03/01/2014

Especificación de Requerimientos del Sistema

Versión 2.0r

Preparado por: Juan José Rojas Valverde

Iteración II

SISTEMA AUTOMATIZADO DE INCLUSIONES

Tabla de Contenidos

1 Control del Documento

1.1 Historial de cambios

Versión	Fecha	Autor	Cambios realizados
0.0	11/12/2013	Ana Irina Calvo Carvajal	Creación de la plantilla del documento de especificación de requisitos de software
0.1	15/12/2013	Jose Arnoldo Segura Campos	Agregar partes del documento ya especificadas en el documento de Visión.
1.0	17/12/2013	Ana Irina Calvo Carvajal	Agregar interfaz y darle formato
1.1	23/12/2013	Jose Arnoldo Segura Campos	Correcciones al Propósito del documento, a la Visión general del documento, diagramas de actividad y entregas.
1.2	12/23/2013	Ana Irina Calvo Carvajal	Corrección del diagrama de contexto
1.3	23/12/2013	Andrés González Ortiz	Corrección del objetivo general y criterios de éxito, corrección de diagrama de casos de uso
1.4i	02/01/2014	Ana Irina Calvo Carvajal	Modificación de la plantilla para el documento de especificación de la iteración 2, para el caso de uso individual.
1.5r	03/01/2014	Juan José Rojas Valverde	Modificación de la Información para su correspondencia con el caso de uso particular de este documento.
2.0r	03/01/2014	Juan José Rojas Valverde	Correcciones a Diagramas de Actividad y Estado.

1.2 Aprobación del documento

Fecha	Nombre	Título	Firma
03/01/2014	Adriana Álvarez Figueroa	Profesora de la Escuela de Ingeniería en Computación del TEC (cliente)	
03/01/2014	Jaime Solano Soto	Profesor del curso de Proyecto de la Escuela de Ingeniería en Computación del TEC	

2 Introducción

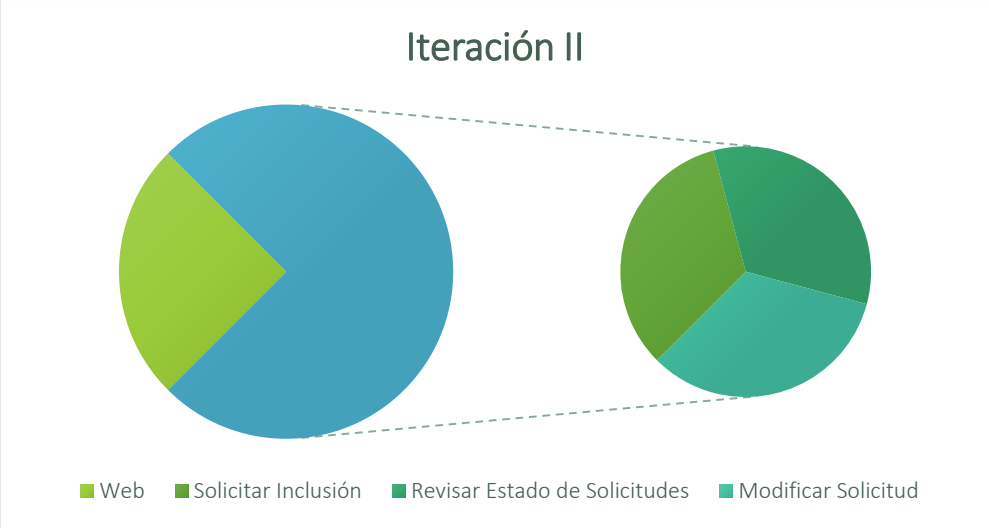
2.1 Propósito del documento

El propósito de este documento es definir y detallar los requerimientos funcionales y no funcionales del software del Sistema Automatizado de Inclusiones de Ingeniería en Computación o Inclusiones Tec, como será llamado en el resto del documento. Esto será realizado mediante la enumeración de los objetivos de la sección de la segunda iteración, los criterios de éxito, los casos de uso involucrados (descritos de forma amplia y detallada), así como otros requerimientos de interfaz, de rendimiento, de flexibilidad, de control de acceso y comerciales.

2.2 Objetivos del sistema

2.2.1 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. Para esta iteración, el trabajo fue dividido en 4 partes, tres para la aplicación Android que se va a generar y una parte para la página web. El siguiente gráfico muestra dichas partes.



La asignación de partes de la segunda iteración en el equipo de trabajo se muestra a continuación:

- Android CU-06 Solicitar Inclusión – Ana Irina Calvo Carvajal
- Android CU-07 Revisar Estado de Solicitudes – Andrés Eduardo González O
- Web CU-08 Gestionar Excepciones de Proceso – Juan José Rojas Valverde
- Android CU-09 Modificar Solicitud – Jose Arnoldo Segura Campos

2.2.2 Objetivo Específico

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.

2.2.3 Criterio de éxito

Para poder comprobar el éxito del objetivo anteriormente planteado, se requiere que el coordinador pueda crear, modificar y eliminar una excepción en particular desde la aplicación Web sin tener ningún problema.

2.3 Perspectiva del producto por desarrollar

El sistema es para la Coordinación de la Escuela de Ingeniería en Computación del Tecnológico de Costa Rica, quienes necesitan agilizar los procesos de solicitud y trámite de inclusión en los cursos semestrales que imparte la escuela. ”Inclusiones Tec” es una aplicación web que Recibe, analiza y califica las solicitudes de inclusiones de forma eficiente y automatizada y que, a diferencia del proceso manual:

- Reduce el tiempo de análisis y calificación de solicitudes de inclusión.
- Notifica automáticamente tanto al estudiante vía correo electrónico si sus solicitudes fueron aprobadas o reprobadas.
- Anuncia al profesor, sobre los cambios en las listas de sus cursos y sobre quiénes son los estudiantes que ahora se encuentran en el curso.
- Genera reportes sobre los resultados y estadísticas del proceso de inclusiones una vez terminado.

2.4 Visión general de la estructura documento

A continuación se detallan las secciones del documento:

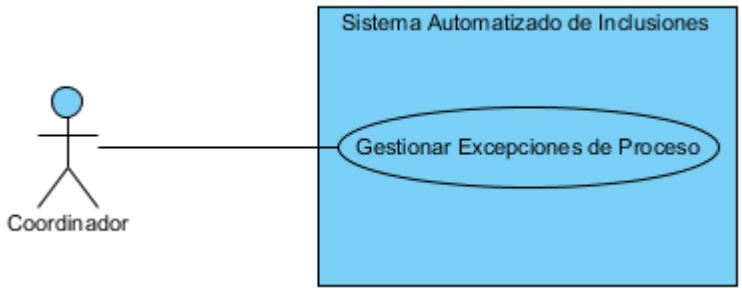
- **Introducción:** define el problema general que va a solucionar el producto, el trabajo realizado para la segunda iteración, la división de trabajo de dicha iteración y el objetivo específico, así como criterios de éxito) del caso de uso correspondiente al encargado de esta parte de la iteración.
- **Requerimientos funcionales:** En esta sección se analiza la eventual funcionalidad que se le debe dar al sistema. También, se definen los casos de uso que van a ser desarrollados para el sistema para este proyecto.
- **Requerimientos no funcionales:** En este apartado se analizan estándares y acuerdos que han sido definidos para el proyecto y los cuales servirán como guía en el desarrollo del sistema.
- **Apéndice:** una serie de secciones añadidas para dar completitud y sentido al documento, incluyendo glosario para abreviaturas y conceptos, así como diagramas Gantt.

3 Requerimientos funcionales

3.1 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. En esta sección, definiremos el contexto o ambiente en el cual será utilizado el sistema.

3.1.1 Diagrama de Casos de Uso



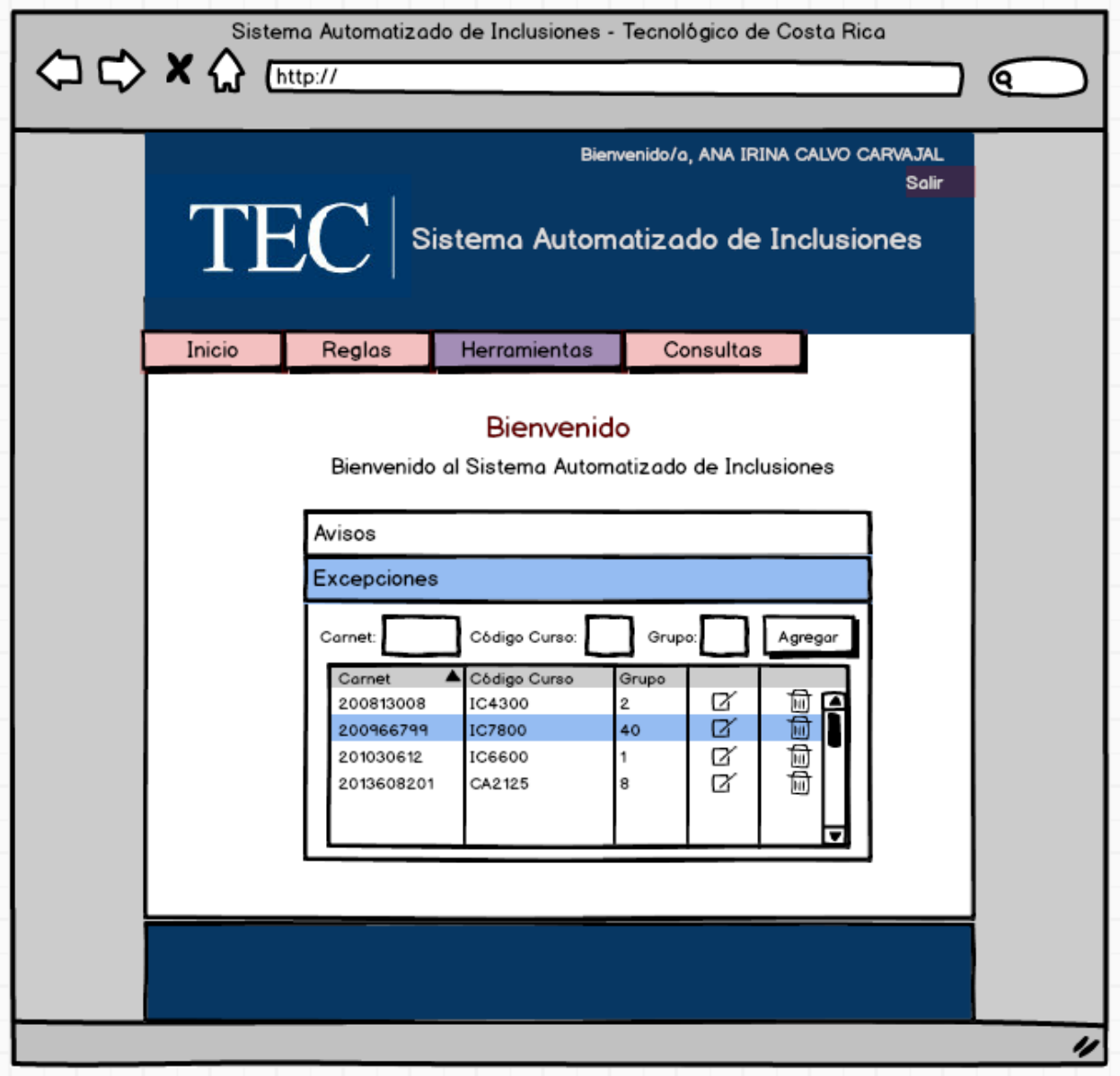
3.2 Descripción detallada de cada Caso de Uso.

3.2.1 Caso de Uso CU-08: Gestionar Excepciones de Proceso

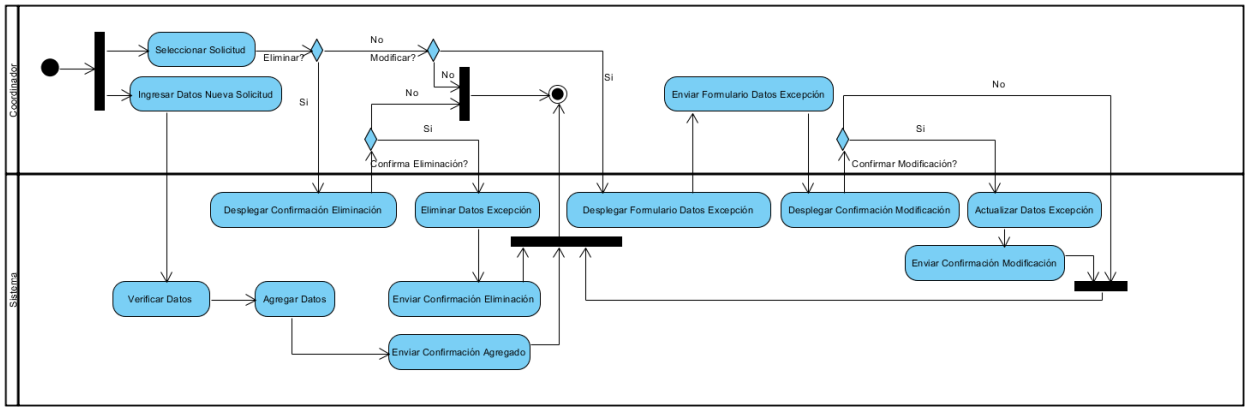
3.2.1.1 Texto del CU

ID de Caso de Uso:	CU-08	**Nombre del Caso de Uso:**	Gestionar Excepciones de Proceso
Creado Por:	Juan José Rojas	**Modificado Por:**	Juan José Rojas
Fecha Creación:	26/12/2013	**Fecha Modificación:**	03/01/2014
Actores:	Coordinador		
Descripción:	El Coordinador crea, modifica o elimina una excepción para que ésta sea tomada en cuenta como prioridad en el proceso de Asignación Automática de cupos.		
Precondiciones:	1. El coordinador debe estar autenticado. 2. La solicitud de inclusión debe ser válida y en estado "Pendiente".		
Postcondiciones:	1. Se actualiza la información de las excepciones en la Base de Datos del sistema.		
Escenario:	**1.0 Modificar una Excepción** 1. El coordinador selecciona el botón de edición de la excepción a modificar. 2. El sistema despliega la información de la excepción. 3. El coordinador cambia la información de la excepción y aplica los cambios. 4. El sistema actualiza los datos en la pantalla inicial y en la Base de Datos e informa al estudiante acerca del éxito del proceso.		
Alternativas:	**1.1 Eliminar una Excepción** **1.2 Agregar una Excepción**		
Escenarios Alternativos:	**1.1 Eliminar una Excepción (en el paso 1)** 1. El coordinador selecciona el botón de borrado de la excepción a eliminar. 2. El sistema despliega un aviso de confirmación de eliminación. 3. El coordinador confirma la eliminación de la excepción. 4. El sistema borra la excepción de la pantalla inicial y de la Base de Datos. **1.2 Agregar una Excepción (en el paso 1)** 1. El coordinador ingresa la información de la excepción y lo envía. 2. El sistema verifica los datos y agrega la excepción de la pantalla inicial y en la Base de Datos.		
Excepciones:	**1.0.E.1 El sistema no puede acceder a la Base de Datos (en el paso 6)** 1. El sistema informa acerca del error al coordinador. 2. El sistema termina el caso de uso.		
Prioridad:	Alta		
Frecuencia de Uso:	Aproximadamente 100 excepciones serán agregadas por período.		
Reglas de Negocio:	El caso de uso sólo podrá ser accedido por el Coordinador de Carrera o encargado designado al proceso de inclusiones.		
Requerimientos Especiales:	1. El sistema debe poder accesar la información de estudiantes y cursos del Departamento de Admisión y Registro para poder verificar la existencia y validez de los datos.		
Supuestos:	1. Todos los cursos descritos en este caso son cursos válidos. 2. Todos los cursos descritos en este caso se encuentran habilitados para el período en cuestión.		
Notas y Detalles:	Fecha Límite: 03/01/2014.		

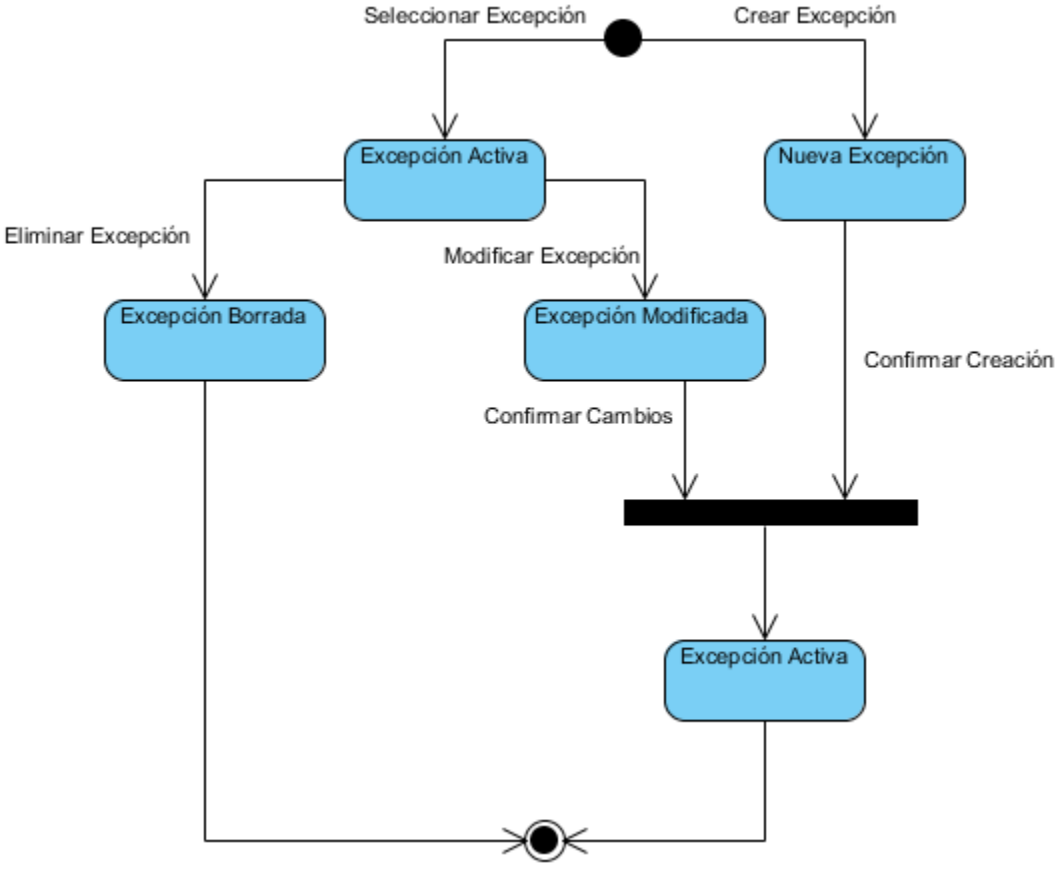
3.2.1.2 Pantalla (s) y/o reporte (s) del CU



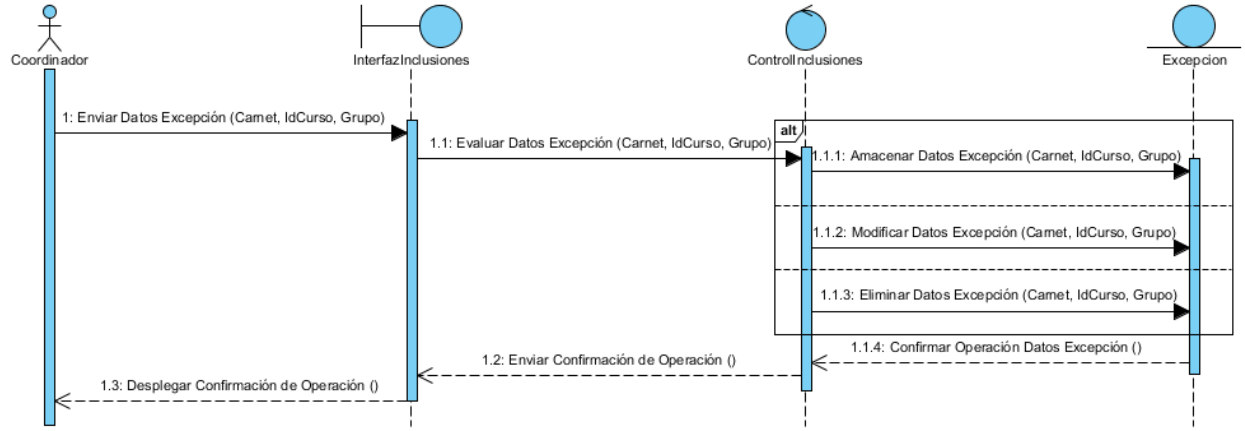
3.2.1.3 Diagrama de actividades del CU



3.2.1.4 Diagrama de estados del CU



3.2.1.5 Diagrama secuencia del sistema (DSS)



3.2.1.6 Contrato de Operaciones

Contrato C08: Gestionar Excepciones de Proceso

Operación	modificarSolicitud()
Parámetros	• Carnet : integer • idCurso: varchar(6) • idGrupo : integer
Referencias	CU-08
Precondiciones	1. El coordinador debe estar autenticado. 2. La solicitud de inclusión debe ser válida y en estado "Pendiente".
Postcondiciones	1. Se actualiza la información de las excepciones en la Base de Datos del sistema.

3.3 Patrones de requerimientos

3.3.1 Entidad de datos

Los datos que se necesitarán en este caso de uso provienen de dos bases de datos distintas: la base de datos de Admisión y Registro, a la cual se accede mediante conexión a un web service wsDAR, y la base de datos del sistema, en la cual se guardan los datos referentes a las inclusiones.

Para este caso de uso se necesitan los datos del estudiante, de los cursos a los que se puede realizar inclusión en el periodo, los grupos por curso, así como los cursos y grupos ya matriculados en el periodo.

3.3.2 Funciones de usuarios

Los usuarios del sistema, estudiantes, deberán escoger la opción de modificar la solicitud y cambiar los grupos que se relacionan con la solicitud, de esta manera el usuario cumplirá con el caso de uso.

4 Requerimientos no funcionales

4.1 RNF-01: Patrones de requerimientos del producto

Para el sistema automatizado de inclusiones se tienen dos módulos distintos: a) el módulo web y b) el módulo móvil. En los párrafos siguientes describiremos los estándares en cuanto a interfaz gráfica para ambos módulos, así como mostrar algunas pantallas de ejemplo del prototipo.

En general, las características gráficas que se quisieron lograr fueron que la interfaz fuera:

a. Minimalista
b. Plana
c. Amigable
d. Obvia

4.1.1 Interfaz web del usuario

4.1.1.1 Descripción

La idea general en cuanto a la interfaz del módulo web del sistema de inclusiones es que además de las funcionalidades principales solicitadas por el cliente, los datos puedan mostrarse de forma ordenada y minimalista, pero aprovechando el espacio de una ventana de computadora. Una de las grandes diferencias con la aplicación de Android es que la representación del formulario digital de solicitud de inclusión tiene un formato más parecido al antiguo formulario en papel.

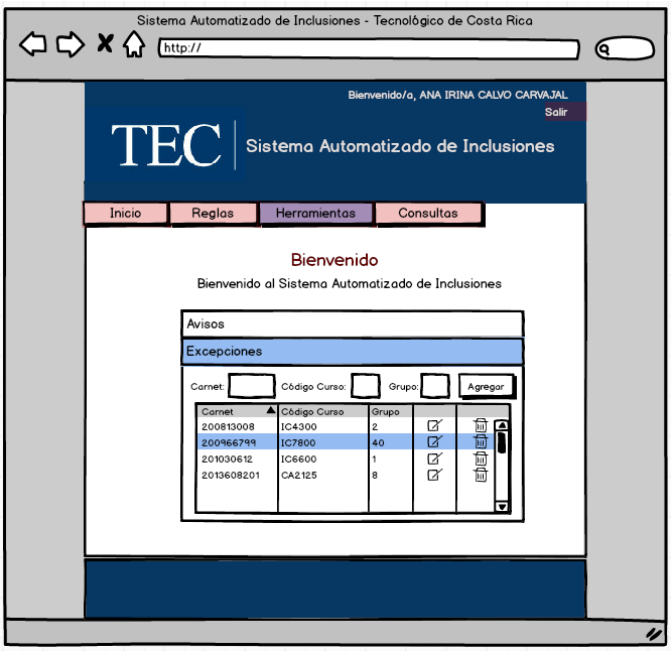
4.1.1.2 Estándar de tipos de fuente

A continuación describiremos el estándar de tipos de fuente para el módulo web del sistema que va a ser desarrollado. Se utilizará un tipo de fuente Serif para el Título 1 con el resto de los tipos de fuente simples.

- Título 1: Serif 30pt negrita
- Título 2: Sans 20pt
- Título 3: Sans 16pt
- Cuerpo: Sans 12pt
- Notas: Sans 10pt
- Botón: Sans 16pt

4.1.1.3 Esquema de colores

4.1.1.4 Ejemplo de pantalla



4.1.1.5 Formato de reportes

Para este caso de uso no existen reportes agregados, ya que para el caso de uso de dicha iteración solamente es necesaria una vista general de las excepciones a tomar en consideración en la asignación automática de cupos.

4.1.2 Rendimiento

El rendimiento de la aplicación se ve afectada por los servicios web de la aplicación, operaciones en la base de datos y los servicios web del departamento de Admisión y Registro.

4.1.3 Flexibilidad

La aplicación solo puede ser utilizada cuando el dispositivo con sistema operativo Android que posea conexión a Internet, ya que todas las operaciones de la aplicación se hacen mediante servicios web.

4.1.4 Control de acceso

El ingreso a la aplicación se da mediante el carné y el pin del estudiante. Al ingresar en la aplicación, se despliega una pantalla de inicio de sesión en la cual se utiliza el carné y el pin. Una vez que el estudiante ha iniciado sesión, se le permite el acceso a la aplicación.

5 Apéndices

5.1 Carta de recibido del usuario

1/2/2014
Tecnológico de Costa Rica, Cartago

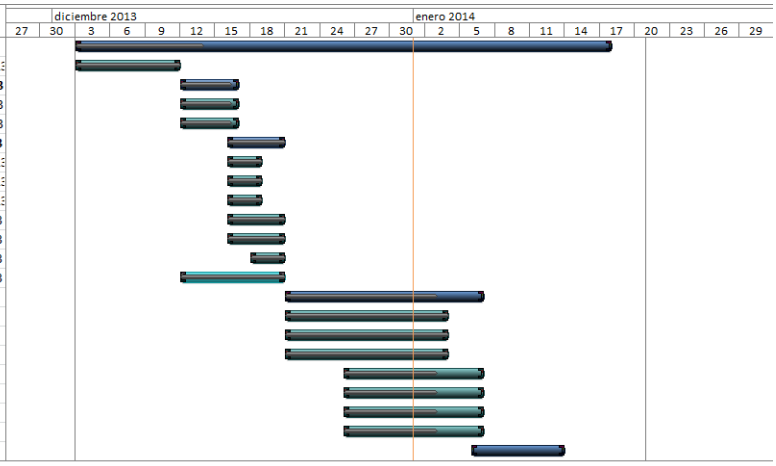
A quien corresponda:

Por este medio hago constar que he revisado y aprobado la ejecución de la segunda iteración del proyecto del "Sistema automatizado de inclusiones", que se encuentra en una computadora de la Escuela de Ingeniería en Computación, con el cual estoy de acuerdo, así como los documentos de especificación de requerimientos, arquitectura y el plan de pruebas, elaborados por los estudiantes Ana Irina Calvo Carvajal carné 200966799, Andrés Eduardo González Ortíz carné 201016317, Jose Arnoldo Segura Campos carné 201030612 y Juan José Rojas Valverde carné 200813008.

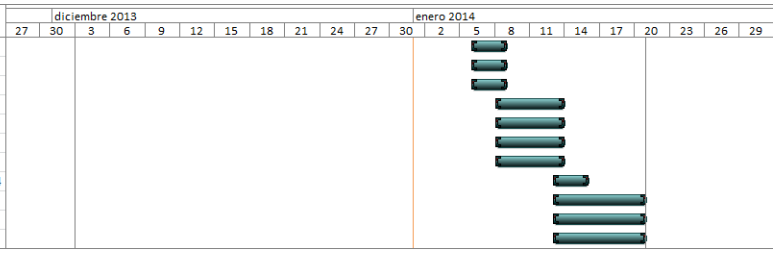
Adriana Álvarez Figueroa Cliente del proyecto Profesora de la Escuela de Ingeniería en Computación	Jose Arnoldo Segura Campos Coordinador del proyecto Estudiante de la Escuela de Ingeniería en Computación

5.2 Plan de proyecto actual

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1		**Sistema Automatizado**	**34 días**	**mar 12/3/13**	**vie 1/17/14**
2		Búsqueda del Proyecto	7 días	mar 12/3/13	mié 12/11/13
3		**Visión y Prototipo**	**3 días**	**jue 12/12/13**	**lun 12/16/13**
4		Documento de Visión	3 días	jue 12/12/13	lun 12/16/13
5		Prototipo completo	3 días	jue 12/12/13	lun 12/16/13
6		**Iteración 1**	**5 días**	**lun 12/16/13**	**vie 12/20/13**
7		It.1 = ERS	3 días	lun 12/16/13	mié 12/18/13
8		It.1 = Plan de pruebas	3 días	lun 12/16/13	mié 12/18/13
9		It.1 = DAS	3 días	lun 12/16/13	mié 12/18/13
10		It.1 = Ejecución	5 días	lun 12/16/13	vie 12/20/13
11		It.1 = Informe de prueb	5 días	lun 12/16/13	vie 12/20/13
12		It.1 = Usuario: Manual, c	3 días	mié 12/18/13	vie 12/20/13
13		It.1 = Portafolio individ	7 días	jue 12/12/13	vie 12/20/13
14		**Iteración 2**	**12 días**	**sáb 12/21/13**	**lun 1/6/14**
15		It.2 = ERS	11 días	sáb 12/21/13	vie 1/3/14
16		It.2 = Plan de pruebas	11 días	sáb 12/21/13	vie 1/3/14
17		It.2 = DAS	11 días	sáb 12/21/13	vie 1/3/14
18		It.2 = Ejecución	8 días	jue 12/26/13	lun 1/6/14
19		It.2 = Informe de prueb	8 días	jue 12/26/13	lun 1/6/14
20		It.2 = Usuario: Manual, c	8 días	jue 12/26/13	lun 1/6/14
21		It.2 = Portafolio individ	8 días	jue 12/26/13	lun 1/6/14
22		**Iteración 3**	**6 días**	**lun 1/6/14**	**lun 1/13/14**

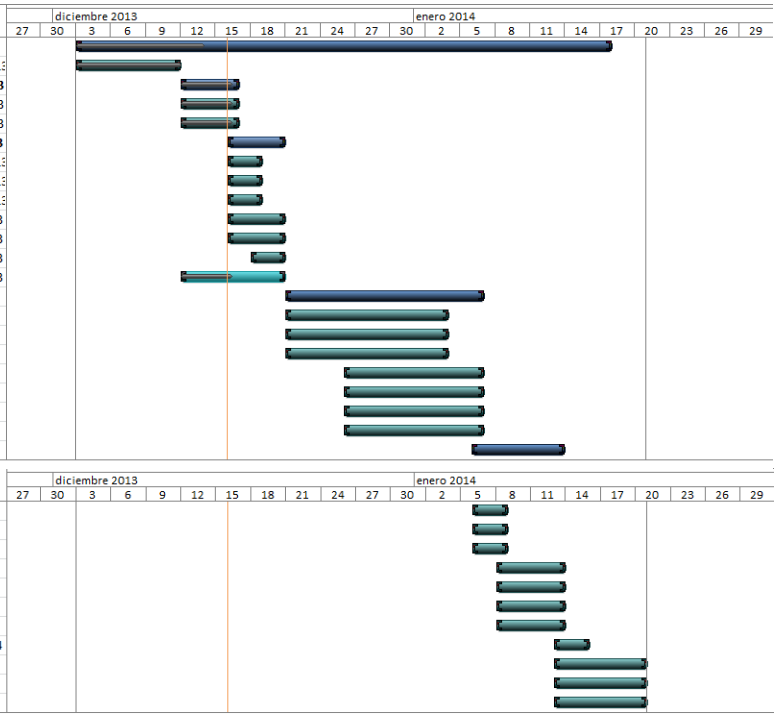


Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
23		It.3 = ERS	3 días	lun 1/6/14	mié 1/8/14
24		It.3 = Plan de pruebas	3 días	lun 1/6/14	mié 1/8/14
25		It.3 = DAS	3 días	lun 1/6/14	mié 1/8/14
26		It.3 = Ejecución	4 días	mié 1/8/14	lun 1/13/14
27		It.3 = Informe de prueb	4 días	mié 1/8/14	lun 1/13/14
28		It.3 = Usuario: Manual, c	4 días	mié 1/8/14	lun 1/13/14
29		It.3 = Portafolio individ	4 días	mié 1/8/14	lun 1/13/14
30		**Entrega**	**3 días**	**lun 1/13/14**	**mié 1/15/14**
31		Pruebas de sistema	6 días	lun 1/13/14	lun 1/20/14
32		Manual del Sistema	6 días	lun 1/13/14	lun 1/20/14
33		Calificación y minutas e	6 días	lun 1/13/14	lun 1/20/14



5.3 Plan de proyecto original

Id	Modo de tarea	Nombre de tarea	Duración	Comienzo	Fin
1		**Sistema Automatizado**	**34 días**	**mar 12/3/13**	**vie 1/17/14**
2		Búsqueda del Proyecto	7 días	mar 12/3/13	mié 12/11/13
3		**Visión y Prototipo**	**3 días**	**jue 12/12/13**	**lun 12/16/13**
4		Documento de Visión	3 días	jue 12/12/13	lun 12/16/13
5		Prototipo completo	3 días	jue 12/12/13	lun 12/16/13
6		**Iteración 1**	**5 días**	**lun 12/16/13**	**vie 12/20/13**
7		It.1 = ERS	3 días	lun 12/16/13	mié 12/18/13
8		It.1 = Plan de pruebas	3 días	lun 12/16/13	mié 12/18/13
9		It.1 = DAS	3 días	lun 12/16/13	mié 12/18/13
10		It.1 = Ejecución	5 días	lun 12/16/13	vie 12/20/13
11		It.1 = Informe de prueb	5 días	lun 12/16/13	vie 12/20/13
12		It.1 = Usuario: Manual, c	3 días	mié 12/18/13	vie 12/20/13
13		It.1 = Portafolio individi	7 días	jue 12/12/13	vie 12/20/13
14		**Iteración 2**	**12 días**	**sáb 12/21/13**	**lun 1/6/14**
15		It.2 = ERS	11 días	sáb 12/21/13	vie 1/3/14
16		It.2 = Plan de pruebas	11 días	sáb 12/21/13	vie 1/3/14
17		It.2 = DAS	11 días	sáb 12/21/13	vie 1/3/14
18		It.2 = Ejecución	8 días	jue 12/26/13	lun 1/6/14
19		It.2 = Informe de prueb	8 días	jue 12/26/13	lun 1/6/14
20		It.2 = Usuario: Manual, c	8 días	jue 12/26/13	lun 1/6/14
21		It.2 = Portafolio individi	8 días	jue 12/26/13	lun 1/6/14
22		**Iteración 3**	**6 días**	**lun 1/6/14**	**lun 1/13/14**
23		It.3 = ERS	3 días	lun 1/6/14	mié 1/8/14
24		It.3 = Plan de pruebas	3 días	lun 1/6/14	mié 1/8/14
25		It.3 = DAS	3 días	lun 1/6/14	mié 1/8/14
26		It.3 = Ejecución	4 días	mié 1/8/14	lun 1/13/14
27		It.3 = Informe de prueb	4 días	mié 1/8/14	lun 1/13/14
28		It.3 = Usuario: Manual, c	4 días	mié 1/8/14	lun 1/13/14
29		It.3 = Portafolio individi	4 días	mié 1/8/14	lun 1/13/14
30		**Entrega**	**3 días**	**lun 1/13/14**	**mié 1/15/14**
31		Pruebas de sistema	6 días	lun 1/13/14	lun 1/20/14
32		Manual del Sistema	6 días	lun 1/13/14	lun 1/20/14
33		Calificación y minutas e	6 días	lun 1/13/14	lun 1/20/14



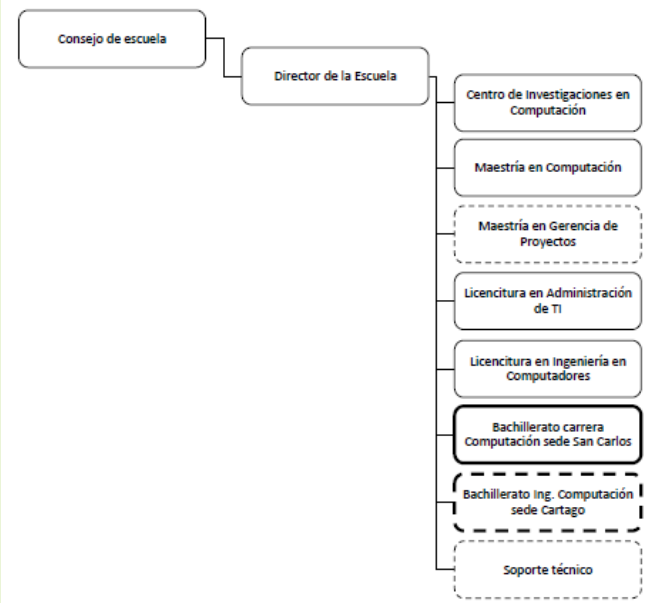
5.4 Glosario de términos y abreviaturas

Abreviatura	Significado
CU-001	Caso de Uso 1.
DAS	Documento de Arquitectura de Software.
ERS	Especificación de Requerimientos de Software.
It.	Iteración.

5.5 Lista de riesgos

Riesgo	*Nivel de riesgo*	*Probabilidad de ocurrencia*	*Plan de mitigación*
Problemas de conectividad con el web service de datos de estudiantes.	Medio	Probable	Tener una buena comunicación con el Centro de Cómputo desde el inicio, de manera que esa información sea fácilmente accesible por el sistema.
Periodo de vacaciones de los funcionarios del Tec, limitando la comunicación con Centro de Cómputo y Registro	Medio - Alto	Probable	Identificar previamente las actividades que dependen de esas entidades y realizarlas con tiempo.
Inconsistencias con los estándares y políticas de desarrollo necesarios.	Medio	Poco probable	Tener claros los estándares de desarrollo del CC antes de trabajar con el mismo.
Incompatibilidad entre las herramientas o plataformas utilizadas.	Bajo	Poco probable	Tener definido desde temprano las plataformas y herramientas estándar utilizadas por el CC y por el proyecto.
Conocimiento limitado de desarrollo para móviles por parte del equipo de trabajo.	Bajo	Probable	Capacitación previa al desarrollo del app móvil.
Fallas en el hardware de la escuela de IC o el centro de cómputo.	Alto	Muy poco probable	Tener una buena comunicación con el CC para conocer sobre el estado actual del equipo, posibles períodos de downtime, entre otros.

5.6 Descripción de la empresa (departamento) (nombre, dirección, descripción general, organigrama (indicar depto del sistema), persona contacto, email, teléfono)

Institución	Escuela de Ingeniería en Computación
Dirección	Tecnológico de Costa Rica
Descripción General	
Organigrama	
Persona Contacto	Adriana Álvarez
Email	adriana.alvarezf@gmail.com
Teléfono	

5.7 Especificación de estándares Programación (Ejemplos de Interfaz nivel local y Web, Base de datos, nombres de atributos, clases, etc.)

5.7.1 Estándar de programación código fuente

5.7.1.1 Distribución del Código

El código fuente debe tener el siguiente orden en su estructura:

- ✓ Encabezado.
- ✓ Imports.
- ✓ Definición del *namespace.*
- ✓ Declaración de clase debidamente comentada.
- ✓ Constantes: de la visibilidad menos restrictiva a la más restrictiva. Se recomienda crear clases o interfaces únicamente para almacenar constantes.
- ✓ Atributos: de la visibilidad menos restrictiva a la más restrictiva Propiedades: de la visibilidad menos restrictiva a la más restrictiva.

- ✓ Constructor(es).
- ✓ Destructor en caso de que exista.
- ✓ Métodos: de la visibilidad menos restrictiva a la más restrictiva.
- ✓ Properties (métodos *set* y *get).*

5.7.1.2 Nombres de Proyectos y Soluciones

Los nombres que se le den a las soluciones quedan a criterio del desarrollador, pero debe ser un nombre significativo para el proyecto en desarrollo.

El nombre de la solución debe ser:
ITCR.Nombre de la solución

Los nombres de los proyectos deben tener el siguiente formato:
ITCR.Nombre de la solución + Capa

- ✓ Capa se refiere a la capa en la cual se está trabajando (Base, Negocios, Datos)

Ejemplo:
- ✓ ITCR.SeguridadITCR.Datos
- ✓ ITCR.Cajas.Negocios

En Visual Studio se desestimó el uso de namespaces para la capa de interfaz en aplicaciones web desde su versión 2005. Por lo que no se incluyen los namespaces .Interfaz en este estándar. Los de Datos y Negocios se siguen utilizando normalmente.

Al definir estos nombres para los proyectos de la solución, los namespaces quedarán definidos.

5.7.1.3 Nombres de Namespaces

El namespace debe tener la siguiente estructura para cualquier clase y capa. Esto ya debe estar definido desde que se creó la solución y los proyectos:

ITCR. + Sistema.Capa
- ✓ Sistema: nombre del sistema que se está desarrollando.
- ✓ Capa: capa en la cual se está trabajando.

Ejemplos:
- ✓ ITCR.Compras.Datos
- ✓ ITCR.SeguridadITCR.Negocios
- ✓ ITCR.Cajas.Negocios

5.7.1.4 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.

5.7.1.4.1 Documentación interna de clases

Cada clase debe iniciar con el nombre del proyecto, descripción, fecha y hora de creación y si es necesario observaciones.

Ejemplo:

```
#region Acerca de...
        ///////////////////////////////////////////////////////////////////////
        // Instituto Tecnológico de Costa Rica
        // Proyecto: Framework Matricula
        // Descripción: Clase de acceso a datos para tabla 'IEPEFDEPTOS'
        // Generado por ITCR Gen v1.0.0.0
        // Fecha: Martes, 11 de Mayo de 2004, 11:50:00 a.m.
        // Dado que esta clase implementa IDispose, las clases derivadas no
        // deben hacerlo.
        ///////////////////////////////////////////////////////////////////////
#endregion
```

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();
....
}
```

5.7.1.4.2 Estándar de la clase base

La clase base no debe modificarse, siempre que se tenga que hacer un cambio debe hacerse en su respectiva clase de datos o negocios. Esto debido a que si se da un cambio en la tabla respectiva en la base de datos se puede remplazar esta clase y los cambios por herencia se verán reflejados en las capas de datos y negocios debido a su relación de herencia.

5.7.1.4.3 Estándar de la clase de negocios

Debe seguir el estándar para cualquier otra clase y si es necesario debe derivarse de la clase de Datos.

5.7.1.4.4 Estándar de la clase de datos

Debe seguir el estándar para cualquier otra clase y debe derivarse de la clase Base.

5.7.1.5 Interfaces

Los nombres de las interfaces deben ser adjetivos.

5.7.1.5.1 Interface Web

WIConPalabrasInternasPrimerLetraEnMayúscula

5.7.1.5.2 Interface Local

IConPalabrasInternasPrimerLetraEnMayúscula

5.7.1.5.3 IAccesoADatos

Toda clase debe cumplir con la Interface IaccesoADatos y por lo tanto tener los métodos Insertar, Actualizar, Eliminar, SeleccionarUno, SeleccionarTodos, Buscar. En esta clase se hace uso de un archivo ".config" que contiene el string de conexión.

Ejemplo:

```
public interface IAccesoADatos
{
      bool Insertar();
      bool Actualizar();
      bool Eliminar();
      DataTable SeleccionarUno();
      DataTable SeleccionarTodos();
      DataTable Buscar();
}
```

5.7.1.6 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"

5.7.1.6.1 Parámetros

- ✓ Debe seguir el estándar de las variables y agregar la letra 'p' antes del underscore (_). Ejemplo: p_usrName.
- ✓ Un método no debe tener más de 6 parámetros ni ser mayor a 100 líneas de código.

5.7.1.7 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).

5.7.1.7.1 Indicadores de tipo

Este indicador debe ser el prefijo para todas las variables en las clases y métodos, los más comunes son:

Tipo	Indicador
Integer	i
String	s
DateTime	da
Bit (equivalente a Bool)	b

5.5.1.8 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

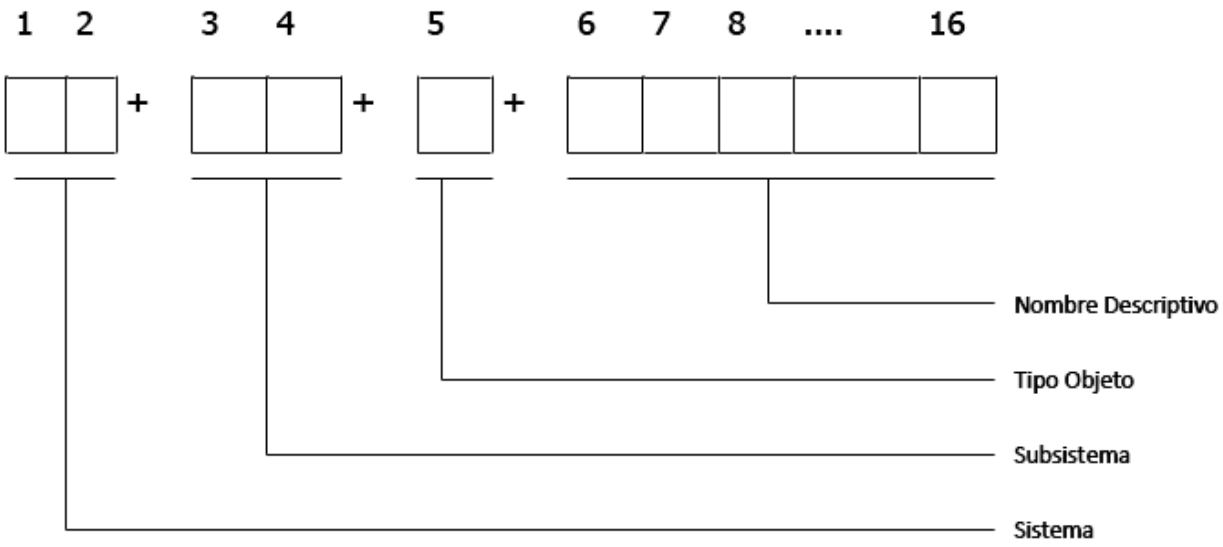
5.7.1.8 Properties

- ✓ 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.

5.7.2 Estándar de objetos de la 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:



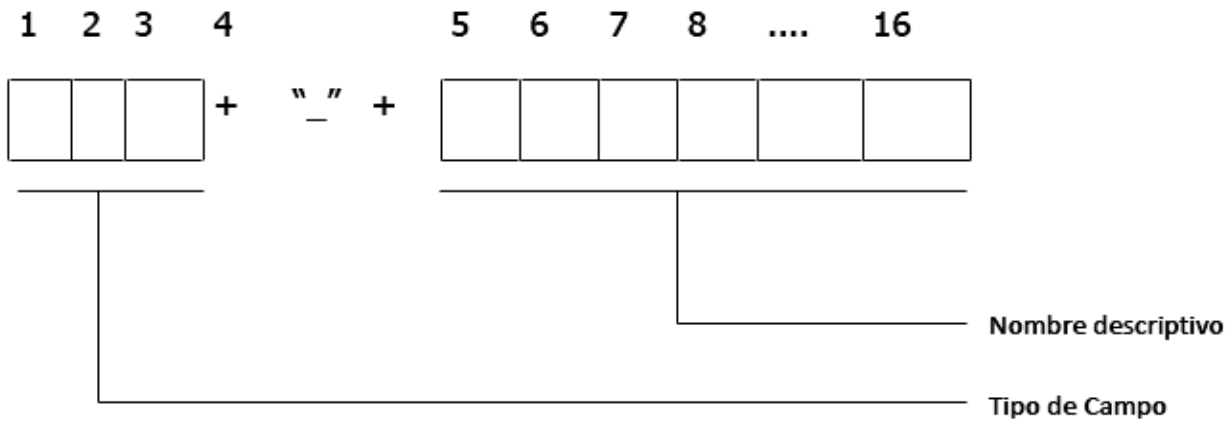
- ✓ Sistema: Corresponde a una tira de dos caracteres que identifica el sistema al que pertenece el objeto.
- ✓ Subsistema: Corresponde a una tira de dos caracteres que identifica el subsistema al que pertenece el objeto.
- ✓ Tipo Objeto: Si es una tabla puede ser “F” (datos fijos) o “T” (temporal). 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).

5.7.2.1 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.

5.7.2.2 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

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

5.7.3.1 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:

```
-----------------------------------------------------------------
-- Procedimiento almacenado que selecciona una fila de la tabla
-- 'IEMAFESTUD_ACADE'
-- basado en la llave primaria.
-- Recibe: @sIDE_ESTUDIANTE varchar(12)
-- Retorna: @iCodError int
```

- ✓ 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>
```