12/23/2013

SISTEMA AUTOMATIZADO DE INCLUSIONES

Especificación de Requerimientos del Sistema

Versión 1.3

# 

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](#_Toc375669902)

[1.1 Historial de cambios 3](#_Toc375669903)

[1.2 Aprobación del documento 3](#_Toc375669904)

[2 Introducción 4](#_Toc375669905)

[2.1 Propósito del documento 4](#_Toc375669906)

[2.2 Descripción del problema 4](#_Toc375669907)

[2.3 Lista de problemas detectados 5](#_Toc375669908)

[2.4 Lista de fortalezas detectadas 5](#_Toc375669909)

[2.5 Objetivos del sistema 5](#_Toc375669910)

[2.5.1 Objetivo general 5](#_Toc375669911)

[2.5.2 Objetivos específicos (Como Casos Uso) 6](#_Toc375669912)

[2.6 Perspectiva del producto por desarrollar 6](#_Toc375669913)

[2.7 Reglas del negocio 7](#_Toc375669914)

[2.8 Suposiciones y dependencias 7](#_Toc375669915)

[2.8.1 Suposiciones 7](#_Toc375669916)

[2.8.2 Dependencias 8](#_Toc375669917)

[2.9 Alcances del sistema 8](#_Toc375669918)

[2.10 Limitaciones o restricciones 9](#_Toc375669919)

[2.11 Stakeholders y sus necesidades 9](#_Toc375669920)

[2.11.1 Tabla de perfiles 10](#_Toc375669921)

[2.12 Visión general de la estructura documento 10](#_Toc375669922)

[3 Requerimientos funcionales 11](#_Toc375669923)

[3.1 Contexto del sistema 11](#_Toc375669924)

[3.1.1 Diagrama de contexto 11](#_Toc375669925)

[3.1.2 Modelo dominio del sistema 12](#_Toc375669926)

[3.1.3 Descripción modelo dominio (Para cada concepto indicar intención y extensión) 12](#_Toc375669927)

[3.1.4 Diagrama de casos de uso 1 iteración 13](#_Toc375669928)

[3.2 Descripción detallada cada CU 1 iteración (ordenados forma descend. por prioridad). 15](#_Toc375669929)

[3.2.1 Caso de uso 1: Crear Solicitud 15](#_Toc375669930)

[3.2.2 Caso de uso 2: Revisar Estado de Solicitudes 19](#_Toc375669931)

[3.2.3 Caso de uso 3: Definir Periodo de Recepción de Solicitudes 22](#_Toc375669932)

[3.2.4 Caso de uso 4: Modificar una solicitud 26](#_Toc375669933)

[3.2.5 Caso de uso 5: Anular Solicitud 30](#_Toc375669934)

[4 Requerimientos no funcionales 34](#_Toc375669935)

[4.1 RNF-01: Patrones de requerimientos del producto 34](#_Toc375669936)

[4.1.1 Interfaz web del usuario 34](#_Toc375669937)

[4.1.2 Interfaz móvil del usuario 35](#_Toc375669938)

[4.2 RNF-02: Organizacionales 36](#_Toc375669939)

[4.2.1 Patrón documentación 36](#_Toc375669940)

[4.2.2 Entregas 37](#_Toc375669941)

[4.2.3 Implementación 37](#_Toc375669942)

[4.3 RNF-03: Externos 37](#_Toc375669943)

[4.3.1 Patrón Interface e interacción entre sistemas 37](#_Toc375669944)

[4.3.2 Legales 38](#_Toc375669945)

[5 Apéndices 39](#_Toc375669946)

[5.1 Plan del proyecto en sólo 1 pág (Gantt en Project) (Desglose semanal y no diario) 39](#_Toc375669947)

[5.2 Glosario de términos y abreviaturas (orden alfabético) 40](#_Toc375669948)

[5.3 Lista de riesgos (orden descendente de prioridad) Incluir nombre, descrip, prioridad, responsable, etc 40](#_Toc375669949)

[5.4 Descripción de la empresa (departamento) (nombre, dirección, descripción general, organigrama (indicar depto del sistema), persona contacto, email, teléfono) 41](#_Toc375669950)

[5.5 Especificación de estándares Programación (Ejemplos de Interfaz nivel local y Web, Base de datos, nombres de atributos, clases, etc.) 41](#_Toc375669951)

[5.5.1 Estándar de programación código fuente 41](#_Toc375669952)

[5.5.2 Estándar de objetos de la base de datos 47](#_Toc375669953)

[5.5.3 Estándar de programación de la base de datos (SQL) 49](#_Toc375669954)

# Control del Documento

## 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 | 12/17/2013 | Ana Irina Calvo Carvajal | Agregar interfaz y darle formato |
| 1.1 | 12/23/2013 | 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 | 12/23/2013 | Andrés González Ortiz | Corrección del objetivo general y criterios de éxito, corrección de diagrama de casos de uso |

## 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 |  |

# Introducción

## Propósito del documento

Este documento define los requerimientos funcionales y no funcionales del software del Sistema de Inclusiones de la Escuela de Ingeniería en Computación.

## Descripción del problema

|  |  |
| --- | --- |
| El problema de | Manejo de inclusiones |
| Afecta a | * Estudiantes. * Profesores. * Escuela de Ingeniería en Computación. |
| El impacto del cual es | * Lentitud en el proceso de solicitud de inclusiones. * Posibles errores en los formularios. * Notificación tardía de aceptación en grupos a los estudiantes. * Carga de trabajo excesiva por cantidad de solicitudes recibidas. * Gasto de papel en la Escuela. |
| Una solución exitosa debería ser | El desarrollo de una aplicación web que automatice el proceso de envío y revisión de solicitudes de inclusión, que aligere la carga de trabajo de los encargados de revisar las solicitudes, agilizando el proceso y reduciendo el gasto de papel en la Escuela de IC. |

## Lista de problemas detectados

Los problemas del cliente que se solucionarán con este sistema son:

* El paso del sistema manual actual a un sistema virtual.
* La dificultad de manejar todas las solicitudes pertenecientes a las tres carreras de la Escuela de Ingeniería en Computación.
* El crecimiento semestral de solicitudes de inclusión en la escuela.
* La dificultad de aplicar las reglas de negocios a todas las solicitudes.
* La gran cantidad de tiempo que toma el procesar las solicitudes (entre dos y tres semanas).

## Lista de fortalezas detectadas

Las fortalezas del sistema actual que nos beneficiarán con el desarrollo de este sistema son:

* El proceso de solicitar la inclusión se encuentra muy bien definido.
* Se conocen las reglas de negocio a implementar.
* Se tienen bien definidos los reportes o consultas a implementar.

## Objetivos del sistema

### 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 Casos Uso)

#### Objetivo específico

1. Aceptar el envío de solicitudes de inclusión de parte de estudiantes de la escuela de IC durante un lapso de tiempo determinado.
2. Verificar el estado actual de las solicitudes enviadas.

#### Criterio de éxito

Una medida clave del éxito de este proyecto es cumplir con los principales casos de uso, es decir, la creación, anulación, modificación y verificación de solicitudes electrónicas de inclusión de parte de los estudiantes de la escuela de IC del ITCR, además de la definición de períodos de recepción de solicitudes de inclusión de parte de un coordinador de la escuela; todo esto para más tardar el 7 de enero del 2014.

## Perspectiva del producto por desarrollar

|  |  |
| --- | --- |
| Para | Coordinación de la Escuela de Ingeniería en Computación, Tecnológico de Costa Rica. |
| Quién | Necesita agilizar los procesos solicitud y trámite de inclusión en los cursos semestrales que imparte la escuela. |
| El Sistema Automatizado de Inclusiones | Es una aplicación web. |
| Qué | Recibe, analiza y califica las solicitudes de inclusiones de forma eficiente y automatizada. |
| A diferencia del | Proceso manual. |
| Nuestro product | * 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. |

## Reglas del negocio

La funcionalidad de este sistema se rige bajo reglamentos establecidos por el departamento de Admisión y Registro. Estos reglamentos se citan a continuación:

* Estudiantes cuya diferencia entre el año de ingreso a la carrera y el año en el que se va a matricular sea mayor o igual a ocho años.
* Estudiantes cuya cantidad de cursos faltantes sea menor a cinco.
* Estudiantes que solo tengan opción a matricular menos de 10 créditos según su plan de estudio.
* Casos especiales:
  + Estudiantes beneficiados con algún tipo de beca que requiera un mínimo de créditos matriculados y que no tengan opciones suficientes para matricular.
  + Estudiantes procedentes de programas de intercambio, que requieran llevar un mínimo de cursos.
  + Estudiantes que presenten alguna discapacidad que requiera una matrícula en un curso específico.
* Asignación por fecha y hora de cita de matrícula ordinaria.

## Suposiciones y dependencias

### Suposiciones

* El ambiente final de la aplicación, Windows Server 2012 se encontrará listo en el departamento de Soporte una vez que comience en la instalación de la aplicación.
* La aplicación no presentará problemas de adaptación o compatibilidad con el entorno final.
* El sistema desarrollado durante la etapa I funcionará como es descrito en la parte de “Resultados del Test” del documento de “Documentación Técnica v1.0” de la misma etapa.
* Se trabajará con las mismas herramientas con las que se trabajó en la etapa I del proyecto.
* La aplicación se utilizará exclusivamente por la Escuela de Ingeniería en Computación pero en etapas posteriores se podrá expandir a otras escuelas del Tecnológico de Costa Rica.

### Dependencias

* Se necesitará de la información almacenada en las bases de datos del departamento de Admisión y Registro. Esta información corresponde a los datos de los estudiantes, ejemplo: citas de matrícula, historial académico, etc.; y a los datos de los cursos, ejemplo: nombre del curso y campos matriculados durante el periodo de matrícula ordinaria y extraordinaria del semestre actual.
* Se dependerá del Centro de Cómputo, ya que éste proveerá acceso a las bases de datos del departamento de Admisión y Registro por medio de web services.
* Se requerirá del departamento de Soporte de la Escuela de Ingeniería en Computación, ya que en este departamento se encuentran tanto el ambiente físico como el software que corresponde al entorno final de la aplicación.
* Se dependerá de los métodos de registro de la solicitud de inclusión de la etapa I del proyecto.

## Alcances del sistema

Una vez terminado el desarrollo del producto el estudiante podrá realizar, ya sea desde su computadora o dispositivo móvil, sus solicitudes de inclusión de una manera más rápida y eficaz.

Además, se garantiza tener un sistema que reduzca considerablemente el tiempo de respuesta de dichas solicitudes mediante la automatización de los procesos de selección, aprobación y notificación del resultado de estas inclusiones, acorde a los criterios de selección establecidos en los reglamentos vigentes.

Por último, se espera poder obtener información estadística adicional que permita solventar problemas como la toma de decisiones en cuanto a la apertura de nuevos grupos en un curso en particular de la escuela.

## Limitaciones o restricciones

A pesar de no ser contemplado en ésta etapa del proyecto, se podría tomar en cuenta para una futura etapa el uso de ésta información como medio para predecir la cantidad de grupos a abrir en un curso específico en períodos lectivos futuros.

## Stakeholders y sus necesidades

En cada proyecto se tienen interesados, los cuales influyen en el desarrollo y la aceptación del mismo.

Para este sistema se pueden encontrar distintos interesados, los cuales serán clasificados en varios perfiles:

* **Usuarios**: quiénes van a usar el sistema.
  + Estudiantes
  + Administradores
* **Clientes**: a quienes se les entrega el producto.
  + Profesora Adriana Álvarez
  + Escuela de Ingeniería en Computación
* **Desarrolladores**: quienes desarrollan el sistema.
  + Ana Irina Calvo Carvajal
  + Andrés Eduardo González Ortíz
  + Jose Arnoldo Segura Campos
  + Juan José Rojas Valverde

### Tabla de perfiles

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Interesado | Mayor valor | Actitudes | Mayor Intereses | Restricciones |
| Estudiante | * Agilización del proceso. * Reducir el tiempo de recepción de respuesta. | Usa el sistema que haya sido definido por la Escuela de la carrera. | Facilidad para realizar la inclusión, poder obtener respuesta rápida, interfaz amigable. | Solo puede acceder para enviar el formulario y revisar el estado de la inclusión. |
| Administrador | * Agilizar el proceso. * Automatización de tareas. | Entusiasmo por usar una herramienta que facilite el trabajo en el proceso. | Automatización del proceso de asignación, parametrización de los pasos. | Para realizar labores de administración en el sistema debe estar registrado como funcionario administrador. |
| Cliente | * Mejora en la productividad. * Automatización de tareasmanuales. * Cumplir con reglas de priorización. * Ahorro en tiempo. | Receptivo a ideas nuevas para resolver el problema. | Automatización del proceso de asignación, parametrización de los pasos. | Dificultad para reunirse en persona con el equipo de trabajo, tiempo correspondiente a verano. |
| Desarrollador | * Automatización de tareas manuales. | Abierto a sugerencias sobre el proyecto. | Desarrollo de todos los casos de uso, alineamiento con el tiempo estimado, proceso de automatización. | Solo se tiene el tiempo correspondiente al periodo de verano para desarrollar el sistema. |

## Visión general de la estructura documento

A continuación se detallan las secciones del documento:

* **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.

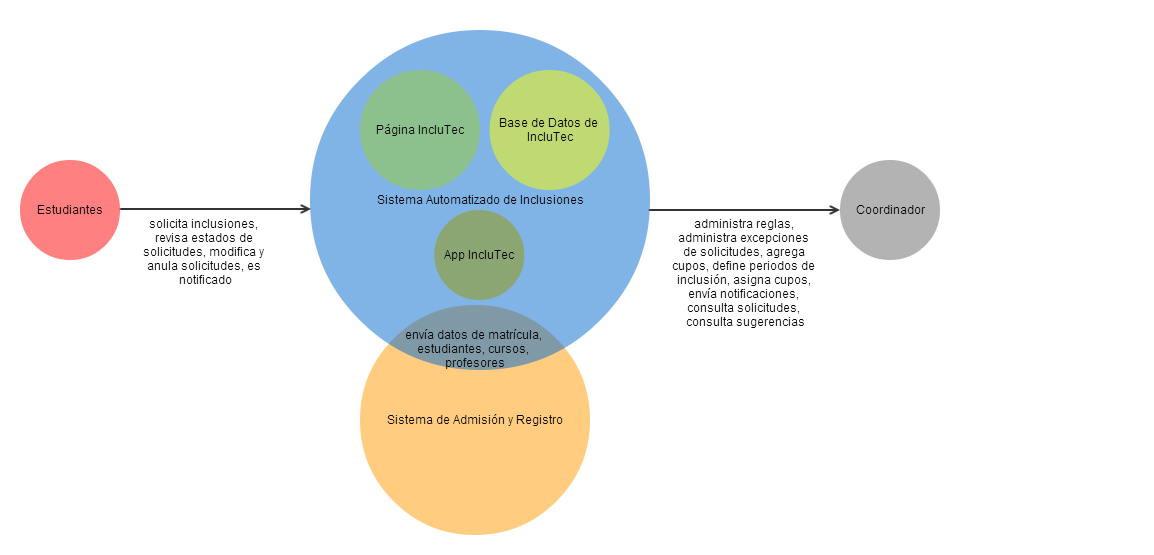
# Requerimientos funcionales

## Contexto del sistema

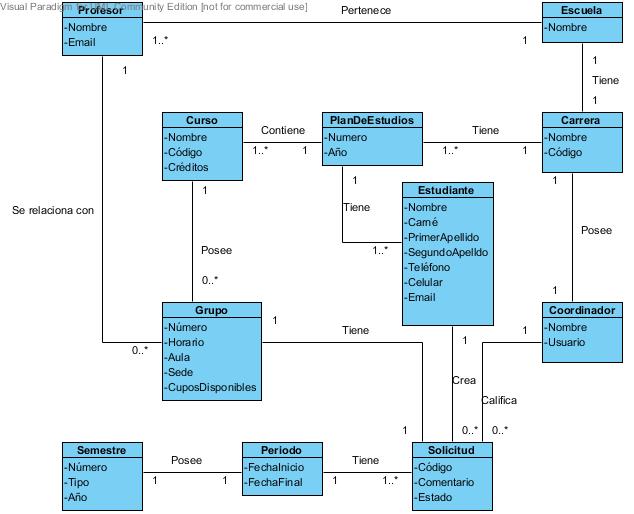
El sistema de inclusiones a desarrollar tiene como parte de sus características, que no solo recopilará formularios de solicitud de inclusión, sino que se integrará en un ambiente de software del Tecnológico de Costa Rica, por lo cual debe ser tomada en cuenta toda conexión que realice con otros sistemas 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.

### Diagrama de contexto

El siguiente es un diagrama que identificará los elementos pertenecientes al ambiente en el cual será utilizado el producto del proyecto de desarrollo del sistema automatizado de inclusiones, así como las interrelaciones entre ellos.



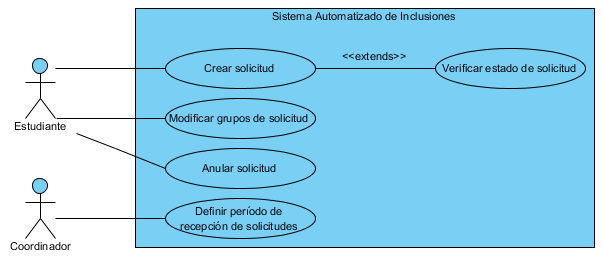
### Modelo dominio del sistema



### Descripción modelo dominio (Para cada concepto indicar intención y extensión)

|  |  |  |
| --- | --- | --- |
| Clase | Intención | Extensión |
| Profesor | Representa a los profesores a cargo de los cursos y que debe recibir notificaciones sobre las inclusiones en sus cursos. | Profesor  Nombre: Jaime Solano.  Email: Jaime.tec.10@gmail.com. |
| Escuela | Representa a departamento para el cual se está ejecutando el proceso de Inclusión. | Escuela  Nombre: Escuela de Ingeniería en Computación. |
| Curso | Representa a la materia del plan de estudios con la que se relacionan los grupos. | Curso  Nombre: Proyecto.  Código: IC7840.  Créditos: 4. |
| PlanDeEstudios | Corresponde a un conjunto de cursos que debe seguir el estudiante para graduarse. | Plan de Estudios  Número: 410.  Año: 2012. |
| Carrera | Simboliza la carrera a la que cual pertenece el plan de estudios al que se le está realizando el proceso de inclusiones. | Carrera  Nombre: Ingeniería en Computación.  Código: CA. |
| Grupo | Corresponde al grupo al cual el estudiante busca entrar mediante el proceso de inclusión. | Grupo  Número: 1.  Horario: L: 8:30am – 11:30, K: 8:30am – 11:30, M: 8:30am – 11:30.  Aula: B2-07.  Sede: Cartago.  CuposDisponibles: 5. |
| Coordinador | Representa a la persona encargada del proceso de inclusión de una determinada carrera. | Coordinador  Nombre: María Estrada.  Usuario: mestrada. |
| Semestre | Corresponde al actual periodo del año para el que se está ejecutando el proceso. | Semestre  Número: II.  Tipo: Verano.  Año: 2013. |
| Periodo | Corresponde al periodo habilitado para la recepción de solicitudes al sistema. | Periodo  FechaInicio: 2-12-2013.  FechaFinal: 3-12-2013. |
| Solicitud | Representa al objeto de negocio que pretende buscar la inclusión para el estudiante. | Solicitud  Comentario: “Ocupo los créditos para la beca”.  Estado: Pendiente. |
| Estudiante | Representa al estudiante que está ejecutando y que le pertenece la solicitud de inclusión. | Estudiante  Nombre: Jose Arnoldo.  PrimerApellido: Segura.  SegundoApellido: Campos.  Carné: 201030612.  Teléfono: 25771122.  Celular: 83370577.  Email: jarnoldo2809@gmail.com. |

### Diagrama de casos de uso 1 iteración



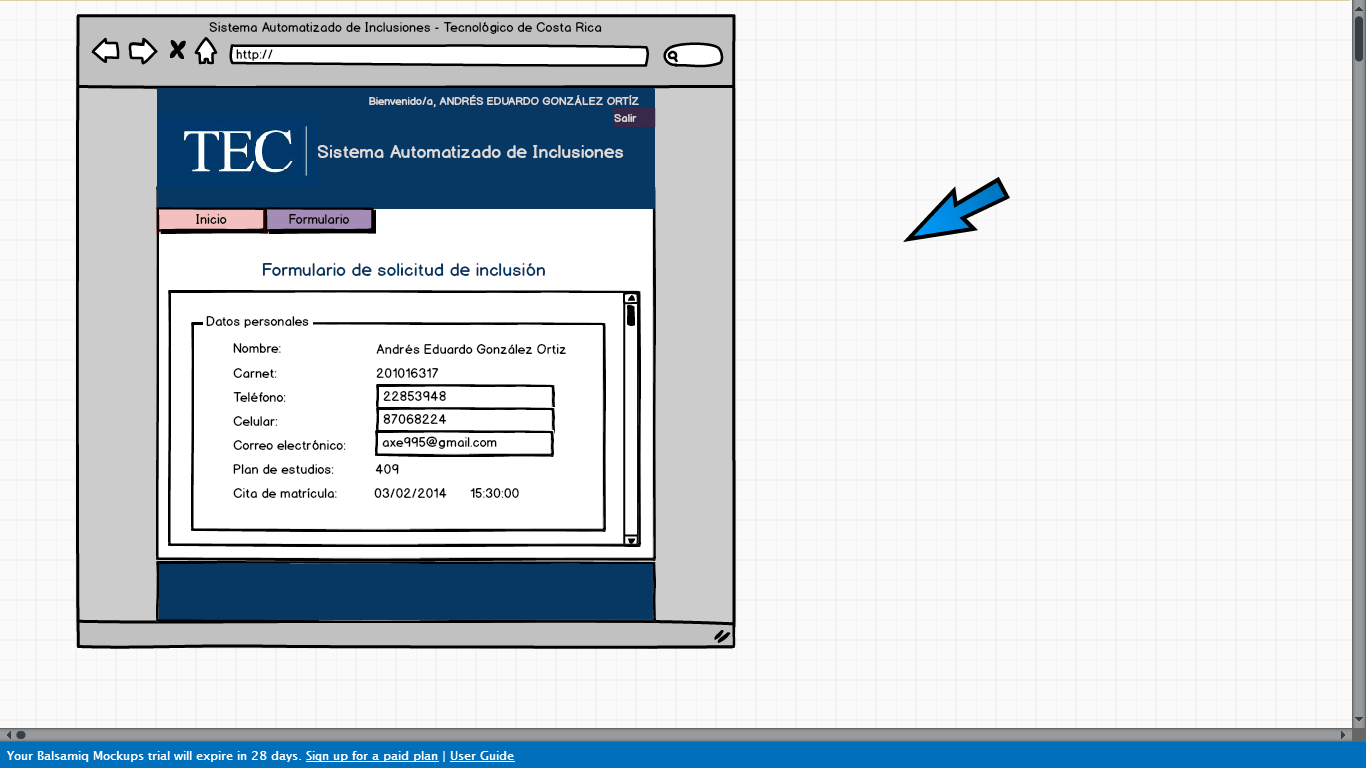
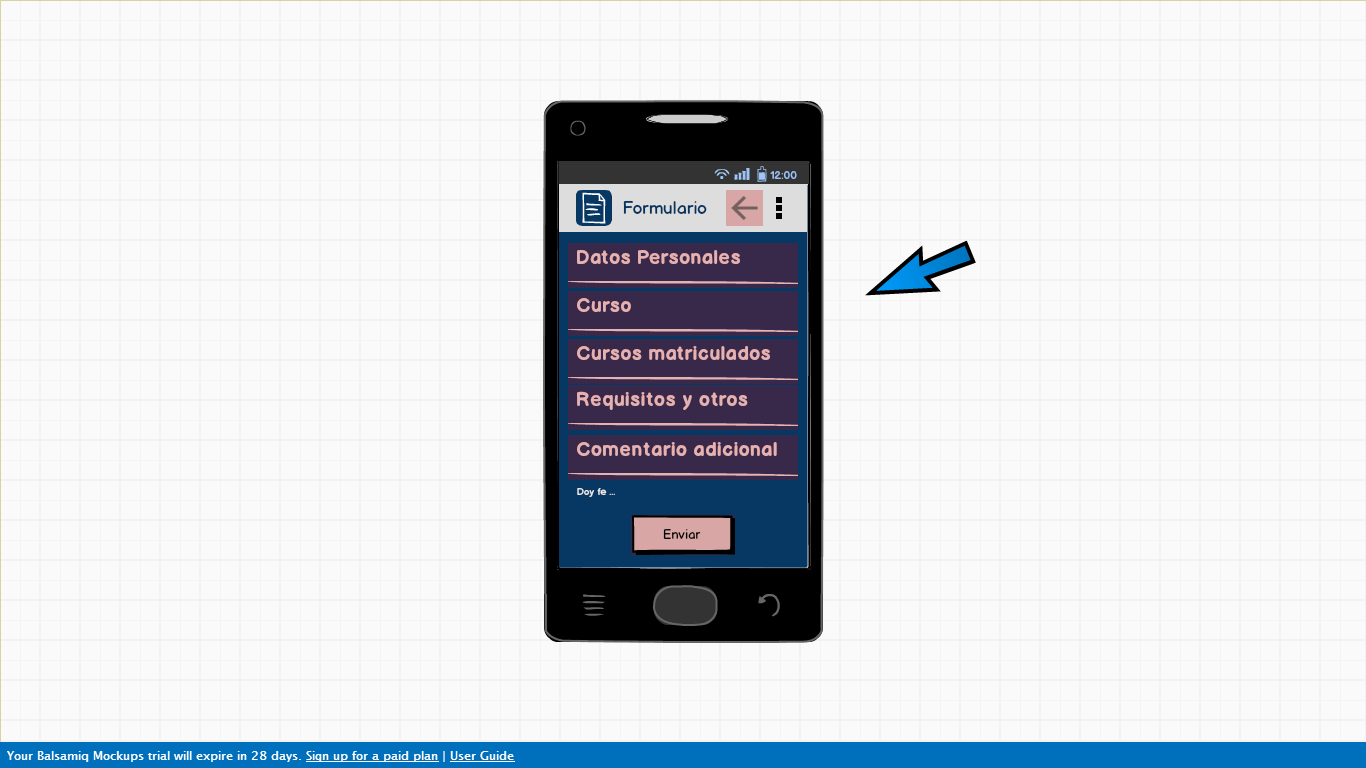
## Descripción detallada cada CU 1 iteración (ordenados forma descend. por prioridad).

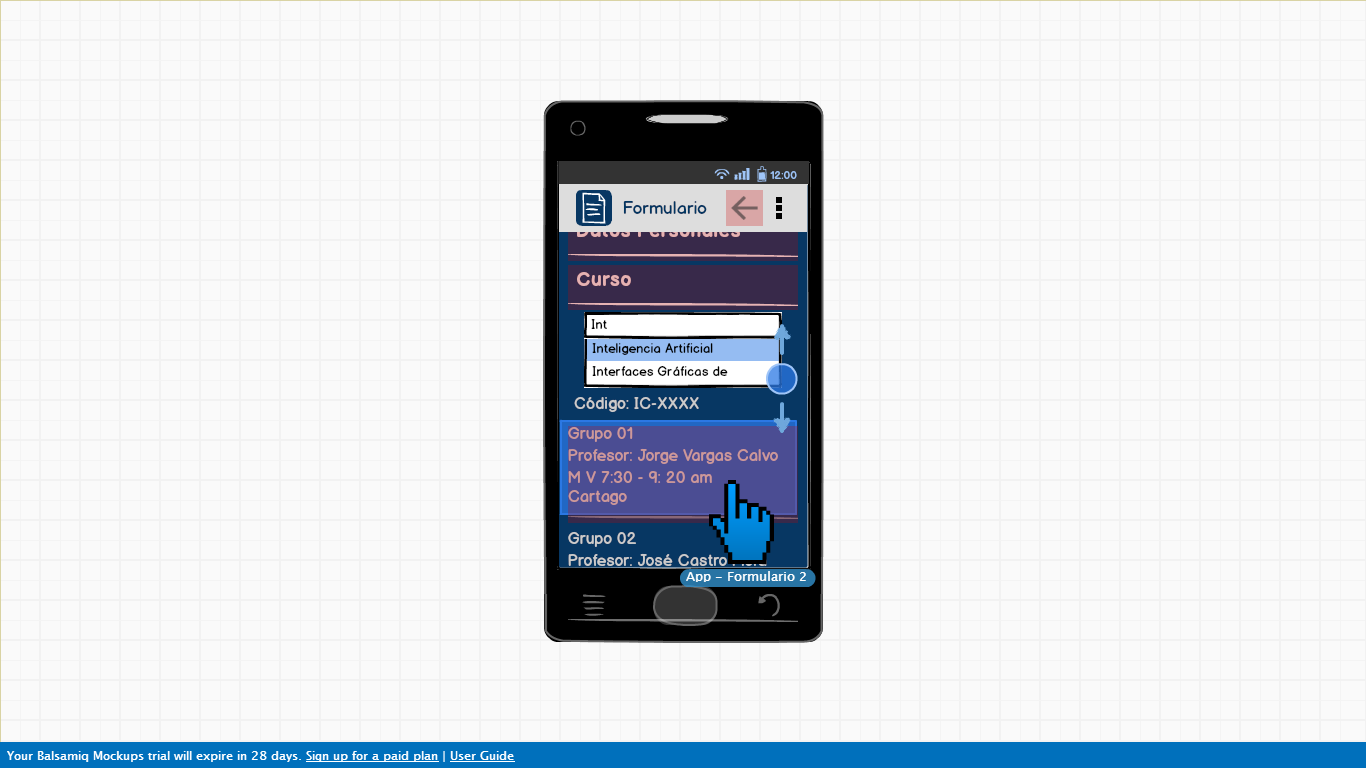
### Caso de uso 1: Crear Solicitud

#### Texto del CU

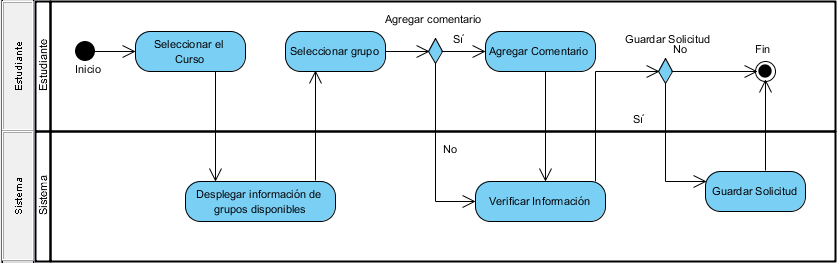


#### Pantalla (s) y/o reporte (s) del CU

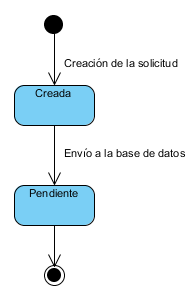




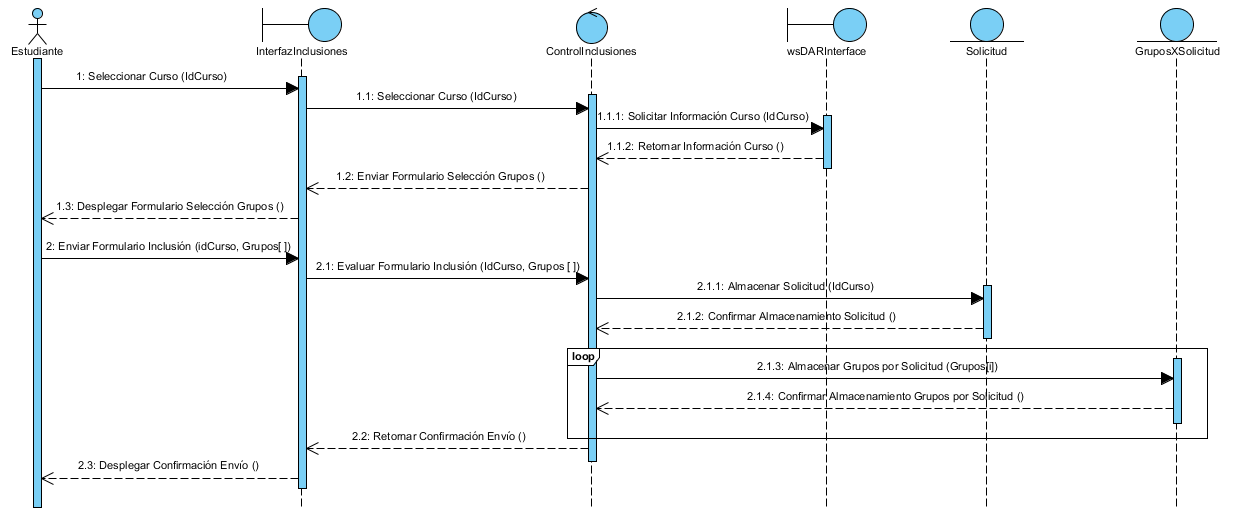
#### Diagrama de actividades del CU



#### Diagrama de estados del CU



#### Diagrama secuencia del sistema (DSS)



#### Contratos de operaciones

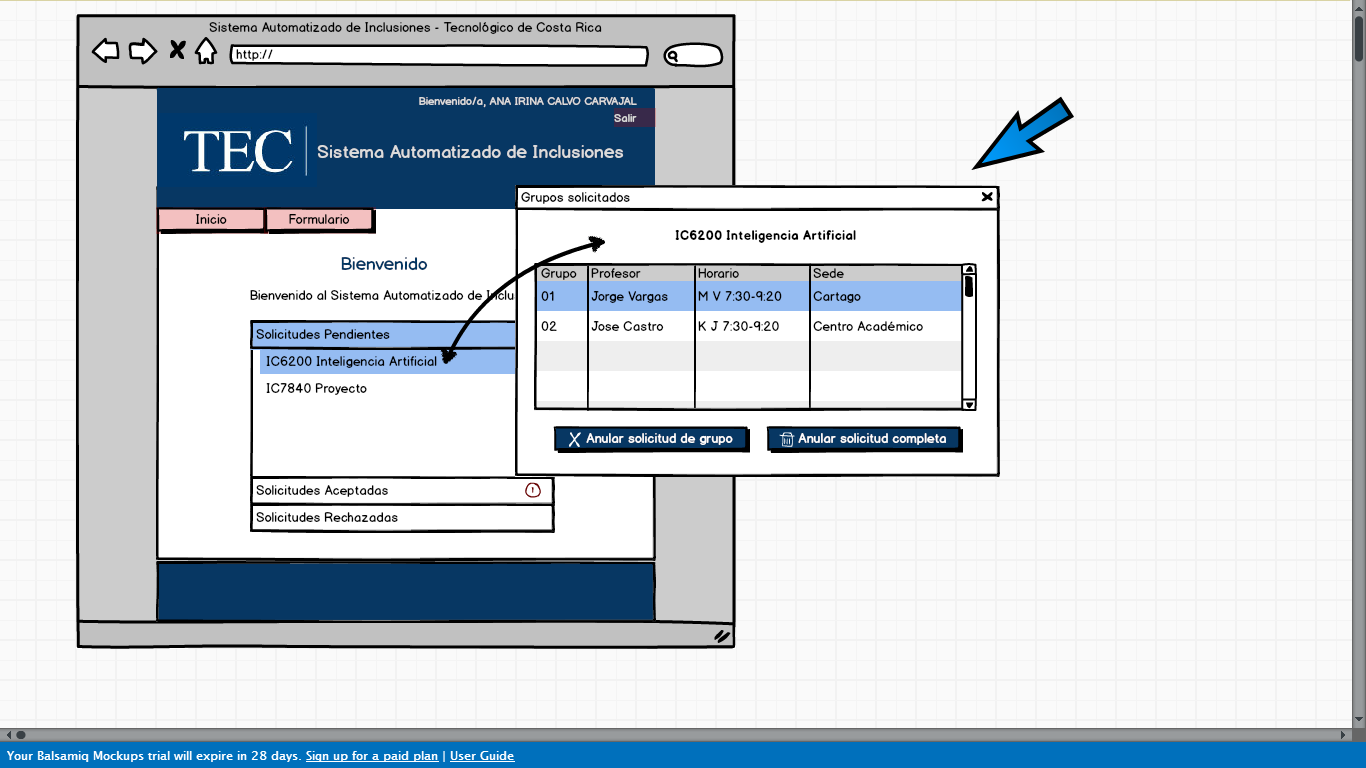
|  |  |
| --- | --- |
| Contrato C01: Solicitar una inclusión | |
| Operación | solicitarInclusión() |
| Parámetros | * nombre : string * carnet : integer * telefono : integer * celular : integer * correoElectronico : integer * planEstudios : integer * citaMatricula : string |
| Referencias | CU-01 |
| Precondiciones | 1. El estudiante debe estar autentificado. 2. El ingreso al formulario de inclusiones del sistema está habilitado por encontrarse el día de hoy dentro del período de recepción de solicitudes. |
| Postcondiciones | 1. Al enviar el formulario la base de datos recibe la información correspondiente, en donde es almacenada. 2. El estudiante es redirigido a una pantalla para que pueda ver |

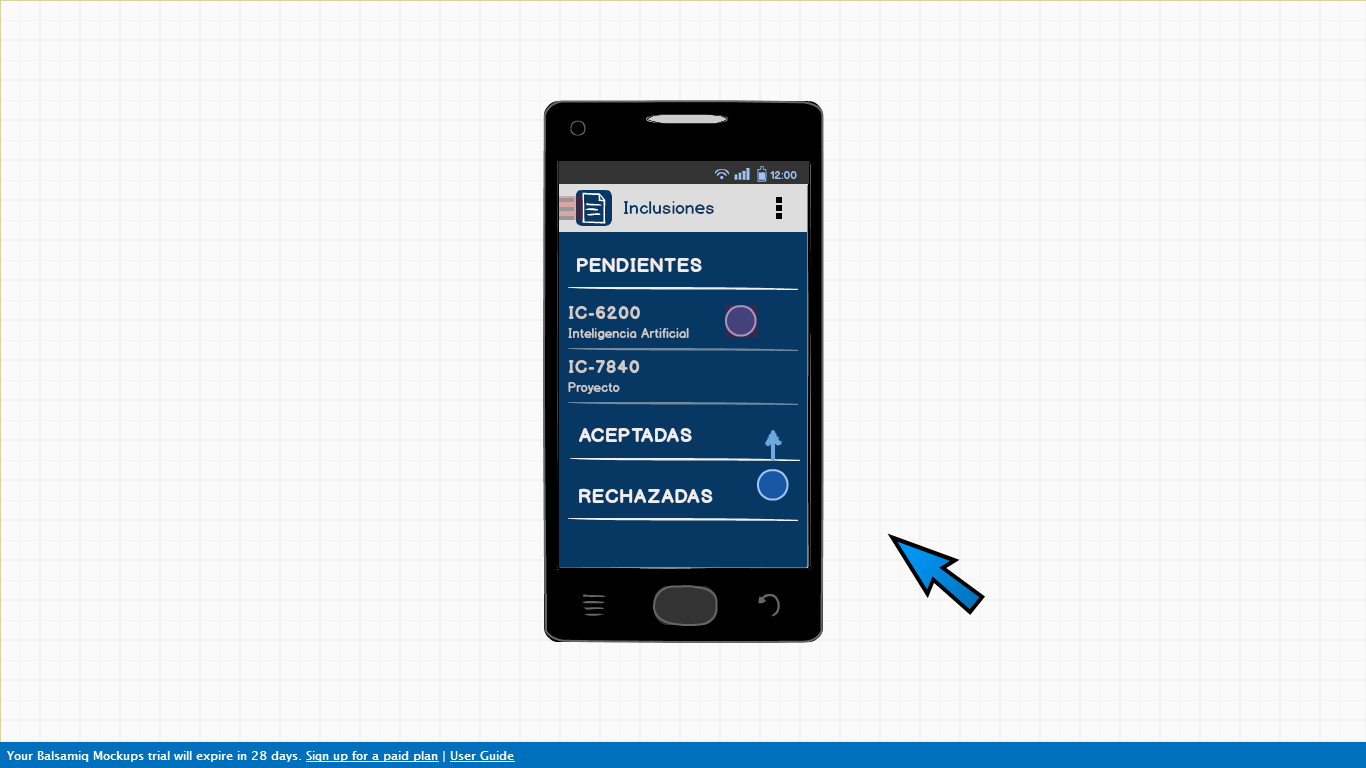
### Caso de uso 2: Revisar Estado de Solicitudes

#### Texto del CU

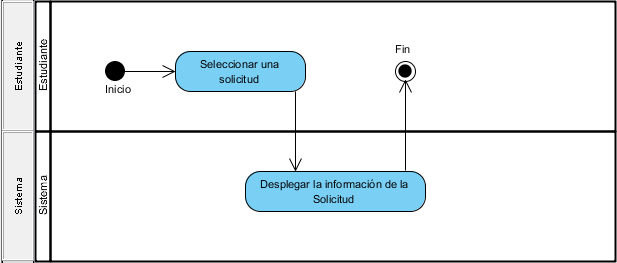


#### Pantalla (s) y/o reporte (s) del CU

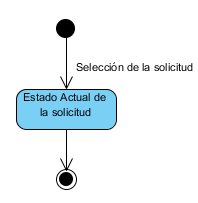




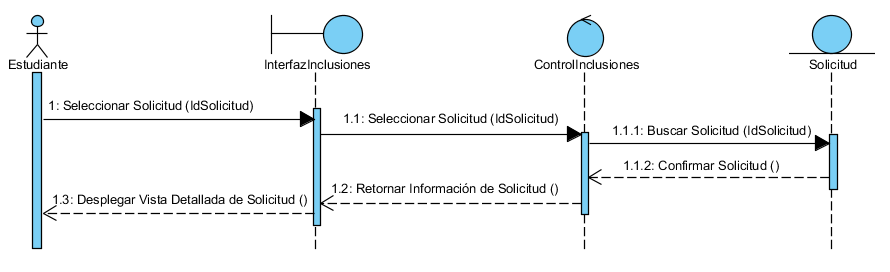
#### Diagrama de actividades del CU



#### Diagrama de estados del CU



#### Diagrama secuencia del sistema (DSS)



#### Contratos de operaciones

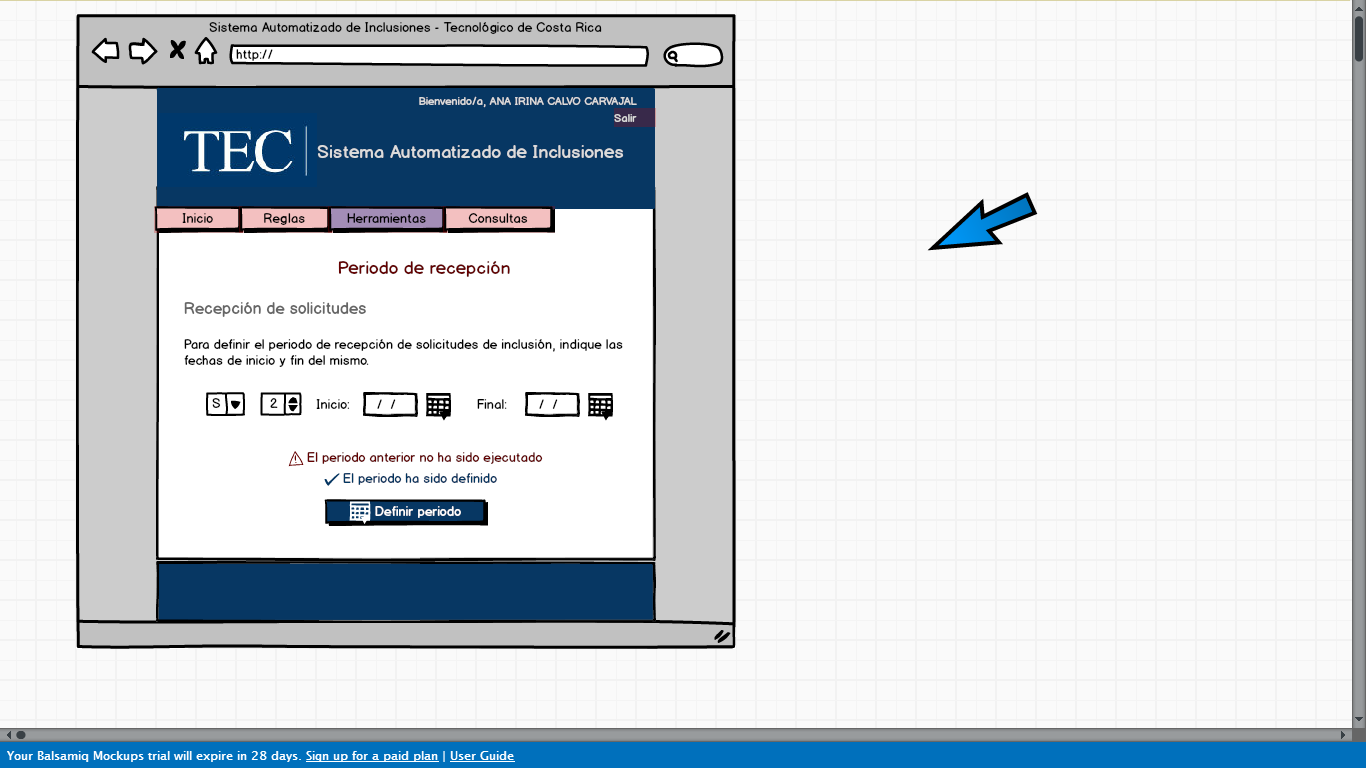
|  |  |
| --- | --- |
| Contrato C02: Revisar estado de solicitudes | |
| Operación | revisarEstadoSolicitudes() |
| Parámetros | * carnetEstudiante : integer |
| Referencias | CU-02 |
| Precondiciones | 1. El estudiante debe estar autenticado. |
| Postcondiciones | 1. Se despliegan las solicitudes asociadas al estudiante, clasificadas en “Pendientes”, “Aceptadas” y “Rechazadas”. |

### Caso de uso 3: Definir Periodo de Recepción de Solicitudes

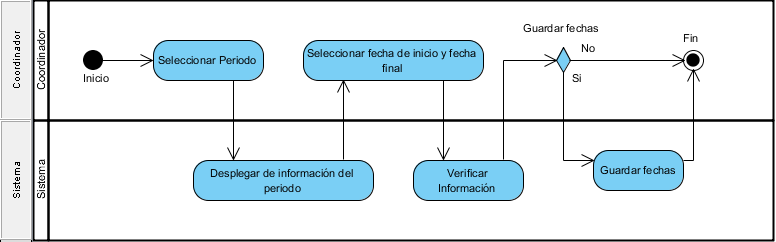
#### Texto del CU

#### 

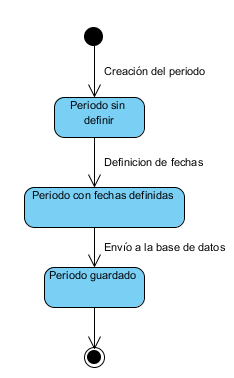
#### Pantalla (s) y/o reporte (s) del CU



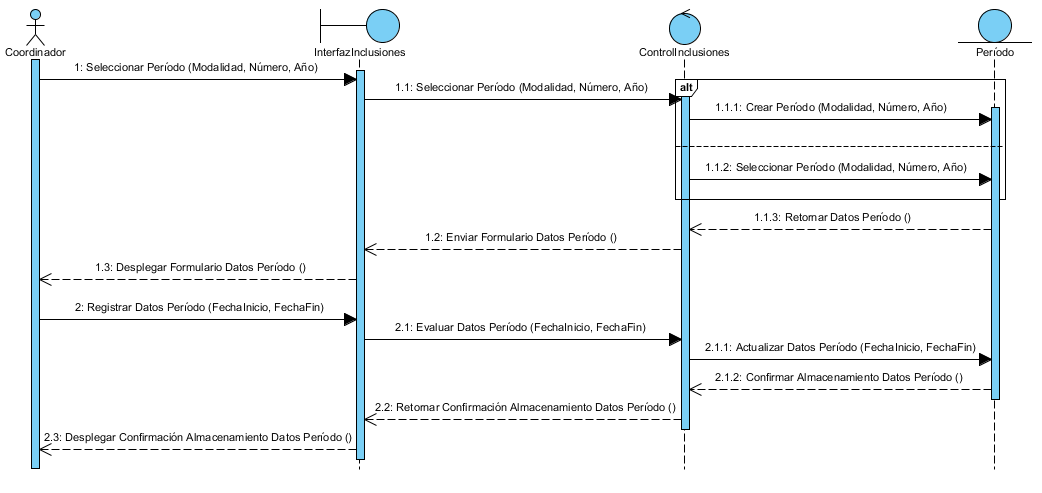
#### Diagrama de actividades del CU



#### Diagrama de estados del CU



#### Diagrama secuencia del sistema (DSS)



#### Contratos de operaciones

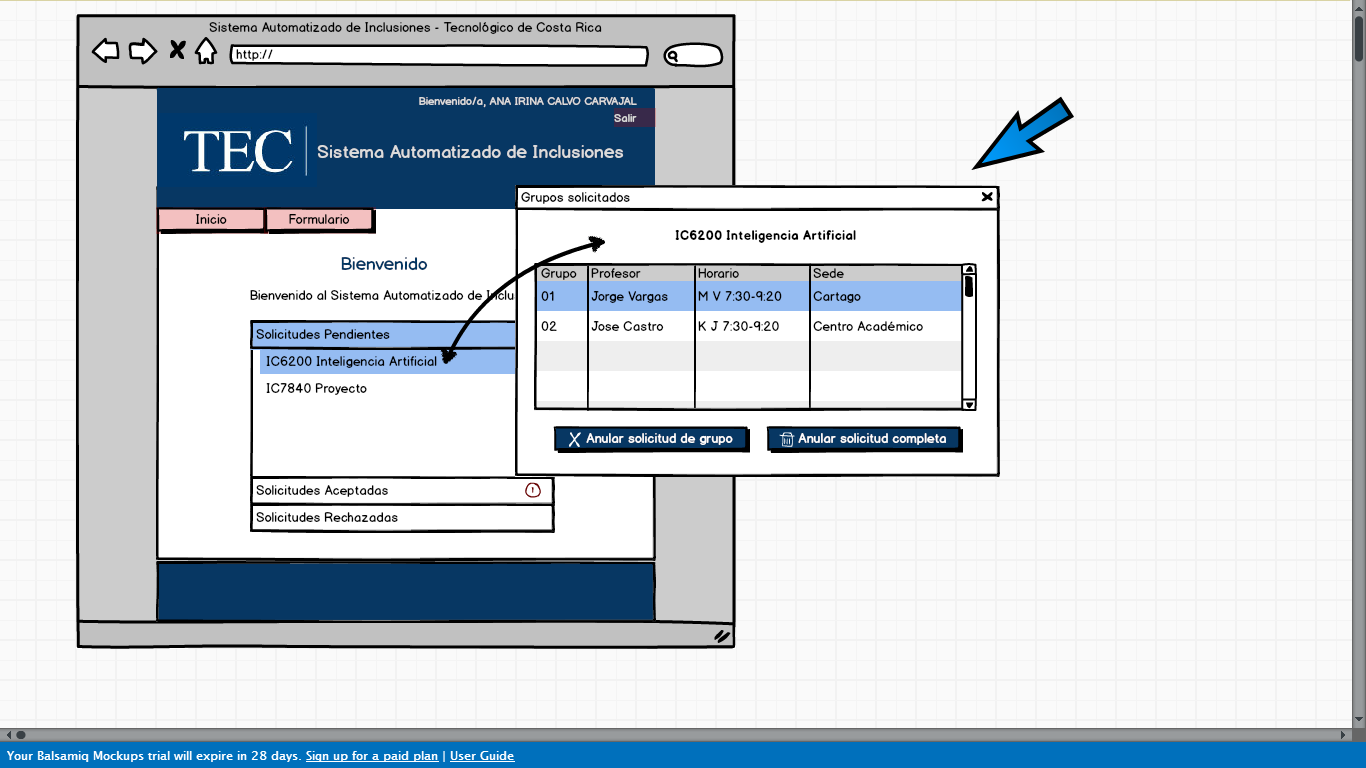
|  |  |
| --- | --- |
| Contrato C03: Definir período de recepción de solicitudes | |
| Operación | definirPeriodo() |
| Parámetros | * fechaInicio : datetime * fechaFinal: datetime |
| Referencias | CU-03 |
| Precondiciones | 1. El coordinador debe estar autenticado. 2. El proceso de asignación no ha sido ejecutado. |
| Postcondiciones | 1. Se guarda un nuevo período en la base de datos del sistema en caso de no existir, caso contrario solamente actualiza el rango de fechas. |

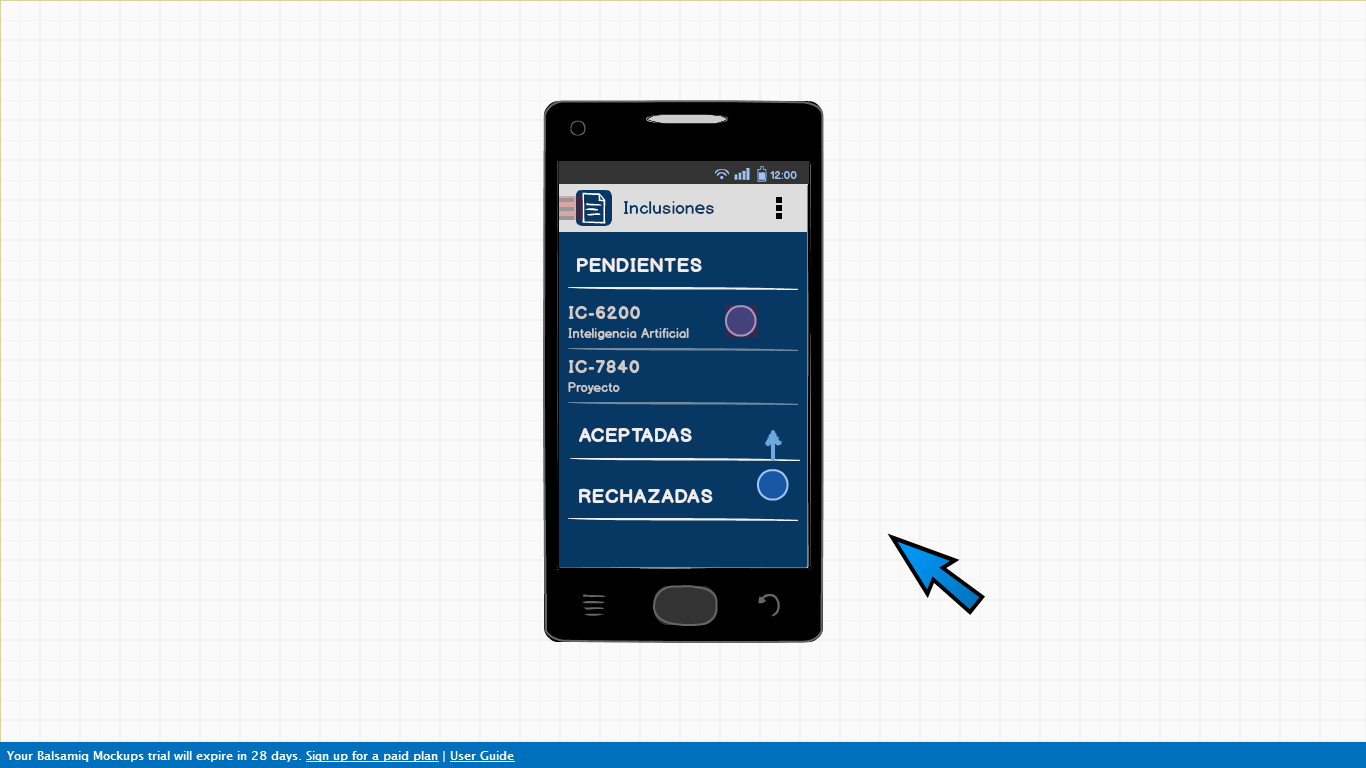
### Caso de uso 4: Modificar una solicitud

#### Texto del CU

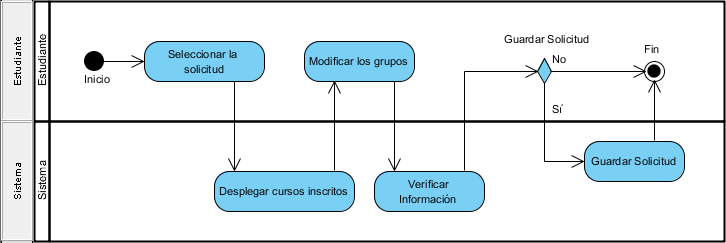


#### Pantalla (s) y/o reporte (s) del CU

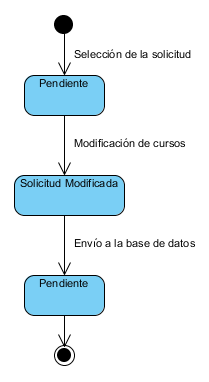




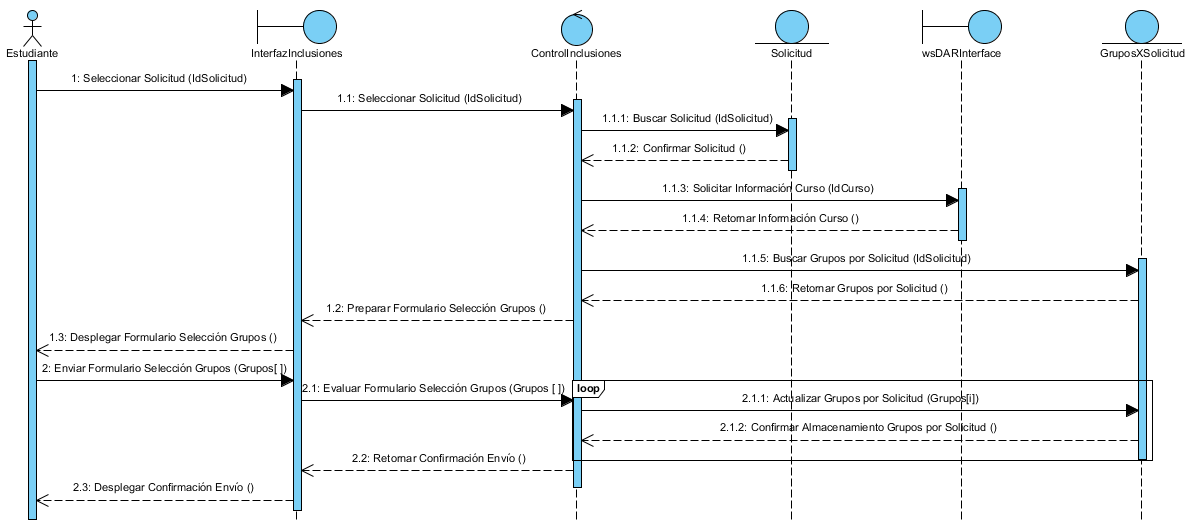
#### Diagrama de actividades del CU



#### Diagrama de estados del CU



#### Diagrama secuencia del sistema (DSS)



#### Contratos de operaciones

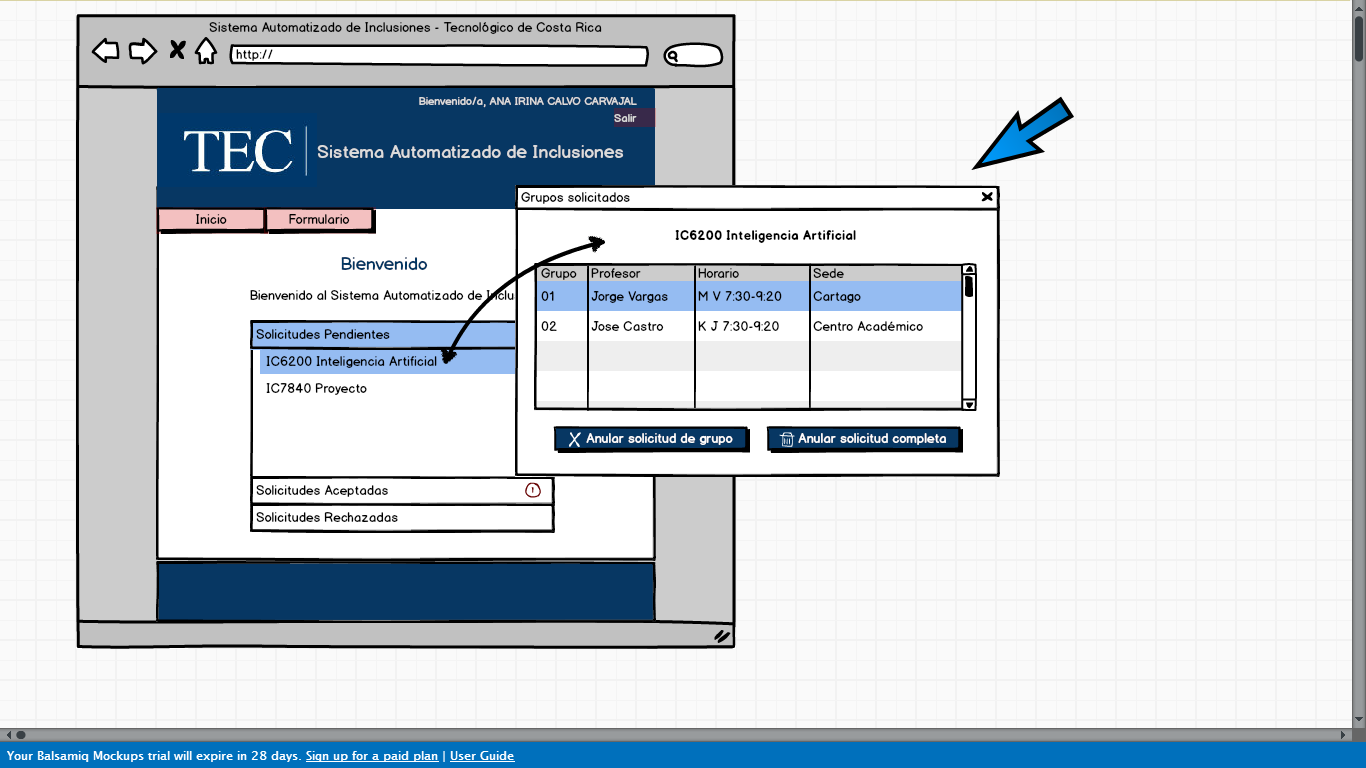
|  |  |
| --- | --- |
| Contrato C04: Modificar una solicitud | |
| Operación | modificarSolicitud() |
| Parámetros | * idSolicitud : integer * idGrupo : integer |
| Referencias | CU-04 |
| Precondiciones | 1. El estudiante debe estar autenticado. 2. El ingreso al formulario de inclusiones del sistema está habilitado por encontrarse el día de hoy dentro del período de recepción de solicitudes. 3. Solo puede ser modificada una solicitud en estado “Pendiente”. |
| Postcondiciones | 1. Se actualiza la solicitud en la base de datos del sistema. |

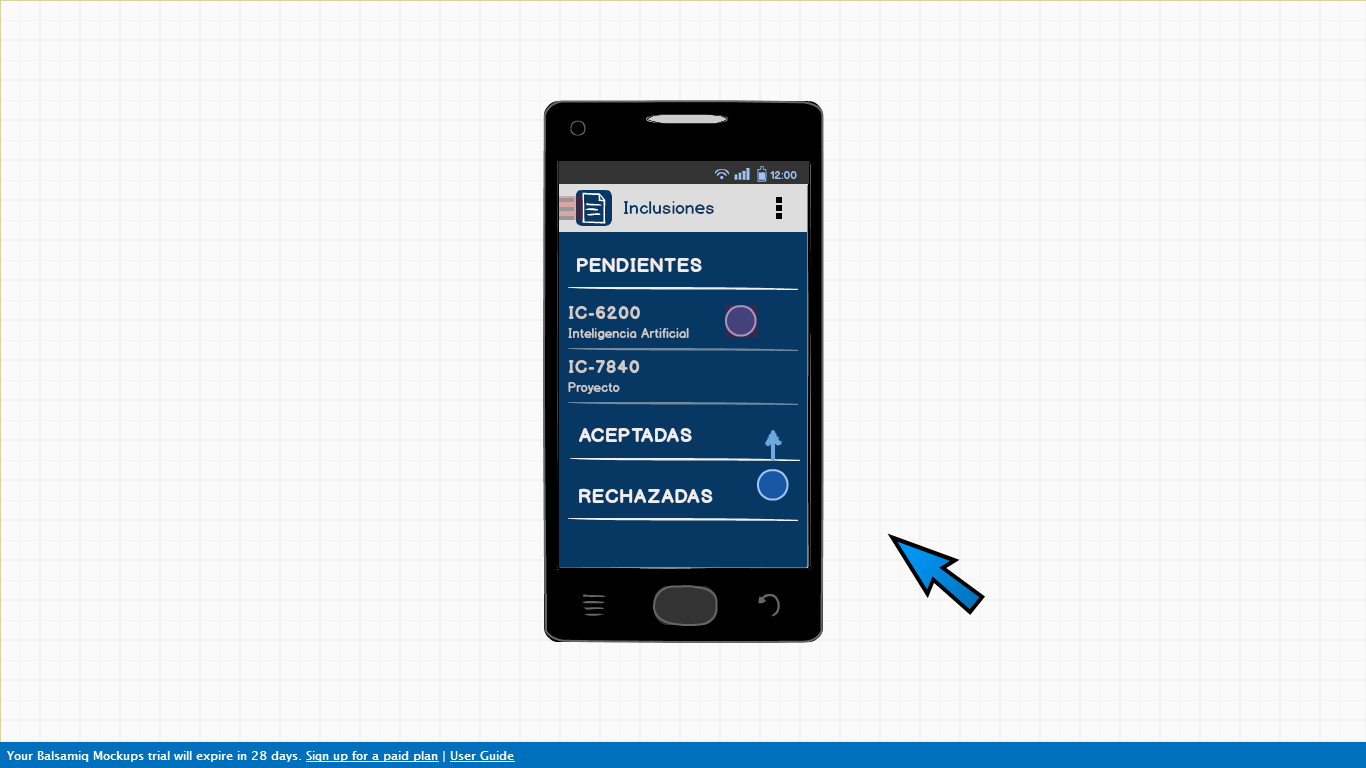
### Caso de uso 5: Anular Solicitud

#### Texto del CU

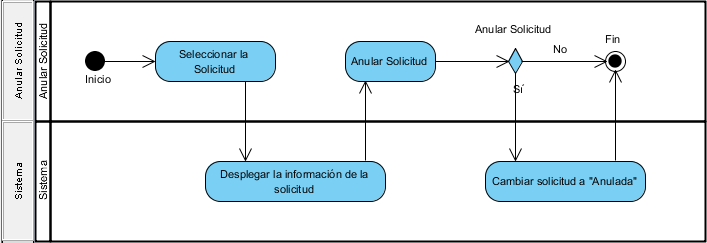


#### Pantalla (s) y/o reporte (s) del CU

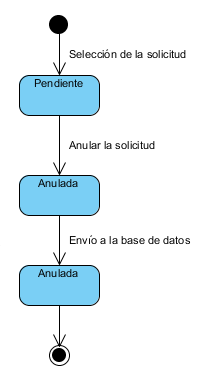




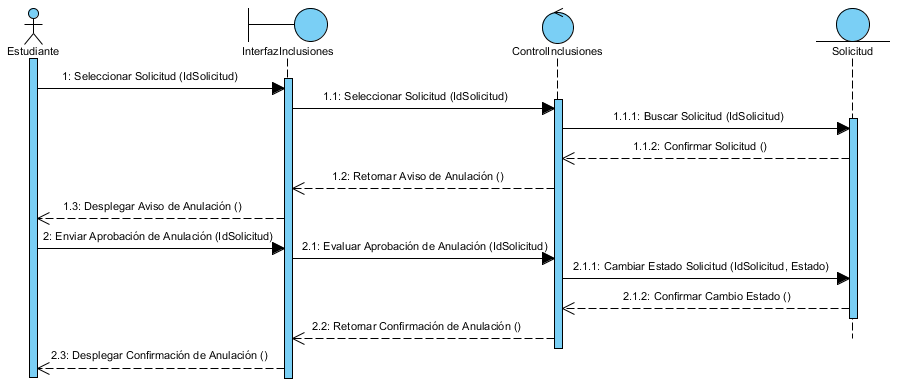
#### Diagrama de actividades del CU



#### Diagrama de estados del CU



#### Diagrama secuencia del sistema (DSS)



#### Contratos de operaciones

|  |  |
| --- | --- |
| Contrato C05: Anular una solicitud | |
| Operación | anularSolicitud() |
| Parámetros | * idSolicitud : integer |
| Referencias | CU-05 |
| Precondiciones | 1. El estudiante debe estar autenticado. 2. El ingreso al formulario de inclusiones del sistema está habilitado por encontrarse el día de hoy dentro del período de recepción de solicitudes. 3. Solo puede ser modificada una solicitud en estado “Pendiente”. |
| Postcondiciones | 1. Se cambia el estado de la solicitud a “Anulada” en la base de datos del sistema. |

# Requerimientos no funcionales

## 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:

* 1. Minimalista
  2. Plana
  3. Amigable
  4. Obvia

### Interfaz web del usuario

#### 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.

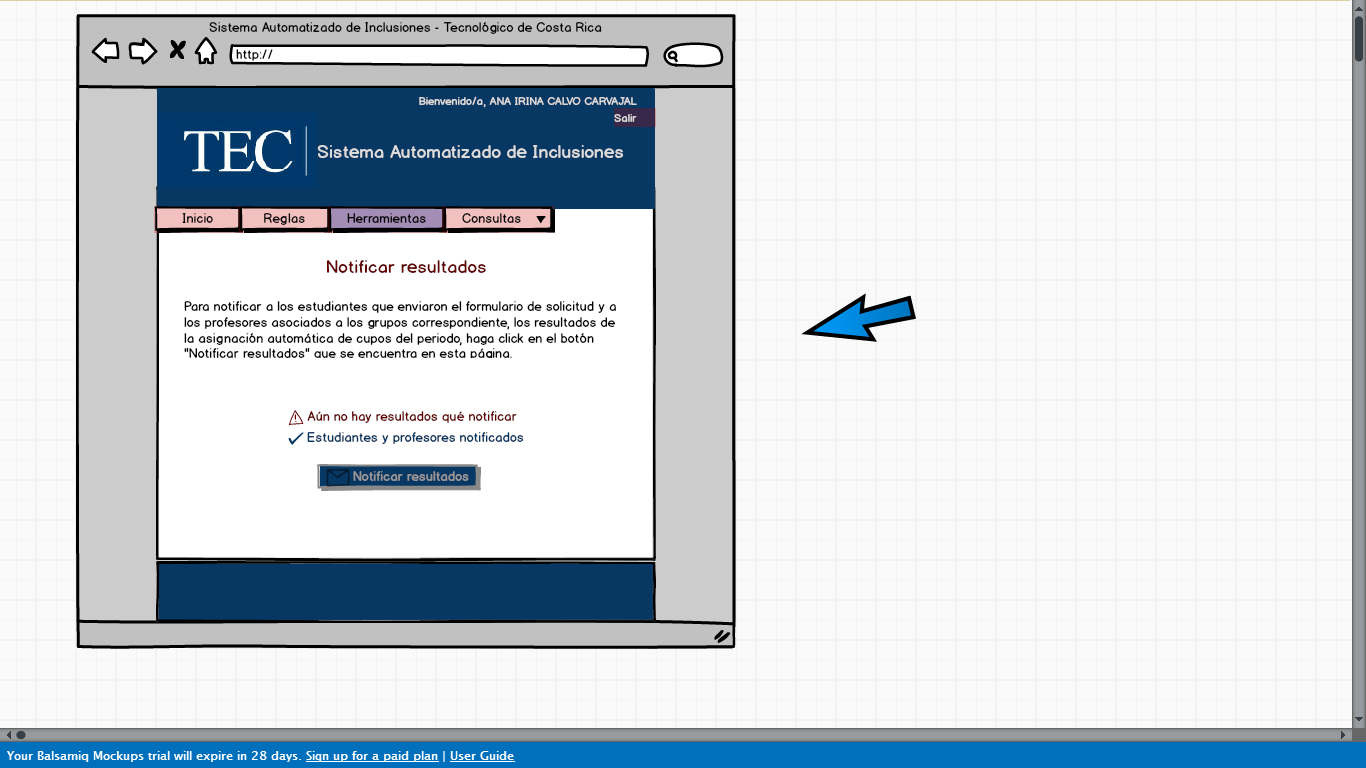
#### Estándar de tipos de fuente

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

#### Esquema de colores

#### Ejemplo de pantalla



### Interfaz móvil del usuario

#### Descripción

La interfaz gráfica para la aplicación de Android se basa en los patrones de diseño gráfico y estilos oficiales de Android, por lo que la aplicación tiene el aspecto más parecido posible a otras aplicaciones que el estudiante puede haber utilizado antes.

#### Estándar de tipos de fuente

A continuación se definen los tipos de fuente para la aplicación de Android, que serían básicamente correspondientes a la aplicación web, pero utilizando una fuente Roboto, la original de Android, como tipo Sans.

* + Título 1: Serif16pt negrita
  + Título 2: Roboto 12pt negrita
  + Título 3: Roboto 10pt negrita
  + Cuerpo: Roboto 10pt
  + Notas: Roboto 10pt
  + Botón: Roboto 12pt

#### Esquema de colores

#### C:\Users\Irina\Dropbox\Proyecto\Iteración I\Prototipo\Screenshots\Screenshot 2013-12-17 18.53.46.pngEjemplo de pantalla

## RNF-02: Organizacionales

### Patrón documentación

Durante las tres iteraciones el proyecto se generará documentación que facilite el desarrollo del software de manera que tanto el desarrollador como el cliente se entiendan. Estos documentos son los documentos de Especificación de Requerimientos de Software y el documento de Arquitectura de Software.

Además, por cada iteración se entregará un manual de usuario el cual explicará paso a paso como realizar cada una de las funciones desarrolladas en cada iteración.

### Entregas

1. Primera Iteración:
   1. Fecha de entrega:6-1-2014.
   2. Entrega al profesor Jaime Solano.
   3. Medio digital y presentación de la iteración.
2. Segunda Iteración:
   1. Fecha de entrega:6-1-2014.
   2. Entrega al profesor Jaime Solano.
   3. Medio digital y presentación de la iteración.
3. Entrega al cliente:
   1. Fecha: 10-1-2014.
   2. Entrega a la profesora Adriana Álvarez.
   3. Presentación del sistema.
4. Tercera Iteración:
   1. Fecha de entrega:13-1-2014.
   2. Entrega al profesor Jaime Solano.
   3. Medio digital y presentación de la iteración.
5. Instalación del sistema:
   1. Fecha de instalación: Por definir, en un rango entre el 13 y el 26 de enero del 2014.
   2. Entrega a la profesora Adriana Álvarez.
   3. Instalación del sistema.

### Implementación

* La implementación será llevada a cabo en los servidores del departamento de Soporte de la Escuela de Ingeniería en Computación.
* El framework será el 4.0.
* El servidor de base de datos debe ser SQL Server 2008 R2.

## RNF-03: Externos

### Patrón Interface e interacción entre sistemas

Este patrón se caracteriza porque el sistema va a estar fuertemente ligado a los servicios web del Departamento de Admision y Registro,es por esto que las dependencias de interfaces del sistema son asociadas con el servicio wsDAR que es proporcionado por el Centro de Cómputo.

Este sistema, además, no publica ninguna interfaz hacia otro subsistema.

### Legales

#### Propiedad del producto desarrollado

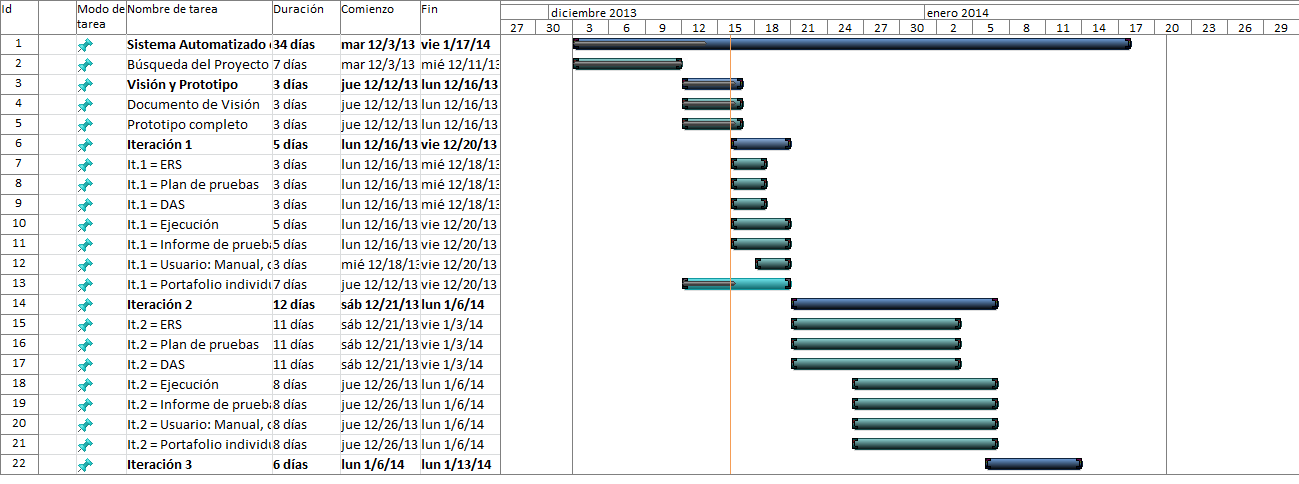
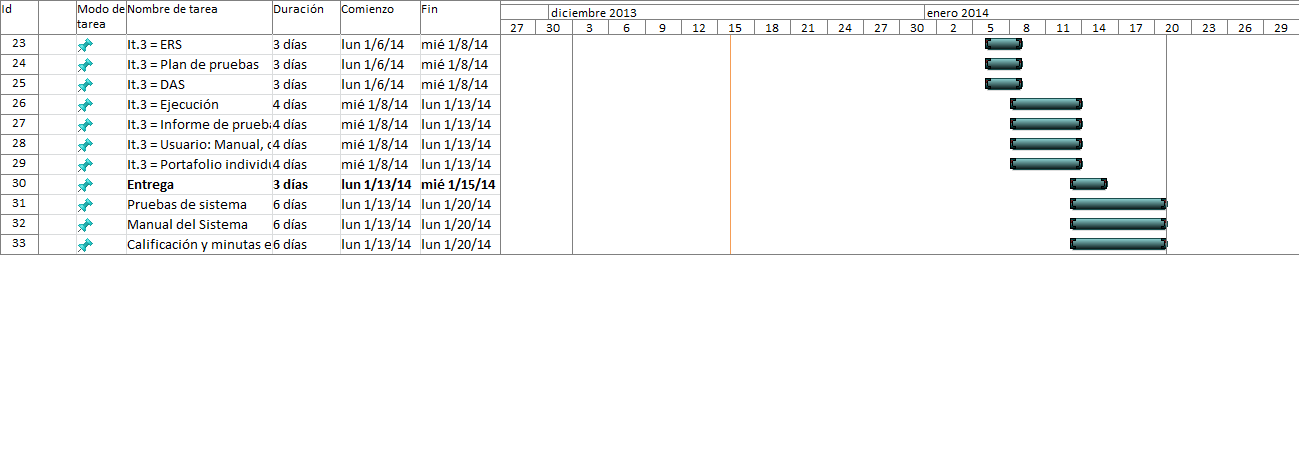
El sistema desarrollado por el equipo de trabajo no tendrá ningún costo ni derechos de autor relacionados con los desarrolladores, y este producto será de uso exclusivo por el Instituto Tecnológico de Costa Rica, dándole título de propietario del sistema.

#### Licenciamiento de aplicaciones de desarrollo y ambientes de ejecución

Las herramientas de desarrollo y los ambientes de ejecución mencionados anteriormente serán provistos por la Escuela de Ingeniería en Computación la cual posee licencias tanto como para los desarrolladores que llevarán a cabo el proyecto y el ambiente de ejecución.

# Apéndices

## Plan del proyecto en sólo 1 pág (Gantt en Project) (Desglose semanal y no diario)



## Glosario de términos y abreviaturas (orden alfabético)

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

## Lista de riesgos (orden descendente de prioridad) Incluir nombre, descrip, prioridad, responsable, etc

|  |  |  |  |
| --- | --- | --- | --- |
| *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. |

## 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.alvaresf@gmail.com |
| Teléfono |  |

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

### Estándar de programación código fuente

#### 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).*

#### 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.

#### 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

#### 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.

##### 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();

....

}

##### 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.

##### 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.

##### Estándar de la clase de datos

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

#### Interfaces

Los nombres de las interfaces deben ser adjetivos.

##### Interface Web

WIConPalabrasInternasPrimerLetraEnMayúscula

##### Interface Local

IConPalabrasInternasPrimerLetraEnMayúscula

##### 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();

}

#### 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”

##### 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.

#### 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).

##### 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 |

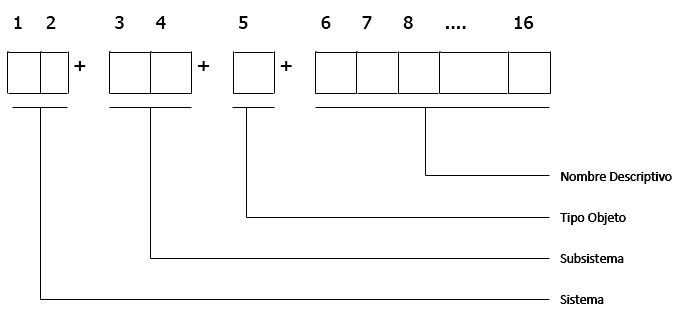
#### Properties

* 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.

### 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:



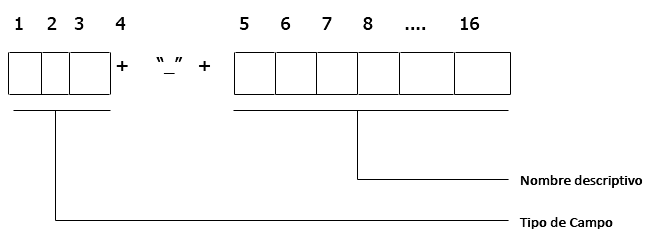
* Sistema: Corresponde a una tira de dos caracteres que identifica el sistema al que pertenece el objeto.
* Subsistema: Corresponde a una tira de dos caracteres que identifica el subsistema al que pertenece el objeto.
* Tipo Objeto: Si es una tabla puede ser “F” (datos fijos) o “T” (temporal). Si es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

#### 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:

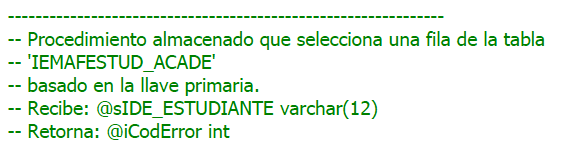
* pr\_IEMAFESTUD\_ACADE\_Insertar

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>