de puerto va a ser asociado al *tModel* con una de las plantillas del *businessService*, a continuación se muestra un ejemplo simple de un *tModel* (Figura 2.13).

```
<tModel tModelKey="uuid:xyz987..."
  operator="http://www.ibm.com"
  authorizedName="John Doe">
<name>HelloWorldInterface Port Type</name>
<description>
Una vista de un Web service
</description>
<overviewDoc>
<overviewURL>
  http://localhost/helloworld.wsdl
</overviewURL>
</overviewDoc>
</tModel>
```

**Figura 2.13.** Ejemplo de *tModel* que representa el *HelloWorldInterface port type*.

#### ***publisherAssertion data structure.***

Es una estructura relacional, que asociada con dos o más estructuras *businessEntity* acuerdan un tipo específico de relación, como filial o departamento.

La estructura *publisherAssertion* consiste de tres elementos *fromKey* (la primer *businessKey*), *toKey* (el segundo *businessKey*) y *keyedReference*.

El *keyedReference* designa el tipo de relación en términos de *keyName* *keyValue* dentro de un *tModel*, únicamente hace referencia a un *tModelKey*.

Un *UDDI publisherAssertion*, permite a una organización publicar información acerca de las relaciones entre múltiples archivos *UDDI businessEntity*. El *UDDI publisherAssertion* es particularmente usado para organizaciones grandes que deben permitir a diferentes partes de la organización operar los servicios de manera independiente, a continuación se muestra la estructura (Figura2.14). **[UDDI 05]**

```
<element name="publisherAssertion" type="uddi:publisherAssertion" />
<complexType name="publisherAssertion">
<sequence>
<element ref="uddi:fromKey" />
<element ref="uddi:toKey" />
<element ref="uddi:keyedReference" />
</sequence>
</complexType>
```

**Figura 2.14.** Estructura de *publisherAssertion*.

## **3 Sistema gestor de calificaciones.**

### **3.1 Justificación.**

Los profesores de la *FCC* requieren contar con un sistema web flexible que les permita registrar y administrar sus grupos, de tal forma, que puedan adaptar sus listas de calificaciones de acuerdo a las diversas necesidades de los grupos y de la materias que imparten, necesitan dar de alta alumnos, insertar o eliminar columnas de calificaciones parciales, asistencias, tareas, presentaciones, proyectos y otro tipo de

calificaciones, también obtener un promedio final y una serie de reportes que le sean útiles para llevar un control del aprovechamiento de sus alumnos y diversas gestiones académicas que tiene que llevar a cabo en la FCC. Éste sistema debe ser accesible en la web, sin importar el sistema operativo sobre el cual trabaje.

### 3.2 Análisis de requisitos.

El sistema a desarrollar, va a ser capaz de guardar listas en una *Base de Datos* para su consulta o modificación en cualquier lugar con acceso a internet. Para consultar las listas o subirlas, el usuario, en este caso profesor, podrá acceder al sistema con su número de trabajador y una contraseña previamente dada de alta en la *Base de Datos*, se autenticara, si la autenticación fue exitosa, el profesor, podrá consultar sus datos personales, como nombre, apellidos, correo electrónico y podrá cambiarlos, así como también podrá cambiar su contraseña de acceso al sistema, esta contraseña por seguridad será codificada y guardada en una *Base de Datos*, en caso de que el profesor olvide su contraseña tendrá la opción de recuperarla a su correo electrónico capturado inicialmente en la *Base de Datos*.

Dentro del sistema, el profesor podrá cargar listas de archivos *Excel* proporcionados por la Secretaría Académica, una vez subida la lista de calificaciones, se asignara el porcentaje que tendrán las calificaciones del grupo a evaluar, así como el tipo de calificaciones que se van a agregar como: exámenes, tareas, exposiciones, proyectos y otras calificaciones. El formato que deberán tener los archivos de Excel que contengan las listas de calificaciones se muestra en la siguiente figura (Figura 3.1).

1	Registro	Nombre Alumno	ID	Status Reg	Nivel	Créditos	Parcial	Final	Detalle de Calificaciones
2	1	AGUILAR AVENDAÑO, ALEJANDRA	200700321	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
3	2	ANGELES HUERTA, IVETH	200606401	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
4	3	APONTE AGUILAR, OSCAR G.	200508575	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
5	4	ARROYO CARMONA, MISAEL	200632376	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
6	5	BAEZ DE LA CRUZ, VICTOR	200516774	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
7	6	BAZQUEZ OCOTITLA, MARCOS	200529535	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
8	7	CABALLERO JUÁREZ, DEISY	200609598	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
9	8	CARRERA ARRIAGA, JOEL	200518491	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
10	9	CORONA PEREZ, XELHUA J.	200511359	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
11	10	CORTÉS HERNÁNDEZ, FERNANDO.	200526332	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
12	11	FERNANDEZ TORRES, LUIS A.	200612199	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
13	12	GOYRI AGUIRRE, OSCAR D.	200619115	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
14	13	GUTIERREZ JIMENEZ, UBALDO	200609504	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
15	14	JIMENEZ HERRERA, CALEB	200519195	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
16	15	LEDESMA PUENTE, MARIA M.	200612453	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
17	16	LUNA PEREZ, MAURICIO	200632297	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
18	17	MARTINEZ HERRERA, ALEJANDRO U.	200607806	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
19	18	MARTÍNEZ VALERA, VICTOR H.	200523538	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
20	19	MIRANDA FRANCO, MAURICIO	200512835	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
21	20	MIRANDA PEREZ, EDUARDO	200729516	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
22	21	ORTEGA RIVERA, GISELA R.	200606175	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
23	22	PAREDES VILLEGAS, HEBER	200620133	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
24	23	PEREZ CRUZ, JOSE R.	200606339	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
25	24	PEREZ GARCIA, ALICIA	200601204	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
26	25	PUENTE TERCERO, JESUS	200606337	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
27	26	RAMOS CRUZ, FERNANDO	200601588	**Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	

Figura 3.1. Formato del Archivo en Excel de la lista de calificaciones.

La carga de listas puede ser por medio de copiado y pegado de información, con el siguiente formato (Figura 3.2).

	A	B	C	D	E
2	200700321	AGUILAR AVENDAÑO, ALEJANDRA	Licenciatura Créditos	10	
3	200606401	ANGELES HUERTA, IVETH	Licenciatura Créditos	10	
4	200508575	APONTE AGUILAR, OSCAR G.	Licenciatura Créditos	10	
5	200632376	ARROYO CARMONA, MISAEI	Licenciatura Créditos	10	
6	200516774	BAEZ DE LA CRUZ, VICTOR	Licenciatura Créditos	10	
7	200525935	BAZQUEZ OCOTITLA, MARCOS	Licenciatura Créditos	10	
8	200606998	CABALLERO JUÁREZ, DEISY	Licenciatura Créditos	10	
9	200618491	CARRERA ARRIAGA, JOEL	Licenciatura Créditos	10	
10	200513599	CORONA PEREZ, XELHUA J.	Licenciatura Créditos	10	
11	200526332	CORTÉS HERNÁNDEZ, FERNANDO	Licenciatura Créditos	10	
12	200612199	FERNANDEZ TORRES, LUIS A.	Licenciatura Créditos	10	
13	200619115	GOYRI AGUIRRE, OSCAR D.	Licenciatura Créditos	10	
14	200606504	GUTIERREZ JIMENEZ, UBALDO	Licenciatura Créditos	10	
15	200519195	JIMENEZ HERRERA, CALEB	Licenciatura Créditos	10	
16	200612453	LEDESMA PUENTE, MARIA M.	Licenciatura Créditos	10	
17	200633297	LUNA PEREZ, MAURICIO	Licenciatura Créditos	10	
18	200607806	MARTINEZ HERRERA, ALEJANDRO U.	Licenciatura Créditos	10	
19	200523528	MARTÍNEZ VALERA, VICTOR H.	Licenciatura Créditos	10	
20	200512835	MIRANDA FRANCO, MAURICIO	Licenciatura Créditos	10	
21	200729616	OLIVERA PEREZ, EDUARDO	Licenciatura Créditos	10	
22	200606175	ORTEGA RIVERA, GISELA R.	Licenciatura Créditos	10	
23	200620133	PAREDES VILLEGAS, HEBER	Licenciatura Créditos	10	
24	200606339	PEREZ CRUZ, JOSE R.	Licenciatura Créditos	10	
25	200601204	PÉREZ GARCIA, ALICIA	Licenciatura Créditos	10	
26	200606337	PUENTE TERCERO, JESUS	Licenciatura Créditos	10	
27	200601598	RAMOS CRUZ, FERNANDO	Licenciatura Créditos	10	
28	200515827	ROJAS YARCE, FELIPE A.	Licenciatura Créditos	10	
29	200628799	ROLDAN ARAUJO, MARTIN	Licenciatura Créditos	10	
30	200502209	SALGADO PARRA, RENE	Licenciatura Créditos	10	
31	200634349	SANCHEZ MARTINEZ, LUIS G.	Licenciatura Créditos	10	
32	200503101	SANCHEZ MELENDEZ, CARLOS J.	Licenciatura Créditos	10	
33	200702583	SANCHEZ VELAZQUEZ, ARIANA P.	Licenciatura Créditos	10	

**Figura 3.2.** Formato del Archivo para copiar y pegar la lista de calificaciones.

Una vez cargada una materia, con su *NRC* que es único, un nombre, sección y periodo a evaluar, se podrá realizar la calificación del grupo de acuerdo al criterio establecido por el profesor o acordado con los alumnos del grupo. En la lista de calificaciones el profesor podrá agregar algún tipo de calificación, o quitarla, como máximo se contara con 4 espacios por tipo de calificación, por ejemplo: *examen 1*, *examen 2*, *examen 3* y *examen 4*. Al final de la lista de calificaciones (columna final) se encontrara la calificación final calculada de acuerdo a los criterios establecidos al cargar la lista inicial (porcentaje). Se podrá actualizar la lista de calificaciones cuantas veces se desee por parte del profesor, además de tener la opción de exportarla a un nuevo documento de *Excel*, con un formato distinto al que proporciona la Secretaría Académica y que no se podrá cargar nuevamente en el sistema. Además, se tendrá la opción de Imprimir la lista de calificaciones con el promedio de cada tipo de calificación y el promedio final, el reporte será con *Crystal Report* y la exportación de la lista con formato *.pdf*. También se tendrá la opción de borrar este grupo o lista de calificaciones sin opción a recuperarlo, si se desea recuperar será necesario repetir todo el proceso (subir el archivo de *Excel* proporcionado por la Secretaría Académica (Figura 3.1) o copiar y pegar la lista en el formato establecido (Figura 3.2), asignar tipo de calificación, porcentajes y calificar el grupo).

Además de estas características el sistema contara con dos tipos de cuentas de usuario, la de *Administrador* y la de *Usuario*. La cuenta de *Usuario* es la que ya fue descrita y la de *Administrador*, que además de tener las mismas características que la de un *Usuario* también tiene el privilegio de poder insertar nuevos usuarios o borrarlos, la inserción puede ser de manera grupal a través de un archivo *Excel* con los datos de los profesores proporcionado por la Secretaría Académica (Figura 3.3), o con la misma dinámica de copiar y pegar en un formato con el siguiente orden: número de trabajador, nombre, apellido paterno, apellido materno, correo electrónico (Figura 3.4) o de manera individual, la baja de profesores del sistema se realiza de manera individual.

Para asignar el tipo de usuario si se hace de manera grupal, todos tendrán el rol de *Usuario*, si es de manera individual se puede asignar el rol de *Administrador* o *Usuario*.

	A	B	C	D	E	F	G	H
1	ID		NOMBRE		Dirección de Correo FCC		Dirección de Correo Externa	
2	1	100398900	ADOLFO	AGUILAR	RICO		<a href="mailto:adolforico2@hotmail.com">adolforico2@hotmail.com</a> <a href="mailto:adolforico2@gmail.com">adolforico2@gmail.com</a>	
3	2	100223144	LUIS CARLOS	ALTAMIRANO	ROBLES	<a href="mailto:altamirano@cs.buap.mx">altamirano@cs.buap.mx</a>	<a href="mailto:LCARLOS6@prodigy.net.mx">LCARLOS6@prodigy.net.mx</a>	
4	3	100203233	ALMA DELIZ	AMBROSIO	VAZQUEZ	<a href="mailto:ambrosio@cs.buap.mx">ambrosio@cs.buap.mx</a>		
5	4	100210511	MARIO	ANZURES	GARCIA	<a href="mailto:anzures@cs.buap.mx">anzures@cs.buap.mx</a>	<a href="mailto:mario_anzures@hotmail.com">mario_anzures@hotmail.com</a> <a href="mailto:manzures@siu.buap.mx">manzures@siu.buap.mx</a>	
6	5	100238799	ETELVINA	ARCHUNDIA	SIERRA	<a href="mailto:etelvina@cs.buap.mx">etelvina@cs.buap.mx</a>		
7	6	100378566	EDUARDO	ARIZA	VELAZQUEZ	<a href="mailto:eariza@cs.buap.mx">eariza@cs.buap.mx</a>		
8	7	100003911	APOLONIO	ATA	PEREZ	<a href="mailto:apolonio@cs.buap.mx">apolonio@cs.buap.mx</a>		
9	8	100471499	IRENE OLAYA	AYAQUICA	MARTINEZ	<a href="mailto:ayaquica@cs.buap.mx">ayaquica@cs.buap.mx</a>	<a href="mailto:ayaquica@gmail.com">ayaquica@gmail.com</a>	
10	9	100442133	VERONICA EDITH	BAUTISTA	LOPEZ	<a href="mailto:vbautista@cs.buap.mx">vbautista@cs.buap.mx</a>	<a href="mailto:veromaster@gmail.com">veromaster@gmail.com</a>	
11	10	100178444	CESAR	BAUTISTA	RAMOS	<a href="mailto:bautista@cs.buap.mx">bautista@cs.buap.mx</a>		
12	11	100299377	PEDRO	BELLO	LOPEZ	<a href="mailto:pbello@cs.buap.mx">pbello@cs.buap.mx</a>		
13	12	100239644	BEATRIZ	BELTRAN	MARTINEZ	<a href="mailto:bbeltran@cs.buap.mx">bbeltran@cs.buap.mx</a>		
14	13	100076511	MARIA BLANCA DEL CARMEN	BERMUDEZ	JUAREZ	<a href="mailto:bbj@cs.buap.mx">bbj@cs.buap.mx</a>		

Figura 3.3. Formato del Archivo en Excel de la plantilla de profesores.

	A	B	C	D	E
1	100203233	ALMA DELIZ	AMBROSIO	VAZQUEZ	<a href="mailto:ambrosio@cs.buap.mx">ambrosio@cs.buap.mx</a>
2	100210511	MARIO	ANZURES	GARCIA	<a href="mailto:anzures@cs.buap.mx">anzures@cs.buap.mx</a>
3	100238799	ETELVINA	ARCHUNDIA	SIERRA	<a href="mailto:etelvina@cs.buap.mx">etelvina@cs.buap.mx</a>
4	100378566	EDUARDO	ARIZA	VELAZQUEZ	<a href="mailto:eariza@cs.buap.mx">eariza@cs.buap.mx</a>
5	100003911	APOLONIO	ATA	PEREZ	<a href="mailto:apolonio@cs.buap.mx">apolonio@cs.buap.mx</a>
6	100471499	IRENE OLAYA	AYAQUICA	MARTINEZ	<a href="mailto:ayaquica@cs.buap.mx">ayaquica@cs.buap.mx</a>
7	100442133	VERONICA EDITH	BAUTISTA	LOPEZ	<a href="mailto:vbautista@cs.buap.mx">vbautista@cs.buap.mx</a>
8	100178444	CESAR	BAUTISTA	RAMOS	<a href="mailto:bautista@cs.buap.mx">bautista@cs.buap.mx</a>
9	100299377	PEDRO	BELLO	LOPEZ	<a href="mailto:pbello@cs.buap.mx">pbello@cs.buap.mx</a>
10	100239644	BEATRIZ	BELTRAN	MARTINEZ	<a href="mailto:bbeltran@cs.buap.mx">bbeltran@cs.buap.mx</a>
11	100076511	MARIA BLANCA DEL CARMEN	BERMUDEZ	JUAREZ	<a href="mailto:bbj@cs.buap.mx">bbj@cs.buap.mx</a>
12	100207200	MARIA BEATRIZ	BERNABE	LORANCA	<a href="mailto:beti@cs.buap.mx">beti@cs.buap.mx</a>
13	100430844	ERIKA	BONFIL	BARAGAN	<a href="mailto:erika.bonfil@cs.buap.mx">erika.bonfil@cs.buap.mx</a>
14	100074766	MARIA DEL ROCIO	BOONE	ROJAS	<a href="mailto:rboone@cs.buap.mx">rboone@cs.buap.mx</a>
15	100005888	MARIO MAURICIO	BUSTILLO	DIAZ	<a href="mailto:bustillo@cs.buap.mx">bustillo@cs.buap.mx</a>
16	100470544	JOSE LUIS	CARBALLID O	CARRANZA	<a href="mailto:jcarballedo@cs.buap.mx">jcarballedo@cs.buap.mx</a>
17	100299344	MAYA	CARRILLO	RUIZ	<a href="mailto:cmaya@cs.buap.mx">cmaya@cs.buap.mx</a>
18	100238122	HILDA	CASTILLO	ZACATELCO	<a href="mailto:hilda@cs.buap.mx">hilda@cs.buap.mx</a>
19	100036755	MAURICIO	CASTRO	CARDONA	<a href="mailto:mcastro@cs.buap.mx">mcastro@cs.buap.mx</a>
20	100195599	CARMEN	CERON	GARNICA	<a href="mailto:mceron@cs.buap.mx">mceron@cs.buap.mx</a>
21	100150822	ANA PATRICIA	CERVANTE S	MARQUEZ	<a href="mailto:patfy@cs.buap.mx">patfy@cs.buap.mx</a>
22	100010677	ELSA	CHAVIRA	MARTINEZ	<a href="mailto:echavira@cs.buap.mx">echavira@cs.buap.mx</a>

Figura 3.4. Formato del Archivo para copiar y pegar la plantilla de profesores.

Además el usuario *Administrador*, tendrá permisos para dar de alta nuevos periodos en los cuales se encontraran los cursos.

Para cualquier movimiento de alta, baja o cambio en la *Base de Datos*, se podrá hacer solo con una llave codificada creada al ingresar al sistema, ya que al estar estas funciones en servicios web podrán ser utilizados por cualquier persona y es necesario contar con seguridad.

### 3.2.1 Ámbito del sistema.

El sistema será capaz de permitir a un profesor ingresar al sistema a través de un *login*. Para que el profesor pueda ingresar, deberá estar previamente dado de alta en

la *Base de Datos*, el alta o baja de profesores se realizara a través de los usuarios *Administrador*.

Una vez que el profesor ingrese al sistema puede acceder a un menú con las siguientes opciones:

*Actualizar datos*: se puede realizar la actualización de los datos generales del profesor que ingreso al sistema y solo se pueden actualizar: nombre o nombres, apellido paterno, apellido materno, correo electrónico y también la contraseña pero se mencionara posteriormente, el número de trabajador que identifica a cada profesor no se puede cambiar.

*Cambiar contraseña*: se podrá cambiar la contraseña previamente guardada en la *Base de Datos*, será necesario ingresar la contraseña actual, escribir la nueva contraseña, confirmar la nueva contraseña y posteriormente se actualizara en la *Base de Datos*.

*Cargar grupo*: en esta opción del menú, se podrán cargar archivos en formato *Excel* en un formato predeterminado por la Secretaría Académica (Figura 3.1), también se podrá copiar y pegar la lista de alumnos en el formato previamente establecido (Figura 3.2), si el archivo *Excel* esta en otro formato, o el texto pegado no está en el formato establecido el sistema lanzara una excepción.

La lista que se va a cargar se guardara en la *Base de Datos*, se guardara el *NRC*, la sección, el nombre del grupo, el periodo al que pertenece y la lista de alumnos, la cual constara de matrícula y nombre, si el alumno existe no se insertara, si aún no está dado de alta en la *Base de Datos*, se cargaran sus datos en la tabla correspondiente. Una vez cargada una lista de calificaciones no se podrá cambiar la información de los alumnos, ni eliminar, solo se podrá eliminar el grupo, pero no el alumno. Una vez que se cargue el grupo en la *Base de Datos*, se tendrán que asignar los porcentajes con los cuales se calificará el grupo y también el tipo de calificación con el que se evaluará.

*Calificar grupo*: en esta opción se podrá elegir algún grupo correspondiente al profesor que se encuentra utilizando el sistema, una vez que se elija el grupo, se puede calificar, añadir o quitar columnas de calificaciones correspondientes a los porcentajes previamente cargados. Si no se ha asignado el porcentaje al grupo que se quiere calificar el sistema re direccionara a la pantalla correspondiente para asignar el porcentaje, una vez que se actualice el porcentaje a cien por ciento podrá evaluarse el grupo. Una vez calificado el grupo o antes (solo para visualización de la lista) el sistema permite imprimir la lista de calificaciones o guardarla en un archivo *Excel* que no puede ser subido al sistema como otro grupo, la lista que se puede imprimir no cambiará de formato.

*Administrar usuarios*: es solo valido para el usuario *Administrador*, en esta opción podrá dar de alta o de baja usuarios, asignándoles un rol o tipo de usuario para el ingreso al sistema. El alta de usuarios puede ser por grupo, donde todos los usuarios tendrán un rol de *Usuario* y como contraseña su número de trabajador, que por seguridad deberá ser cambiada al primer ingreso por parte de cada profesor o individualmente especificando el rol y contraseña. La baja de los usuarios debe ser conociendo el id o número de trabajador del profesor, se confirmaran los datos y posteriormente se dará de baja.

*Alta de Periodos*: también estará activo para los usuarios de tipo *Administrador*, a donde se seleccionara el periodo que se desea dar de alta: primavera, verano u otoño, solo se podrán dar de alta esos tres periodos por año.

### 3.2.2 Funciones del sistema.

En la siguiente tabla (Figura 3.5) se describen las funciones que el sistema va a tener, así como los parámetros de entrada requeridos por la función y los parámetros devueltos.

<b>Nombre.</b>	<b>Función</b>	<b>Parámetros de Entrada</b>	<b>Parámetros Devueltos o resultado Obtenido</b>
<i>Acceder al Sistema.</i>	Se valida en la Base de Datos si el usuario existe y si la contraseña ingresada es correcta.	Número de trabajador del profesor y contraseña	Resultado booleano y una llave. - Verdadero (permite ingresar al sistema) y asigna una llave para poder realizar modificaciones en la Base de Datos. - Falso (datos inválidos)
<i>Actualizar Datos</i>	Se actualizan los datos generales del profesor en la Base de Datos	Datos generales (Nombre, Apellido Materno, Apellido Paterno, correo electrónico, llave)	Resultado booleano - Verdadero (mensaje de éxito) - Falso (mensaje de error por excepción)
<i>Cambiar Contraseña</i>	Se actualiza la contraseña en la Base de Datos, se confirma la nueva contraseña antes de realizar el cambio.	Número de trabajador del profesor, contraseña anterior, contraseña nueva y llave.	Resultado booleano - Verdadero (mensaje de éxito) - Falso (mensaje de error por excepción)
<i>Cargar lista de Archivo</i>	Permite cargar una lista de alumnos a la Base de Datos.	Lista de Alumnos con Matricula y nombre, NRC, Nombre de la Materia, Sección, período y llave.	Resultado booleano - Verdadero (mensaje de éxito, inserción de Grupo en Base de Datos) - Falso (mensaje de error por excepción)
<i>Asignar Porcentaje de Calificaciones (Dependiente de Cargar lista de Archivo)</i>	Permite elegir el tipo de calificación con el que se va a evaluar al grupo y el porcentaje que se le asignara a cada tipo de calificación.	Grupo previamente cargado, llave.	Resultado booleano - Verdadero (mensaje de éxito) - Falso (mensaje de error por excepción)
<i>Calificar Grupo</i>	Se ingresa el NRC del grupo previamente cargado, se realiza la calificación del grupo y posteriormente se guardan las actualizaciones en la Base de Datos.	El NRC de algún grupo previamente cargado.	Resultado booleano - Verdadero (mensaje de éxito) - Falso (mensaje de error por excepción)
<i>Generar Reportes.</i>	Tiene dos opciones: -Imprimir la lista de calificación que se está calificando. -Guardar en formato Excel la lista.	Para Imprimir el reporte o guardarlo solo se necesita el NRC del Grupo que se está calificando.	Impresión del reporte en una nueva página web o archivo guardado con formato Excel.
<i>Administrar Usuarios (Administrador)</i>	Tiene tres opciones: -Dar de alta usuarios a través de un listado de profesores con formato Excel. -Alta individual. -Baja individual.	Número de Trabajador, nombre y apellidos del profesor, correo electrónico, contraseña, rol y llave.	Resultado booleano - Verdadero (mensaje de éxito) - Falso (mensaje de error por excepción)
<i>Alta de Períodos</i>	Se elige el año del cual	El año, el periodo y	Resultado booleano

(Administrador)	se va crear el período, se selecciona el período.	llave.	<ul style="list-style-type: none"> <li>- Verdadero (mensaje de éxito, inserción de Período en Base de Datos)</li> <li>- Falso (mensaje de error por excepción o porque ya existe el período)</li> </ul>
-----------------	---	--------	---

**Figura 3.5. Funciones del sistema.**

### 3.2.3 Características de los actores del sistema.

En las siguientes tablas (Figura 3.6 y Figura 3.7), se muestran los actores que van a interactuar con el sistema, las funciones que van a utilizar y su descripción.

<b>Actor</b>	Profesor.
<b>Casos de Uso</b>	Acceder al Sistema, Actualizar Datos, Cargar Listas, Asignar Porcentaje de Calificaciones, Calificar Grupo, Generar Reportes, Administrar Usuarios ( <i>Administrador</i> ), Alta de Períodos ( <i>Administrador</i> ).
<b>Tipo</b>	Primario.
<b>Descripción</b>	Este actor es la parte esencial en el uso del sistema, el profesor puede: <ul style="list-style-type: none"> <li>- Ingresar al Sistema.</li> <li>- Actualizar sus Datos.</li> <li>- Cambiar su Contraseña.</li> <li>- Cargar listas y asignarles su porcentaje, calificar las listas y generar reportes de las listas generadas.</li> <li>- Altas y bajas de usuarios (<i>Administrador</i>).</li> <li>- Alta de períodos (<i>Administrador</i>).</li> </ul>

**Figura 3.6. Descripción del Actor Profesor.**

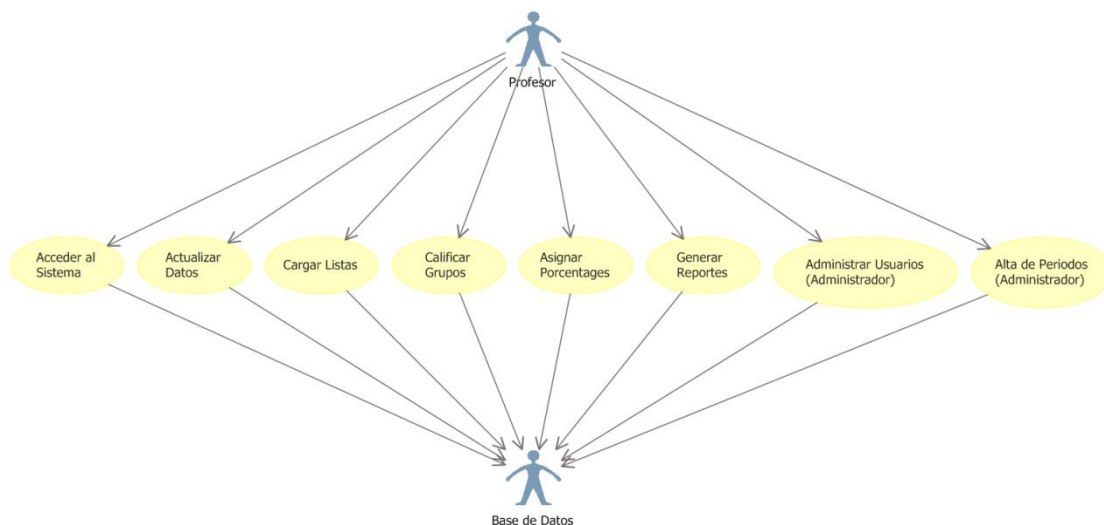
<b>Actor</b>	Base de Datos.
<b>Casos de Uso</b>	Acceder al Sistema, Actualizar Datos, Cargar listas, Asignar Porcentaje de Calificaciones, Calificar Grupo, Generar Reportes, Modificar Usuarios y Alta de Períodos.
<b>Tipo</b>	Primario.
<b>Descripción</b>	Este actor es también parte esencial en el uso del sistema, ya que almacena toda la información de un profesor, de los períodos, de los grupos que califica y de los alumnos que toman las materias.

**Figura 3.7. Descripción del Actor Base de Datos.**

### 3.2.4 Casos de uso.

En el siguiente caso de uso (Figura 3.8) se describe de manera general la funcionalidad del sistema, un profesor puede acceder al sistema, actualizar sus datos, cargar listas, calificar grupos, administrar usuarios, dar de alta períodos y asignar el porcentajes a cada tipo de calificación, también puede generar reportes después de

haber calificado un grupo (guardar reportes en formato Excel e imprimir el reporte de calificaciones).



**Figura 3.8.** Diagrama general de casos de uso.

**Validar usuario.**

En el caso de uso *Validar Usuario* (Figura 3.9 y Figura 3.10) se describe como el profesor puede ingresar al sistema.

<b>Caso de Uso</b>	Validar Usuario (S-1)
<b>Actores</b>	Usuario, Base de Datos.
<b>Tipo</b>	Básico.
<b>Propósito</b>	Permitir a un usuario ingresar al el sistema.
<b>Resumen</b>	El usuario inicia este caso de uso, ofrece la funcionalidad para ingresar al Sistema.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Usuario Ingresa sus datos de acceso al sistema.</li> <li>2. Se validan datos por parte de la Base de Datos.</li> <li>3. Si los datos son correctos se permite el acceso al sistema al usuario y se genera una llave para hacer modificaciones en la Base de Datos.</li> </ol>
<b>Flujo Alternativo.</b>	<ol style="list-style-type: none"> <li>3. Si los datos no son correctos se avisa al usuario para la corrección de datos.</li> </ol>
<b>Precondiciones</b>	Ninguna.

**Figura 3.9.** Caso de Uso Validar Usuario.

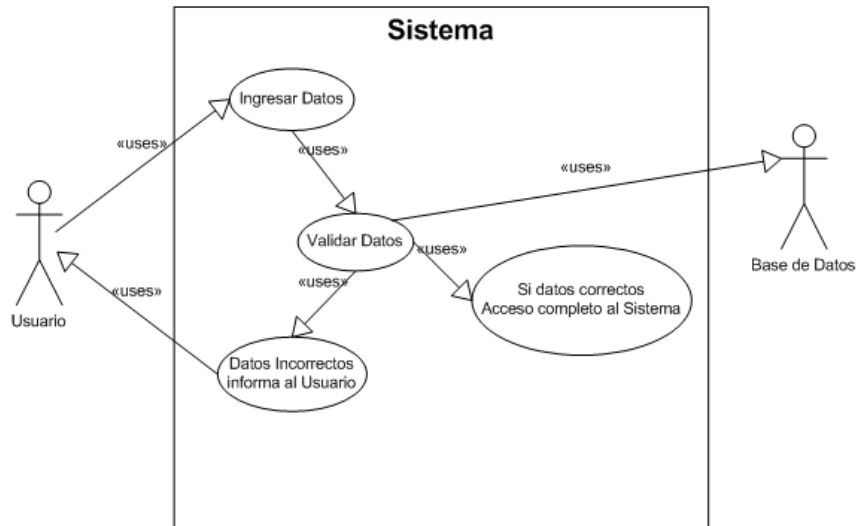


Figura 3.10. Diagrama Caso de Uso Validar Usuario.

### Actualizar datos de usuario.

En el caso de uso *Actualizar Datos de Usuario* (Figura 3.11 y Figura 3.12) se describe como el profesor puede modificar sus datos generales.

<b>Caso de Uso</b>	Actualizar Datos de Usuario (S-2)
<b>Actores</b>	Usuario, Base de Datos.
<b>Tipo</b>	Secundario.
<b>Propósito</b>	Permitir a un usuario modificar sus datos.
<b>Resumen</b>	El usuario inicia este caso de uso. Ofrece la funcionalidad para modificar datos personales.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Usuario Ingresa datos a modificar.</li> <li>2. Se validan datos por parte de la Base de Datos.</li> <li>3. Si los datos son correctos se realiza la actualización en la Base de Datos.</li> </ol>
<b>Flujo Alternativo.</b>	<ol style="list-style-type: none"> <li>3. Si los datos ingresados no tiene caracteres validos se avisa al usuario para la corrección de datos.</li> </ol>
<b>Precondiciones</b>	Requiere ejecutar inicialmente el caso de uso <i>Validar Usuario(S-1)</i> .

Figura 3.11. Caso de Uso Actualizar Datos de Usuario.

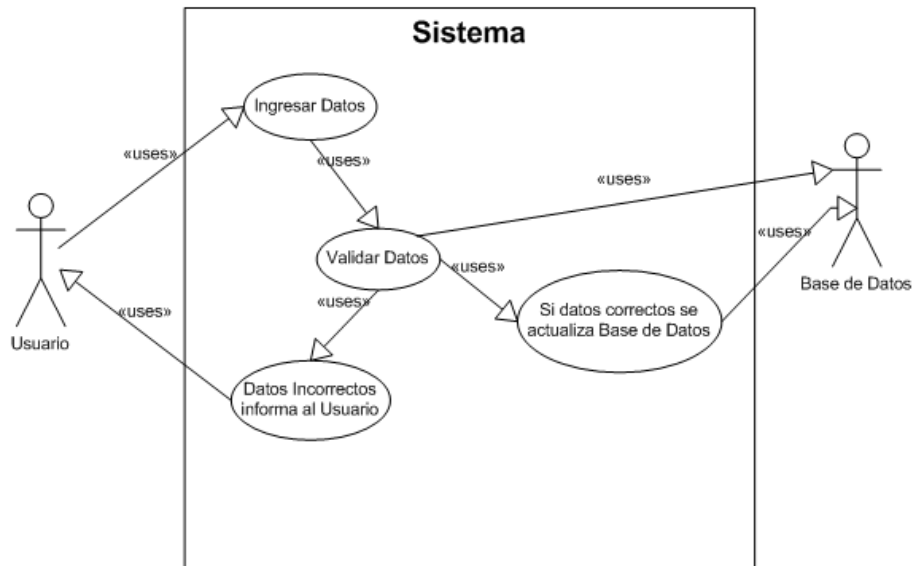


Figura 3.12. Diagrama Caso de Uso Actualizar Datos de Usuario.

### Actualizar contraseña de usuario.

En el caso de uso *Actualizar Contraseña de Usuario* (Figura 3.13 y Figura 3.14) se describe como el profesor puede cambiar su contraseña actual.

<b>Caso de Uso</b>	Actualizar Contraseña de Usuario (S-3)
<b>Actores</b>	Usuario, Base de Datos.
<b>Tipo</b>	Secundario.
<b>Propósito</b>	Permitir a un usuario modificar su contraseña.
<b>Resumen</b>	El usuario inicia este caso de uso. Ofrece la funcionalidad para modificar su contraseña.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Usuario Ingresa contraseña actual y la contraseña.</li> <li>2. Se validan datos por parte del Sistema y de la Base de Datos.</li> <li>3. Si los datos son correctos se realiza la actualización en la Base de Datos.</li> </ol>
<b>Flujo Alternativo.</b>	<ol style="list-style-type: none"> <li>3. Si la contraseña anterior no es válida o no se confirma la nueva contraseña se avisa al usuario para la corrección de datos.</li> </ol>
<b>Precondiciones</b>	Requiere ejecutar inicialmente el caso de uso <i>Validar Usuario(S-1)</i> .

Figura 3.13. Caso de Uso Actualizar Contraseña de Usuario.

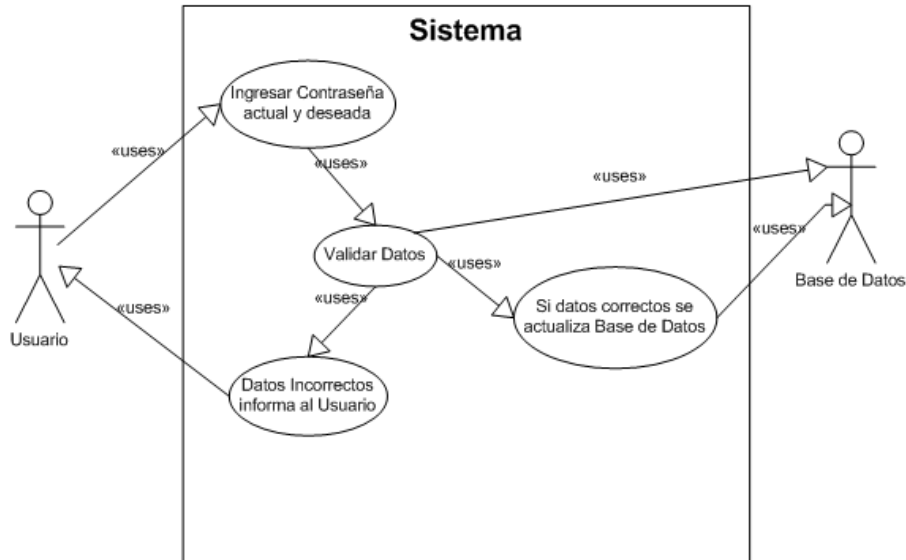


Figura 3.14. Diagrama Caso de Uso Actualizar Contraseña de Usuario.

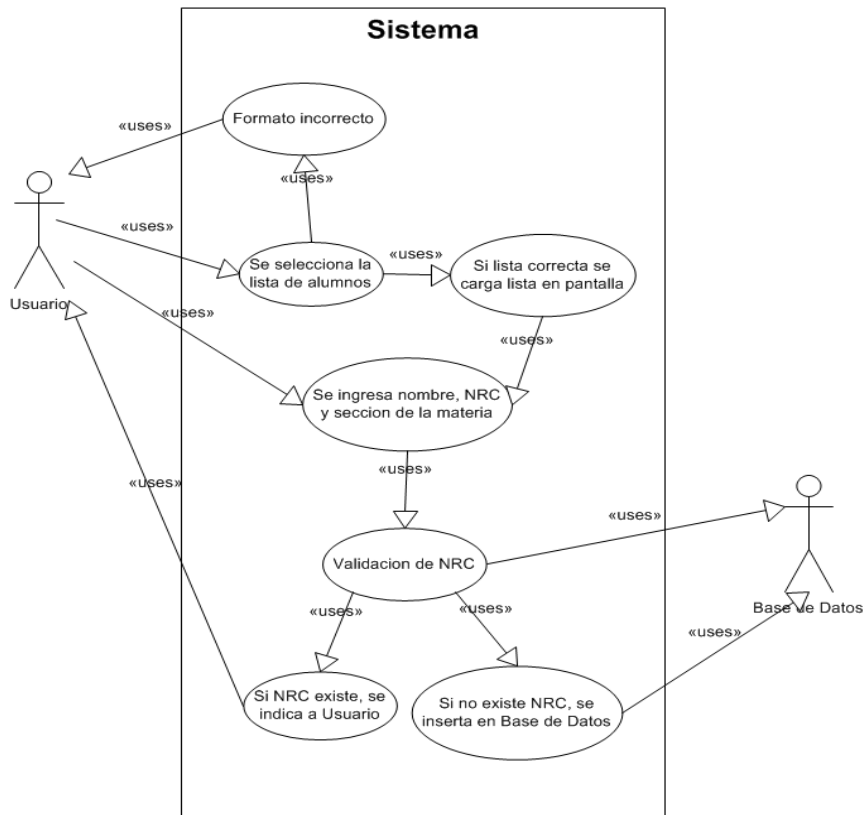
### Cargar lista.

En el caso de uso *Cargar Lista* (Figura 3.15 y Figura 3.16) se describe como el profesor puede insertar nuevos grupos en el sistema.

<b>Caso de Uso</b>	Cargar Lista (S-4)
<b>Actores</b>	Usuario, Base de Datos.
<b>Tipo</b>	Básico.
<b>Propósito</b>	Permitir a un usuario cargar listas de calificaciones a la Base de Datos.
<b>Resumen</b>	El usuario inicia este caso de uso. Ofrece la funcionalidad para subir listas a la Base de Datos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Se selecciona un archivo en formato <i>Excel</i> predeterminado, o se copia texto en formato predeterminado.</li> <li>2. Si el formato del archivo o el texto copiado es correcto, se carga la lista de calificaciones en el sistema.</li> <li>3. Se ingresa nombre de la materia, <i>NRC</i>, sección de la materia y período.</li> <li>4. Se valida que el <i>NRC</i> no exista en la Base de Datos, que se halla asignado nombre y sección a la materia, si no existe se carga la lista de calificaciones.</li> </ol>
<b>Flujo Alternativo.</b>	<ol style="list-style-type: none"> <li>2. Si el Formato de la lista de calificaciones no es correcto se indica al usuario para intentarlo nuevamente.</li> <li>4. Si el <i>NRC</i> existe se indica para asignar un nuevo <i>NRC</i> o si no se asignó nombre de la Materia, número de Sección o período a evaluar.</li> </ol>

<b>Precondiciones</b>	Requiere ejecutar inicialmente el caso de uso <i>Validar Usuario(S-1)</i> .
-----------------------	---

**Figura 3.15.** Caso de Uso Cargar Lista.



**Figura 3.16.** Diagrama Caso de Uso Cargar Lista.

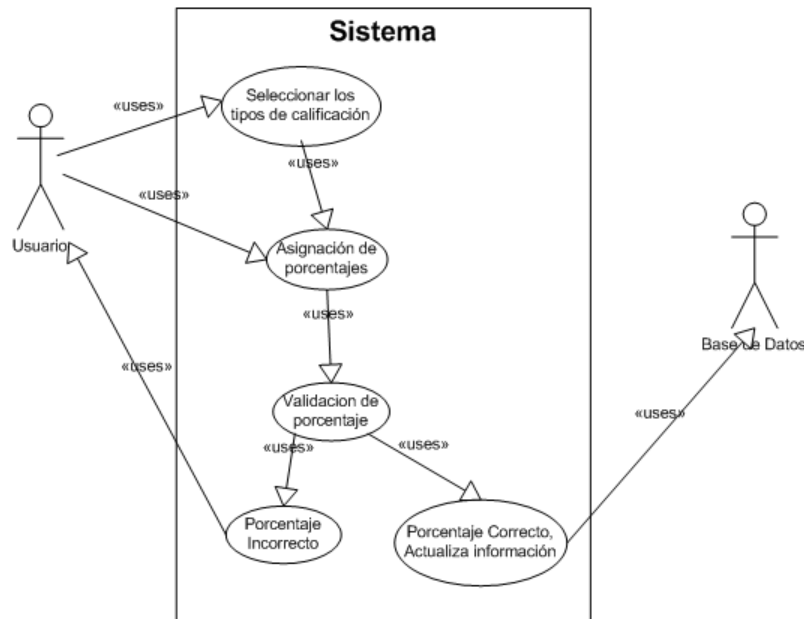
**Asignar porcentaje a calificaciones.**

En el caso de uso *Asignar Porcentaje a Calificaciones* (Figura 3.17 y Figura 3.18) se describe como el profesor puede asignar porcentaje al grupo que se cargó, para poder evaluarlo después.

<b>Caso de Uso</b>	Asignar Porcentaje a Calificaciones (S-5)
<b>Actores</b>	Usuario, Base de Datos.
<b>Tipo</b>	Básico.
<b>Propósito</b>	Permitir a un usuario asignar porcentajes y tipo calificación a una lista.
<b>Resumen</b>	El usuario inicia este caso de uso. Ofrece la funcionalidad para asignar tipos de calificación y porcentajes a un grupo.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Se selecciona el tipo de calificación con el cual se va a calificar el grupo.</li> <li>2. Se le asigna un porcentaje a cada tipo de calificación.</li> <li>3. Si el porcentaje es correcto se actualiza en la Base de Datos con la información</li> </ol>

	correspondiente al grupo.
<b>Flujo Alternativo.</b>	3. Si el porcentaje no es correcto se indica al usuario para corregir.
<b>Precondiciones</b>	Requiere ejecutar inicialmente el caso de uso <i>Validar Usuario(S-1)</i> , también requiere <i>Cargar Lista(S-4)</i> .

**Figura 3.17.** Caso de Uso Asignar Porcentaje a Calificaciones.



**Figura 3.18.** Diagrama Caso de Uso Asignar Porcentaje a Calificaciones.

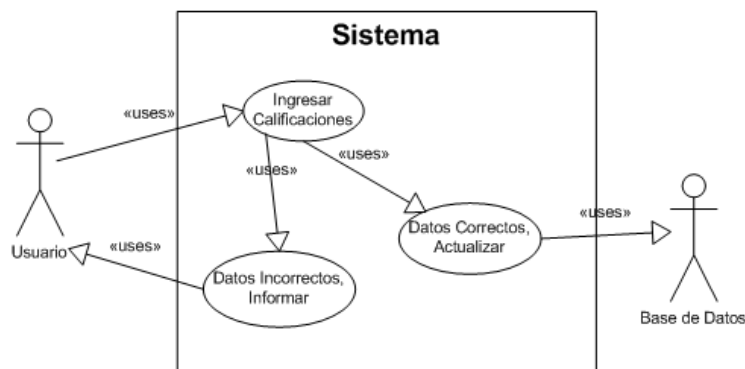
### Calificar grupo.

En el caso de uso *Calificar Grupo* (Figura 3.19 y Figura 3.20) se describe como el profesor asigna calificaciones al grupo que se está evaluando.

<b>Caso de Uso</b>	Calificar Grupo (S-6)
<b>Actores</b>	Usuario, Base de Datos.
<b>Tipo</b>	Básico.
<b>Propósito</b>	Permitir a un usuario asignar calificaciones a una lista de alumnos perteneciente a un grupo.
<b>Resumen</b>	El usuario inicia este caso de uso. Ofrece la funcionalidad para asignar calificaciones a un grupo, también se pueden añadir o quitar columnas de calificaciones.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Se ingresan las calificaciones al grupo.</li> <li>2. Si las Calificaciones son correctas se puede actualizar la Base de Datos con las calificaciones.</li> </ol>
<b>Flujo Alternativo.</b>	2. Si alguna hay algún error se indica al usuario para corregir.

<b>Precondiciones</b>	Requiere ejecutar inicialmente el caso de uso <i>Validar Usuario(S-1)</i> y <i>Asignar Porcentaje a Calificaciones (S-5)</i> .
-----------------------	--

**Figura 3.19.** Caso de Uso *Calificar Grupo*.



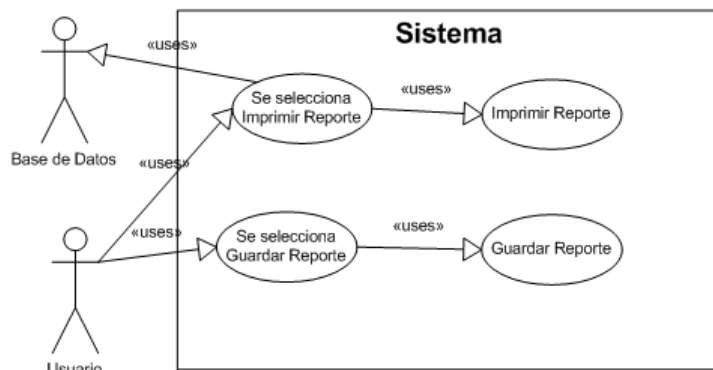
**Figura 3.20.** Diagrama Caso de Uso *Calificar Grupo*.

### Generar reportes.

En el caso de uso *Generar Reportes* (Figura 3.21 y Figura 3.22) se describe como el profesor puede generar reportes del grupo que está evaluando.

<b>Caso de Uso</b>	Generar Reportes (S-6)
<b>Actores</b>	Usuario, Base de Datos.
<b>Tipo</b>	Básico.
<b>Propósito</b>	Permitir a un usuario generar reportes de calificaciones de una lista de calificaciones.
<b>Resumen</b>	El usuario inicia este caso de uso. Ofrece la funcionalidad para imprimir reportes de las listas de calificaciones o también poder guardarlos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Se selecciona imprimir el reporte.</li> <li>2. Se imprime el reporte de calificaciones de la lista que se está calificando actualmente.</li> </ol>
<b>Flujo Alternativo.</b>	<ol style="list-style-type: none"> <li>1. Se selecciona guardar el reporte.</li> <li>2. Se guarda el reporte de calificaciones de la lista que se está calificando actualmente en formato <i>Excel</i>.</li> </ol>
<b>Precondiciones</b>	Requiere ejecutar inicialmente el caso de uso <i>Validar Usuario(S-1)</i> y <i>Calificar Grupo (S-6)</i> .

**Figura 3.21.** Caso de Uso *Generar Reportes*.



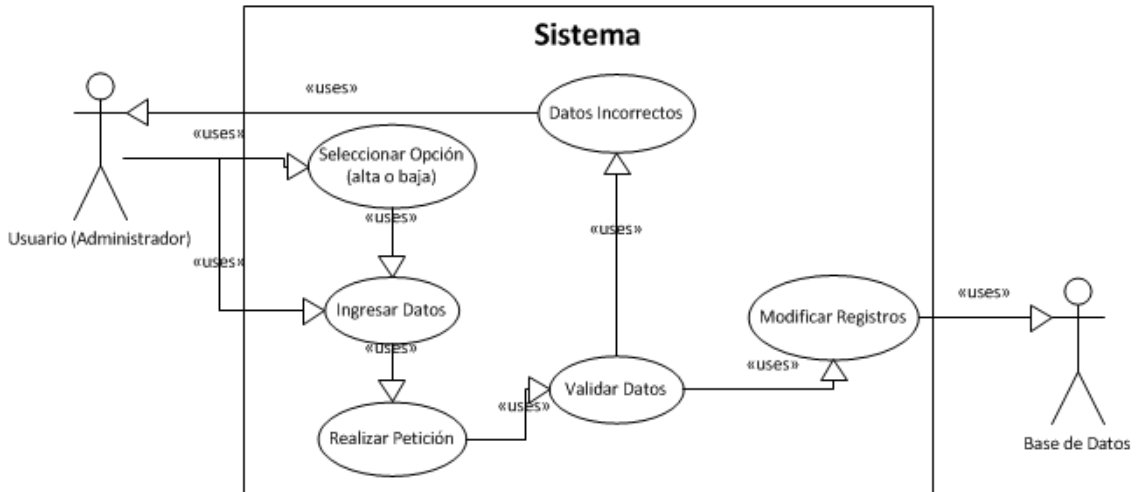
**Figura 3.22.** Diagrama *Caso de Uso Generar Reportes*.

### Administrar usuarios.

En el caso de uso *Administrar Usuarios* (Figura 3.23 y Figura 3.24) se describe como el profesor con un rol de *Administrador* puede dar de alta o baja usuarios.

<b>Caso de Uso</b>	Administrar Usuarios (S-7)
<b>Actores</b>	Usuario ( <i>Administrador</i> ), Base de Datos.
<b>Tipo</b>	Básico.
<b>Propósito</b>	Permitir a un usuario <i>Administrador</i> dar de alta o baja usuarios para la el sistema.
<b>Resumen</b>	El usuario <i>Administrador</i> inicia este caso de uso. Ofrece la funcionalidad para dar de alta usuarios con un rol o darlos de baja.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Se selecciona la opción alta de usuarios, alta individual o baja individual.</li> <li>2. De acuerdo a la opción seleccionada se realiza la inserción o se borra el registro correspondiente en la Base de Datos en caso de que los datos sean correctos.</li> </ol>
<b>Flujo Alternativo.</b>	<ol style="list-style-type: none"> <li>2. En caso de existir alguna excepción o error se indica al usuario.</li> </ol>
<b>Precondiciones</b>	Requiere ingresar al sistema con una cuenta de <i>Administrador</i> .

**Figura 3.23.** *Caso de Uso Administrar Usuarios*.



**Figura 3.24.** Diagrama Caso de Uso Administrar Usuarios.

**Alta de período.**

En el caso de uso *Alta de Período* (Figura 3.25 y Figura 3.26) se describe como el profesor con un rol de *Administrador* puede dar de alta periodos para evaluar grupos.

<b>Caso de Uso</b>	Alta de Período(S-8)
<b>Actores</b>	Usuario ( <i>Administrador</i> ), Base de Datos.
<b>Tipo</b>	Básico.
<b>Propósito</b>	Permitir a un usuario <i>Administrador</i> dar de alta periodos en los que estarán los grupos.
<b>Resumen</b>	El usuario <i>Administrador</i> inicia este caso de uso. Ofrece la funcionalidad para dar de alta periodos.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1. Se selecciona la opción alta de períodos.</li> <li>2. Se realiza la inserción de un nuevo periodo en la Base de Datos en caso de que los datos sean correctos.</li> </ol>
<b>Flujo Alternativo.</b>	<ol style="list-style-type: none"> <li>2. En caso de existir alguna excepción o error se indica al usuario.</li> </ol>
<b>Precondiciones</b>	Requiere ingresar al sistema con una cuenta de <i>Administrador</i> .

**Figura 3.25.** Caso de Uso Alta de Período.

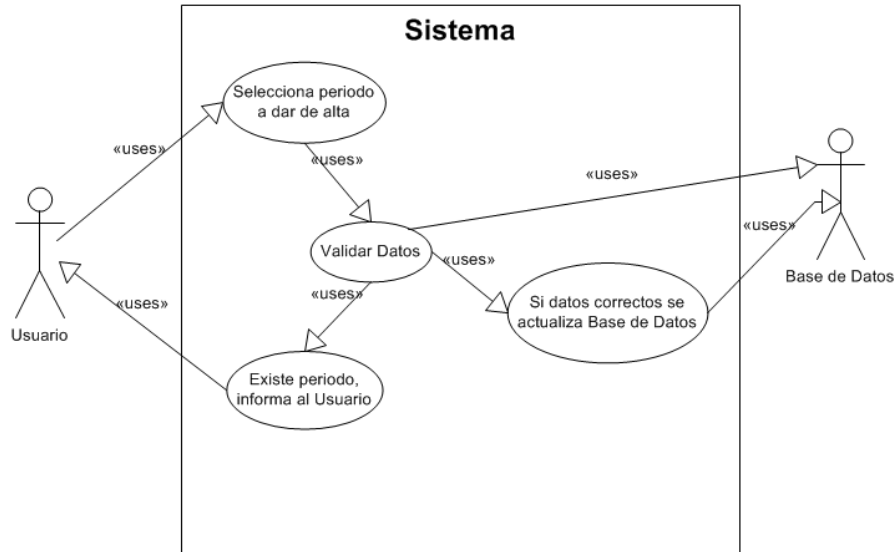


Figura 3.26. Diagrama Caso de Uso Alta de Período.

### 3.3 Diseño.

#### 3.3.1 Diseño de la Base de Datos.

En la siguiente figura (Figura 3.27) se muestra el modelo entidad relación de la Base de Datos que se va a crear.

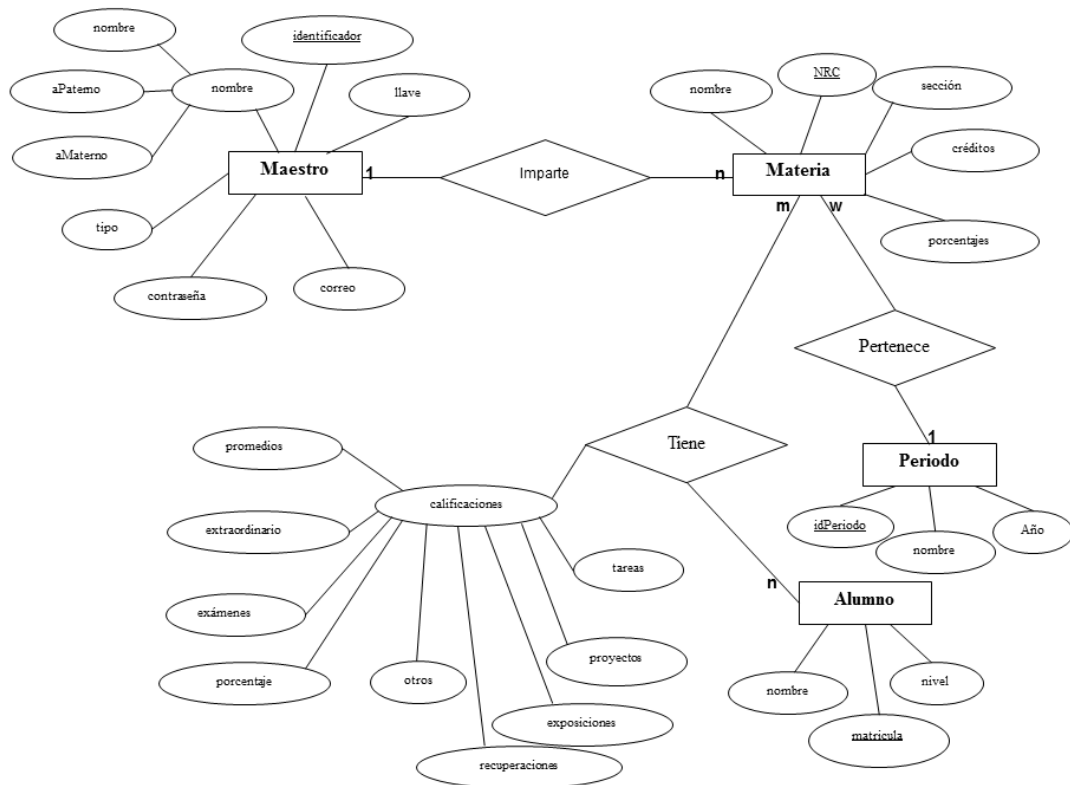


Figura 3.27. Modelo Entidad Relación.

Para mejor entendimiento del diagrama, se abreviaron los campos:

porcentajes = Porcentaje de exámenes, de tareas, exposiciones, proyectos y otros.  
promedios = Promedio de exámenes, de tareas, exposiciones, proyectos, otros y final.  
exámenes =examen1, examen2, examen3, examen4.  
tarea = tarea1, tarea2, tarea3, tarea4.  
proyectos = proyecto1, proyecto2, proyecto3, proyecto4.  
exposiciones = exposicion1, exposicion2, exposicion3, exposicion4.  
otros = otros1, otros2, otros3, otros4.  
porcentaje = porcentaje examen, porcentaje tarea, porcentaje otros, porcentaje exposiciones, porcentaje proyectos.  
recuperaciones = recuperacion1, recuperacion2, recuperacion3, recuperacion4.  
Cada uno de los campos simplificados para mejor entendimiento en los diagramas y cualquier otro tipo de notación son atómicos.

### **Modelo relacional.**

Mapeo de Modelo Entidad relación a Relaciones.

1.- Por cada entidad crear una tabla.

Maestro ()  
Materia ()  
Alumno ()  
Periodo ()

2.- Cada atributo simple y nodo hoja de los atributos compuestos es una columna de las tablas.

Maestro (no trabajador, nombre, a\_paterno, a\_materno, email, contraseña, tipo, llave).

Materia (nrc, nombre, sección, créditos, porcentajes).

Alumno (matricula, nombre, nivel).

Periodo (idPeriodo, año, nombre).

3.- Cardinalidad 1:n.

Maestro (no trabajador, nombre, a\_paterno, a\_materno, email, contraseña, tipo, llave).

Materia (nrc, no\_trabajador(FK), idPeriodo(FK), nombre, sección, créditos, porcentajes).

Alumno (matricula, nombre, nivel).

Periodo (idPeriodo, año, nombre).

4.- Cardinalidad m:n.

Maestro (no trabajador, nombre, a\_paterno, a\_materno, email, contraseña, tipo, llave).

Materia (nrc, no\_trabajador(FK), idPeriodo(FK), nombre, sección, créditos, porcentajes).

Alumno (matricula, nombre, nivel).

Periodo (idPeriodo, año, nombre).

Tiene (nrc, matricula, exámenes, tareas, exposiciones, proyectos, otros, porcentaje, recuperaciones, extraordinario, promedios).

NOTA: Para mejor entendimiento de la tabla generada en base a la relación que existe entre Alumno y Materia, se le cambiara el nombre de "Tiene", por el de "Calificación", entonces las tablas hasta este paso quedan de la siguiente manera.

Maestro (no trabajador, nombre, a\_paterno, a\_materno, email, contraseña, tipo, llave).

Materia (nrc, no\_trabajador(FK), idPeriodo(FK), nombre, sección, créditos, porcentajes).

Alumno (matricula, nombre, nivel).

Periodo (idPeriodo, año, nombre).

Calificación (nrc, matricula, exámenes, tareas, exposiciones, proyectos, otros, porcentaje, recuperaciones, extraordinario, promedios).

Aplicando normalización.

1FN.- Todas las tablas están en 1ra forma normal ya que todos los atributos son atómicos y dependientes de la llave primaria, en el caso de la tabla Alumno en el campo nombre, es atómico por la especificación y requerimientos del sistema su nombre completo será una sola cadena a diferencia del nombre del profesor, y los campos de la tabla calificación también son atómicos por las especificaciones descritas después del modelo entidad relación.

2FN.- Cumple la segunda forma normal ya que todos los atributos no clave de cada tabla son dependientes completamente de la clave primaria de cada tabla.

3FN.- También cumple la tercera forma normal, ya que cada campo de cada tabla es totalmente independiente de los demás campos, excepto de la llave primaria.

Realizados estos pasos, se procede la implementación de la *Base de Datos*.

Para la implementación de la *Base de Datos* en *SQL Server* se asignara el nombre Escuela. La estructura de la *Base de Datos* en *SQL Server* (Figura 3.28) y la implementación física (Figura 3.29) se muestran a continuación.

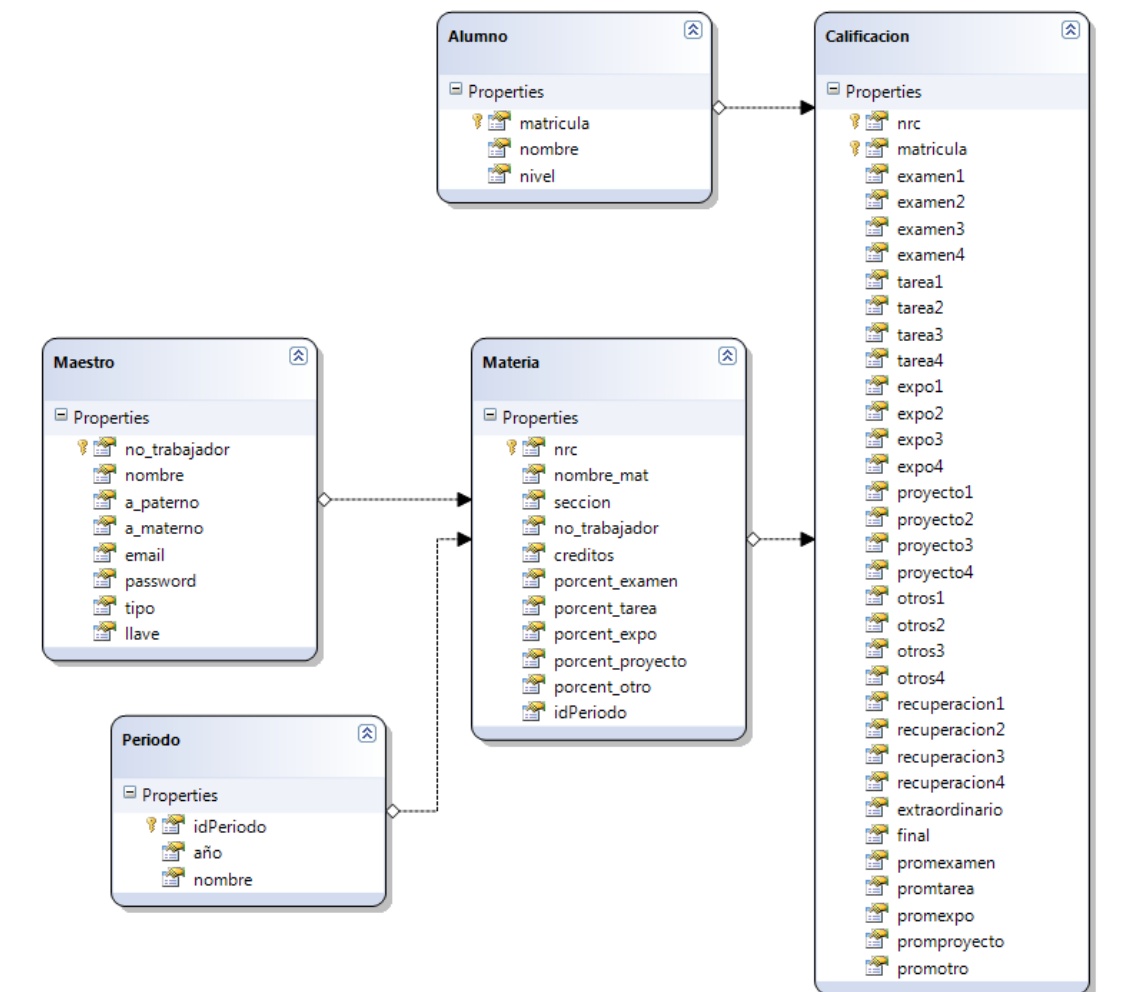
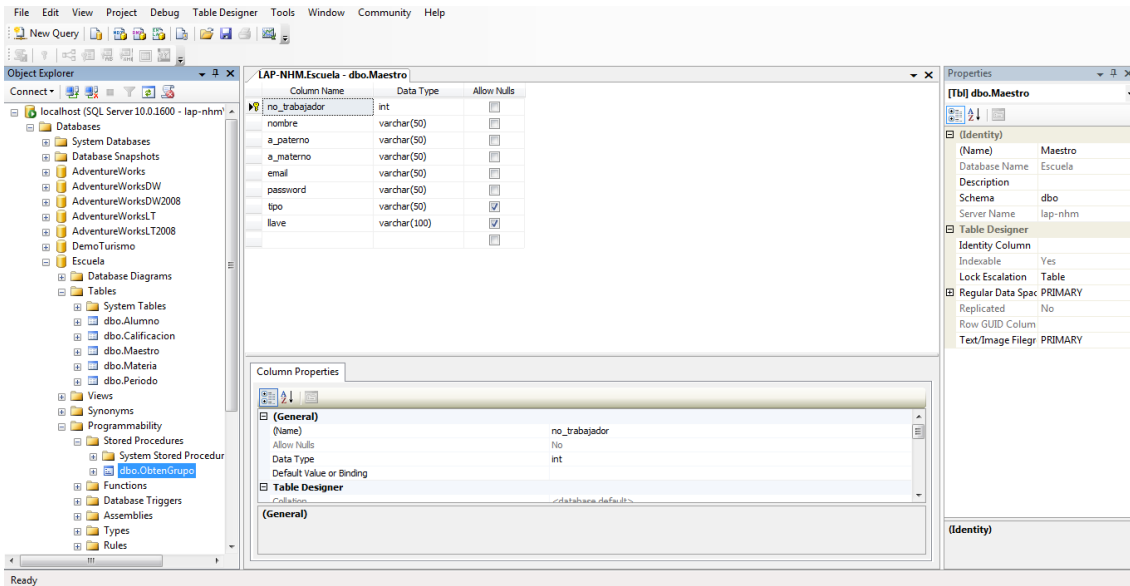


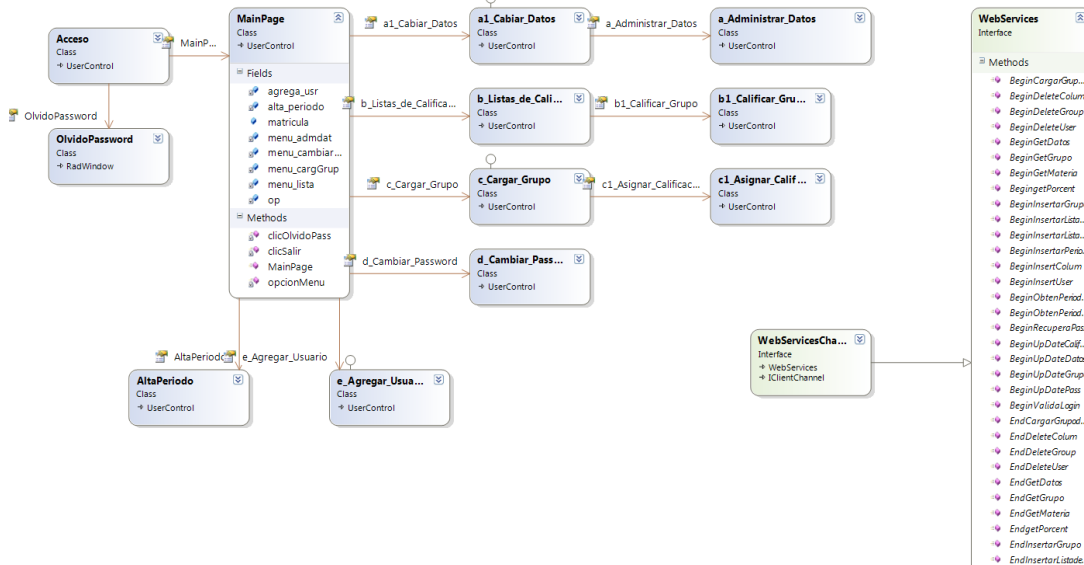
Figura 3.28. Diagrama de la Base de Datos en SQL Server 2008.



**Figura 3.29.** Implementación física de la Base de Datos en SQL Server 2008 (Además como se puede observar en la imagen se implementó un procedimiento almacenado para agilizar la consulta de los grupos, como resultado trae el nombre y matrícula de los estudiantes y sus calificaciones por NRC de Grupo).

### 3.3.2 Diagramas de clases.

El diagrama de clases se generó en la herramienta para desarrollo de diagramas de clases de *Visual Studio 2010*, a donde se liga cada control de usuario, con el control principal de la aplicación (*MainPage*). En una clase que no tiene conexión con las otras se montaron los servicios web (*WebServicesClient*) que serán consumidos vía web, como a continuación se muestra (Figura 3.30).



**Figura 3.30.** Diagrama de Clases.

### 3.3.3 Diagramas de actividades.

#### Validar usuario.

Se ingresa número de trabajador y contraseña, si los datos ingresados son válidos entonces se ingresa al sistema, en caso contrario se indica al usuario (Figura 3.31).

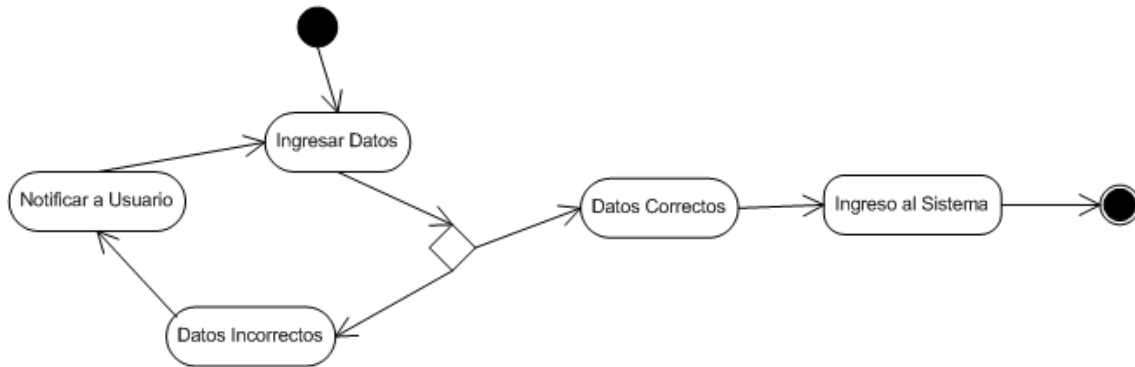


Figura 3.31. Validar usuario.

#### Actualizar datos.

Una vez ingresado al sistema, se tendrá la opción de cambiar datos personales. Se captura la información que se desea cambiar y se actualiza en la *Base de Datos* (no se puede actualizar el número de trabajador), en caso de que se ingresen caracteres no válidos u ocurra alguna excepción se indica al usuario (Figura 3.32).

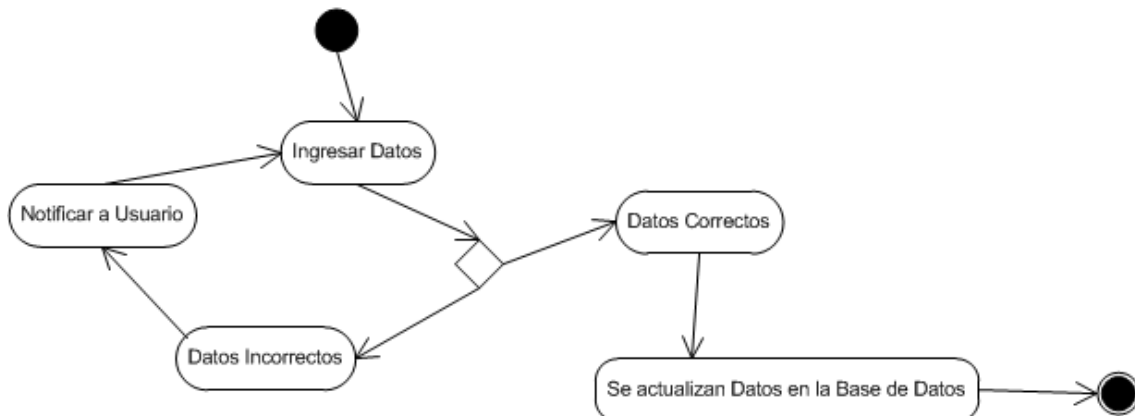
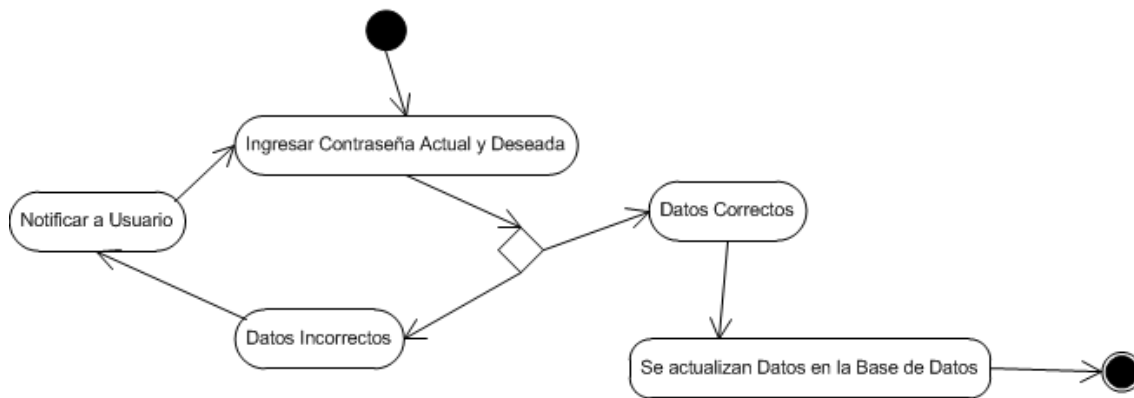


Figura 3.32. Actualizar datos de usuario.

#### Actualizar contraseña de usuario.

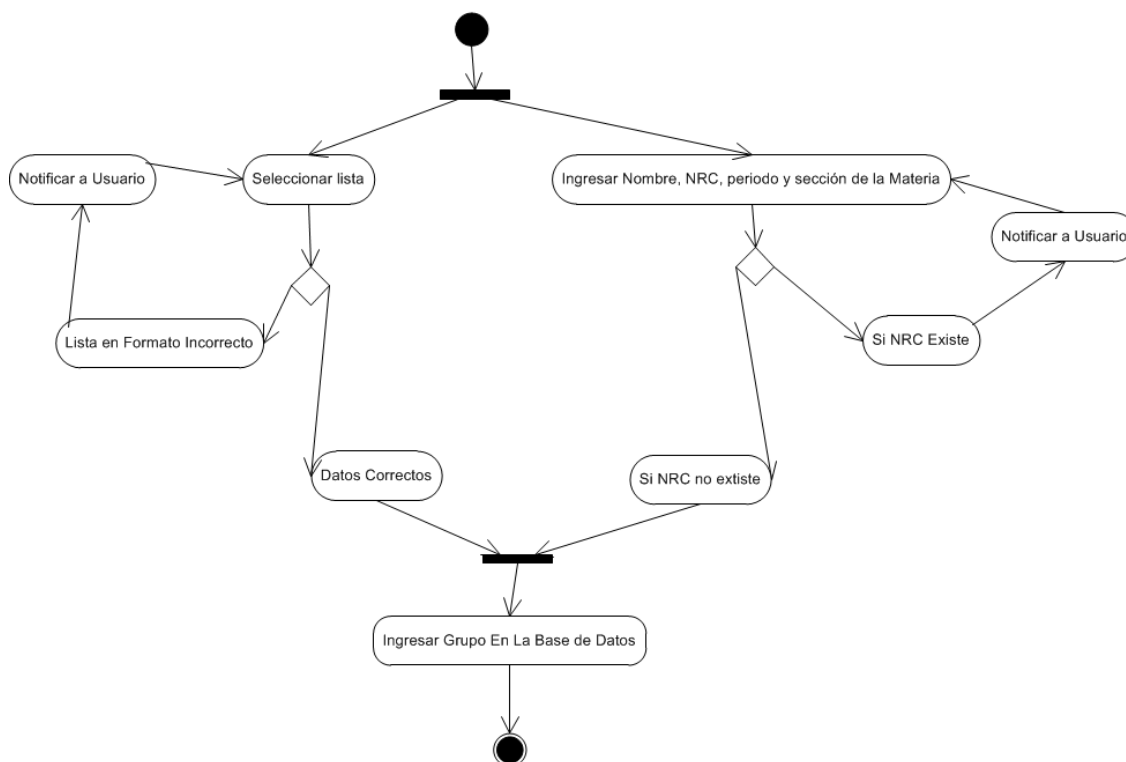
Otra opción que se tiene es cambiar la contraseña. Se ingresa la contraseña actual, se ingresa la contraseña deseada, se confirma y si todo es correcto se actualiza la nueva contraseña en la *Base de Datos*, en caso contrario se indica al usuario (Figura 3.33).



**Figura 3.33.** Actualizar contraseña de usuario.

**Cargar lista.**

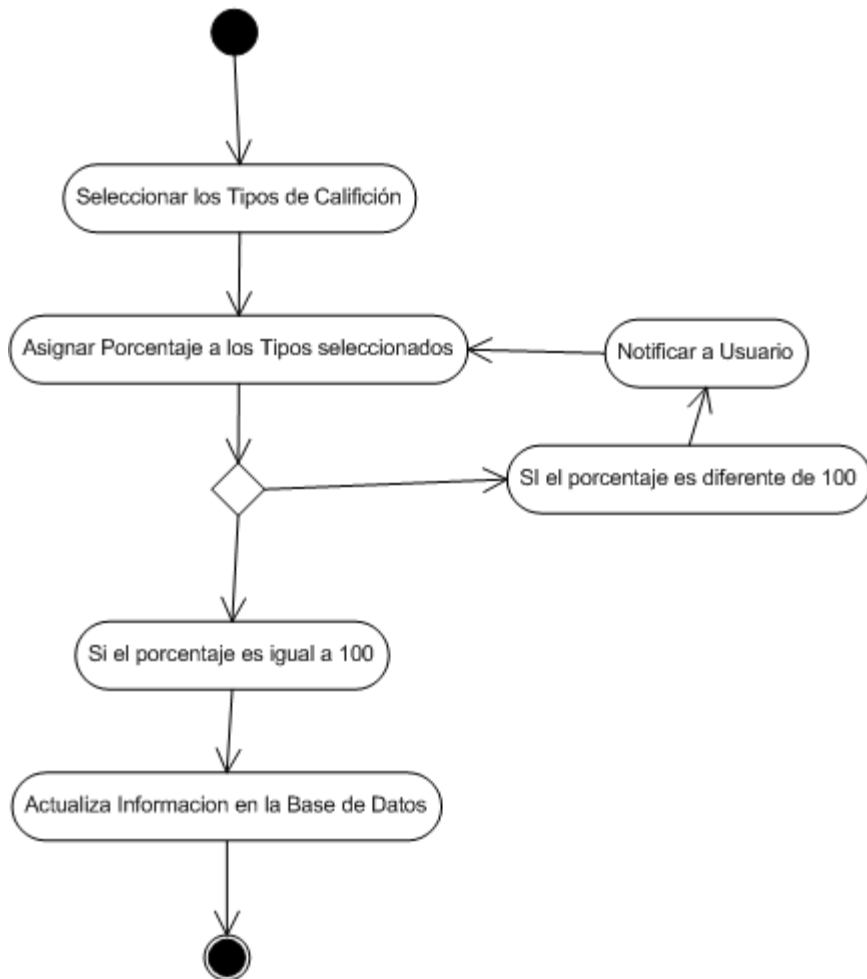
Para cargar una lista de calificaciones se debe seleccionar la lista que se quiere cargar, si el formato correcto entonces la lista se mostrara en pantalla, sino se indica al usuario. También se debe ingresar el nombre, sección, *NRC* de la materia y periodo, se valida que el *NRC* no exista en la *Base de Datos* y si no existe y el grupo ya se cargó en pantalla, entonces se puede guardar la lista de calificaciones en la *Base de Datos*, en caso contrario se informa al usuario (Figura 3.34).



**Figura 3.34.** Cargar lista.

**Asignar porcentajes.**

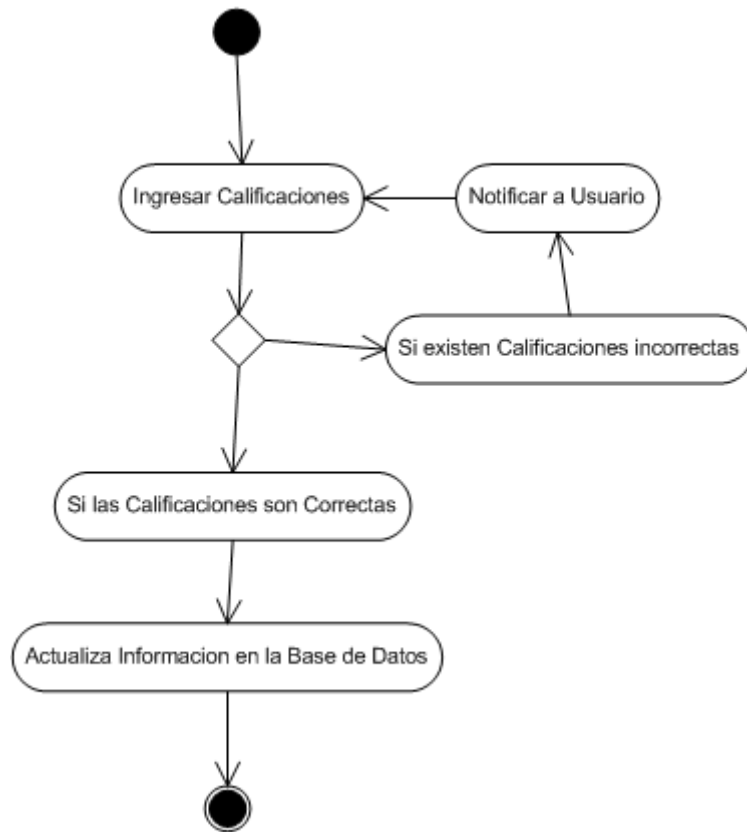
Aquí se elige como se va a evaluar el grupo, por ejemplo con exámenes proyectos, exposiciones, etc., una vez que se elijan los campos, se le asigna un porcentaje a cada tipo de calificación, para poder actualizar el porcentaje en la *Base de Datos*, el porcentaje tiene que ser igual a 100, posteriormente se puede calificar el grupo (Figura 3.35).



**Figura 3.35.** *Asignar porcentajes.*

**Calificar grupo.**

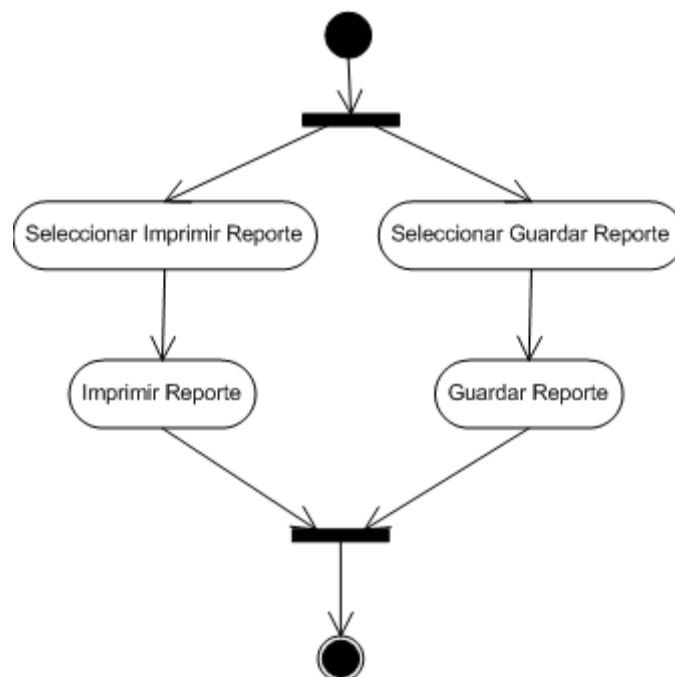
Se ingresan calificaciones a un grupo seleccionado, se valida si son correctas para actualizarlas en la *Base de Datos*, en caso de que exista algún error se notifica al usuario (Figura 3.36).



**Figura 3.36.** *Calificar grupo.*

**Generar reportes.**

Se selecciona la opción, si la opción seleccionada es imprimir reporte, se muestra en pantalla el reporte para ser impreso, si la opción seleccionada es guardar reporte, entonces se elige el nombre y la ubicación del reporte para que sea almacenado en formato *Excel* (Figura 3.37).



**Figura 3.37.** *Diagrama de actividades generar reportes.*

### Administrar usuarios (*Administrador*).

Se selecciona la opción (la opción se podrá elegir siempre y cuando se esté en una sesión de *Administrador*), una vez seleccionada se mostraran 3 opciones:

- La primera es agregar usuarios, donde se debe elegir la lista en un formato predeterminado y los maestros que no estén dados de alta aun en la *Base de Datos* serán añadidos a ella.
- La segunda es el alta individual, donde se deben llenar todos los campos con la información del maestro para posteriormente darlo de alta siempre y cuando sean correctos los datos y no exista ya el profesor.
- La tercera opción es la baja individual, a donde se debe ingresar el número de trabajador del profesor que se va a dar de baja, consultar sus datos y en caso de que exista se podrá dar de baja de la *Base de Datos*.

A continuación se muestra el diagrama correspondiente (Figura 3.38).

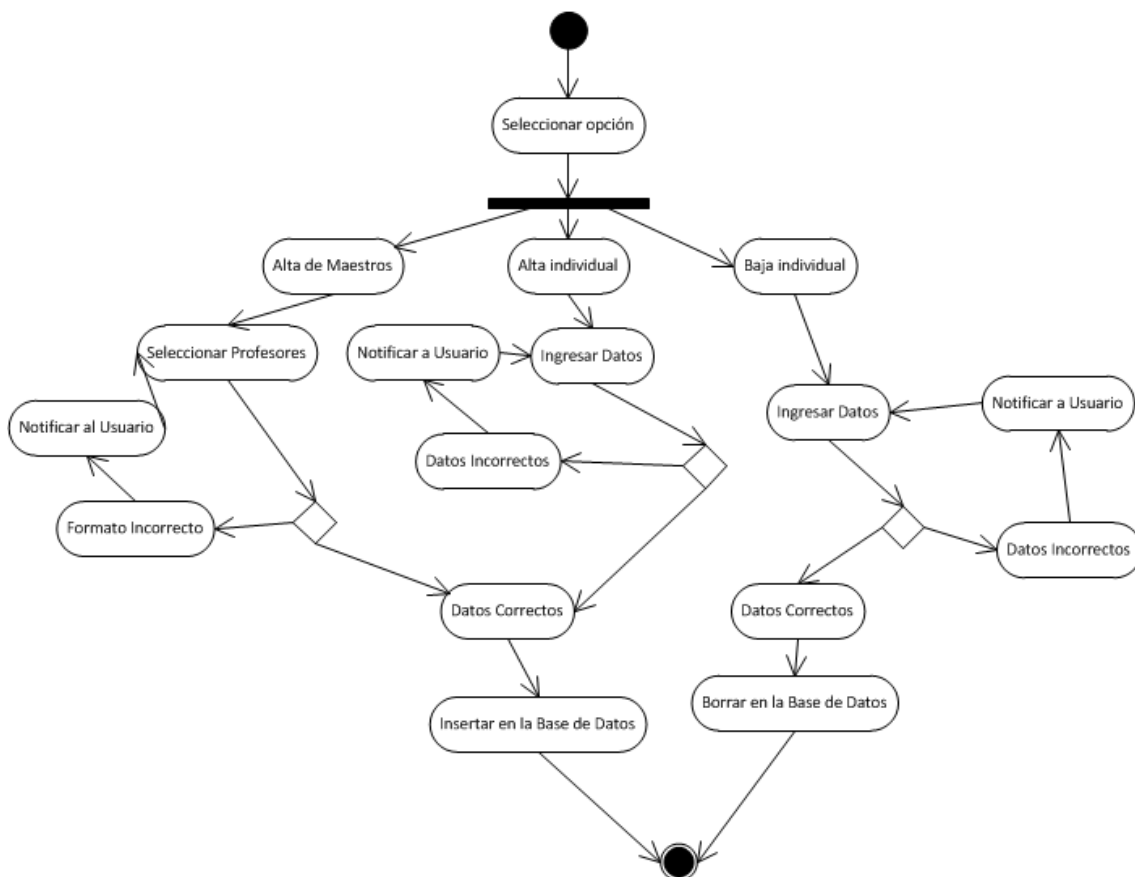


Figura 3.38. Diagrama de actividades administrar usuarios.

### Alta de períodos (*Administrador*).

Se selecciona el periodo (la opción se podrá elegir siempre y cuando se esté en una sesión de *Administrador*), solo pueden elegirse tres periodos por año, y solo se pueden dar de alta los periodos del año actual y del siguiente año (Figura 3.39).

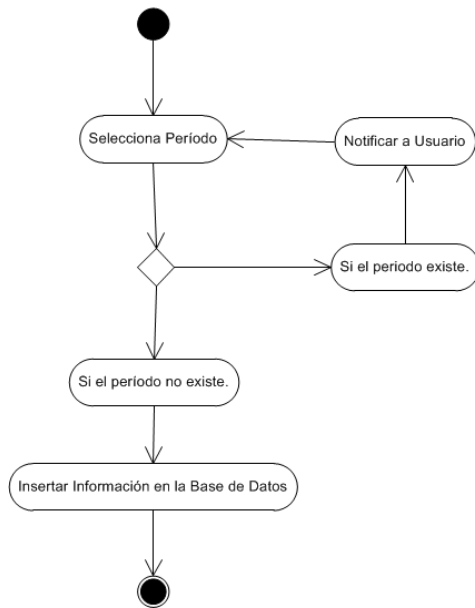


Figura 3.39. Diagrama de actividades alta de períodos.

### Flujo general.

El flujo de datos de la aplicación en general se muestra a continuación (Figura 3.40).

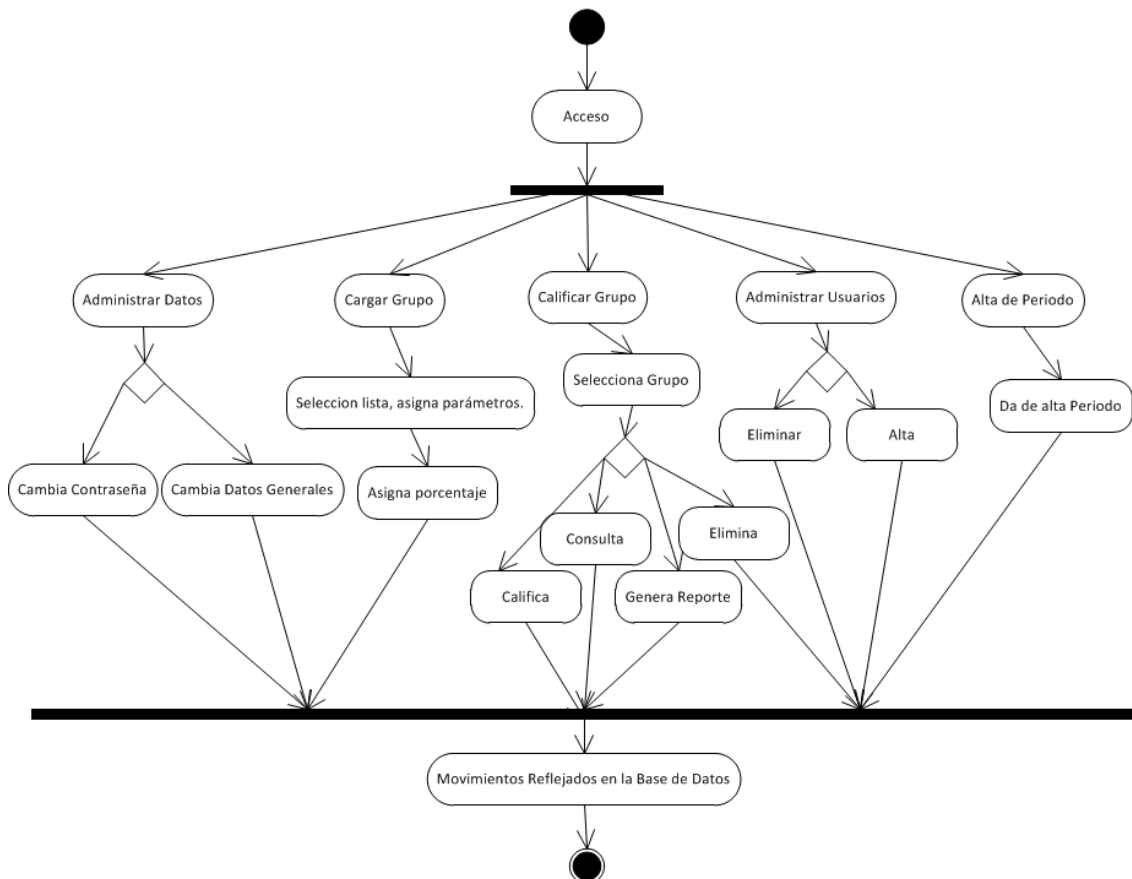


Figura 3.40. Diagrama general de flujo de datos de un usuario.

### 3.3.4 Diagramas de secuencia.

En la siguiente figura (Figura 3.41), se muestra la secuencia de la aplicación.

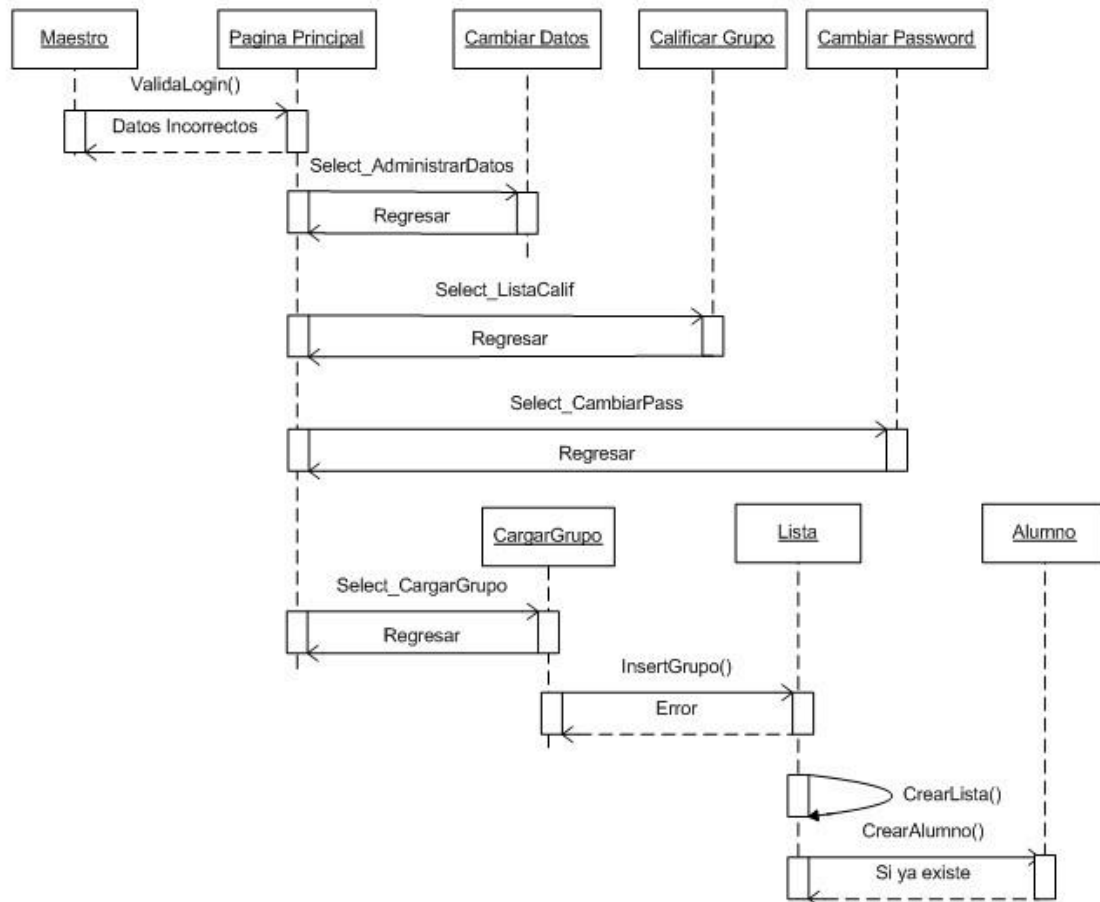
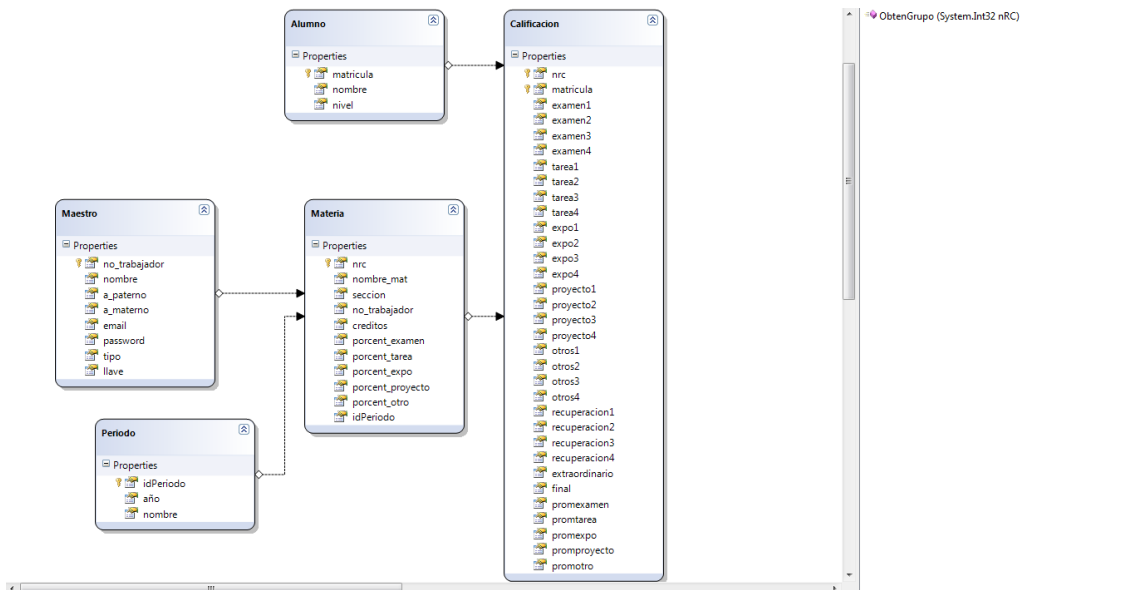


Figura 3.41. Diagrama de secuencia.

### 3.4 Implementación.

#### 3.4.1 Desarrollo del sistema mediante Silverlight.

Para empezar con el desarrollo de la aplicación es necesario pasar la estructura de la *Base de Datos* al proyecto que se va a realizar, para esto se copia a la estructura de la *Base de Datos* con ayuda de la clase *DataContext* [Johnson 17], para pasar el esquema de la *Base de Datos* a clases y poder trabajar con todo el proyecto con objetos incluidas las consultas que se harán con ayuda de *LINQ*, y el esquema que queda se muestra en la siguiente figura (Figura 3.42).



**Figura 3.42.** Esquema de la Base de Datos mapeada a clases con Visual Studio 2010.

Posteriormente para el desarrollo de las interfaces se utilizan los controles que provee Silverlight como *DataGrid*, *TextBox*, *TextBlock*, *ComboBox*, *RadioButton*, *Button*, entre los más importantes, se utilizaron controles de *Telerik* solo por diseño de la aplicación, pero que funcionan igual que los controles de usuario nativos de Silverlight, el control utilizado fue el *RadGridView* parecido al *DataGrid*. Se realizó la creación de formularios para las distintas vistas.

### Vista inicial de la aplicación.

En la parte inicial de la aplicación, se creó un formulario que consta de cuatro partes, el título en la parte superior, un menú en la parte izquierda, la ventana de Acceso para inicio de sesión, y en caso de que los datos sean correctos, se puede acceder a la última parte del diseño, que es el contenido o área de trabajo a donde se mostraran de manera asíncrona todos los contenidos de cada menú (Figura 3.43).

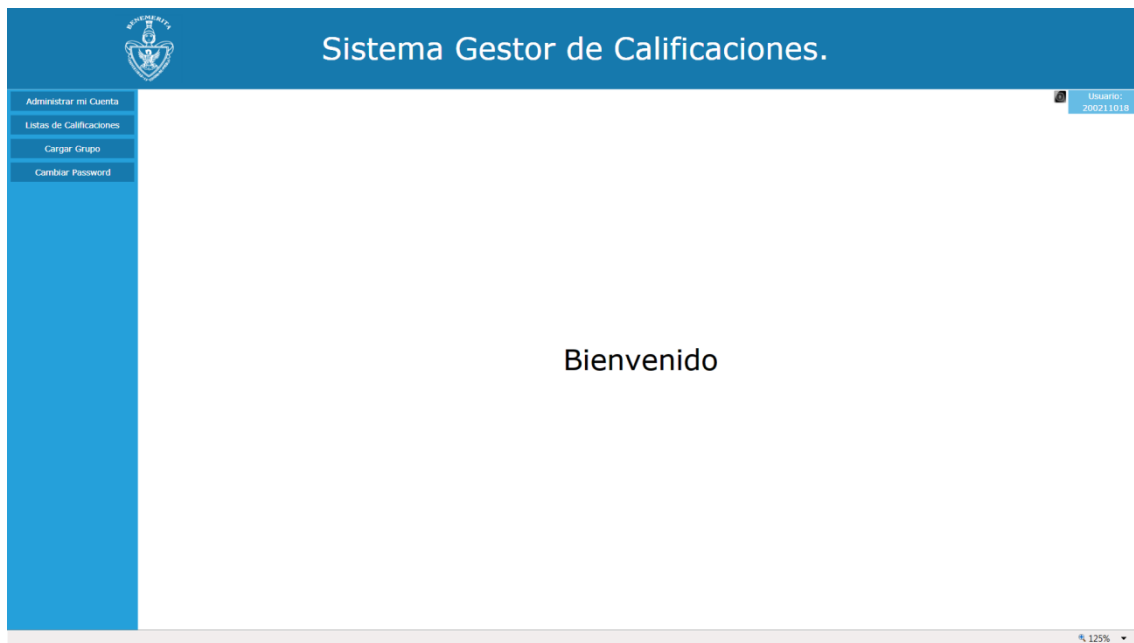


**Figura 3.43.** Vista Inicial, está compuesta por un Grid que se divide en 2 filas, la primera con el título y la segunda dividida en 2 columnas, que contienen el menú de la aplicación y el espacio de trabajo, ahí cambiara todo el contenido y se mostraran

*mensajes, alertas, otras ventanas y controles, como en este caso se muestra la ventana para la autenticación, que es estática y permite validar datos de acceso.*

### **Vista inicial. (Ingreso al sistema).**

Una vez ingresando al sistema se desbloquea el contenido y el menú para poder trabajar en la aplicación, en la parte superior derecha del contenido se mostrara el número de trabajador del usuario que está utilizando la aplicación y un botón para poder salir de la aplicación. Una ventaja de la aplicación realizada con *Silverlight* es que al salir del sistema o cambiar de página web, no existe la opción de regresar para seguir viendo el contenido de la aplicación (Figura 3.44).



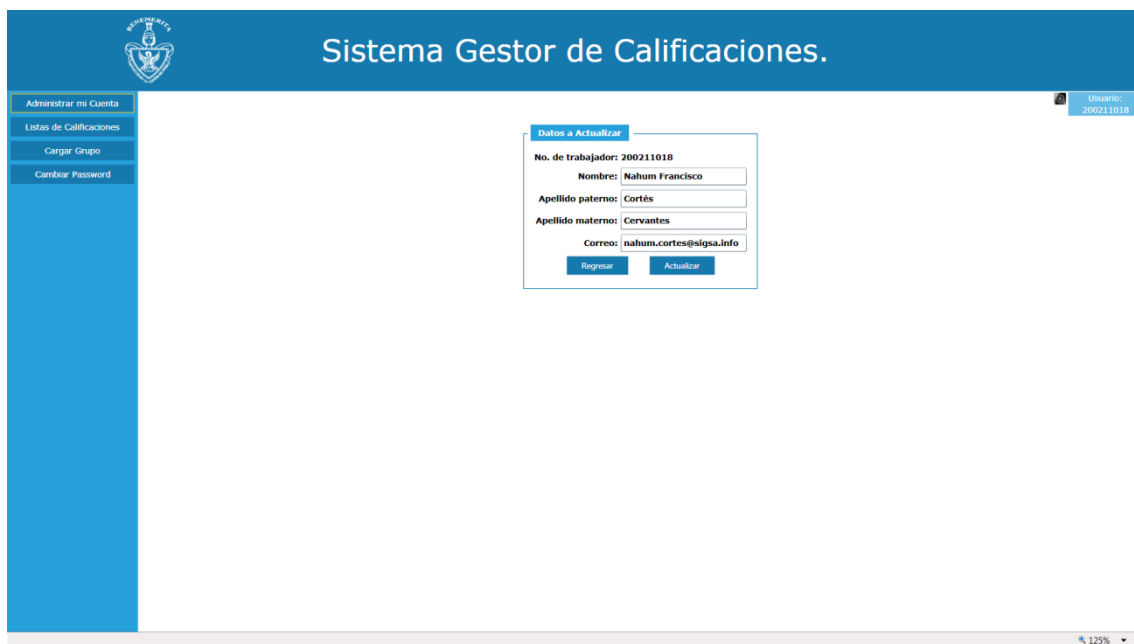
**Figura 3.44.** *Ingreso al sistema, menú habilitado y contenido de la aplicación.*

### **Administrar datos.**

En la opción *Administrar mi Cuenta*, se mostrara un *UserControl* en el contenido de la aplicación, para corroborar nuestros datos (Figura 3.45) o poder cambiarlos, en caso de querer cambiarlos, se podrá hacer con el botón "*Cambiar Datos*" y se mostrara el siguiente *UserControl* (Figura 3.46), en el cual se pueden capturar los datos a modificar después de oprimir el botón "*Actualizar*" o se podrá regresar a la vista anterior con el botón "*Regresar*".



**Figura 3.45.** Muestra de datos generales.



**Figura 3.46.** Cambio de datos.

### **Cargar grupo.**

En esta opción (Figura 3.47) se mostrara otro *UserControl* en el contenido de la aplicación, en la parte superior izquierda, si se quiere cargar un grupo se oprimirá el botón “*Cargar Grupo*” y una nueva ventana se abrirá a donde podremos ubicar el archivo *Excel* proporcionado por la Secretaría Académica, se selecciona el archivo y si está en formato correcto, en la parte central de la aplicación que tiene un control *RadGridView* cargara la lista de alumnos. También pueden cargarse los grupos utilizando la opción *Copiar Datos*, donde se copia el texto en formato predeterminado y se oprime el botón “*Pegar Datos*” y en un control *RadGridView* se cargara la lista de alumnos.

Deben llenarse los controles *TextBox* que están en el apartado *Información Del Grupo* ubicados a la derecha en el área de trabajo y seleccionar el periodo en el *ComboBox*,

si al oprimir el botón “*Subir Lista*”, el *NRC* del grupo ya existe, se mostrara un mensaje con el error correspondiente, en caso de que aún no exista en la *Base de Datos* el *NRC*, el grupo será grabado y se pasara a la siguiente vista o *UserControl* (Figura 3.48) en el cual se debe asignar el porcentaje acordado a cada tipo de calificación, antes debe seleccionarse el *CheckBox* ubicado a la izquierda de cada tipo de calificación para habilitar el *TextBox* y poder actualizar el porcentaje con el botón “*Guardar Cambios*”, se verifica que el total sea igual a 100, de no ser así el botón estará deshabilitado. Si se actualiza el porcentaje de manera correcta entonces el sistema re direcciona al *UserControl* *Asignar Calificaciones* que se describirá posteriormente.

Figura 3.47. Cargar un grupo.

Figura 3.48. Asignar porcentaje.

### Elegir lista de calificaciones.

En la opción *Calificar Grupo*, se mostrara otro *UserControl* en el área de trabajo de la aplicación, en este *UserControl* se cargan dos *ComboBox*, uno con los periodos en los que tenga calificaciones el profesor y otro con todos los grupos guardados en la *Base de Datos* correspondientes al profesor y al periodo seleccionado (Figura 3.49), se selecciona un grupo, se oprime el botón “*Siguiente*” y el sistema re direcciona al *UserControl Calificar Grupo* (Figura 3.50).



Figura 3.49. Elegir grupo.

### Calificar grupo.

En el *UserControl Calificar Grupo* (Figura 3.50) se muestra en la parte central un *RadGridView* con las calificaciones correspondientes de cada alumno del grupo previamente seleccionado, en la parte superior se cuenta con dos *RadioButton*, un *ComboBox* y un *Button* para agregar o quitar columnas a las listas de calificaciones, como máximo por cada tipo de calificación (por las especificaciones del sistema) solo se pueden agregar cuatro columnas. De acuerdo a los parámetros establecidos en la parte de *Asignar Porcentajes*, (si por algún motivo aún no se asigna el porcentaje, antes de pasar a esta parte el sistema re direcciona a *Asignar Porcentajes* (Figura 3.48)).

En la parte inferior del *UserControl* se cuenta con un conjunto de botones que a continuación se describen:

- *Regresar*: permite regresar al *UserControl Elegir Grupo*.
- *Grabar Actualizaciones*: al ser seleccionado se actualizan las calificaciones correspondientes al grupo en la *Base de Datos*.
- *Imprimir*: Muestra una nueva ventana con un reporte del grupo que se está visualizando.
- *Exportar a Excel*: permite guardar el grupo en un formato *Excel XML*.
- *Borrar Grupo*: elimina el grupo que se está evaluando, sin opción a recuperarse (Solo repitiendo todo el proceso, cargar grupo, asignar porcentaje y evaluar), y si la eliminación es exitosa se re direcciona al *UserControl Elegir Grupo*.



Figura 3.50. Calificar grupo.

### Cambiar contraseña.

En la opción *Cambiar Contraseña* (Figura 3.51), se mostrara un *UserControl* en el área de trabajo de la aplicación, se debe ingresar en el campo contraseña actual, la contraseña actual y en las otras dos casillas se debe ingresar la nueva contraseña, y para poder actualizarla se oprime el botón “*Actualizar*”, si la contraseña actual coincide con la que está almacenada en la *Base de Datos*, y la contraseña nueva coincide en los dos campos capturados, se realiza el cambio en la *Base de Datos*.

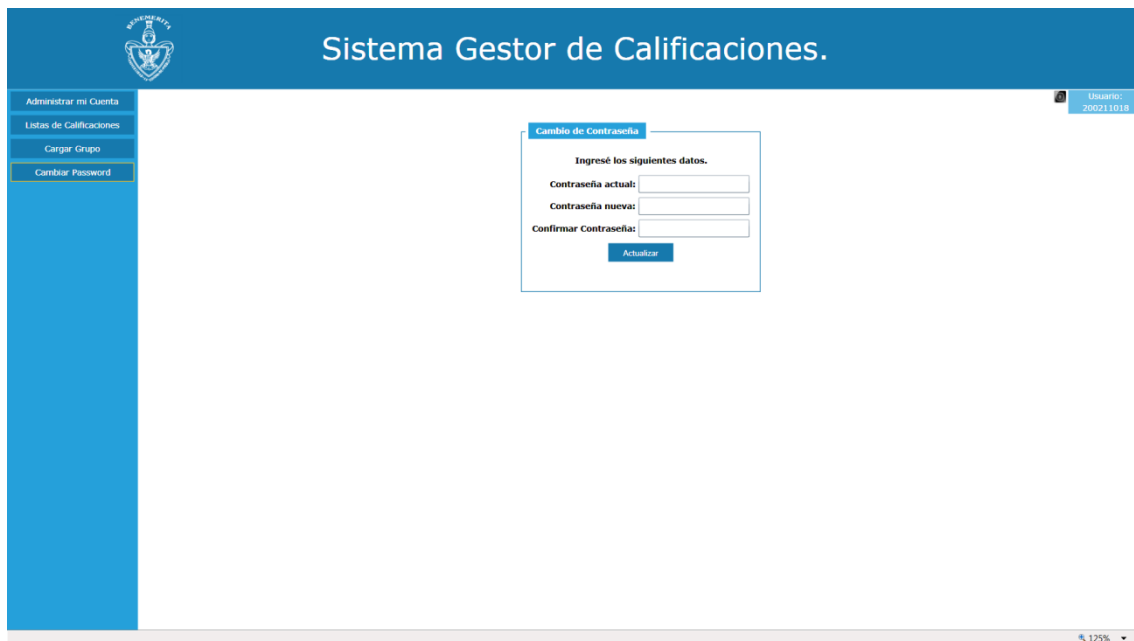
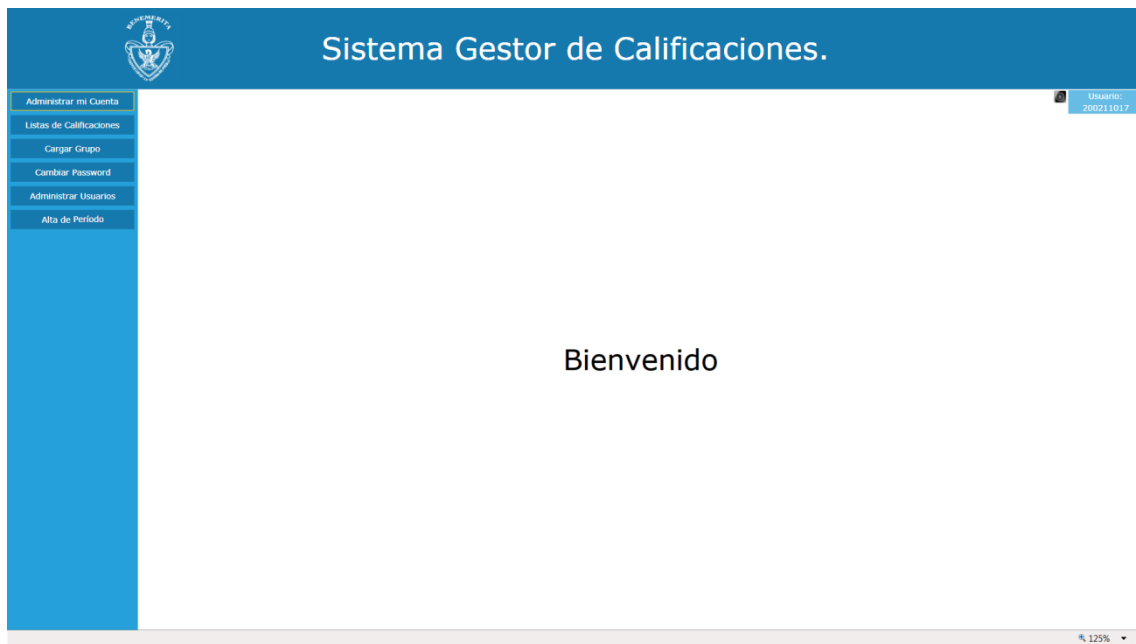


Figura 3.51. Cambiar contraseña.

### Administrar usuarios (Administrador).

En caso de ingresar al sistema con una cuenta de *Administrador*, en el menú aparecerán 2 opciones más, la primera que aquí se describe es la de *Administrar Usuarios* (Figura 3.52).

Al elegir *Administrar Usuarios* se mostrara un *UserControl* en el contenido de la aplicación, este *UserControl* cuenta con un *TabControl* que a su vez tiene tres *TabItems*, el contenido del primero tiene la opción “*Alta de Maestros*” (Figura 3.53), en esta opción se puede copiar o seleccionar la información de un archivo *Excel* con datos de los profesores de la facultad y se mostrara en un *RadGridView* la información que sea dada de alta en la *Base de Datos*, el control consta de un *Textbox* donde se muestra el nombre del archivo, un botón para insertar los registros del archivo, un par de *RadioButtons* para elegir como se va a subir la información, un botón para pegar el contenido de un archivo *Excel*.



**Figura 3.52.** *Administrar usuarios.*



**Figura 3.53.** *Alta de maestros.*

En el siguiente *TabItem* se encuentra la opción de “*Alta Individual*”, al seleccionarlo se mostrara un formulario que debe llenarse con información general del profesor y elegir

el tipo de usuario con un *ComboBox* y un botón “*Aceptar*” para guardar la información (Figura 3.54).

The screenshot shows the 'Sistema Gestor de Calificaciones' interface. The top navigation bar includes 'Administrar mi Cuenta', 'Listas de Calificaciones', 'Cargar Grupo', 'Cambiar Password', 'Administrar Usuarios', and 'Alta de Período'. The 'Alta de Período' menu item is selected. The main content area displays the 'Alta Individual' form, which prompts the user to 'Ingresar los datos del nuevo usuario'. The form contains the following fields: 'No. de trabajador', 'Nombre', 'Apellido paterno', 'Apellido materno', 'Correo', 'Contraseña', and a 'Tipo' dropdown menu. An 'Aceptar' button is located at the bottom of the form. The user's session information 'Usuario: 200211017' is visible in the top right corner.

Figura 3.54. Alta individual.

En el último *TabItem* se encuentra la opción de “*Baja Individual*”, al seleccionarlo, se mostrara un control a donde se debe ingresar el identificador del profesor que se quiere dar de baja, posteriormente se mostrara la información y se podrá dar de baja (Figura 3.55), el control consta de un *TextBox* a donde se ingresa el identificador y un botón “*Borrar*” para eliminar la información del usuario en *Base de Datos* una vez visualizado los datos enfrente del *TextBlock* No. De Trabajador y Nombre.

The screenshot shows the 'Sistema Gestor de Calificaciones' interface with the 'Baja Individual' menu item selected. The main content area displays the 'Baja' form, which prompts the user to 'Ingresar el id del profesor que quiere dar de baja'. The form contains the following fields: 'No. de trabajador', 'No. de Trabajador', and 'Nombre'. A 'Borrar' button is located at the bottom of the form. The user's session information 'Usuario: 200211017' is visible in the top right corner.

Figura 3.55. Baja individual.

#### Alta de período (Administrador).

Esta es la última opción del menú y es para el usuario *Administrador*, al seleccionar esta opción *Alta de Período*, se podrá seleccionar el año del cual se quiere dar de alta

el periodo, que deberá ser el año en curso o el siguiente año, seleccionar el nombre del periodo del *ComboBox* y posteriormente con el botón “Alta”, dar de alta el periodo en la *Base de Datos*, en caso de que el periodo seleccionado ya exista se notificara al usuario (Figura 3.56).

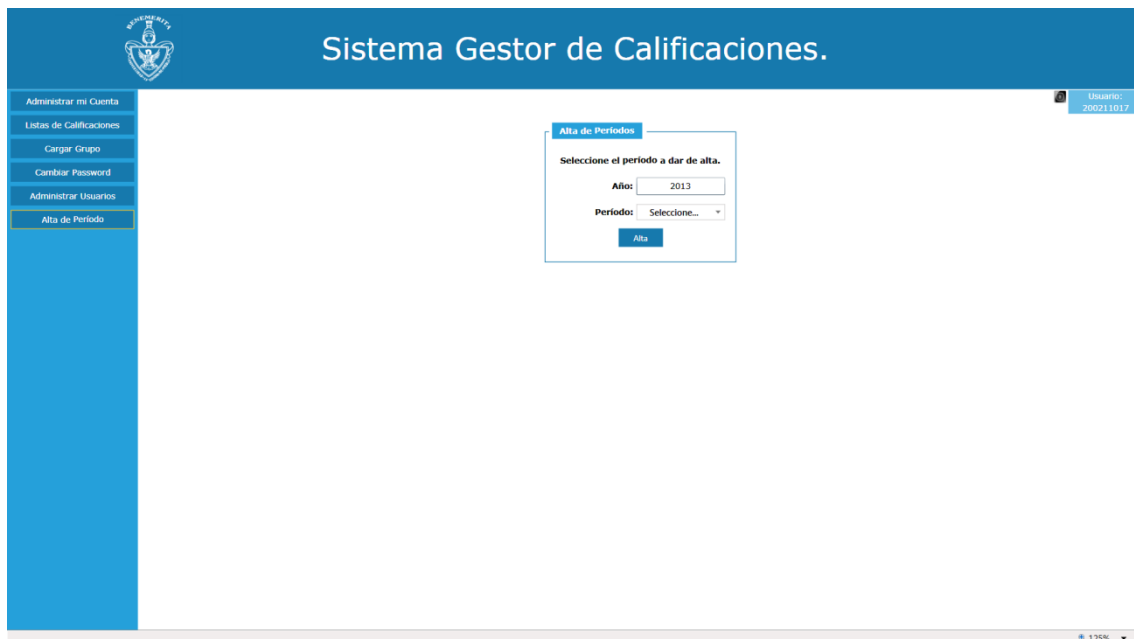


Figura 3.56. Alta de periodo.

### 3.4.2 Servicios realizados.

Los servicios realizados para la implementación del sistema, con metodologías de *WCF* fueron los siguientes:

**GetMD5.-** Dada una cadena, la codifica y regresa la cadena codificada y tiene la opción también de decodificar la cadena, el archivo que se genera al crear el servicio web es *GetMD5.asmx.cs*, la ruta temporal creada para su consulta local es <http://<NombreServidor>/GetMD5.asmx>, en el archivo que se genera, se deben tener las librerías que se requieran, se agrega el servicio web al espacio de nombres del proyecto en el cual se está trabajando, se realiza alguna breve descripción del servicio web, posteriormente se crea la clase del servicio web que debe heredar la funcionalidad *WebService*, dentro de la clase se declaran las variables globales que se utilizaran, y posteriormente las funciones que contendrá el servicio web, estas deben ser declaradas después de la sentencia *[WebMethod]*, a continuación se muestra el código para la implementación de este servicio web.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Services;
using System.Security.Cryptography;
using System.Text;
using System.Configuration;

namespace Proyecto.Web
{
    [WebService(Namespace = "http://tempuri.org/", Description="Servicio que codifica una cadena")]
    [WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
    [System.ComponentModel.ToolboxItem(false)]
    public class GetMD5 : System.Web.Services.WebService
    {
```

```

private const string cryptoKey = "cryptoKey";

private static readonly byte[] IV =
new byte[8] { 240, 3, 45, 29, 0, 76, 173, 59 };
[WebMethod]
public string Encrypt(string s)
{
if (s == null || s.Length == 0) return string.Empty;
string result = string.Empty;
try
{
byte[] buffer = Encoding.ASCII.GetBytes(s);

TripleDESCryptoServiceProvider des =
new TripleDESCryptoServiceProvider();

MD5CryptoServiceProvider MD5 =
new MD5CryptoServiceProvider();

des.Key =
MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(cryptoKey));

des.IV = IV;
result = Convert.ToBase64String(
des.CreateEncryptor().TransformFinalBlock(
buffer, 0, buffer.Length));
}
catch
{
throw;
}

return result;
}

[WebMethod]
public string Decrypt(string s)
{
if (s == null || s.Length == 0) return string.Empty;

string result = string.Empty;

try
{
byte[] buffer = Convert.FromBase64String(s);

TripleDESCryptoServiceProvider des =
new TripleDESCryptoServiceProvider();

MD5CryptoServiceProvider MD5 =
new MD5CryptoServiceProvider();

des.Key =
MD5.ComputeHash(ASCIIEncoding.ASCII.GetBytes(cryptoKey));

des.IV = IV;

result = Encoding.ASCII.GetString(
des.CreateDecryptor().TransformFinalBlock(
buffer, 0, buffer.Length));
}
catch
{
throw;
}

return result;
}
}
}

```

Una vez en el servidor, se pueden visualizar las funciones que contienen (Figura 3.51) y la descripción que se colocó.

# GetMD5

Servicio que encripta una cadena

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [Decrypt](#)
- [Encrypt](#)

**Figura 3.57.** Vista del servicio web publicado (<http://<NombreServidor>/GetMD5.asmx>).

Se puede ver el código en formato *XML* del *Web Service Definition Language*, como a continuación se muestra (<http://<NombreServidor>/GetMD5.asmx?WSDL>).

```
<?xml version="1.0" encoding="utf-8" ?>
<wsdl:definitions xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/" xmlns:tns="http://tempuri.org/"
  xmlns:s="http://www.w3.org/2001/XMLSchema"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/" targetNamespace="http://tempuri.org/"
  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">
  <wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Servicio que codifica una
    cadena</wsdl:documentation>
  <wsdl:types>
    <s:schema elementFormDefault="qualified" targetNamespace="http://tempuri.org/">
      <s:element name="Encrypt">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="s" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="EncryptResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="EncryptResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="Decrypt">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="s" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
      <s:element name="DecryptResponse">
        <s:complexType>
          <s:sequence>
            <s:element minOccurs="0" maxOccurs="1" name="DecryptResult" type="s:string" />
          </s:sequence>
        </s:complexType>
      </s:element>
    </s:schema>
  </wsdl:types>
  <wsdl:message name="EncryptSoapIn">
    <wsdl:part name="parameters" element="tns:Encrypt" />
  </wsdl:message>
  <wsdl:message name="EncryptSoapOut">
```

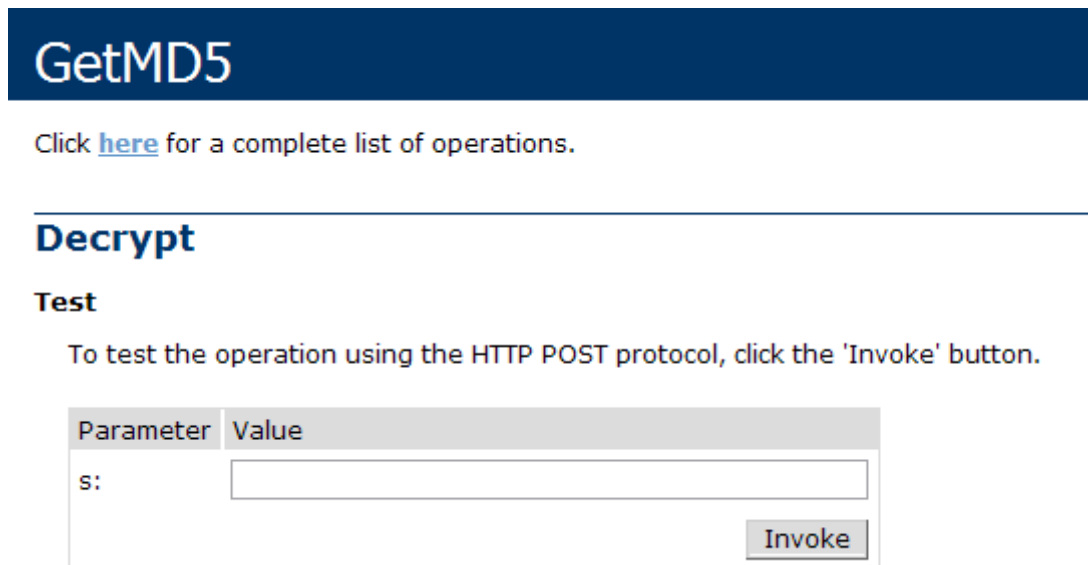
```

<wsdl:partname="parameters" element="tns:EncryptResponse" />
</wsdl:message>
<wsdl:message name="DecryptSoapIn">
<wsdl:partname="parameters" element="tns:Decrypt" />
</wsdl:message>
<wsdl:message name="DecryptSoapOut">
<wsdl:partname="parameters" element="tns:DecryptResponse" />
</wsdl:message>
<wsdl:portType name="GetMD5Soap">
<wsdl:operation name="Encrypt">
<wsdl:inputmessage="tns:EncryptSoapIn" />
<wsdl:outputmessage="tns:EncryptSoapOut" />
</wsdl:operation>
<wsdl:operation name="Decrypt">
<wsdl:inputmessage="tns:DecryptSoapIn" />
<wsdl:outputmessage="tns:DecryptSoapOut" />
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="GetMD5Soap" type="tns:GetMD5Soap">
<soap:bindingtransport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="Encrypt">
<soap:operationsoapAction="http://tempuri.org/Encrypt" style="document" />
<wsdl:input>
<soap:bodyuse="literal" />
</wsdl:input>
<wsdl:output>
<soap:bodyuse="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Decrypt">
<soap:operationsoapAction="http://tempuri.org/Decrypt" style="document" />
<wsdl:input>
<soap:bodyuse="literal" />
</wsdl:input>
<wsdl:output>
<soap:bodyuse="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="GetMD5Soap12" type="tns:GetMD5Soap">
<soap12:bindingtransport="http://schemas.xmlsoap.org/soap/http" />
<wsdl:operation name="Encrypt">
<soap12:operationsoapAction="http://tempuri.org/Encrypt" style="document" />
<wsdl:input>
<soap12:bodyuse="literal" />
</wsdl:input>
<wsdl:output>
<soap12:bodyuse="literal" />
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="Decrypt">
<soap12:operationsoapAction="http://tempuri.org/Decrypt" style="document" />
<wsdl:input>
<soap12:bodyuse="literal" />
</wsdl:input>
<wsdl:output>
<soap12:bodyuse="literal" />
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="GetMD5">
<wsdl:documentation xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/">Servicio que codifica una
cadena</wsdl:documentation>
<wsdl:port name="GetMD5Soap" binding="tns:GetMD5Soap">
<soap:addresslocation="http://localhost:105/GetMD5.asmx" />
</wsdl:port>
<wsdl:port name="GetMD5Soap12" binding="tns:GetMD5Soap12">

```

```
<soap12:addresslocation="http://localhost:105/GetMD5.asmx" />
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

Si se selecciona la opción en la liga principal para decodificar (Figura 3.58) el servicio web se muestra una interfaz en la que se puede probar la función del servicio web y se muestra una descripción en formato XML de la función, al igual que la opción de codificado (Figura 3.59).



**Figura 3.58.** Método Decrypt del servicio web publicado (<http://<NombreServidor>/GetMD5.asmx?op=Decrypt>)

- Protocolo SOAP correspondiente al método Decrypt.

#### SOAP

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /GetMD5.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Decrypt"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<Decrypt xmlns="http://tempuri.org">
<s>string</s>
</Decrypt>
</soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
<soap:Body>
<DecryptResponse xmlns="http://tempuri.org">
```

```
<DecryptResult>string</DecryptResult>
</DecryptResponse>
</soap:Body>
</soap:Envelope>
```

#### HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /GetMD5.aspx/Decrypt HTTP/1.1
Host: localhost
Content-Type: application/x-www-form-urlencoded
Content-Length: length
s=string
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>
<string xmlns="http://tempuri.org/">string</string>
```

## GetMD5

Click [here](#) for a complete list of operations.

### Encrypt

#### Test

To test the operation using the HTTP POST protocol, click the 'Invoke' button.

Parameter	Value
s:	<input type="text"/>

**Figura 3.59.** Método *Decrypt* del servicio web publicado (<http://<NombreServidor>/GetMD5.aspx?op=Encrypt>)

- Protocolo SOAP correspondiente al método *Encrypt*.

#### SOAP 1.1

The following is a sample SOAP 1.1 request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /GetMD5.aspx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://tempuri.org/Encrypt"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <Encrypt xmlns="http://tempuri.org/">
      <s>string</s>
    </Encrypt>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: **length**

```
<?xml version="1.0" encoding="utf-8"?>  
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
xmlns:xsd="http://www.w3.org/2001/XMLSchema"  
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">  
<soap:Body>  
<EncryptResponse xmlns="http://tempuri.org/">  
<EncryptResult>string</EncryptResult>  
</EncryptResponse>  
</soap:Body>  
</soap:Envelope>
```

HTTP POST

The following is a sample HTTP POST request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /GetMD5.aspx/Encrypt HTTP/1.1  
Host: localhost  
Content-Type: application/x-www-form-urlencoded  
Content-Length: length  
s=string  
HTTP/1.1 200 OK  
Content-Type: text/xml; charset=utf-8  
Content-Length: length
```

```
<?xml version="1.0" encoding="utf-8"?>  
<string xmlns="http://tempuri.org/">string</string>
```

De esta manera se realiza la implementación y pruebas de un servicio web, así funcionan los servicios realizados, por lo cual solo se describió a detalle este servicio web y del resto solo se mencionara su funcionalidad.

**GetPorcent.**- Este servicio recibe como parámetro el *NRC* de algún grupo y regresa el porcentaje asignado a cada tipo de calificación para ese grupo.

**ValidaLogin.**- Recibe el número de trabajador del maestro y su contraseña, realiza una validación en la *Base de Datos* utilizando otro servicio de codificación *GetMD5*, y valida si el usuario es válido regresa un valor booleano y una llave para poder modificar la *Base de Datos* posteriormente.

**CargarGrupodeArchivo.**- Recibe como parámetro la ruta del archivo *Excel* que se quiere subir, con el formato adecuado, y regresa en una lista de elementos, la lista para mostrar en pantalla.

**InsertarGrupo.**- Recibe como parámetros una lista con la matrícula y nombre de los alumnos, el número de trabajador del maestro, el *NRC*, sección, nombre de la materia, el tipo de inserción, el porcentaje de cada tipo de calificación, periodo y llave, si el campo tipo de inserción es igual a grupo, se verifica si la materia existe, si es así regresa un valor booleano falso, en caso contrario se procede con la inserción del grupo y se regresa como valor booleano verdadero. En caso de que el tipo de inserción sea porcentajes, se asigna el porcentaje a cada tipo de calificación y se regresa como valor booleano verdadero, en caso contrario se regresa como valor booleano falso, ya que no se realizó alguna actualización en la *Base de Datos*.

**GetDatos.**- Dado el identificador de trabajador del profesor, regresa una lista con sus datos personales, si existe alguna excepción al realizar la consulta regresa un elemento vacío.

**UpDateDatos.**- Dado el identificador del profesor, su nombre, apellido paterno, apellido materno, correo electrónico y llave, se realiza una actualización de los datos recibidos en la *Base de Datos* a excepción del número de trabajador, en caso de que la actualización haya sido exitosa se regresa un valor booleano verdadero, en caso contrario un valor falso.

**UpDatePass.**- Dado un número de trabajador, su contraseña y la llave, se realiza la actualización de la contraseña en la *Base de Datos* utilizando el servicio de codificación *GetMD5*, si la actualización es exitosa se regresa un valor booleano verdadero, en caso contrario falso.

**GetMateria.**- Dado el identificador de un maestro y el de un período, se devuelve una lista con los grupos cargados por el maestro.

**UpDateCalifGrupo.**- Dado un *NRC*, una lista de calificaciones y una llave, inicializa en la *Base de Datos* la calificación, para determinar cómo se va a calificar el grupo, regresa un valor booleano verdadero si se actualiza la información, en caso contrario un valor falso.

**UpDateGrupo.**- Dado un *NRC*, una lista de calificaciones y una llave, realiza la actualización de las calificaciones y calcula de acuerdo a los criterios establecidos la calificación final, regresa la lista de calificaciones actualizada.

**GetGrupo.**- Dado un *NRC*, el sistema regresa la lista de calificaciones correspondiente a él.

**InsertColum.**- Dado un *NRC*, el nombre del campo que se quiere añadir y una llave, el sistema inserta 0's en la columna deseada para habilitarla en el sistema, como resultado, regresa la lista actualizada, consultando al servicio *GetGrupo*.

**DeleteColum.**- Dado un *NRC*, el nombre del campo que se quiere quitar y una llave, el sistema inserta valores negativos en la columna deseada para deshabilitarla en el sistema, como resultado, regresa la lista actualizada, consultando al servicio *GetGrupo*.

**DeleteGroup.**- Dado un *NRC* y una llave, se elimina en la *Base de Datos* el grupo con el *NRC* correspondiente, en caso de eliminación exitosa el servicio regresa un valor booleano verdadero, en caso contrario un valor falso.

**RecuperaPass.**- Dado un identificador de maestro, manda un correo al correo registrado en la *Base de Datos* con la contraseña para poder ingresar nuevamente al sistema en caso de olvidar la contraseña.

**InsertUser.**- Dado los datos de un maestro y una llave, se realiza la inserción en la *Base de Datos* con los datos del profesor, la contraseña se guarda codificada, regresa un valor booleano, verdadero en caso de insertarse con éxito y falso en caso de ya existir el profesor u ocurrir algún error de conexión que no permita realizar la inserción.

**DeleteUser.**- Dado el identificador de un maestro y una llave, se da de baja de la *Base de Datos*, si se borra la información regresa un valor booleano verdadero, en caso contrario un valor falso.

**InsertarListadeMaestros.**- Dada la ruta de un archivo *Excel* y una llave, en el sistema se realiza la inserción de los profesores que estén en la lista utilizando el servicio *InsertUser*, la función regresa una lista con la información de los profesores que se

insertaron, adicionalmente utiliza otra función *VerificaCorreos*, que recibe 2 campos, el correo institucional en el primero y otros correos que el profesor tenga en el otro campo, se guarda solo uno dependiendo de qué campo tenga valor, se da prioridad a guardar el correo institucional.

**InsertarListadeMaestros.**- Dada una lista de maestros y una llave, se realiza la inserción de ellos en la *Base de Datos*, todos con un rol de *Usuario*, en caso de que la inserción sea exitosa, se regresa la lista de maestros insertadas para ser mostrada en pantalla, en caso contrario se regresa una lista vacía. Si algún usuario que se reciba en la lista, ya está dado de alta, se omite en la lista de vuelta ya que no se vuelve a insertar.

**InsertarPeriodo.**- Dado un año, un nombre de periodo y una llave, se realiza el alta de un nuevo periodo, en caso de que el alta se realice con éxito, se devuelve un valor booleano verdadero, en caso de que ya exista el periodo un valor booleano falso ya que no se insertó.

**ObtenPeriodos.**- Obtiene todos los periodos dados de alta, regresa la lista de periodos, en caso de alguna excepción, regresa una lista vacía.

**ObtenPeriodosPorProfesor.**- Dado el número de trabajador de un profesor, se devuelve una lista de todos los periodos en los cuales el profesor tenga grupos dados de alta, en caso de excepción o de no encontrar se devuelve una lista vacía.

## 4. Conclusión y trabajo futuro.

### 4.1. Conclusión.

Los servicios web forman una parte esencial en estos días para el desarrollo de aplicaciones, no solo a nivel de aplicaciones web, también en dispositivos móviles y aplicaciones de escritorio.

Ahora no existe una tecnología estándar y estable en el mundo para poder tener aplicaciones ricas en contenido y diseño para cualquier tipo de dispositivo (Computadoras de escritorio, tabletas y teléfonos inteligentes), por este motivo los servicios web toman importancia, ya que para desarrollar una aplicación para un dispositivo móvil, escritorio o página web, es necesario crear una aplicación para cada tipo de dispositivo, por ejemplo:

- Para la parte web, se pueden utilizar tecnologías como *.NET*, *JSP*, *PHP*, *Javascript*, etc.
- Para la parte de escritorio, dependiendo del sistema operativo se utilizara un programa para generar la aplicación, por mencionar algunos *Java*, *WPF* o *C++*.
- Para los teléfonos inteligentes o tabletas, también depende el sistema que utilice ya sea *Windows Phone*, *Android*, *BlackBerry OS* o *iOS*, por mencionar los dispositivos más comerciales.

La importancia de contar con un servicio web o un conjunto de servicios disponibles en internet, es que gracias a los estándares que manejan, se puede reutilizar cada uno de estos servicios en la aplicación que queramos para el dispositivo que se desee, así podemos tener la mayor parte de la funcionalidad de la aplicación englobada en los servicios web y solo diseñar con la herramienta adecuada al dispositivo la interfaz que consume los servicios, ahorrando trabajo y consecuentemente dinero y esfuerzo, ampliando la tecnología que desee distribuirse a un amplio mercado hasta poder cubrirlo todo.

## 4.2. Trabajo futuro.

Se cuenta ya con una aplicación web dinámica para la gestión de calificaciones, que puede ser visible en cualquier computadora personal de escritorio y laptop con conexión a internet, pero aún se tiene la limitación de no poder visualizarse en tabletas y teléfonos inteligentes.

Al ser una aplicación robusta de captura de calificaciones, no es conveniente que se pueda utilizar con todas sus funciones en un dispositivo pequeño, pero si puede servir en un dispositivo móvil para consulta, consumiendo los servicios existentes, realizando más servicios en base a ellos o realizar nuevas funcionalidades como las notificaciones o alertas en teléfonos inteligentes.

La aplicación puede escalarse para cualquier institución educativa que tenga un sistema de calificaciones igual o parecido al de la BUAP. Y también puede ampliarse para poder ser utilizada por los estudiantes.

## 5. Bibliografía.

- [1] Georgakopoulos Dimitrios, Papazoglou Michael P. *Service-Oriented Computing*, 2009, Cambridge, Massachusetts. London, England. Massachusetts Institute of Technology.
- [2] Erl Thomas. *Service-Oriented Architecture: Concepts, Technology, and Design*, Estados Unidos de América, 2005, Prentice Hall.
- [3] M. Arboleda Liliana. *Servicios WEB: Distribución e integración*, Universidad Icesi, vol, p. 107-120, 2004. Sistemas & Telemática.
- [4] Rotem-Gal-Oz Arnon. *SOA Patterns*, 2012, Estados Unidos de América, MANNING.
- [5] Roshen Waseem. *SOA-Based Enterprise Integration*, 2009, Estados Unidos de América, Mc Graw Hill.
- [6] Manouvrier Bernard, Ménard Laurent. *Application Integration: EAI, B2B, BPM and SOA*, 2008, Gran Bretaña, Estados Unidos de América, WILEY.
- [7] Cummins Fred A. *Building the Agile Enterprise with SOA, BPM and MBM*, 2009, Estados Unidos de América, Elsevier.
- [8] Web completa de XML, <http://www.xml.com/>
- [9] Web de auto aprendizaje tema SOAP, <http://www.w3schools.com/soap>
- [10] Web de auto aprendizaje tema WSDL, <http://www.w3schools.com/wsdl>
- [11] Web de auto aprendizaje tema UDDI, <http://www.tutorialspoint.com/uddi/>
- [12] Henning Michi, Vinoski, Steve. *Advanced CORBA Programming with C++*, Primera Edición, 1999, Estados Unidos de América, Addison-Wesley.
- [13] Deitel Paul, Deitel Harvey. *C# 2010 FOR PROGRAMMERS*, cuarta edición, Estados Unidos de América, 2010.
- [14] Stoecker Matthew. *Windows Applications Development with Microsoft .NET Framework 4*, Estados Unidos de América, 2011.
- [15] Löwy Juval. *Programming WCF Services*, tercera edición, 2010, Estados Unidos de América, O'Reilly.
- [16] Johnson Bruce, Madziak Peter, Morgan Sara. *Microsoft .NET Framework 3.5 Windows Communication Foundation*, 2009, Estados Unidos de América, Microsoft Press.
- [17] Johnson Glenn. *Accessing Data with Microsoft .NET Framework 4.0*, 2011, Estados Unidos de América, Microsoft Press.
- [18] Northrup Tony, Snell Mike. *Web Applications Development with Microsoft .NET Framework 4*, 2011, Estados Unidos de América, Microsoft Press.

- [19] Sharp, John. *Microsoft® Visual C#® 2010, Step by Step*, 2010, Estados Unidos de América, Microsoft Press.
- [20] Aitchison Alastair, Machanic Adam. *Expert SQL Server 2008 Development*, 2009, Estados Unidos de América, Apress.
- [21] Beaulieu Alan. *Learning SQL*, segunda edición, 2009, Estados Unidos de América, O'Reilly.
- [22] Piaorsi Paolo, Russo Marco. *Programming Microsoft LINQ in Microsoft .NET Framework 4*, 2010, Estados Unidos de América, Microsoft Press.
- [23] Deitel Harvey M, Deitel Paul J. *Como Programar en Java*, quinta edición, 2004, Prentice Hall.
- [24] Goodman Danny, Eich Brendan. *JavaScript Bible*, Gold Edition, 2001, Estados Unidos de América, Hungry Minds.
- [25] Pfaffenberger Bryan, Schafer Steven M, White Chuck, Karow Bill. *HTML, XHTML, and CSS*, tercera edición, 2004, Estados Unidos de América, WILEY.

## 6. Apéndices.

### A. Acrónimos.

- *.xap* (*XAML Application*) Aplicación Zammel.
- *API* (*Application Programming Interface*) Interfaz de programación de aplicaciones.
- *App* (*Application*) Aplicación.
- *ASP* (*Active Server Pages*) Páginas activas del servidor.
- *Aspx* (*Active Server Pages XML*) Páginas activas del servidor utilizando formatos web.
- *BPEL* (*Business Process Execution Language*) Lenguaje de ejecución de procesos de negocio.
- *BPM* (*Bussines Process Management*) Sistema de administración de procesos.
- *BUAP* Benemérita universidad autónoma de puebla.
- *CORBA* (*Common Object Request Broker Architecture*) Arquitectura común de intermediarios en peticiones a objetos.
- *CS* (*Class*) Archivo de tipo clase.
- *DAE* Dirección de administración escolar.
- *DTD* (*Document Type Definition*) Definición de tipo de documento.
- *DTD* (*Document Type Definition*) Definición de tipo de documento.
- *EAI* (*Enterprise Application Integration*) Integración de aplicaciones empresariales.
- *EbXML* (*Electronic Business XML*) Negocios electrónicos con XML.
- *FCC* Facultad de ciencias de la computación.
- *FK* (*Foreign Key*) Llave foranea.
- *GUI* (*Graphical User Interface*) Interfaz gráfica de usuario.
- *HTTP* (*Hypertext Transfer Protocol*) Protocolo de transferencia de hipertexto.
- *IBM* (*International Business Machines*) Máquinas de oficina internacionales.
- *IDE* (*Integrated Development Environment*) Entorno de desarrollo integrado.
- *IDL* (*Interface Definition Language*) Lenguaje de definición de

•	JSP	(JavaServer Pages)	interfaz. Servidor de páginas java.
•	LINQ	(Language Integrated Query)	Lenguaje de consulta estructurado.
•	MDI window	(Multiple Document Interface Window)	Interfaz de múltiples documentos en ventana.
•	MOM	(Middleware-Oriented Messages)	Software intermedio orientado a mensajes.
•	NRC		Número de referencia del curso.
•	ORB	(Object Request Broker)	Corredor de peticiones de objetos.
•	PC	(Personal Computer)	Computadora personal.
•	PDA	(Personal Digital Assistant)	Asistente digital personal.
•	RMI	(Remote Method Invocation)	Invocación de método remoto
•	RPC	(Remote Procedure Call)	Llamada a procedimiento remoto.
•	SOA	(Service-Oriented Architecture)	Arquitectura orientada a servicios.
•	SOAP	(Simple Object Access Protocol)	Protocolo de acceso de objeto simple.
•	SOC	(Service-Oriented Computing)	Computo orientado a servicios.
•	SQL	(Structured Query Language)	Lenguaje de consulta estructurado.
•	UDDI	(Universal Discovery, Description and Integration)	Descubrimiento universal, descripción e integración.
•	USB	(Universal Serial Bus)	Bus universal en serie.
•	W3C's	(World Wide Web Consortium's)	Consortio de la red informática mundial.
•	WPF	(Windows Presentation Foundation)	Capa de presentación básica de Windows.
•	WSDL	(Web Services Definition Language)	Lenguaje para la descripción de un servicio web.
•	WWW	(World Wide Web)	Red informática mundial.
•	XAML	(eXtensible Application Markup Language)	Lenguaje extensible de formato para aplicaciones –Zammel.
•	XML	(eXtensible Markup Language)	Lenguaje extensible de marcas.
•	XPath	(XML Path Language)	Lenguaje para acceder a partes de un documento XML.
•	XSL	(Extensible Stylesheet Language)	Lenguaje extensible de hojas de estilo.
•	XSL-FO	(XSL Formatting Objects)	Vocabulario XML utilizado para describir el formato de los documentos XML.
•	XSLT	(XSL Transformations)	Lenguaje para transformar documentos XML en otros documentos.

## B. Definiciones.

- **.aspx**: Extensión de las páginas web realizadas con el marco de desarrollo de .NET.
- **.NET**: Es un marco de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones.

- **.xap**: Aplicación XAML empaquetada, que se genera al compilar una aplicación Silverlight o WPF.
- **3D**: Visualización de un objeto en 3 dimensiones.
- **Adobe Flash**: Interfaz que utiliza gráficos vectoriales y gráficos rasterizados, sonido, código de programa, flujo de vídeo y audio bidireccional.
- **Android**: Sistema operativo basado en Linux y respaldado por la empresa Google para teléfonos inteligentes y tabletas.
- **API**: Conjunto de funciones y procedimientos de apoyo para desarrollo de software.
- **Aplicación**: Es un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos.
- **ASP.NET**: Marco para desarrollar aplicaciones web desarrolladas y comercializadas por Microsoft.
- **BANNER**: Herramienta de software para gestión administrativa en la BUAP.
- **Base de Datos relacional**: una Base de Datos que cumple con el modelo relacional.
- **Base de Datos**: Conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- **BlackBerry OS**: Sistema operativo móvil desarrollado por Research In Motion para sus dispositivos BlackBerry.
- **BPEL**: Lenguaje común normalizado que tiene como objetivo principal optimizar la gestión de los procesos empresariales cada vez más complejos.
- **BPM**: Conjunto de servicios y herramientas que facilitan la administración de procesos de negocio: análisis, definición, ejecución, monitoreo, y control de los procesos.
- **C#**: Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET
- **C**: Lenguaje de programación orientado a la implementación de Sistemas Operativos.
- **C++**: Lenguaje de programación diseñado con la intención de extender al exitoso lenguaje de programación C con mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, el C++ es un lenguaje híbrido.
- **Clases E numerables**: En programación es una clase que puede ser recorrida por un ciclo.
- **Clases**: Es la estructura de un objeto, es decir, la definición de todos los elementos de que está hecho un objeto.
- **Ciente-Servidor**: Modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, que le da respuesta.
- **CORBA**: Es un estándar que permite que diversos componentes de software escritos en múltiples lenguajes de programación y que corren en diferentes computadoras puedan trabajar juntos.
- **Crystal Report**: Es una aplicación de inteligencia empresarial utilizada para diseñar y generar informes desde una amplia gama de fuentes de datos.
- **DataContext**: Modelo de datos mapeado a clases dentro de la estructura de un programa con tecnología .NET.
- **Decodificar**: la acción de decodificar una clave.
- **DirectX**: Es una colección de API desarrolladas para facilitar complejas tareas relacionadas con multimedia, especialmente programación de juegos y vídeo.
- **Document Type Definition (DTD)**: Es una descripción de estructura y sintaxis de un documento XML, su función básica es la descripción de la estructura de

datos, para usar una estructura común y mantener la consistencia entre todos los documentos que utilicen la misma DTD. De esta forma, dichos documentos pueden ser validados, conocen la estructura de los elementos y la descripción de los datos que trae consigo cada documento, y pueden además compartir la misma descripción y forma de validación dentro de un grupo de trabajo que usa el mismo tipo de información.

- **EAI:** Es el intercambio de datos y procesos de negocios, sin restricciones, a través de aplicaciones en red, o fuentes de datos, en una organización.
- **e-business:** Se refiere a las actividades y prácticas de gestión empresarial resultantes de la incorporación a los negocios de las tecnologías de la información y la comunicación generales y particularmente de Internet.
- **EbXML:** Es una iniciativa internacional establecida por el CEFC (Centre for Trade Facilitation and Electronic Business) de las Naciones Unidas y por OASIS para definir el marco tecnológico sobre el que XML puede ser normalizado.
- **Eclipse:** Entorno de desarrollo integrado de código abierto multiplataforma para desarrollar "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.
- **Codificar:** es la acción de proteger información para que no pueda ser leída sin una clave.
- **Excel:** es una aplicación distribuida por Microsoft office para hojas de cálculo.
- **Extensible Stylesheet Language (XSL):** Familia de lenguajes basados en el estándar XML que permite describir cómo la información contenida en un documento XML cualquiera debe ser transformada o formateada para su presentación en un medio.
- **Formulario:** Colección de datos a visualizar en una interfaz de usuario.
- **GUI:** Programa informático que actúa de interfaz de usuario.
- **Hardware:** En una computadora, se refiere a todos los componentes físicos que lo conforman.
- **http:** Protocolo usado en cada transacción de la World Wide Web.
- **IDL:** Lenguaje de especificación de interfaces que se utiliza en software de computación distribuida. Ofrece la sintaxis necesaria para definir los procedimientos o métodos que queremos invocar remotamente.
- **IntelliSense:** Permite autocompletar en tiempo de programación el código de una aplicación que se está desarrollando.
- **Interfaz IEnumerable:** En programación, interfaz que expone un enumerador para poder ser iterada por un ciclo sobre una colección de datos.
- **Interfaz IQueryable:** En programación, proporciona funcionalidad para evaluar consultas con respecto a un origen de datos concreto en el que se especifica el tipo de los datos.
- **Interfaz:** En informática, esta noción se utiliza para nombrar a la conexión física y funcional entre dos sistemas o dispositivos de cualquier tipo dando una comunicación entre distintos niveles. A nivel de usuario es la vista que se tiene del Software.
- **Internet:** Conjunto descentralizado de redes de comunicación interconectadas que utilizan la familia de protocolos TCP/IP, garantizando que las redes físicas heterogéneas que la componen funcionen como una red lógica única, de alcance mundial.
- **Intranet:** Red de ordenadores privados que utiliza tecnología Internet para compartir dentro de una organización parte de sus sistemas de información y sistemas operacionales.
- **iOS:** Sistema operativo de la empresa Apple para iPhone y iPad.
- **Iron-Python:** Es una implementación del intérprete Python (CPython) escrita totalmente en C# que se integra con el marco de Microsoft .NET.

- **IronRuby:** es una implementación del lenguaje de programación Ruby, enfocado al marco de Microsoft .NET.
- **JAVA:** Lenguaje de programación de alto nivel Orientado a Objetos.
- **JavaScript:** Es un lenguaje de programación que permite a los desarrolladores crear acciones en sus páginas web.
- **Jedit:** Editor de texto libre.
- **JSP:** Tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo.
- **Librería de Clases de .NET Framework:** Es una biblioteca de clases, interfaces y tipos de valor que se incluye en Microsoft .NET Framework SDK auxiliar en el desarrollo de aplicaciones dentro del entorno.
- **LINQ:** Lenguaje de consulta de datos con el modelo de Programación Orientada a Objetos, que agrega capacidades de consulta a datos de manera nativa a los lenguajes .NET.
- **Log In:** Término que se usa en computación para referirse al ingreso a las cuentas de usuario, a los sistemas o servicios.
- **MDI Window:** Tecnología que permite mostrar múltiples interfaces a la vez.
- **Microsoft Expression Blend:** Herramienta para desarrollar interfaces y animaciones para el entorno de programación de .NET.
- **Middleware:** Es un software que conecta componentes de software por lo regular en un ambiente distribuido o aplicaciones para que puedan intercambiar datos. Esto incluye servidores web, servidores de aplicaciones, sistemas de gestión de contenido y herramientas similares. Middleware es especialmente esencial para tecnologías como XML, SOAP, servicios web y arquitecturas orientada a servicios.
- **Modelo-Vista-Controlador:** Modelo para separar los datos de una aplicación, la interfaz de usuario, y la lógica de negocio en tres componentes distintos.
- **MOM:** Infraestructura de Hardware o Software que soporta el envío y recepción de mensajes entre sistemas distribuidos.
- **Mono:** Software que provee las herramientas necesarias para desarrollar y ejecutar aplicaciones .NET Cliente-Servidor en diferentes plataformas.
- **MySQL:** Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario.
- **Navegador Web:** Es una aplicación que opera a través de Internet, interpretando la información de archivos y sitios web para que estos puedan ser leídos.
- **NRC:** Identificador de las materias asignadas en la BUAP.
- **Oracle:** Sistema de gestión de Base de Datos objeto-relacional (o ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation.
- **ORB:** Capa intermedia de Software que permite a los objetos realizar llamadas a métodos situados en máquinas remotas.
- **PC:** Computadora personal.
- **PDA:** Agenda electrónica de bolsillo.
- **Plug-in:** Es una aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.
- **PostgreSQL:** Es un Sistema de Gestión de Bases de Datos relacional orientado a objetos y libre.
- **Programación Orientada a Objetos:** Paradigma de programación que usa los objetos en sus interacciones, para diseñar aplicaciones y programas informáticos.
- **RMI:** Es un software que permite la invocación de métodos remotos utilizando el lenguaje Java.

- **RPC:** Protocolo de comunicación que permite ejecutar código en otras máquinas si preocuparse por la comunicación.
- **Silverlight:** Estructura para aplicaciones web que agrega nuevas funciones multimedia como la reproducción de vídeos, gráficos vectoriales, animaciones e interactividad, en forma similar a lo que hace Adobe Flash.
- **Sistema Distribuido:** Conjunto de computadoras separadas físicamente pero conectadas entre sí por una red, en la que cada computadora percibe como un solo sistema los recursos locales, como los recursos de las otras computadoras de la red.
- **Sistema Operativo:** Conjunto de programas que permiten interactuar con un dispositivo de Hardware.
- **SOAP:** Protocolo estándar que define cómo dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML.
- **SOC:** Paradigma de computación que sitúa a los servicios como componentes software fundamentales, expuestos a través de interfaces en red, neutrales a plataformas y lenguajes de programación, y que permiten la composición de aplicaciones distribuidas, posiblemente complejas, a partir de componentes débilmente acoplados.
- **Sockets:** Método de comunicación entre un programa cliente y un servidor.
- **Software:** Es el equipamiento lógico o soporte lógico de un sistema informático.
- **SQL:** Lenguaje de consulta estructurado para consulta o modificaciones en bases de datos.
- **SQLServer:** Sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional.
- **SubmitChanges:** Acción para subir cambios.
- **UDDI:** Es el registro de un catálogo que se hace en XML. UDDI es una iniciativa industrial abierta, entroncada en el contexto de los servicios web.
- **USB:** Es un estándar industrial desarrollado en los años 1990 que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre ordenadores y periféricos y dispositivos electrónicos.
- **UserControl:** Parte de una aplicación WPF o Silverlight, que sirve para dividir la aplicación o reutilización de controles para obtener mayor rendimiento.
- **Vectores:** estructura para representar una colección de datos.
- **Visual Basic:** Lenguaje de programación dirigido por eventos.
- **Visual Studio:** Entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta varios lenguajes de programación tales como Visual C++, Visual C#, Visual J#, y Visual Basic .NET, al igual que entornos de desarrollo web como ASP.NET. aunque actualmente se han desarrollado las extensiones necesarias para muchos otros.
- **Windows Phone:** Sistema operativo de teléfonos inteligentes de la compañía Microsoft.
- **Windows:** Sistema operativo de Microsoft.
- **World Wide Web Consortium's (W3C's):** Consorcio internacional que produce recomendaciones para la World Wide Web.
- **WPF:** Modelo de programación de Microsoft para aplicaciones de escritorio, que separa la interfaz de la lógica, con un lenguaje parecido a XML y los lenguajes propios de Microsoft (C# y Visual Basic).
- **WSDL:** Formato XML que se utiliza para describir servicios web.
- **XAML:** Lenguaje de formato para la interfaz de usuario para aplicaciones WPF y Silverlight.

- **XML:** Lenguaje de marcas que permite estandarizar la comunicación en la internet.
- **XPath:** Es un lenguaje que permite construir expresiones que recorren y procesan un documento XML.
- **XSL-FO:** es un documento XML en el que se especifica cómo se van a formatear los datos para presentarlos en pantalla, papel u otros medios.
- **XSLT:** Estándar de la organización W3C que presenta una forma de transformar documentos XML en otros e incluso a formatos que no son XML.

## C. Manual de usuario.

En el siguiente manual se muestra la funcionalidad de la aplicación y como utilizarla.

### C.1. Página principal.

Ingresa a la liga donde esta publicada la aplicación <http://<NombreDelServidor>/PortalEscolar/>, en la pantalla principal se muestra un banner con el título del proyecto, el logo de la Universidad y una ventana de acceso (Figura C.1.1).



**Figura C.1.1.** Vista inicial de la aplicación.

### C.2. Validando datos.

Para ingresar al sistema es necesario estar previamente registrado, con un número de trabajador y una contraseña, una vez que los datos son ingresados se validan (FiguraC.2.1).



**Figura C.2.1.** Descripción del acceso.

A continuación se describen los elementos del acceso.

- Número de Trabajador.- Identificador que cada maestro tiene como parte de la institución y que debe conocer.
- Contraseña.- Contraseña previamente establecida.
- Botón Ingreso.- Al oprimirlo se realiza la validación de los datos.
- Mensaje de Notificación.- El mensaje muestra la acción a realizar o información que es requerida, al inicializarse pide el ingreso de un usuario y su contraseña, en caso de que se intente ingresar, falten datos o sean incorrectos el mensaje será mostrado, en caso de que los datos sean correctos se cierra la ventana y se ingresa al sistema.
- Botón Olvido Contraseña.- Al no recordar la contraseña, debe oprimir el botón y mostrara una ventana (Figura C.2.2) con un formulario a donde se debe ingresar el número de trabajador y oprimir el botón de *Enviar*, si el id es correcto, recibirá en la cuenta de correo registrada un correo con su actual contraseña para acceder al sistema y esto se mostrara en una alerta (Figura C.2.3), en caso de no recordar su número de trabajador no tendrá la opción de recuperar su contraseña (Figura C.2.4).

**Figura C.2.2.** Olvido contraseña.

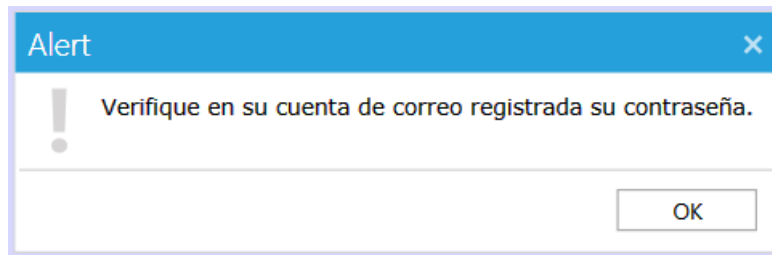


Figura C.2.3. Mensaje de envío exitoso.

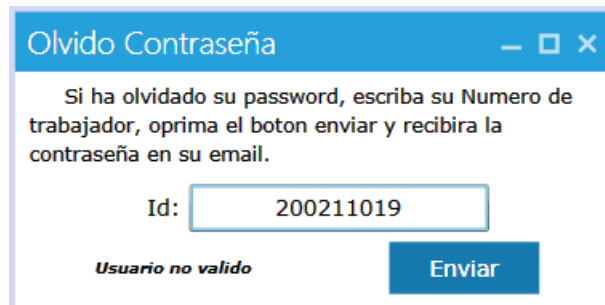


Figura C.2.4. El usuario no está registrado.

### C.3. Ingreso al sistema.

Al haber ingresado al sistema se verá un mensaje de bienvenida en la parte central de la página y se podrá visualizar el menú, existen dos tipos de usuarios el "Administrador" y el "Usuario" con menos privilegios en caso de ser un "Usuario" la pantalla inicial solo tendrá cuatro elementos en el menú principal que son a los que tendrá acceso (Figura C.3.1) y el usuario "Administrador" cuenta con dos opciones más en el menú (Figura C.3.2), que es la *Administración de Usuarios* y *Alta de Períodos*.

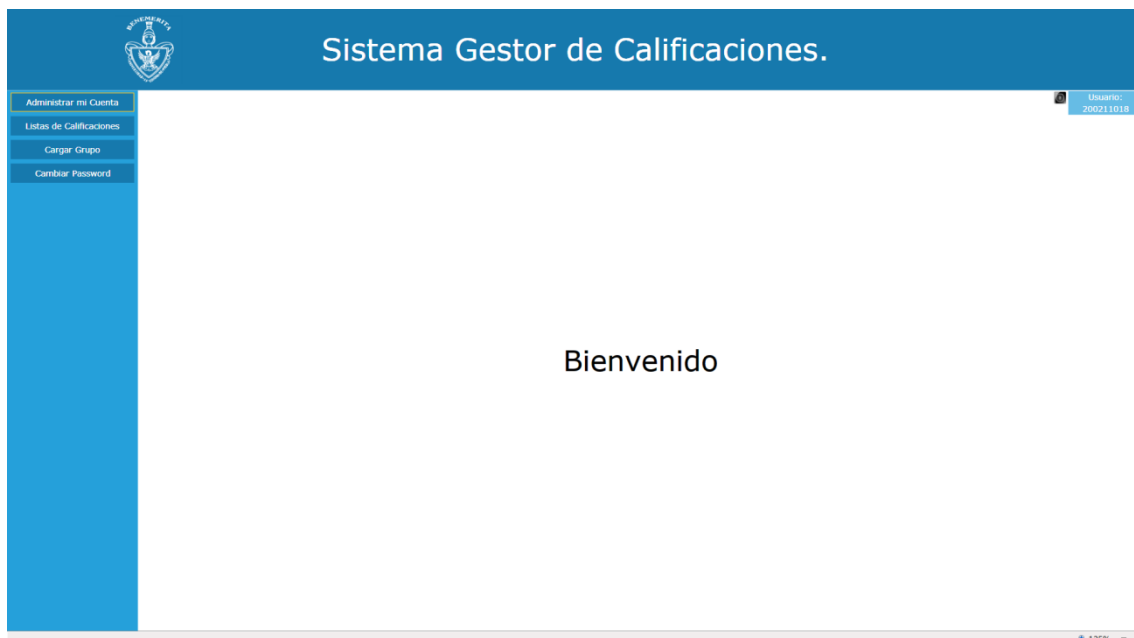
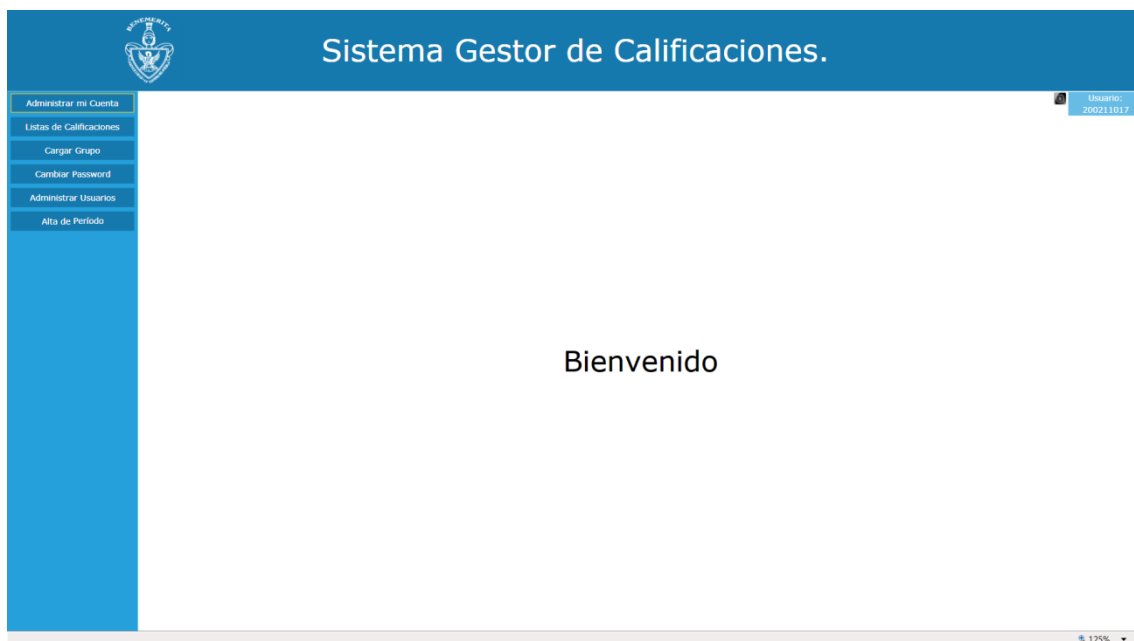


Figura C.3.1. Vista de ingreso al sistema de un "Usuario".



**Figura C.3.2.** Vista de ingreso al sistema de un “Administrador”.

A la derecha de la pantalla que se habilita en cualquiera de los dos casos, se muestra un cuadro con el número de trabajador y un botón para salir del sistema. A la izquierda de la pantalla también se habilita un menú, el cual consta de cuatro opciones: *Administrar mi Cuenta*, *Listas de Calificaciones*, *Cargar Grupo* y *Cambiar Password*, para el usuario “Administrador” se agrega la opción de *Administrar Usuarios* y *Alta de Período* que posteriormente se describirán.

## **C.4. Administrar mi cuenta.**

### **C.4.1. Visualizar datos.**

Al seleccionar esta opción se muestra la información general del usuario que está utilizando la aplicación, su número de trabajador, nombre y correo electrónico (Figura C.1.8), para cambiar algún dato, se puede oprimir el botón *Cambiar Datos* y se direccionará a otra ventana dentro de la misma pantalla (Figura C.4.1).

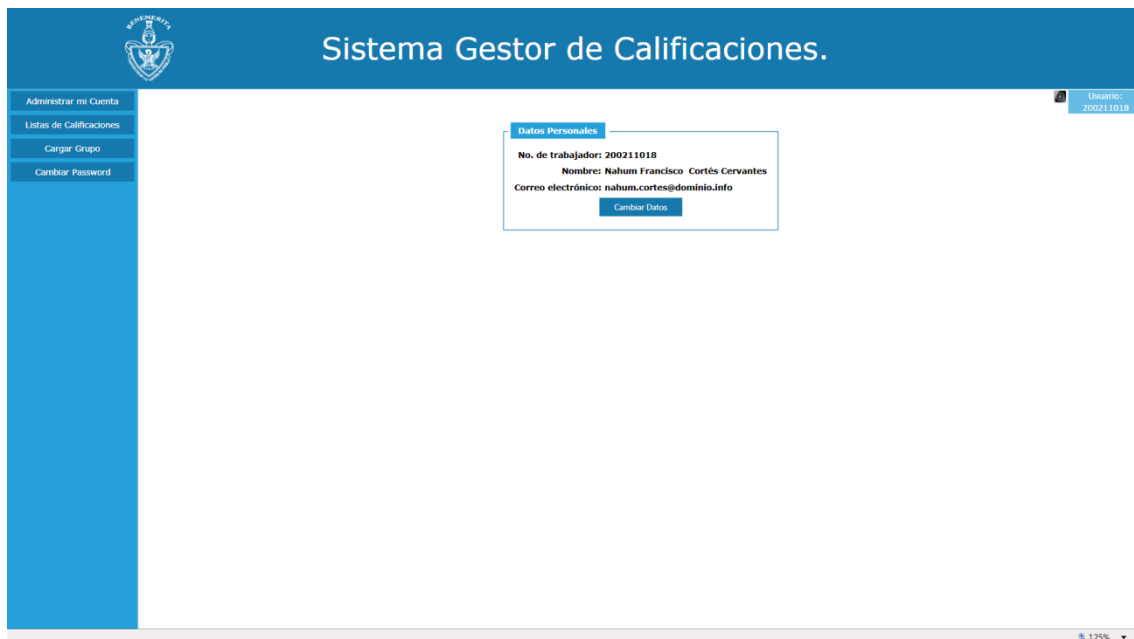


Figura C.4.1. Vista de datos generales del Profesor.

#### C.4.2 Cambiar datos.

Al seleccionar esta opción, se abrirá un nuevo formulario a donde se podrán cambiar los datos, menos el número de trabajador (Figura C.4.2), para actualizarlos se oprime el botón *Actualizar* al realizar la actualización se muestra una ventana indicando que se realizó de manera exitosa (Figura C.4.3) y posteriormente se re direcciona a la pantalla anterior, para regresar solo a la pantalla anterior se oprime el botón *Regresar* o se selecciona otro elemento del menú.

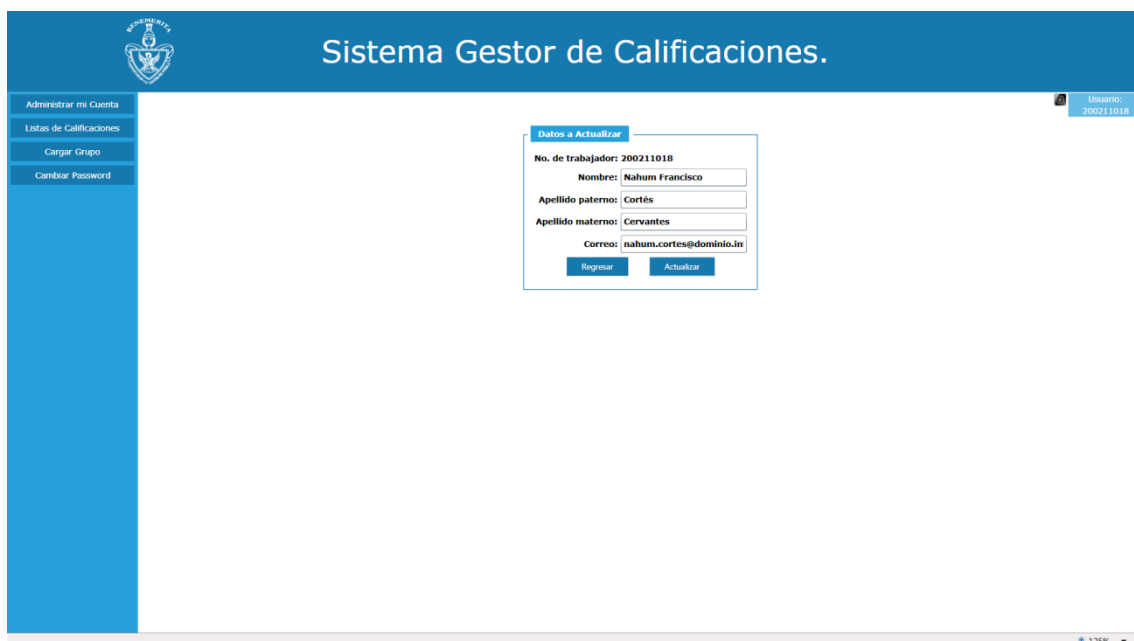
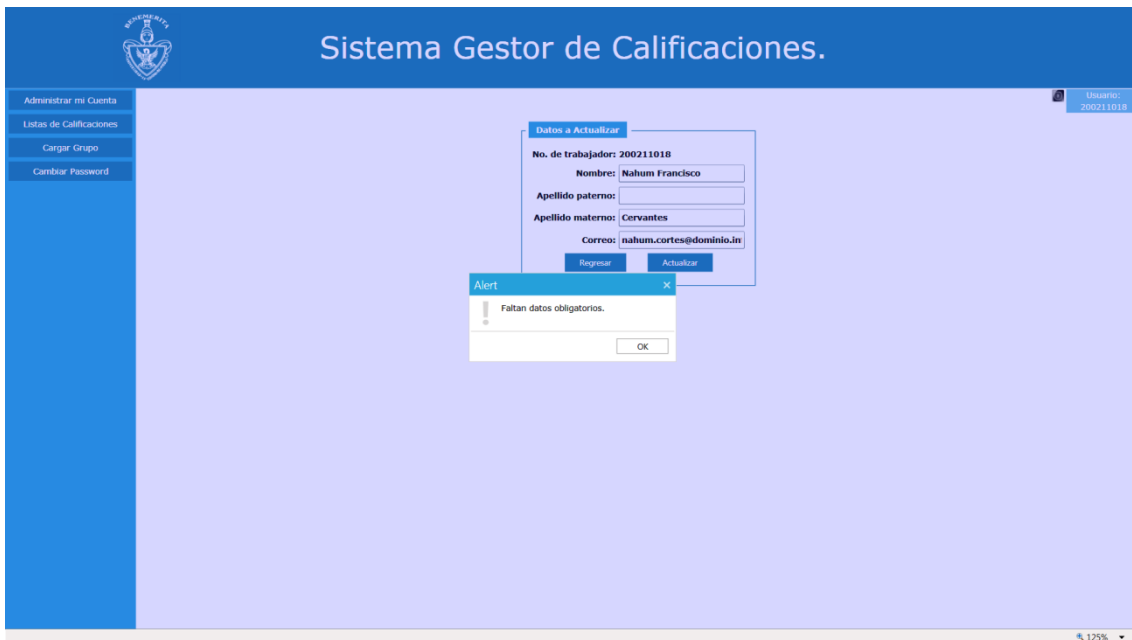


Figura C.4.2. Vista de datos generales del Profesor para ser editados.



**Figura C.4.3.** Vista al cambiar datos.

En los campos obligatorios (Nombre, apellido paterno y apellido materno) solo se aceptaran letras, si falta alguno de ellos, se mostrara un mensaje indicándolo y no se permitirá realizar la actualización (Figura C.4.4), si algún campo tiene caracteres no validos se pintara el elemento con el error para indicarlo (Figura C.4.5), para el campo del correo electrónico se valida que el texto que entra tenga el formato de un correo electrónico.



**Figura C.4.4.** Mensaje de error en caso de faltar campos obligatorios.

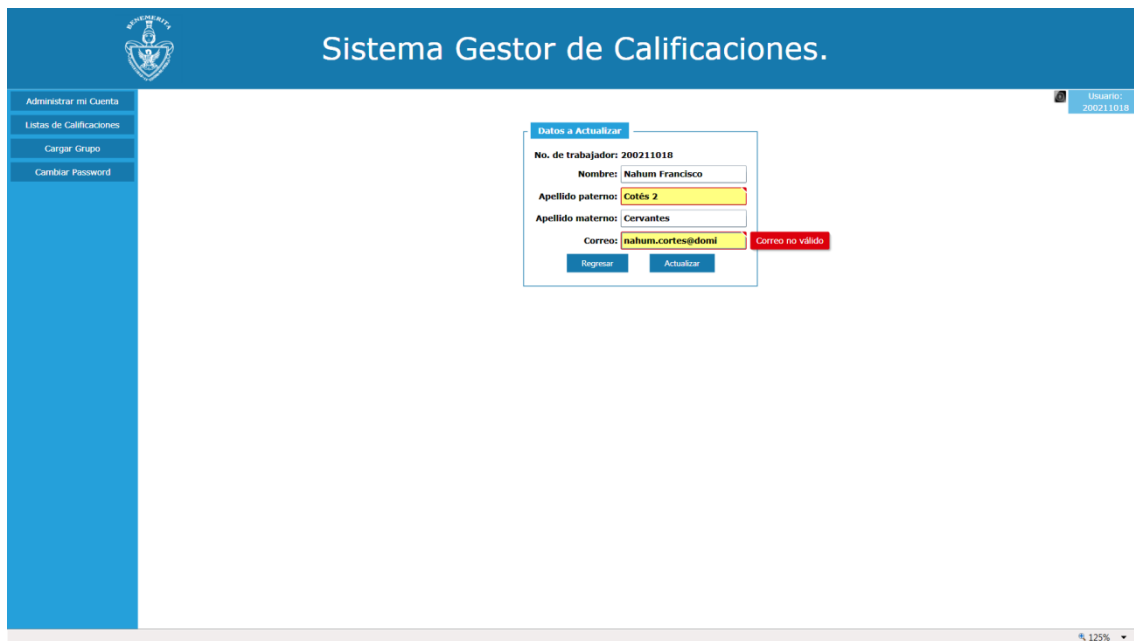


Figura C.4.5. Mensaje de error en caso de ingresar caracteres inválidos.

## C.5. Listas de calificaciones.

### C.5.1. Seleccionar lista.

Al seleccionar la opción *Listas de Calificaciones*, se mostrara un combo con los periodos en los cuales el profesor tiene calificaciones, al seleccionar alguno, en el siguiente combo aparecerán los grupos que califico en ese período (Figura C.5.1), se selecciona un grupo, se oprime el botón *Siguiente* para mostrar la lista de calificaciones del grupo (Figura C.5.2).

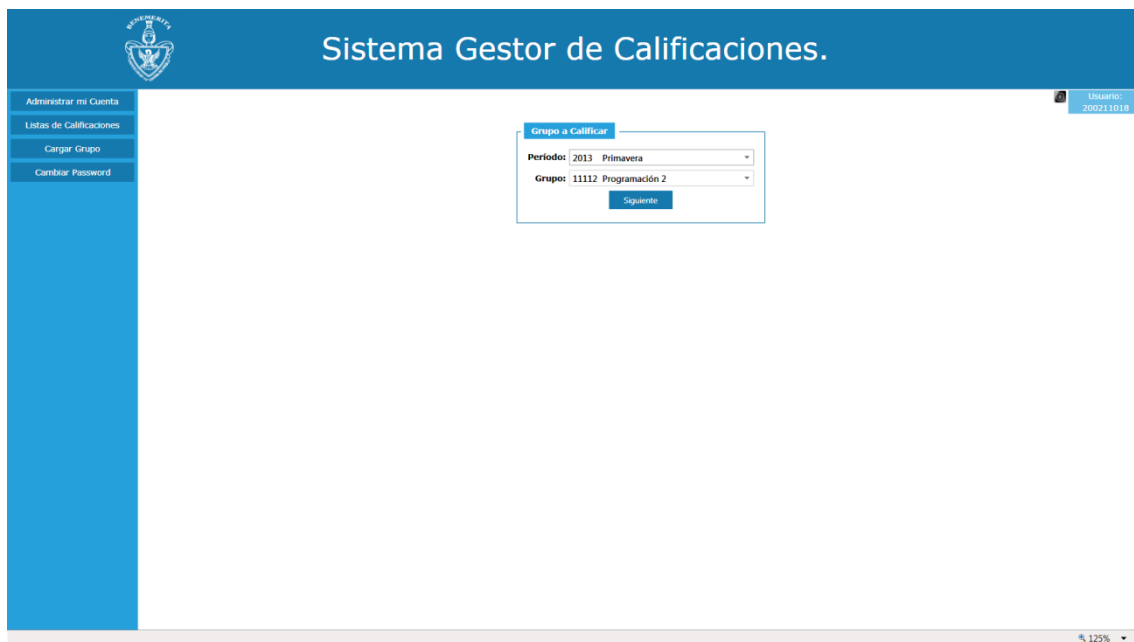


Figura C.5.1. Vista para seleccionar la lista que se desea evaluar o consultar.

## C.5.2. Calificar grupo.

Una vez seleccionado el grupo que se quiere evaluar o consultar, se mostrara una pantalla con la lista de calificaciones del grupo y opciones para agregar o quitar determinado tipo de columnas (Figura C.5.2).

Matricula	Nombre	Examen	Tarea	Final
200502209	SALGADO PARRA, RENE	0	0	0
200503101	SANCHEZ MELENDEZ, CARLOS J.	0	0	0
200508575	APONTE AGUILAR, OSCAR G.	0	0	0
200512835	MIRANDA FRANCO, MAURICIO	0	0	0
200513599	CORONA PEREZ, XELHUA J.	0	0	0
200515827	ROIAS YARCE, FELIPE A.	0	0	0
200516774	BAEZ DE LA CRUZ, VICTOR	0	0	0
200519195	JIMENEZ HERRERA, CALEB	0	0	0
200523528	MARTINEZ VALERA, VICTOR H.	0	0	0
200525935	BAZQUEZ OCOTITLA, MARCOS	0	0	0
200526332	CORTES HERNANDEZ, FERNANDO	0	0	0
200529029	XICOHTENCATI, ROMANO, ISIDRO	0	0	0
200601204	PEÑEZ GARCIA, ALICIA	0	0	0
200601598	RAMOS CRUZ, FERNANDO	0	0	0
200602342	SERRANO ESCOBAR, VICTOR	0	0	0
200605195	VILLEGAS CANTORAN, MARISSOL	0	0	0
200606175	ORTEGA RIVERA, GISELA R.	0	0	0
200606337	PUENTE TERCERO, JESUS	0	0	0
200606339	PEREZ CRUZ, JOSE R.	0	0	0
200606401	ANGELES HUERTA, IVETH	0	0	0
200606998	CABALLERO JUÁREZ, DEISY	0	0	0
200607806	MARTINEZ HERRERA, ALEJANDRO UL	0	0	0
200609504	GUTIERREZ JIMENEZ, UBALDO	0	0	0
200612199	FERNANDEZ TORRES, LUIS A.	0	0	0
200612453	LEDESMA PUENTE, MARIA M.	0	0	0

Figura C.5.2. Lista de calificaciones.

En el combo para agregar o quitar columnas, aparecerá el tipo de calificación con que se va a evaluar al grupo, como máximo se pueden agregar cuatro columnas de cada tipo y se promediara de acuerdo a lo establecido a principios de curso, además de estas opciones se contara con la calificación de extraordinario y exámenes de recuperación, los exámenes de recuperación aparecerán siempre y cuando se halla acordado evaluar exámenes en el curso, si no se le asignó un porcentaje a los exámenes (se mostrara posteriormente), entonces no aparecerán ni los exámenes, ni las recuperaciones, para poder agregar un examen de recuperación es necesario tener el examen del periodo normal (Figura C.5.3 y Figura C.5.4). El examen extraordinario estará siempre disponible, y al agregarlo y darle un valor mayor a cero este se cargara como calificación final.

Matricula	Nombre	Examen	Inicial	Tercer Tercer	Final
200502209	SALGADO PARRA, RENE	Tarea	0	0	0
200503101	SANCHEZ MELLENDEZ, CARLOS J.	Recuperación	0	0	0
200508575	APONTE AGUILAR, OSCAR G.	Extraordinario	0	0	0
200512835	MIRANDA FRANCO, MAURICIO		0	0	0
200513599	CORONA PEREZ, XELHUA J.		0	0	0
200515827	ROJAS YARCE, FELIPE A.		0	0	0
200516774	BAEZ DE LA CRUZ, VICTOR		0	0	0
200519195	JIMENEZ HERRERA, CALEB		0	0	0
200523528	MARTINEZ VALERA, VICTOR H.		0	0	0
200525935	BAZQUEZ OCOTITLA, MARCOS		0	0	0
200526332	CORTES HERNANDEZ, FERNANDO		0	0	0
200529029	XICOHTENCATL ROMANO, ISIDRO		0	0	0
200601204	PEREZ GARCIA, ALICIA		0	0	0
200601598	RAMOS CRUZ, FERNANDO		0	0	0
200602342	SERRANO ESCOBAR, VICTOR		0	0	0
200605195	VILLEGAS CANTORAN, MARSOL		0	0	0
200606175	ORTEGA RIVERA, GISELA R.		0	0	0
200606337	PUENTE TERCERO, JESUS		0	0	0
200606339	PEREZ CRUZ, JOSE R.		0	0	0
200606401	ANGELES HUERTA, IVETH		0	0	0
200606998	CABALLERO JUAREZ, DESY		0	0	0
200607806	MARTINEZ HERRERA, ALEJANDRO UL		0	0	0
200609504	GUTIERREZ JIMENEZ, UBALDO		0	0	0
200612199	FERNANDEZ TORRES, LUIS A.		0	0	0
200612453	LEDESMA PUENTE, MARIA M.		0	0	0

Figura C.5.3. Lista de calificaciones con exámenes.

Matricula	Nombre	Inicial	Final
200502209	SALGADO PARRA, RENE	0	0
200503101	SANCHEZ MELLENDEZ, CARLOS J.	0	0
200508575	APONTE AGUILAR, OSCAR G.	0	0
200512835	MIRANDA FRANCO, MAURICIO	0	0
200513599	CORONA PEREZ, XELHUA J.	0	0
200515827	ROJAS YARCE, FELIPE A.	0	0
200516774	BAEZ DE LA CRUZ, VICTOR	0	0
200519195	JIMENEZ HERRERA, CALEB	0	0
200523528	MARTINEZ VALERA, VICTOR H.	0	0
200525935	BAZQUEZ OCOTITLA, MARCOS	0	0
200526332	CORTES HERNANDEZ, FERNANDO	0	0
200529029	XICOHTENCATL ROMANO, ISIDRO	0	0
200601204	PEREZ GARCIA, ALICIA	0	0
200601598	RAMOS CRUZ, FERNANDO	0	0
200602342	SERRANO ESCOBAR, VICTOR	0	0
200605195	VILLEGAS CANTORAN, MARSOL	0	0
200606175	ORTEGA RIVERA, GISELA R.	0	0
200606337	PUENTE TERCERO, JESUS	0	0
200606339	PEREZ CRUZ, JOSE R.	0	0
200606401	ANGELES HUERTA, IVETH	0	0
200606998	CABALLERO JUAREZ, DESY	0	0
200607806	MARTINEZ HERRERA, ALEJANDRO UL	0	0
200609504	GUTIERREZ JIMENEZ, UBALDO	0	0
200612199	FERNANDEZ TORRES, LUIS A.	0	0
200612453	LEDESMA PUENTE, MARIA M.	0	0

Figura C.5.4. Lista de calificaciones sin exámenes.

La pantalla cuenta con más botones que a continuación se describen:

- **Regresar.**- Permite regresar a la vista anterior para elegir otro grupo.
- **Grabar Actualizaciones.**- Actualiza el contenido de la lista en la *Base de Datos*, en caso de que la actualización no sea exitosa se muestra un mensaje (Figura C.5.5), en caso de ingresar alguna cadena o carácter al momento de actualizar, no se realizara ningún cambio en la *Base de Datos* y se pintara el contorno de rojo indicando un error (Figura C.5.6).



**Figura C.5.5.** En caso de ingresar un valor que no esté entre el intervalo de 1 a 10 se muestra el mensaje con el alumno para corregir.



**Figura C.5.6.** En caso de ingresar un carácter se pinta el contorno de rojo y no se permite guardar en la Base de Datos.

- *Imprimir.*- Al seleccionar esta opción se muestra una nueva ventana con el reporte que contiene las calificaciones del grupo (Figura C.5.7).

**Lista de Calificaciones**

Materia: **Calculo Diferencial**    23456    102    Profesor: **Nahum d**  
 Nombre: **NRC**    Sección    ID: **200211018**

Matricula	Nombre	Examen	Tarea	Exposición	Proyecto	Otros	Final
200502209	SALGADO PARRA, RENE	0.00	10.00	0.00	7.00	0.00	9
200503101	SANCHEZ MELENDEZ, CARLOS J.	0.00	6.00	0.00	9.00	0.00	8
200508575	APONTE AGUILAR, OSCAR G.	0.00	10.00	0.00	9.00	0.00	10
200512835	MIRANDA FRANCO, MAURICIO	0.00	8.00	0.00	8.00	0.00	8
200513599	CORONA PEREZ, YELHUA J.	0.00	8.00	0.00	9.00	0.00	9
200515827	ROJAS YARCE, FELIPE A.	0.00	6.00	0.00	7.00	0.00	7
200516774	BAEZ DE LA CRUZ, VICTOR	0.00	10.00	0.00	10.00	0.00	10
200519195	JIMENEZ HERRERA, CALEB	0.00	10.00	0.00	10.00	0.00	10
200523528	MARTINEZ VALERA, VICTOR H.	0.00	8.00	0.00	10.00	0.00	9
200525935	BAZQUEZ OCOTITLA, MARCOS	0.00	0.00	0.00	10.00	0.00	5
200526332	CORTES HERNANDEZ, FERNANDO	0.00	8.00	0.00	10.00	0.00	9
200529029	XICOHTENCATL ROMANO, ISIDRO	0.00	8.00	0.00	0.00	0.00	4
200601204	PEREZ GARCIA, ALICIA	0.00	8.00	0.00	10.00	0.00	9
200601598	RAMOS CRUZ, FERNANDO	0.00	8.00	0.00	10.00	0.00	9
200602342	SERRANO ESCOBAR, VICTOR	0.00	8.00	0.00	0.00	0.00	4
200605195	VILLEGAS CANTORAN, MARISOL	0.00	8.00	0.00	10.00	0.00	9
200606175	ORTEGA RIVERA, GISELA R.	0.00	8.00	0.00	0.00	0.00	4
200606337	PUENTE TERCERO, JESUS	0.00	8.00	0.00	10.00	0.00	9
200606339	PEREZ CRUZ, JOSE R.	0.00	8.00	0.00	10.00	0.00	9
200606401	ANGELES HUERTA, IVETH	0.00	8.00	0.00	0.00	0.00	4
200606998	CABALLERO JUAREZ, DEISY	0.00	0.00	0.00	0.00	0.00	0
200607806	MARTINEZ HERRERA, ALEJANDRO U.	0.00	0.00	0.00	0.00	0.00	0
200609504	GUTIERREZ JIMENEZ, UBALDO	0.00	0.00	0.00	0.00	0.00	0
200612199	FERNANDEZ TORRES, LUIS A.	0.00	0.00	0.00	0.00	0.00	0
200612453	LEDESMA PUENTE, MARIA M.	0.00	0.00	0.00	0.00	0.00	0
200618491	CARRERA ARRIAGA, JOEL	0.00	0.00	0.00	0.00	0.00	0
200619115	GOYRI AGUIRRE, OSCAR D.	0.00	0.00	0.00	0.00	0.00	0
200620133	PEREZ VILLEGAS, HEBER	0.00	0.00	0.00	0.00	0.00	0
200622019	TORRES BARRIENTOS, LIZBETH	0.00	0.00	0.00	0.00	0.00	0
200626757	SANTIAGO HERNANDEZ, JOSE L.	0.00	0.00	0.00	0.00	0.00	0
200628799	ROLDAN ARAUJO, MARTIN	0.00	0.00	0.00	0.00	0.00	0
200632376	ARROYO CARMONA, MISAEL	0.00	0.00	0.00	0.00	0.00	0
200633297	LUNA PEREZ, MAURICIO	0.00	0.00	0.00	0.00	0.00	0
200634338	SANDOVAL HERNANDEZ, GAMALIEL O.	0.00	0.00	0.00	0.00	0.00	0

Figura C.5.7. Información del grupo en un reporte.

- **Exportar a Excel.**- Al elegir esta opción se muestra una nueva ventana (Figura C.5.8), a donde se deberá elegir la ubicación y el nombre del archivo, el archivo se guardara con extensión XML compatible con Excel.

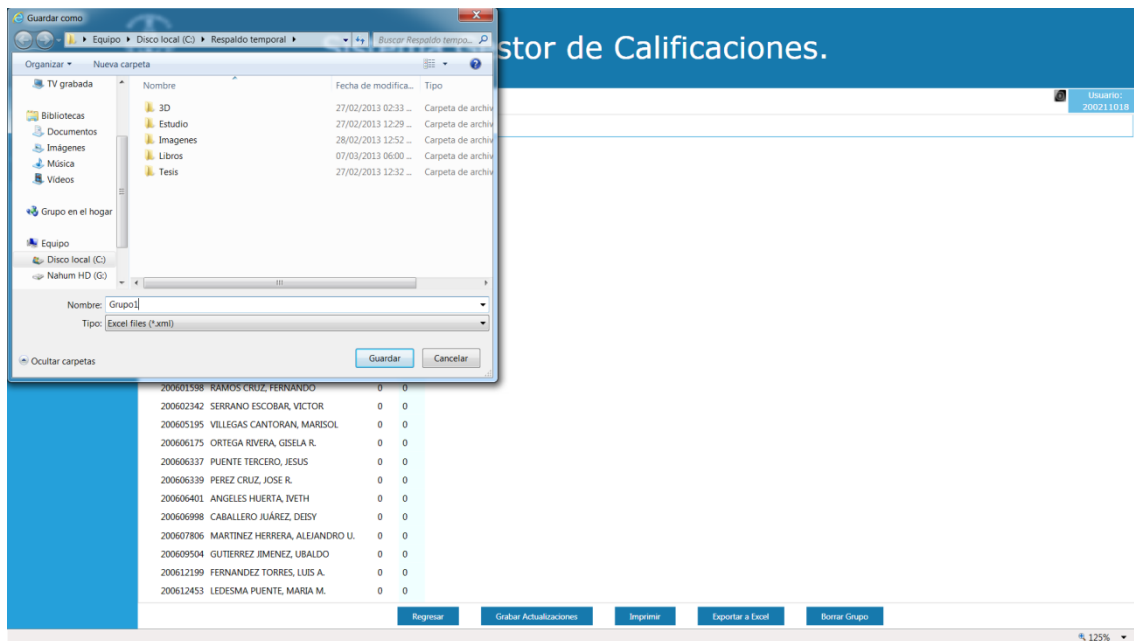


Figura C.5.8. Información del grupo en un reporte.

- **Borrar Grupo.**- Al elegir esta opción se mostrara una venta para confirmar que se quiere eliminar (Figura C.5.9), a donde en caso de confirmar se eliminara el grupo de la Base de Datos y se re direccionará a la pantalla anterior para seleccionar un nuevo grupo.

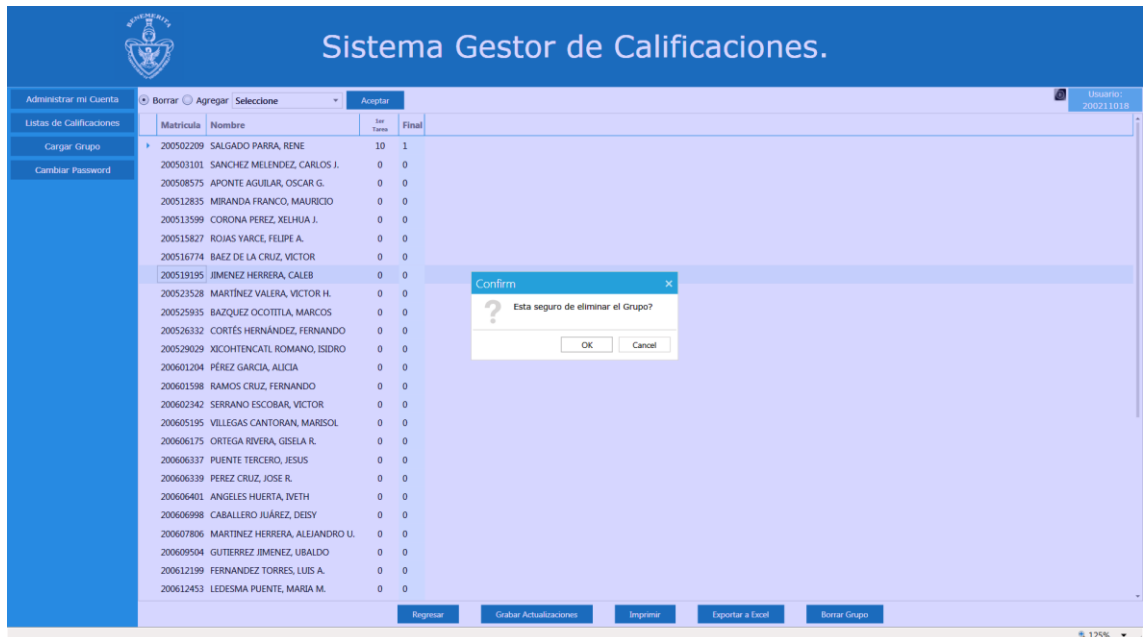


Figura C.5.9. Eliminar el grupo.

## C.6. Cargar grupo.

### C.6.1. Cargar grupo.

Al elegir esta opción se mostrara en la pantalla un formulario para cargar una lista de calificaciones (Figura C.6.1).

Si se selecciona la opción en la parte de la izquierda de *Subir Archivo* se debe seleccionar el botón *Cargar Grupo* y se mostrara una ventana para ubicar el archivo que se desea subir (Figura C.6.3).

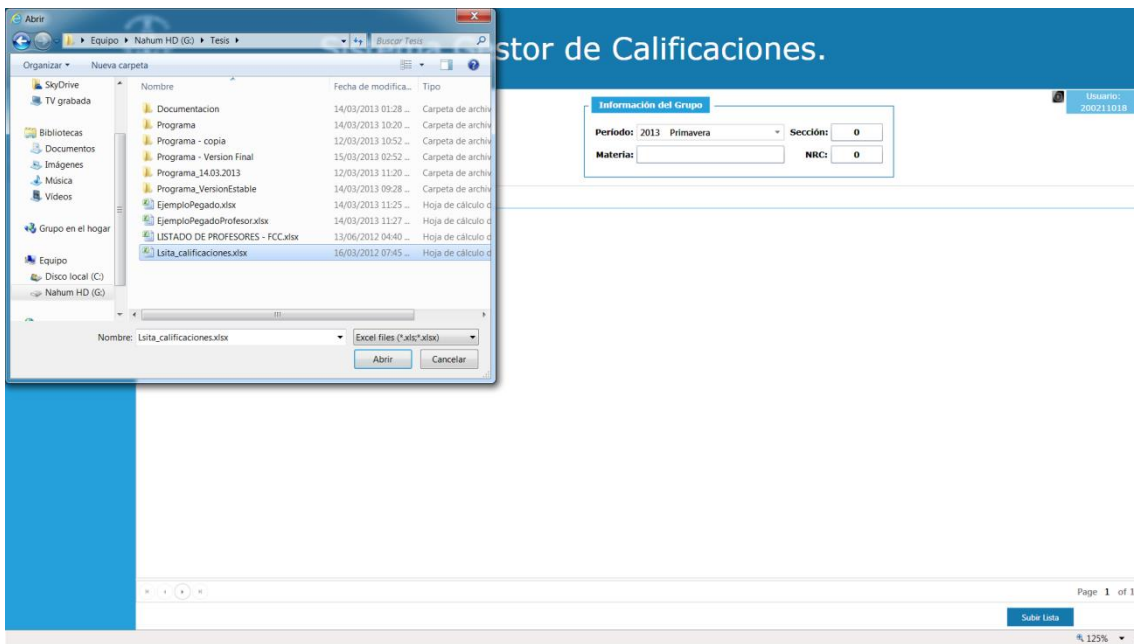
Si se selecciona la opción *Copiar Datos* solo se debe copiar los datos que se deseen cargar y posteriormente oprimir el botón *Pegar Datos* (Figura C.6.2).



Figura C.6.1. Vista inicial de Cargar Grupo por un archivo de Excel.



**Figura C.6.2.** Vista inicial de Cargar Grupo por pegado de datos.



**Figura C.6.3.** Cargar el archivo Excel.

En caso de elegir un archivo que no tenga el formato dado por la Secretaría Académica (Figura C.6.5), o no se realice de manera adecuada el copiado de datos (Figura C.6.6 y Figura C.6.7) se mostrara un error (Figura C.6.4).

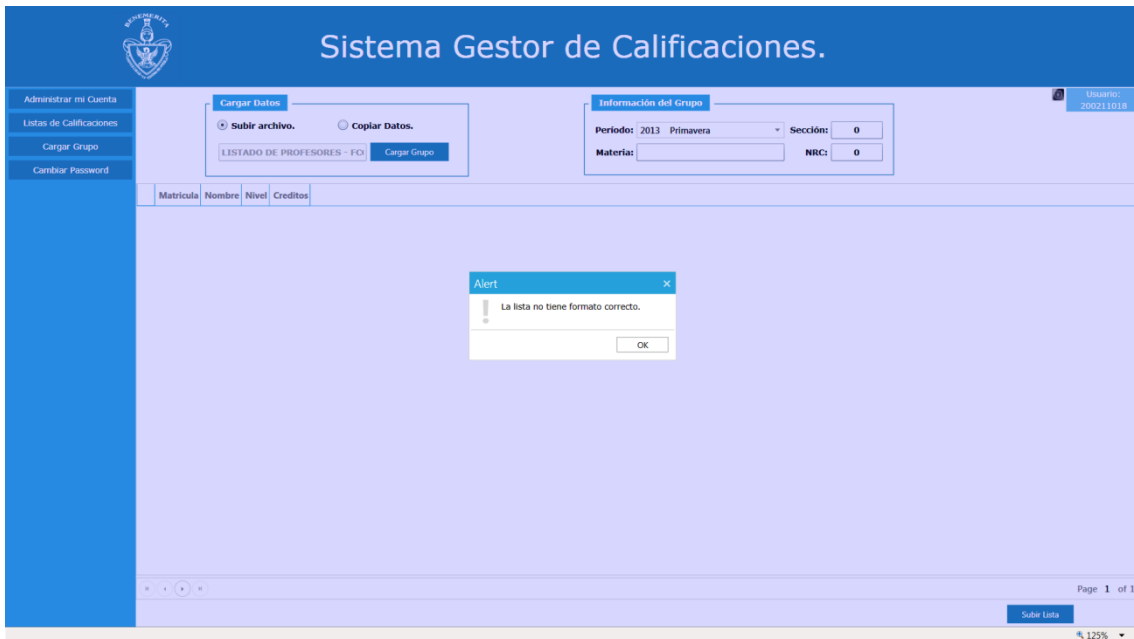


Figura C.6.4. Formato no coincide con el formato para subir la lista.

Registro	Nombre Alumno	ID	Status Reg	Nivel	Créditos	Parcial	Final	Detalle de Calificaciones
1	AGUILAR AVENDAÑO, ALEJANDRA	200700321	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
2	ANGELES HUERTA, IVETH	200606401	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
3	APONTE AGUILAR, OSCAR G.	200608575	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
4	ARROYO CARMONA, MISAL	200632376	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
5	BAEZ DE LA CRUZ, VICTOR	200516774	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
6	BAZQUEZ OCOTITLA, MARCOS	200525935	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
7	CABALLERO JUÁREZ, DEISY	200606998	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
8	CARRERA ARRIAGA, JOEL	200618491	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
9	CORONA PEREZ, XELHUA J.	200513599	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
10	CORTÉS HERNÁNDEZ, FERNANDO	200526332	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
11	FERNANDEZ TORRES, LUIS A.	200512199	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
12	GOYRI AGUIRRE, OSCAR D.	200519115	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
13	GUTIERREZ JIMENEZ, UBALDO	200608504	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
14	JIMENEZ HERRERA, CALEB	200519195	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
15	LEDESMA PUENTE, MARIA M.	200512453	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
16	LUNA PEREZ, MAURICIO	200633297	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
17	MARTINEZ HERRERA, ALEJANDRO U.	200607806	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
18	MARTÍNEZ VALERA, VICTOR H.	200523528	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
19	MIRANDA FRANCO, MAURICIO	200512835	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
20	OLIVERA PEREZ, EDUARDO	200729516	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
21	ORTEGA RIVERA, GISELA R.	200606175	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	
22	PARADES VILLEGAS, HEBER	200620133	***Inscrito Via Web**	Licenciatura Créditos	10,000	Ingrese	Ingrese	

Figura C.6.5. Formato de la lista de calificaciones proporcionado por la Secretaría Académica.

	A	B	C	D	E	F
1						
2	200700321	AGUILAR AVENDAÑO, ALEJANDRA	Licenciatura Créditos	10		
3	200606401	ANGELES HUERTA, IVETH	Licenciatura Créditos	10		
4	200508575	APONTE AGUILAR, OSCAR G.	Licenciatura Créditos	10		
5	200632376	ARROYO CARMONA, MISAEL	Licenciatura Créditos	10		
6	200516774	BAEZ DE LA CRUZ, VICTOR	Licenciatura Créditos	10		
7	200525935	BAZQUEZ OCOTITLA, MARCOS	Licenciatura Créditos	10		
8	200606998	CABALLERO JUÁREZ, DEISY	Licenciatura Créditos	10		
9	200618491	CARRERA ARRIAGA, JOEL	Licenciatura Créditos	10		
10	200513599	CORONA PEREZ, XELHUA J.	Licenciatura Créditos	10		
11	200526332	CORTÉS HERNÁNDEZ, FERNANDO	Licenciatura Créditos	10		
12	200612199	FERNANDEZ TORRES, LUIS A.	Licenciatura Créditos	10		
13	200619115	GOYRI AGUIRRE, OSCAR D.	Licenciatura Créditos	10		
14	200609504	GUTIERREZ JIMÉNEZ, UBALDO	Licenciatura Créditos	10		
15	200519195	JIMENEZ HERRERA, CALEB	Licenciatura Créditos	10		
16	200612453	LEDESMA PUENTE, MARIA M.	Licenciatura Créditos	10		
17	200633297	LUNA PEREZ, MAURICIO	Licenciatura Créditos	10		
18	200607806	MARTINEZ HERRERA, ALEJANDRO U.	Licenciatura Créditos	10		
19	200523528	MARTÍNEZ VALERA, VICTOR H.	Licenciatura Créditos	10		
20	200512835	MIRANDA FRANCO, MAURICIO	Licenciatura Créditos	10		
21	200729616	OLIVERA PEREZ, EDUARDO	Licenciatura Créditos	10		
22	200606175	ORTEGA RIVERA, GISELA R.	Licenciatura Créditos	10		
23	200620133	PAREDES VILLEGAS, HEBER	Licenciatura Créditos	10		
24	200606339	PEREZ CRUZ, JOSE R.	Licenciatura Créditos	10		
25	200601204	PEREZ GARCÍA, ALICIA	Licenciatura Créditos	10		
26	200606337	PUENTE TERCERO, JESUS	Licenciatura Créditos	10		
27	200601598	RAMOS CRUZ, FERNANDO	Licenciatura Créditos	10		
28	200515827	ROJAS YARCE, FELIPE A.	Licenciatura Créditos	10		
29	200628799	ROLDAN ARAUJO, MARTIN	Licenciatura Créditos	10		
30	200502209	SALGADO PARRA, RENE	Licenciatura Créditos	10		
31	200634349	SANCHEZ MARTINEZ, LUIS G.	Licenciatura Créditos	10		
32	200503101	SANCHEZ MELENDEZ, CARLOS J.	Licenciatura Créditos	10		
		SANCHEZ VELAZQUEZ, LAURA P.	Licenciatura Créditos	10		

**Figura C.6.6.** Formato de la lista de calificaciones para pegar la lista (Se deben seleccionar solo las columnas que tengan información, en el formato que se muestra y posteriormente elegir la opción de copiar).

Archivo	Edición	Formato	Ver	Ayuda
200700321	AGUILAR AVENDAÑO, ALEJANDRA	Licenciatura Créditos	10	
200606401	ANGELES HUERTA, IVETH	Licenciatura Créditos	10	
200508575	APONTE AGUILAR, OSCAR G.	Licenciatura Créditos	10	
200632376	ARROYO CARMONA, MISAEL	Licenciatura Créditos	10	
200516774	BAEZ DE LA CRUZ, VICTOR	Licenciatura Créditos	10	
200525935	BAZQUEZ OCOTITLA, MARCOS	Licenciatura Créditos	10	
200606998	CABALLERO JUÁREZ, DEISY	Licenciatura Créditos	10	
200618491	CARRERA ARRIAGA, JOEL	Licenciatura Créditos	10	
200513599	CORONA PEREZ, XELHUA J.	Licenciatura Créditos	10	
200526332	CORTÉS HERNÁNDEZ, FERNANDO	Licenciatura Créditos	10	
200612199	FERNANDEZ TORRES, LUIS A.	Licenciatura Créditos	10	
200619115	GOYRI AGUIRRE, OSCAR D.	Licenciatura Créditos	10	
200609504	GUTIERREZ JIMÉNEZ, UBALDO	Licenciatura Créditos	10	
200519195	JIMENEZ HERRERA, CALEB	Licenciatura Créditos	10	
200612453	LEDESMA PUENTE, MARIA M.	Licenciatura Créditos	10	
200633297	LUNA PEREZ, MAURICIO	Licenciatura Créditos	10	
200607806	MARTINEZ HERRERA, ALEJANDRO U.	Licenciatura Créditos	10	
200523528	MARTÍNEZ VALERA, VICTOR H.	Licenciatura Créditos	10	
200512835	MIRANDA FRANCO, MAURICIO	Licenciatura Créditos	10	
200729616	OLIVERA PEREZ, EDUARDO	Licenciatura Créditos	10	
200606175	ORTEGA RIVERA, GISELA R.	Licenciatura Créditos	10	
200620133	PAREDES VILLEGAS, HEBER	Licenciatura Créditos	10	
200606339	PEREZ CRUZ, JOSE R.	Licenciatura Créditos	10	
200601204	PEREZ GARCÍA, ALICIA	Licenciatura Créditos	10	
200606337	PUENTE TERCERO, JESUS	Licenciatura Créditos	10	
200601598	RAMOS CRUZ, FERNANDO	Licenciatura Créditos	10	
200515827	ROJAS YARCE, FELIPE A.	Licenciatura Créditos	10	
200628799	ROLDAN ARAUJO, MARTIN	Licenciatura Créditos	10	
200502209	SALGADO PARRA, RENE	Licenciatura Créditos	10	
200634349	SANCHEZ MARTINEZ, LUIS G.	Licenciatura Créditos	10	
200503101	SANCHEZ MELENDEZ, CARLOS J.	Licenciatura Créditos	10	
200702583	SANCHEZ VELAZQUEZ, LAURA P.	Licenciatura Créditos	10	

**Figura C.6.7.** Formato alternativo de la lista de calificaciones para pegar la lista (Se

puede seleccionar de un blog de notas siempre y cuando estén separadas las columnas por un tabulador).

Si se elige un archivo con formato correcto (Figura C.6.5, Figura C.6.6 y Figura C.6.7), se cargara la lista de alumnos, será necesario ingresar el NRC, la Sección, el nombre de la materia y el periodo en el que será evaluado, en caso de que el NRC no se repita y estén los datos completos, se podrá continuar (Figura C.6.8).

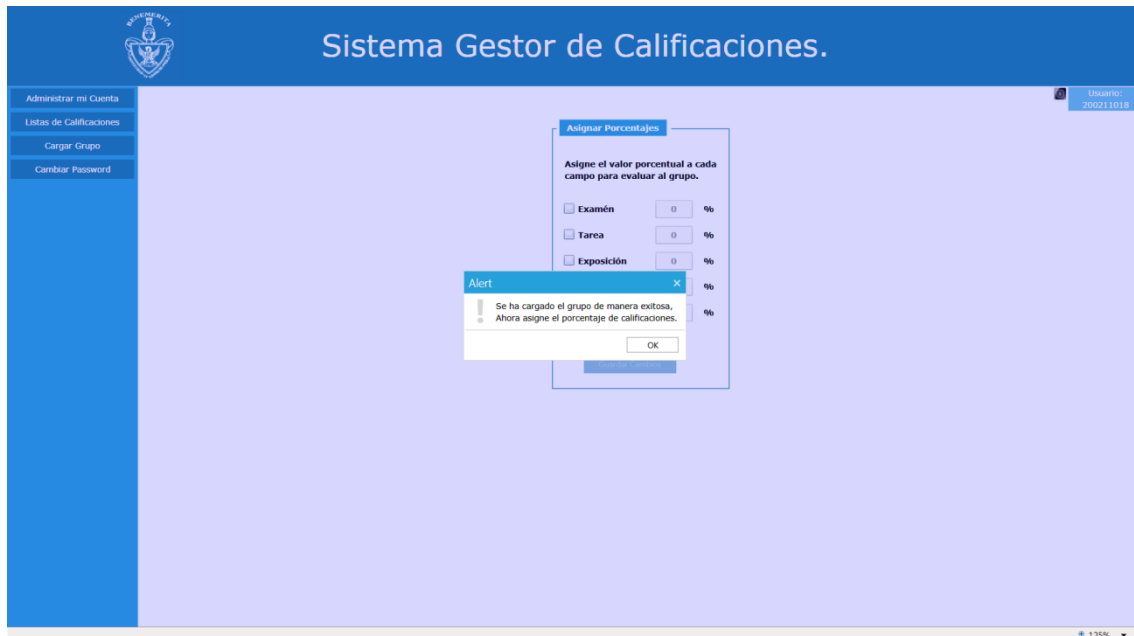


Figura C.6.8. Lista cargada.

Antes de poder subir la lista se valida que la sección y el NRC, tengan un formato correcto, para la sección un número entero de 3 dígitos y para el NRC un número entero de 5 dígitos, en caso de que no tengan el formato correcto se mostrara una advertencia y no podrá subirse el grupo (Figura C.6.9).



Figura C.6.9. Lista cargada con porcentaje asignado.

## C.6.2. Asignar porcentajes y tipos de calificación.

Una vez que se han cargado los datos de la lista de manera correcta es necesario asignar el porcentaje con que se va a calificar el grupo (Figura C.6.10), se muestra un formulario en el cual se debe checar el tipo de calificación con el que se va a evaluar, y asignar un porcentaje, una vez que el porcentaje sea igual a 100 se habilita el botón *Guardar Cambios* y se podrán guardar los cambios (Figura C.6.11), en caso de que el porcentaje no de los 100 el botón estará deshabilitado.

Sistema Gestor de Calificaciones.

Administrador de Cuenta

Usuarios: 000011018

Asignar Porcentajes

Asigne el valor porcentual a cada campo para evaluar al grupo.

Examen 0 %

Tarea 0 %

Exposición 0 %

Proyecto 0 %

Otro 0 %

Total: 0

Guardar Cambios

Figura C.6.10. Asignar porcentajes.

Sistema Gestor de Calificaciones.

Administrador de Cuenta

Usuarios: 000011018

Acciones: Borrar, Agregar, Seleccione, Aceptar

Matrícula	Nombre	Ser Examen	Ser Proyecto	Final
200502209	SALGADO PARRA, RENE	0	0	-1
200503101	SANCHEZ MELENDEZ, CARLOS J.	0	0	-1
200508575	APONTE AGUILAR, OSCAR G.	0	0	-1
200512835	MIRANDA FRANCO, MAURICIO	0	0	-1
200513599	CORONA PEREZ, XELHUA J.	0	0	-1
200515827	ROIAS YARCE, FELIPE A.	0	0	-1
200516774	BAEZ DE LA CRUZ, VICTOR	0	0	-1
200519195	JIMENEZ HERRERA, CALEB	0	0	-1
200523528	MARTÍNEZ VALERA, VICTOR H.	0	0	-1
200525935	BAZQUEZ OCOTITLA, MARCOS	0	0	-1
200526332	CORTÉS HERNÁNDEZ, FERNANDO	0	0	-1
200529029	XICOHTENCATI ROMANO, ISIDRO	0	0	-1
200601204	PÉREZ GARCÍA, ALICIA	0	0	-1
200601598	RAMOS CRUZ, FERNANDO	0	0	-1
200602342	SERRANO ESCOBAR, VICTOR	0	0	-1
200605195	VILLEGAS CANTORAN, MARSOL	0	0	-1
200606175	ORTEGA RIVERA, GISELA R.	0	0	-1
200606337	PUENTE TERCERO, JESUS	0	0	-1
200606339	PEREZ CRUZ, JOSE R.	0	0	-1
200606401	ANGELES HUERTA, IVETH	0	0	-1
200606998	CABALLERO JUÁREZ, DEISY	0	0	-1
200607806	MARTÍNEZ HERRERA, ALEJANDRO LL.	0	0	-1
200609504	GUTIERREZ JIMENEZ, URBALDO	0	0	-1
200612199	FERNANDEZ TORRES, LUIS A.	0	0	-1
200612453	LEDESMA PUENTE, MARIA M.	0	0	-1

Alert

Porcentaje actualizado con éxito.

OK

Regresar, Grabar Actualizaciones, Imprimir, Exportar a Excel, Borrar Grupo

Figura C.6.11. Porcentaje correcto.

Para poder calificar un grupo es necesario se halla asignado el porcentaje correspondiente al grupo, si al elegir un grupo aún no se guarda el porcentaje de evaluación se re direccionara a asignar porcentajes (Figura C.6.12).



Figura C.6.12. Falta asignar porcentaje antes de calificar un grupo.

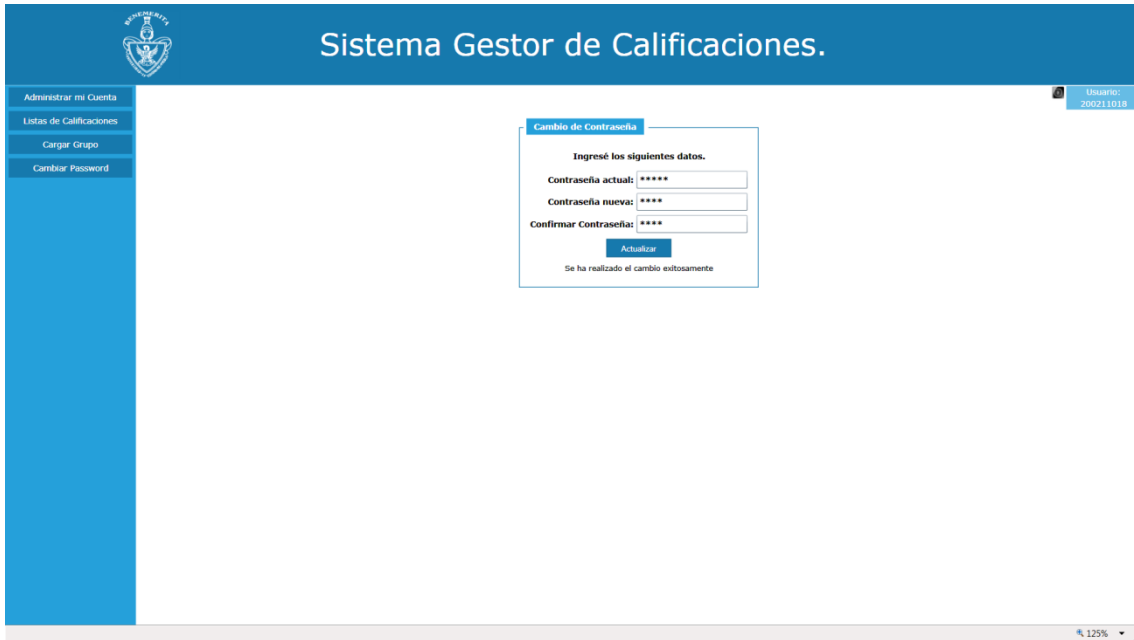
## C.7. Cambiar contraseña.

### C.7.1. Cambiar contraseña.

Si se selecciona esta opción se puede cambiar la contraseña actual por una nueva (Figura C.7.1), solo es necesario ingresar la contraseña actual, ingresar la contraseña nueva y confirmarla y si todo es correcto, se realizara la actualización (Figura C.7.2).

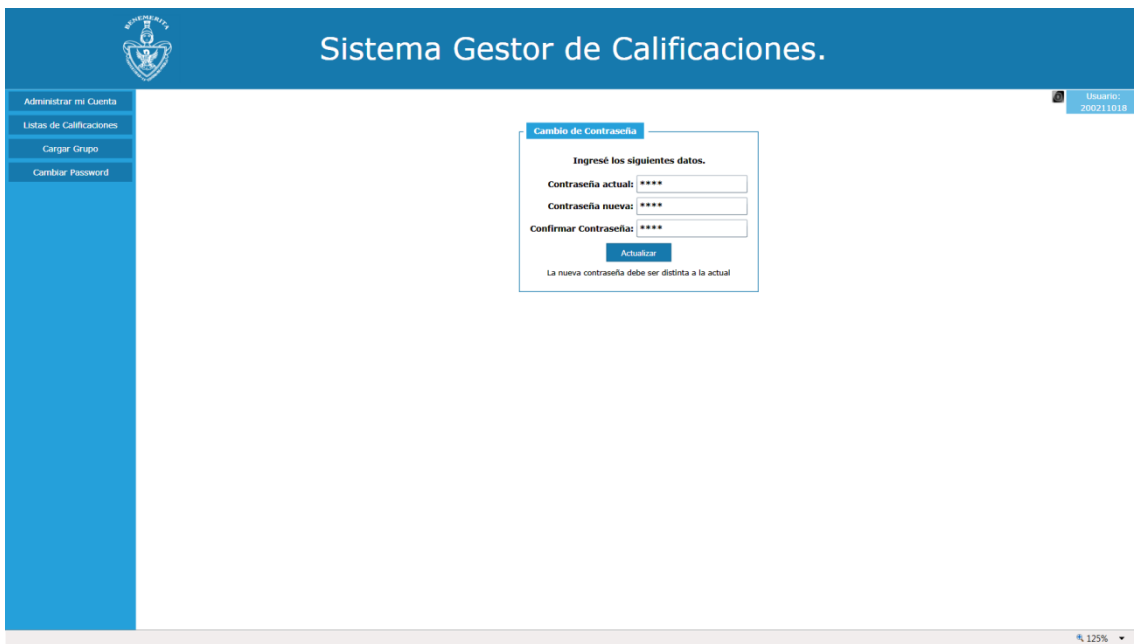


Figura C.7.1. Cambiar contraseña.



**Figura C.7.2.** *Contraseña actualizada.*

En caso de faltar datos o no coincidir a continuación se muestran los posibles mensajes de error (Figura C.7.3, Figura C.7.4, Figura C.7.5 y Figura C.7.6):



**Figura C.7.3.** *La contraseña debe ser diferente a la actual.*

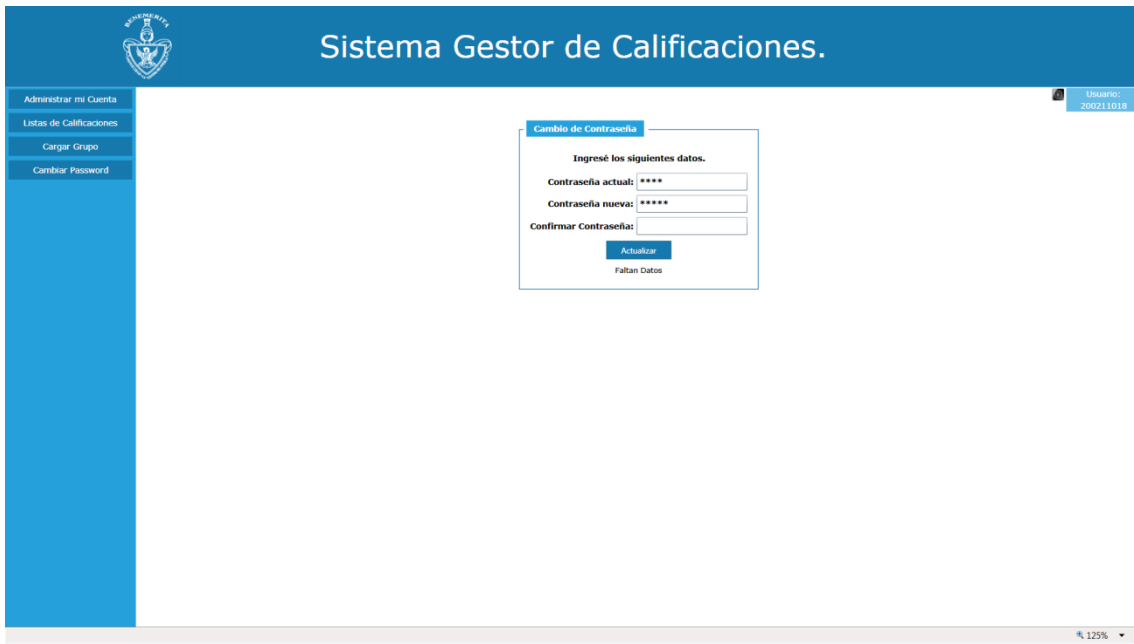


Figura C.7.4. *Faltan datos para realizar la actualización.*

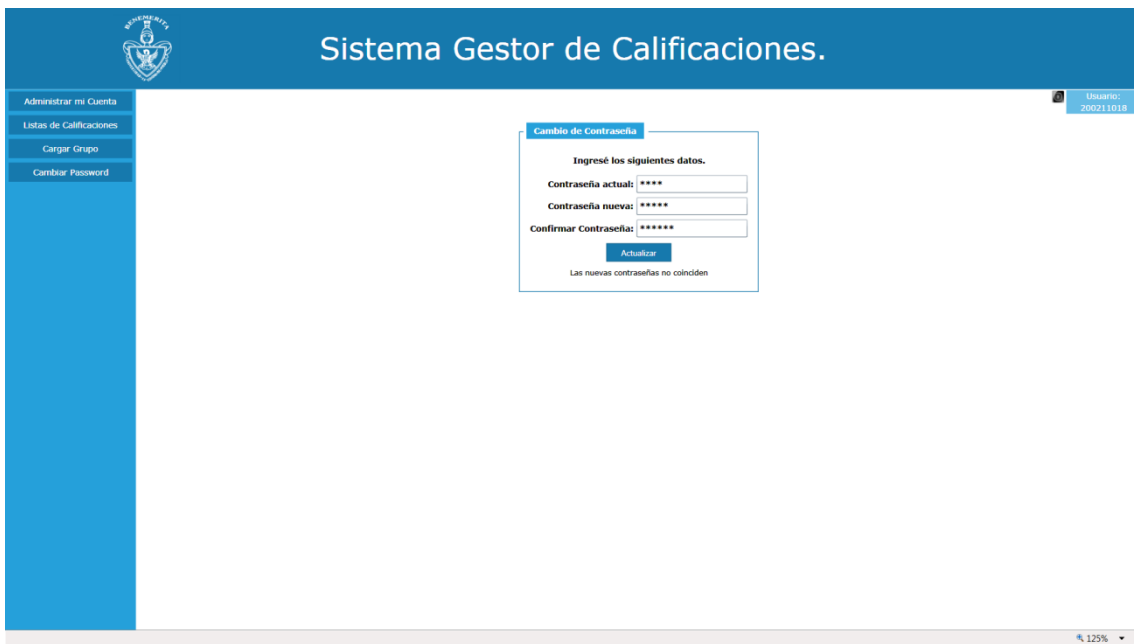


Figura C.7.5. *La contraseña nueva no coincide.*

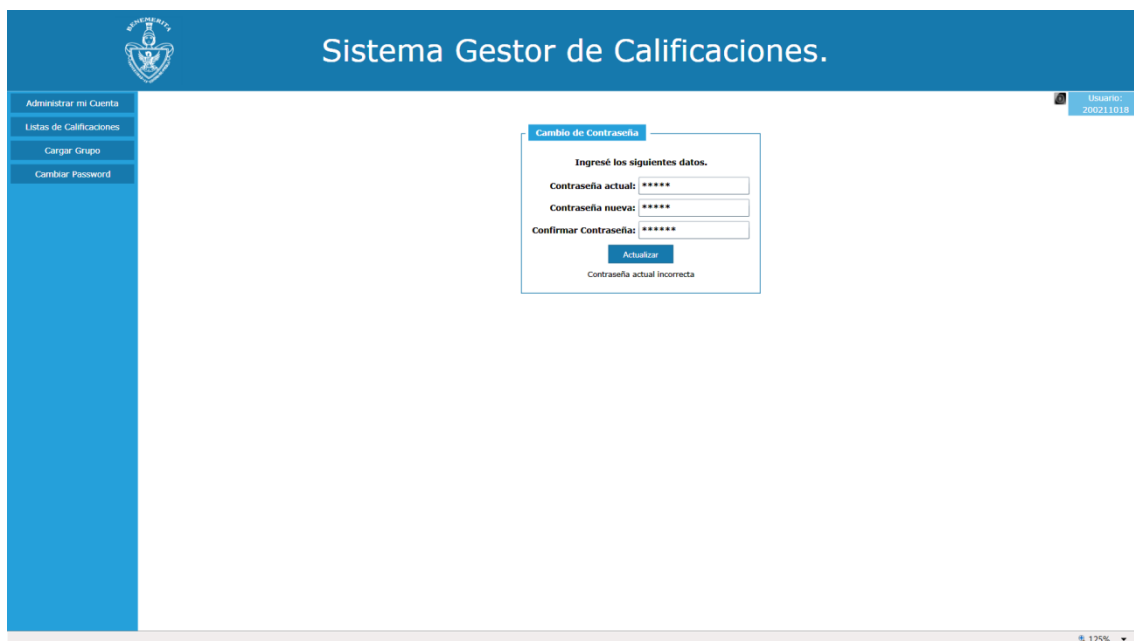


Figura C.7.6. La contraseña actual no es correcta.

## C.8. Administrar usuarios (Administrador).

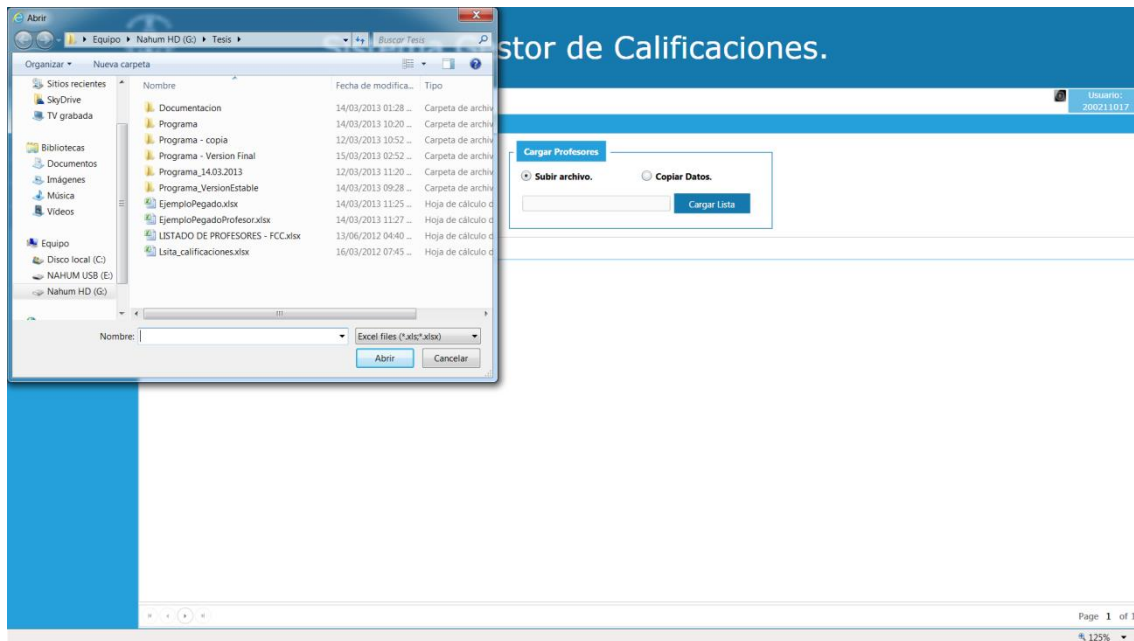
Esta opción solo está disponible para los usuarios de tipo “Administrador”, se pueden agregar y borrar nuevos usuarios y se les puede asignar el rol o tipo de usuario que van a ser. Esta opción está formada por tres partes (Figura C.8.1) que a continuación serán descritas.



Figura C.8.1. Administrar usuarios.

### C.8.1. Alta de maestros.

Para poder cargar la lista de maestros es necesario oprimir el botón *Cargar Lista de Maestros*, se mostrara una ventana a donde se debe ubicar el archivo que contiene la información de los profesores (Figura C.8.2), si la lista está en formato correcto (Figura C.8.3).



**Figura C.8.2.** Cargar la lista de profesores por archivo de Excel.

ID	NOMBRE	Dirección de Correo FCC	Dirección de Correo Externa
1	ADOLFO AGUILAR RICO		adolforico2@hotmail.com adolforico2@gmail.com
2	LUIS CARLOS ALTAMIRANO ROBLES	altamirano@cs.buap.mx	LCARLOS6@prodigy.net.mx
3	ALMA DELIZ AMBROSIO VAZQUEZ	ambrosio@cs.buap.mx	
4	MARIO ANZURES GARCIA	anzures@cs.buap.mx	mario_anzures@hotmail.com manzures@siu.buap.mx
5	ETELVINA ARCHUNDIA SIERRA	etelvina@cs.buap.mx	
6	EDUARDO ARIZA VELAZQUEZ	eariza@cs.buap.mx	
7	APOLONIO ATA PEREZ	apolonio@cs.buap.mx	
8	IRENE OLAYA AYAQUICA MARTINEZ	ayaquica@cs.buap.mx	ayaquica@gmail.com
9	VERONICA EDITH BAUTISTA LOPEZ	vbautista@cs.buap.mx	veromaster@gmail.com
10	CESAR BAUTISTA RAMOS	bautista@cs.buap.mx	
11	PEDRO BELLO LOPEZ	pbello@cs.buap.mx	
12	BEATRIZ BELTRAN MARTINEZ	bbeltran@cs.buap.mx	
13	MARIA BLANCA DEL CARMEN BERMUDEZ JUAREZ	bbj@cs.buap.mx	

**Figura C.8.3.** Formato correcto de la lista de profesores proporcionada por la Secretaría Académica.

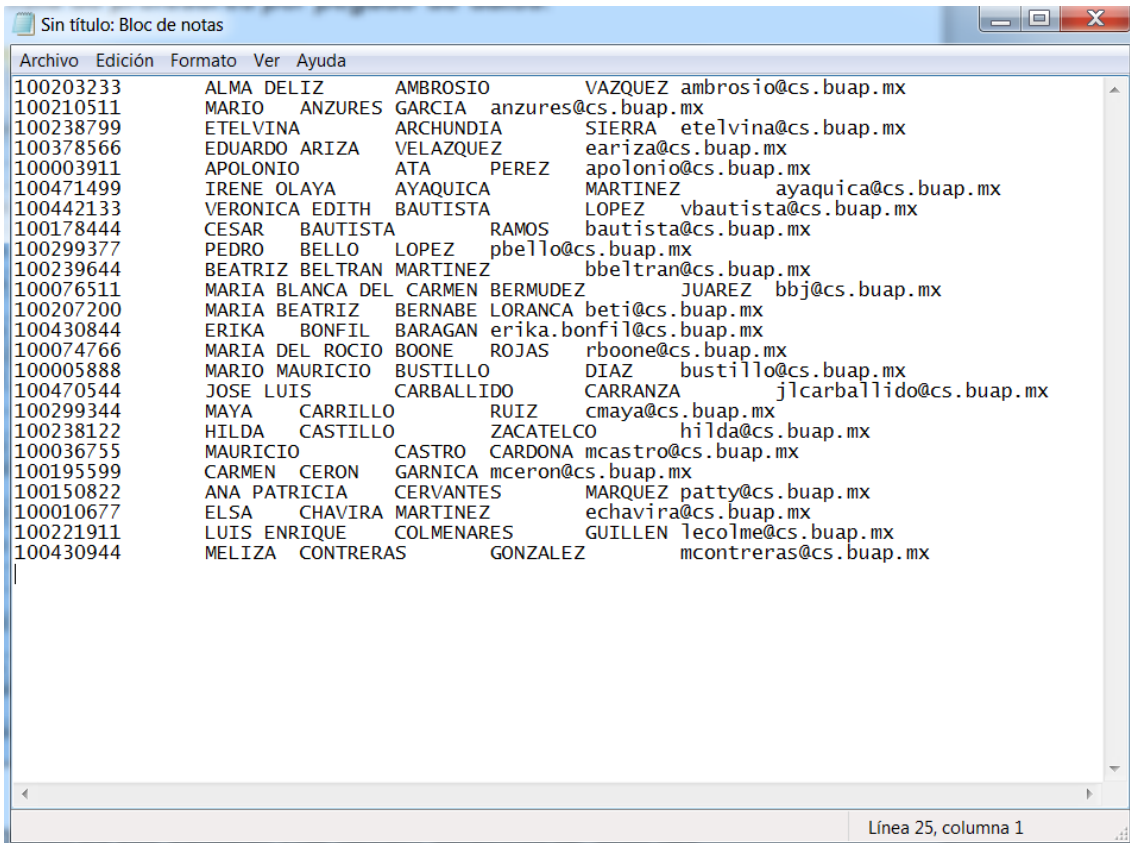
También puede realizarse el alta de maestros por pegado de datos, se selecciona la opción de copiar datos y posteriormente se presiona el botón *Pegar Datos*, y si la lista está en el formato correcto (Figura C.8.5 y Figura C.8.6) se realiza la inserción de los profesores y se muestra la lista (Figura C.8.7).



**Figura C.8.4.** Cargar la lista de profesores por pegado de datos.

	A	B	C	D	E	F	G	H	I
1	100203233			VAZQUEZ	ambrosio@cs.buap.mx				
2	100210511			GARCIA	anzures@cs.buap.mx				
3	100238799	ETELVINA	ARCHUNDIA	SIERRA	etelvina@cs.buap.mx				
4	100378566			IZA	eariza@cs.buap.mx				
5	100003911			TA	apolonio@cs.buap.mx				
6	100471499			UICA	avaquica@cs.buap.mx				
7	100442133			ISTA	vbautista@cs.buap.mx				
8	100178444			ISTA	bautista@cs.buap.mx				
9	100299377			LO	pbello@cs.buap.mx				
10	100239644			TRAN	bbeltran@cs.buap.mx				
11	100076511	MARIA		UDEZ	bbi@cs.buap.mx				
12	100207200			IABE	beti@cs.buap.mx				
13	100430844			IFIL	erika.bonfil@cs.buap.mx				
14	100074766			ONE	rboone@cs.buap.mx				
15	100005888			TILLO	bustillo@cs.buap.mx				
16	100470544			ALLID	jcarballido@cs.buap.mx				
17	100299344			RILLO	cmaya@cs.buap.mx				
18	100238122			TILLO	hilda@cs.buap.mx				
19	100036755			CASTRO	mcastro@cs.buap.mx				
20	100195599	CARMEN	CERON	GARNICA	mceron@cs.buap.mx				
21	100150822	ANA PATRICIA	CERVANTE S	MARQUEZ	patty@cs.buap.mx				
22	100010677	ELSA	CHAVIRA	MARTINEZ	echavira@cs.buap.mx				
23	100221911	LUIS ENRIQUE	COLMENAR ES	GUILLEN	lecolme@cs.buap.mx				
24	100430944	MELIZA	CONTRERA S	GONZALEZ	mcontreras@cs.buap.mx				
25									

**Figura C.8.5.** Cargar la lista de profesores por pegado (Se deben seleccionar solo las columnas que tengan información, en el formato que se muestra y posteriormente elegir la opción de copiar).



**Figura C.8.6.** Cargar la lista de profesores alterna por pegado (Se todo el texto en el formato que se muestra y posteriormente elegir la opción de copiar, el texto debe estar separado por una tabulación).

Una vez que se cargue la lista de profesores al sistema se mostrara en la pantalla, la contraseña para su ingreso será su número de trabajador, por lo cual será recomendable que después del primer ingreso cambien la contraseña (Figura C.8.7).



**Figura C.8.7.** Lista de profesores cargados en la Base de Datos.

En caso de que la lista no esté en formato correcto o ya no existan profesores que agregar se mostrara un mensaje indicándolo (Figura C.8.8).



Figura C.8.8. No se puede cargar la lista.

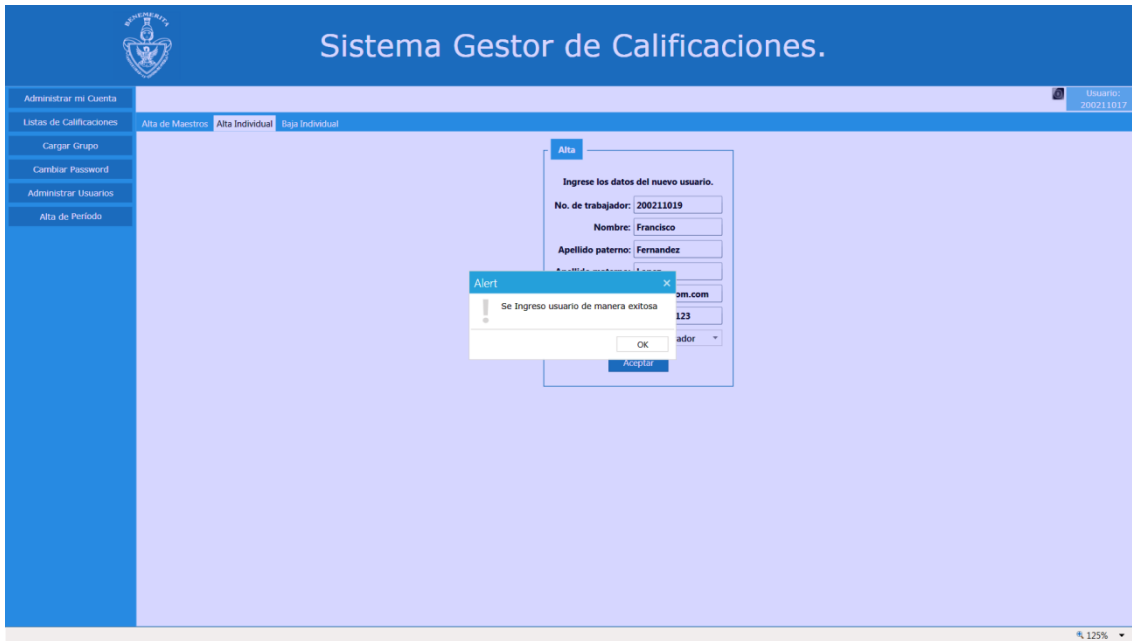
### C.8.2. Alta individual.

Para dar de alta profesores individualmente es necesario llenar el formulario (Figura C.8.9), seleccionar el tipo de usuario (en esta opción se pueden dar de alta usuarios con el rol de “Administrador” o “Usuario”, ya que en la opción anterior todos los que se dan de alta tienen solo los privilegios de “Usuario”), esta opción permite también asignar la contraseña, con la recomendación de cambiarla después del primer ingreso.



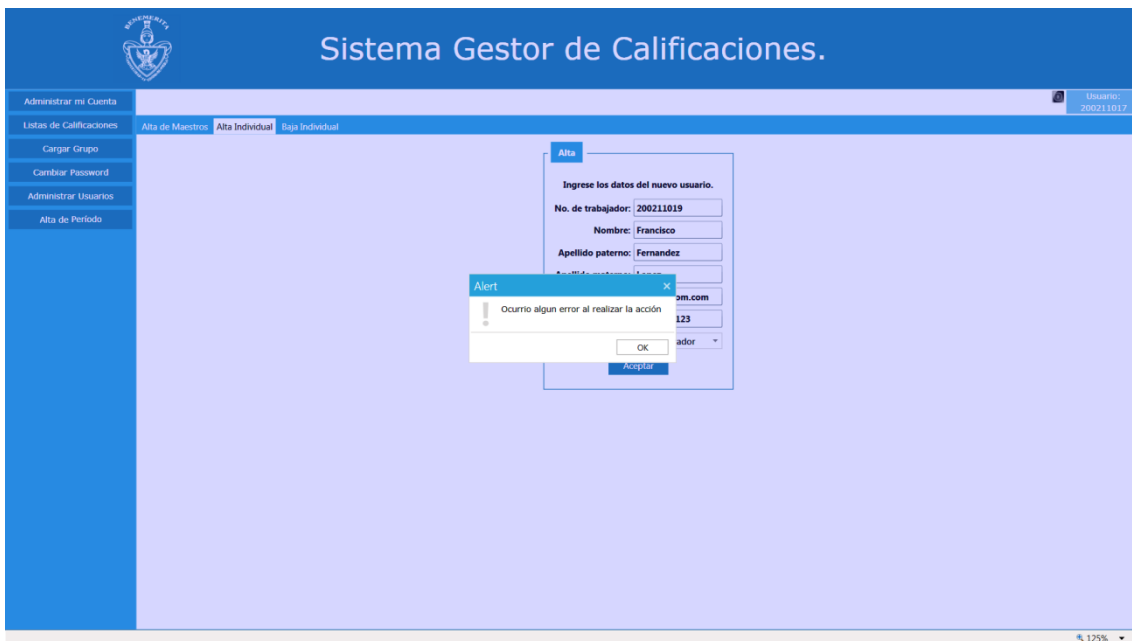
Figura C.8.9. Alta individual.

Una vez que se llene el formulario, si todo es correcto se habrá dado de alta un nuevo usuario (Figura C.8.10).



**Figura C.8.10.** *Usuario nuevo insertado.*

En caso de que exista algún error de conexión y no se complete el alta o el usuario ya exista se mostrara un mensaje de error (Figura C.8.11).



**Figura C.8.11.** *Error al insertar el usuario.*

En caso de que falten datos antes de insertar se mostrara el mensaje (Figura C.8.12), o en caso de que el formato de los datos ingresados no sea válido (Figura C.8.13), es decir, que en los nombres existan números o caracteres extraños y que el formato del correo sea el correcto.

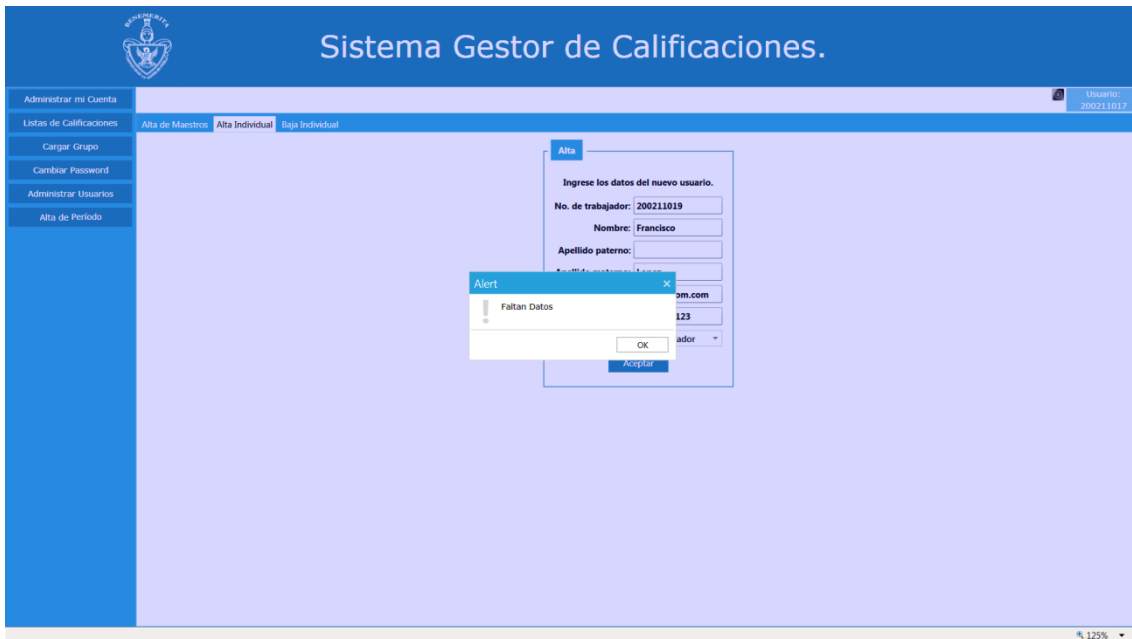


Figura C.8.12. Faltan datos para la inserción.



Figura C.8.13. Datos incorrectos para inserción.

### C.8.3. Baja individual.

La última opción es la baja de usuarios (Figura 8.14), es necesario conocer el identificador del profesor (número de trabajador), consultar los datos con un *Enter* y si existe se mostraran los datos (Figura C.8.15), en caso de que aún no esté dado de alta o no exista también se mostrara un mensaje de notificación (Figura C.8.16).

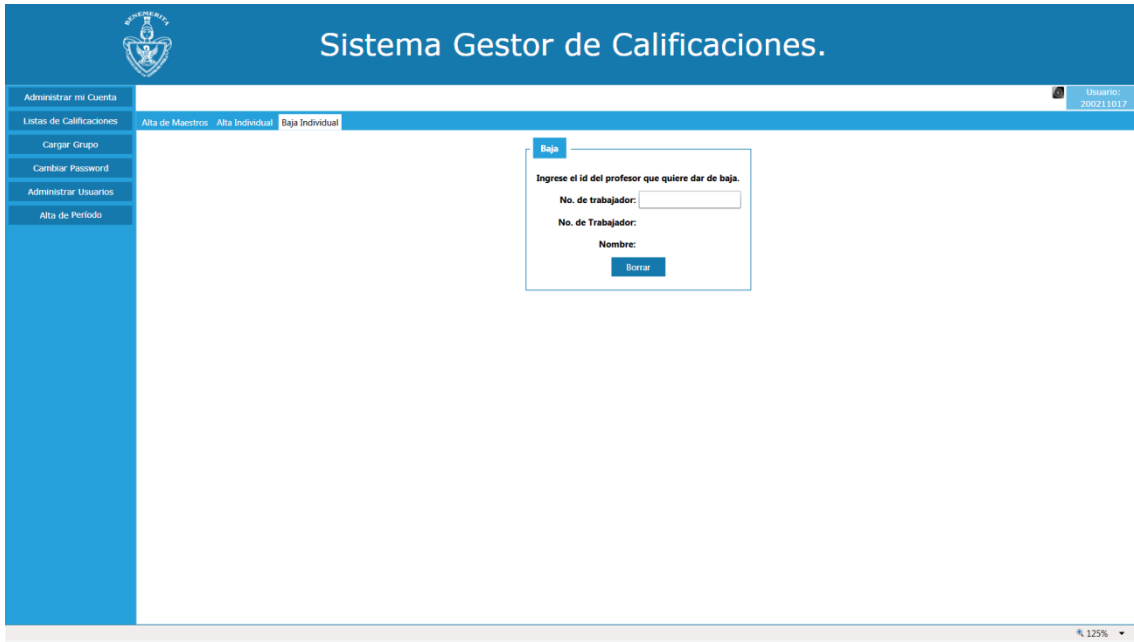
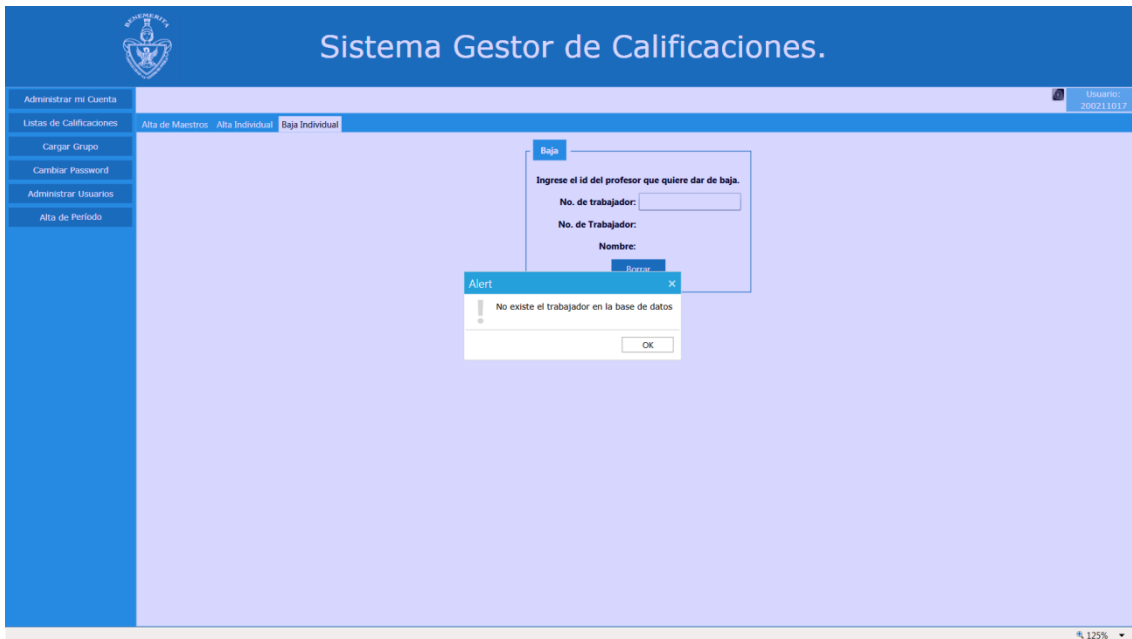


Figura C.8.14. Baja individual.

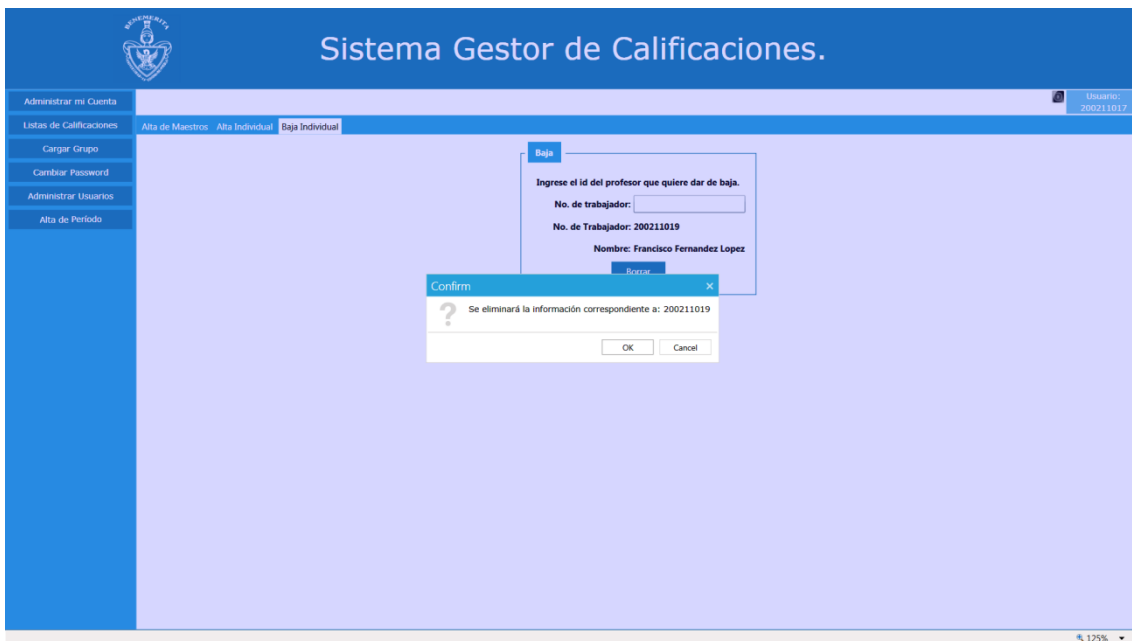


Figura C.8.15. Datos del usuario a dar de baja.



**Figura C.8.16.** El usuario no está registrado.

Una vez que se ha consultado la información del profesor que se quiere dar de baja, se puede borrar, antes de borrarlos se mostrara un mensaje para confirmar que se quiere eliminar de la *Base de Datos* (Figura C.8.17).



**Figura C.8.17.** Confirmación de la baja del profesor.

Una vez confirmado, se muestra un mensaje con el identificador del profesor que se dio de baja (Figura C.8.18).

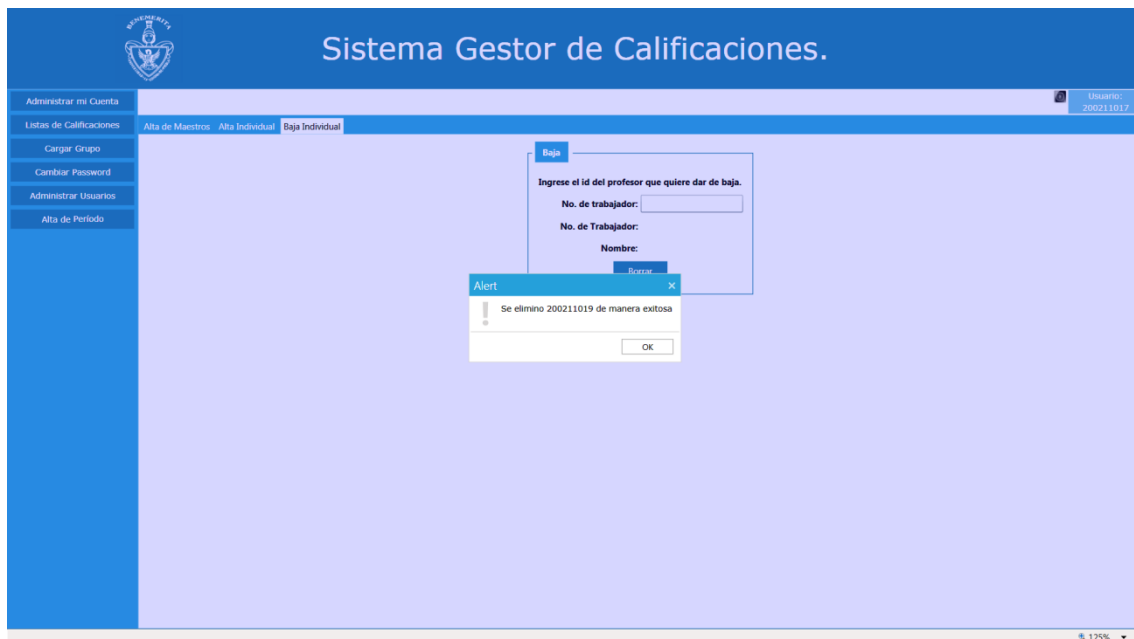


Figura C.8.18. Baja completa.

### C.9. Alta de período (Administrador).

Esta parte del menú está disponible para los usuarios de tipo “Administrador”, permite agregar nuevos periodos para los ciclos escolares (Figura C.8.19).

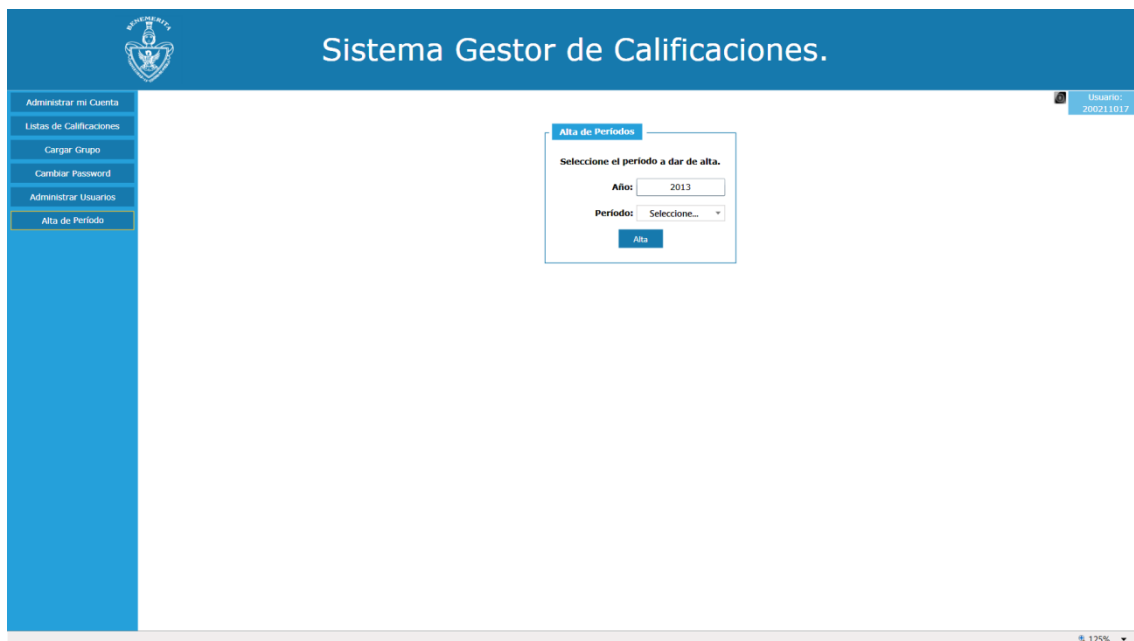
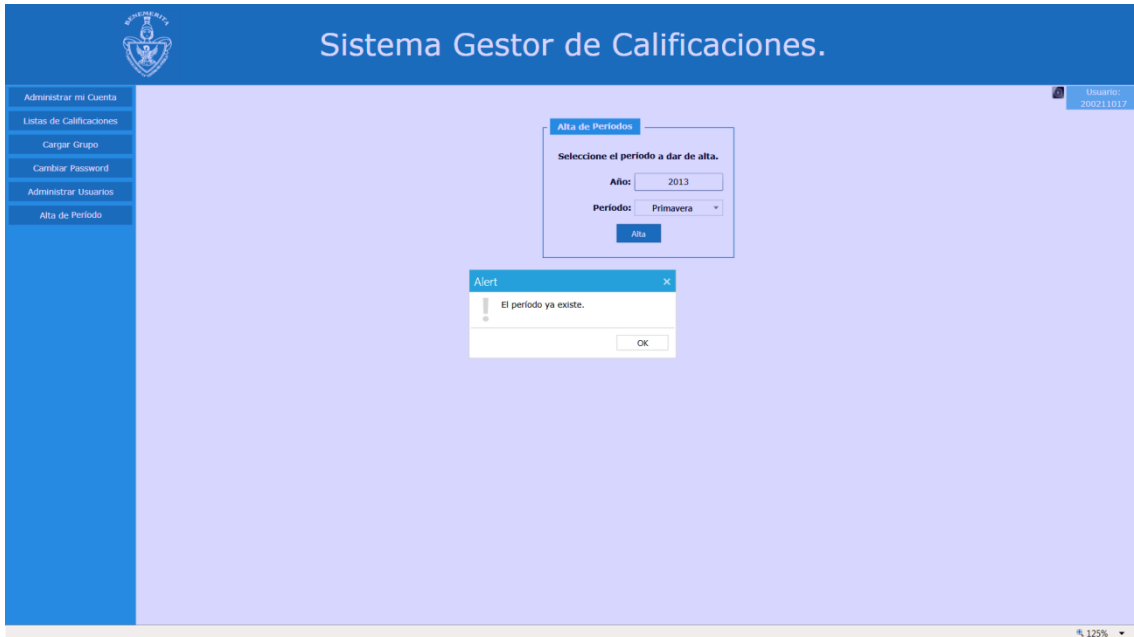


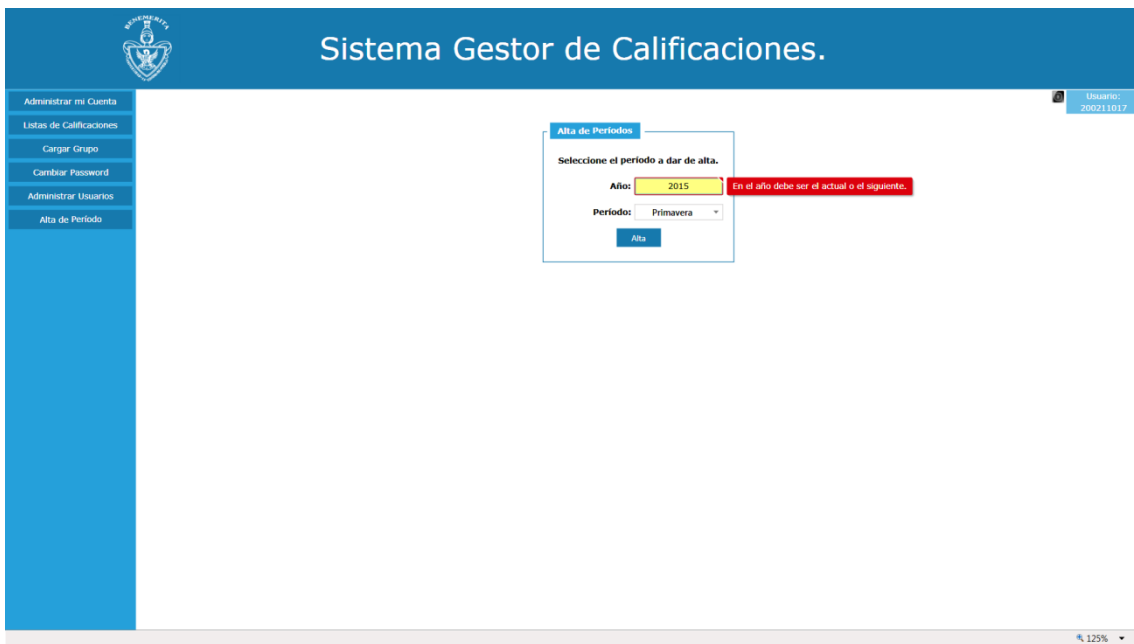
Figura C.8.19. Alta de períodos.

Solo se pueden insertar 3 periodos por año, que son: primavera, verano y otoño, en caso de que el periodo que se quiera añadir ya exista se notificara con una alerta (Figura C.8.20).



**Figura C.8.20.** El período ya existe.

Solo se pueden dar de alta periodos para el año actual o el siguiente año, en caso de ingresar otro año el sistema lo notificara y no permitirá realizar el alta (Figura C.8.21).



**Figura C.8.21.** Año incorrecto.

Si el periodo no existe, se podrá dar de alta y se notificara la alta del periodo (Figura C.8.22).

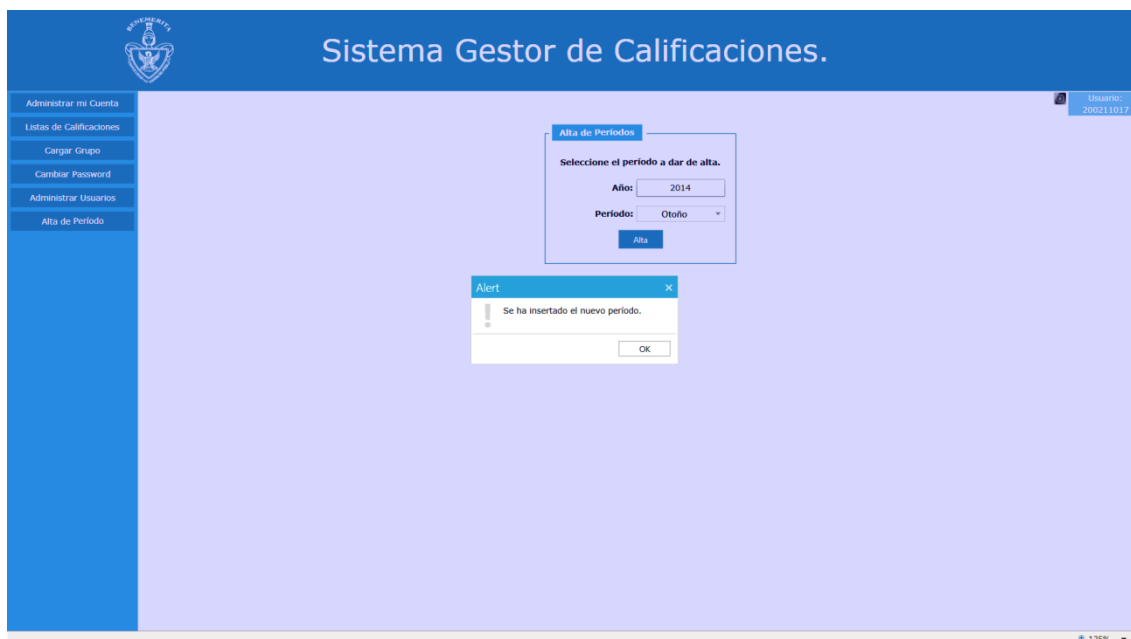


Figura C.8.22. Inserción de un nuevo período.

## D. Manual técnico.

### D.1. Descripción general.

En este documento se describen los módulos del programa y se da su descripción para futuras modificaciones.

El proyecto está realizado con el lenguaje de programación *C#*, *XAML* y *LINQ* se divide en dos partes (Figura D.1.1), la parte del proyecto *Silverlight* llamada “*Proyecto*” que es encapsulado en un archivo *.xap*, para poder ser consumido en el navegador web, la otra parte del proyecto es la parte web llamada “*Proyecto.Web*” que alojara con un *script* el proyecto *Silverlight* con ayuda del archivo *.xap* que se genera al compilar.

La herramienta principal para el desarrollo de la aplicación es *Visual Studio 2010*, incorporando *Crystal Reports* y controles *Telerik*, se utilizó *Expression Blend* para realizar efectos y como gestor de *Base de Datos SQL Server 2008*.

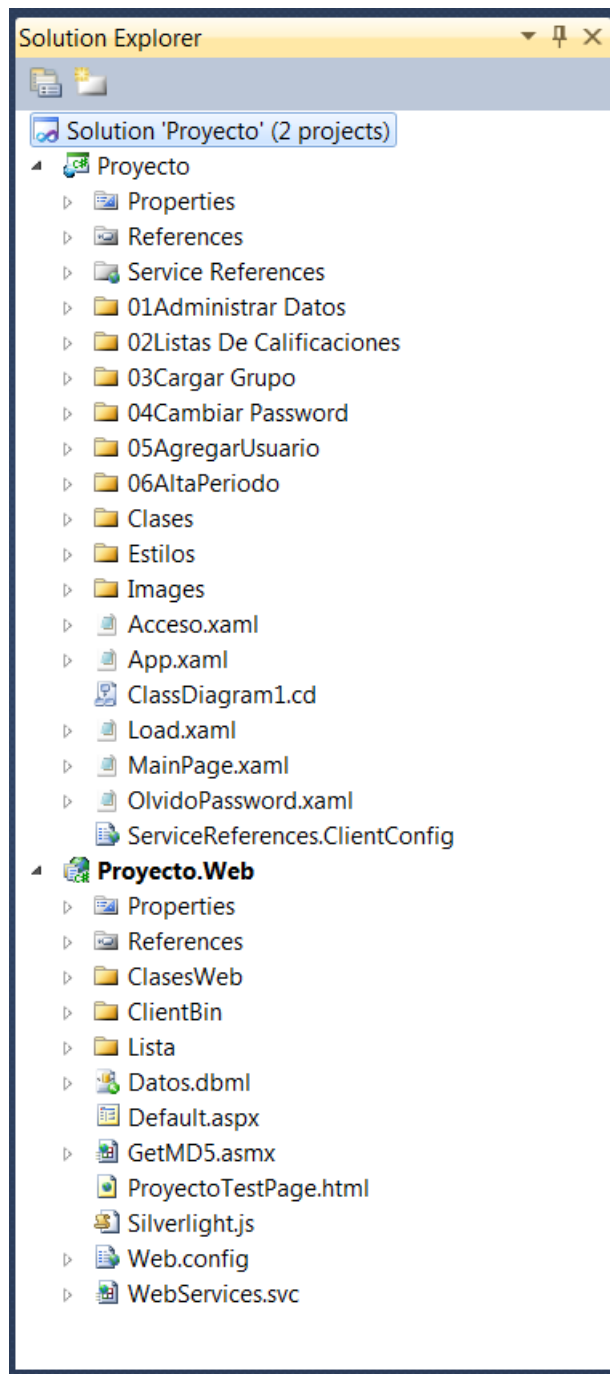
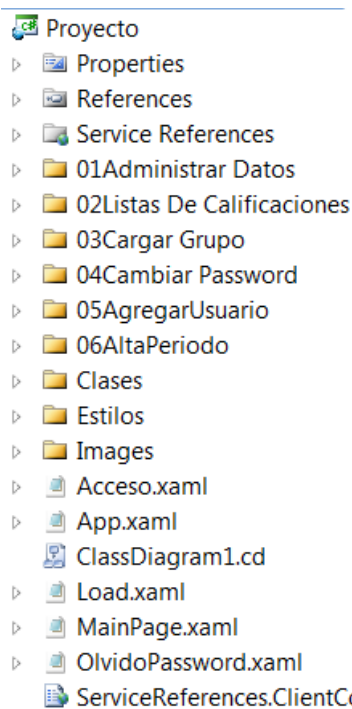


Figura D.1.1. Imagen de las carpetas del proyecto con Visual Studio 2010.

## D.2. Parte Silverlight “Proyecto”.

La parte correspondiente a “Proyecto” (Figura D.2.1), integra la mayor parte de la interfaz para el cliente, en esta parte se realizan las ventanas y controles que serán utilizadas durante la interacción del cliente con la aplicación y a continuación se describen las carpetas y archivos.



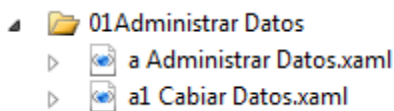
**Figura D.2.1.** Parte Silverlight del proyecto.

### D.2.1. Carpetas internas del proyecto.

Las carpetas *Properties*, *References* y *Service References* son propias de los proyectos de *Silverlight* donde *Properties* contiene tres archivos *AppManifest.xml*, *AssemblyInfo.cs* y *InBrowserSettings.xml* utilizados para declarar los manifiestos, ensamblados y permisos elevados con los que cuenta el proyecto, *References* contiene las referencias a las *dll*'s utilizadas en el proyecto y *Service References* contiene las referencias a los servicios web utilizados y alojados en el servidor.

### D.2.2. Administrar datos.

Esta carpeta contiene los controles para visualizar los datos y poderlos actualizar (Figura D.2.2 y Figura D.2.3).



**Figura D.2.2.** Carpeta Administrar Datos.

Control	Descripción	Funciones
Administrar Datos	Permite visualizar la información del usuario que ingreso a la aplicación.	<ul style="list-style-type: none"> <li>· a_Administrar_Datos.- es el constructor del UserControl.</li> <li>· ws_GetDatosCompleted.- De acuerdo al resultado del servicio web llena el formulario con los datos generales.</li> <li>· clicActualizar.-Evento del botón actualizar, manda a llamar al UserControl <i>Cambiar Datos</i>.</li> </ul>
Cambiar Datos	Permite cambiar los datos generales de	<ul style="list-style-type: none"> <li>· a1_Cabiar_Datos.- Constructor del UserControl.</li> </ul>

	los profesores.	<ul style="list-style-type: none"> <li>· ws_GetDatosCompleted.- De acuerdo al resultado del servicio web llena el formulario con los datos generales.</li> <li>· ws_UpdateDatosCompleted.- De acuerdo a los datos regresados por el servicio web para actualizar llena el formulario con los datos actualizados y re direcciona al UserControl <i>Administrar Datos</i>.</li> <li>· clicRegresar.-Evento de botón que permite regresar al UserControl anterior <i>Administrar Datos</i>.</li> <li>· clicActualizar.- Valida los campos y posteriormente manda a ejecutar el Servicio web para actualizar los datos.</li> </ul>
--	-----------------	--

Figura D.2.3. Descripción carpeta *Administrar Datos*.

**D.2.3. Listas de calificaciones.**

En esta carpeta se encuentran los controles para poder escoger el grupo que se quiere calificar o visualizarlo además de calificarlo (Figura D.2.4 y Figura D.2.5).

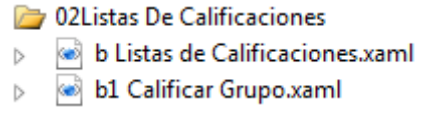


Figura D.2.4. Carpeta *Listas De Calificaciones*.

<b>Control</b>	<b>Descripción</b>	<b>Funciones</b>
Listas de Calificaciones	Permite elegir el grupo que se quiere calificar.	<ul style="list-style-type: none"> <li>· b_Listas_de_Calificaciones.- Constructor del UserControl, inicializa el combo de período con los periodos en los que el profesor tiene materias.</li> <li>· ws_GetMateriaCompleted.- Permite llenar un Combo Box con los elementos resultantes de una consulta con un servicio web de los grupos que tiene el profesor.</li> <li>· clicSiguiente.- Valida que se haya seleccionado un grupo y permite visualizar su contenido pasando al siguiente UserControl <i>Calificar Grupo</i>.</li> <li>· cambioSeleccion.- Evento que al cambiar de período seleccionado, manda a traer las materias correspondientes con ese período y con el profesor que está usando el sistema.</li> </ul>
Calificar Grupo	Permite calificar un	<ul style="list-style-type: none"> <li>· b1_Calificar_Grupo.- Constructor</li> </ul>

	<p>grupo, actualizar la lista, imprimirla, guardarla o eliminarla.</p>	<p>del UserControl.</p> <ul style="list-style-type: none"> <li>· ws_getPorcentCompleted.- Obtiene el porcentaje de cómo se va a calificar el grupo y llena el combo de tipo de calificaciones para poder agregar o quitar columnas dependiendo del resultado, valida que el grupo ya cuente con porcentaje y llena el resultado en un RadGridView, de no tenerlo re direcciona al UserControl correspondiente para asignar porcentajes.</li> <li>· inicializarGrupo.-Función que manda a llamar al servicio web correspondiente para obtener el Grupo a calificar.</li> <li>· ws_DeleteColumCompleted.- Regresa el resultado de borrar un tipo de columna y recarga el Grid con los resultados del grupo actualizados.</li> <li>· ws_InsertColumCompleted.- Regresa el resultado de insertar un tipo de columna y recarga el Grid con los resultados del grupo actualizados.</li> <li>· ws_GetDatosCompleted.- De acuerdo al resultado del servicio web llena el formulario con los datos generales.</li> <li>· Regresar_Click.- Evento de botón que permite regresar al UserControl anterior <i>Listas de Calificaciones</i>.</li> <li>· Aceptar_Click.- Valida que un RadioButton este checado, un elemento del Combo Box este seleccionado, para agregar o quitar columnas a la lista de calificaciones.</li> <li>· btnExcel_Click.-Abre una ventana para guardar la lista de calificaciones en un formato xml compatible con Excel.</li> <li>· rd_AutoGeneratingColumn.- Evento para construir en tiempo de ejecución las columnas necesarias para guardarse en el archivo xml que se abrirá con Excel.</li> <li>· btnPrint_Click.- Manda a imprimir el reporte en un nuevo link con la información correspondiente al</li> </ul>
--	--	---

		<p>grupo.</p> <ul style="list-style-type: none"> <li>· ws_UpDateGrupoCompleted.- Actualizada la Base de Datos con las calificaciones llena el Grid incluyendo los nuevos resultados finales.</li> <li>· btnDelete_Click.- Manda a ejecutar la función del servicio web correspondiente para eliminar la lista de calificaciones.</li> <li>· ws_DeleteGroupCompleted.- Dependiendo del resultado del servicio web, si se eliminó el grupo manda un mensaje indicándolo y re direccionando al UserControl <i>Listas de Calificaciones</i>, sino se invita a intentarlo posteriormente por posibles fallas de conexión.</li> <li>· clicGrabarGrupo.-Botón para mandar la lista de calificaciones a la Base de Datos y actualizarla, antes de invocar a la función del servicio web, valida los datos ingresados.</li> </ul>
--	--	--

**Figura D.2.5.** Descripción carpeta Listas De Calificaciones.

#### D.2.4. Cargar grupo.

En esta carpeta se encuentran los controles para poder cargar un archivo *Excel* a la base datos y hacer un grupo y posteriormente poderle asignar el tipo de calificación y el porcentaje con el que será evaluado el grupo (Figura D.2.6 y Figura D.2.7).



**Figura D.2.6.** Carpeta Cargar Grupo.

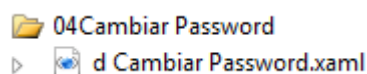
Control	Descripción	Funciones
Cargar Grupo	Permite elegir un archivo Excel de la máquina y subirlo a la Base de Datos.	<ul style="list-style-type: none"> <li>· c_Cargar_Grupo.-Constructor del UserControl.</li> <li>· ws_CargarGrupodeArchivoCompleted.- Una vez elegido el archivo, si el archivo tuvo el formato correcto se carga en el RadGridView, sino se muestra un mensaje.</li> <li>· ws_InsertarGrupoCompleted.- Ya que se insertó el grupo, si el resultado fue correcto se re direcciona al siguiente UserControl Asignar Calificaciones, sino se muestra un mensaje.</li> <li>· clicCargarGrupo.- Abre la venta para elegir el archivo a subir ayudado de un servicio web (CargarGrupodeArchivo).</li> </ul>

		<ul style="list-style-type: none"> <li>· clicSubirLista.- Evento de botón que valida que los datos sean correctos y los envía a un servicio web (CargarGrupodeArchivo) para insertar el grupo en la Base de Datos.</li> <li>· clicPegarGrupo.- Lee un texto copiado al Oprimir el botón <i>Pegar Datos</i>.</li> <li>· CargaDatos.- Carga los datos del grupo en pantalla o indica si el texto que se copió no tiene formato correcto.</li> <li>· soloNumeros.- Valida que solo se acepten caracteres numéricos.</li> <li>· OnPropertyChanged.- Notifica a una propiedad si existió algún cambio por parte de algún control, utilizado para las validaciones.</li> </ul>
Asignar Calificaciones	Permite elegir el criterio de evaluación del grupo.	<ul style="list-style-type: none"> <li>· c1_Asignar_Calificaciones.- Constructor del UserControl.</li> <li>· ws_InsertarGrupoCompleted.-Una vez asignado el porcentaje si todo fue correcto muestra un mensaje de éxito por la actualización y re direcciona al UserControl Calificar Grupo, en caso contrario muestra un mensaje de error.</li> <li>· cbexamen_Checked.- Habilita la visibilidad de los TextBox según las opciones seleccionadas.</li> <li>· cbexamen_Unchecked.- Deshabilita la visibilidad de los TextBox según las opciones que se quiten.</li> <li>· cambiaTexto.-Usa la función calculaTotal al cambiar el texto.</li> <li>· entraCaracter.- Valida que solo se acepten caracteres numéricos.</li> <li>· calculaTotal.- Habilita el botón para guardar cambios si el total de la suma del porcentaje es igual a 100.</li> <li>· clicGuardarPorcentaje.-Valida que el porcentaje sea igual a 100 y llama a la función del servicio web para actualizar el porcentaje.</li> </ul>

**Figura D.2.7.** Descripción carpeta Cargar Grupo.

### D.2.5. Cambiar contraseña.

En esta carpeta se encuentran el control para poder cambiar la contraseña actual (Figura D.2.7 y Figura D.2.8).



**Figura D.2.7.** Carpeta Cambiar Contraseña.

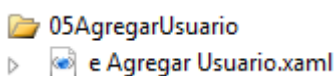
Control	Descripción	Funciones
Cambiar	Permite renovar	· d_Cambiar_Password.-Constructor del

Contraseña	la contraseña.	UserControl. <ul style="list-style-type: none"> <li>· ws_ValidaLoginCompleted.- se validan los datos que se quieren actualizar y el login anterior.</li> <li>· ws_UpDatePassCompleted.-Muestra si el resultado de la actualización se logró o no.</li> <li>· clicActualizar.-Evento del botón, valida los datos a enviar.</li> </ul>
------------	----------------	--

**Figura D.2.8.** Descripción carpeta Cambiar Contraseña.

### D.2.6. Agregar usuario.

En esta carpeta se encuentran el control para dar de alta o baja usuarios (Figura D.2.9 y Figura D.2.10).



**Figura D.2.9.** Carpeta Agregar Usuario.

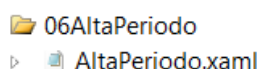
Control	Descripción	Funciones
Agregar Usuario	Permite dar de baja o alta usuarios con un rol.	<ul style="list-style-type: none"> <li>· e_Agregar_Usuario.-Constructor del UserControl.</li> <li>· ws_InsertUserCompleted.-Muestra en un mensaje si se realizó correctamente la inserción del usuario o no de acuerdo a la ejecución del servicio web.</li> <li>· clicAceptar.-Evento del botón Aceptar que valida que no falten datos antes de realizar la petición para que sean insertados.</li> <li>· btnConsultaDatos_Click.-Evento de botón que valida la información para llenar el formulario para borrar un usuario.</li> <li>· ws_DeleteUserCompleted.- Dependiendo del resultado obtenido por el servicio web para borrar, informa si fue exitosa la eliminación o no.</li> <li>· ws_GetDatosCompleted.- Verifica si existe un trabajador de acuerdo al resultado de un servicio web.</li> <li>· ws_InsertarListadeMaestrosCompleted.- De acuerdo a los datos obtenidos por un servicio web para insertar una lista de maestros, verifica si se pudieron insertar los nuevos maestros o no.</li> <li>· ws_InsertarListadeMaestrosPorPegadoCompleted.- De acuerdo a los datos obtenidos por un servicio web para insertar una lista de maestros, verifica si se pudieron insertar los nuevos maestros o no, por pegado.</li> <li>· clicCargarMaestros.- Evento de un botón que abre una nueva ventana para elegir un archivo Excel que debe contener información de los maestros y subir su información a la Base de Datos.</li> <li>· clicPegarProfesores.- Lee un conjunto de caracteres que hayan sido copiados y después pegados con el botón al que le pertenece este evento.</li> <li>· CargaMaestros.- convierte el texto leído en el</li> </ul>

		<p>evento anterior a un formato para ser insertado en la tabla de maestros.</p> <ul style="list-style-type: none"> <li>· Enter.- Valida para la baja de datos que solo se puedan ingresar números y que al dar <i>Enter</i> se llame al servicio para traer los datos de un profesor.</li> <li>· validaNumero.- Permite que solo se puedan ingresar números.</li> </ul>
--	--	---

**Figura D.2.10.** Descripción carpeta Agregar Usuario.

### D.2.7. Alta de períodos.

En esta carpeta se encuentran el control para dar de alta períodos (Figura D.2.11 y Figura D.2.12).



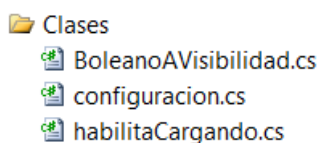
**Figura D.2.11.** Carpeta Alta Periodo.

Control	Descripción	Funciones
Alta Período	Permite dar de alta períodos.	<ul style="list-style-type: none"> <li>· AltaPeriodo.-Constructor del UserControl.</li> <li>· ws_InsertUserCompleted.-Muestra en un mensaje si se realizó correctamente la inserción del usuario o no de acuerdo a la ejecución del servicio web.</li> <li>· clicAlta.-Evento del botón Alta que valida que no falten datos antes de realizar la petición para que sea insertado un nuevo período.</li> <li>· btnConsultaDatos_Click.-Evento de botón que valida la información para llenar el formulario para borrar un usuario.</li> <li>· OnPropertyChanged.- Notifica a las propiedades si hay algún cambio, usado para validar datos.</li> <li>· entraAño.- Permite que solo se puedan ingresar números.</li> </ul>

**Figura D.2.12.** Descripción carpeta Alta Período.

### D.2.8. Clases.

Esta carpeta contiene clases auxiliares para el manejo global de variables y controles (Figura D.2.13 y Figura D.2.14).



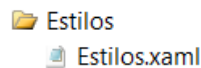
**Figura D.2.13.** Carpeta Clases.

Clase	Descripción	Funciones
Configuración	Permite controlar el identificador del maestro para poder utilizarlo en toda la aplicación.	<ul style="list-style-type: none"> <li>Identificador.- Variable global tipo texto y estática para manejar el identificador.</li> </ul>
habilitaCargando	Permite habilitar o deshabilitar un control cargando.	<ul style="list-style-type: none"> <li>openCargando.- manda a la pantalla principal el control cargando.</li> <li>closeCargando.- manda a cerrar en la pantalla principal el control cargando.</li> </ul>
BooleanoAVisibilidad	Implementa la interfaz IValueConverter, para convertir un valor bool a visibility	<ul style="list-style-type: none"> <li>Convert.- Convierte un valor true a visible y un valor false a collapsed.</li> <li>ConvertBack.- Método no implementado, pero creado al implementar la interfaz IValueConverter.</li> </ul>

**Figura D.2.14.** Descripción carpeta Clases.

### D.2.9. Estilos.

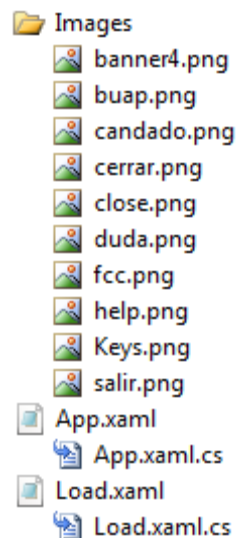
Esta carpeta contiene un diccionario de recursos, para establecer estilos en los controles y poder utilizarlos a cualquier nivel de la aplicación (Figura D.2.15).



**Figura D.2.15.** Carpeta de Estilos.

### D.2.10. Images, App y Load.

La carpeta *Images* contiene las imágenes contenidas en el proyecto que se usan como botones o símbolos y los archivos *App*, se encargan de tener los diccionarios que se utilizan en la aplicación y la comunicación entre ella, y los archivos *Load* contienen al control que aparece al realizar una consulta para controlar los eventos y congelar la pantalla principal mientras se ejecuta la tarea (Figura D.2.16).



**Figura D.2.16.** Carpeta Images, archivos App y Load.

### D.2.11. MainPage.

Este es el control principal de la aplicación, ya que en él se inicializan los demás controles ya que están contenidos ahí y es la pantalla principal (Figura D.2.17 y Figura D.2.18).

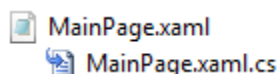


Figura D.2.17. Control MainPage.

Control	Descripción	Funciones
MainPage.	Contiene y controla todos los controles.	<ul style="list-style-type: none"><li>· MainPage.-Constructor del UserControl.</li><li>· Salida.- Función que permite que al salir de la aplicación se inhabilite la llave de acceso.</li><li>· clicSalir.- Al oprimir la imagen de salir se lanza un evento para limpiar las variables y reiniciar la aplicación.</li><li>· clicLogin.- Valida los datos en el formulario del login y de ser correctos inicial el servicio web para validarlos, en caso contrario muestra un mensaje mostrando el error.</li><li>· clicOlvidoPass.- Evento de botón que abre la ventana que contiene el formulario <i>olvidoPassword</i>.</li><li>· opcionMenu.- De acuerdo a un elemento del menú seleccionado, carga en el contenido la interfaz correspondiente al menú seleccionado.</li></ul>

Figura D.2.18. Descripción del control MainPage.

### D.2.12. OlvidoPassword.

En caso de olvidar la contraseña de acceso al sistema, este control permite recuperar la contraseña recuperándola vía mail (Figura D.2.19 y Figura D.2.20).

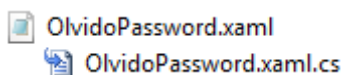


Figura D.2.19. Control OlvidoPassword.

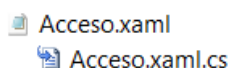
Control	Descripción	Funciones
OlvidoContraseña.	Permite recuperar la contraseña vía mail.	<ul style="list-style-type: none"><li>· OlvidoPassword.-Constructor del UserControl.</li><li>· ws_RecuperaPassCompleted.- Una vez ejecutado el servicio web correspondiente para recuperar la contraseña, de acuerdo al resultado indica si se pudo recuperar o no.</li><li>· Button_Click.- Evento de botón que valida que se ingrese el identificador del trabajador y manda a llamar al</li></ul>

		<p>servicio web para validar los datos y enviar el correo con la contraseña.</p> <ul style="list-style-type: none"> <li>· validaCaracter.- Valida que en el número de trabajador solo se ingresen números.</li> </ul>
--	--	---

**Figura D.2.20.** Descripción del control OlvidoContraseña.

### D.2.13. Acceso.

Control que permite realizar el ingreso al sistema (Figura D.2.21 y Figura D.2.22).



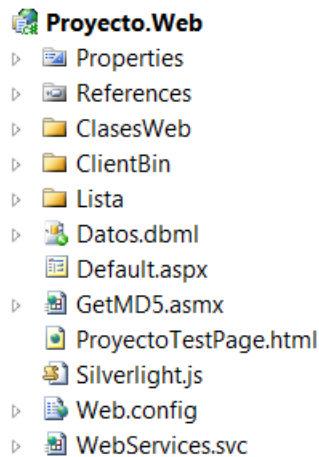
**Figura D.2.21.** Control Acceso.

Control	Descripción	Funciones
Acceso.	Permite ingresar al sistema.	<ul style="list-style-type: none"> <li>· Acceso.-Constructor del UserControl.</li> <li>· ws_ValidaLoginCompleted.- Dependiendo del resultado de la validación del login si fue correcto habilita la pantalla principal, de no serlo muestra un mensaje con el error correspondiente.</li> <li>· clicLogin.- Valida los datos en el formulario de acceso y de ser correctos inicia el servicio web para validarlos, en caso contrario muestra un mensaje mostrando el error.</li> <li>· clicOlvidoPass.- Evento de botón que abre la ventana que contiene el formulario <i>olvidoPassword</i>.</li> <li>· enterKeyDown.- Al estar posicionado en el campo de la contraseña, si se tecléa <i>Enter</i> llama al evento clicLogin.</li> <li>· enterKeyUsuario.- Valida que solo se puedan ingresar números al número de maestro.</li> </ul>

**Figura D.2.22.** Descripción del control Acceso.

### D.3. Parte web “Proyecto.Web”.

La parte correspondiente a “Proyecto.Web” (Figura D.3.1), es la encargada de contener la aplicación *Silverlight* y es lo que se publica en el servidor, además de contener el mapeo a la *Base de Datos* y los servicios web necesarios para consultar los datos de la *Base de Datos*, y también contiene el reporte que se muestra para imprimir en otra página web.



**Figura D.3.1.** Estructura del proyecto en su parte web.

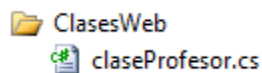
### D.3.1. Carpetas internas del proyecto web.

Las carpetas *Properties*, *References*, *Web.config* y *ClientBin* son propias de los proyectos web donde *Properties* contiene el archivo *AssemblyInfo.cs* utilizado para declarar los ensamblados, *References* contiene las referencias a las *dll*'s utilizadas en el proyecto y *Web.config* contiene las referencias a las conexiones a la *Base de Datos* y declaración de otras variables. La carpeta *ClientBin* contiene el archivo *Proyecto.xap* que contiene la aplicación *Silverlight*.

También contiene el archivo *Default.aspx*, que es a donde se contiene con ayuda de scripts la aplicación *Silverlight* con ayuda del archivo *Silverlight.js*.

### D.3.2. Clases web.

Esta carpeta contiene clases auxiliares para los servicios web o para el proyecto *Silverlight* (Figura D.3.2 y Figura D.3.3).



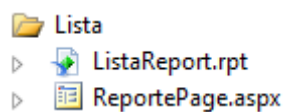
**Figura D.3.2.** Carpeta *ClasesWeb*.

Clase	Descripción
claseProfesor	Permite crear Objeto de tipo Profesor para crear filas y mandarse entre el servicio web y el proyecto Silverlight.

**Figura D.3.3.** Descripción carpeta *ClasesWeb*.

### D.3.3. Lista.

Esta carpeta contiene el reporte para mostrar la lista de calificaciones y la página que lo contiene (Figura D.3.4 y Figura D.3.5).



**Figura D.3.4.** Carpeta *Lista*.

Archivo	Descripción
ListaReport.rpt	Contiene el diseño del CrystalReport y la referencia a la Base de Datos.
ReportePage.aspx	Inicializa el reporte y lo carga ya que es una página web aspx.

Figura D.3.5. Descripción carpeta Lista.

### D.3.4. Datos.

Este archivo contiene el modelo o mapeo de la *Base de Datos* para poder trabajar con un modelo orientado a objetos con los datos de la *Base de Datos* (Figura D.3.5).

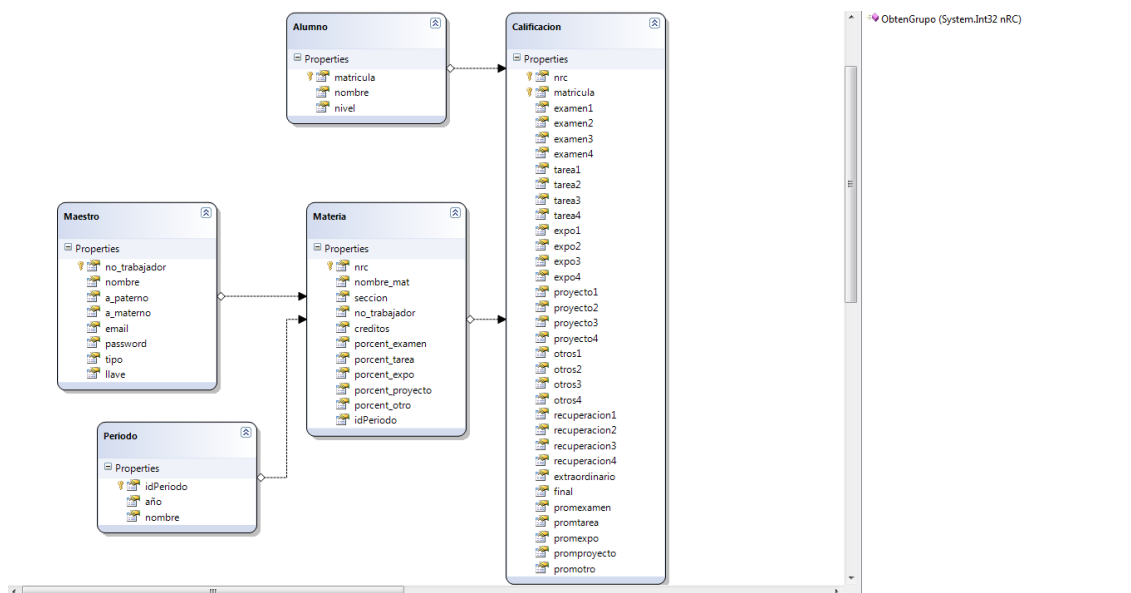


Figura D.3.5. Modelo de la Base de Datos.

### D.3.5. GetMD5.

Este archivo es un servicio web con extensión *asmx* y contiene las funciones para codificar información y también decodificarla (Figura D.3.6).

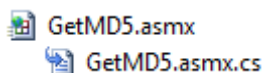


Figura D.3.6. Archivos GetMD5.

### D.3.6. Web services.

En este archivo se encuentran los servicios web restantes o funciones (Figura D.3.7 y Figura D.3.18) y que a continuación se describen.

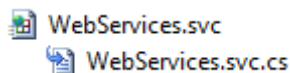


Figura D.3.7. Archivos WebServices.

Función	Descripción	Parámetros que recibe.	Que devuelve.
getPorcent	Obtiene el porcentaje con el cual se califica	El NRC del Grupo.	Un objeto de tipo ObtenGrupoResult que es un grupo y es del tipo

	un grupo.		devuelto por un Stored Procedure.
ValidaLogin	Valida si los datos ingresados son válidos.	El identificador de usuario y contraseña.	Un objeto de tipo <i>accesoUsuario</i> , que está formado de un valor booleano para indicar si existe el usuario, una cadena de caracteres para saber qué tipo de rol tiene el usuario y otra cadena, que es la llave de acceso para modificaciones en la Base de Datos.
AlSalir	Limpia la llave del profesor, por seguridad al salir de la aplicación.	El número de trabajador del profesor.	Nada.
CargarGrupodeArchivo	Lee un archivo en Excel y lo almacena en una variable.	La ruta del archivo Excel.	List<filaGrupoCargado>
InsertarGrupo	Inserta un nuevo grupo a la Base de Datos.	Una lista del grupo cargado, el identificador del profesor, el NRC, sección, nombre de la materia, tipo de inserción, porcentaje examen, porcentaje tarea, porcentaje exposición, porcentaje proyecto, porcentaje de otro, identificador del Periodo y la llave.	Valor booleano.
GetDatos	Obtiene los datos del profesor.	El identificador del profesor.	List<filaMaestro>
UpDateDatos	Actualiza los datos del profesor.	El identificador del profesor, nombre, apellido paterno, apellido materno, correo	Valor booleano.

		electrónico y llave.	
UpDatePass	Actualízala contraseña.	El identificador del profesor, la nueva contraseña y llave.	Valor booleano.
GetMateria	Obtiene los grupos que tiene cargados un profesor.	El identificador del profesor y el período.	List<filaMateria>
UpDateCalifGrupo	Actualiza el porcentaje de cómo se califica un grupo.	El NRC, valor entero para tarea, examen, proyecto, expo, otro y llave.	Valor booleano.
UpDateGrupo	Actualiza la lista de calificaciones.	La lista de calificaciones, el NRC del grupo y llave.	List<ObtenGrupoResult>
InsertColum	Inserta algún tipo de calificación.	El NRC del grupo, el tipo de calificación a agregar y llave.	List<ObtenGrupoResult>
DeleteColum	Borra algún tipo de calificación.	El NRC del grupo, el tipo de calificación a borrar y llave.	List<ObtenGrupoResult>
GetGrupo	Obtiene la lista de calificaciones.	El NRC del Grupo.	List<ObtenGrupoResult>
DeleteGroup	Borra un grupo de la Base de Datos.	El NRC del Grupo y llave.	Valor booleano.
RecuperaPass	Manda vía correo electrónico la contraseña.	El identificador del Profesor.	Valor booleano.
InsertUser	Inserta un nuevo profesor en la Base de Datos.	Identificador del profesor, nombre, apellido paterno, apellido materno, correo, tipo de usuario, contraseña y llave.	Valor booleano.
DeleteUser	Borra un profesor de la Base de	Identificador del profesor y	Valor booleano.

	Datos.	llave.	
InsertarListadeMaestros	Dada la ruta de un archivo Excel con la información de los profesores, inserta a cada profesor y regresa una lista de los profesores que se insertaron en la Base de Datos.	La ruta del archivo y llave.	List<claseProfesor>
VerificaCorreos	Verifica si un maestro tiene correo institucional y de no ser así, si tiene un correo alterno y lo regresa para ser insertado por la función <i>InsertUser</i> .	El correo institucional en una cadena y en otro los o el correo alternativo.	Una cadena con el correo apropiado.
InsertarPeriodo	Permite insertar un nuevo período.	Año del periodo, nombre y llave.	Valor booleano.
ObtenPeriodos	Obtiene la lista de todos los periodos.	Nada	List<Periodo>
ObtenPeriodosPorProfesor	Obtiene los periodos en los cuales el profesor ha tenido materias.	Número de trabajador del profesor.	List<Periodo>
ValidaLLave	Valida que la llave exista en la Base de Datos.	La Llave.	Valor booleano.

**Figura D.3.8.** Descripción de las funciones del archivo *WebServices*.