19/01/2014

SISTEMA AUTOMATIZADO DE INCLUSIONES

Manual Técnico del Sistema

Versión 1.0

Preparado por: Ana Irina Calvo Carvajal

Andrés Eduardo González Ortíz

Jose Arnoldo Segura Campos

Juan José Rojas Valverde

Tabla de Contenidos

[1 Control del Documento 3](#_Toc377930480)

[1.1 Historial de cambios 3](#_Toc377930481)

[1.2 Aprobación del documento 3](#_Toc377930482)

[2 Requisitos 4](#_Toc377930483)

[2.1 Visión General 4](#_Toc377930484)

[2.1.1 Objetivo General 4](#_Toc377930485)

[2.1.2 Objetivos Específicos 4](#_Toc377930486)

[2.2 Especificación Revisada 5](#_Toc377930487)

[2.3 Manual de Usuario 6](#_Toc377930488)

[2.3.1 Plataforma Web 6](#_Toc377930489)

[2.3.2 Plataforma Móvil 12](#_Toc377930490)

[2.4 Funcionamiento 16](#_Toc377930491)

[2.4.1 Ambiente Operativo 16](#_Toc377930492)

[2.4.2 Rendimiento 16](#_Toc377930493)

[2.4.3 Tamaño 16](#_Toc377930494)

[2.5 Análisis del Problema 18](#_Toc377930495)

[2.5.1 Contexto del Sistema 18](#_Toc377930496)

[2.5.2 Diagrama de Secuencia de los Casos de Uso 20](#_Toc377930497)

[2.5.3 Diagrama de Modelo de Dominio 26](#_Toc377930498)

[3 Diseño 27](#_Toc377930499)

[3.1 Visión General 27](#_Toc377930500)

[3.2 Estructura en Tiempo de Ejecución 27](#_Toc377930501)

[3.3 Estructura del Módulo 28](#_Toc377930502)

[4 Pruebas 29](#_Toc377930503)

[4.1 Estrategia 29](#_Toc377930504)

[5 Reflexión 30](#_Toc377930505)

[5.1 Evaluación 30](#_Toc377930506)

[5.2 Lecciones 30](#_Toc377930507)

[5.3 Errores y Limitaciones Conocidos 30](#_Toc377930508)

[6 Apéndice 31](#_Toc377930509)

[6.1 Formatos 31](#_Toc377930510)

[6.1.1 Estándares de Programación 31](#_Toc377930511)

[6.1.2 Estándar de Programación Base de Datos 31](#_Toc377930512)

[6.2 Especificaciones del Módulo 31](#_Toc377930513)

[6.2.1 Capa de Presentación 31](#_Toc377930514)

[6.2.2 Capa de Negocio 31](#_Toc377930515)

[6.2.3 Capa de Datos 31](#_Toc377930516)

[6.3 Casos de Prueba 31](#_Toc377930517)

# Control del Documento

## Historial de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Autor | Cambios realizados |
| 0.0 | 16/01/2014 | Juan José Rojas Valverde | Creación de la plantilla del Manual Técnico del Sistema |
| 0.1 | 17/01/2014 | Juan José Rojas Valverde | Modificación Sección Requisitos |
| 0.2 | 18/01/2014 | Juan José Rojas Valverde | Modificación Sección Diseño |
| 0.3 | 19/01/2014 | Juan José Rojas Valverde | Modificación Sección Pruebas |
| 1.0 | 19/01/2014 | Juan José Rojas Valverde | Adición de imágenes en Manual de Usuario, Diseño |

## Aprobación del documento

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Nombre | Título | Firma |
| 11/12/2013 | Adriana Álvarez Figueroa | Profesora de la Escuela de Ingeniería en Computación del TEC (cliente) |  |
| 11/12/2013 | Jaime Solano Soto | Profesor del curso de Proyecto de la Escuela de Ingeniería en Computación del TEC |  |

# Requisitos

## Visión General

El Sistema Automatizado de Inclusiones, o IncluTec, es un software dirigido a los estudiantes y profesores de carrera de la Escuela de Ingeniería en Computación. Esta aplicación fue creada para una plataforma de ambiente web mediante la cual los estudiantes pueden realizar en línea el trámite de solicitar una inclusión y la consulta de resultados sobre la misma, y además permitir a los encargados del proceso de inclusiones tramitar dichas solicitudes y presentar los resultados al departamento correspondiente. Además, el proyecto cuenta con una aplicación diseñada para dispositivos móviles con el Sistema Operativo Android mediante la cual los estudiantes pueden realizar el trámite y consulta de las solicitudes.

### Objetivo General

Desarrollar un sistema que administre las solicitudes de inclusiones de la escuela de Ingeniería en Computación del Tecnológico de Costa Rica de manera electrónica y automatizada utilizando plataformas web y móviles.

### Objetivos Específicos

Como objetivos específicos del proyecto, se pueden mencionar los siguientes:

1. Permitir al estudiante crear una solicitud de inclusión en la que se puedan escoger el curso a solicitar, así como los grupos de interés en un orden de prioridad deseado; ya sea mediante el uso de la plataforma web, o mediante el uso de la aplicación para Android desde su dispositivo móvil.
2. Permitir al estudiante la consulta sobre una solicitud tramitada para saber el estado de la misma; ya sea mediante el uso de la plataforma web, o mediante el uso de la aplicación para Android desde su dispositivo móvil.
3. Generar un período de recepción de solicitudes de inclusión durante el cual los estudiantes puedan hacer los trámites necesarios para su participación en el proceso.
4. Permitir al estudiante el poder modificar los grupos deseados en una solicitud de inclusión en particular; ya sea mediante el uso de la plataforma web, o mediante el uso de la aplicación para Android desde su dispositivo móvil.
5. Permitir que los estudiantes anulen el trámite de una solicitud de inclusión en particular; ya sea mediante el uso de la plataforma web, o mediante el uso de la aplicación para Android desde su dispositivo móvil.
6. Permitir al coordinador de carrera la gestión adecuada de los casos excepcionales de un proceso de inclusión en el sistema para su priorización en la asignación automática de cupos.
7. Permitir la consulta e impresión de las solicitudes como parte de los resultados del proceso automático de asignación de cupos.
8. Permitir la consulta e impresión del reporte de sugerencias como parte de los resultados del proceso automático de asignación de cupos.
9. Gestionar de manera adecuada las Reglas de Proceso a aplicar previo al proceso automático de asignación de cupos.
10. Habilitar o deshabilitar la ejecución de una determinada regla de proceso a aplicar previo al proceso automático de asignación de cupos.
11. Determinar la prioridad de ejecución de una determinada regla de proceso a aplicar previo al proceso automático de asignación de cupos.
12. Permitir el envío de notificaciones al estudiantado acerca de los resultados de sus solicitudes posterior a la aplicación del proceso automático de asignación de cupos.
13. Ejecutar de manera automática el proceso que permite asignar los cupos disponibles para el proceso de inclusión, con base en las excepciones y reglas previamente establecidas.

## Especificación Revisada

Entre los requerimientos principales del cliente se encuentra que el sistema sea capaz de llevar a cabo todo el trámite de inclusión desde que un estudiante crea una solicitud, hasta que el sistema de manera automática apruebe o rechace la misma. Debe ser también capaz de enviar notificaciones a los estudiantes y profesores, generar reportes de resultados y de sugerencias para la apertura o ampliación de cupos y consultar el historial de solicitudes realizadas.

Según los requerimientos del cliente, el sistema podrá recibir en promedio unas 1400 solicitudes de inclusión anuales, por lo que se deduce que análogamente se generarán 1400 entradas aproximadamente en la tabla principal del sistema de datos. Además, se debe tomar en consideración el crecimiento de la población estudiantil que utilizará este sistema en un futuro cercano, debido a la apertura y crecimiento de la carrera de Ingeniería en Computación en las sedes de Alajuela, Centro Académico y Limón, ya que la cantidad de usuarios se vería reflejada directamente en un incremento aún mayor en el promedio anteriormente mencionado.

Entre las características del nuevo sistema tenemos que:

* Permite realizar las solicitudes de inclusión mediante el acceso vía internet.
* Es escalable.
* Compatible con dispositivos móviles.
* Multiplataforma.
* Interfaz de usuario amigable e intuitiva.
* Integración con sistemas institucionales existentes.

El sistema tiene las siguientes dependencias:

* El Departamento de Admisión y Registro debe proveer las interfaces de comunicación con los sistemas institucionales para el sistema que se desarrollará.
* La Escuela de Ingeniería en Computación debe proveer la infraestructura física necesaria para la ejecución continua del sistema a desarrollar.

## Manual de Usuario

A continuación se presenta el manual de usuario de la aplicación escrita en el documento.

### Plataforma Web

#### Acceso al Sistema

Al ser una aplicación creada para una plataforma web ésta puede ser accedida desde cualquier navegador de internet, por lo cual se supone un conocimiento básico en el manejo de algún navegador de internet.

Para ingresar al sistema ingresamos en el campo de la URL del navegador de internet la siguiente dirección: <http://iis.ic-itcr.ac.cr/inclutec>



#### Sección de Autenticación

Para ingresar al sistema, es necesario ser estudiante de la institución, o un funcionario con credenciales de administrador (por lo general el Coordinador de la Carrera), por lo que al ingresar a la dirección del sistema se desplegará una pantalla para la autenticación de cada uno de ellos.

Esta sección es obligatoria para todos los usuarios del sistema y permite o rechaza el acceso a las demás secciones del sistema.



En caso de ser estudiante:

* El Nombre de Usuario es el número de carnet estudiantil.
* La Contraseña es el PIN de estudiante brindado por Admisión y Registro.

En caso de ser administrador:

* El Nombre de Usuario es el mismo con el que el funcionario ingresa a los sistemas del TEC.
* La Contraseña es la misma con la que el funcionario ingresa a los sistemas del TEC.

#### Sección del Estudiante

El estudiante, al ingresar al sistema, procede a observar la siguiente pantalla, mediante la cual se pueden revisar las solicitudes divididas en 4 estados: Solicitudes Pendientes, Solicitudes Anuladas, Solicitudes Aprobadas y Solicitudes Rechazadas.



Dentro de la sección, se puede acceder al área de Solicitudes, en la cual se puede acceder al formulario de solicitud de inclusión:



#### Sección de Administrador

Esta sección se encuentra dividida en 3 áreas principales: Reglas, Herramientas y Consultas. Al ingresar a la sección de coordinador se muestra la siguiente pantalla:



##### Área de Reglas

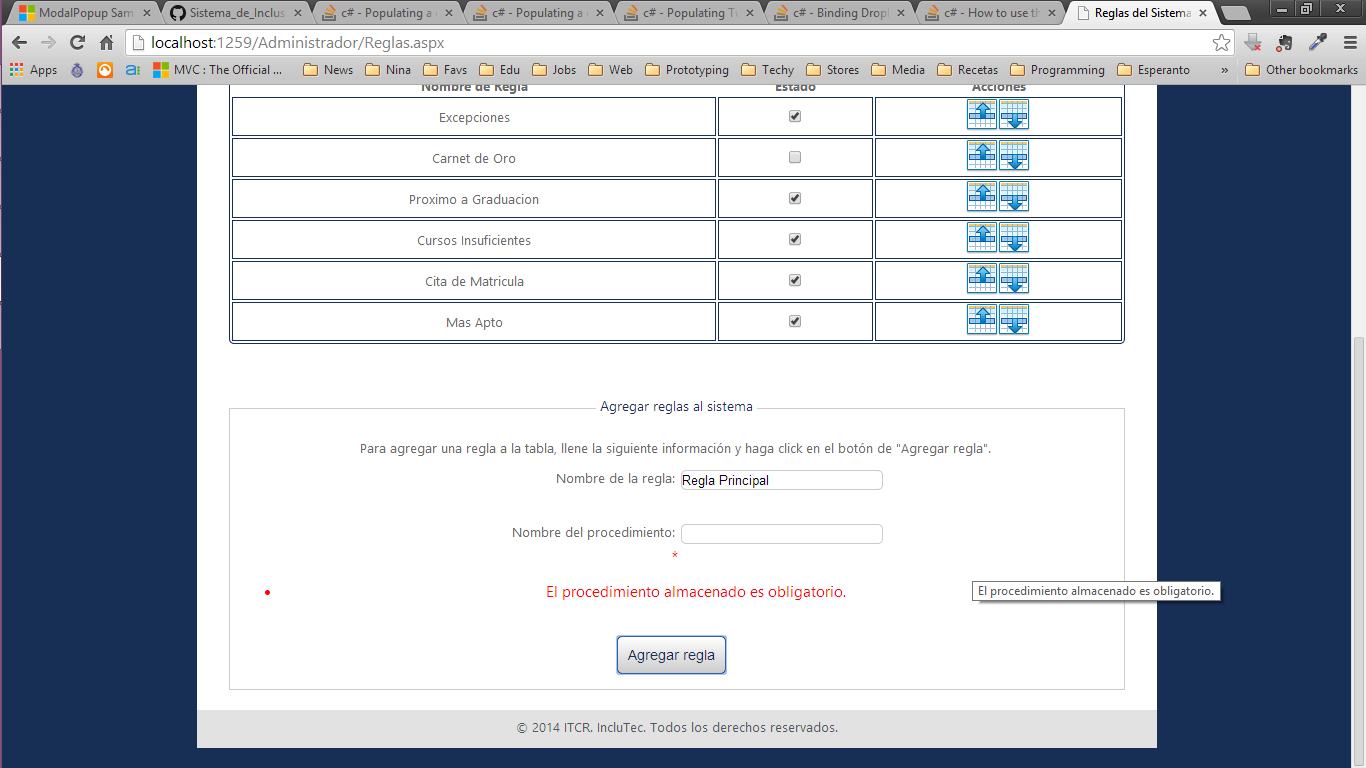
###### Opción de Reglas de Asignación

En esta opción el coordinador puede gestionar las reglas con las cuales el sistema realizará la asignación automática. La pantalla en la que se ubica dicha opción es la mostrada a continuación:



Acá se pueden realizar las siguientes operaciones:

* Habilitar/deshabilitar la aplicación de una regla, dicha acción se realiza mediante la selección del checkbox ubicado a la derecha de cada una de las reglas.
* Ordenar la ejecución de las reglas, mediante el uso de los botones de arriba y abajo ubicados a la derecha de cada regla.
* Agregar nuevas regla, mediante el ingreso de datos en el formulario ubicado en la parte inferior de la página, como el mostrado a continuación:



###### Opción de Excepciones

En esta opción se muestran aquellos casos especiales que requieren una mayor prioridad en el proceso de asignación de cupos. La pantalla que se muestra es la siguiente:



Acá se pueden realizar las siguientes operaciones:

* Eliminar la excepción de proceso, mediante el botón de eliminar ubicado al lado derecho de la excepción.
* Agregar una nueva excepción, mediante el ingreso de datos en el formulario ubicado en la parte inferior de la página.

##### Área de Herramientas

###### Opción de Período de Recepción

En esta opción se permite al coordinador establecer las fechas durante las cuales se permite la recepción de solicitudes. La pantalla que se muestra es la siguiente:



La única operación a realizar en ésta página es la de definir las fechas de un período.

###### Opción de Asignación Automática de Cupos

En esta opción el coordinador procede a ejecutar el sistema que se encarga de asignar los cupos por inclusión, con base en las reglas y excepciones dadas. La pantalla que se muestra es la siguiente:



###### Opción de Notificaciones

En esta opción el coordinador procede a enviar notificaciones vía correo electrónico a los estudiantes y profesores acerca de los resultados del proceso de inclusiones. La pantalla que se muestra es la siguiente:



##### Área de Consultas

###### Opción de Consulta de Período

En esta opción el coordinador puede consultar los períodos definidos en el sistema, así como las fechas que éstos comprenden. La pantalla a mostrar es la siguiente:



###### Opción de Consulta de Solicitudes

En esta opción el coordinador puede consultar los resultados del proceso de inclusiones, además de tener la posibilidad de imprimirlos para su publicación en la Escuela. La pantalla a mostrar es similar a la mostrada en la Consulta de Períodos.

###### Opción de Consulta de Sugerencias

En esta opción el coordinador puede consultar la cantidad de solicitudes rechazadas por curso, así como un informe de sugerencias en cuanto a la acción de contingencia a realizar, ya sea la de buscar la apertura de un nuevo grupo, o la de ampliar el espacio en alguno de los ya existentes; además de poder imprimir dicho informe para su gestión posterior. La pantalla a mostrar es similar a la mostrada en la Consulta de Períodos.

### Plataforma Móvil

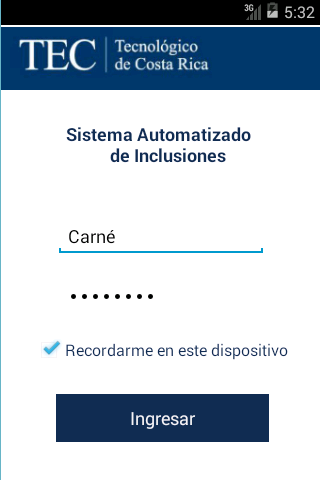
#### Instalación

En lo que respecta a la aplicación Móvil, se requiere descargarla desde la página de la Escuela de Computación. No requiere de la instalación de ninguna extensión adicional.

#### Sección de Autenticación

Para ingresar al sistema, es necesario ser estudiante de la institución, por lo que al ingresar a la dirección del sistema se desplegará una pantalla para la autenticación de los mismos.

Esta sección es obligatoria para todos los usuarios del sistema y permite o rechaza el acceso a las demás secciones del sistema.

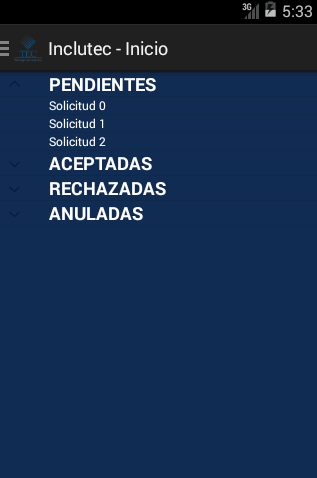


Para el registro de autenticación se debe ingresar:

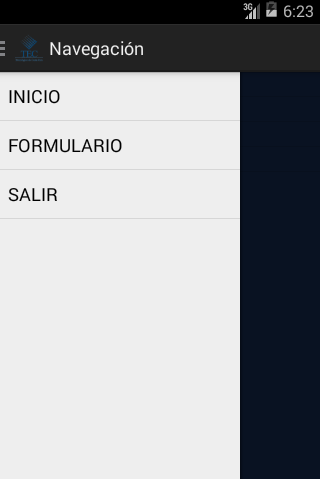
* El Número de carnet estudiantil.
* La Contraseña, que es el PIN de estudiante brindado por Admisión y Registro.

#### Pantalla de Inicio del Sistema

El estudiante, al ingresar al sistema, procede a observar la siguiente pantalla, mediante la cual se pueden revisar las solicitudes divididas en 4 estados: Solicitudes Pendientes, Solicitudes Anuladas, Solicitudes Aprobadas y Solicitudes Rechazadas.

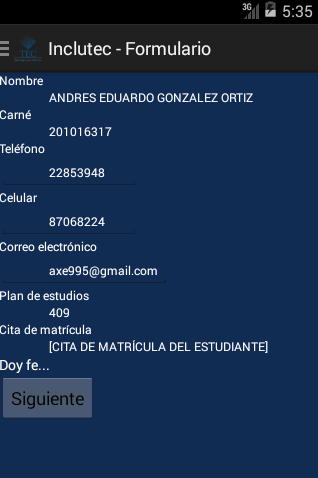


Dentro de la sección, se puede acceder a un Menú de Navegación que permite el acceso al área de Solicitudes; así como a la salida de la aplicación.

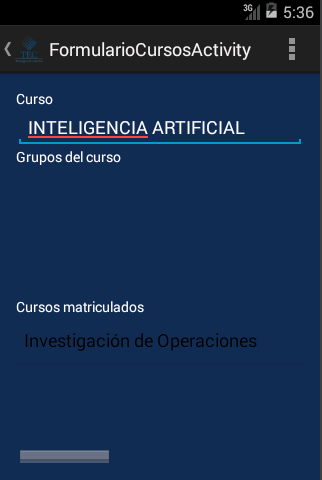


#### Pantalla de Formulario de Solicitud de Inclusión

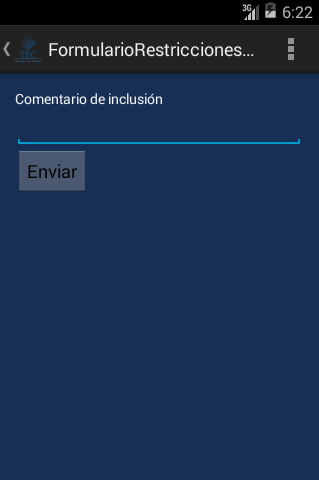
Al acceder a la opción Formulario desde el Navegador del sistema, se ingresa a la pantalla de formulario, en la cual se pueden editar los campos de números telefónicos, o correo electrónico, la cual tiene un aspecto como el siguiente:



Al seleccionar el botón siguiente, se procede a la pantalla de selección de cursos, que es como la mostrada a continuación:



Al seleccionar el curso y grupos de interés, se procede a seleccionar siguiente y se muestra la pantalla de comentarios, tal y como la siguiente:



Al seleccionar el botón enviar, se procesará la solicitud y se retornará a la pantalla de inicio del sistema.

#### Pantalla de Vista/Edición de Solicitudes

En la pantalla de inicio, al seleccionar una de las solicitudes presentadas, se desplegará una pantalla como la siguiente:



Dependiendo del estado de la solicitud, el estudiante será capaz de, mediante el arrastre del grupo, seleccionar su prioridad de selección en la solicitud.

## Funcionamiento

### Ambiente Operativo

El sistema está compuesto por una aplicación instalada en un servidor de la Escuela de Ingeniería en Computación. Este sistema utiliza una Base de Datos sobre el motor SQL Server 2008 R2, la cual está en un servidor distinto a donde está instalada la aplicación. Estos servidores, tanto donde se ubica la aplicación como donde se ubica la Base de Datos, utiliza un sistema operativo Windows Server 2008 R2.

Para conseguir la información necesaria de los estudiantes y profesores, así como la información de matrícula (cursos, grupos, horarios, citas de matrícula) se utiliza la interfaz de Web Services del Departamento de Admisión y Registro (wsDAR), la cual se encuentra a disposición de los servidores vía IP institucional desde el Centro de Cómputo del TEC.

El sistema será accedido vía internet por los estudiantes de Ingeniería en Computación interesados en la realización de una inclusión en un curso en particular, estas solicitudes pueden ser revisadas una vez emitidas para verificar su estado. Ambas operaciones se pueden realizar mediante plataformas web o mediante la aplicación para dispositivos móviles Android. El sistema deberá ser intuitivo y permitir una mayor agilidad en el proceso de inclusiones.

### Rendimiento

El rendimiento de la aplicación está determinado por la cantidad de usuarios que soporte los equipos donde se hará el despliegue de la aplicación, así también por las características de hardware que este tenga.

La eficiencia de la aplicación está determinada por los requerimientos del usuario y al ser principalmente implementado el sistema como una solución Web, la manera de comunicación entre el cliente (dado un por Web browser) y el servidor Web, está establecida por el protocolo HTTP.

El funcionamiento de dicho protocolo es mediante dos acciones Request (por parte del cliente) y Response (por parte del servidor), por lo cual se requiere que la comunicación entre ambas partes establecida por el protocolo no sea mayor a 2 minutos para respuesta de los casos de uso planteados anteriormente.

### Tamaño

Según los requerimientos del cliente, el sistema podrá recibir en promedio unas 1400 solicitudes de inclusión anuales, por lo que se deduce que análogamente se generarán 1400 entradas aproximadamente en la tabla principal del sistema de datos. Además, se debe tomar en consideración el crecimiento de la población estudiantil que utilizará este sistema en un futuro cercano, debido a la apertura y crecimiento de la carrera de Ingeniería en Computación en las sedes de Alajuela, Centro Académico y Limón, ya que la cantidad de usuarios se vería reflejada directamente en un incremento aún mayor en el promedio anteriormente mencionado.

El crecimiento de las tablas está determinado por el máximo de inclusiones recibidas en un año, según lo estimado por el cliente. Tomando esto como base, podemos estimar el tamaño potencial que puede llegar a alcanzar:

#### Tamaño de las Tablas en Bytes

El detalle de tamaños en bytes de las tablas se presenta a continuación:

|  |  |
| --- | --- |
| Nombre de la Tabla | Tamaño en Bytes |
| SIFCurso | 128 |
| SIFEstudiante | 128 |
| SIFGrupo | 16 |
| SIFGrupo\_Por\_Solicitud | 32 |
| SIFPeriodo | 4 |
| SIFPlanEstudios | 64 |
| SIFProfesor | 128 |
| SIFRegla | 32 |
| SIFSolicitud | 316 |
| SITDatos\_Estudiante | 64 |
| SIT\_Excepcion | 32 |
| SITHorario | 32 |
| Total en Bytes | **976** |
| Crecimiento Anual | ***1366400*** |

## Análisis del Problema

El crecimiento de las plataformas tecnológicas basadas en web y el aumento de estudiantes dentro de la Escuela de Ingeniería en Computación conlleva al desarrollo de un sistema que permita realizar inclusiones de matrícula en uno o más grupos por parte de los estudiantes.

Actualmente, el proceso de inclusiones se realiza de manera manual por lo que se debe revisar una gran cantidad de boletas de inclusión en un tiempo determinado, lo cual hace que esto sea un problema creciente en la escuela. El sistema de inclusiones optimizará y mejorará el proceso para realizar las solicitudes.

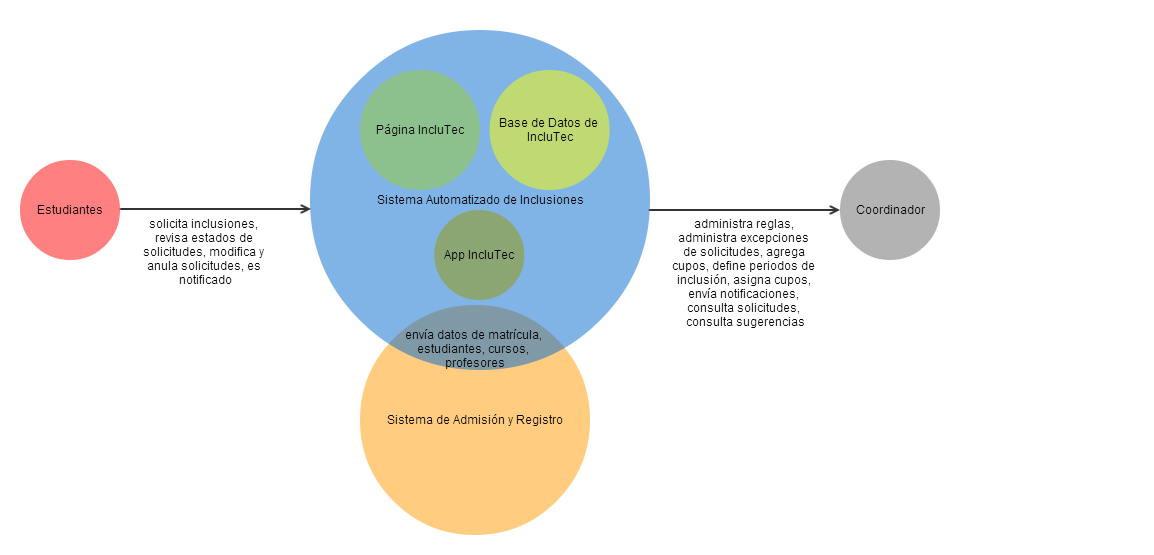
El sistema de inclusiones es una aplicación que permite a los estudiantes solicitar las inclusiones desde cualquier computadora o dispositivo móvil con acceso a internet, además la información del estudiante y de las posibles inclusiones que pueda hacer estará dada por una interfaz de comunicación que proveerá el Departamento de Admisión y Registro.

Además, el coordinador de la carrera o encargado del proceso revisará los resultados del proceso automático de inclusiones desde el mismo sistema mientras cuente con los requisitos básicos de funcionamiento (acceso a internet y un navegador web).

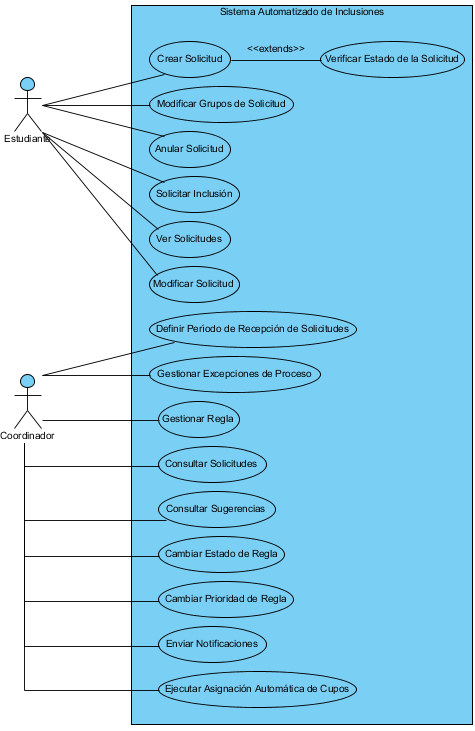
### Contexto del Sistema

El sistema de inclusiones a desarrollar tiene como parte de sus características, que no solo recopilará formularios de solicitud de inclusión, sino que se integrará en un ambiente de software del Tecnológico de Costa Rica, por lo cual debe ser tomada en cuenta toda conexión que realice con otros sistemas de la institución, y la relación que tendrá con dichos programas, así como las entradas y salidas del mismo.

El siguiente es un diagrama que representa las diferentes interacciones que tendrá el sistema de inclusiones en el contexto anteriormente planteado:



A partir del análisis del contexto del sistema y de las expectativas y deseos de los interesados, se inició la recolección de requerimientos. Como producto de dicho proceso, se determinó realizar los casos de uso presentados en el siguiente diagrama:

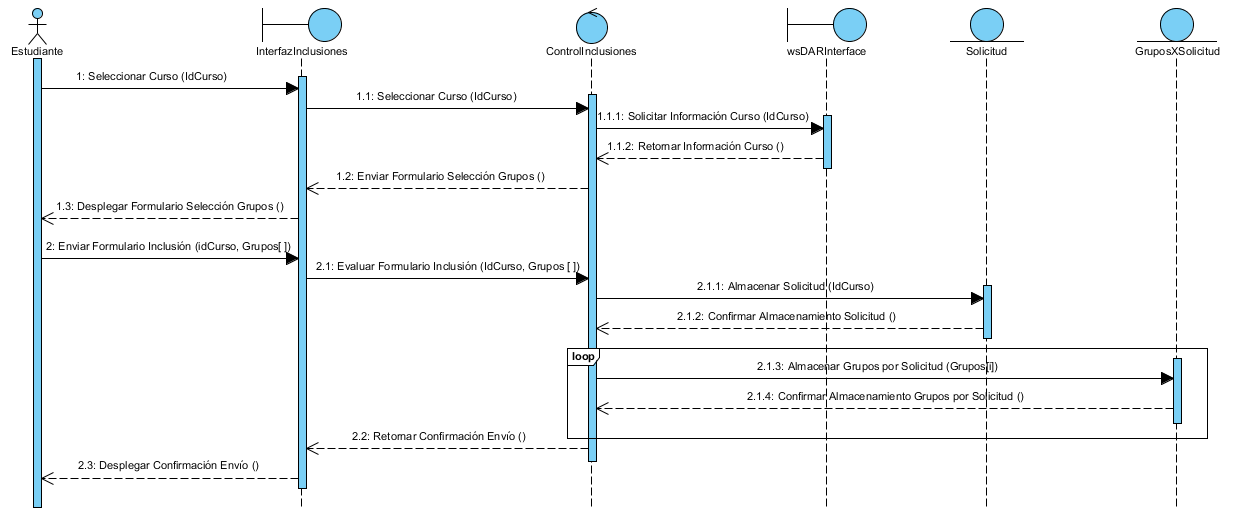


Los textos y diagramas de los casos de uso presentados anteriormente fueron incluidos en el Documento de Especificación de Requerimientos (ERS).

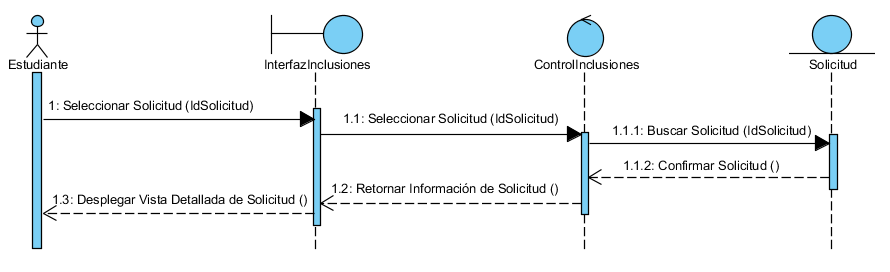
### Diagrama de Secuencia de los Casos de Uso

A continuación se presentan los diagramas de secuencia del sistema, de cada uno de los casos de uso anteriores:

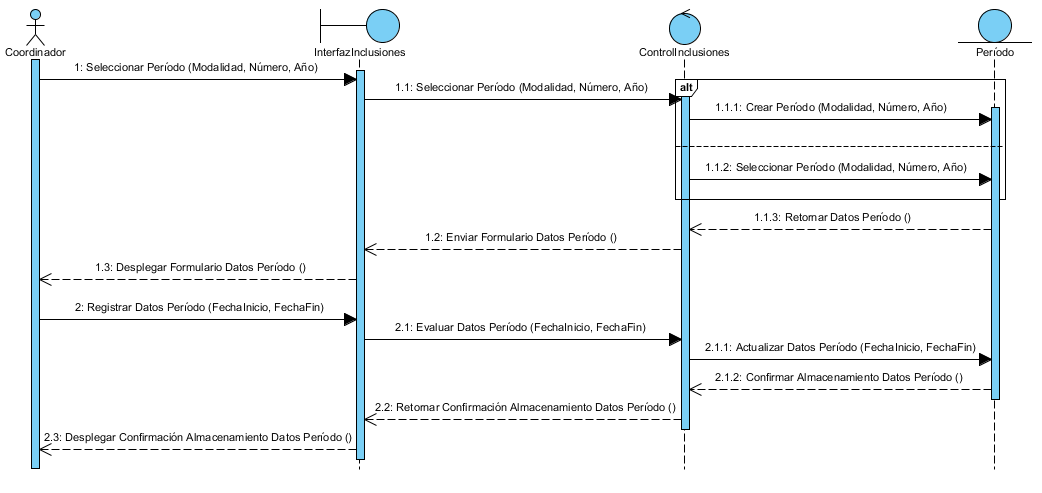
#### CU-01: Crear Solicitud



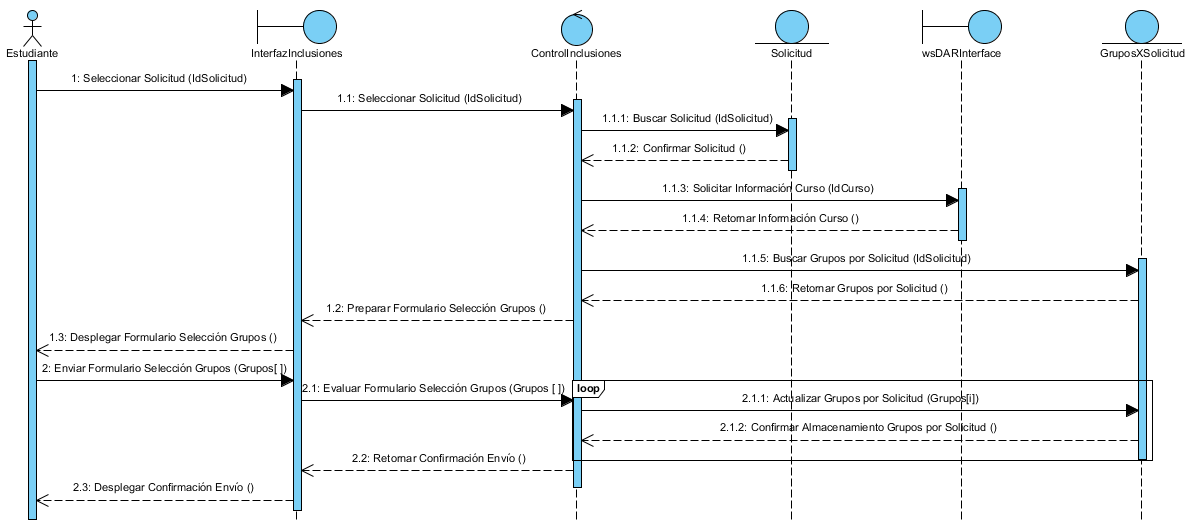
#### CU-02: Revisar Estado de Solicitudes



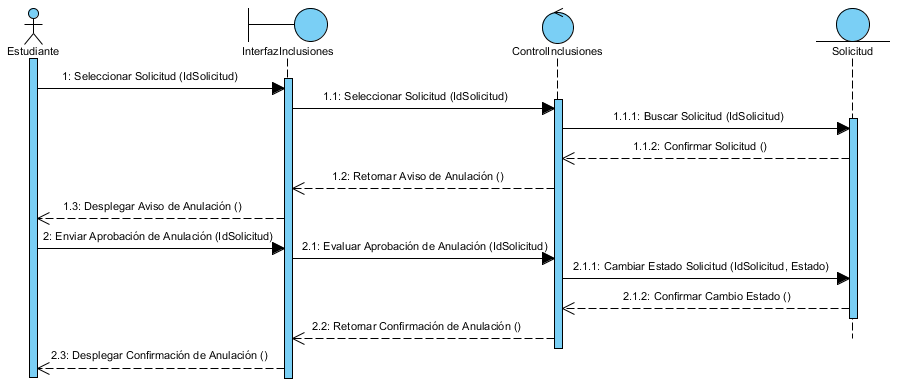
#### CU-03: Definir Período de Recepción de Solicitudes



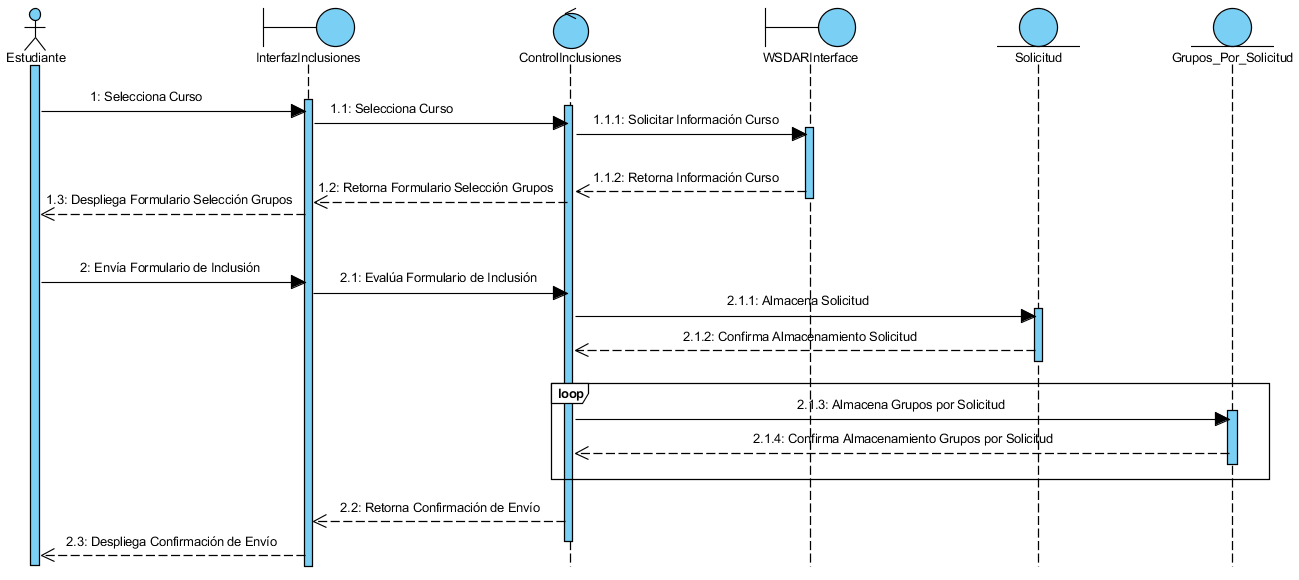
#### CU-04: Modificar Solicitud de Grupo



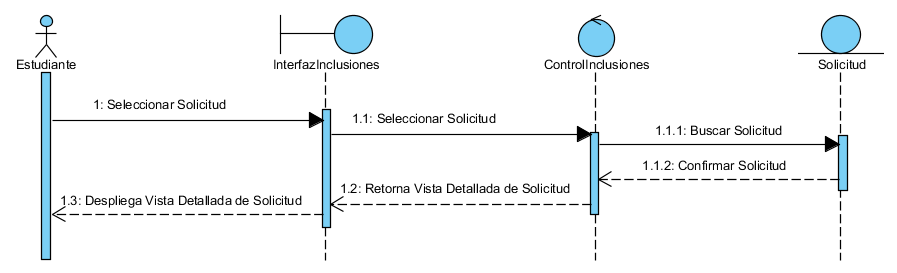
#### CU-05: Anular Solicitud



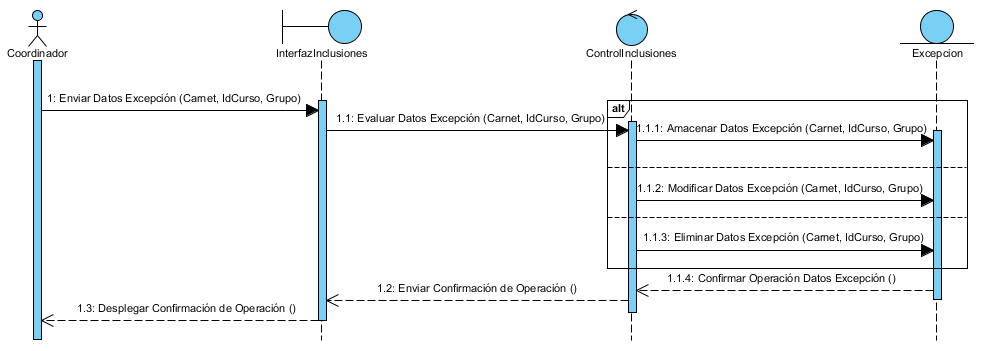
#### CU-06: Solicitar Inclusión



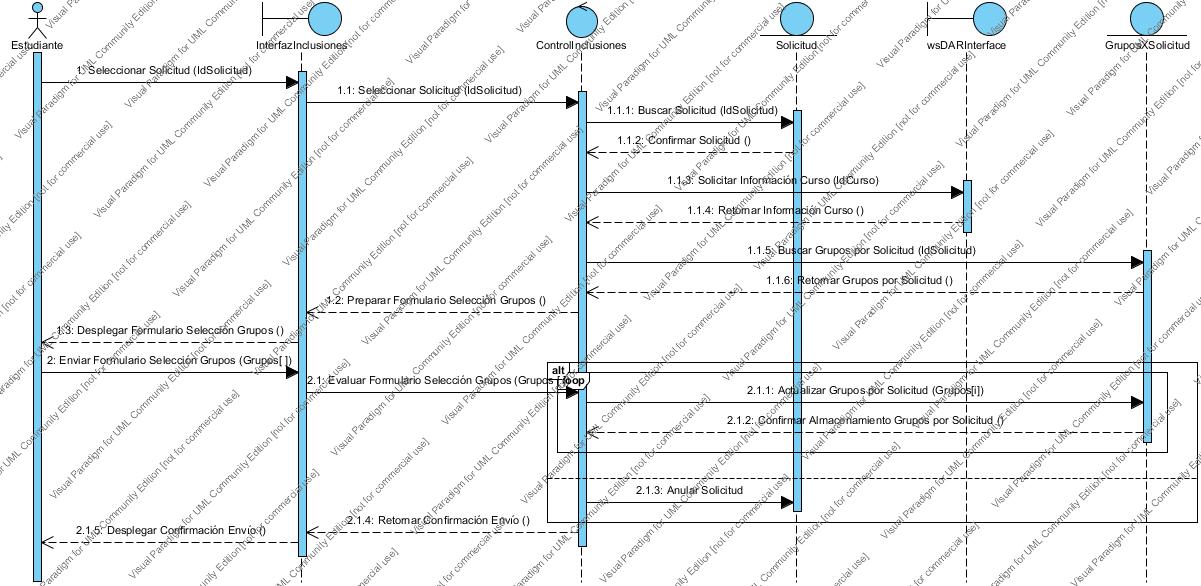
#### CU-07: Ver Solicitudes



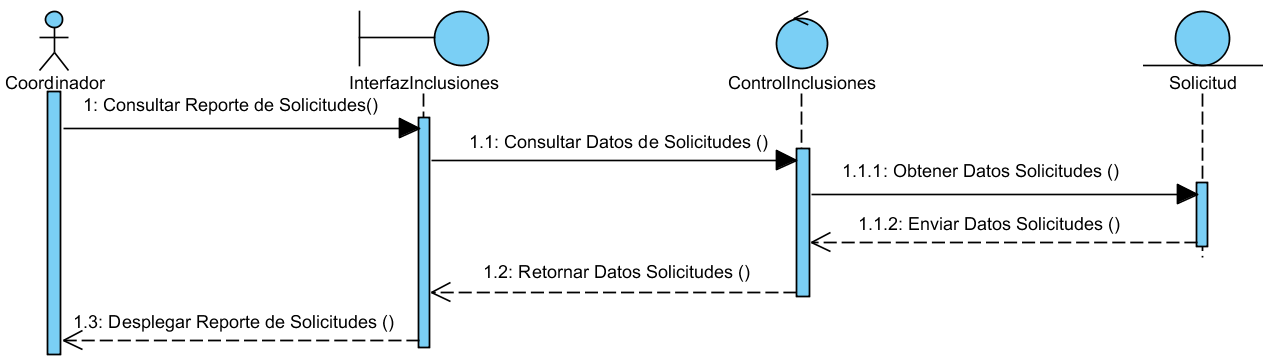
#### CU-08: Gestionar Excepciones de Proceso



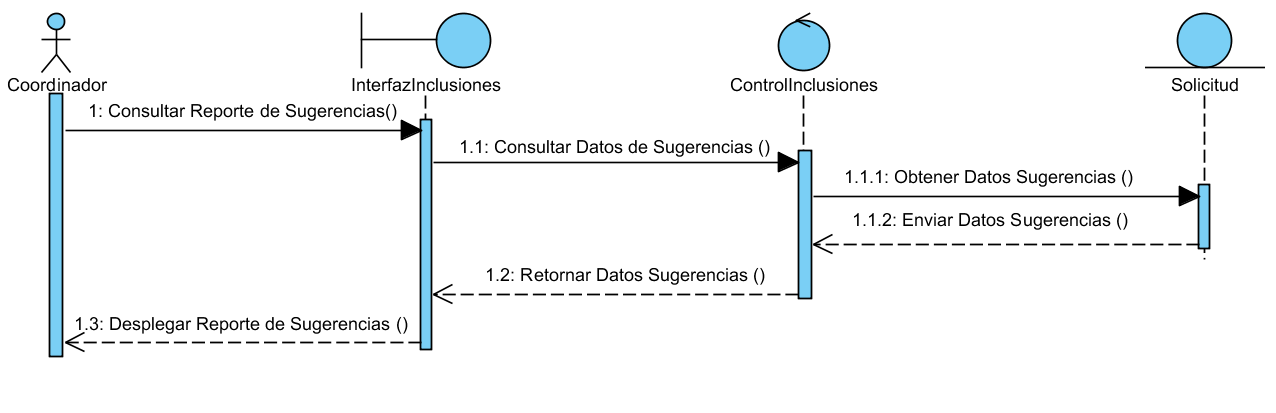
#### CU-09: Modificar Solicitud



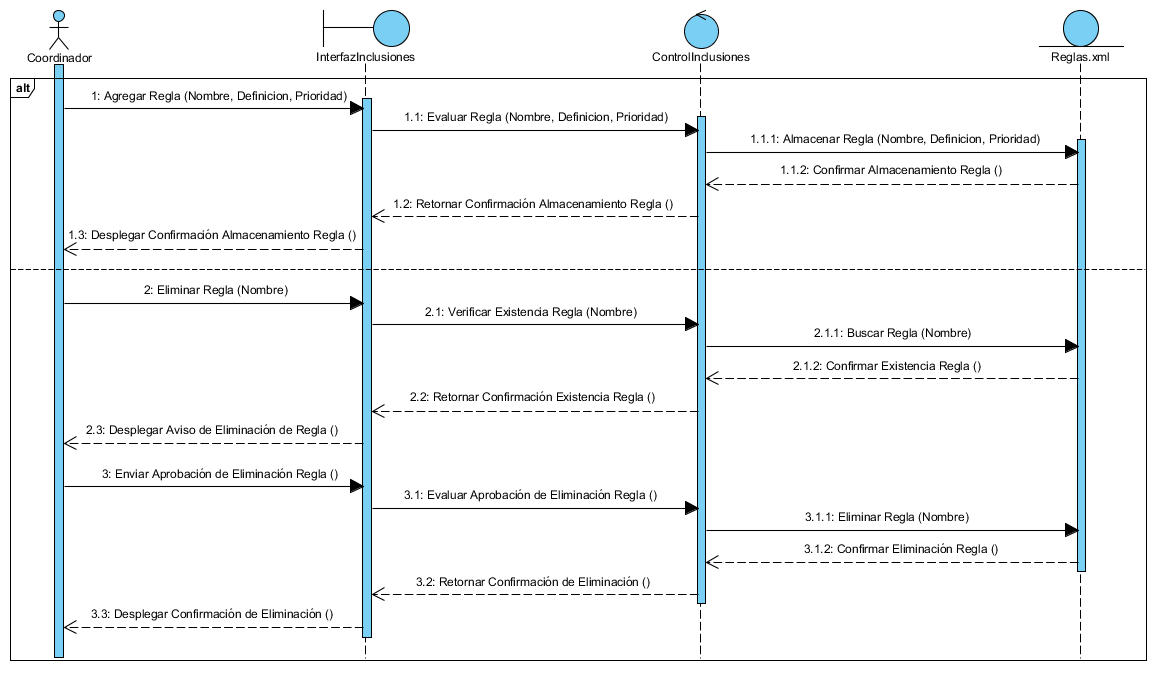
#### CU-10: Consultar Solicitudes



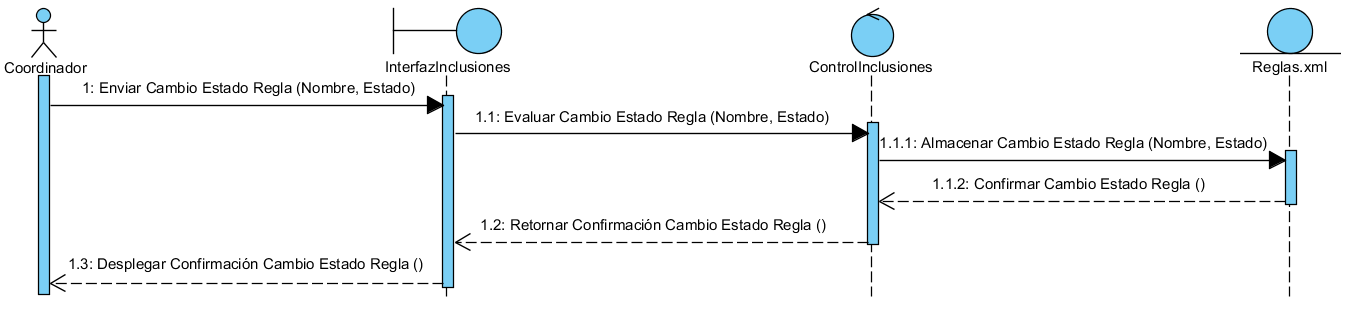
#### CU-11: Consultar Sugerencias



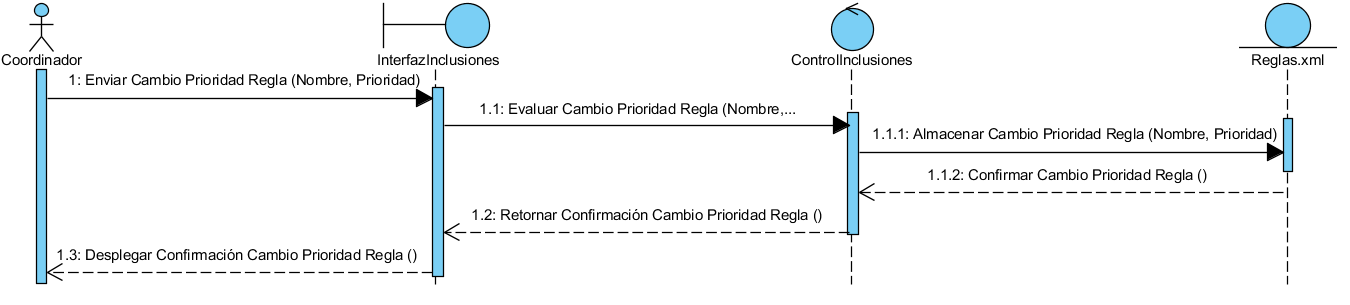
#### CU-12: Gestionar Regla



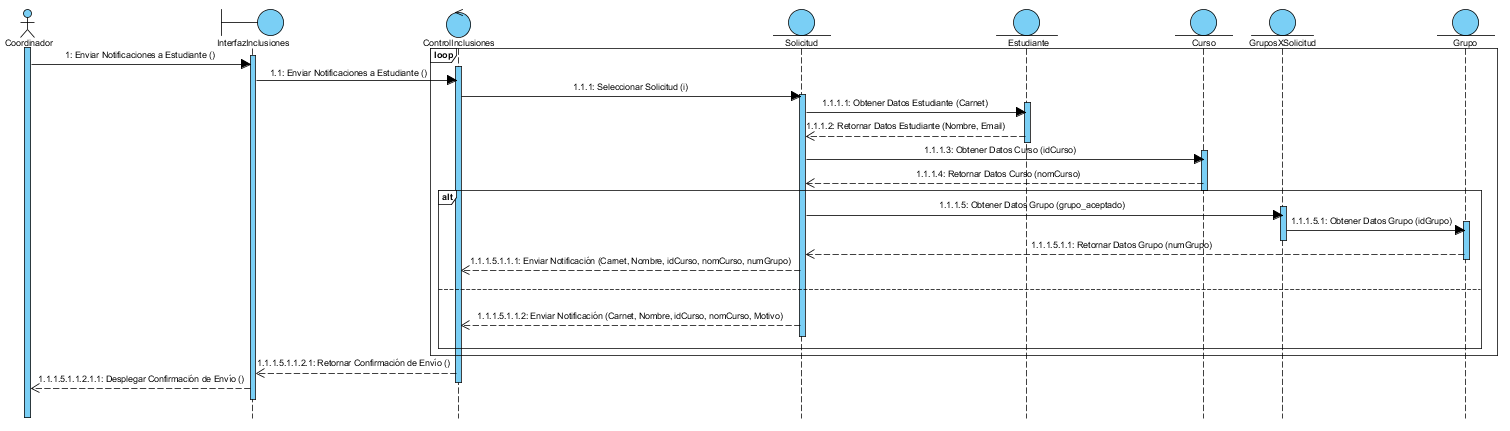
#### CU-13: Cambiar Estado de Regla



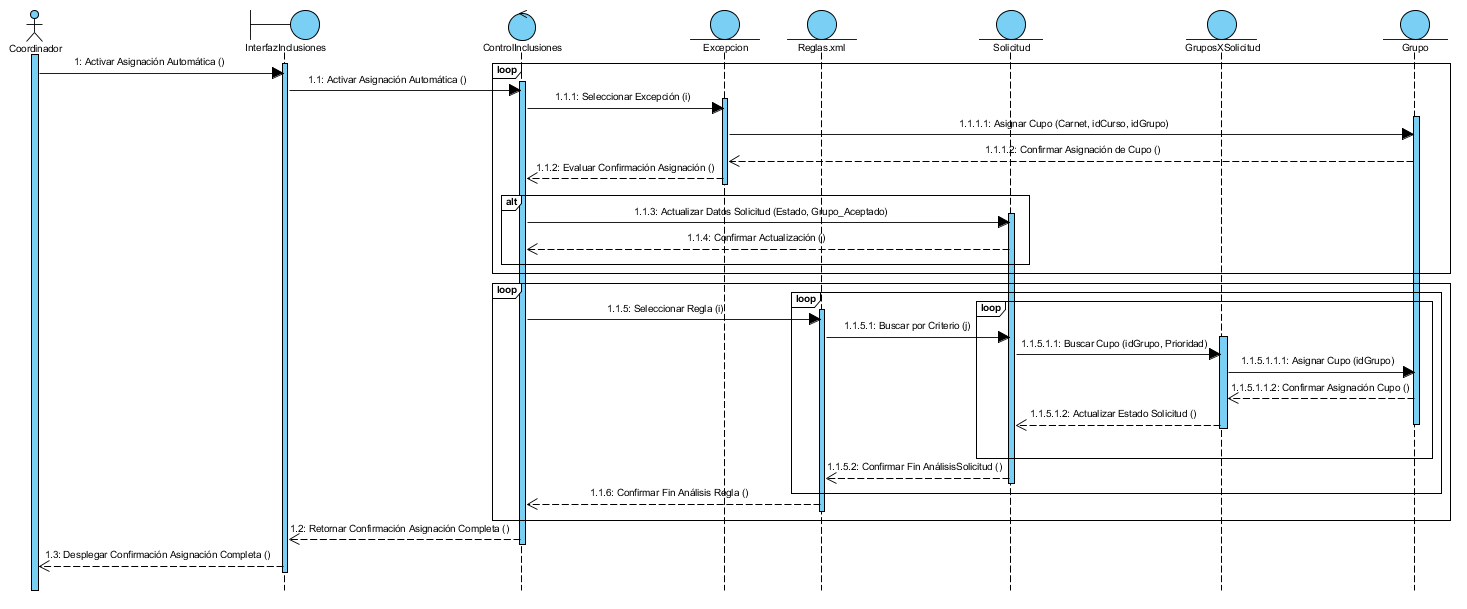
#### CU-14: Cambiar Prioridad de Regla



#### CU-15: Enviar Notificaciones

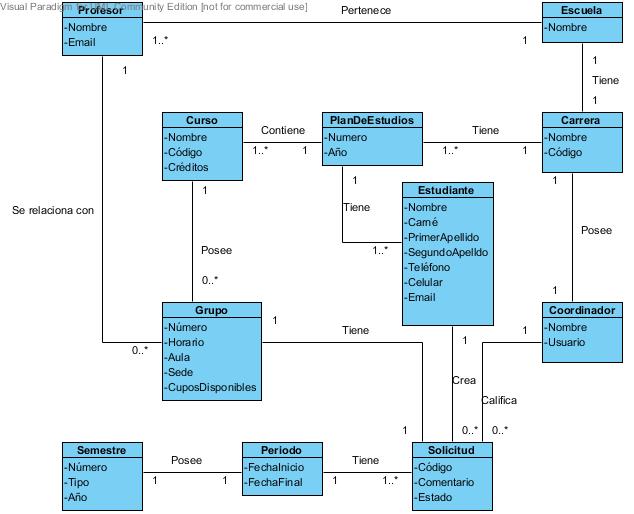


#### CU-16: Ejecutar Asignación Automática de Cupos



### Diagrama de Modelo de Dominio

Con base en las funcionalidades que debe tener el sistema, se construyó el siguiente Modelo de Dominio:



# Diseño

## Visión General

Una de las decisiones tomadas en la arquitectura del programa fue la realización del mismo en capas. Dicha decisión desencadenó en la división del proyecto en tres capas principales:

* Capa de Presentación.
* Capa de Negocios.
* Capa de Acceso a Datos.

Además de ésta división tenemos el módulo del estudiante y el módulo del coordinador.

En este proyecto debido a estándares institucionales se decidió realizar el sistema con ASP.NET y una Base de Datos en SQL Server R2, los cuales realizan una conexión entre ellos mediante el Enterprise Library. Esto para realizar una conexión efectiva y segura entre ambos mediante la utilización de los datos de conexión provistos por el .config en el sistema.

Otro punto clave del diseño a partir de la arquitectura utilizada es que al utilizar el modelo en capas y dividir su funcionalidad en módulos permite la adición de funcionalidades según se necesite sin afectar directamente a otras funcionalidades del sistema.

## Estructura en Tiempo de Ejecución

El sistema se encuentra dividido en diferentes ventanas y clases dependiendo de la funcionalidad deseada, cada ventana dirigirá a otra ventana dependiendo de la acción a realizar por parte del usuario, además cada ventana de la capa de presentación posee una relación a una o más clases de la capa de negocios dependiendo del módulo en el cual se encuentre y la acción a realizar. Además las clases de lógica se encuentran relacionadas directamente con una clase en la capa de acceso a datos, la cual se encarga de todo lo relacionado con la conexión a la base de datos y al web service.

Como se mencionó anteriormente, el sistema se encuentra dividido en tres capas que a su vez se encuentran subdivididas. Un diagrama que muestra dichas subdivisiones, así como sus relaciones es el siguiente:

## Estructura del Módulo

Para el desarrollo del sistema se empleó una arquitectura de tres capas, lo cual facilita el entendimiento del código fuente, además de una mayor modularidad en el sistema por lo cual facilita la extensión del sistema en caso que se desee ampliar en un futuro por medio de la agregación de otros módulos, o modificar el funcionamiento de un módulo, sin afectar el buen funcionamiento de otras partes del sistema. Las tres capas utilizadas son las siguientes:

* Capa de Presentación: Esta es la encargada de comunicar a la aplicación con el usuario, así también como es el encargado de iniciar la ejecución de las acciones que el usuario desea realizar. En esta se encuentra la interfaz del sistema.
* Capa de Negocio: Esta contiene las clases que se encargan de la comunicación entre la capa de presentación y la de datos. Además esta es la encargada de la lógica de la aplicación.
* Capa de Datos: Esta es la encargada de la comunicación con la Base de Datos, así también como la comunicación con el Web Service para la lectura o escritura de datos en la misma.

# Pruebas

## Estrategia

El sistema de inclusiones es un sistema orientado al manejo de datos de las inclusiones y el manejo de la información de los estudiantes para la lógica de los cursos posibles a inclusión por parte del estudiantado, por lo cual se realizó mayor énfasis en las pruebas de rendimiento, donde el tiempo de respuesta de las peticiones

# Reflexión

## Evaluación

## Lecciones

El grupo ha aprendido mucho acerca de cómo deben ser las aplicaciones en el campo laboral. Es importante tener toda la retroalimentación que el cliente pueda dar pues ellos son los que al final van a utilizar el software, por lo que además del grupo de trabajo, el cliente tiene que estar comprometido completamente con el desarrollo de este, pues sino el proyecto puede fracasar.

Además se debe evitar depender de otros sistemas a menos que sea completamente necesario, pues si deja de funcionar alguno de ellos el sistema también puede llegar a verse comprometido en cuanto a disponibilidad. En caso de depender de los servidores, tener un servidor local en el cual se puedan probar las instalaciones para detectar posibles errores en el proceso mismo, así como corregirlos antes de realizar la instalación en los servidores de la Escuela.

Otra lección aprendida es llevar un buen control del tiempo utilizado para cada una de las actividades que se realizan en un proyecto, de esta manera se evita apuros al último minuto de entrega.

## Errores y Limitaciones Conocidos

En cuanto a errores y limitaciones se refiere, podemos afirmar que uno de los mayores problemas encontrados fue la gran dependencia de sistemas externos como los Web Services administrados por el Centro de Cómputo, así como el acceso restringido a través de las direcciones IP de la Escuela, ya que su disponibilidad no está garantizada al 100% y, por ende, conlleva a posibles fallos del sistema provocados por ellos. Además con respecto a este tema, se puede presentar el caso de una modificación a los Web Services o a la Base de Datos de Admisión y Registro que podría perjudicar gravemente el funcionamiento de este software.

Al haber tenido muy pocas pruebas debido a riesgos materializados a lo largo de la creación del producto, pueden surgir errores no contemplados en este documento que serían corregidos en una etapa posterior.

# Apéndice

## Formatos

### Estándar de Programación Código Fuente

#### Nombres de Clases

Los nombres de las clases deben tener el siguiente formato:

**‘c’ + Nombre descriptivo + Capa**

O bien, para las clases de negocios que requieren nombres sencillos:

**‘c’ + Nombre descriptivo**

Nombre descriptivo hace referencia al nombre de la clase y su función (la primera letra del nombre debe ser en mayúscula).

Capa se refiere a la capa en la cual se esta trabajando (Base, Negocios, Datos)

Ejemplo:

* cManejoErroresDatos
* cSeguridadITCRDatos
* cSolicitudNegocios

Se usan sustantivos en singular para los nombres de las clases.

Los nombres de excepciones personalizadas deben terminar con el postfijo “Exception”.

Namespaces or Packages deben seguir el mismo nombre que los folders que contienen el código fuente y en minúscula.

#### Métodos

* Los nombres de métodos deben iniciar con mayúscula y el resto de su nombre debe ir en minúscula, siguiendo el estándar de C#. Ej. Insertar, RegistrarEmpresa.
* Para un método que es un procedimiento utilice un verbo seguido de un objeto, por ejemplo SaveCustomer, MoverFicha, InsertarNodo.
* Para el nombre de una función use una descripción del valor de retorno de la función, por ejemplo NextCustomerId, GetName, RetornarListaAutos.
* Utilice una propiedad en lugar de métodos cuando lo que se va a retornar es un dato de la clase. En el caso de Java use los métodos llamados “setters” y “getters”

#### Variables

Los nombres de variables de clase privadas tendrán el siguiente formato:

**‘\_’(underscore) + indicador de tipo + Nombre significativo**

* Nombre significativo corresponde a la función específica que se realizara con esa variable y debe iniciar con minúscula. Ejemplo: \_iCodFuncionalidad, \_sNumeroAula.
* No utilice el término “temp” o “index” para nombres de variables, tampoco los nombres de índices i,j,k ni cualquier uso de nombres de una letra o silábicos, en su lugar utilice nombres largos bien descriptivos.
* No utilice nombres ambiguos en variables booleanas, ejemplo: done, status.
* No utilice nombres negativos en las variables como NotFound, NoListado.
* Nunca declare variables de instancia públicas, todas deben ser protected o private.
* Variables locales se escriben con la primer letra en minúscula indicando su tipo sin el “\_” (underscore).

#### Constantes

* MAYUSCULAS\_SEPARADAS\_POR\_UNDERSCORE
* Siempre utilice constantes en lugar de literales hard-coded.
* Hay ciertos casos en los que no es necesario declarar una constante para ciertas literales hardcode, tales se muestran en la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Nombre | Descripción | Ejemplo |
| Cero | Use el valor cero cuando inicializa índices, variables o requiere comparar el largo o cantidad de elementos de una colección | * Variable = 0; * for (int IndeXYZ=0; …. * AnyKindCollection.Length > 0 * AnyKindCollection.Length < 0 * AnyKindCollection.Length == 0 * AnyKindCollection.Length >= 0 * AnyKindCollection.Length =< 0   Lo anterior aplica para arrays, arraylist, collections, hashtable, datarows, etc cuando usan las propiedades o métodos Length, Size, Count |
| * Length-1 * Size-1 | Algunos Collections requieren algún ajuste de tamaño para hacer comparaciones o para ajustarse a algún algoritmo | for(int indexXYZ=0;indexXYZ<array.Length-1… |
| Inicio de índice  1,2,3 | Cuando un ciclo inicia en los valores 1, 2, 3 | for (int indexXYZ=1; indexXYZ<array.length …. |
| Evaluación par o impar | Cuando se requiere saber si un número es par o impar | Cuando se requiere saber si un numero es par o impar |

#### Properties

* Deben iniciar con minúscula indicado su tipo y seguir la el estándar de C#. Ej. sNombrePersona, iTipoSolicitud
* Si el lenguaje no soporta properties utilice los métodos de Set y Get que sean necesarios.

#### Comentarios

Los procedimientos que se utilicen en la clase deben ser comentados, así como cualquier elemento que así lo requiera utilizando el estándar de documentación XML de .NET.

Ejemplo:

/// <summary>

/// Propósito: Inicializa los miembros de la clase.

/// </summary>

private void InitClass()

{

// Crea todos los objetos e inicializa otros miembros.

\_conexionBD = new OleDbConnection();

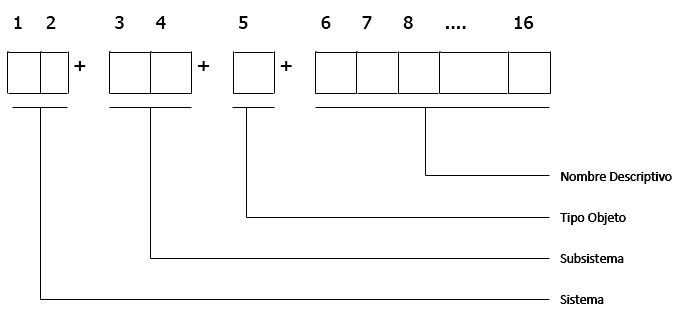
....

}

### Estándar de Programación Base de Datos

A continuación se describen las normas generales que se deben utilizar para definir los nombres de los objetos incluidos en la base de datos de un sistema, esto incluye nombres de tablas y vistas.

Para asignar el nombre a objetos de este tipo se debe utilizar letras mayúsculas de acuerdo con el siguiente esquema:



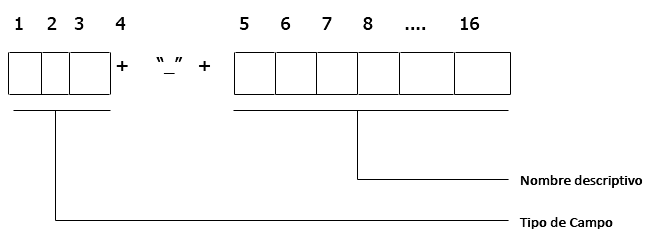
* Sistema: Corresponde a una tira de dos caracteres que identifica el sistema al que pertenece el objeto.
* Subsistema: Corresponde a una tira de dos caracteres que identifica el subsistema al que pertenece el objeto.
* Tipo Objeto: Si es una tabla puede ser “F” (datos fijos) o “T” (temporal). Si es otro tipo de objeto se utiliza lo siguiente: “V” para vistas.
* Nombre descriptivo: es una tira de caracteres que permite identificar la naturaleza del objeto.

Ejemplos:

* SGTPFSOLICITUDES (Sistemas de Servicios Generales, Sistema del Taller de Publicaciones, tabla fija de solicitudes).
* IEPEFDEPTOS (Información Estudiantil, Planes de Estudio, tabla fija de departamentos).

#### Nombres de los campos

Para asignar el nombre a los campos en cada tabla de la base de datos, se debe utilizar el siguiente esquema:



* Tipo de Campo: corresponde a una tira de caracteres que indica el uso del campo de acuerdo a la siguiente tabla:

|  |  |  |
| --- | --- | --- |
| Código | Tipo de Campo | Ejemplo |
| Id, Ide | Identificador | ID\_USUARIO |
| Can | Cantidad | CAN\_PAGINAS |
| Cod | Código | COD\_DEPTO |
| Dir | Dirección | DIR\_HABITACION |
| Dsc | Descripción | DSC\_SOLICITUD |
| Fec | Fecha | FEC\_INGRESO |
| Img | Imagen | IMG\_EMPLEADO |
| Int | Interés | INT\_ANUAL |
| Mon | Monto | MON\_SALDO |
| Nom | Nombre | NOM\_EMPLEADO |
| Num | Número | NUM\_CEDULA |
| Por | Porcentaje | POR\_NOTA |
| Txt | Texto | TXT\_EMAIL |

* Nombre descriptivo: es una tira de caracteres que permite identificar la naturaleza del campo.

#### Nombres de procedimientos almacenados

Los procedimientos deben nombrarse iniciando con “pr\_” seguido del nombre descriptivo del procedimiento.

Ejemplo:

* pr\_ConsultaSalario
* pr\_ObtenerDepartamentoEmpleado
* pr\_ReporteAnual

Si el procedimiento es de mantenimiento, tal como insertar, actualizar, eliminar, etc. para una tabla, el nombre del procedimiento debe ser “pr\_” seguido del nombre de la tabla, un underscore y el nombre de la función que realiza, las cuales son:

* Insertar
* Eliminar
* Actualizar
* Buscar
* SeleccionarUno
* SeleccionarTodos

Ejemplos:

* pr\_SGTPFSOLICITUD\_Insertar
* pr\_SGTPFCENTROCOSTO\_SeleccionarTodos

### Estándar de programación de la base de datos (SQL)

#### Estándar para procedimientos almacenados

* Las palabras reservadas deben escribirse en mayúscula.
* Los nombres de tablas se rigen por los estándares de diseño establecidos.
* Los nombres de procedimientos almacenados definidos por el usuario deben seguir el siguiente formato:

**‘pr’ + \_ (underscore) + objeto sobre el cual trabaja + nombre de la función que realiza**

Ejemplo:

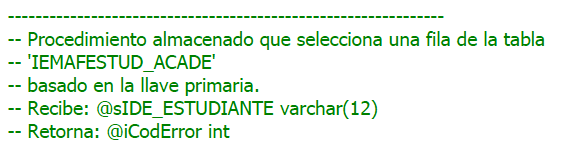
* pr\_IEMAFESTUD\_ACADE\_Insertar

En caso de que el procedimiento actúe sobre varios objetos de la base de datos al mismo tiempo, entonces puede omitirse el nombre del objeto.

Ejemplo:

* pr\_GenerarIndice
* Cada procedimiento almacenado debe contener como comentario un encabezado con la siguiente estructura:
* Nombre del procedimiento
* Función que realiza
* Fecha y hora de creación
* Precondiciones
* Postcondiciones (debe incluir los valores de retorno y su descripción)
* Lista de parámetros y su descripción

Ejemplo:



* Declaración de parámetros: debe declararse un parámetro por línea acompañados por su respectivo tipo. Ejemplo:

CREATE PROCEDURE dbo.pr\_IEMAFESTUD\_ACADE\_Insertar

@siCOD\_CLA\_ACC smallint,

@sNOM\_ESTUDIANTE varchar(40),

@sIDE\_SEDE varchar(2)

AS

INSERT dbo.IEMAFESTUD\_ACADE

(

COD\_CLA\_ACC,

NOM\_ESTUDIANTE,

IDE\_SEDE

)

VALUES

(

@siCOD\_CLA\_ACC,

@sNOM\_ESTUDIANTE,

@sIDE\_SEDE

)

* Para cualquier mandato SQL, la cláusula inicial funcionará como encabezado y cualquier otra cláusula del mandato tendrá una identación de 3 caracteres con respecto a la cláusula inicial. Ejemplo:

SELECT <Lista Campos>

FROM <Lista Tablas>

WHERE <Condiciones\_de\_Seleccion>

## Especificaciones del Módulo

### Capa de Presentación

La capa de presentación posee únicamente la interfaz del sistema y en estas las funciones son llamadas desde las clases de la capa lógica. Las ventanas deben tener su nombre iniciando con frm. Ejemplo: frmPrueba.

### Capa de Negocio

Las clases de esta capa deben iniciar su nombre con una c y terminar con la palabra Negocios. Ejemplo: cPruebaNegocios. Las clases en esta capa poseen los métodos de lógica de la aplicación, así como también el llamado a los métodos necesarios de las clases de la capa de datos, y a los métodos del web service necesario para llevar a cabo las funciones del sistema. Todos los métodos escritos deben de ser documentados presentando los datos de entrada y salida.

### Capa de Datos

Las clases de esta capa deben presentar su nombre iniciando con una c y terminando con la palabra datos. Ejemplo: cPruebaDatos. Estas clases son las que realizan la conexión con la Base de Datos, cada método escrito debe ser documentado presentando los datos de entrada y salida. Posee además una clase llamada cAccesoDatos, la cual provee métodos para la conexión con la BD, así como métodos para la ejecución de Procedures sobre la misma.

## Casos de Prueba