

DESARROLLO DEL NUEVO SISTEMA DE INFORMACIÓN PARA  
LA INSCRIPCIÓN, ADJUDICACIÓN, ADMINISTRACIÓN Y  
CONTROL DEL SERVICIO DE COMEDORES DE BIENESTAR  
UNIVERSITARIO DE LA UNIVERSIDAD INDUSTRIAL DE  
SANTANDER

FIDEL DAVID JIMÉNEZ MARTÍNEZ

JUAN MARTÍNEZ MARTÍNEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICOMECAÑICAS

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

2011

DESARROLLO DEL NUEVO SISTEMA DE INFORMACIÓN PARA  
LA INSCRIPCIÓN, ADJUDICACIÓN, ADMINISTRACIÓN Y  
CONTROL DEL SERVICIO DE COMEDORES DE BIENESTAR  
UNIVERSITARIO DE LA UNIVERSIDAD INDUSTRIAL DE  
SANTANDER

FIDEL DAVID JIMÉNEZ MARTÍNEZ

JUAN MARTÍNEZ MARTÍNEZ

trabajo de grado presentado como requisito para optar al título de  
Ingeniero de Sistemas

DIRECTOR

ingeniero ENRIQUE TORRES LÓPEZ

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE INGENIERÍAS FISICOMECAÑICAS

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

2011

amgno p̄oyin cyen weryic  
p̄r̄p̄er w̄l̄t̄er̄o c̄ w̄m̄er̄

fidel david

*al profesor Albus Percival Dumbledore*

juan

# Agradecimientos

fidel david

A mis padres, por su enorme tarea.

A los ingenieros Jacksson González y Enrique Torres, por su paciencia, orientación y ejemplo de profesionalismo.

A los profesores Fabio Reyes y Fernando Rojas, por ser los mejores, por ser una luz en nuestra oscura escuela.

juan

A mis papás por tratar de hacerme una persona amorosa y sensata. A los ingenieros de la División de Servicios de Información de la UIS que nos ayudaron en este proyecto. A los hombres de ciencia que no se conforman con la superstición.

# Tabla de contenido

Tabla de contenido.....	9
<b>Resumen.....</b>	<b>14</b>
Introducción .....	15
Presentación del proyecto .....	17
1.1 Descripción del proyecto .....	17
1.1.1 Objetivo general .....	17
1.1.2 Objetivos específicos.....	17
1.2 Justificación .....	19
1.2.1 Antecedentes y descripción del problema.....	19
1.2.2 Impacto.....	19
1.2.3 Viabilidad.....	20
Marco teórico .....	21
2.1 Tecnología usada para el proyecto.....	21
2.1.1 Entorno Web .....	21
2.1.2 Arquitectura multicapa .....	21
2.1.3 Entorno de desarrollo .....	23
2.1.4 Herramientas de desarrollo.....	25
2.2 Sistemas de información.....	26
2.2.1 Definición.....	26
2.2.2 Ciclo de vida de un sistema de información.....	26
2.3 Generalidades del sistema de comedores.....	28
2.3.1 Definiciones y siglas.....	28

2.3.2 Descripción general.....	29
2.3.3 Documentación.....	30
2.4 Metodología de desarrollo .....	30
2.4.1 Descripción de la metodología .....	30
2.4.2 Razones para la elección de la metodología.....	32
Desarrollo del sistema .....	33
3.1 Plan de trabajo.....	33
3.1.1 Acoplamiento y capacitación .....	33
3.1.2 Análisis y diseño global del nuevo sistema.....	33
3.1.3 Diseño e implementación de los módulos .....	34
3.1.4 Puesta en marcha, y entrega del producto final .....	35
3.2 Acoplamiento y capacitación.....	35
3.3 Análisis y diseño global del nuevo sistema.....	36
3.3.1 Análisis y conocimiento de los requisitos del sistema .....	36
3.3.2 Planificación de los módulos.....	36
3.3.3 elaboración de los diagramas de casos de uso .....	38
3.3.4 Modelo de datos .....	45
3.3.5 Diseño del modelo de entidad-relación.....	46
3.3.6 Diseño del prototipo no funcional .....	59
3.3 Diseño e implementación de los módulos .....	68
3.4.1 Componentes reutilizables.....	69
3.4.2 Inscripciones .....	75
3.4.3 Adjudicación .....	102
3.4.4 Control de la asistencia .....	108
3.4.5 Administración .....	117
3.4.6 Esquema de seguridad.....	139
Conclusiones .....	143
Recomendaciones .....	145
Anexos.....	147
Bibliografía .....	148

# Lista de figuras

3.4.6 Esquema de seguridad.....	139
figura 01 - arquitectura multicapa .....	22
figura 02 - estructura de la aplicación .....	25
figura 03 - contextos del seam .....	25
figura 04 – descripción de la metodología de desarrollo .....	31
figura 05 – diagrama de casos de uso del módulo de inscripciones .....	42
figura 06 – diagrama de casos de uso del módulo de adjudicación .....	43
figura 07 – diagrama de casos de uso del módulo de administración .....	44
figura 08 – diagrama de casos de uso del módulo de control .....	45
figura 09 – menú del prototipo no funcional .....	60
figura 10 – prototipo caso de uso ‘crear inscripción’ .....	61
figura 11 – prototipo del caso de uso ‘búsqueda de inscrito’ .....	62
figura 12 – prototipo del caso de uso ‘modificar estado de servicios solicitados’ ...	63
Figura 13 – prototipo del caso de uso ‘gestionar cupos de servicio’ .....	64
figura 14 – prototipo del caso de uso ‘gestionar cupos de servicio’ .....	65
figura 15 – prototipo del caso de uso ‘gestionar servicios servidos’ .....	66
figura 16 – prototipo del caso de uso ‘hacer mantenimiento de tipos de usuario’ .	67
figura 17 – submenús del sistema.....	68
figura 18 – captura de interfaz de ‘búsqueda de inscrito’ por documento.....	71
figura 19 - captura de interfaz de ‘búsqueda de inscrito’ por nombre .....	72
figura 20 – captura de interfaz de ‘búsqueda de usuario’ .....	74
figura 21 – captura de interfaz de ‘búsqueda de usuario’ con resultados .....	75
figura 22 – captura de interfaz de ‘crear inscripción’ con un documento que no pertenece a ningún estudiante .....	80
figura 23 – captura de la interfaz de ‘crear inscripción’ con el mensaje de validación de requisitos.....	81
figura 24 – captura de interfaz de ‘modificar tipo de usuario’ .....	84
figura 25 – captura de la interfaz de ‘modificar inscripción’ .....	86
figura 26 – captura de interfaz de ‘modificar inscripción’ con el mensaje de validación .....	87
figura 27 – captura de la interfaz de ‘gestionar recibos’ .....	90
figura 28– captura interfaz de ‘ver recibo de pago’ .....	91
figura 29 – captura de interfaz de ‘crear aplazamiento’ .....	93
figura 30 – captura de interfaz de ‘crear aplazamiento’ con mensaje de validación .....	94
figura 31 – captura de interfaz de ‘ver aplazamientos’ .....	95
figura 32 – captura de interfaz ‘gestionar requisitos exceptuados’ .....	97
figura 34 – captura de interfaz de ‘ver lista de requisitos exceptuados’ .....	99
figura 35 – captura de interfaz de ‘administrar cupos de prueba de servicios’ .....	105
figura 36 – captura de interfaz de ‘adjudicar cupos de servicios iniciales’ .....	106

figura 37 – captura de interfaz de ‘modificar cupos disponibles’ .....	107
figura 38 – captura de interfaz de ‘iniciar una sesión de servicio’ .....	111
figura 39 – captura de interfaz de ‘crear asistencia a servicio’ .....	112
figura 40 – captura de interfaz de ‘crear asistencia a servicio’ sin el servicio adjudicado .....	113
figura 41 – captura de interfaz de ‘crear asistencia a servicio’ sin el servicio habilitado .....	114
figura 42 – captura de interfaz de ‘crear asistencia a servicio’ con paso libre .....	115
figura 43 – captura de interfaz de ‘abrir sesión de servicio’ .....	116
figura 44 – captura de interfaz de ‘modificar tipo de usuario’ .....	122
figura 45 – captura de interfaz de ‘ver detalles de tipo de usuario’ .....	124
figura 46 – captura de interfaz de ‘modificar un tipo de usuario’ .....	125
figura 47 – captura de la interfaz de crear periodos de pago .....	131
figura 48 – captura del mensaje de validación en ‘crear periodo’ .....	132
figura 49 – captura de interfaz de ‘ver periodo’ .....	133
figura 50 – captura de interfaz de ‘modificar periodo’ .....	134
figura 51 – captura de interfaz de ‘eliminar periodo’ .....	135
tabla 01 – tabla ‘anos_semestres_academicos’ .....	47
tabla 02 – tabla ‘aplazamientos_servicios’ .....	48
tabla 03 – tabla ‘asistencias’ .....	48
tabla 04 – tabla ‘cambios_cumplimiento_requisitos’ .....	49
tabla 05 – tabla ‘criterios’ .....	50
tabla 06 – tabla ‘cupos_servicios’ .....	50
tabla 07 – tabla ‘detalles_recibos’ .....	51
tabla 08 – tabla ‘grupos_especiales’ .....	51
tabla 09 – tabla ‘inscripciones’ .....	52
tabla 10 – tabla ‘inscritos_grupos_especiales’ .....	52
tabla 11 – tabla ‘parametros_comedores’ .....	53
tabla 12 – tabla ‘periodos’ .....	53
tabla 13 – tabla ‘recibos’ .....	54
tabla 14 – tabla ‘requisitos’ .....	54
tabla 15 – tabla ‘requisitos_exceptuados’ .....	55
tabla 16 – tabla ‘sanciones’ .....	56
tabla 17 – tabla ‘servicios’ .....	56
tabla 18 – tabla ‘servicios_servidos’ .....	57
tabla 19 – tabla ‘servicios_solicitados’ .....	58
tabla 20 – tabla ‘tipos_usuario’ .....	59
tabla 21 – caso de uso ‘búsqueda de inscrito’ .....	70
tabla 22 – caso de uso ‘búsqueda de usuario’ .....	73
tabla 23 – caso de uso ‘gestionar inscripciones’ .....	77
tabla 24 – caso de uso ‘crear inscripción’ .....	78
tabla 25 – caso de uso ‘modificar inscripción’ .....	82
tabla 26 – caso de uso ‘modificar tipo de usuario’ .....	83
tabla 27 – caso de uso ‘modificar estado de servicios solicitados’ .....	85
tabla 28 – caso de uso ‘ver detalles de inscrito’ .....	88

tabla 29 – caso de uso ‘gestionar recibos’.....	89
tabla 30 – caso de uso ‘gestionar aplazamiento de servicios’ .....	92
tabla 31 – caso de uso ‘gestionar requisitos exceptuados’ .....	96
tabla 32 – caso de uso ‘gestionar sanciones’ .....	100
tabla 33 – caso de uso ‘gestionar excusas’.....	101
tabla 34 – caso de uso ‘gestionar cupos de servicio’ .....	103
tabla 35 – caso de uso ‘gestionar cupos de prueba’ .....	103
tabla 36 – caso de uso ‘adjudicar cupos de servicio’ .....	104
tabla 37 – caso de uso ‘gestionar servicios servidos’ .....	109
tabla 38 – caso de uso ‘gestionar asistencias a servicio’ .....	110
tabla 39 – caso de uso ‘hacer mantenimiento de años y semestres academicos’..	118
tabla 40 – caso de uso ‘hacer mantenimiento de parámetros’ .....	120
tabla 40 – caso de uso ‘modificar cupos adicionales’ .....	105
tabla 41 – caso de uso ‘hacer mantenimiento de tipos de usuario’ .....	121
tabla 42 – caso de uso ‘hacer mantenimiento de criterios’ .....	126
tabla 43 – caso de uso ‘hacer mantenimiento de servicios’ .....	127
tabla 44 – caso de uso ‘hacer mantenimiento de requisitos’ .....	129
tabla 45 – caso de uso ‘hacer mantenimiento de periodos de pago’ .....	130
tabla 46 – caso de uso ‘hacer mantenimiento de grupos especiales’ .....	136
tabla 47 - caso de uso ‘validar requisitos’ .....	137
tabla 48 - caso de uso ‘consultar’ .....	139
tabla 6 – tabla ‘cupos_servicios’ .....	50

# Resumen

## Título

Desarrollo del nuevo sistema de información para la inscripción, adjudicación, administración y control del servicio de comedores de Bienestar Universitario de la Universidad Industrial de Santander

## Autores

fidel david jiménez<sup>1</sup>  
juan martínez martínez<sup>1</sup>

## Palabras clave

Sistema de información, plataforma Web, comedores, Bienestar Universitario, servicios, Java, JBoss-Seam

## Descripción

El presente trabajo de grado fue desarrollado en conjunto con la División de Servicios de Información, encargada de administrar y desarrollar software que soporta el funcionamiento de la Universidad Industrial de Santander, y con la División de Bienestar Universitario. Este proyecto tiene como objetivo desarrollar un nuevo sistema de información para el servicio de comedores de Bienestar Universitario de la Universidad Industrial de Santander.

La nueva plataforma está basada en el sistema anterior, pero lo reemplaza completamente. Además el nuevo sistema implementa varias características con las que no contaba el sistema anterior, y soluciona sus problemas de fragmentación en varios módulos inconexos, su baja eficiencia, su poca facilidad de uso y su escasa flexibilidad. El nuevo sistema funciona en un ambiente completamente Web, lo cual permite a cualquiera de sus usuarios, desde sus administradores hasta los beneficiarios del servicio de comedores, ejecutarlo desde cualquier navegador de Internet.

Como todos los sistemas desarrollados por la División de Servicios de Información de la Universidad Industrial de Santander, el proyecto es realizado respetando los estándares de calidad y eficiencia; y además usando tecnologías de punta que permiten asegurar la robustez, escalabilidad y cohesión en los sistemas de la Universidad Industrial de Santander.

---

<sup>1</sup> facultad de Ingenierías Fisicomecánicas  
escuela de Ingeniería de Sistemas e Informática  
dirigido por el ingeniero Enrique Torres López

# Summary

## **Title**

Development of the new information system for the inscription, adjudication, administration and control of the dining room service provided by the Institutional Welfare division of the Industrial University of Santander

## **Authors**

Jiménez Fidel David<sup>2</sup>  
Martínez Martínez Juan<sup>2</sup>

## **Keywords**

Information system, web app, web platform, dining room, Institutional Welfare, University, services, Java, JBoss-Seam

## **Description**

The present project was developed in joint Information Services Division, a branch in charge of managing and developing the software that supports operations of the Industrial University of Santander, and with the Institutional Welfare Division. This project's goal is to develop a new information system for the dining room service provided by the Institutional Welfare division of the Industrial University of Santander.

The new platform is based on the previous system, but it will replace it completely. It also implements several new features that were absent in the previous system, and solves its problems of fragmentation in various disconnected modules, its low efficiency, its sparse ease of use, and its lack of flexibility and scalability. The new system operates in a completely web environment and on a single platform, which allows any of its users, from administrators to the recipients of the dining hall service, to use the software from any internet browser.

Like all software developed by the Information Services Division of the Industrial University of Santander, the project is developed respecting its standards of quality and efficiency; and also using advanced technologies which ensure the robustness, scalability and cohesion in the information systems of the Industrial University of Santander.

---

<sup>2</sup> department of Physique-Mechanics Sciences  
Systems Engineering school  
directed by the engineer Enrique Torres López

# Introducción

La División de Servicios de Información de la Universidad Industrial de Santander tiene la importante tarea de impulsar la innovación tecnológica en la Universidad, y de promover la participación de la comunidad universitaria en la generación de soluciones informáticas de alta calidad, que facilitan el proceso de modernización institucional.

Teniendo en cuenta lo anterior, este proyecto se concentra en el servicio de comedores de la División de Bienestar Universitario, la cual enfrenta una serie de dificultades que podrían ser identificadas y resueltas por la División de Servicios de Información, a través de este trabajo, con el que además se cumplirá el requisito de realizar nuestro Proyecto de Grado.

Este proyecto pretende ofrecer la estructura tecnológica necesaria para garantizar un mejor servicio en el área de comedores, siendo esta parte fundamental del soporte social que ofrece Bienestar Universitario, que actualmente beneficia a más de tres mil (3000) estudiantes.

Conscientes de que ya existe un sistema para la inscripción, adjudicación, administración y control del servicio de comedores, se sabe también que este sistema puede actualizarse y mejorarse considerablemente, en beneficio de todas las partes implicadas, desde los estudiantes que hacen uso del servicio de forma regular hasta los encargados de administrar el servicio de comedores, pasando por todos sus usuarios casuales y las directivas mismas de la Universidad que supervisan el proceso.

Los nuevos requisitos, el cambio de reglamentación, la nueva orientación que se le debe dar a los sistemas de información, hace necesario la construcción del nuevo sistema que soporte los procesos asociados a la prestación del servicio de comedores de Bienestar Universitario.

# Presentación del proyecto

## 1.1 Descripción del proyecto

### 1.1.1 Objetivo general

Desarrollar el Sistema de Información en ambiente web para el servicio de comedores de Bienestar Universitario de la Universidad Industrial de Santander.

### 1.1.2 Objetivos específicos

- Desarrollar un módulo para el manejo del proceso de inscripción del servicio de Comedores basado en las siguientes funcionalidades
  - Permitir a los usuarios registrar su inscripción, y modificarla en los plazos establecidos
  - Validar los requisitos necesarios para la inscripción de los usuarios
- Desarrollar un módulo para la administración del proceso de adjudicación de los usuarios inscritos al servicio de Comedores, considerando las siguientes funcionalidades

- Permitir a los administradores del sistema adjudicar el servicio de comedores a los usuarios inscritos teniendo en cuenta los criterios establecidos
- Permitir a los usuarios la consulta de los resultados de las adjudicaciones, vía web
- Emitir los recibos de pago con códigos de barras para agilizar la cancelación, registro y control de los mismos. Este módulo debe integrarse con el sistema financiero de la universidad
- Desarrollar un módulo que permita ejercer control de la asistencia y uso del servicio por parte de los usuarios
- Realizar la integración del sistema de comedores con el sistema para el manejo de Minutas, con el sistema para el manejo de Beneficios de Bienestar Universitario y con el sistema de información Financiero de la Universidad Industrial de Santander
- Establecer niveles de seguridad y de auditoría para controlar y hacer seguimiento de los accesos realizados al sistema
- Generar informes y estadísticas que permitan conocer
  - El registro histórico del número de usuarios inscritos y adjudicados
  - La cantidad de inscritos y adjudicados, clasificados por cada servicio, y estratificados por el valor base de matrícula
  - La lista de todos los inscritos por periodo académico, ordenados por el número de servicios adjudicados
  - El valor de los criterios mínimos que fueron necesarios para la adjudicación en cada servicio

## 1.2 Justificación

### 1.2.1 Antecedentes y descripción del problema

La división de Bienestar Universitario de la UIS ya contaba con un sistema de comedores desarrollado hace varios años y al que se le habían hecho diferentes mejoras conforme fue necesario, aunque no siempre utilizando las mismas plataformas de desarrollo. El resultado fue la división del sistema en varios módulos, algunas veces totalmente inconexos.

Algunos de estos módulos aún tenían interfaces basadas puramente en texto o Interfaces Textuales de Usuario, mientras que otros exigían que los usuarios del sistema en Bienestar Universitario pidieran asistencia a los ingenieros de la División de Servicios de Información para poder realizar ciertos procesos. Para agravar la situación, el sistema no cubría algunas tareas importantes de ejecución única en el semestre. Esto implicaba que un ingeniero de la División de Servicios de Información debía perder varias horas de trabajo haciendo modificaciones a las bases de datos, trabajando con consultas de SQL puro.

Esta situación hizo necesario crear un nuevo sistema más consistente y sensiblemente más amigable con el usuario, que no sólo cumpliera con los requisitos del sistema anterior sino que incluyera módulos nuevos para atender a los requisitos actuales de Bienestar Universitario. Este proyecto atiende a esa necesidad.

### 1.2.2 Impacto

El nuevo sistema de comedores facilitará considerablemente el trabajo de los empleados de Bienestar Universitario encargados de administrar y supervisar la operación del servicio de comedores, y facilitará el proceso de realizar análisis estadísticos relacionados con el mismo. Adicionalmente, la división de Bienestar Universitario dejará de depender de los ingenieros de la División de Servicios de Información para realizar tareas que el sistema actual no cumple.

Este sistema también será utilizado por los beneficiarios del servicio de comedores de la Universidad Industrial de Santander; actualmente un promedio por semestre

de 5 mil estudiantes, decenas de invitados a la universidad y algunos trabajadores de la UIS. En el futuro los profesores de la institución también podrían verse incluidos en el grupo. Todos ellos disfrutarán de mejoras en el sistema.

### 1.2.3 Viabilidad

Para la realización del proyecto se tuvieron a disposición entornos de desarrollo de software de distribución libre como JBoss Developer Studio 3.0 GA, el servidor de aplicaciones JBoss-seam patrocinado por la compañía Red Hat Software Inc., que actualmente tiene un contrato de soporte con la Universidad Industrial de Santander. Estas herramientas facilitaron el desarrollo del proyecto e hicieron posible que el mismo cumpliera con estándares altos de calidad y seguridad. Adicionalmente su popularidad implica que existe en la Red una gran cantidad de documentación disponible para el grupo de trabajo.

Por otra parte los desarrolladores del proyecto tuvieron a su disposición espacio y equipos apropiados ubicados en el Centro de las Tecnologías de la Información y Comunicación, CENTIC, de la Universidad Industrial de Santander, en donde se contó con el soporte técnico necesario para su realización.

Finalmente los ingenieros Enrique Torres López y Jackson Sonny de la División de Servicios de Información prestaron asistencia y asesoría necesarias para la realización del proyecto.

# 2

## Marco teórico

### 2.1 Tecnología usada para el proyecto

#### 2.1.1 Entorno Web

Considerando que las tecnologías Web modernas permiten crear aplicaciones con un nivel de dinamismo, complejidad y confiabilidad considerable, este proyecto fue desarrollado para que funcione completamente dentro de un entorno Web. Como resultado todos sus usuarios, desde los empleados de Bienestar Universitario hasta los beneficiarios del servicio, podrán utilizarlo desde un navegador de Internet sin instalar software adicional.

#### 2.1.2 Arquitectura multicapa

El objetivo de la arquitectura es la separación de todo el sistema en capas diferentes en las que se dividen las funciones del sistema. La ventaja principal de este estilo es que el desarrollo se puede llevar a cabo en varios niveles y, en caso de que sobrevenga algún cambio, sólo se ataca al nivel requerido sin tener que revisar entre código mezclado.

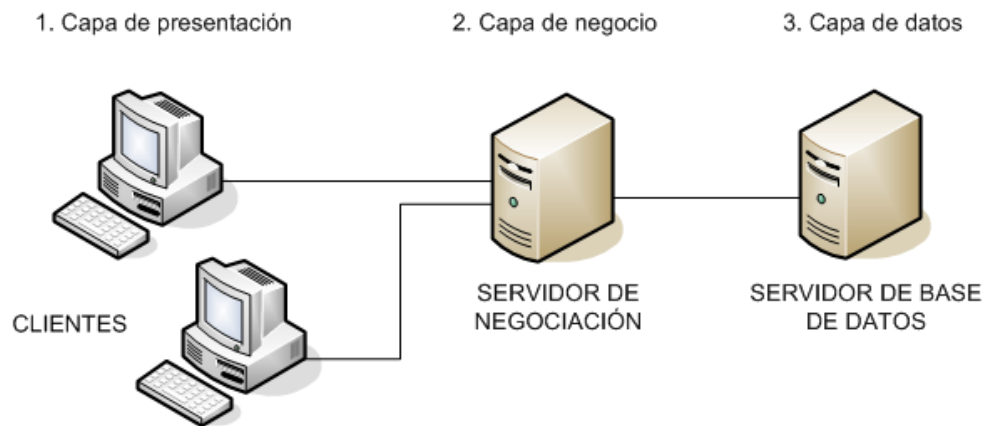
En el diseño de sistemas informáticos actual se suelen usar las arquitecturas multinivel o Programación por capas. En dichas arquitecturas a cada nivel se le confía una misión simple, lo que permite el diseño de arquitecturas escalables (que

pueden ampliarse con facilidad en caso de que las necesidades aumenten). El diseño más utilizado actualmente es el diseño en tres capas:

**Capa de presentación:** es la que se tiene contacto con el usuario. Presenta el sistema al usuario, le comunica la información y captura la información del usuario dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio. También es conocida como interfaz gráfica.

**Capa de la lógica de negocio:** es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Se denomina capa de negocio pues es aquí donde se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para solicitar al gestor de base de datos para almacenar o recuperar datos de él. También se consideran aquí los programas de aplicación.

**Capa de datos:** es donde residen los datos y es la encargada de acceder a los mismos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de datos, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.



*figura 1 - arquitectura multicapa*

Por lo general, en sistemas pequeños todas estas capas pueden residir en un único ordenador, si bien lo más usual es que haya una multitud de ordenadores en donde

reside la capa de presentación (son los clientes de la arquitectura cliente/servidor). Las capas de negocio y de datos pueden residir en el mismo ordenador, y si el crecimiento de las necesidades lo aconseja se pueden separar en dos o más ordenadores. Así, si el tamaño o complejidad de la base de datos aumenta, se puede separar en varios ordenadores los cuales recibirán las peticiones del ordenador en que resida la capa de negocio.

### 2.1.3 Entorno de desarrollo

La División de Servicios de Información desarrolla sus nuevos sistemas en la plataforma Java Enterprise Edition 5, que usa Java como lenguaje de programación, en la que se puede implementar aplicaciones multicapa y se apoya ampliamente en componentes de software modulares.

Dentro de las especificaciones de la plataforma, para el desarrollo del proyecto se usaron los siguientes frameworks.

**Java Server Faces (JSF)**<sup>3</sup>: es un framework para aplicaciones Java basadas en web que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE, y se usó en el proyecto para el desarrollo de la capa de presentación.

JSF incluye un conjunto de aplicaciones para representar componentes de una interfaz de usuario y administrar su estado, manejar eventos, validar entrada, definir un esquema de navegación de las páginas y dar soporte para internacionalización y accesibilidad; también contempla un conjunto por defecto de componentes para la interfaz de usuario y un modelo de eventos en el lado del servidor.

**Enterprise Java Beans 3.0**<sup>4</sup>: es una interfaz de programación de aplicaciones (API) que implementa un modelo de componentes distribuido estándar del lado del servidor. El objetivo de los EJB es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (conurrencia, transacciones, persistencia, seguridad, etc.) para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables.

---

<sup>3</sup>[javaserverfaces.org](http://javaserverfaces.org)

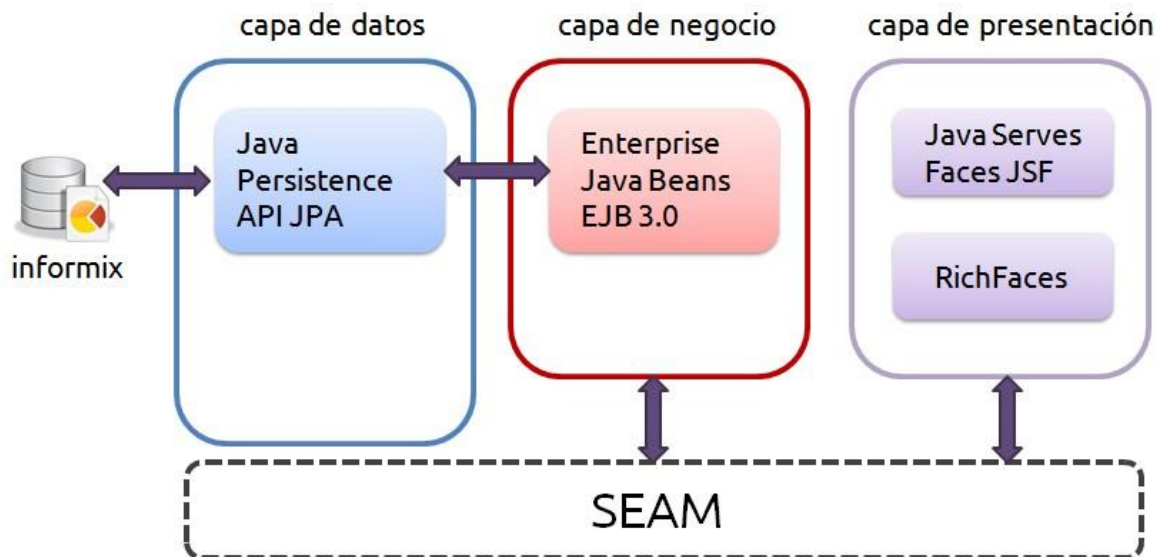
<sup>4</sup>[oracle.com/technetwork/java/javaee/ejb](http://oracle.com/technetwork/java/javaee/ejb)

Se usó para el desarrollo de la lógica del negocio del sistema.

**Java Persistence API<sup>5</sup>:** conocida como JPA, es un API de persistencia programado en el lenguaje Java que maneja datos relacionales en aplicaciones JEE 5. Este framework permite relacionar el modelo de dominio de la aplicación (clases) con las tablas en la base de datos (el mapeo objeto-relacional). El mapeo permite manejar las clases y sus objetos en la lógica del negocio, y no tablas y registros que son de conocimiento sólo del sistema de gestión de base de datos. Con JPA se crea una capa de abstracción de los datos, que permite la comunicación entre la capa de la lógica del negocio y la base de datos.

Se usó este framework para el desarrollo de la capa de datos (la capa de abstracción de los datos).

**Jboss Seam<sup>6</sup>:** es un framework para el manejo de la lógica del negocio, y como plataforma base para otros dos frameworks para el desarrollo de otras capas. Seam permite la definición de componentes implementados con EJB 3.0, que pueden ser accedidos mediante una interfaz por la capa de presentación para así establecer una comunicación entre ésta y la lógica de negocio. Combina a los 2 frameworks Enterprise JavaBeans EJB3 y Java Server Faces JSF y es el que permite ensamblar todas las capas de la aplicación.

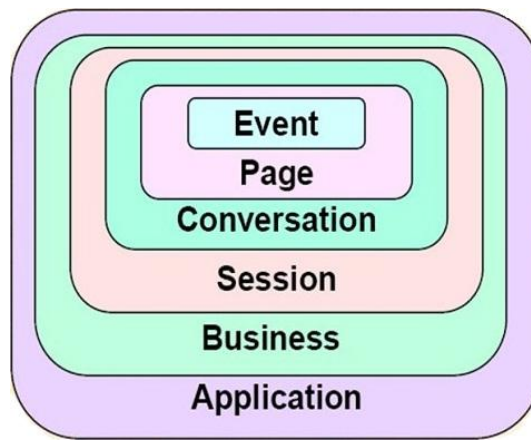


<sup>5</sup> [javaserverfaces.org](http://javaserverfaces.org)

<sup>6</sup> [seamframework.org](http://seamframework.org)

*figura 2 - estructura de la aplicación*

Además es un meta-contenedor de componentes agrupados en contextos definidos por seam. Cada contexto puede guardar los componentes que se definan y tiene un ciclo de vida propio del contexto donde se encuentren.



*figura 3 - contextos del seam*

## 2.1.4 Herramientas de desarrollo

Para el análisis, diseño, e implementación del sistema, se usaron las siguientes herramientas software:

- **Enterprise Architect<sup>7</sup>**: es una herramienta desarrollada por Sparx Systems para crear los diagramas UML en la etapa de análisis y diseño

---

<sup>7</sup> sparxsystems.com

- **Jboss Developer Studio**<sup>8</sup>: es un entorno integrado de desarrollo basado en Eclipse, que asocia el servidor de aplicaciones Jboss con otros plug-ins para facilitar la programación del sistema
- **Axure RP**<sup>9</sup>: una herramienta para construir prototipos semifuncionales de aplicaciones Web

## 2.2 Sistemas de información

### 2.2.1 Definición

Un sistema de información es un conjunto organizado de elementos, que pueden ser personas, datos, actividades o recursos materiales en general. Estos elementos interactúan entre sí para procesar información y distribuirla de manera adecuada en función de los objetivos de una organización.

Cabe resaltar que el concepto de sistema de información suele ser utilizado como sinónimo de sistema de información informático, aunque no son lo mismo. Este último pertenece al campo de estudio de la tecnología de la información y puede formar parte de un sistema de información como recurso material. De todas formas, se dice que los sistemas de información tratan el desarrollo y la administración de la infraestructura tecnológica de una organización.

### 2.2.2 Ciclo de vida de un sistema de información

El método de ciclo de vida para el desarrollo de sistemas es el conjunto de actividades que los analistas, diseñadores y usuarios realizan para desarrollar e implantar un sistema de información.

**Análisis de requisitos.** En esta fase se analizan las necesidades de los usuarios finales del software para determinar qué objetivos debe cubrir. De esta fase surge una memoria llamada SRD (documento de especificación de requisitos), que

---

<sup>8</sup>[devstudio.jboss.com/download](http://devstudio.jboss.com/download)

<sup>9</sup>[axure.com](http://axure.com)

contiene la especificación completa de lo que debe hacer el sistema sin entrar en detalles internos

Es importante señalar que en esta etapa se debe consensuar todo lo que se requiere del sistema, tomando en cuenta que realizar cambios en etapas posteriores será considerablemente más dispendioso.

**Diseño del sistema.** Se descompone y organiza el sistema en elementos que puedan elaborarse por separado, aprovechando las ventajas del desarrollo en equipo. Como resultado surge el SDD (Documento de Diseño del Software), que contiene la descripción de la estructura relacional global del sistema y la especificación de lo que debe hacer cada una de sus partes, así como la manera en que se combinan unas con otras. También se define el modelo de datos, y el modelo funcional del sistema.

Es conveniente distinguir entre diseño de alto nivel o arquitectónico y diseño detallado. El primero de ellos tiene como objetivo definir la estructura de la solución (una vez que la fase de análisis ha descrito el problema) identificando grandes módulos (conjuntos de funciones que van a estar asociadas) y sus relaciones. Con ello se define la arquitectura de la solución elegida. El segundo define los algoritmos empleados y la organización del código para comenzar la implementación.

**Codificación o implementación.** Es la fase en donde se implementa el código fuente, haciendo uso de prototipos así como de pruebas y ensayos para corregir errores. En términos generales, es la etapa en la que se construye el sistema basado en las especificaciones de requisitos y diseño.

**Pruebas y mantenimiento.** Los elementos, ya programados, se ensamblan para componer el sistema y se comprueba que funciona correctamente y que cumple con los requisitos, antes de ser entregado al usuario final.

Finalmente ingenieros con los conocimientos apropiados deben asegurarse de que el sistema funciona de forma apropiada y tienen que tener la capacidad de corregir errores en él en caso de que se presente una situación imprevista.

## 2.3 Generalidades del sistema de comedores

### 2.3.1 Definiciones y siglas

- **Bienestar Universitario:** dependencia administrativa de la Universidad Industrial de Santander que brinda apoyo para el buen desarrollo de la actividad académica, contribuyendo activamente en la formación integral de los estudiantes a través del desarrollo de programas y el ofrecimiento de servicios que propenden por el mejoramiento de su calidad de vida. Uno de estos servicios es el servicio de comedores.
- **División de Servicios de Información (DSI):** División de la Universidad Industrial de Santander que se encarga de desarrollar, administrar y modernizar las soluciones informáticas de la universidad.
- **Empleado:** persona vinculada a la UIS a través de un contrato de trabajo y que es beneficiaria del servicio de comedores por renunciar a un bono de auxilio de alimentación.
- **Estudiante:** persona matriculada a algún programa académico de la UIS.
- **Inscrito:** persona que solicitó el servicio de comedores dentro del lapso apropiado para ello.
- **Adjudicado:** un inscrito que solicitó el servicio de comedores, cumplió con los requisitos para recibirlo y alcanzó una posición en el concurso que le permitió ser elegido como beneficiario del servicio.
- **Sesión:** cada una de las sesiones de comida servidas en las instalaciones de Bienestar Universitario de la UIS. Por ejemplo:
  - desayuno
  - almuerzo

- comida

### 2.3.2 Descripción general

El sistema debe permitir que, al inicio de cada semestre, los operarios de la división de Bienestar Universitario definan una serie de parámetros que deberán cumplir los alumnos, trabajadores o cualquier otra persona que desee ser beneficiada con el servicio.

Estos parámetros incluyen tanto los requisitos que deben cumplir para inscribirse como los que deben cumplir para continuar recibiendo el servicio de comedores si fueron elegidos como beneficiarios.

Un módulo al que pueden acceder todos los matriculados a la UIS, permite inscribirse a aquellos que cumplan con los requisitos establecidos por Bienestar Universitario, para ser evaluados como posibles beneficiados por el servicio.

Un módulo del sistema permite a Bienestar Universitario establecer cierta cantidad de cupos para cada comida que servirá al día. El sistema hará un ordenamiento de las personas interesadas en recibir el servicio, seleccionando a los más aptos según criterios establecidos en otro módulo por Bienestar Universitario, y adjudicará el servicio al número de personas indicado por Bienestar Universitario.

Una vez se ha adjudicado el servicio de comedores a los estudiantes que cumplen los requisitos y quedan dentro de los cupos establecidos, el sistema debe generarles recibos de pago por sus servicios asignados, y una vez pasada la fecha límite para realizar la consignación el sistema debe reconocer que quienes pagaron son los inscritos aptos para recibir el servicio. En adelante, el sistema también llevará un control minucioso de los requisitos que deben cumplir estos usuarios para mantenerse como beneficiarios del sistema.

El sistema tiene un módulo que entrega cifras detalladas sobre los inscritos de un semestre determinado.

### 2.3.3 Documentación

La documentación relacionada con el diseño del sistema se encuentra en el archivo del programa Enterpirse Architect con el que se diagramó el mismo.

La documentación del código fuente se hizo en cada una de las clases, siguiendo el estándar de JAVADOC y documentando la definición de la clase, descripción de los métodos 'getters' y 'setters' y descripción de los parámetros de entrada y salida de cada uno de los métodos que componen la clase.

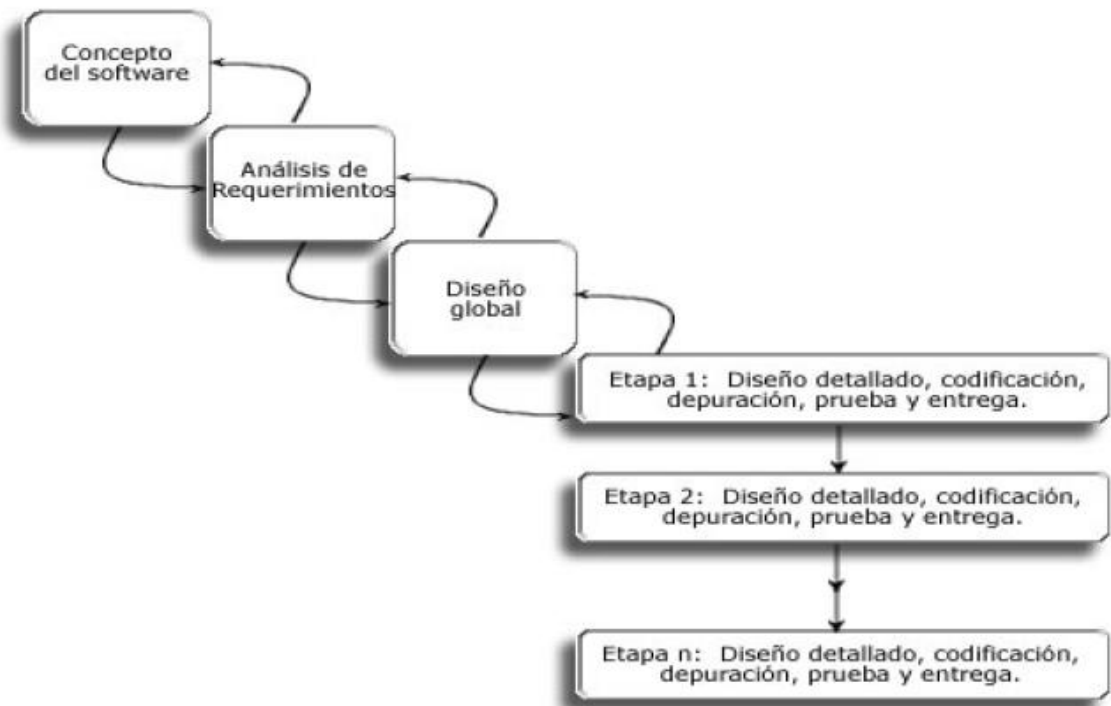
## 2.4 Metodología de desarrollo

### 2.4.1 Descripción de la metodología

Se utilizó para el desarrollo del sistema la metodología de 'Entrega por Etapas' o 'Entrega Incremental' usando prototipos, definida por Steve McConnell en Desarrollo y Gestión de Proyectos Informáticos<sup>10</sup>

---

<sup>10</sup> McConnell, Steve. Desarrollo y Gestión de Proyectos Informáticos. Madrid, McGraw-Hill. 1997



*figura 4 – descripción de la metodología de desarrollo*

La entrega por etapas no disminuyó el tiempo necesario para desarrollar el software, pero redujo sustancialmente los riesgos implícitos en su construcción, y también proporcionó signos tangibles de progreso que fueron visibles para el cliente y útiles para la directiva en la evaluación del estado del proyecto. Otra de las ventajas de usar esta metodología fue que proporcionó la seguridad de que la arquitectura del producto era sólida, y obligó a definir cuidadosamente las etapas (módulos) de entrega.

Como sugiere la **figura 4**, con la entrega de las etapas se atraviesan los pasos de definición del concepto de software, análisis de requisitos y creación del diseño global. Luego se procede a hacer el diseño detallado, codificación, depuración y prueba con cada etapa, usando prototipos para su refinamiento y creando un producto para entregar al final de cada etapa.

Debido a que se conocía muy bien la Lógica del Negocio gracias a la experiencia vivida con el sistema antiguo, y tomando en cuenta que se tenían los nuevos

requisitos definidos en reuniones conjuntas con los usuarios del sistema, se concluyó que había claridad sobre qué módulos debían contemplarse y el alcance que convenía dar al nuevo sistema de comedores. La estructura por módulos facilitó e hizo más eficiente el desarrollo con una metodología incremental.

## 2.4.2 Razones para la elección de la metodología

Las siguientes razones son las que se tuvieron en cuenta para la elección de la metodología de Desarrollo Incremental:

- **El sistema se conforma por módulos muy definibles:** en este punto es más eficiente utilizar entregas incrementales que lineales.
- **Se tiene muy bien definido el sistema a implementar:** es un factor fundamental el tener un sistema ya existente, lo cual facilita conocer con detalle la lógica del negocio y los nuevos requisitos del nuevo sistema.
- **Se podrán ver resultados más rápido y funcionando:** a pesar que no es una exigencia la visibilidad inmediata del sistema, el cumplimiento de tiempos sí es fundamental, permitiendo proporcionar una funcionalidad útil en las manos de los usuarios antes de entregar la totalidad del proyecto.
- **Se construirá un sistema dinámico en su crecimiento funcional:** será más fácil modificar el sistema desarrollado en cobertura y diversidad durante su tiempo de vida.

# 3

## Desarrollo del sistema

### 3.1 Plan de trabajo

#### 3.1.1 Acoplamiento y capacitación

En esta fase se hará una familiarización y capacitación con las políticas de la División de Servicios de Información, sus metodologías, estándares, herramientas, software, y modelos ya implementados con los que se desarrollan los nuevos sistemas para la universidad.

Se establecerán prioridades para el desarrollo del plan y se realizará un análisis al sistema actual que apoya las actividades del servicio de comedores de Bienestar Universitario.

#### 3.1.2 Análisis y diseño global del nuevo sistema

Teniendo una clara visión del sistema actual, se empieza la segunda fase, definida en las siguientes etapas:

- análisis y conocimiento de los requisitos del sistema
- planificación de los módulos
- elaboración de los diagramas de casos de uso

- diseño del modelo de datos
- diseño del prototipo no funcional

### 3.1.3 Diseño e implementación de los módulos

Con los requisitos y el diseño global aprobados, y las etapas(módulos) planificadas, se pasó a hacer un diseño funcional detallado de cada etapa, considerando la construcción de su respectivo prototipo. Implementación de cada etapa:

- **Etapa 1:** módulo de inscripciones
  - Diseño funcional detallado
  - Implementación y construcción con base en el prototipo
  - Refinamiento del prototipo
- **Etapa 2:** módulo de adjudicación
  - Diseño funcional detallado.
  - Implementación y construcción con base en el prototipo.
  - Refinamiento del prototipo
- **Etapa 3:** módulo para el control de la asistencia
  - Diseño funcional detallado
  - Implementación y construcción con base en el prototipo
  - Refinamiento del prototipo

- **Etapa 4:** módulo de administración
  - Diseño funcional detallado
  - Implementación y construcción con base en el prototipo
  - Refinamiento del prototipo
- **Etapa 5:** implementación de la seguridad

Finalmente se hacen pruebas individuales.

### 3.1.4 Puesta en marcha, y entrega del producto final

Con todas las entregas implementadas, se entra a la cuarta fase, donde se integran los módulos desarrollados, definiendo las siguientes etapas:

- Pruebas al sistema completo e integrado
- Capacitación general de los usuarios
- Seguimiento y ajustes al sistema implantado
- Entrega del producto final

## 3.2 Acoplamiento y capacitación

Para asegurarse de tener los conocimientos necesarios para manejar las herramientas utilizadas en la realización del proyecto, los desarrolladores recibieron un curso básico de Java, seguido por un curso más avanzado, donde aprendieron a utilizar las siguientes herramientas:

- **JBoss Seam:** un framework de desarrollo de aplicaciones Web
- **Java Server Faces:** un framework de desarrollo de interfaces para aplicaciones Web
- **Java Persistence API, o JPA por su sigla en inglés:** una interface de programación de aplicaciones en Java para administrar datos relacionales. Este curso incluyó la utilización del lenguaje de consultas a bases de datos Java Persistence Query Language, o JPQL por su sigla en inglés

## 3.3 Análisis y diseño global del nuevo sistema

### 3.3.1 Análisis y conocimiento de los requisitos del sistema

Para construir la lista de requisitos del proyecto se hizo un trabajo previo de análisis del antiguo sistema de comedores de Bienestar Universitario que incluyó entrevistas con sus usuarios, así como con los ingenieros encargados de prestar apoyo a los empleados de Bienestar Universitario para definir las funciones faltantes en el software antiguo.

Posteriormente se sostuvieron reuniones con directivos de Bienestar Universitario (el cliente), así como con algunos de los que serían los principales usuarios del nuevo sistema, para establecer nuevos requisitos y sentar una base para iniciar el análisis y diseño del nuevo sistema de comedores para Bienestar Universitario de la UIS.

Con la información del análisis se prosiguió a dividir las funcionalidades en módulos, y a elaborar los diagramas de casos de uso correspondientes.

### 3.3.2 Planificación de los módulos

Un análisis del sistema antiguo de comedores junto con los requisitos del nuevo sistema permitieron llegar a la conclusión de que este necesitaría los siguientes 4 módulos:

- **Administración:** un módulo donde los empleados de Bienestar Universitario pueden establecer parámetros del sistema como:
  - Los requisitos para determinar qué personas pueden ser beneficiadas por el servicio.
  - Las fechas en las que es válido presentar una petición para recibir el servicio de comedores.
  - Las fechas en las que empiezan los periodos de pago.
  - Las raciones que se servirán al día.
  - El valor de cada ración.
  - Los tipos de usuarios que tendrá su sistema.
- **Inscripciones:** un módulo donde los interesados en ser beneficiarios del servicio de comedores pueden presentar su petición para recibirlo, y luego modificar esa petición o cancelarla. Adicionalmente quienes resulten beneficiados por el servicio pueden usar el módulo para presentar excusas si un día no van a poder asistir. Por su parte los empleados de Bienestar Universitario pueden utilizar este módulo para modificar el estado de un beneficiario del servicio, para adjudicar a una persona el servicio manualmente o para retirar el servicio a algún beneficiario.
- **Adjudicación:** un módulo que permite a los empleados de Bienestar Universitario indicar cuántas personas serán beneficiadas por el servicio, y que se encarga de calcular cuáles de los candidatos que se presentaron serán beneficiados basándose en los parámetros de prioridad establecidos por los mismos empleados de Bienestar Universitario.
- **Control:** un módulo que permite tomar asistencia de los beneficiados por el servicio a las raciones que se sirven cada día, y que revisa que los beneficiarios cumplan con los requisitos para continuar recibiendo el servicio.

### 3.3.3 elaboración de los diagramas de casos de uso

Diagrama de casos de uso del módulo de inscripción:

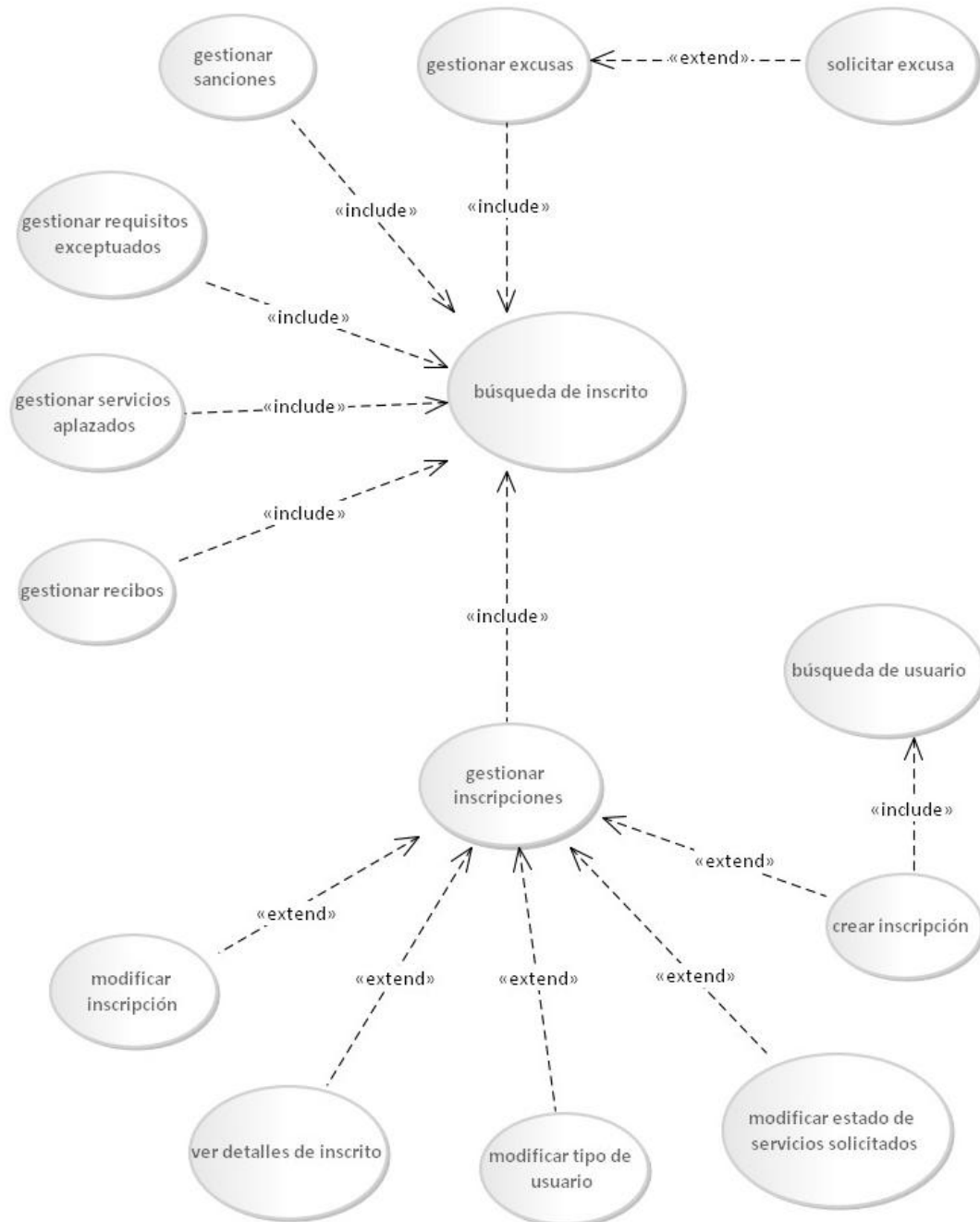


figura 5 – diagrama de casos de uso del módulo de inscripción

Relación del actor Operario con los casos de uso del módulo de inscripción:

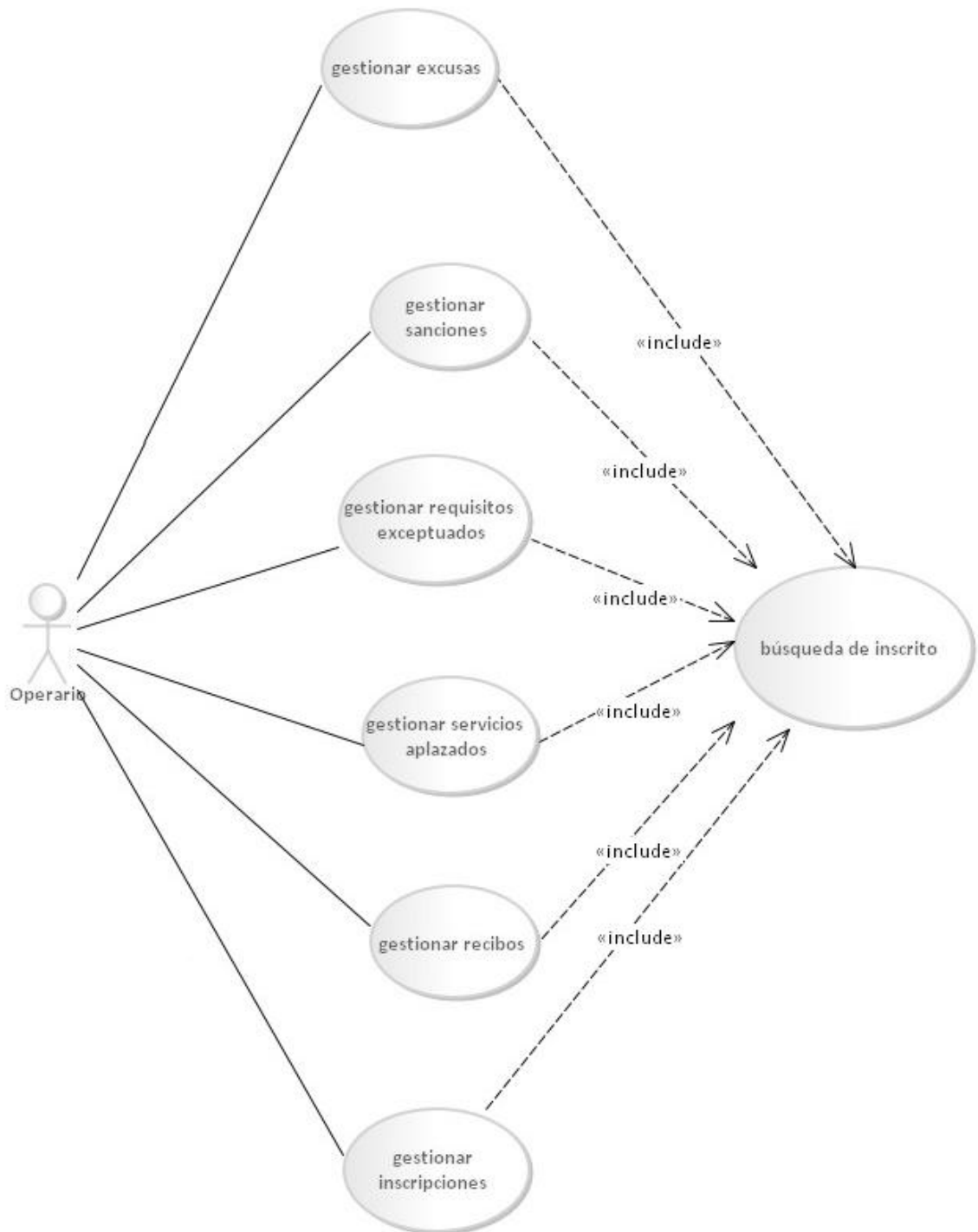
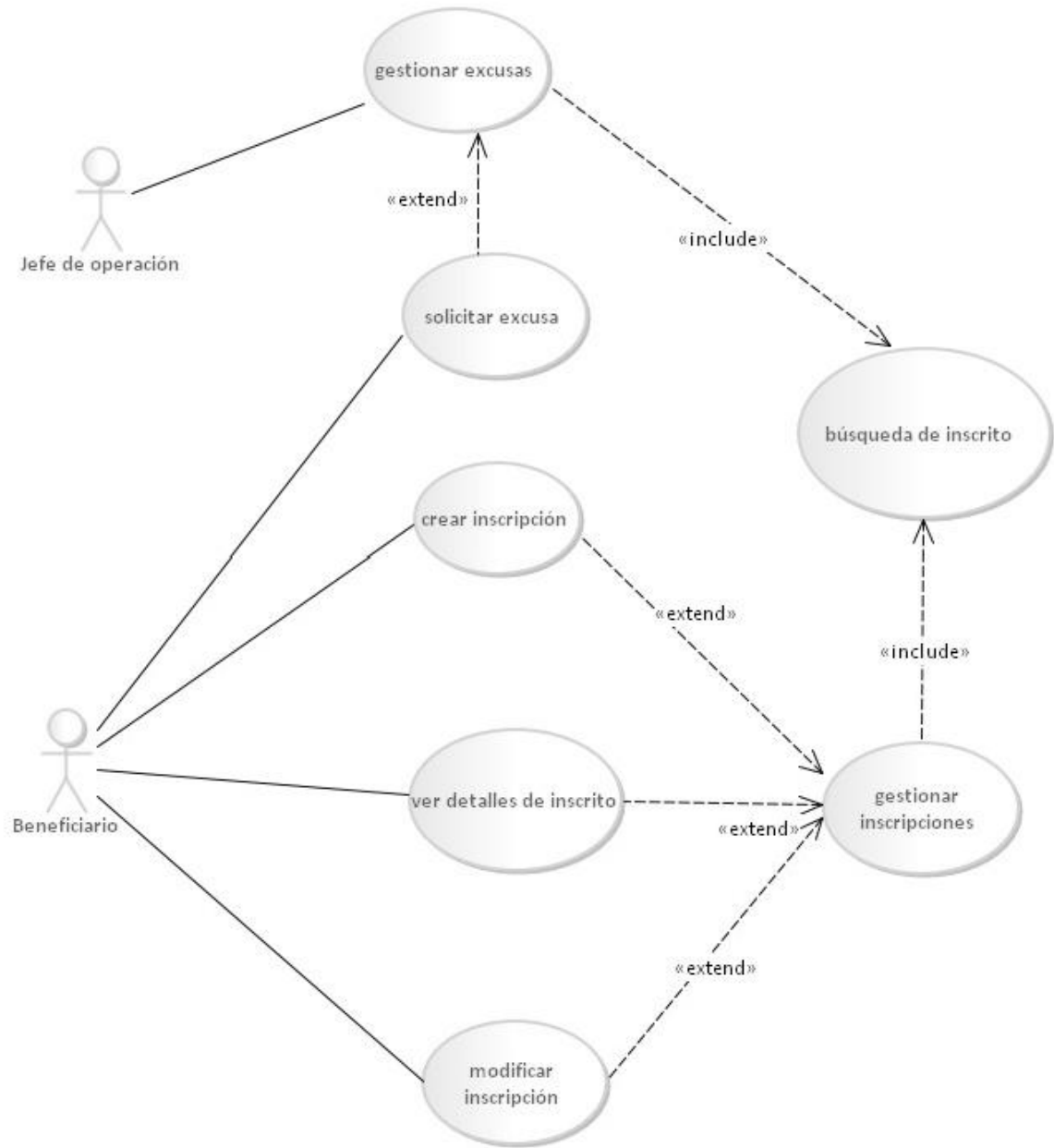


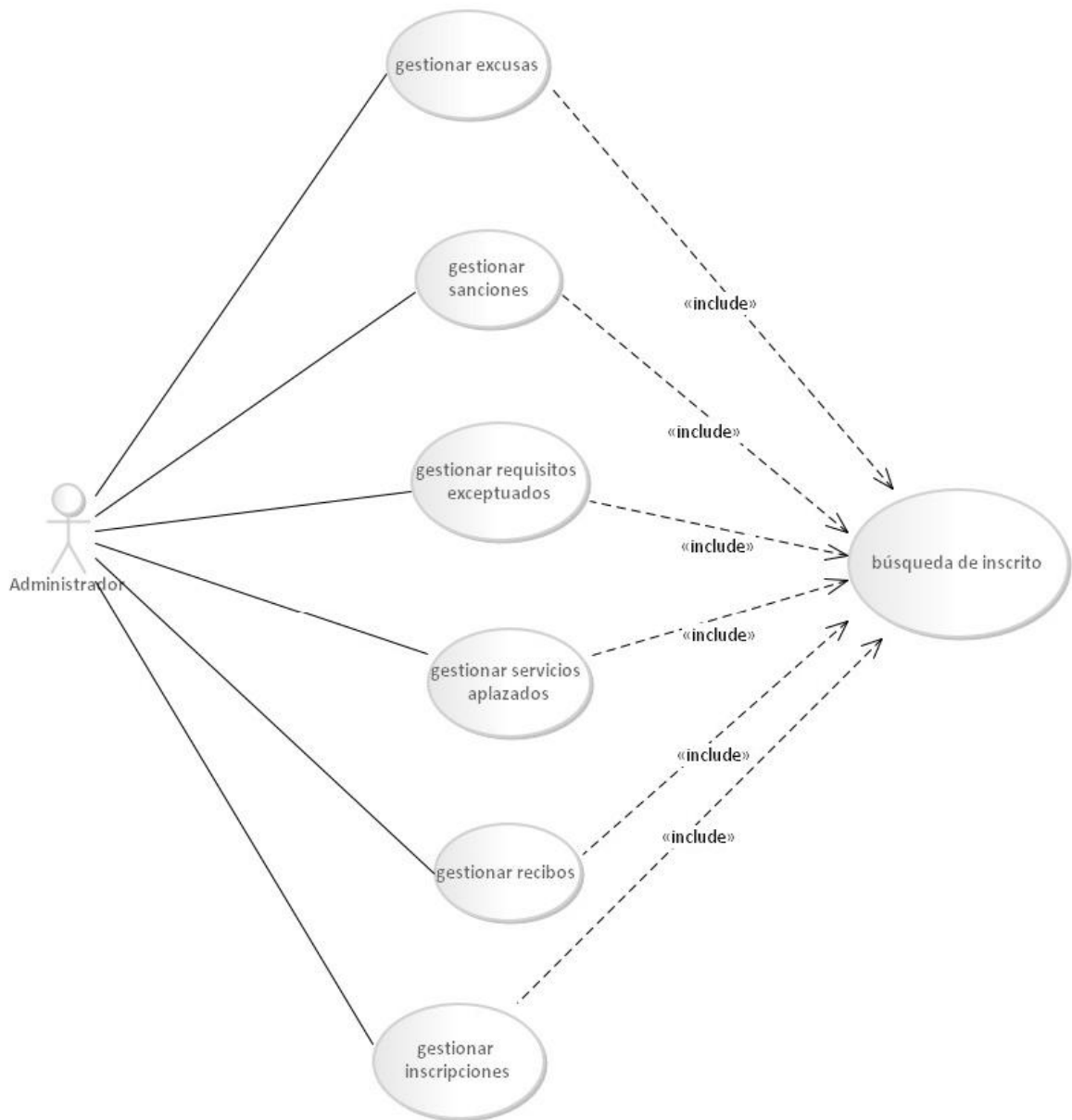
figura 5 A - relación del actor Operario con los casos de uso del módulo de inscripción

Relación de los actores Jefe de operación y Beneficiario con los casos de uso del módulo de inscripción:



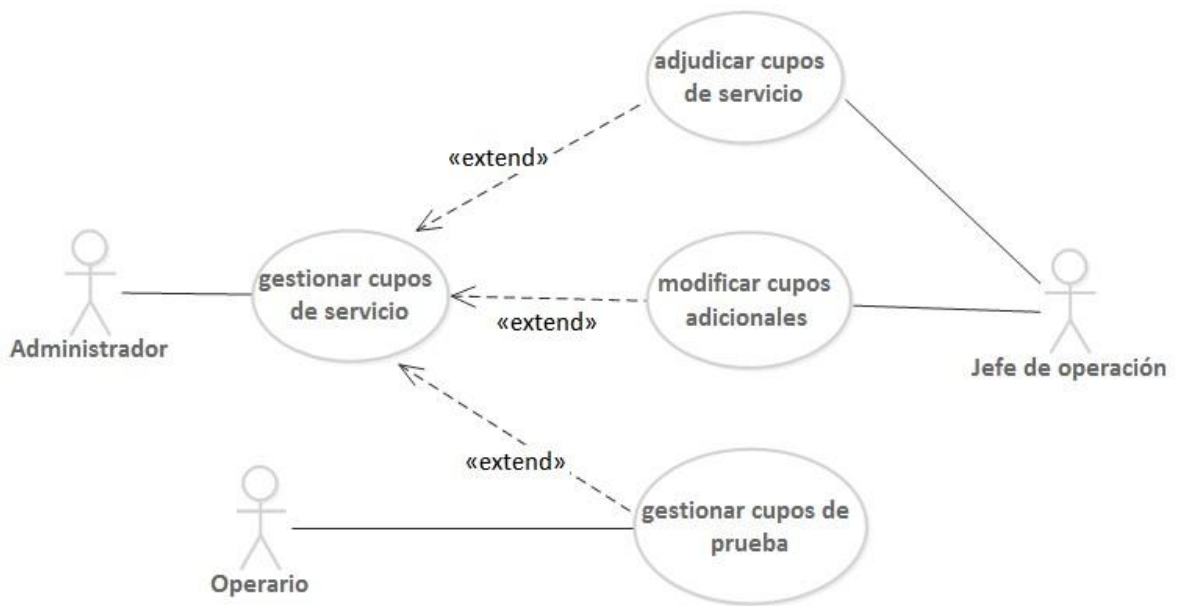
*figura 5 B - Relación de los actores Jefe de operación y Beneficiario con los casos de uso del módulo de inscripción*

Relación del actor Administrador con los casos de uso del módulo de inscripción:



*figura 5 C - Relación del actor Administrador con los casos de uso del módulo de inscripción*

Diagrama de casos de uso del módulo de adjudicación:



*figura 6 – diagrama de casos de uso del módulo de adjudicación*

Diagrama de casos de uso del módulo de administración:

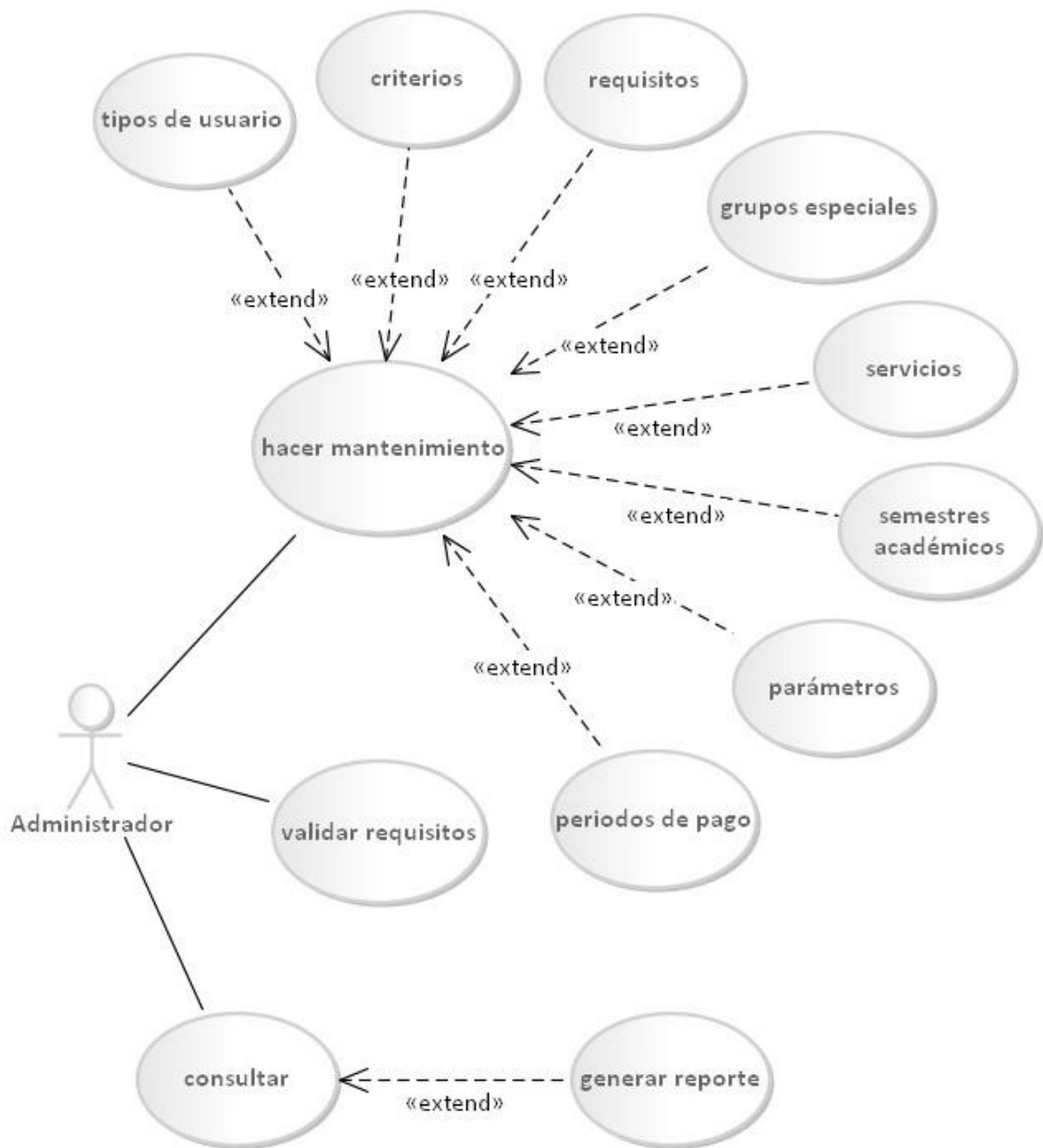
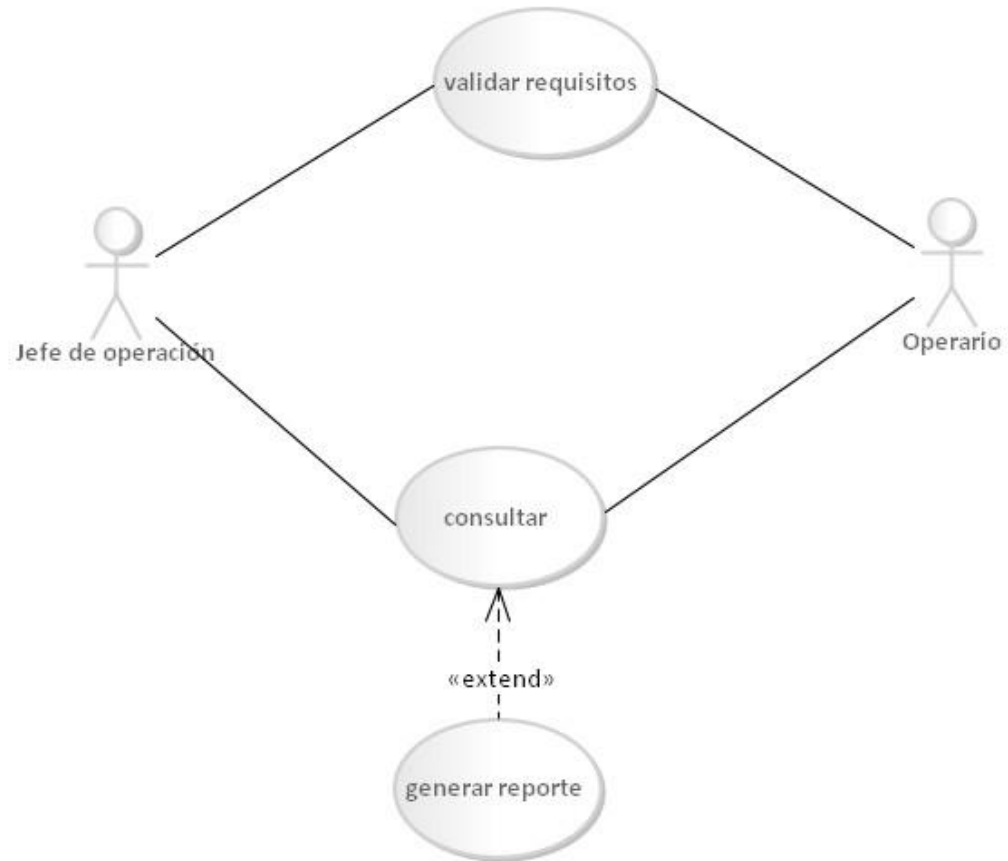


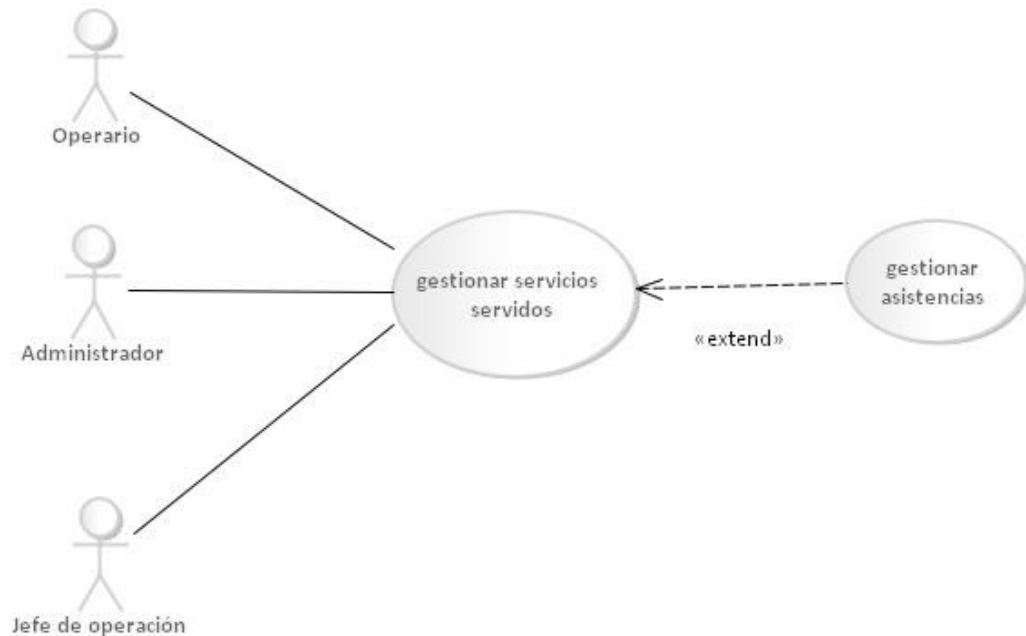
figura 7 – diagrama de casos de uso del módulo de administración

Relación de los actores Jefe de operación y Operario con los casos de uso del módulo de administración:



*figura 7 A – Relación de los actores Jefe de operación y Operario con los casos de uso del módulo de administración*

## Diagrama de casos de uso del módulo de Control

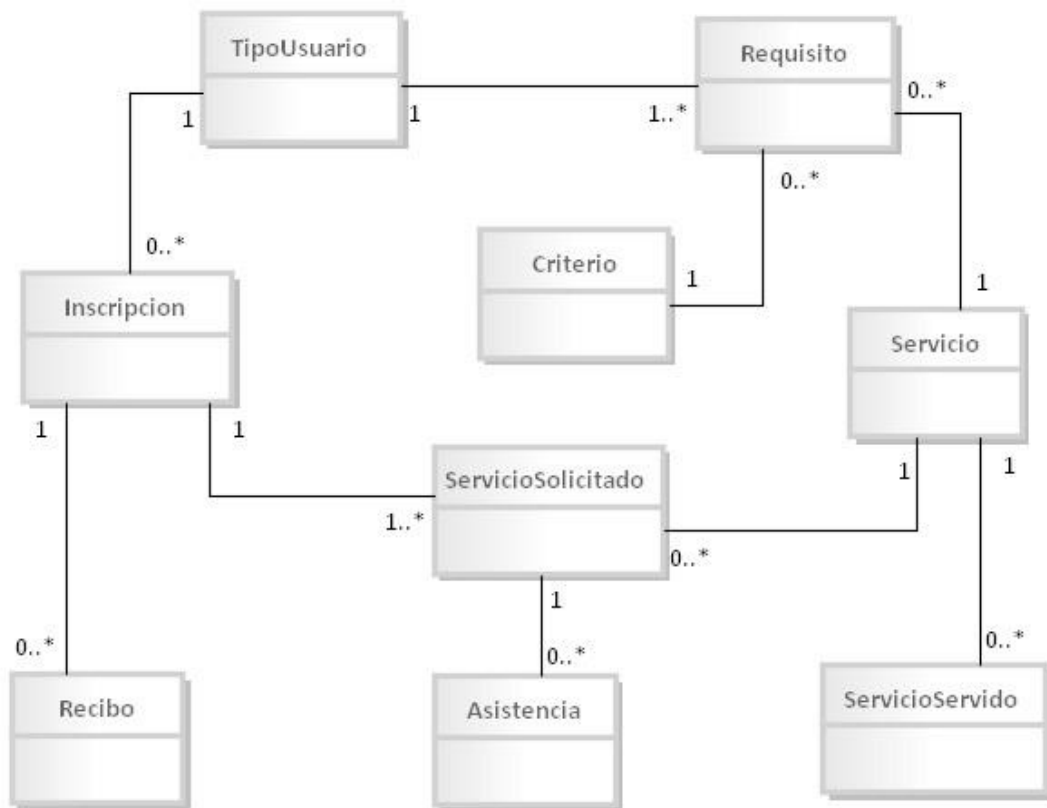


*figura 8 – diagrama de casos de uso del módulo de control*

### 3.3.4 Modelo de datos

A continuación se presenta un fragmento importante, aunque simplificado, del modelo de datos del sistema. Un modelo detallado se encuentra adjunto a este documento en un medio digital, y puede explorarse utilizando el software Enterprise Architect.

En la capa de la lógica del negocio, no se utilizan tablas sino clases. El sistema relaciona las dos a través de un 'mapeo' de los datos. Este es un diagrama de clases parcial del nuevo sistema de comedores, idéntico al mismo fragmento del diagrama entidad-relación del sistema:



*figura 9 – diagrama de clases parcial del sistema*

### 3.3.5 Diseño del modelo de entidad-relación

El diagrama de entidad relación se encuentra en un archivo adjunto a este documento en un medio digital, y puede explorarse utilizando el software Enterprise Architect. A continuación, se hace una descripción breve de cada tabla en la base de datos.

tabla	anos_semestres_academicos
descripción	<p>Esta tabla almacena la información de los periodos académicos, para que el sistema de comedores pueda conocer el año y el semestre en el que se presta el servicio.</p> <p>Se diseñó de tal manera que el último registro insertado sea el periodo académico actual.</p>
Campos	<p>llave primaria</p> <p>ano (SMALLINT)</p> <p>semestre (SMALLINT)</p> <p>fecha_inicio (DATE)</p> <p>fecha_fin (DATE)</p> <p>fecha_inicio_inscripciones (DATE)</p> <p>fecha_fin_inscripciones (DATE)</p>

*tabla 1 – tabla 'anos\_semestres\_academicos'*

tabla	aplazamientos_servicios
descripción	<p>Almacena los aplazamientos realizados a un inscrito para un servicio y un periodo de pago.</p> <p>Un aplazamiento hace que el inscrito mantenga su cupo para el siguiente periodo de pago, pero no puede usar el servicio que aplazado en el periodo de pago.</p> <p>Se relaciona con las tablas 'servicios_solicitados' y con 'periodos' de las cuales trae sus llaves primarias.</p>

campos	llave primaria	
	ano_inscripcion	SMALLINT NOT NULL,
	semestre_inscripcion	SMALLINT NOT NULL,
	documento_usuario	DECIMAL(13,0) NOT NULL,
	codigo_tipo_documento	SMALLINT NOT NULL,
	codigo_servicio	INTEGER NOT NULL
	consecutivo_periodo	SMALLINT NOT NULL,
	ano_periodo	SMALLINT NOT NULL,
	semestre_periodo	SMALLINT NOT NULL
	observaciones	TEXT NOT NULL

*tabla 2 – tabla 'aplazamientos\_servicios'*

tabla	Asistencias	
descripción	Almacena las asistencias a una sesión de servicio (fecha y servicio). El estado de la asistencia nos indica si el inscrito se presentó al servicio o no. Se relaciona con la tabla 'servicios_solicitados' de la cual trae su llave primaria.	
campos	llave primaria	
	ano	SMALLINT NOT NULL,
	semestre	SMALLINT NOT NULL,
	documento_usuario	DECIMAL(13,0) NOT NULL,
	codigo_tipo_documento	SMALLINT NOT NULL,
	codigo_servicio	INTEGER NOT NULL,
	fecha_asistencia	DATE NOT NULL
	estado	VARCHAR(2) NOT NULL

*tabla 3 – tabla 'asistencias'*

tabla	cambios_cumplimiento_requisitos
descripción	Guarda la información de un requisito, cuando éste ha cambiado su estado de cumplimiento (S = sí cumple, N = no cumple). Se relaciona con las tablas 'inscripciones' y con 'requisitos' de las cuales trae sus llaves primarias.
campos	llave primaria codigo_cambio_cumplimiento_requisito SERIAL NOT NULL,  ano SMALLINT NOT NULL, semestre SMALLINT NOT NULL, documento_usuario DECIMAL(13,0) NOT NULL, codigo_tipo_documento SMALLINT NOT NULL, codigo_servicio INTEGER NOT NULL, codigo_criterio INTEGER NOT NULL, codigo_tipo_usuario INTEGER NOT NULL, fecha_cambio DATE NOT NULL, hora_cambio DATETIME HOUR TO FRACTION(3) NOT NULL, estado_cambio VARCHAR(2) NOT NULL

*tabla 4 – tabla 'cambios\_cumplimiento\_requisitos'*

tabla	criterios
descripción	Guarda la información de un criterio. Un criterio almacena una consulta en la base de datos para hallar algún valor que será usado para compararlo.

campos	llave primaria	
	codigo_criterio	SERIAL NOT NULL,
	nombre_criterio	VARCHAR(100) NOT NULL,
	nombre_tabla	VARCHAR(100) NOT NULL,
	nombre_atributo	VARCHAR(100) NOT NULL,
	funcion	VARCHAR(25),
	lista_atributos_ordenamiento	LVARCHAR(700),
	tipo_dato_retornado	VARCHAR(70) NOT NULL,
	lista_condiciones_where	LVARCHAR(700) NOT NULL,
prioridad	INTEGER	

*tabla 5 – tabla 'criterios'*

tabla	cupos_servicios	
descripción	<p>Guarda la información de cuantos cupos se adjudicarán para cada servicio en un periodo académico. También guardan cuantos cupos se han adicionado sobre los primeros adjudicados. Se relaciona con las tablas 'anos_semestres_academicos' y con 'servicios' de las cuales trae sus llaves primarias.</p>	
campos	llave primaria	
	ano	SMALLINT NOT NULL,
	semestre	SMALLINT NOT NULL,
	codigo_servicio	INTEGER NOT NULL,
	cupos_adjudicados_inicio	INTEGER NOT NULL,
	cupos_pruebas	INTEGER NOT NULL,
	cupos_adicionales	INTEGER NOT NULL

*tabla 6 – tabla 'cupos\_servicios'*

tabla	detalles_recibos
descripción	Guarda la información de los detalles (servicios) de facturación de un recibo de pago. El valor del campo 'valor_total_detalle' se calcula multiplicando el valor de la ración del servicio por el número de días facturados. Se relaciona con la tabla 'recibos' y con 'servicios' de las cuales trae sus llaves primarias.
campos	llave primaria numero_recibo     INTEGER NOT NULL, codigo_servicio    INTEGER NOT NULL,  valor_total_detalle DECIMAL(7,0) NOT NULL

*tabla 7 – tabla 'detalles\_recibos'*

tabla	grupos_especiales
descripción	Los grupos especiales representan los grupos deportivos y artísticos de la universidad en los que, si un estudiante participa en ellos obtiene beneficios para la adjudicación de servicios de comedores.
campos	llave primaria codigo_grupo_especial    SERIAL NOT NULL,  nombre_grupo_especial    VARCHAR(150) NOT NULL

*tabla 8 – tabla 'grupos\_especiales'*

tabla	Inscripciones
descripción	Esta tabla almacena la información básica de un inscrito para un periodo académico. Se relaciona con la tabla 'anos_semestres_academicos' de la cual trae su llave primaria.
campos	<p>llave primaria</p> <p>ano SMALLINT NOT NULL,  semestre SMALLINT NOT NULL,  documento_usuario DECIMAL(13,0) NOT NULL,  codigo_tipo_documento SMALLINT NOT NULL,</p> <p>codigo_tipo_usuario INTEGER NOT NULL,  correo_electronico VARCHAR(100)</p>

*tabla 9 – tabla 'inscripciones'*

tabla	inscritos_grupos_especiales
descripción	Esta tabla almacena las relaciones entre un inscrito y un grupo especial, y un registro representa que el inscrito pertenece a un grupo especial. Se relaciona con las tablas 'inscripciones' y 'grupos_especiales' de las cuales trae sus llaves primarias.
campos	<p>llave primaria</p> <p>ano SMALLINT NOT NULL,  semestre SMALLINT NOT NULL,  documento_usuario DECIMAL(13,0) NOT NULL,  codigo_tipo_documento SMALLINT NOT NULL,  codigo_grupo_especial INTEGER NOT NULL</p>

*tabla 10 – tabla 'inscritos\_grupos\_especiales'*

tabla	parametros_comedores	
descripción	Esta tabla almacena los parámetros que son comunes a todo el sistema de comedores. funciona como un arreglo asociativo, donde el nombre del parámetro es único y el valor es una cadena.	
campos	llave primaria codigo_parametro	SERIAL NOT NULL,
	nombre_parametro descripcion_parametro valor_parametro	VARCHAR(200) NOT NULL, VARCHAR(100), VARCHAR(50) NOT NULL

*tabla 11 – tabla 'parametros\_comedores'*

tabla	periodos	
descripción	Esta tabla almacena la información de los periodos de pago. Se obliga a que las fechas de inicio y fin de periodos consecutivos tengan coherencia temporal. Se relaciona con la tabla 'anos_semestres_academicos' de la cual trae su llave primaria.	
campos	llave primaria ano semestre consecutivo	SMALLINT NOT NULL, SMALLINT NOT NULL, SMALLINT NOT NULL,
	fecha_inicio fecha_fin numero_dias	DATE NOT NULL, DATE NOT NULL, SMALLINT NOT NULL

*tabla 12 – tabla 'periodos'*

tabla	recibos
descripción	<p>Almacena los recibos de pago de cada inscrito en un periodo de pago. Los recibos son generados facturando los servicios que tenga adjudicado el inscrito, y por cada periodo sólo puede haber un recibo válido.</p> <p>Se relaciona con las tablas 'inscripciones' y 'periodos' de las cuales trae sus llaves primarias.</p>

*tabla 13 – tabla 'recibos'*

tabla	requisitos														
descripción	<p>Esta tabla almacena los requisitos que se exigen para la adjudicación y habilitación de servicios de comedores. Un requisito es la exigencia en el valor de un criterio para todos los tipos de usuario para un servicio en particular.</p> <p>Se relaciona con las tablas 'tipos_usuario', 'criterios' y 'servicios' de las cuales trae sus llaves primarias.</p>														
campos	<p>llave primaria</p> <table> <tr> <td>codigo_criterio</td> <td>INTEGER NOT NULL,</td> </tr> <tr> <td>codigo_servicio</td> <td>INTEGER NOT NULL,</td> </tr> <tr> <td>codigo_tipo_usuario</td> <td>INTEGER NOT NULL,</td> </tr> </table> <table> <tr> <td>valor_maximo</td> <td>VARCHAR(30),</td> </tr> <tr> <td>valor_minimo</td> <td>VARCHAR(30),</td> </tr> <tr> <td>orden_revision</td> <td>INTEGER,</td> </tr> <tr> <td>tipo_revision</td> <td>VARCHAR(100)</td> </tr> </table>	codigo_criterio	INTEGER NOT NULL,	codigo_servicio	INTEGER NOT NULL,	codigo_tipo_usuario	INTEGER NOT NULL,	valor_maximo	VARCHAR(30),	valor_minimo	VARCHAR(30),	orden_revision	INTEGER,	tipo_revision	VARCHAR(100)
codigo_criterio	INTEGER NOT NULL,														
codigo_servicio	INTEGER NOT NULL,														
codigo_tipo_usuario	INTEGER NOT NULL,														
valor_maximo	VARCHAR(30),														
valor_minimo	VARCHAR(30),														
orden_revision	INTEGER,														
tipo_revision	VARCHAR(100)														

*tabla 14 – tabla 'requisitos'*

tabla	requisitos_exceptuados
descripción	<p>Esta tabla almacena los requisitos que se exceptúan a un inscrito y por lo tanto no son tomados en cuenta para la validación.</p> <p>Se relaciona con las tablas 'requisitos' e 'inscripciones' de las cuales trae sus llaves primarias.</p>
campos	<p>llave primaria</p> <p>ano SMALLINT NOT NULL,  documento_usuario DECIMAL(13,0) NOT NULL,  codigo_tipo_documento SMALLINT NOT NULL,  semestre SMALLINT NOT NULL,  codigo_criterio INTEGER NOT NULL,  codigo_servicio INTEGER NOT NULL,  codigo_tipo_usuario INTEGER NOT NULL,</p> <p>observaciones TEXT NOT NULL</p>

*tabla 15 – tabla 'requisitos\_exceptuados'*

tabla	sanciones
descripción	<p>Esta tabla almacena la información de las sanciones impuestas a los inscritos. Las sanciones tienen una duración cuya información se almacena en los campos 'fecha_inicio' y 'fecha_fin'.</p> <p>Se relaciona con la tabla 'servicios_solicitados' de la cual trae su llave primaria.</p>

campos	llave primaria	
	codigo_sancion	SERIAL NOT NULL,
	ano	SMALLINT NOT NULL,
	semestre	SMALLINT NOT NULL,
	documento_usuario	DECIMAL(13,0) NOT NULL,
	codigo_tipo_documento	SMALLINT NOT NULL,
	codigo_servicio	INTEGER NOT NULL,
	observaciones	TEXT NOT NULL,
fecha_inicio	DATE NOT NULL,	
fecha_fin	DATE NOT NULL	

*tabla 16 – tabla ‘sanciones’*

tabla	Servicios	
descripción	Esta tabla almacena la información de los servicios que presta la sección de comedores.	
campos	llave primaria	
	codigo_servicio	INTEGER NOT NULL,
	nombre_servicio	VARCHAR(70) NOT NULL,
	nombre_servicio_plural	VARCHAR(70) NOT NULL,
	nombre_servicio_abreviado	VARCHAR(3) NOT NULL,
	valor_racion	DECIMAL(5,0) NOT NULL,
	hora_inicio	DATETIME HOUR TO
	MINUTE NOT NULL,	
	hora_fin	DATETIME HOUR TO
	MINUTE NOT NULL,	
estado	BOOLEAN NOT NULL	

*tabla 17 – tabla ‘servicios’*

tabla	servicios_servidos	
descripción	<p>Esta tabla almacena registros de las sesiones de servicios. Una sesión es la ocurrencia de cuando la sección de comedores despliega la logística para 'servir' cada servicio.</p> <p>Un registro tiene dos campos que guardan los estados 'esta_abierto' y 'se_presto'. El primero indica si la sesión del servicio está aún abierta, y en la tabla sólo puede haber un sólo servicio abierto. 'se_presto' indica si esa sesión es válida o no para contabilizar las asistencias y fallas de los beneficiarios al servicio (un servicio puede 'servirse' pero quizá un evento fortuito obligó a un cierre adelantado, luego, para no contabilizar fallas se asigna 'se_presto' en N)</p> <p>Se relaciona con la tabla 'servicios' de la cual trae su llave primaria.</p>	
campos	llave primaria codigo_servicio fecha_servicio_servido	INTEGER NOT NULL, DATE NOT NULL,
	se_presto esta_abierto hora_interrupcion observaciones	VARCHAR(2) NOT NULL, VARCHAR(2) NOT NULL, DATETIME DAY TO SECOND, TEXT

*tabla 18 – tabla 'servicios\_servidos'*

tabla	servicios_solicitados	
Descripción	<p>Esta tabla almacena la información de los servicios que un inscrito solicitó. Entre esa información, se encuentra si el servicio está adjudicado y si está habilitado.</p> <p>Se relaciona con las tablas 'servicios' e 'inscripciones' de las cuales trae sus llaves primarias.</p>	
Campos	llave primaria	
	ano	SMALLINT NOT NULL,
	semestre	SMALLINT NOT NULL,
	documento_usuario	DECIMAL(13,0) NOT NULL,
	codigo_tipo_documento	SMALLINT NOT NULL,
	codigo_servicio	INTEGER NOT NULL,
	servicio_adjudicado	VARCHAR(2) NOT NULL,
	servicio_habilitado	VARCHAR(2) NOT NULL,
	servicio_en_espera	VARCHAR(2) NOT NULL,
	observacion	VARCHAR(200)

*tabla 19 – tabla 'servicios\_solicitados'*

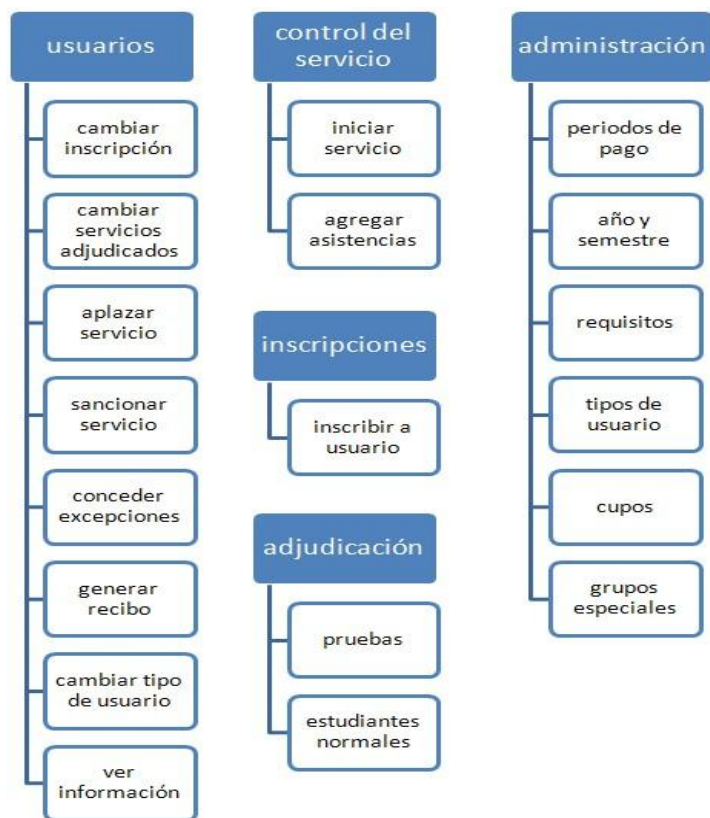
Tabla	tipos_usuario
Descripción	En esta tabla se guardan los tipos de usuario (beneficiarios) del servicio de comedores. Un tipo de usuario existe como abstracción de varios usuarios que sus condiciones le permiten tener otros requisitos.
Campos	<p>llave primaria  codigo_tipo_usuario SERIAL NOT NULL,</p> <p>nombre_tipo_usuario VARCHAR(150) NOT NULL,  descripcion_tipo_usuario VARCHAR(200) NOT NULL,  codigo_tipo_documento SMALLINT NOT NULL,  tabla_consulta VARCHAR(100) NOT NULL</p>

*tabla 20 – tabla 'tipos\_usuario'*

### 3.3.6 Diseño del prototipo no funcional

El proceso de construcción del prototipo inicial comenzó desde el momento en que se elaboraron los diagramas de casos de uso, y cuando se tuvo un esbozo de los módulos del sistema. El prototipo sería como una carta de navegación en el transcurso del desarrollo del sistema, y como parte importante en la realimentación y refinamiento de los prototipos siguientes.

El prototipo no funcional se creó como una página web básica sin ningún tipo de programación de fondo, usando una herramienta especial para el prototipado de aplicaciones web (Axure RP, en su versión trial) que mostraría gráficamente el esquema de navegación del sistema, de ésta manera se definieron detalles de alta importancia tales como la información que el usuario quiere ver, la que desea ocultar, el contenido de cada una de las vistas y los datos que se le van a solicitar a cada uno de los usuarios que interactuarían con el sistema.




*figura 9 – menú del prototipo no funcional*

Por cuestiones prácticas se presentan sólo algunos pantallazos del prototipo no funcional. El prototipo completo se encuentra en un medio digital adjunto a este documento.

Vista del formulario que se despliega al acceder al menú 'inscripciones -> inscribir usuario'. Permite al usuario inscribir a un beneficiario al sistema de comedores:

# inscribir usuario

tipo de usuario  

[búsqueda avanzada](#)

Fidel David Jiménez Martínez

## servicios a solicitar

desayuno

almuerzo

comida

*figura 10 – prototipo caso de uso 'crear inscripción'*

Vista del formulario que se despliega al acceder al menú 'usuarios -> cambiar servicios adjudicados', que se encarga primero de la búsqueda del inscrito. Una vez elegido el inscrito, se muestra el pantallazo 'cambiar servicios adjudicados'.

## 5 cambiar servicios adjudicados

joaquin [búsqueda avanzada](#)

buscar

se han encontrado 3 resultados

joaquín mariano mosquera y arboleda	2040051
joaquín josé mosquera vélez	2247861
joaquín elías mosquera zúñiga	2489921

*figura 11 – prototipo del caso de uso 'búsqueda de inscrito'*

El formulario 'cambiar servicios adjudicados' que permite al administrador adjudicar o des-adjudicar los servicios de un inscrito manualmente.

Joaquín Mariano Mosquera

estudiante normal

código: 2040051

### cambiar servicios adjudicados

servicios adjudicados

desayuno

almuerzo

comida

adjudicar



*figura 12 – prototipo del caso de uso 'modificar estado de servicios solicitados'*

Vista del formulario que se despliega al acceder al menú 'adjudicación -> pruebas', que permite administrar los cupos a adjudicar.

## prueba de adjudicación *por raciones*

cantidad **desayunos**  [ver resultados](#)

cantidad **almuerzos**  [ver resultados](#)

cantidad **comidas**  [ver resultados](#)

[ver resultados todas las comidas](#)

*Figura 13 – prototipo del caso de uso 'gestionar cupos de servicio'*

Vista del formulario que se despliega al acceder al menú 'adjudicación -> estudiantes normales', que permite adjudicar los cupos normales.

## adjudicación de cupos normales

cantidad desayunos 670

cantidad almuerzos 1565

cantidad comidas 1267

adjudicar

*figura 14 – prototipo del caso de uso 'gestionar cupos de servicio'*

Vista del formulario que se despliega al acceder al menú 'control -> iniciar servicio'. Permite al administrador iniciar una sesión para que pueda agregar asistencias.

## control de servicio



sesión de servicio

servicio

fecha  

*figura 15 – prototipo del caso de uso 'gestionar servicios servidos'*



Vista del formulario que se despliega al acceder al menú 'administración -> tipos de usuario'. Permite la gestión de crear, modificar, y eliminar registros de la tabla tipos de usuario.

## administrar tipos de usuario

nombre

descripción

**crear**

nombre	acción
estudiante normal	
empleado	

*figura 16 – prototipo del caso de uso 'hacer mantenimiento de tipos de usuario'*

### 3.3 Diseño e implementación de los módulos

Después del análisis de los requisitos, y de planificar los módulos, se elaboraron los diagramas de casos de uso agrupados en los módulos definidos. A continuación se expone el desarrollo de cada módulo, especificando cada caso de uso, y mostrando el último prototipo funcional. El diseño final del menú es el siguiente, detallando cada submenú:

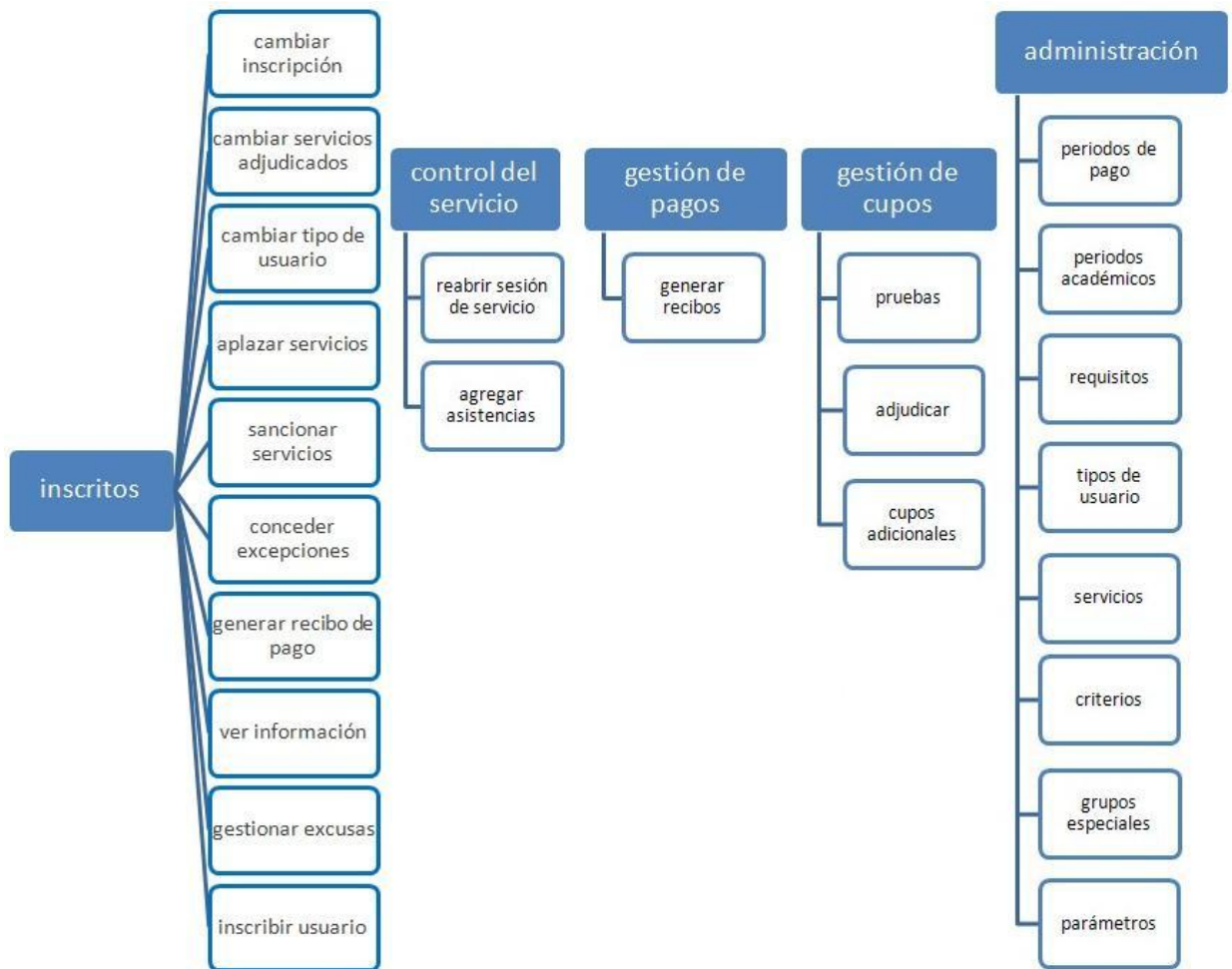


figura 17 – submenús del sistema

### 3.4.1 Componentes reutilizables

#### Casos de uso

Este módulo implementa los casos de uso de realizar la búsqueda de un inscrito, lo cual puede hacerse mediante su nombre, partes de su nombre, su código o a través de una búsqueda avanzada que permite especificar otros parámetros de búsqueda. Una vez seleccionado un inscrito, la interfaz de búsqueda desaparece y da paso a otros módulos, recordando el inscrito seleccionado. En algunos módulos, a interfaz de búsqueda permite seleccionar a varios usuarios para realizar un cambio a todos ellos.

A continuación una lista de los casos de uso que pertenecen al módulo, seguida por su descripción detallada. Algunos de ellos fueron ilustrados además con capturas de pantalla:

- Búsqueda de inscrito
- Búsqueda de usuario

#### Búsqueda de inscrito

caso de uso	Búsqueda de inscrito
actores	administrador, operario
propósito	Buscar a un inscrito del servicio de comedores.
resumen	El actor ingresa el nombre o el código, y el sistema presenta los resultados. El actor elige al inscrito o a los inscritos (depende del caso de uso que use éste)
precondiciones	Se requiere que el actor esté validado correctamente.

flujo principal	<p>Se presenta el formulario de búsqueda.El actor ingresa el nombre o el código y se realiza la búsqueda.</p> <p>Dependiendo del caso de uso, el actor puede elegir a un inscrito o a varios, para posteriormente hacer la operación respectiva.</p>
subflujos	<p><b>Subflujo 1:</b> en caso que el actor necesite elegir varios inscritos, el sistema mostrará una lista con los inscritos elegidos hasta el momento.</p> <p>El actor puede eliminar inscritos de esa lista, y terminar la búsqueda para hacer la operación respectiva.</p>
excepciones	<p>Si la búsqueda no da resultados, se despliega el mensaje.</p>

*tabla 21 – caso de uso ‘búsqueda de inscrito’*

# gestionar inscripción

Modificar inscripción

The image shows two sequential screenshots of a web application interface. The top screenshot shows a search form with the title 'criterios de búsqueda'. It contains a text input field labeled 'nombre o código', a 'busqueda avanzada' link with a question mark icon, and two buttons: 'buscar' and 'Cancelar'. A large blue arrow points downwards from this form to the second screenshot. The second screenshot shows the same search form, but now the input field contains the value '2040027'. Below the search form is a section titled 'resultados de la búsqueda' which contains a table with one row of results. The table has three columns: 'documento', 'nombre', and 'tipo de usuario'. The row contains the values '2040027', 'fidel david jimenez martinez', and 'estudiante normal'. Below the table is a pagination control showing '1 registros' and navigation buttons.

documento	nombre	tipo de usuario
2040027	<a href="#">fidel david jimenez martinez</a>	estudiante normal

figura 18 – captura de interfaz de 'búsqueda de inscrito' por documento

Éste caso de uso se invoca desde muchos otros, pues ofrece la funcionalidad de buscar un inscrito y le permite al administrador, una vez seleccionado el inscrito, operar sobre él según el caso de uso lo haya invocado.

El sistema despliega el un formulario para que el actor que usa el caso de uso, ingrese el documento del inscrito, o parte de su nombre. Una vez ejecutado el proceso de búsqueda, se despliega los inscritos que tengan coincidencias con los datos ingresados.

En la imagen de arriba, se muestra una búsqueda ingresando el documento. Como el documento es único (código de estudiante) se muestra sólo el inscrito con ese código.



*figura 19 - captura de interfaz de 'búsqueda de inscrito' por nombre*

En la búsqueda por nombre, despliega la lista de los inscritos que tengan coincidencias en el nombre (en la imagen, con la letra 'f'). El operario selecciona al inscrito de la lista, y luego lo entrega al caso de uso que lo haya invocado.

caso de uso	Búsqueda de usuario
actores	administrador, operario

propósito	Buscar un aspirante (en su mayoría estudiantes) basado en los criterios de búsqueda establecidos por el actor.
resumen	El actor elige los parámetros de búsqueda y el sistema le muestra una lista de registros.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	Se presenta el formulario de búsqueda en una ventana modal.  El actor debe escoger el tipo de usuario, y llenar los campos (primer nombre, segundo nombre, primer apellido, segundo apellido) que desee. Se muestran los resultados, el actor selecciona y se cierra la ventana modal.
subflujos	Ninguno.
excepciones	Ninguna.

*tabla 22 – caso de uso ‘búsqueda de usuario’*

Éste caso de uso se invoca desde otros, y ofrece la funcionalidad de buscar un posible beneficiario del servicio (estudiantes y empleados). Muestra un formulario en una ventana modal, con 5 criterios de búsqueda: el tipo de usuario, los dos nombres y los dos apellidos.

gestionar inscripción

criterios de búsqueda

tipo de usuario

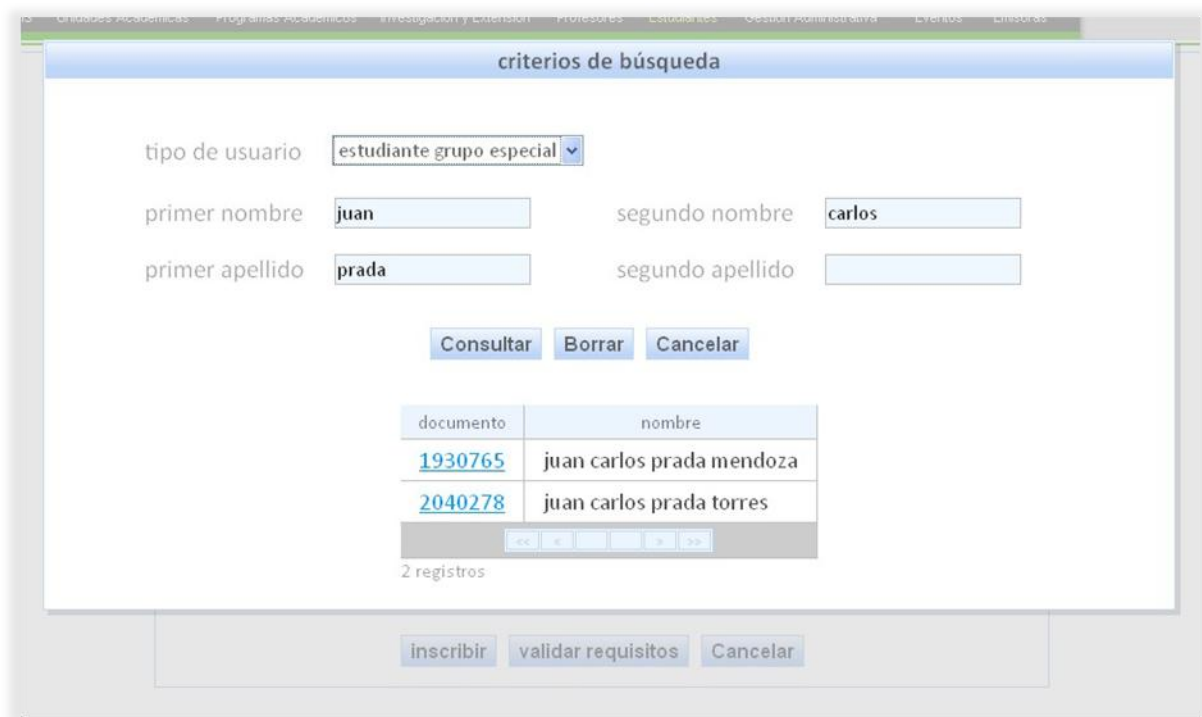
primer nombre  segundo nombre

primer apellido  segundo apellido

comida

*figura 20 – captura de interfaz de 'búsqueda de usuario'*

el botón 'borrar' limpia todos los campos del formulario, y el botón 'cancelar' cierra la ventana modal. El botón consultar, una vez llenos los campos, busca y despliega una lista con los usuarios (estudiantes o empleados, inscritos o no en el servicio). Cuando se escoja el usuario de la lista, la ventana modal se cierra y el caso de uso que invocó a 'búsqueda de usuario' tendrá la información del usuario inscrito. (ver imagen)



*figura 21 – captura de interfaz de 'búsqueda de usuario' con resultados*

### 3.4.2 Inscripciones

#### Casos de uso

Este módulo implementa los casos de uso que permiten a los usuarios inscribirse en el servicio y modificar su inscripción, y a los operarios de Bienestar Universitario administrar la información de los inscritos o crear inscritos nuevos de forma manual; un servicio que se presta, por ejemplo, a los estudiantes invidentes.

Adicionalmente cubre los casos de uso para generación de recibos por lote o únicos, gestión de excusas por inasistencia, aplazamiento de servicios por motivos de fuerza mayor, permisos especiales, y sanciones.

A continuación una lista de los casos de uso que pertenecen al módulo, seguida por su descripción detallada. Algunos de ellos fueron ilustrados además con capturas de pantalla:

- gestionar inscripciones
- crear inscripción
- modificar inscripción
- modificar estado de servicios solicitados
- modificar tipo de usuario
- ver detalles de inscrito
- gestionar recibos
- gestionar excusas
- gestionar sanciones
- gestionar servicios aplazados
- gestionar requisitos exceptuados

### gestionar inscripciones

caso de uso	Gestionar inscripciones
actores	administrador, operario
propósito	realizar diversas operaciones para llevar a cabo el proceso de inscripciones.
resumen	Le ofrece al actor realizar diversas operaciones para administrar el proceso de inscripciones .

precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	Le permite al actor ejecutar los siguientes casos de uso: "Crear inscripción" "Modificar inscripción" "Modificar tipo de usuario" "Modificar estado de servicios solicitados"
subflujos	Ninguno.
excepciones	Ninguna.

*tabla 23 – caso de uso ‘gestionar inscripciones’*

## crear inscripción

caso de uso	Crear inscripción
actores	administrador, operario, beneficiario
propósito	Inscribir a un aspirante al proceso de adjudicación del servicio de comedores.
resumen	El actor elige los servicios que desea solicitar, y opcionalmente puede ver de los requisitos que el aspirante incumple.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	<p>Se presenta el formulario de inscripción, donde se muestran los servicios.</p> <p>Si no es 'beneficiario', el actor puede elegir el tipo de usuario, y debe ingresar el documento del aspirante a beneficiario. En caso de no saber el documento del usuario, puede usar el caso de uso "búsqueda de usuario".</p> <p>El actor tiene la opción de revisar cuales requisitos incumple.</p>
subflujos	Ninguno.
excepciones	Si el documento del aspirante no corresponde a ninguna persona, muestra un mensaje al actor.

*tabla 24 – caso de uso 'crear inscripción'*

## gestionar inscripción

**datos de usuario**

tipo de usuario  \* [?](#)

codigo estudiante  \* [búsqueda avanzada](#) [?](#)

servicios a solicitar

desayuno

almuerzo

comida



## gestionar inscripción

**datos de usuario**

tipo de usuario  \* [?](#)

codigo estudiante  \* [búsqueda avanzada](#) [?](#)

nombre **fidel david jimenez martinez**

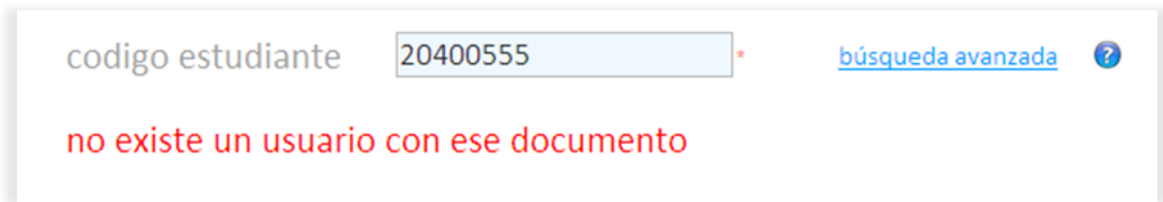
servicios a solicitar

desayuno

almuerzo

comida

en la interfaz para crear una inscripción el operario debe indicar un código de estudiante. El sistema automáticamente busca los datos de esa persona y, si existe, muestra el nombre del estudiante. De lo contrario, muestra un error:



The screenshot shows a search interface with the following elements:

- Label: "codigo estudiante"
- Input field: "20400555"
- Link: "búsqueda avanzada" with a question mark icon
- Error message: "no existe un usuario con ese documento" (displayed in red text)

*figura 22 – captura de interfaz de 'crear inscripción' con un documento que no pertenece a ningún estudiante*

Adicionalmente el operario puede indicar qué servicios solicitará el nuevo inscrito. En el momento de realizar la inscripción, si el estudiante cumple con los requisitos mínimos para inscribirse a comedores y solicitar servicios, se muestra un aviso de confirmación.



*figura 23 – captura de la interfaz de 'crear inscripción' con el mensaje de validación de requisitos*

### Modificar inscripción

caso de uso	Modificar inscripción
actores	administrador, operario, beneficiario
propósito	cambiar los servicios solicitados en una inscripción.

resumen	Se presenta al actor la inscripción realizada, y permite cambiar los servicios solicitados.
precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso 'búsqueda de inscrito' (en caso que el actor no sea 'beneficiario').
flujo principal	<p>Si el actor no es 'beneficiario', se ejecuta el caso de uso 'búsqueda de inscrito', y una vez elegido el inscrito, se muestra la información en un formulario y el actor puede cambiar los servicios solicitados.</p> <p>En caso de que el actor no elija ningún servicio, el sistema eliminará la inscripción y se presentará un aviso de confirmación.</p> <p>Si el actor es 'beneficiario' se muestra directamente la información de la inscripción. El resto del flujo es igual.</p>
subflujos	Ninguno.
excepciones	<b>excepción 1:</b> si al desagregar un servicio, éste se encuentra adjudicado, se deshace la operación y se muestra un mensaje de error.

*tabla 25 – caso de uso 'modificar inscripción'*

### Modificar tipo de usuario

caso de uso	Modificar tipo de usuario
actores	administrador, operario
propósito	Cambiar el tipo de usuario de uno o más inscritos.
resumen	Se eligen uno o más inscritos y luego se selecciona el nuevo tipo de usuario al que se van a cambiar los inscritos elegidos.

precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso 'búsqueda de inscrito' habiendo elegido uno o más inscritos.
flujo principal	<p>Se despliega el formulario, con los tipos de usuarios para que el actor escoja.</p> <p>El sistema cambia el tipo de usuario de los seleccionados mientras el documento del inscrito no sea incompatible con el tipo de usuario.</p> <p>Al final despliega una lista con los inscritos a los que se les modificó el tipo de usuario correctamente.</p>
subflujos	<b>Subflujo 1:</b> si se escoge el tipo de usuario 'estudiante grupo especial' se despliega el listado de todos los grupos especiales para asociar a los inscritos con el grupo especial elegido.
excepciones	Ninguna.

*tabla 26 – caso de uso 'modificar tipo de usuario'*

## Administración de tipos de usuario

Modificar tipo de usuario

nombre	<input type="text" value="estudiante normal"/> *	?
descripción	<input type="text" value="."/> *	?
tipo de documento	<input type="text" value="codigo estudiante"/> *	?
tabla de consulta	<input type="text" value="GnrEstudianteUis"/> *	?

*figura 24 – captura de interfaz de 'modificar tipo de usuario'*

La interfaz de modificar tipo de usuario permite al operario crear nuevos tipos de usuario y relacionarlos con un tipo de documento. Adicionalmente, puede escribirse una descripción de los tipos de usuario. Algunos ejemplos de tipos de usuario son:

- estudiante
- estudiante de primer nivel
- estudiante en trabajo de grado
- estudiante miembro de selección deportiva
- estudiante miembro de grupo artístico

- empleado UIS

Diferentes tipos de usuario tienen diferentes requisitos para recibir el servicio de comedores.

### Modificar el estado de servicios solicitados

caso de uso	Modificar el estado de servicios solicitados
actores	administrador, operario
propósito	Cambiar el estado los servicios solicitados de una inscripción.
resumen	Se presenta al actor la información de una inscripción y permite adjudicar o des-adjudicar servicios solicitados.
precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso 'búsqueda de inscrito'.
flujo principal	Se muestra la información en un formulario y el actor puede adjudicar o des-adjudicar los servicios solicitados del inscrito.
subflujos	Ninguno.
excepciones	Si no hay cupos para algún servicio que se desea adjudicar, se muestra un mensaje de alerta.

*tabla 27 – caso de uso 'modificar estado de servicios solicitados'*

## gestionar inscripción

### Modificar inscripción

datos de inscrito

fidel david jimenez martinez

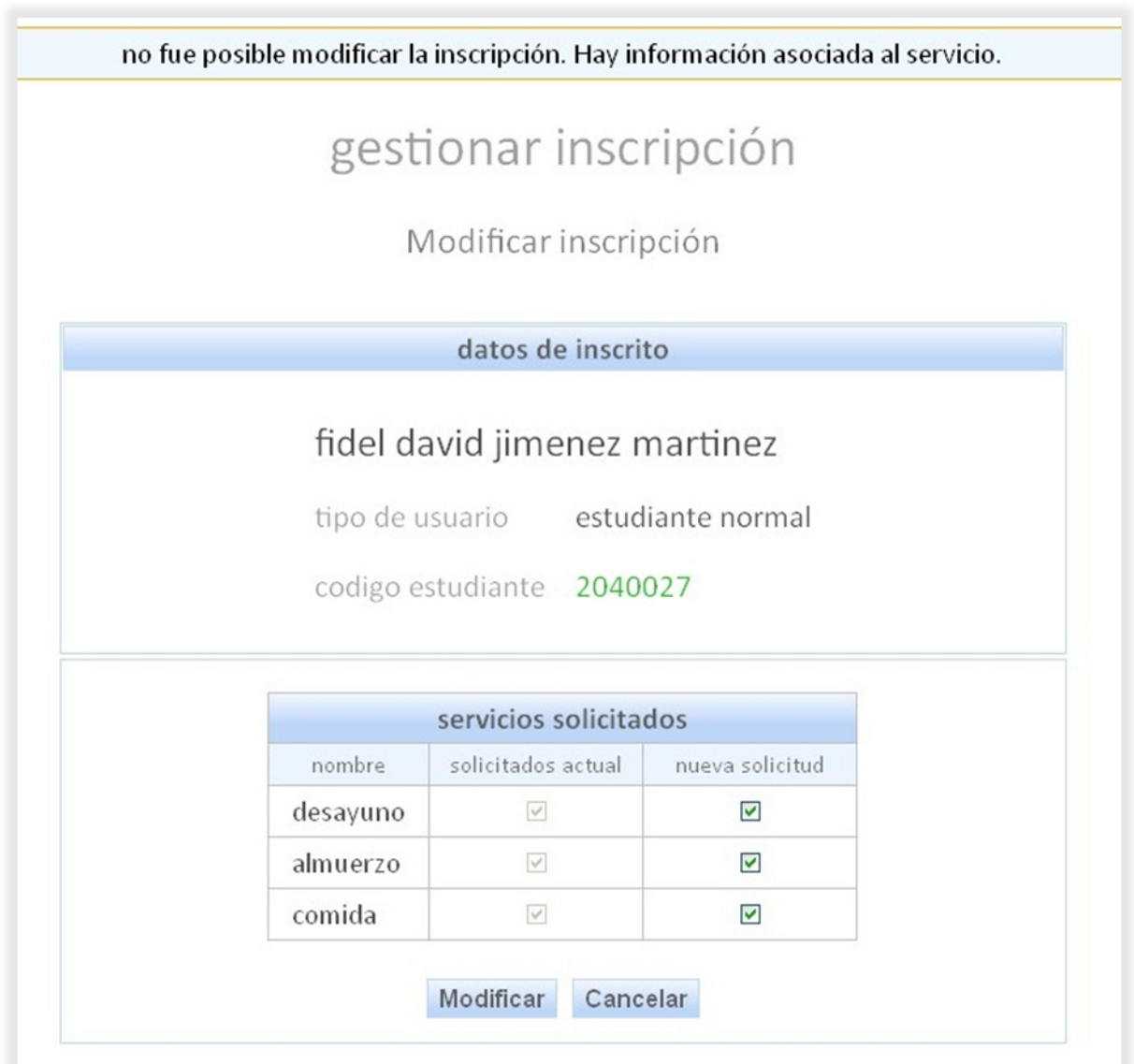
tipo de usuario      estudiante normal

codigo estudiante    2040027

servicios solicitados		
nombre	solicitados actual	nueva solicitud
desayuno	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
almuerzo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
comida	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

*figura 25 – captura de la interfaz de ‘modificar inscripción’*

En la interfaz para modificar una inscripción, el operario puede cambiar los servicios que solicita el inscrito. Si el operario no marca ningún servicio, se entiende que el usuario en cuestión ya no estará inscrito, y se borra su inscripción. Aunque si el inscrito ya tiene información asociada en el sistema no permite eliminarlo y despliega este mensaje:



*figura 26 – captura de interfaz de 'modificar inscripción' con el mensaje de validación*

### Ver detalles de inscrito

caso de uso	Ver detalles de inscrito
actores	administrador, operario, beneficiario

propósito	ver los detalles de un inscrito.
resumen	Se presenta al actor el inscrito seleccionado, y permite ver su información.
precondiciones	Se requiere que el actor esté validado correctamente, y que se haya ejecutado el caso de uso 'búsqueda de inscrito' (en caso que el actor no sea 'beneficiario').
flujo principal	<p>Si el actor no es 'beneficiario', se ejecuta el caso de uso 'búsqueda de inscrito', y una vez elegido el inscrito, se muestra su información básica para comedores, el cumplimiento histórico de sus requisitos, su historial de asistencias, etc.</p> <p>Si el actor es 'beneficiario' se muestra directamente la información de la inscripción. El resto del flujo es igual.</p>
subflujos	Ninguno.
excepciones	Ninguna.

*tabla 28 – caso de uso 'ver detalles de inscrito'*

## Gestionar recibos

caso de uso	Gestionar recibos
actores	administrador, operario, beneficiario
propósito	Generar, ver y anular los recibos a un inscrito.
resumen	El actor selecciona a un inscrito y el sistema permite crear, consultar y anular los recibos.
precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso 'búsqueda de inscrito' (en caso que el actor no sea 'beneficiario').

flujo principal	<p>Si el actor no es 'beneficiario' Se muestra la información de la inscripción. El actor escoge el periodo de pago, la fecha límite de pago y otros datos del recibo para luego generarlo (para que el inscrito posteriormente lo consulte e imprima).</p> <p>Si el actor es 'beneficiario', se muestra un enlace que permite generar el recibo en formato PDF para su impresión.</p> <p>Se muestra también un listado con los recibos generados en un semestre académico, donde se muestran las opciones de 'ver' y 'anular' (si el actor no es 'beneficiario') cada registro de la lista.</p>
subflujos	<p><b>Subflujo 1:</b> si el inscrito no tiene ningún servicio adjudicado, muestra un mensaje de la imposibilidad de generar el recibo.</p> <p><b>Subflujo 2:</b> si el actor elije la opción 'ver' de algún registro de la lista, se despliega la información detallada del recibo elegido.</p> <p><b>Subflujo 3:</b> si el actor elije la opción 'anular' de algún registro de la lista, se despliega un formulario para que se pueda anular el recibo elegido.</p>
excepciones	Ninguna.

*tabla 29 – caso de uso 'gestionar recibos'*

## gestionar recibo de pago

### Crear recibo de pago

fidel david jimenez martinez

tipo de usuario estudiante normal

codigo estudiante 2040027

semestre académico 2 del año 2010

periodo de pago

5 (actual) \*



fecha límite de pago



\*



cantidad de días a facturar

32 \* de 32



observaciones

\*



generar

Cancelar

### lista de recibos

número	periodo de pago	número de servicios	estado	acción
19	5	3	válido	
22	6	3	válido	

«< >>»

2 registros

figura 27 – captura de la interfaz de 'gestionar recibos'

La interfaz de gestionar recibos de pago permite crear nuevos recibos a un inscrito seleccionado previamente con el buscador, ver los que ya tiene y eliminar recibos anteriores. La interfaz de ver un recibo de pago es la siguiente:

## gestionar recibo de pago

**ver detalle recibo de pago**

semestre académico **2** del año **2010**

periodo de pago                      5

número                                    24

fecha de expedición                21 de julio de 2011

fecha límite de pago                27 de julio de 2011

fecha de pago                         no ha pagado

cantidad de días facturados      32

observaciones                        al estudiante se le adjudicaron los servicios desués del proceso con la psicóloga y con autorización de la doctora consuelo.

estado                                    válido

servicio	valor de la ración	subtotal
comida	\$ 860	\$ 27.520
almuerzo	\$ 890	\$ 28.480
desayuno	\$ 350	\$ 11.200
<b>total</b>		<b>\$ 67.200</b>

Regresar

*figura 28– captura interfaz de ‘ver recibo de pago’*

## Gestionar aplazamiento de servicios

caso de uso	Gestionar aplazamiento de servicios
actores	administrador, operario
propósito	Crear, ver y eliminar aplazamientos de los servicios adjudicados a un inscrito.
resumen	El actor selecciona a un inscrito y el sistema permite crear, ver y eliminar los aplazamientos.
precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso 'búsqueda de inscrito'.
flujo principal	Se muestra la información de la inscripción. Se despliega el formulario, donde el actor escoge el periodo de pago y los servicios a aplazar. También se muestra un listado de todos los aplazamientos, donde se muestran las opciones de 'ver' y 'eliminar' cada registro de la lista.
subflujos	<b>Subflujo 1:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada del aplazamiento elegido. <b>Subflujo 2:</b> si el usuario elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario para que se pueda eliminar el aplazamiento elegido.
excepciones	Ninguna.

*tabla 30 – caso de uso 'gestionar aplazamiento de servicios'*

## Administración de aplazamientos

### Crear aplazamiento

fidel david jimenez martinez

tipo de usuario    **estudiante normal**

codigo estudiante    **2040027**

periodo de pago    5 (actual) ▼ \* ?

servicios

desayuno

almuerzo

comida

observaciones

la doctora consuelo autorizó el aplazamiento

*figura 29 – captura de interfaz de 'crear aplazamiento'*

La interfaz de crear aplazamientos permite al operario de Bienestar Universitario aplazar un servicio a un inscrito que tenga dificultades para pagarlo o para asistir

durante un periodo de pago de comedores. El sistema no va a requerir la asistencia de ese inscrito durante el periodo que hays sido aplazado, de forma que no acumule fallas. Cuando termine su aplazo el inscrito tendrá que pagar sus servicios y asistir a comedores. Cuando un aplazamiento es creado exitosamente el sistema notifica al operario con este mensaje:



The screenshot shows a web interface with a light blue header bar containing the text "se han creado correctamente". Below this is a large heading "Administración de aplazamientos". Underneath is a form titled "Crear aplazamiento" with a blue header. The form contains the following fields:

- Nombre: fidel david jimenez martinez
- tipo de usuario: estudiante normal
- codigo estudiante: 2040027
- periodo de pago: 5 (actual) [dropdown arrow] \* [help icon]

*figura 30 – captura de interfaz de 'crear aplazamiento' con mensaje de validación*

La interfaz permite también ver los aplazamientos que un inscrito tenga previamente, y eliminarlos:

lista de aplazamientos		
servicio	periodo de pago	acción
desayuno	2	
desayuno	5	

2 registros

*figura 31 – captura de interfaz de ‘ver aplazamientos’*

### Gestionar requisitos exceptuados

caso de uso	Gestionar requisitos exceptuados
actores	administrador, operario
propósito	Crear, ver y eliminar requisitos exceptuados a un inscrito.
resumen	El actor selecciona a un inscrito y el sistema permite crear, ver y eliminar requisitos exceptuados.
precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso ‘búsqueda de inscrito’.
flujo principal	Se muestra la información de la inscripción. Se despliega el formulario, donde el actor escoge los servicios y los criterios, para crear los requisitos a exceptuar. También se muestra un listado de todos los requisitos exceptuados, donde se muestran las opciones de ‘ver’ y ‘eliminar’ cada registro de la lista.

subflujos	<p><b>Subflujo 1:</b> si el actor elije la opción 'ver' de algún registro de la lista, se despliega la información detallada del requisito exceptuado elegido.</p> <p><b>Subflujo 2:</b> si el actor elije la opción 'eliminar' de algún registro de la lista, se despliega un formulario para que se pueda eliminar el requisito exceptuado elegido.</p>
excepciones	Ninguna.

*tabla 31 – caso de uso 'gestionar requisitos exceptuados'*


## gestionar requisitos exceptuados

### Crear requisito exceptuado

fidel david jimenez martinez

tipo de usuario      estudiante normal

codigo estudiante    2040027

servicios    todos  

critérios a exceptuar

- promedio acumulado
- numero creditos matriculados
- numero creditos aprobados
- numero sanciones
- numero semestres con comedores
- numero beneficios incompatibles
- numero auxiliaturas

observaciones

la documentación se encuentra en el folio  
23, autorizado por la psicóloga

Guardar

Cancelar

figura 32 – captura de interfaz 'gestionar requisitos exceptuados'

La interfaz de gestionar requisitos exceptuados permite indicarle al sistema que no revise ciertos requisitos para un inscrito seleccionado previamente con el buscador. Adicionalmente, el operario de Bienestar Universitario puede exceptuar la revisión de requisitos del inscrito únicamente para ciertos servicios; por ejemplo para el desayuno y el almuerzo, pero no para la comida.

Cuando se seleccionan todos los servicios, el sistema crea o modifica una lista de requisitos que no deben revisarse en todos los servicios de comida que reciba de un inscrito:



se han creado (y quizá modificado) los registros.

## gestionar requisitos exceptuados

**Crear requisito exceptuado**

fidel david jimenez martinez

tipo de usuario      estudiante normal

codigo estudiante    2040027

*figura 33 – captura de interfaz de 'crear requisitos exceptuados' con mensaje de validación*

La interfaz también permite ver una lista de requisitos exceptuados anteriormente para otros servicios y eliminarlos o revisar sus detalles:

lista de requisitos exceptuados			
tipo de usuario	criterio	servicio	acción
estudiante normal	numero auxiliaturas	comida	 
estudiante normal	numero beneficios incompatibles	almuerzo	 
estudiante normal	numero beneficios incompatibles	comida	 
estudiante normal	numero creditos aprobados	desayuno	 
estudiante normal	numero creditos aprobados	comida	 
estudiante normal	numero creditos matriculados	comida	 
estudiante normal	numero sanciones	desayuno	 
estudiante normal	numero sanciones	almuerzo	 
estudiante normal	numero sanciones	comida	 
estudiante normal	numero semestres con comedores	almuerzo	 

13 registros

figura 34 – captura de interfaz de ‘ver lista de requisitos exceptuados’

### Gestionar sanciones

caso de uso	Gestionar sanciones
actores	administrador, operario
propósito	Crear, ver, modificar y eliminar sanciones de un inscrito.
resumen	El actor selecciona a un inscrito y el sistema permite crear, ver, modificar y eliminar las sanciones.

precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso 'búsqueda de inscrito'.
flujo principal	Se muestra la información de la inscripción. Se despliega el formulario, donde el actor asigna las fechas de inicio y fin de la sanción, y escoge los servicios a sancionar (entre los servicios solicitados). También se muestra un listado de todas las sanciones, donde se muestran las opciones de 'ver', 'modificar' y 'eliminar' cada registro de la lista.
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada de la sanción.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega un formulario para que se pueda modificar la sanción.</p> <p><b>Subflujo 3:</b> si el actor elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario para que se pueda eliminar la sanción.</p>
excepciones	Ninguna.

*tabla 32 – caso de uso 'gestionar sanciones'*

La interfaz de gestionar sanciones permite a un operario sancionar a un inscrito durante un periodo de pago, varios periodos de pago, u opcionalmente entre dos fechas arbitrarias. Adicionalmente el operario puede agregar una observación a la sanción. La interfaz también permite ver los detalles de sanciones aplicadas anteriormente a un inscrito.

Las sanciones pueden igualmente eliminarse para permitir que un inscrito sancionado recupere sus derechos de comedores.

## Gestionar excusas

caso de uso	Gestionar excusas
actores	administrador, operario, jefe de operación, beneficiario
propósito	Crear (solicitar, en caso de que el actor sea 'beneficiario'), modificar y eliminar excusas de un inscrito.
resumen	El actor selecciona a un inscrito y el sistema permite crear (solicitar, en caso de que el actor sea 'beneficiario'), aprobar y eliminar las excusas.
precondiciones	Se requiere que el actor esté validado correctamente, y requiere que se haya ejecutado el caso de uso 'búsqueda de inscrito' (en caso que el actor no sea 'beneficiario').
flujo principal	Se muestra la información de la inscripción. Se despliega el formulario, donde el actor ingresa toda la información de una excusa. También se muestra un listado de todas las sanciones, donde (si el actor no es 'beneficiario') se muestran las opciones de 'modificar' y 'eliminar' cada registro de la lista.
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega la información detallada de la excusa, donde se permite aprobarla o desaprobala.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario para que se pueda eliminar la excusa.</p>
excepciones	Si una excusa a eliminar es para un día que ya pasó, muestra un mensaje de error e impide la operación.

*tabla 33 – caso de uso 'gestionar excusas'*

### 3.4.3 Adjudicación

#### Casos de uso

Este módulo implementa los casos de uso que ejecutan los procesos de adjudicación de servicios en lote y los que administran la información necesaria para que esos procesos se lleven a cabo.

A continuación una lista de los casos de uso que pertenecen al módulo, seguida por su descripción detallada. Algunos de ellos fueron ilustrados además con capturas de pantalla:

- gestionar cupos de servicio
- gestionar cupos de prueba
- adjudicar cupos de servicio
- modificar cupos adicionales

#### Gestionar cupos de servicio

caso de uso	Gestionar cupos de servicio
actores	Administrador
propósito	realizar diversas operaciones para llevar a cabo el proceso de adjudicación de cupos.
resumen	Le ofrece al actor realizar diversas operaciones para administrar el proceso de adjudicación de cupos .

precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	Le permite al actor ejecutar los siguientes casos de uso: "Adjudicar cupos de servicio" "modificar cupos de adicionales" "Gestionar cupos de prueba"
subflujos	Ninguno.
excepciones	Ninguna.

*tabla 34 – caso de uso ‘gestionar cupos de servicio’*

### Gestionar cupos de prueba

caso de uso	Gestionar cupos de prueba
actores	administrador, operario
propósito	Asignar y modificar los cupos de prueba normales de los servicios para un año y semestre académico.
resumen	El actor establece la cantidad de cupos de prueba (para los beneficiarios normales) para cada servicio activo a adjudicar.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	Se muestra un formulario para que el actor ingrese las cantidades, las cuales se pueden modificar mientras no se hayan adjudicado los cupos.
excepciones	Ninguna.

*tabla 35 – caso de uso ‘gestionar cupos de prueba’*

## Adjudicar cupos de servicio

caso de uso	Adjudicar cupos de servicio
actores	administrador, jefe de operación
propósito	Adjudicar los servicios de comedores a los inscritos que los solicitaron.
resumen	Permite al actor ejecutar el proceso de adjudicación.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	Se muestra al actor las opciones siguientes:  Realizar el proceso de adjudicación de cupos normales, basados en los cupos de pruebas asignados finalmente.  Realizar el proceso de adjudicación de cupos especiales.
subflujos	Ninguno.
excepciones	Ninguna.

*tabla 36 – caso de uso ‘adjudicar cupos de servicio’*

## Modificar cupos adicionales

caso de uso	Modificar cupos adicionales
actores	administrador, jefe de operación
propósito	Adicionar y quitar cupos para su ` posible adjudicación.

resumen	El actor establece la cantidad de cupos adicionales.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	Se muestra un formulario para que el actor ingrese las cantidades de los cupos adicionales, y los modifique, agregando o quitando.
subflujos	Ninguno.
excepciones	Ninguna.

*tabla 40 – caso de uso ‘modificar cupos adicionales’*

The image shows a screenshot of a web application interface. At the top, the text 'gestión de cupos de servicios' is displayed. Below it is a modal dialog box titled 'Modificar cupos'. The dialog contains three rows of data, each with a label and a numeric input field with a small up/down arrow icon to its right:

- desayuno: 10
- almuerzo: 15
- comida: 12

At the bottom of the dialog, there are two buttons: 'Modificar' and 'Cancelar'.

*figura 35 – captura de interfaz de ‘administrar cupos de prueba de servicios’*

La interfaz de administración de cupos permite establecer un número tentativo de cupos para cada uno de los servicios de comedores que ofrece Bienestar Universitario. Después de llegar a una decisión final sobre la cantidad de cupos que se adjudicarán a cada servicio de comida, el operario puede utilizar la interfaz de adjudicar cupos iniciales de servicios:



*figura 36 – captura de interfaz de 'adjudicar cupos de servicios iniciales'*

Esta interfaz ya no permite cambiar el número de cupos que van a asignarse a cada servicio de comida; únicamente sirve para indicarle al sistema que puede hacer un ordenamiento de los inscritos según los criterios definidos, y adjudicar el servicio a quienes lo merezcan, hasta agotar los cupos de cada servicio.

## gestión de cupos

administrar cupos adicionales				
servicio	aprobados al inicio	adjudicados	disponibles	acción
desayuno	50	0	<input type="text" value="50"/>	
almuerzo	20	0	<input type="text" value="20"/>	
comida	10	0	<input type="text" value="10"/>	

*figura 37 – captura de interfaz de ‘modificar cupos disponibles’*

Una vez han sido adjudicados los cupos iniciales, los inscritos que se presenten como pertenecientes a grupos especiales, y en general cualquier otro inscrito que vaya a adjudicarse, tendrá que ser adjudicado con cupos disponibles. Si no hay suficientes cupos disponibles, un jefe de operación de Bienestar Universitario puede cambiar la cantidad de cupos disponibles que hay en el sistema.

### 3.4.4 Control de la asistencia

#### Casos de uso

Este módulo implementa los casos de uso que ejecutan administran y gestionan la información relacionada con las asistencias de los beneficiarios a las sesiones de servicios.

A continuación una lista de los casos de uso que pertenecen al módulo, seguida por su descripción detallada. Algunos de ellos fueron ilustrados además con capturas de pantalla:

- gestionar servicios servidos
- gestionar asistencia a servicio

#### Gestionar servicios servidos

caso de uso	Gestionar servicios servidos
actores	administrador, operario
propósito	Permite al actor iniciar, cerrar, suspender y abrir sesiones de servicios.
resumen	El actor puede iniciar sesión de servicio para agregar asistencias, luego cerrar el servicio o suspenderlo en caso de cancelarse la sesión por eventos de fuerza mayor.
precondiciones	Se requiere que el actor esté validado correctamente.

flujo principal	<p>Se muestra un formulario para que el actor escoja la fecha y el servicio para iniciar sesión. Una vez iniciada la sesión se ejecuta el caso de uso 'gestionar asistencias a servicio' para agregar asistencias a la sesión.</p> <p>El actor puede cerrar la sesión, o suspenderla en caso que el servicio no se termine adecuadamente (no termine normalmente, sino por eventos de fuerza mayor).</p> <p>También se le permite al actor deshacer una suspensión de un servicio.</p> <p>Si el actor es 'administrador', le permite abrir servicio que se haya cerrado.</p>
subflujos	<p><b>Subflujo 1:</b> abrir servicio; se muestra un formulario donde el actor escoge una sesión que haya sido cerrada.</p> <p><b>Subflujo 2:</b> suspender servicio; se muestra un formulario donde el actor ingresa las observaciones y temporalmente se suspende el servicio. Luego, se muestra al usuario los datos de la sesión, un mensaje con la información de la suspensión. Le permite al usuario finalizar la sesión o reanudarla.</p>
excepciones	Ninguna.

*tabla 37 – caso de uso 'gestionar servicios servidos'*

## Gestionar asistencias a servicio


caso de uso	Gestionar asistencias a servicio
actores	administrador, operario
propósito	Permite al actor agregar asistencias a una sesión de servicio.


resumen	una vez elegida la sesión de servicio, el actor ingresa el documento del inscrito y se registra su asistencia.
precondiciones	Se requiere que el actor esté validado correctamente, y que se haya iniciado alguna sesión de servicio.
flujo principal	<p>Si ya hay una sesión abierta, se escoge ésta y se muestra un formulario para que el actor ingrese el documento del inscrito. Si no hay una sesión abierta, se ejecuta el caso de uso "gestionar asistencias a servicio".</p> <p>Si al agregar una asistencia el inscrito tiene el servicio de la sesión en el estado 'habilitado', se despliega un mensaje de éxito. En caso contrario, un mensaje que especifica que el inscrito no puede hacer uso del servicio.</p>
subflujos	<b>Subflujo 1:</b> si el inscrito no tiene el servicio habilitado, pero le fue adjudicado, le permite al actor registrar esa asistencia forzosamente.
excepciones	<p><b>Excepción 1:</b> si el documento no corresponde a ningún inscrito, se muestra un aviso.</p> <p><b>Excepción 2:</b> si el inscrito no solicitó ese servicio, se despliega un mensaje de advertencia.</p> <p><b>Excepción 3:</b> si el inscrito si solicitó ese servicio, pero no lo tiene adjudicado, se despliega un mensaje de advertencia.</p> <p><b>Excepción 4:</b> si el inscrito so solicitó ese servicio, pero no lo tiene habilitado, se ejecuta el subflujo 1.</p> <p><b>Excepción 5:</b> si el inscrito ya tiene registrada una asistencia para esa sesión, se despliega un mensaje de advertencia.</p>

*tabla 38 – caso de uso ‘gestionar asistencias a servicio’*

## control de asistencia

**asistencia a sesión de servicio**

fecha  

servicio   \*

**agregar asistencias**



## control de asistencia

**asistencia a sesión de servicio**

fecha  

servicio   \*

**iniciar sesión**

¿está seguro de que desea iniciar una sesión?;  
las sesiones deben iniciarse únicamente  
cuando un servicio está a punto de ser servido

*figura 38 – captura de interfaz de 'iniciar una sesión de servicio'*

La interfaz de toma de asistencia permite al operario indicar cuándo está por comenzar una sesión de comida en las instalaciones de Bienestar Universitario, y

pide que confirme su intención de dar inicio a una sesión. El hecho de que el mensaje se despliegue sin mover el botón de lugar permite al usuario experimentado hacer dos clics muy rápidos y no detenerse en este mensaje ni perder tiempo con él. Una vez iniciado un servicio, el operario de Bienestar Universitario puede tomar asistencia a las personas que se presenten:



The image shows a web interface titled "control de asistencia". Below the title is a blue header bar with the text "opciones de la sesión". Underneath, the text "desayuno" is displayed in a large font, followed by the date "20 de julio de 2011". There is a label "código" next to a light blue rectangular input field. At the bottom of the interface, there are two buttons: "cerrar" and "suspender", both with a blue gradient and white text.

*figura 39 – captura de interfaz de 'crear asistencia a servicio'*

En este momento el sistema está listo para recibir un código de inscrito; ya sea obtenido con un lector de códigos de barras o digitado manualmente en caso de que el inscrito haya perdido su carné estudiantil o no tenga uno: el caso de los empleados de la UIS que reciben el servicio de comedores.

La interfaz también permite cerrar la sesión, indicando que pasó la hora límite para asistir, o suspender la sesión por causas imprevistas, describiendo esas causas en una observación.



*figura 40 – captura de interfaz de 'crear asistencia a servicio' sin el servicio adjudicado*

Si el inscrito que desea ingresar a la sesión de comida en el comedor de Bienestar Universitario no tiene adjudicado el servicio, una notificación se muestra en la pantalla y el operario no puede hacer nada más que negarle la entrada. Si el inscrito tiene adjudicado el servicio, pero no cumple con los requisitos necesarios para asistir a él, la interfaz también muestra un mensaje de paso no admitido:



*figura 41 – captura de interfaz de ‘crear asistencia a servicio’ sin el servicio habilitado*

En este caso la interfaz muestra además un botón de ‘ajuste manual’. El botón tiene esa etiqueta intencionalmente imprecisa para que, en caso de que un inscrito vea la pantalla, no encuentre un botón para permitir su ingreso a pesar de no cumplir con los requisitos. Pero eso es precisamente lo que hace el botón de ajuste manual: deja ingresar al inscrito que tenga el servicio adjudicado pero no habilitado, y registra su asistencia a la sesión de comida de esa fecha.



*figura 42 – captura de interfaz de 'crear asistencia a servicio' con paso libre*

La interfaz también muestra un mensaje de paso no permitido cuando un código no pertenece a un estudiante que se haya inscrito en el servicio de comedores, o cuando el inscrito ya ingresó al servicio e intenta entrar de nuevo.

En caso de que un operario desee agregar la asistencia de un inscrito a una sesión de comida después de que esa sesión fue cerrada, él o un superior con autoridad debe utilizar la interfaz de abrir sesión de servicio para volver a abrir la sesión:

## control de asistencia

abrir sesión

fecha

servicio  •

*figura 43 – captura de interfaz de 'abrir sesión de servicio'*

Esta interfaz muestra únicamente un listado de los que ya fueron servidos y cerrados para la fecha elegida; en el momento de tomar esta captura, ningún servicio se había servido el 20 de julio de 2011. Pero para días donde sí se sirvieron servicios de comida, el listado se cargaría con esos servicios servidos, y la interfaz permitiría volver a abrirlos para agregar asistencias de forma extemporánea.

## 3.4.5 Administración

### Casos de uso

Este módulo implementa los casos de uso que manipulan la información de las tablas de soporte para el sistema, y que ofrecen también las consultas de la información del sistema.

A continuación una lista de los casos de uso que pertenecen al módulo, seguida por su descripción detallada. Algunos de ellos fueron ilustrados además con capturas de pantalla:

- Hacer mantenimiento de años y semestres académicos
- Hacer mantenimiento de parámetros
- Hacer mantenimiento de tipos de usuario
- Hacer mantenimiento de criterios
- Hacer mantenimiento de servicios
- Hacer mantenimiento de requisitos
- Hacer mantenimiento de periodos de pago
- Hacer mantenimiento de grupos especiales
- Validar requisitos
- Consultar

## Hacer mantenimiento de años y semestres académicos

caso de uso	Hacer mantenimiento de años y semestres académicos
actores	administrador
propósito	Crear, modificar, y consultar los registros de la tabla de años y semestres académicos.
resumen	el sistema permite al actor crear, modificar, y consultar los registros de la tabla de años y semestres académicos.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	<p>Se despliega el formulario que le permite al actor crear un nuevo año y semestre académico, que será el que empiece a regir en el momento de la creación.</p> <p>También se muestra un listado de todos los registros creados, donde se muestran las opciones de 'modificar' y 'ver' cada registro de la lista.</p>
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del año y semestre académico para su edición.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada del año y semestre académico.</p>
excepciones	Si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.

*tabla 39 – caso de uso 'hacer mantenimiento de años y semestres academicos'*

## Hacer mantenimiento de parámetros

caso de uso	Hacer mantenimiento de parámetros
actores	administrador
propósito	Crear, modificar, consultar y eliminar los registros de la tabla de parámetros.
resumen	el sistema permite al actor crear, modificar, consultar y eliminar los registros de la tabla de parámetros.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	<p>Se despliega el formulario que le permite al actor crear un nuevo parámetro.</p> <p>También se muestra un listado de todos los registros creados, donde se muestran las opciones de 'modificar', 'ver' y 'eliminar' cada registro de la lista.</p>
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del parámetro para su edición.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada del parámetro.</p> <p><b>Subflujo 3:</b> si el actor elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario con la información detallada del parámetro para su eliminación.</p>

excepciones	<p><b>Excepción 1:</b> si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.</p> <p><b>Excepción 2:</b> si al eliminar un registro se viola una restricción de integridad en la base de datos, se muestra un mensaje de advertencia y se impide llevar a cabo la operación.</p>
-------------	---

*tabla 40 – caso de uso ‘hacer mantenimiento de parámetros’*

### Hacer mantenimiento de tipos de usuario

caso de uso	Hacer mantenimiento de tipos de usuario
actores	administrador
propósito	Crear, modificar, consultar y eliminar los registros de la tabla de tipos de usuario.
resumen	el sistema permite al actor crear, modificar, consultar y eliminar los registros de la tabla de tipos de usuario.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	<p>Se despliega el formulario que le permite al actor crear un nuevo tipo de usuario.</p> <p>También se muestra un listado de todos los registros creados, donde se muestran las opciones de ‘modificar’, ‘ver’ y ‘eliminar’ cada registro de la lista.</p>

subflujos	<p><b>Subflujo 1:</b> si el actor elije la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del tipo de usuario para su edición.</p> <p><b>Subflujo 2:</b> si el actor elije la opción 'ver' de algún registro de la lista, se despliega la información detallada del tipo de usuario.</p> <p><b>Subflujo 3:</b> si el actor elije la opción 'eliminar' de algún registro de la lista, se despliega un formulario con la información detallada del tipo de usuario para su eliminación.</p>
excepciones	<p><b>Excepción 1:</b> si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.</p> <p><b>Excepción 2:</b> si al eliminar un registro se viola una restricción de integridad en la base de datos, se muestra un mensaje de advertencia y se impide llevar a cabo la operación.</p>

*tabla 41 – caso de uso 'hacer mantenimiento de tipos de usuario'*

## Administración de tipos de usuario

### Crear tipo de usuario

nombre  \*

descripción  \*

tipo de documento  \*

tabla de consulta  \*

lista de tipos de usuario		
nombre	descripción	acción
estudiante normal	.	 
estudiante primer nivel	.	 
empleado	.	 
estudiante grupo especial	.	 

4 registros

*figura 44 – captura de interfaz de 'modificar tipo de usuario'*

La interfaz de modificar tipo de usuario permite al operario crear nuevos tipos de usuario y relacionarlos con un tipo de documento. Adicionalmente, puede escribirse una descripción de los tipos de usuario. Algunos ejemplos de tipos de usuario son:

- estudiante
- estudiante de primer nivel
- estudiante en trabajo de grado
- estudiante miembro de selección deportiva
- estudiante miembro de grupo artístico
- empleado UIS

Diferentes tipos de usuario tienen diferentes requisitos para recibir el servicio de comedores.

La interfaz también permite ver los tipos de usuario creados hasta el momento, y si es necesario ver sus detalles:

## Administración de tipos de usuario

ver detalle tipo de usuario	
código	1
nombre	estudiante normal
descripción	.
tipo de documento	codigo estudiante
tabla de consulta	GnrEstudianteUis

[Regresar](#)

*figura 45 – captura de interfaz de 'ver detalles de tipo de usuario'*

Si el operario lo desea puede modificar un tipo de usuario, o eliminarlo:

## Administración de tipos de usuario

### Modificar tipo de usuario

nombre	<input type="text" value="estudiante normal"/> *	<a href="#">?</a>
descripción	<input type="text" value="."/> *	<a href="#">?</a>
tipo de documento	<input type="text" value="codigo estudiante"/> *	<a href="#">?</a>
tabla de consulta	<input type="text" value="GnrEstudianteUis"/> *	<a href="#">?</a>

figura 46 – captura de interfaz de 'modificar un tipo de usuario'

### Hacer mantenimiento de criterios

caso de uso	Hacer mantenimiento de criterios
actores	administrador
propósito	Crear, modificar, consultar y eliminar los registros de la tabla de criterios.
resumen	el sistema permite al actor crear, modificar, consultar y eliminar los registros de la tabla de criterios.
precondiciones	Se requiere que el actor esté validado correctamente.

flujo principal	<p>Se despliega el formulario que le permite al actor crear un nuevo criterio.</p> <p>También se muestra un listado de todos los registros creados, donde se muestran las opciones de 'modificar', 'ver' y 'eliminar' cada registro de la lista.</p>
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del criterio para su edición.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada del criterio.</p> <p><b>Subflujo 3:</b> si el actor elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario con la información detallada del criterio para su eliminación.</p>
excepciones	<p><b>Excepción 1:</b> si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.</p> <p><b>Excepción 2:</b> si al eliminar un registro se viola una restricción de integridad en la base de datos, se muestra un mensaje de advertencia y se impide llevar a cabo la operación.</p>

*tabla 42 – caso de uso 'hacer mantenimiento de criterios'*

## Hacer mantenimiento de servicios

caso de uso	Hacer mantenimiento de servicios
actores	administrador
propósito	Crear, modificar, consultar y eliminar los registros de la tabla de servicios.

resumen	el sistema permite al actor crear, modificar, consultar y eliminar los registros de la tabla de servicios.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	<p>Se despliega el formulario que le permite al actor crear un nuevo servicio.</p> <p>También se muestra un listado de todos los registros creados, donde se muestran las opciones de 'modificar', 'ver' y 'eliminar' cada registro de la lista.</p>
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del servicio para su edición.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada del servicio.</p> <p><b>Subflujo 3:</b> si el actor elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario con la información detallada del servicio para su eliminación.</p>
excepciones	<p><b>Excepción 1:</b> si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.</p> <p><b>Excepción 2:</b> si al eliminar un registro se viola una restricción de integridad en la base de datos, se muestra un mensaje de advertencia y se impide llevar a cabo la operación.</p>

*tabla 43 – caso de uso 'hacer mantenimiento de servicios'*

## Hacer mantenimiento de requisitos

caso de uso	Hacer mantenimiento de requisitos
actores	administrador
propósito	Crear, modificar, consultar y eliminar los registros de la tabla de requisitos.
resumen	el sistema permite al actor crear, modificar, consultar y eliminar los registros de la tabla de requisitos.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	<p>Se despliega el formulario que le permite al actor crear un nuevo requisito.</p> <p>También se muestra un listado de todos los registros creados, donde se muestran las opciones de 'modificar', 'ver' y 'eliminar' cada registro de la lista.</p>
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del requisito para su edición.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada del requisito.</p> <p><b>Subflujo 3:</b> si el actor elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario con la información detallada del requisito para su eliminación.</p>

excepciones	<p><b>Excepción 1:</b> si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.</p> <p><b>Excepción 2:</b> si al eliminar un registro se viola una restricción de integridad en la base de datos, se muestra un mensaje de advertencia y se impide llevar a cabo la operación.</p>
-------------	---

*tabla 44 – caso de uso ‘hacer mantenimiento de requisitos’*

### Hacer mantenimiento de periodos de pago

caso de uso	Hacer mantenimiento de periodos de pago
actores	administrador
propósito	Crear, modificar, consultar y eliminar los registros de la tabla de periodos de pago.
resumen	el sistema permite al actor crear, modificar, consultar y eliminar los registros de la tabla de periodos de pago.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	Se despliega el formulario que le permite al actor crear un nuevo periodo de pago. También se muestra un listado de todos los registros creados, donde se muestran las opciones de ‘modificar’, ‘ver’ y ‘eliminar’ cada registro de la lista.

subflujos	<p><b>Subflujo 1:</b> si el actor elije la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del periodo de pago para su edición.</p> <p><b>Subflujo 2:</b> si el actor elije la opción 'ver' de algún registro de la lista, se despliega la información detallada del periodo de pago.</p> <p><b>Subflujo 3:</b> si el actor elije la opción 'eliminar' de algún registro de la lista, se despliega un formulario con la información detallada del periodo de pago para su eliminación.</p>
excepciones	<p><b>Excepción 1:</b> si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.</p> <p><b>Excepción 2:</b> si al eliminar un registro se viola una restricción de integridad en la base de datos, se muestra un mensaje de advertencia y se impide llevar a cabo la operación.</p>

*tabla 45 – caso de uso 'hacer mantenimiento de periodos de pago'*

## Administración de periodo de pago

### Crear periodo de pago











año 2010 semestre académico 2

periodo 7

fecha de inicio   \* 

fecha fin   \* 

cantidad de días  \* 

periodos de pago				
periodo	fecha de inicio	fecha fin	cantidad de días	acción
1	ago/03/2010	sep/15/2010	30	
2	sep/16/2010	nov/17/2010	42	
3	nov/18/2010	jun/30/2011	10	
4	jul/01/2011	jul/05/2011	2	
5	jul/06/2011	oct/30/2011	32	  
6	oct/31/2011	nov/14/2011	9	  

6 registros

figura 47 – captura de la interfaz de crear periodos de pago

Para el mantenimiento de la tabla de periodos de pago, el caso de uso despliega inicialmente la anterior interfaz, donde el operario de Bienestar Universitario puede hacer uso del formulario de creación, que por defecto ajusta las fechas según los periodos que ya han sido creados (que la fecha de inicio del nuevo periodo sea un día después del último periodo creado), y el número del periodo sea uno más que el anterior.

En caso de enviar el formulario con campos faltantes se despliega la siguiente pantalla:

Administración de periodo de pago

Crear periodo de pago

año 2010 semestre académico 2

periodo 7

fecha de inicio nov/15/2011 \*

fecha fin dic/01/2011 \*

cantidad de días \* ✘ este campo es obligatorio

Guardar Cancelar

*figura 48 – captura del mensaje de validación en 'crear periodo'*

En la interfaz de crear periodo se muestra también el listado de los periodos de pago creados. En cada registro de la lista se encuentra la opción de 'ver', 'modificar' y 'eliminar' para los periodos creados que aún no se hayan vencido.

Si el operario elige 'ver' en el registro que escoja, se despliega la siguiente interfaz

## Administración de periodo de pago






actualizar periodo			
año	2010	semestre académico	2
periodo	6		
fecha de inicio	31 de octubre de 2011		
fecha fin	14 de noviembre de 2011		
cantidad de días	9		
<input type="button" value="Regresar"/>			


*figura 49 – captura de interfaz de 'ver periodo'*

El botón de regresar devuelve al operario a la pantalla de 'crear periodo'. Si se elige la opción de modificar (en ícono) se presenta la pantalla de modificar, cargando la información del periodo en el formulario, y aplicando la misma validación de cuando se crea el periodo. (ver **figura 49**)

## Administración de periodo de pago

actualizar periodo

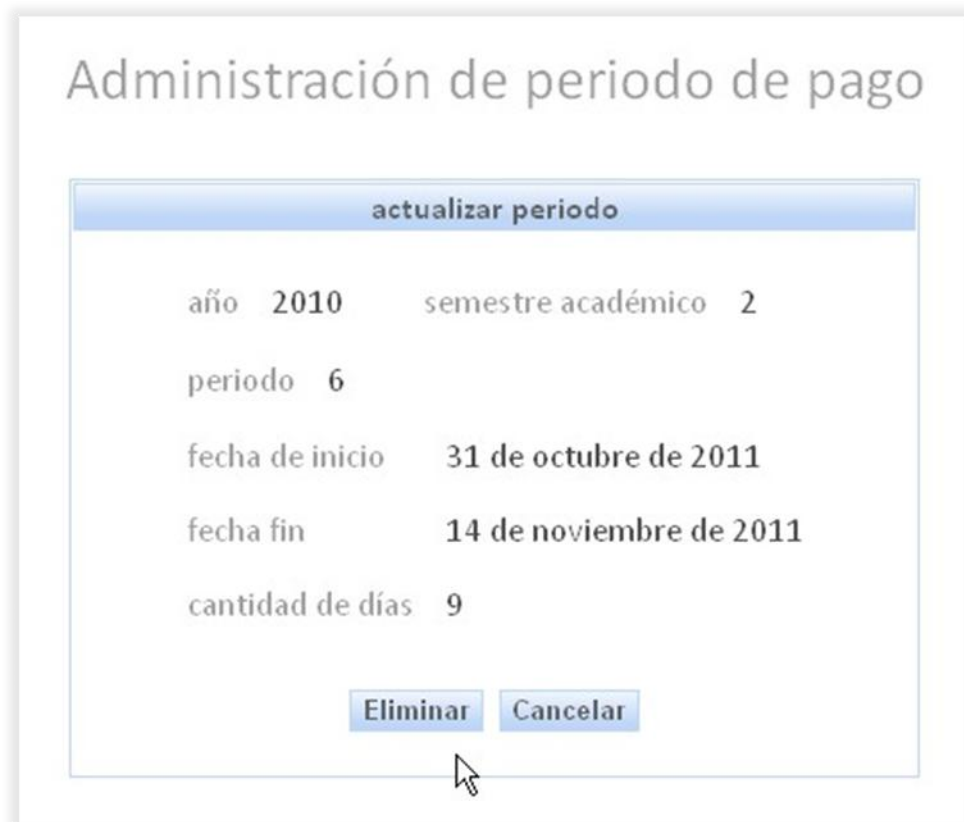
año	2010	semestre académico	2	
periodo	6			
fecha de inicio	<input type="text" value="oct/31/2011"/>		*	
fecha fin	<input type="text" value="nov/14/2011"/>		*	
cantidad de días	<input type="text" value="9"/>		*	



*figura 50 – captura de interfaz de ‘modificar periodo’*

El botón de ‘cancelar’, como dice su nombre, cancela la acción y vuelve a mostrar la pantalla de de ‘crear periodo’. El botón ‘modificar’ ejecuta la acción y despliega un mensaje de error si la validación no es satisfactoria.

Si el operario selecciona la opción de ‘eliminar’ (el ícono en la columna de acciones) se despliega la pantalla de eliminación, que es similar a la de modificación. Presenta los botones de ‘eliminar’ y ‘cancelar’. El botón de ‘cancelar’ ejecuta la misma tarea que en la pantalla de ‘modificar periodo’. El botón de ‘eliminar’ valida que borrar el registro no tenga conflictos funcionales, ni que esté asociado en la base de datos.



*figura 51 – captura de interfaz de 'eliminar periodo'*

### Hacer mantenimiento de grupos especiales

caso de uso	Hacer mantenimiento de grupos especiales
actores	administrador
propósito	Crear, modificar, consultar y eliminar los registros de la tabla de grupos especiales.
resumen	el sistema permite al actor crear, modificar, consultar y eliminar los registros de la tabla de grupos especiales.
precondiciones	Se requiere que el actor esté validado correctamente.

flujo principal	<p>Se despliega el formulario que le permite al actor crear un nuevo grupo especial.</p> <p>También se muestra un listado de todos los registros creados, donde se muestran las opciones de 'modificar', 'ver' y 'eliminar' cada registro de la lista.</p>
subflujos	<p><b>Subflujo 1:</b> si el actor elige la opción 'modificar' de algún registro de la lista, se despliega un formulario con la información detallada del grupo especial para su edición.</p> <p><b>Subflujo 2:</b> si el actor elige la opción 'ver' de algún registro de la lista, se despliega la información detallada del grupo especial.</p> <p><b>Subflujo 3:</b> si el actor elige la opción 'eliminar' de algún registro de la lista, se despliega un formulario con la información detallada del grupo especial para su eliminación.</p>
excepciones	<p><b>Excepción 1:</b> si hay información inconsistente en los campos de los formularios de creación y modificación, se muestran los mensajes de advertencia y se impide llevar a cabo la operación.</p> <p><b>Excepción 2:</b> si al eliminar un registro se viola una restricción de integridad en la base de datos, se muestra un mensaje de advertencia y se impide llevar a cabo la operación.</p>

*tabla 46 – caso de uso 'hacer mantenimiento de grupos especiales'*

## Validar requisitos

caso de uso	Validar requisitos
actores	administrador, operario, jefe de operación
propósito	validar los requisitos de todos los inscritos que tienen adjudicado algún servicio.
resumen	El actor ejecuta el proceso de revisión y validación de requisitos.
precondiciones	Se requiere que el actor esté validado correctamente y que no haya sesiones abiertas.
flujo principal	El actor ejecuta el proceso de validación de requisitos. cuando se termine el proceso, se muestran la cantidad de inscritos habilitados en cada servicio.
subflujos	Ninguno.
excepciones	Ninguna.

*tabla 47 - caso de uso 'validar requisitos'*

## Consultar

caso de uso	Consultar
actores	administrador, operario, jefe de operación
propósito	Ejecutar la consulta que el actor indique y mostrar los resultados.
resumen	Se despliega el listado de consultas para que el actor seleccione la que desee ejecutar.
precondiciones	Se requiere que el actor esté validado correctamente.
flujo principal	<p>Se despliega el siguiente listado de consultas:</p> <ul style="list-style-type: none"><li>consultar registro histórico del número de usuarios inscritos y adjudicados.</li><li>consultar cantidad de inscritos y adjudicados detallado.</li><li>consultar lista de todos los inscritos por periodo académico.</li><li>consultar el valor de los criterios mínimos para adjudicación.</li><li>consulta la cantidad de inscritos habilitados para cada servicio en el instante.</li></ul> <p>Una vez ejecutada, se permite la opción de generar el reporte en *.pdf para su impresión.</p>
subflujos	Ninguno.
excepciones	Ninguna.

### 3.4.6 Esquema de seguridad

Para este proyecto se utiliza el esquema de seguridad definido por la División de Servicios de Información para los diferentes sistemas de información que apoyan la gestión de la Universidad Industrial de Santander, el cual está basado en la estructura de roles – usuarios.

Los roles se establecen en cada una de las unidades académico administrativas, UAA, responsables de cada sistema, de acuerdo a las actividades que realizan. A cada uno de los roles definidos se le asocian los usuarios de acuerdo a las funciones que desempeñen.

#### **Estructura de la base de datos soporte**

La base de datos que soporta el esquema de seguridad contempla básicamente las siguientes tablas:

**Sistema:** Contiene información de los sistemas de información de la universidad. Para cada sistema se especifica: Nombre, descripción del sistema, fecha y hora de creación en la base de datos, fecha y hora de inicio de vigencia del sistema, fecha y hora de cierre de vigencia del sistema.

**Rol:** contiene información de los diferentes roles definidos para cada sistema de información, como: Nombre asignado al rol, descripción del rol, fecha y hora de creación, fecha y hora de inicio de vigencia del rol, fecha y hora de cierre de vigencia del rol.

**Usuario:** Contiene información de los posibles usuarios de los sistemas de información. Entre esta información está: tipo y número de documento de identidad del usuario, fecha y hora de creación del usuario, fecha y hora de inicio de vigencia del usuario, fecha y hora de cierre de vigencia del usuario.

**Sistema–rol:** Contiene los roles definidos para cada uno de los sistemas de información, indicando: rol, sistema, fecha y hora de creación del rol – sistema, fecha y hora de inicio de vigencia del rol en el sistema, fecha y hora de cierre de vigencia del rol en el sistema.

Rol–usuario: Contempla los usuarios asociados a cada uno de los roles definidos, considerando: Rol, usuario, fecha y hora de creación del rol – usuario, fecha y hora de inicio de vigencia del usuario en el rol, fecha y hora de cierre de vigencia del usuario en el rol.

Menú–rol–sistema: Contiene los menús asociados a los roles en los distintos sistemas de información, contemplando: Sistema de información, nombre del menú, descripción del menú, fecha y hora de creación del menú, fecha y hora de inicio de vigencia del menú asociado al rol, fecha y hora de cierre de vigencia del menú asociado al rol.

Opción–menú–rol: Contempla las opciones definidas para cada una de las posibles menús establecidos para cada sistema de información. Contiene: Nombre de la opción, descripción de la opción, nombre del menú superior, nombre del menú que contiene la opción, nombre del programa a ejecutar cuando la opción es la de más bajo nivel, fecha y hora de creación de la opción del menú, fecha y hora de inicio de vigencia de la opción, fecha y hora de cierre de la opción.

Tabla–sistema: Contiene información de las tablas que conforman la base de datos que soporta cada uno de los sistemas de información. Considera: Sistema de información, nombre de la tabla, descripción de la tabla.

Tipo–permiso: Establece para cada tabla de un sistema de información, los roles que tienen permisos para incluir registros, para modificar registros o para eliminar registros en ella. Contiene: Sistema de información, nombre de la tabla, clase de permiso (inclusión, modificación, eliminación de registros), fecha y hora de creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.

Acceso–tabla: Define para las tablas de un sistema de información si un rol tiene permiso sobre toda la información de la tabla o sobre una parte de esta. Considera: Sistema, nombre de la tabla, clase de acceso (total, parcial), fecha y hora de creación del permiso, fecha y hora de inicio de vigencia del permiso, fecha y hora de fin de vigencia del permiso.

Atributo–tabla: Establece los atributos sobre los cuales se debe controlar el acceso a una tabla, cuando a un rol se le concede permiso para hacer uso parcial de la información existente en una tabla. Contiene: Sistema de información, nombre de la tabla, nombre del atributo sobre el cual se controla el acceso a la tabla,

descripción del atributo, fecha y hora de creación del atributo, fecha y hora de inicio de vigencia del atributo, fecha y hora de fin de vigencia del atributo.

Valor–atributo-proceso: Contiene los valores que deben tener los atributos definidos en cada tabla en la tabla atributo – tabla que permiten el acceso a la información asociada a estos valores. Específica: Sistema de información, nombre de la tabla, nombre del atributo, valor del atributo, descripción, fecha y hora de creación del valor del atributo, fecha y hora de inicio de vigencia del valor del atributo, fecha y hora de fin de vigencia del valor del atributo.

Acceso-sistema: Contempla el histórico de acceso que un usuario ha realizado a un sistema, identificando las opciones que ha seleccionado. Contiene: Login de usuario, rol, identificación de la sesión, sistema, opción seleccionada, fecha y hora de ingreso, fecha y hora de salida

### **Entorno de navegación**

Para cada sistema de información, la UAA responsable define los roles necesarios para el adecuado uso del sistema de información de acuerdo a las funciones que realice y establece los usuarios asociados a cada uno de ellos.

Para cada rol se define el menú de inicio, el cual le permite a cada usuario que hace parte de este rol, empezar la navegación por las distintas opciones que le ofrece el sistema, hasta llegar al nivel más bajo en el cual se ejecuta el proceso que soporta la actividad que desea realizar.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, menú-rol, opción-menu rol, descritas en “ESTRUCTURA DE LA BASE DE DATOS SOPORTE”.

### **Entorno de control de datos**

Para los roles definidos en cada uno de los sistemas de información se especifican las tablas a las cuales puede acceder, el tipo de transacción que puede realizar sobre estas tablas (inclusión, modificación o eliminación de registros), si tiene acceso total o parcial a la información que contiene la tabla.

Para el acceso a la información de la tabla de manera parcial, se debe establecer el atributo o atributos seleccionados, los valores que estos atributos deben tener para autorizar el acceso solicitado.

Este entorno está soportado por las siguientes tablas de las base de datos del esquema de seguridad: Rol, usuario, sistema, sistema-rol, usuario-rol, tabla-sistema, tipo-permiso, acceso-tabla, atributo-tabla, valor atributo proceso, descritas en "ESTRUCTURA DE LA BASE DE DATOS SOPORTE".

### **Auditoría**

Todas las tablas que conforman la base de datos soporte del esquema de seguridad tienen el historial de las transacciones realizadas sobre cada una de ellas.

El historial de las transacciones de cada tabla contiene información de los registros incluidos en la tabla, de los registros modificados y de los registros eliminados. Adicionalmente, en cada transacción se especifica: Fecha de la transacción, hora de la transacción, tipo de transacción (I/U/D), tipo y número de documento de identidad del usuario que realizó la transacción, login, rol asociado, dirección IP y MAC del equipo desde el cual llevó a cabo la transacción.

# 4

## Conclusiones

- La nueva versión del sistema de información del servicio de comedores de la Universidad Industrial de Santander se convierte en la solución a los problemas de segmentación, funcionalidad y practicidad que experimentaba el sistema anterior. Como consecuencia el nuevo sistema agiliza y aumenta la eficiencia toda la gestión de los procesos del servicio de comedores de Bienestar Universitario de la UIS.

Adicionalmente el sistema desarrollado en este proyecto responde a los requisitos que tiene la prestación del servicio que no estaban contemplados en la versión anterior con la que trabajaba Bienestar Universitario.

El nuevo sistema fue desarrollado para funcionar en un entorno completamente Web, de forma que tanto sus administradores como cada uno de sus usuarios puede aprovecharlo desde un navegador de Internet, sin instalar software adicional.

- Para el desarrollo de este proyecto se utilizaron varios frameworks para aplicaciones Web con la intención de reducir la carga asociada con tareas comunes y repetitivas en el proceso de desarrollo.

Los frameworks y especificaciones de arquitecturas avanzadas, que hacen posible crear componentes reutilizables, tienen la desventaja de tener una curva de aprendizaje lenta, pero proporcionan como resultado el desarrollo de una aplicación empresarial potente y de fácil mantenimiento.

- La metodología de desarrollo por etapas resultó apropiada para este proyecto pues permitió que tanto los supervisores como el cliente hicieran observaciones importantes a lo largo del proceso, lo que hizo posible que los desarrolladores corrigieran los problemas en etapas tempranas y asegurar la calidad y funcionalidad de cada módulo antes de iniciar el desarrollo del siguiente.
- La construcción de prototipos no funcionales para ilustrar el resultado final del proyecto se tradujo en un ahorro considerable de tiempo, debido a que el usuario interactuó con las entradas y salidas que iba a tener el sistema y pudo hacer pruebas reales a su esquema de navegación, para luego hacer observaciones, correcciones y sugerencias, sin que en el proceso los desarrolladores perdieran tiempo en la programación de módulos que iban a ser descartados o largamente revisados.

# 5

## Recomendaciones

- **Política de cero papel:**

Los desarrolladores de este sistema recomiendan que sus usuarios lo aprovechen para reducir al mínimo los trámites realizados en papel, aprovechando su capacidad para guardar registro de todas las transacciones como un método igualmente efectivo, o incluso más eficiente, de documentar todos los procesos del servicio de comedores de Bienestar Universitario, incluyendo las situaciones completamente excepcionales.

De ser posible, la plataforma debería estar acompañada de un módulo de 'notificaciones' o 'comunicaciones'; un módulo de mensajería entre los operarios del sistema, que permita llevar a cabo procesos que actualmente se realizan con papel, sellos y firmas, de forma digital y limpia.

- **Sistema de notificaciones a inscritos:**

Los desarrolladores de este sistema recomiendan a la División de Servicios de Información hacer énfasis en la importancia de que los estudiantes registren un correo electrónico actualizado en su información en el portal de la Universidad Industrial de Santander, de forma que ese dato pueda usarse, si ellos así lo desean, para enviarles notificaciones importantes.

De ser posible, la información de contacto también debería incluir redes sociales como Facebook, Twitter, LinkedIn o Google+, herramientas en extremo útiles para notificar a las personas de eventualidades que merecen su atención inmediata.

- **plataforma de soporte para casos de emergencia**

Los desarrolladores de este sistema recomiendan que se utilice una herramienta apropiada para registrar la asistencia de los inscritos de comedores de Bienestar Universitario en escenarios donde no se cuente con acceso a la Red, a la red interna de la universidad, o incluso si no hay disponibilidad de luz eléctrica.

Un ordenador portátil con suficiente batería y un módulo de emergencia podrían cumplir con el trabajo, y posteriormente el módulo de Control de la Asistencia serviría para ingresar la información al sistema principal.

# 6

## Anexos

Tanto los diagramas de casos de uso detallados, como los diagramas de entidad relación, se encuentran en línea en las siguientes direcciones

<http://tinyurl.com/comedoresPrototipos>

<http://tinyurl.com/comedoresUML>

# 7

## Bibliografía

McConnell, Steve. Desarrollo y gestión de proyectos informáticos. Madrid, McGraw-Hill. 1997.

Piattini Velthuis, Mario G. Análisis y diseño de aplicaciones informáticas de gestión: Una perspectiva de ingeniería del software. México, Alfaomega. 2007.

Piattini Velthuis, Mario G. Tecnología y diseño de bases de datos. México, Alfaomega. 2007.

Joyanes Aguilar, Luis; Zahonero Martínez, Ignacio. Programación en JAVA 2: Algoritmos, estructuras de datos y programación orientada a objetos. Madrid, McGraw-Hill. 2002.

Allen, Dan. Seam in Action. Manning Publications. 2008.

Panda Debu; Rahman Reza; Lane Derek. EJB3 in Action. Manning Publications. 2007.

Mann, Kito D. JavaServer Faces in Action. Manning Publications. 2004.

Vera, Freddy. Propuesta de un proceso de desarrollo de software de componentes reutilizables, Bucaramanga, UIS.

Estándares de desarrollo. División de Servicios de Información. Bucaramanga, UIS.