

Universidad de las Ciencias Informáticas

Facultad 1



*Trabajo de diploma para optar por el título de  
Ingeniero en Ciencias Informáticas.*

*Título: Subsistema de asignación de comensales para la  
Universidad de las Ciencias Informáticas.*

*Autores: Odrey Esteban Hernández*

*Dayami Alvarez Cobiella*

*Tutores: Ing. Luis Ernesto Acosta Ortíz*

*Ing. Guillermo Gómez Urquiza*

*Ciudad de La Habana, Cuba*

*Curso 2011-2012*

# *Declaración de autoría*

---

Declaramos que somos los únicos autores del trabajo titulado Subsistema de asignación de comensales para la Universidad de las Ciencias Informáticas y autorizamos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo el presente a los \_\_\_\_ días del mes de Junio del año 2012.

---

Firma del autor

Dayami Alvarez Cobiella

---

Firma del autor

Odrey Esteban Hernández

---

Firma del Tutor

Luis Ernesto Acosta Ortíz

---

Firma del Tutor

Guillermo Gómez Urquiza

### **Dayami:**

*Agradezco a mis padres por apoyarme siempre en todas mis decisiones, principalmente por enseñarme a ser la persona que soy y por darme a mi bella hermana, sin ellos este logro no hubiera sido posible. Como tampoco hubiera sido posible sin el amor y la compañía incondicional de mi novio a lo largo de estos cinco años de carrera y la preocupación de sus padres quien hoy son también parte de mi familia, gracias los amo mucho a todos.*

*Agradezco especialmente a mi amigo y compañero de tesis porque sin el nada de esto hoy estaría ocurriendo, gracias por tu entrega y dedicación y sobre todo por tu linda amistad, nunca te olvidaré.*

*Agradezco también a todos mis compañeros que me han acompañado a lo largo de mi carrera, a los que hoy están aquí conmigo y a los que ya no están, muchas gracias por su amistad, siempre los recordaré con mucho cariño a todos.*

*A nuestros tutores y a las profesoras Iviannis Cañizares Rivera y Yulia Fustiel Alvarez por su tiempo y dedicación para con nosotros. En fin a todas las personas que me quieren ver bien y a quienes siempre confiaron en mí, gracias.*

### **Odrey:**

*Resumir en algunos nombres de personas e instituciones todo el agradecimiento por el alcance de un sueño es una tarea bastante difícil, quizás más que su realización. En este empeño por crecer han estado involucradas muchas personas que han sido el sustento y aliento diario a no cejar en el empeño.*

*En primer término quiero mencionar a mis padres, a quienes no solo agradezco sus consejos y el apoyo constante, sino además, y sobre todas las cosas, el privilegio de ser su hijo. En general a toda mi familia por su preocupación, por saber darme la ayuda necesaria en el momento justo y brindarme lo mejor de ellos.*

*Agradezco especialmente a mi compañera de tesis por su paciencia, dedicación y cariño sin los cuales no hubiese sido posible la realización de este trabajo.*

*A todos los amigos y amigas que compartieron trabajo, amor, sonrisas, consejos y momentos inolvidables; especialmente a mis compañeros de cuarto insustituibles amigos y cómplices de muchas alegrías, tristezas, días de fiestas y noches de trabajo.*

*A nuestros tutores y a los profesores Yulia, Iviannis y Yosvanis por su tiempo y ayuda incondicional en todo momento.*

*A todos, de corazón, gracias.*

### **Dayami:**

*Dedico este gran logro de mi vida especialmente a las tres personas que más amo:*

*A mi padre quien siempre me ha aconsejado para que salga adelante por mis propios esfuerzos, por siempre confiar en mí y especialmente por ser mi papá.*

*A mi madre quien siempre me ha enseñado a seguir adelante a pesar de las dificultades y a luchar por las cosas que queremos lograr.*

*A mi hermana por apoyarme siempre a pesar de ser menor que yo, por haberse convertido en una gran mujer y por darme la felicidad de ser tía.*

*También a mis abuelos por darme tanto amor, los adoro.*

### **Odrey:**

*A mis padres y tíos, por su confianza y preocupación. Por creer en mí y darme ejemplos dignos de superación y entrega que han hecho de mí la persona que soy actualmente. Porque en gran parte gracias a ustedes, hoy puedo ver alcanzada mi meta, ya que su apoyo incondicional, consejos y cariño siempre estuvieron impulsándome en los momentos más difíciles de mi carrera, y porque el orgullo que sienten por mí, fue lo que me hizo ir hasta el final.*

## Resumen

La alimentación del personal y al acceso del mismo a cada una de las puertas de los complejos existentes en la Universidad de las Ciencias Informáticas (UCI), son actividades muy importantes que se desarrollan diariamente en esta institución. Es por esto que se necesita tener un control estricto sobre dichas actividades mediante la asignación de comensales, con el objetivo de que dicho proceso se efectúe de manera correcta en la Dirección de Alimentación de la universidad, donde se controla este proceso. El desarrollo del Subsistema de asignación de comensales permite la asignación rápida y eficiente de los comensales a las diferentes puertas de acceso. Mejora aspectos existentes en el sistema anterior que impedían la correcta ejecución del proceso como la navegabilidad del sistema, el tiempo de ejecución y la organización, clasificación y actualización de las personas. Se utilizaron en su mayoría herramientas de desarrollo libres debido al gran impacto económico y social que tienen las mismas para la universidad y el país como son: Visual Paradigm 8.0 para el modelado, *Spring Framework* en su versión 3.0.2 como marco de trabajo y como lenguaje de programación Java. Como resultado se obtuvo una aplicación de escritorio totalmente funcional y que puede ser utilizada en cualquier institución que cuente con una base de datos donde se encuentre desplegado el núcleo del sistema de Gestión Universitaria.

**Palabras claves:** Asignación de comensales, control de acceso, eventos, tiempo de ejecución.

<b>Índice</b>	
Índice de tablas .....	VI
Índice de figuras .....	VII
Capítulo I: Asignación de comensales .....	6
Introducción.....	6
1.1 Conceptos relacionados con el dominio del problema.....	6
1.2 Estudio del estado del arte .....	7
1.3 Herramientas y tecnologías de <i>software</i> utilizadas en el desarrollo del subsistema .....	13
1.4 Herramientas de modelado .....	16
1.5 Herramientas de programación para el desarrollo de la aplicación.....	18
1.6 Sistemas Gestores de Bases de Datos .....	21
1.7 Entorno Integrado de Desarrollo (IDE) .....	23
1.8 Conclusiones parciales .....	23
Capítulo II: Propuesta de solución .....	25
Introducción.....	25
2.1 Modelo de dominio.....	25
2.2 Requisitos funcionales. ....	26
2.3 Requisitos no funcionales.....	29
2.4 Métrica para la calidad de la especificación de los requisitos de <i>software</i> .....	31
2.5 Diccionario de datos.....	32
2.6 Modelo lógico y físico. ....	38
2.7 Patrones de diseño. ....	41
2.8 Arquitectura.....	42
2.9 Descripción del subsistema.....	45
2.10 Conclusiones parciales.....	47
Capítulo III: Validación de la solución. ....	48
Introducción.....	48
3.1 Estándares de codificación.....	48
3.2 Diagrama despliegue .....	49
3.3 Pruebas.....	50
3.4 Método de prueba .....	50
3.5 Técnica empleada mediante el método caja negra.....	51
3.6 Resultados de las pruebas .....	55
3.7 Conclusiones parciales.....	58
Conclusiones.....	59
Recomendaciones.....	60
Bibliografía referenciada.....	61
Bibliografía consultada .....	63
Glosario de términos .....	64
Anexo 1 .....	66
Anexo 2.....	74

## Índice de tablas

Tabla 1. Miembros del equipo de inspección.....	31
Tabla 2. Resultados obtenidos al aplicar la métrica.....	32
Tabla 3. Entidad: Acceso.....	32
Tabla 4. Entidad: Punto de control.....	33
Tabla 5. Entidad: Grupo.....	34
Tabla 6. Entidad: Evento.....	34
Tabla 7. Entidad: Usuario.....	35
Tabla 8. Entidad: Distribución.....	35
Tabla 9. Entidad: Persona autorizada.....	36
Tabla 10. Entidad: Recurso foto.....	36
Tabla 11. Entidad: Estructura.....	36
Tabla 13. Entidad: Sede.....	37
Tabla 14. Entidad: Área.....	38
Tabla 15. CP: Asignar grupos a puntos.....	51
Tabla 16. CP: Gestionar sede.....	52
Tabla 17. CP: Balance de puntos.....	53
Tabla 18. CP: Puntos de control.....	53
Tabla 19. CP: Sin asignar.....	54
Tabla 20. Prueba de Integración Int_1 Subsistema de control de acceso.....	56
Tabla 11. Prueba de Integración Int_2 Subsistema de control de acceso.....	57
Tabla 22. Prueba de Integración Int_3 Subsistema de reservación de alimentación y planificación de menú.....	57
Tabla 23. Prueba de Integración Int_4 Subsistema de reservación de alimentación y planificación de menú.....	57
Tabla 24. RF: Mostrar eventos.....	66
Tabla 25. RF: Adicionar evento.....	67
Tabla 26. RF: Modificar evento.....	68
Tabla 27. RF: Eliminar evento.....	69
Tabla 28. RF Mostrar estructuras y puntos.....	69
Tabla 29. RF Crear estructura y punto de control.....	71
Tabla 30. RF Modificar estructura y punto de control.....	72
Tabla 31. RF Eliminar estructura.....	73
Tabla 32. CP Gestionar estructuras.....	74
Tabla 33. CP: Gestionar grupo.....	75
Tabla 34. CP: Adicionar y eliminar personas a grupos.....	76

## Índice de figuras

Figura 1. Modelo de dominio. ....	26
Figura 2. Resultados de la métrica para la calidad de la especificación de los requisitos de <i>software</i> . ....	32
Figura 3. Modelo lógico del Subsistema de asignación de comensales. ....	39
Figura 4. Modelo físico del Subsistema de asignación de comensales. ....	40
Figura 5. Arquitectura del Subsistema de asignación de comensales. ....	44
Figura 6. Integración del Subsistema de asignación de comensales. ....	46
Figura 7. Diagrama de despliegue. ....	49
Figura 8. Resultado de las pruebas. ....	55

## Introducción

Los sistemas y las tecnologías de información han revolucionado en gran medida el desarrollo de la humanidad, ejerciendo gran influencia en los diferentes sectores de la sociedad y cambiando la forma en que operan las organizaciones y empresas actuales. A través de su uso se logran importantes mejoras, pues suministran una plataforma de información necesaria para la toma de decisiones. Su implantación logra ventajas competitivas y optimizan los procesos de manipulación de la información por lo que actualmente están siendo adoptados por muchas empresas en el mundo.

Cuba como el resto de los países del mundo, se ha venido involucrando gradualmente sobre todo en los últimos años en esta revolución informática. Este proceso tuvo sus inicios con el surgimiento de la batalla de ideas y fue impulsado en gran medida por el Comandante en Jefe Fidel Castro Ruz que percibió en la informática una rama que podría convertirse en una de las más productoras de recursos a nuestra economía. Por esta razón se crea en el año 2002 la Universidad de las Ciencias Informáticas (UCI), institución que se ha convertido en el mayor centro productor de *software* del país. En la actualidad gracias al esfuerzo del gobierno y en medio de una crisis económica que afecta a toda la humanidad, cada vez es mayor el número de empresas e instituciones cubanas que informatizan sus procesos.

En este centro los sistemas informáticos son parte imprescindible de la vida diaria del personal que vive y labora en esta institución. La gran concentración de recursos de *hardware* conectados, disponibilidad de personal que colabora con el desarrollo de proyectos, el gran número de personas que poseen una cultura informática amplia y la prestación de gran variedad de servicios similares a los que se pueden encontrar en cualquier comunidad con la tendencia al incremento de los mismos, hacen de esta universidad un laboratorio ideal para desarrollar un modelo de informatización de la sociedad.

La UCI cuenta con varios centros productivos, dentro de los cuales se encuentra el Centro de Informatización Universitaria (CENIA) que está compuesto por proyectos encargados de informatizar los procesos que se desarrollan en la institución.

Una de las actividades de gran importancia que se desarrolla a diario en la universidad es la alimentación del personal y el acceso del mismo a los comedores, para lo cual se realiza la asignación de comensales por las diferentes puertas de los complejos y se controla su acceso mediante un carnet de identificación.

La asignación de comensales por las diferentes puertas de los comedores se realiza en la Dirección de Alimentación y se utiliza para ello el Sistema de Gestión de Alimentación. Este sistema cuenta con

varios años de explotación y ha sido de gran utilidad para la organización de esta actividad, sin embargo todavía persisten problemas que atentan contra la correcta ejecución del proceso de asignación de comensales, como son:

- El sistema está desarrollado sobre tecnologías propietarias y funciona solamente en el sistema operativo *Windows*.
- Debido al gran volumen de comensales que acceden a los comedores, los cambios por altas, bajas y cambios de credencial son muy comunes por lo que se necesita actualizar diariamente el personal que está activo. Esta actividad se realiza actualmente a través de servicios web, tomando la información de 4 bases de datos diferentes, lo cual requiere de un alto tiempo de ejecución y trae consigo además inconsistencia e inexactitud en dichos datos.
- La organización y clasificación de las personas dadas sus características no es correcta, pues estas pueden ser agrupadas por más de un criterio (Ejemplo: solapín, nombre, No. Carnet) y pertenecer a varios grupos a la vez, lo que trae consigo redundancia en la información, manejo innecesario de datos y poca exactitud a la hora de contabilizar, realizar cálculos y estadísticas. Esto influye en la generación de distribuciones con errores, pues posibilita que las personas puedan ser asignadas a más de una puerta en un mismo evento.
- La navegabilidad del sistema es un problema que dificulta el trabajo de los operadores e influye también en el tiempo de ejecución, debido a que en algunas ocasiones se hace necesario cerrar algunas interfaces, para que estas actualicen los datos que han sido modificados y en otras se necesita reiniciar la propia aplicación para poder realizar algunas funcionalidades.

Una vez analizada la problemática expuesta anteriormente surge el siguiente **problema de la investigación**: ¿Cómo lograr un proceso de asignación de comensales en la Universidad de las Ciencias Informáticas con mayor organización y agilidad?

El **objeto de estudio** de la presente investigación lo constituyen los procesos de asignación de personas, cuyo **campo de acción** se enmarca en los procesos de asignación de comensales en la UCI.

Se define como **objetivo general** de la investigación: Desarrollar el Subsistema de asignación de comensales que permita la gestión de los mismos en la Universidad de las Ciencias Informáticas con mayor organización y agilidad, utilizando para ello tecnologías y herramientas libres.

Para complementar el objetivo de la investigación se han formulado los siguientes objetivos específicos:

## Objetivos específicos:

- 1- Realizar el análisis y diseño del Subsistema de asignación de comensales.
- 2- Implementar los módulos del Subsistema de asignación de comensales.
- 3- Probar el correcto funcionamiento del subsistema desarrollado.

Para cumplir con los objetivos y solucionar la problemática planteada se proponen las siguientes **tareas investigativas:**

1. Desarrollo del marco teórico de la investigación.
2. Identificación de los principales conceptos que intervienen en los procesos del negocio y las relaciones entre estos.
3. Estudio de las tendencias y técnicas relacionadas con este tema en la actualidad y en las que se apoyan para la solución del problema a enfrentar.
4. Caracterización de las tecnologías, lenguajes y herramientas, para el desarrollo de la aplicación.
5. Descripción de los procesos relacionados con la asignación de personas a los comedores de la UCI.
6. Identificación de los requerimientos relacionados con la asignación de personas a los comedores de la UCI.
7. Diseño de los prototipos de interfaz.
8. Diseño de los casos de prueba basado en los requerimientos.
9. Diseño de la base de datos del subsistema y descripción de sus componentes.
10. Descripción de los elementos principales de la arquitectura del *software*.
11. Implementación de los requisitos funcionales definidos.
12. Aplicación de los casos de pruebas al subsistema para garantizar un buen funcionamiento de la aplicación.

Para la realización de esta investigación se plantea la siguiente **idea a defender**: desarrollando el subsistema de asignación de comensales se agilizarán y facilitarán los procesos de asignación y control de acceso de comensales en la universidad.

## Estrategias de la investigación.

Con el objetivo de obtener los conocimientos a partir de los datos de la práctica y de la teoría precedente y además elaborar una estrategia general para enfrentar el problema que se investiga se utilizan los siguientes **métodos científicos de la investigación:**

**Método científico:** es la forma de abordar la realidad, de estudiar la naturaleza, la sociedad y el pensamiento, con el propósito de descubrir su esencia y sus relaciones.

## **Métodos teóricos:**

Análisis Histórico - Lógico: se analizó la forma en que se realiza la asignación de personas actualmente en los distintos niveles, internacional, nacional y en la universidad, así como las técnicas, sugerencias y soluciones similares ya existentes, para poder solucionar el problema planteado.

## **Métodos empíricos:**

Observación: se estudiaron soluciones informáticas similares a la que se desea desarrollar en la presente investigación, con el propósito de ver el funcionamiento de las mismas.

Entrevista: se realizaron entrevistas con personal de experiencia, con el fin de ayudar a definir un buen desarrollo del subsistema.

Como posibles resultados de la investigación:

- ✓ Se obtendrá una base sólida de fundamentos teóricos asociados a los procesos de asignación y la documentación referente al desarrollo de la solución.
- ✓ Una aplicación de escritorio, multiplataforma, desarrollada con herramientas libres, capaz de gestionar la asignación de comensales en la UCI.
- ✓ Un aumento en la organización y agilidad en los procesos de asignación de comensales en la UCI.

## **Estructura del documento de tesis.**

**Introducción:** se hace la introducción de la investigación, se explica la necesidad de la misma y el problema que se necesita resolver.

**Capítulo 1. Asignación de comensales:** en este capítulo se presentan los elementos teóricos que sirven de base a la investigación del problema planteado, analizando los principales conceptos relacionados con el objeto de estudio. Se realiza una valoración del estado del arte, analizando sistemas semejantes existentes y se caracterizan las herramientas, lenguajes de programación y tecnologías seleccionadas para el desarrollo del subsistema.

**Capítulo 2. Propuesta de solución:** en este capítulo se realiza el análisis y el diseño de la propuesta de solución donde se identifican las metas globales, se analizan las perspectivas del cliente y sus necesidades para posteriormente facilitar la implementación del subsistema. Se muestra el Modelo del Dominio correspondiente, se identifican las funcionalidades que debe tener el subsistema, se describe

la composición del mismo, así como la arquitectura que será utilizada y la estructura de la base de datos.

**Capítulo 3. Validación de la solución:** en este capítulo se presenta el diagrama de despliegue del subsistema y se describe el estándar de código utilizado. Se describen las pruebas realizadas al subsistema y los casos de prueba para validar la calidad de cada una de las funcionalidades de la propuesta de solución.

**Conclusiones:** se evidencia lo que se hizo en el transcurso de la investigación y los resultados obtenidos con la misma.

**Recomendaciones:** se recomiendan aspectos que se consideran necesarios incluir en la solución en caso de posteriores modificaciones.

**Referencias:** se exponen datos que permiten identificar alguna parte de la investigación donde se hizo referencia a algún autor, libro, revista, sitio web, tesis, documento o artículo.

## Capítulo I: Asignación de comensales

### Introducción

En el presente capítulo se presentan elementos teóricos que son la base de la investigación del problema planteado, analizando los principales conceptos relacionados con el objeto de estudio. Se realiza una valoración del estado del arte, analizando sistemas semejantes existentes tanto en el ámbito nacional como en el internacional, además se efectúa una caracterización de las herramientas, lenguajes de programación y tecnologías seleccionadas para el desarrollo del subsistema.

### 1.1 Conceptos relacionados con el dominio del problema

Es necesario el conocimiento de algunos conceptos importantes que se ponen de manifiesto durante el desarrollo de la investigación, para un mejor entendimiento del problema planteado. A continuación se hace referencia a dichos conceptos, elaborados teniendo en cuenta el ámbito de la asignación de comensales en el que se ponen de manifiesto los mismos y las definiciones que le son otorgadas por la Real Academia Española las cuales pueden ser consultadas en el glosario de términos:

- ✓ Comensales: son las personas que comen en un determinado lugar. En el caso de la UCI la gran mayoría de las personas que estudian y trabajan en ella, acceden a los servicios de alimentación en el comedor donde le corresponde.
- ✓ Sede: es el lugar o la institución donde se despliega y se explota el Subsistema de asignación de comensales.
- ✓ Punto de control: es un punto específico donde se permite o se niega el acceso de una persona. Se refiere en este caso a las puertas de acceso de cada comedor donde existe un mecanismo que controla la entrada de los comensales al mismo.
- ✓ Grupo: en este caso son un conjunto de personas (comensales) que tienen al menos una característica en común y que se agrupan con el objetivo de lograr una mayor organización en el subsistema.
- ✓ Evento: el evento es un suceso que ocurre en un momento determinado, con un principio y un fin previamente conocido, independientemente de que pueda poseer una periodicidad establecida. Estos eventos en el caso de la asignación de comensales están relacionados principalmente con la alimentación y pueden ser por ejemplo el desayuno, el almuerzo, la comida y la cena de un comedor determinado.
- ✓ Asignación: puede definirse en el ámbito de la asignación de comensales como la acción de decidir las personas que tienen acceso a un determinado punto de control, de un comedor, para

# Capítulo I. Asignación de comensales

---

efectuar los eventos que se desarrollan en el mismo.

- ✓ Distribución: la distribución puede definirse como el resultado final del proceso de asignación, con el cual se logra que cada persona pueda acceder a un punto de control específico, durante la ocurrencia de un evento.

## 1.2 Estudio del estado del arte

Actualmente en el mundo existen múltiples sistemas capaces de realizar asignaciones de todo tipo y que poseen una amplia gama de funcionalidades que facilitan esta tarea en almacenes, empresas e instituciones en general. Por esta razón se hizo necesario realizar un estudio previo sobre los mismos para determinar cuál de ellos posee características y soluciones que puedan ayudar a resolver las necesidades planteadas en el problema descrito anteriormente.

### Nivel internacional

#### ➤ AccPro

Es un moderno *software* desarrollado en Argentina cuyo objetivo principal es tener el control centralizado del estado de todos los accesos de una empresa. Cuenta con múltiples funcionalidades y entre las que presentan relación con las tareas de asignación se pueden citar las siguientes:

- ✓ Funciones de arrastrar y soltar para asignar o quitar privilegios de acceso:

Para facilitar el proceso de asignar privilegios de acceso en determinadas puertas, el sistema permite realizar la tarea de un modo totalmente gráfico, seleccionando los usuarios desde una barra lateral derecha y arrastrándolos hasta el acceso donde se desea permitir el ingreso. También es posible otorgar múltiples accesos en una sola operación arrastrando a los usuarios sobre el ícono que representa al grupo de acceso. En este caso los usuarios serán habilitados en cada una de las puertas definidas dentro del grupo. Por otra parte al arrastrar los usuarios sobre la papelera de reciclaje se quitará automáticamente todos los privilegios de acceso concedidos anteriormente.

- ✓ Pantallas de monitoreo de eventos configurables:

Es posible configurar qué eventos se desean monitorear: Ingreso o Egreso, Acceso bloqueado, Entrada y Salida de trabajo, entre otros. Además permite saber cuáles son los accesos de interés para cada operador y cuál es el modo de visualización preferido por cada uno.

- ✓ Supervisión de personal habilitado en cada acceso:

Es posible visualizar todos los usuarios habilitados en un acceso/puerta. En una pantalla se podrán visualizar todos los datos de los mismos, la restricción horaria que poseen y su estado Activo Si/NO.

# Capítulo I. Asignación de comensales

---

Se pueden definir el tiempo de apertura por nodo. También es posible eliminarlos de la lista marcándolos y enviándolos a la papelera de reciclaje (LARCON-SIA S.R.L, 2009).

## ➤ **Advanced Software – Control de Comedores**

Desarrollado por la empresa *Advanced Software* que comercializa y distribuye el *software* de Control de Comedores, como módulo opcional e integrado en la aplicación de Control de Presencia o de Control de Accesos, ya sea en las versiones bajo base de datos Access o SQLServer / ORACLE, o bien de forma de aplicación independiente. Es una aplicación destinada a resolver con gran seguridad el control de acceso del personal a los comedores de cualquier empresa con posibilidad de definir tipos de menú y valorarlos económicamente. De forma lógica y natural, hereda las máximas prestaciones en cuanto a control de accesos se refiere, en función de la aplicación base en la que se sostiene e integra o bajo la que ha sido adquirida como son la posibilidad de controlar centros remotos, con gestión de calendarios y puertas configurables. Incluye los elementos de gestión y clasificación de trabajadores por empresa, departamento, sección y categoría; la gestión de tarjetas de acceso; el registro y control de acceso a comedores, así como la creación de informes y estadísticas del sistema. De la misma forma pueden configurarse niveles de seguridad, indicando a qué puertas se tiene acceso y en qué franjas horarias independientes por día, asociándolos finalmente a los trabajadores. Se pueden agregar módulos opcionales como: Módulo de Exportación a Nómina, Módulo de Importación/Actualización de datos frecuentes y Módulo de Introducción de Marcajes mediante huella digital, entre otros (*Advanced Software*, 2011).

## **Nivel nacional**

### ➤ **Balance de Aguas**

Es un sistema para la asignación de las aguas disponibles en el país desarrollado por el Centro de *Software* de Holguín en el año 2010. La idea principal del sistema es controlar las entregas de agua que se establecen a diferentes entidades, a la vez que se reciben las demandas de estos, teniendo en cuenta las condiciones de cada una de las fuentes de abasto al inicio del año.

El sistema está estructurado en 4 opciones: Configuración, Gestión de datos, Gestión de procesos y Salidas del sistema o Reportes. Mediante la configuración se establecen los usuarios del sistema y se define el año del primer balance a realizar. Se establece además una gestión de provincias y municipios para la posibilidad de una nueva división político administrativa o cambios nacionales (Peña et al, 2010).

# Capítulo I. Asignación de comensales

---

Como parte de la dinámica de la elaboración del balance, el sistema cuenta con las opciones para la entrada y gestión de datos: organismos con sus ramas y actividades asociadas, entidades usuarias del agua y las actividades que ellas realizan vinculadas al manejo de estos recursos a nivel de territorio, así como las fuentes de agua (regulada, no regulada y subterránea). Una vez que se guarden los datos necesarios, con una configuración previa del año en el cual se desee trabajar, el sistema está listo para trabajar en base a las asignaciones de agua correspondientes. Se permiten gestionar 3 procesos para la realización del balance:

- ✓ Planificación inicial: controla las condiciones en que se encuentran las fuentes de agua a inicios de año, con el propósito de realizar posteriormente el balance en base al agua disponible.
- ✓ Demanda: controla las solicitudes de agua de las entidades de cada uno de los organismos, además se le asignarán fuentes específicas para el abasto y en caso de utilizarse pozos o equipos de bombeo se asocian a la demanda y se guarda para tener en cuenta su gasto o capacidad de instalación, además de registrar las tarifas, la eficiencia en el riego, el nivel de actividad a abastecer y las normas cubanas (netas y brutas).
- ✓ Asignación: gestiona las asignaciones de agua a las entidades clientes una vez realizadas las demandas, según el volumen que va quedando disponible según la fuente asociada a una actividad seleccionada (Peña et al, 2010).

El sistema permite además, generar e imprimir recuperaciones de la información obtenida en formato .pdf a través de diferentes criterios en dependencia del reporte que se escoja, ya sea de un año, trimestre, organismo o fuente de agua específicos; o según una zona, cuenca hidrográfica, sistema, provincia, dependiendo además de los permisos según el rol. Se obtienen salidas como:

- ✓ Modelo oficial de asignación de agua (para cada entidad).
- ✓ Propuesta de Balance de agua.
- ✓ Balance por tipo de fuente, por organismos y por entidades.
- ✓ Resumen general de utilización de las aguas.
- ✓ Balance general de fuentes y organismos para la Gaceta Oficial.
- ✓ Resúmenes comparativos con años anteriores.

## ➤ **TEAMSOF**T

Es un sistema de soporte a la decisión diseñado con el propósito de apoyar el proceso de asignación de personal en las organizaciones de *software*. Fue desarrollado en el Instituto Superior Politécnico “José Antonio Echeverría” (CUJAE) en el 2010. La herramienta sustenta un modelo formal de

# Capítulo I. Asignación de comensales

---

asignación que integra factores que contribuyen a la asignación individual de roles en el proyecto y a la formación del equipo como un todo. Está desarrollada en Java y posee una interfaz amigable y flexible, capaz de tratar los factores ya sea como objetivos o como restricciones, y de ajustar algunos parámetros del modelo.

La herramienta estructura el proceso de asignación de personal en dos etapas, asignación del jefe de proyecto y asignación del equipo. Antes de establecer el equipo de proyecto definitivo, se pueden evaluar tantas propuestas de equipos como deseen, considerado fijo al jefe de proyecto, quien fue previamente asignado.

Durante estas evaluaciones resulta posible ajustar algunos parámetros del modelo, seleccionar los objetivos y las restricciones a tomar en cuenta, así como seleccionar el método y el algoritmo de solución. A pesar que estas últimas facilidades están asociadas a cuestiones más técnicas, el hecho de que la herramienta posea una interfaz flexible y amigable permite que resulte fácil para el decisor interactuar con el sistema, aún sin ser conocedores de la temática de investigación de operaciones y de no dominar el modelo, y los métodos y algoritmos de solución que subyacen.

Los resultados obtenidos en pruebas muestran que su aplicación contribuye a que el proceso de asignación, en general, se torne más objetivo y transparente y ha probado ser un instrumento especialmente útil durante el proceso de asignación de personas a los equipos de proyectos de *software* (Ampuero et al, 2010).

## Nivel universidad

### ➤ Sistema de Gestión de Alimentación

Este sistema brinda la posibilidad de realizar la asignación de comensales en la UCI, atendiendo a una determinada estructura de comedores. Permite seleccionar entre distribuciones existentes para fechas y períodos especiales, además de ofrecer reportes como cantidad de comensales que han pasado y desglosarlo por puerta o por tipo. Actualmente se encuentra en uso y ha demostrado ser una aplicación funcional durante los años de explotación, sin embargo presenta un conjunto de deficiencias que dificultan el trabajo de sus operadores. Entre las funcionalidades principales con que cuenta se pueden encontrar las siguientes:

- ✓ Gestión de eventos: muestra los eventos relacionados con la estructura de comedores con los siguientes datos: nombre, hora de inicio y fin, si el evento está activo, y la cantidad de raciones permitidas.
- ✓ Gestión de estructuras: permite construir y editar la estructura de los comedores, permitiendo la adición y eliminación de los puntos de control de acceso a los mismos.

# Capítulo I. Asignación de comensales

---

- ✓ Gestión de grupos: posibilita organizar al personal de la universidad en grupos más pequeños que responden a determinadas características.

## ➤ Sistema de Control de Acceso

Este sistema fue desarrollado en la UCI en el año 2010 con el objetivo de automatizar el control del personal que accede a la universidad así como a las diferentes áreas dentro de la misma. Para ello cuenta con un sistema administrador encargado de estructurar y configurar los puntos de control que van a ser controlados y asignarle el personal que tiene acceso a cada uno de ellos y otro sistema cliente que es el encargado de la identificación del personal en los puntos de control.

Entre las funcionalidades principales que se identificaron en esta aplicación se pueden citar las siguientes:

- ✓ Gestionar sede: permite crear la sede o institución donde se aplicará el sistema poniendo de manifiesto la genericidad y usabilidad del mismo en cualquier institución donde se desee controlar el acceso.
- ✓ Gestionar grupos: permite organizar y crear grupos de personas según las preferencias del administrador del sistema o teniendo en cuenta características específicas.
- ✓ Gestionar estructuras y locales: brinda la posibilidad de construir y modificar las estructuras a las que se va a controlar el acceso, con sus respectivos puntos de control.
- ✓ Gestionar eventos: mediante esta funcionalidad se crean y modifican los eventos que se desarrollan en las diferentes estructuras y locales, a los cuales el personal solicita el acceso.
- ✓ Asignar grupos a puntos de control: una vez que se han creado los diferentes grupos de personas estos pueden ser asignados a un punto de control determinado para controlar el acceso de estas personas por ese punto.

El sistema cuenta también con opciones de reportes donde se pueden consultar datos como la cantidad de accesos que registró una determinada persona por un punto de control específico y la cantidad de accesos por puntos de control, entre otros.

Después de realizar un análisis sobre estos sistemas se puede concluir que AccPro y Control de Comedores poseen algunas funcionalidades como el monitoreo de eventos y la clasificación y gestión de las personas, todas con un alto grado de seguridad y que podrían ser beneficiosas para la construcción del subsistema. Sin embargo se observan inconvenientes que las hacen soluciones no factibles para gestionar los comensales en la UCI como por ejemplo, son aplicaciones propietarias que no se tiene acceso a su código, por lo que no se pueden modificar y adaptar a las necesidades de la universidad y por las que hay que pagar costosas licencias para su utilización. Además requieren de

# *Capítulo I. Asignación de comensales*

---

equipos y tecnologías de punta con las cuales la UCI no cuenta como es en el caso de AccPro de lectores de huellas dactilares, sistemas de alarmas y cámaras de seguridad. Estos sistemas también tienen el inconveniente de estar desarrollados con un ambiente gráfico de alto nivel y la configuración inicial de las estructuras requiere de representaciones gráficas muy complejas.

Las aplicaciones de Balance de agua y TEAMSOFTE a pesar de estar desarrolladas sobre la base del código abierto, no son sistemas generales que se puedan implantar con otro objetivo, que no sea para el cual se desarrollaron.

En cuanto a las aplicaciones de Gestión de Alimentación y Control de Acceso desarrollados en la universidad, se llega a la conclusión que son en gran medida la base y constituyen el punto de partida para el subsistema a desarrollar. El sistema de Control de acceso es una solución que se acerca mucho a las necesidades planteadas por la Dirección de Alimentación. Sin embargo presenta funcionalidades que no se corresponden con el ámbito de los comedores y que no son necesarias para la solución del problema, como son el registro del personal que visita la institución, el control de vehículos y de áreas a visitar. Carece además de otras, como son el balance de los puntos de control, y la funcionalidad de organización de las personas genera redundancias en el sistema pues permite que estas puedan ser agrupadas por más de un criterio. Además este sistema depende totalmente de la red, no contando con un mecanismo de respaldo para el caso de la ocurrencia de algún fallo en la conexión.

El Sistema de Gestión de Alimentación a pesar de estar brindando servicios en estos momentos, presenta deficiencias que dificultan la correcta asignación del personal a los diferentes puntos de control de los comedores como se explicó anteriormente. Es por ello que se necesita rediseñarlo para mejorar aspectos como la navegabilidad del sistema ya que en ocasiones es necesario cerrar algunas interfaces para actualizar los datos que han sido modificados y en otras se necesita reiniciar la aplicación para poder realizar algunas funcionalidades. El tiempo de ejecución de algunas funcionalidades sobre todo las correspondientes a la actualización del personal, es muy elevado, y los datos que se obtienen en muchas ocasiones son inexactos pues están distribuidos en bases de datos diferentes. La organización de las personas en grupos no es correcta, ya que las mismas pueden pertenecer a varios grupos a la vez, por lo que no es posible contar con la cifra exacta de personas que han sido asignadas y las que faltan por asignar. Esto puede traer como resultado una distribución con errores, donde puede suceder que las personas tengan acceso a más de una puerta durante un mismo evento. Se necesita además contar con la documentación necesaria que respalde el desarrollo del subsistema y permita un correcto mantenimiento del mismo.

# Capítulo I. Asignación de comensales

---

A esto se suma que ambas aplicaciones están desarrolladas sobre tecnologías propietarias y que solamente funcionan en el sistema operativo *Windows* lo que no cumple con las políticas del centro CENIA, donde se desarrollan aplicaciones de código abierto en vistas a la migración de los sistemas de la universidad y del país a este tipo de *software*.

Con este análisis se lograron identificar características y funcionalidades que son comunes en la mayoría de estos sistemas como: gestión de grupos, gestión y clasificación del personal, gestión de estructuras y manejo de registros las cuales serán incluidas en el Subsistema de asignación de comensales con el cual se pretende cubrir las necesidades de los sistemas antes descritos y mejorar los procesos de asignación de comensales en la universidad.

## 1.3 Herramientas y tecnologías de *software* utilizadas en el desarrollo del subsistema

Teniendo en cuenta las políticas del centro CENIA para el desarrollo de aplicaciones de escritorio y las características del subsistema, se estudiaron y seleccionaron las herramientas y tecnologías a utilizar en el desarrollo del mismo. Un factor importante en esta selección, lo constituyó el hecho de que las herramientas fueran de código abierto considerando la necesidad de la migración de las instituciones cubanas a este tipo de *software* dadas las ventajas que este brinda con respecto a los del tipo propietario, entre las que se pueden citar las siguientes:

- Evita el alto costo que supone el pago de las licencias de *software* propietario lo cual eleva el costo total de propiedad de las computadoras.
- Su utilización no implica gastos adicionales por concepto de cambio de plataforma de *software*, ya que es operable en el mismo soporte de *hardware* con que se cuenta.
- Se garantiza su utilización con un mínimo de recursos, puesto que no hay que pagar nada por su uso, modificación y distribución.
- Evita la dependencia tecnológica, de empresas extranjeras, permitiendo la generación de riquezas y recursos nacionales.
- Es posible su adaptación a diferentes contextos de aplicación gracias al acceso al código fuente, lo que garantiza un camino adelantado en ese proceso.
- Se pueden detectar y corregir, con facilidad, las fallas de seguridad informática o errores de programación y obtener de forma rápida las actualizaciones y nuevas versiones.

## Proceso de desarrollo ágil con segundo nivel de CMMI (*Capability Maturity Model Integration*)

Un proceso de desarrollo de *software* tiene como propósito la producción eficaz y eficiente de un producto *software* que reúna los requisitos del cliente. En los últimos tiempos la cantidad y variedad de

# Capítulo I. Asignación de comensales

---

procesos de desarrollo de *software* han aumentado de forma impresionante. Las metodologías de desarrollo ágil surgen como una extensión a las metodologías tradicionales para mejorar el desarrollo de sistemas, según el tipo de proyecto y empresa, añadiendo y mejorando las prácticas de desarrollo de *software*.

CMMI es una aproximación a la mejora de los procesos usados para guiar el desarrollo de los proyectos. Requiere una previa preparación, a menudo la ayuda de un consultor para determinar los procesos y de un asesor externo para evaluar un grupo o proyecto y ver en qué nivel se encuentra (valorando del 1 al 5) (Carvajal, 2008).

CMMI consta de 5 niveles de madurez que están clasificados de la siguiente forma: Inicial, Gestionado, Definido, Gestionado cuantitativamente y Optimizado, diseñados de manera que los niveles inferiores proveen las bases para que de forma progresiva se alcancen los superiores.

La UCI se encuentra inmersa en un proceso de mejora de los procesos que se desarrollan como parte de la construcción del *software* basado en el modelo de CMMI. Este proceso de mejora tiene como objetivo que la universidad alcance el nivel 2 del modelo CMMI que muchas veces es el más difícil de alcanzar y esto es porque requiere cambios en la forma de trabajar de la empresa o proyecto.

Lo que se pretende en este nivel es conseguir que en los proyectos haya una gestión de los requisitos y que los procesos estén planeados, ejecutados, medidos y controlados (Murugarren, 2003).

En la universidad la mayoría de los proyectos que actualmente existen son de corta duración, el equipo de desarrollo con que se cuenta generalmente es pequeño, la implementación de los sistemas es lo más importante y constituye una prioridad en el equipo de desarrollo. Es por esto que en el centro CENIA se aplica el proceso de desarrollo de *software* con un enfoque ágil para el nivel 2 de CMMI, cuyo ciclo de vida está constituido por las siguientes áreas que se describen brevemente a continuación.

## **Administración de Requerimientos (REQM)**

Esta área de proceso tiene como propósito mantener bajo control los requerimientos que el producto a desarrollar deberá satisfacer. Un tema fundamental planteado en esta área de proceso es que cualquier cambio realizado a los requerimientos se efectúe de manera controlada y que el resto de los artefactos del proyecto (planes, especificaciones, diseño) se mantengan consistentes.

# Capítulo I. Asignación de comensales

---

## **Planificación del Proyecto (PP)**

Esta área de proceso tiene como propósito establecer y mantener el plan que será empleado para ejecutar y monitorizar el proyecto. El plan se desarrolla sobre la base de los requerimientos administrados por el área REQM.

Dentro de esta área se incluyen todas las actividades necesarias para determinar el alcance del proyecto, estimar esfuerzo y costos, establecer el cronograma, identificar riesgos y obtener el compromiso de todos los involucrados respecto al plan de proyecto. Las actividades de esta área suelen implementarse mediante la combinación de varios elementos. Por un lado, será necesario establecer algún tipo de mecanismo de estimación que emplee como entrada los requerimientos del proyecto. También será necesario formalizar el plan de proyecto, el ciclo de vida a emplear y los mecanismos de aprobación.

## **Monitoreo y Control del Proyecto (PMC)**

No tiene sentido formular planes para algo que no se tiene intenciones de gestionar. Esta área de proceso es complementaria y es una consecuencia de Planificación del Proyecto (PP): su propósito es monitorear la ejecución del proyecto empleando para ello el plan y gestionar acciones correctivas en el caso de detectarse desvíos.

Para poder cumplir con estos objetivos será necesario implementar prácticas de seguimiento, tales como el reporte de horas trabajadas en el proyecto, el informe de avance periódico y revisiones en puntos particulares del proyecto. Si bien esto suena sencillo, conseguir cambiar la cultura es una tarea muy compleja.

## **Medición y Análisis (MA)**

Una premisa, presente en todos los movimientos de calidad es que lo que no puede medirse, no puede mejorarse. Esta área de proceso apunta, justamente, a desarrollar y mantener capacidades de medición que permitan satisfacer las necesidades de información de la organización.

## **Aseguramiento de la Calidad de Productos y Procesos (PPQA)**

Una vez establecidos procesos y estándares será necesario evaluar su aplicación. El objetivo de esta área es proveer una evaluación objetiva de los procesos y de los artefactos producidos. Es importante aclarar que las prácticas de esta área implican:

- ✓ Evaluar los procesos ejecutados, los artefactos producidos y los servicios provistos contra los estándares y descripciones de proceso aplicables.

# Capítulo I. Asignación de comensales

---

- ✓ Identificar no conformidades, comunicarlas a los responsables y asegurar su resolución.
- ✓ Informar a los interesados (básicamente, el equipo de proyecto y la gerencia) el resultado de las actividades de aseguramiento de la calidad.

## **Administración de la Configuración (CM)**

Esta área de proceso tiene como propósito mantener la integridad de todos los artefactos (entregables o no) producidos por el proyecto. Lo cual implica identificar los elementos de configuración, realizar sobre ellos cambios de manera controlada, generar y mantener líneas base y proveer información precisa acerca del estado de la configuración a todos los interesados.

Esta área de proceso es, probablemente, una de las más difíciles de implementar de todas las incluidas en este nivel. Además de tener problemas para planificar y ejecutar sus actividades de acuerdo a esos planes, las organizaciones de nivel uno suelen tener serias complicaciones para identificar y mantener las versiones correctas de sus productos y artefactos asociados. Es bastante común encontrarse con organizaciones que tienen múltiples versiones activas de una misma aplicación, sin poder llegar a controlarlas del todo e invirtiendo enormes recursos en arreglar los mismos problemas una y otra vez. Estas prácticas apuntan, justamente, a resolver este tipo de problemas.

## **Administración de Acuerdos con Proveedores (SAM)**

Esta área de proceso apunta a resolver otro de los problemas habituales en muchas organizaciones: el de la tercerización. Si bien está originalmente pensada para todo lo relacionado con la adquisición de productos que vayan a ser incorporados en la solución a entregar al cliente, las prácticas incluidas aquí también sirven para todo aquello que sea necesario comprar pero que no será finalmente entregado al cliente, como por ejemplo herramientas de desarrollo (Chrissis et al, 2009).

### **1.4 Herramientas de modelado**

Las herramientas de modelado, permiten crear un "simulacro" del sistema, a bajo costo y riesgo mínimo. A bajo costo porque, al fin y al cabo, es un conjunto de gráficos y textos que representan el sistema, pero no son el sistema físico real (el cual es más costoso). Además, minimizan los riesgos, porque los cambios que se deban realizar (por errores o cambios en los requerimientos), se pueden realizar más fácil y rápidamente sobre el modelo y sobre el sistema ya implementado (Scribd Inc, 2011).

# Capítulo I. Asignación de comensales

---

## Visual Paradigm 8.0

Es una herramienta profesional muy potente que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Diseñado para varios tipos de usuarios, incluyendo Ingenieros de *Software*, Analistas de Sistemas, Analistas de Negocio y Arquitectos de Sistema. Permite realizar Diagramas de procesos de negocios, Modelado UML, Modelos de casos de usos, Modelos de actividad, de interacción, de bases de datos, de entidad-relación. Además posee integración para varios Entornos Integrados de Desarrollo (IDE), puede realizar Ingeniería de Código y también generar documentación, entre otras cosas; todo bajo un modelo colaborativo. Soporta todas las necesidades de diseño y modelado a lo largo del ciclo de vida de desarrollo de *software*, es una herramienta que ayuda a construir aplicaciones de calidad, de manera más rápida, óptima y más barata (Grupo Satelite s.a, 2011).

Es posible generar código desde Visual Paradigm para plataformas como .NET, Java y PHP, así como obtener diagramas a partir del código, esto es de gran utilidad pues ahorra tiempo a los desarrolladores y reduce las posibilidades de cometer errores. Brinda la posibilidad de obtener una base de datos relacional y el código necesario para acceder a esta a partir de un Diagrama Entidad Relación, además se conecta fácilmente a varios servidores de base de datos. Establece interoperabilidad con otras aplicaciones como el *Microsoft Visio* y el *Rational Rose* y permite documentar todo el trabajo y especificaciones de Casos de Usos sin necesidad de utilizar herramientas externas, por ejemplo editores de texto, utilizando plantillas que se encuentran o que pueden ser creadas por los usuarios. Disponible en múltiples lenguajes y plataformas: Microsoft *Windows* (98, 2000, XP, o Vista), GNU Linux, Mac OS X, Solaris o Java (Visual Paradigm, 2010).

## Evolus Pencil 1.3.4

Es una herramienta libre y de código abierto para crear diagramas y prototipos de interfaz gráfica de usuario que todos puedan usar (Evolus, 2010).

Características principales:

- ✓ Construcción de prototipos.
- ✓ Conexión entre páginas.
- ✓ Exportación a formatos HTML, PNG, documento Word y PDF.
- ✓ Multiplataforma: Puede ser instalado tanto en *Windows* como *Linux*, además puede agregarse como complemento para el navegador *Mozilla Firefox*.
- ✓ Tipo de funcionamiento: arrastrar y soltar.

## 1.5 Herramientas de programación para el desarrollo de la aplicación

Las herramientas de programación, son aquellas que permiten realizar aplicativos, programas, rutinas, utilitarios y sistemas para que la parte física del computador u ordenador, funcione y pueda producir resultados (Lenguajes de Programación, 2009).

En la actualidad existen varias herramientas de programación en el mercado tanto para programadores expertos como para inexpertos de las cuales se seleccionaron las siguientes:

### Lenguaje de Programación. Java

Un Lenguaje de Programación es un conjunto de reglas, notaciones, símbolos y/o caracteres que permiten a un programador poder expresar el procesamiento de datos y sus estructuras en la computadora (Terrence et al, 1998).

La gran mayoría de los lenguajes de programación pueden ser utilizados para expresar cualquier algoritmo, sin embargo resulta importante qué lenguaje se utiliza para la resolución de problemas de programación. Muchas veces la elección de un lenguaje de programación para resolver un problema del mundo real se hace por facilidad, experiencia o simplemente por actualidad, que aunque son razones válidas no garantizan la solución más eficiente ni mucho menos la más barata. Es importante saber elegir el lenguaje en función de compatibilidad, portabilidad y facilidad, todo esto equilibrado a lo que se desea hacer para cumplir los objetivos trazados, en base a esto se seleccionó Java como lenguaje de programación para dar cumplimiento al objetivo general de la investigación.

Java es un lenguaje de programación de código abierto orientado a objetos desarrollado por *Sun Microsystems* a inicios de los años noventa. Es el lenguaje de programación más usado en el mundo y que soporta el encapsulamiento, la herencia y el polimorfismo (los tres pilares del paradigma de la POO). El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria (Masterhacks, 2011).

Posee múltiples características que favorecieron a su selección, entre las cuales se encuentran:

- ✓ **Lenguaje simple:** Java posee una curva de aprendizaje muy rápida. Todos aquellos familiarizados con C++ encontrarán que Java es más sencillo, ya que se han eliminado ciertas características, como los punteros. Debido a su semejanza con C y C++, y dado que la mayoría de la gente los conoce aunque sea de forma elemental, resulta muy fácil aprender Java. Los programadores experimentados en C++ pueden migrar muy rápidamente a Java y ser productivos en poco tiempo.
- ✓ **Orientado a objeto:** Java está totalmente orientado a objetos, proporcionando los mecanismos para que el programador haga utilización de todas las técnicas de diseños y programación

# Capítulo I. Asignación de comensales

---

orientadas a objetos como herencia, polimorfismo, abstracción y concurrencia. El lenguaje va acompañado de numerosas librerías de objetos que cubren todas las áreas, desde los tipos básicos de datos a las interfaces de entrada/salida y de red, pasando por el soporte para la construcción de interfaces gráficas de usuario.

- ✓ **Distribuido:** proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir *sockets* y establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas.
- ✓ **Arquitectura neutral:** Java ha sido desarrollado para crear aplicaciones que funcionen en entornos de red, operando en una amplia variedad de arquitecturas y sistemas operativos. Para conseguir esta independencia, el código fuente Java se compila a un código de bytes de alto nivel independiente de la máquina, este código (*bytecode*) está diseñado para ejecutarse en una máquina hipotética que es implementada por un sistema *runtime*, que sí es dependiente de la máquina (Máquina virtual de Java). Por esta razón, y debido a la naturaleza interpretada del lenguaje, el mismo *bytecode* puede ejecutarse sobre cualquier plataforma sin necesidad de recompilación. Para establecer Java como parte integral de la red, el compilador Java compila su código a un fichero objeto de formato independiente de la arquitectura de la máquina en que se ejecutará.
- ✓ **Robusto:** Java fue diseñado para crear *software* altamente fiable. Para ello proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores ya que se ha prescindido por completo los punteros, y la recolección de basura elimina la necesidad de liberación explícita de memoria.
- ✓ **Seguro:** tiene políticas que evitan que se puedan codificar virus con este lenguaje. Existen varias restricciones que limitan lo que se puede y no puede hacerse con los recursos críticos de una computadora. Se implementaron barreras de seguridad en el lenguaje y en el sistema de ejecución en tiempo real (Schildt, 2008).

## Marco de trabajo

Desde el punto de vista del desarrollo de *software* un marco de trabajo es una estructura conceptual y tecnológica compuesta por librerías, componentes y clases que facilitan el desarrollo ágil, seguro y escalable de una aplicación. En otras palabras, puede considerarse como un marco o modelo de aplicación genérica incompleta y configurable, con un conjunto de bibliotecas orientadas a la reutilización, a la cual pueden añadirse las últimas piezas para construir una aplicación concreta.

# Capítulo I. Asignación de comensales

---

Teniendo en cuenta el lenguaje de programación seleccionado y las características de la aplicación a desarrollar, se seleccionaron los siguientes marcos de trabajo:

## **Spring Framework 3.0.2**

*Spring Framework* es un marco de trabajo de aplicación de código abierto desarrollado para aplicaciones escritas en el lenguaje de programación Java. Es una potente herramienta de gestión de configuración basada en *JavaBeans* 6, aplicando los principios de Inversión de Control (IoC). Esto hace que la configuración de aplicaciones sea rápida y sencilla.

Normalmente se suelen extender clases de manera innecesaria. Especialmente problemática es esta práctica cuando se trata de objetos DAO (Access Object) que crean una conexión a la base de datos ya que si cada usuario crea una conexión a la base de datos y el número de usuarios es alto pueden hacer que el servidor se caiga. Para evitar esto, *Spring* propone como patrón la inyección de dependencias que radica en resolver las dependencias de cada clase (atributos) generando los objetos cuando se arranca la aplicación y luego inyectarlos en los demás objetos que los necesiten a través de métodos set o bien a través del constructor, pero estos objetos se instancian una vez, se guardan en una factoría y se comparten por todos los usuarios (al menos en el caso de *Spring*) y así se evita tener que andar extendiendo clases o tumbar el servidor de la base de datos (Walls, 2008).

Dentro de las ventajas que ofrece *Spring*, se encontró que facilita la manipulación de los objetos, elimina la necesidad de usar distintos y variados tipos de ficheros de configuración, mejora la práctica de programación, además se ha demostrado que más del 70% de los desarrolladores y empresas que lo utilizan han ganado en productividad y reducido el tiempo de desarrollo (SpringSource, 2011).

Dentro del subsistema desarrollado la principal ventaja que brinda es el uso del patrón de inyección de dependencia con el cual se delega la responsabilidad de instanciar los objetos en un archivo de configuración XML, esto resulta muy útil puesto que evita cualquier dependencia que pueda existir entre los objetos, facilitando así el desacoplamiento entre las capas.

## **Hibernate**

*Hibernate* es una herramienta de mapeo objeto/relacional para entornos Java. El término de mapeo objeto/relacional (ORM) se refiere a la técnica de mapear una representación de datos desde un modelo de objeto a un modelo de datos relacionales con un esquema basado en SQL.

Esta herramienta no solamente se ocupa del mapeo desde las clases Java a las tablas de las bases de datos (y desde los tipos de datos de Java a los tipos de datos de SQL), sino que también facilita la consulta y recuperación de datos ofreciendo un lenguaje propio de consultas de datos llamado HQL

# Capítulo I. Asignación de comensales

---

(*Hibernate Query Language*). Esto puede reducir de manera importante el tiempo de desarrollo que se tomaría con el manejo de datos de forma manual en SQL y JDBC. Permite generar bases de datos en cualquiera de los Sistemas Gestores de Bases de Datos Relacionales (SGBDR) entre los cuales se encuentra: *PostgreSQL*, *MySQL*, *Oracle*, entre otros y es un *software* libre bajo la licencia GNU/LGPL (HIBERNATE-Comunity, 2010).

## 1.6 Sistemas Gestores de Bases de Datos

Con la necesidad de lograr la gestión de las bases de datos surgen los Sistemas Gestores de Bases de Datos (SGBD), que según publicaciones, no es más que: una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos. Algunos ejemplos de SGBD son *Oracle*, *PostgreSQL*, *MySQL*, *MS SQL Server*, entre otros (CAVSI, 2009).

### PostgreSQL 8.4

Es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales.

Utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Su desarrollo comenzó hace más de dieciséis años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han tenido en cuenta durante su desarrollo.

*PostgreSQL* posee una gran escalabilidad. Es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta. Además implementa el uso de subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en los que *MySQL* no podría. También tiene la capacidad de comprobar la integridad referencial, así como la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser *Oracle*. Por estos motivos, *PostgreSQL* se puede considerar a la altura de los principales sistemas gestores de bases de datos comerciales y presentarse como un

# Capítulo I. Asignación de comensales

---

importante competidor entre las tecnologías disponibles para el diseño de sistemas (PostgreSQL, 2011).

## **PgAdmin III 1.14.0.**

Es la más popular y completa herramienta de código abierto para administración y desarrollo en la plataforma *PostgreSQL*, la más avanzada base de datos de código abierto en el mundo. Puede ser usado en Linux, *FreeBSD*, Solaris, Mac OSX y *Windows* para administrar *PostgreSQL*, así como comerciales y las versiones derivadas de *PostgreSQL* como *Postgres Plus Advanced Server* y *Greenplumdatabase* (pgadmin.org, 2011).

Está diseñado para responder a las necesidades de todos los usuarios, desde simples consultas SQL hasta para el desarrollo de bases de datos complejas. La interfaz gráfica soporta todas las características y hace fácil la administración. La aplicación también incluye editor de resaltado de sintaxis SQL, servidor-editor de código, SQL/*batch/shell* programación de agente de empleo, motor de replicación y mucho más. La conexión con el servidor puede hacerse a través de TCP / IP o *Unix domain sockets* (en plataformas Unix), y puede utilizar SSL para la seguridad.

Es desarrollado por una comunidad de expertos de *PostgreSQL* en todo el mundo y está disponible en más de una docena de idiomas. Es un *software* libre publicado bajo la licencia BSD13 (pgadmin.org, 2011).

## **DB4O 6.4**

Es un novedoso motor de base de datos orientada a objetos. Sus siglas se corresponden con la expresión "*DataBase4 (for) Objects*", que a su vez es el nombre de la compañía que lo desarrolla: *db4objects, Inc.*

Las claves innovadoras de este producto son su alto rendimiento (sobre todo en modo embebido) y el modelo de desarrollo que proporciona a las aplicaciones para su capa de acceso a datos, el cual propugna un abandono completo del paradigma relacional de las bases de datos tradicionales.

La mayor clave del éxito que está teniendo este motor de base de datos frente a otros competidores que han desarrollado tecnologías similares, es que se ha optado por un modelo de licenciamiento idéntico al utilizado por empresas como *MySQL*: licencia dual GPL/comercial. Es decir, si se quiere desarrollar *software* libre con esta librería, su uso no conlleva ningún coste por licencia; sin embargo si se desea aplicar a un *software* privativo, se aplica otro modelo de licenciamiento concreto (VERSANT CORP, 2010).

## 1.7 Entorno Integrado de Desarrollo (IDE)

Un entorno de desarrollo integrado es un programa compuesto por una serie de herramientas que utilizan los programadores para desarrollar código. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o varios de estos.

Las herramientas que normalmente componen un entorno de desarrollo integrado son las siguientes: un editor de texto, un compilador, un intérprete, herramientas para la automatización, un depurador, un sistema de ayuda para la construcción de interfaces gráficas de usuario y, opcionalmente, un sistema de control de versiones. Hoy en día los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado (por ejemplo C, C++, C#, Java, Python y Visual Basic entre otros) (BuenasTareas.com, 2012).

### NetBeans 7.0.1

Es un exitoso proyecto de código abierto con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de cien socios en todo el mundo. Fundado por la compañía Sun Microsystems en junio del 2000. Está codificado en el lenguaje de programación Java y permite desarrollar aplicaciones web, de escritorio y móviles, usando las plataformas de Java. Soporta otros lenguajes de programación como son C/C++, Ruby y PHP. Es de código abierto y gratuito para uso tanto comercial como no comercial.

Es multiplataforma, disponible para diversos sistemas operativos como Open Solaris, *Windows*, *MacOS* y *GNU Linux*. Su instalación y actualización es muy simple y una vez instalado se le pueden adicionar módulos que permiten extender sus funcionalidades.

Brinda facilidades para el modelado con UML y posee un diseñador gráfico para juegos y aplicaciones para celulares, empleando la plataforma J2ME (netbeans.org, 2012).

## 1.8 Conclusiones parciales

Después de haber realizado una investigación relacionada con la asignación, se lograron identificar una serie de funcionalidades y características que son comunes en los sistemas de asignación estudiados. De los sistemas analizados a nivel internacional se concluyó que no pueden ser utilizados, pues no se tiene acceso a su documentación, utilizan tecnologías con las que la UCI no cuenta, son sistemas desarrollados con un objetivo específico y con características muy particulares de la institución a la que pertenecen por lo que no pueden ser aplicados en el contexto de la asignación de comensales.

Por otra parte, los sistemas analizados a nivel de universidad, son la base fundamental para el

# *Capítulo I. Asignación de comensales*

---

desarrollo del subsistema de asignación de comensales pues están pensados con el mismo objetivo y los fines son comunes, por lo que se decidió reutilizar muchas de sus funcionalidades. Además se describieron las herramientas y tecnologías propuestas por el centro, las cuales permitirán un correcto desarrollo del subsistema.

## Capítulo II: Propuesta de solución

### Introducción

El proceso de desarrollo de *software* cuenta con varias etapas y en cada una de ellas se generan varios artefactos que son los que le van dando respuesta a las necesidades del cliente.

El presente capítulo aborda las principales características del subsistema, se describe el objeto de estudio en el cual se llevan a cabo los procesos involucrados en el campo de acción y el desarrollo de la propuesta de solución. Además se describe la arquitectura y los patrones utilizados en la implementación y se identifican las funcionalidades con que contará el subsistema.

### 2.1 Modelo de dominio

El Modelo de Dominio es una representación visual estática del entorno real objeto del proyecto. Es decir, un diagrama con los objetos reales que existen en el proyecto que se llevará a cabo y las relaciones que hay entre ellos. Representa una visión estática de las clases y sus interacciones. Su principal objetivo es ayudar a comprender los conceptos que utilizan los usuarios y con los que deberá trabajar el sistema (Larman, 2004).

#### Cómo se elabora:

El proceso para su elaboración tiene tres pasos:

1. Identificar las Clases Conceptuales.
2. Dibujarlas en un Diagrama de Clases.
3. Añadir Relaciones y Atributos.

A continuación se muestra el modelo de dominio perteneciente al Subsistema de asignación de comensales, el cual está compuesto por 7 clases conceptuales (Ver figura 1).

# Capítulo II. Propuesta de solución

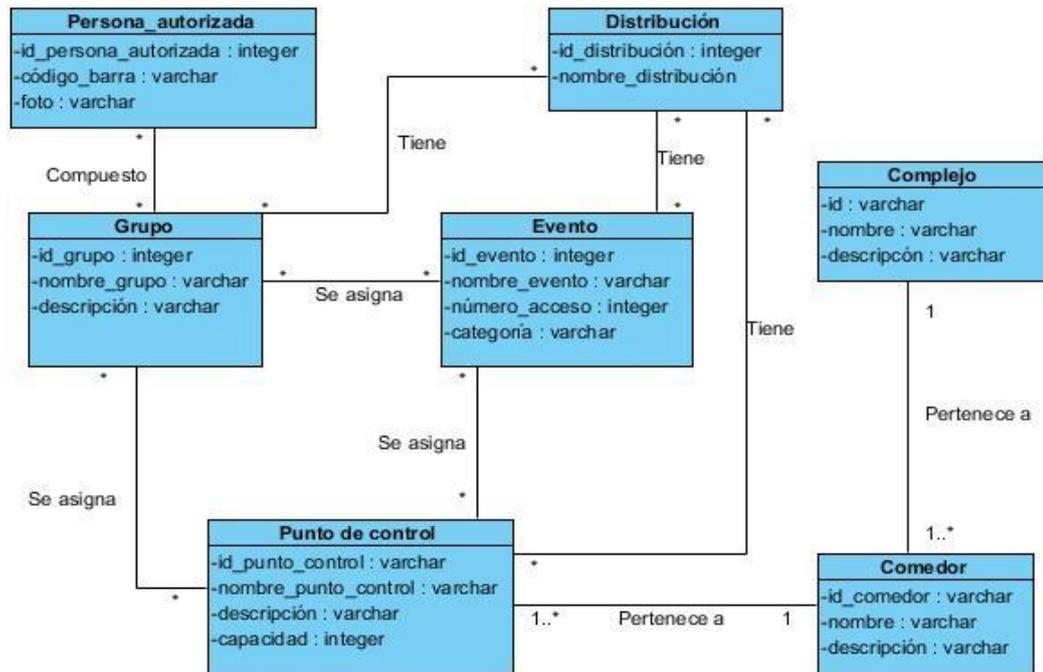


Figura 1. Modelo de dominio.

La clase conceptual **Distribución** representa el resultado de la asignación de los comensales a los puntos de control pertenecientes a los comedores y es una relación entre las personas, los puntos de control y los eventos. Estas personas representadas por el concepto: **Persona autorizada**, se organizan por determinados criterios conformando **Grupos**. Un **Evento** es un concepto que se refiere a una actividad que se realiza en un determinado lugar, por ejemplo el desayuno, el almuerzo y la comida. El acceso a un evento se controla en los **Puntos de control**, donde teniendo en cuenta las reglas definidas en el mismo se permite o se niega la entrada de los comensales. Cada punto de control pertenece a un **Comedor** y cada comedor pertenece a su vez a un **Complejo** de Comedores.

## 2.2 Requisitos funcionales.

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir, sin tomar en consideración ningún tipo de restricción física, de manera que se mantienen invariables sin importar con qué propiedades o cualidades se relacionen (Booch et al, 2004).

Para la posterior implementación del subsistema fueron identificados treinta y ocho requisitos funcionales, los cuales serán explicados a continuación, para más información (Ver anexo 1).

**RF-1** Mostrar eventos: mediante esta funcionalidad la aplicación mostrará los eventos existentes en la misma.

## *Capítulo II. Propuesta de solución*

---

**RF-2** Adicionar evento: la aplicación permitirá adicionar un nuevo evento, editando los campos necesarios para esta acción, como son: Nombre del evento, Categoría del evento, Raciones permitidas, Estado del evento, Hora de Inicio, Hora de Fin.

**RF-3** Modificar evento: la aplicación permitirá modificar los datos de un evento seleccionado.

**RF-4** Eliminar evento: la aplicación permitirá eliminar un evento seleccionado.

**RF-5** Mostrar estructuras y puntos: mediante esta funcionalidad la aplicación muestra todas las estructuras existentes.

**RF-6** Crear estructura y punto de control: la aplicación permitirá adicionar una nueva estructura y al mismo tiempo los puntos de control que le pertenecerán a la misma, editando los campos Nombre y Descripción.

**RF-7** Modificar estructura y punto de control: la aplicación permitirá modificar los datos de una estructura o de un punto de control seleccionado.

**RF-8** Eliminar estructura: la aplicación permitirá eliminar una estructura seleccionada.

**RF-9** Balance de puntos: la aplicación permitirá también balancear la capacidad de acceso a un punto de control seleccionado, editando el campo Capacidad.

**RF-10** Crear grupo: la aplicación permitirá crear un nuevo grupo, editando correctamente los campos Nombre del grupo y Descripción.

**RF-11** Modificar grupo: la aplicación permitirá modificar los datos de un grupo seleccionado.

**RF-12** Eliminar grupo: la aplicación permitirá eliminar un grupo seleccionado.

**RF-13** Eliminar persona y concepto: esta funcionalidad permite eliminar una persona o un concepto seleccionado.

**RF-14** Adicionar personas a grupos: la aplicación permitirá adicionar nuevas personas a un grupo ya existente en la misma.

**RF-15** Gestionar distribuciones: esta funcionalidad posibilita Crear, Modificar y Activar una distribución.

**RF-16** Crear distribución: la aplicación permitirá crear una nueva distribución editando el campo Nombre.

**RF-17** Asignar grupos a puntos: la aplicación permitirá asignar a los puntos de control los grupos anteriormente creados.

## *Capítulo II. Propuesta de solución*

---

- RF- 18** Activar distribución: la aplicación permitirá activar la distribución seleccionada.
- RF- 19** Asignar eventos a puntos: la aplicación permitirá establecer los eventos que tendrán lugar en cada una de las puertas, permitiendo especificar para cada grupo asignado a dicha puerta los eventos particulares a los que estos tendrán acceso.
- RF-20** Crear fichero db4o: la aplicación deberá permitir generar los ficheros db4o que serán usados por la aplicación de Control de acceso.
- RF-21** Crear sede: la aplicación permitirá crear una nueva sede editando los campos necesarios.
- RF-22** Activar sede: la aplicación permitirá activar una sede seleccionada.
- RF-23** Eliminar sede: la aplicación permitirá eliminar la sede seleccionada.
- RF-24** Crear usuario: la aplicación permitirá agregar nuevos usuarios mediante esta funcionalidad.
- RF-25** Modificar usuario: la aplicación brindará la opción de modificar los datos pertenecientes a un usuario ya existente en la misma, antes debe ser seleccionado.
- RF-26** Cambiar contraseña: la aplicación posibilitará también cambiar la contraseña perteneciente a un usuario seleccionado.
- RF-27** Eliminar usuario: la aplicación brindará la opción de eliminar un usuario seleccionado.
- RF-28** Guardar dirección de fotos: la aplicación permitirá crear y guardar la dirección del servidor de fotos, la cual será utilizada por el subsistema de Control de acceso para obtener las fotos de las personas.
- RF-29** Modificar dirección de fotos: la aplicación permitirá modificar y guardar la dirección del servidor de fotos, la cual será utilizada por el subsistema de Control de acceso para obtener las fotos de las personas.
- RF-30** Eliminar dirección de fotos: la aplicación permitirá eliminar las direcciones de servidores de fotos que no sean necesarias.
- RF-31** Adicionar atributo de persona: la aplicación permitirá adicionar un nuevo atributo de persona editando los campos Nombre, Texto a mostrar y Descripción.
- RF-32** Modificar atributo de persona: la aplicación permitirá modificar un atributo de persona seleccionado.
- RF- 33** Eliminar atributo de persona: la aplicación permitirá eliminar un atributo de persona

## Capítulo II. Propuesta de solución

---

seleccionado.

**RF-34** Reporte del punto de control: la aplicación permitirá conocer todo lo relacionado a los puntos de control existentes, brindando también las opciones de exportar en Excel y PDF dicho reporte.

**RF-35** Reporte de asignado a: la aplicación brinda la posibilidad de conocer dado un número de expediente a quien corresponde el mismo; así como el complejo, comedor y puerta a la cual está asignada dicha persona.

**RF-36** Reporte sin asignar: mediante este reporte la aplicación permite conocer el total de las personas que no han sido agrupadas, es decir que no se han asignado a ningún grupo.

**RF-37** Reporte de accesos de una persona: mediante esta funcionalidad la aplicación permite conocer las personas que pasan por el punto de control que el usuario seleccione, y los accesos de cada una de estas personas a este punto de control en un día determinado.

**RF-38** Reporte accesos a un punto de control: permite conocer todas las personas que han accedido a un punto de control, en un evento y una fecha determinada.

### 2.3 Requisitos no funcionales.

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener. Representan las características que hacen al producto atractivo, usable, rápido o confiable. Son fundamentales en el éxito del producto y normalmente están vinculados a requisitos funcionales (Booch et al, 2004).

#### Usabilidad

- ✓ Facilidad de uso: el subsistema debe presentar una interfaz amigable al usuario, permitiendo fácil interacción con el mismo y llegar de manera rápida y efectiva a la información buscada. Además esta interfaz debe ser de manejo cómodo donde la curva de aprendizaje para los usuarios sea lo menos inclinada posible y que posibilite en estos una rápida adaptación.
- ✓ Especificación de la terminología utilizada: el subsistema debe adaptarse al lenguaje y términos utilizados por los clientes en la rama abordada con vista a una mayor comprensión por su parte sobre la herramienta de trabajo.
- ✓ Menús: el subsistema debe presentar una serie de menús, que permitan el acceso rápido a la información por parte de los usuarios, aprovechando así las potencialidades de estas estructuras.

# Capítulo II. Propuesta de solución

---

## **Escalabilidad**

- ✓ El subsistema debe estar en capacidad de permitir en el futuro el desarrollo de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y puesta en marcha inicial.

## **Soporte**

- ✓ Grupo de soporte y asesoría: el subsistema contará con un grupo de soporte y asesoría al cliente.

## **Interfaz**

- ✓ Interfaz externa: la interfaz deberá ser sencilla con colores suaves a la vista y sin cúmulo de imágenes u objetos que distraigan al cliente del objetivo de su empleo.

## **Fiabilidad**

- ✓ Disponibilidad: El subsistema estará disponible las veinticuatro horas del día y los siete días de la semana.
- ✓ Todas las salidas del subsistema deberán tener un cien por ciento de veracidad y precisión.

## **Portabilidad.**

- ✓ El subsistema deberá funcionar tanto en *Windows* como *Linux*.

## **Seguridad**

- ✓ El subsistema garantizará que la información sea modificada y vista únicamente por quien tenga permisos para esto.
- ✓ El acceso al subsistema deberá estar restringido por el uso de claves asignadas a cada uno de los usuarios. Sólo podrán ingresar al subsistema las personas que estén registradas.
- ✓ Respecto a la confidencialidad, el subsistema deberá estar en capacidad de rechazar accesos o modificaciones indebidos (no autorizados) a la información y proveer los servicios requeridos por los usuarios legítimos del mismo.

## **Restricciones de diseño**

- ✓ El subsistema de desarrollará en lenguaje *Java*.
- ✓ El marco de trabajo base de desarrollo que se utilizará es: *Spring framework*.
- ✓ El IDE de desarrollo y mantenimiento del subsistema es *NetBeans*.
- ✓ El sistema gestor de bases de datos deberá ser *PostgreSQL*.
- ✓ El sistema operativo a utilizar en el entorno de desarrollo deberá ser: *GNU Linux*.

## **Hardware**

- ✓ Para el desarrollo: PC Pentium 4 o superior, CPU 1.60GHZ o superior, 1024 MB RAM o superior.
- ✓ Para explotación del cliente: PC Pentium 4 o superior, CPU 1.60 GHZ o superior, 768 MB RAM recomendada o superior.

# Capítulo II. Propuesta de solución

## Software

- ✓ Debe estar instalada la Máquina Virtual JAVA 7.0 o superior.

### 2.4 Métrica para la calidad de la especificación de los requisitos de software.

Actualmente en el proceso de desarrollo de *software* existen un conjunto de métricas que se utilizan para la validación de los requisitos identificados en la realización de un *software*, estas métricas permiten a los analistas validar de manera correcta que los requisitos identificados durante el proceso de desarrollo tienen la calidad requerida.

Los requisitos del Subsistema de asignación de comensales fueron comprobados para determinar su especificidad (ausencia de ambigüedad) a través de la métrica para la calidad de especificación de los requisitos de *software*, basada en la consistencia de la interpretación de los revisores para cada uno de estos. La métrica es el resultado de dividir el número de requisitos para los que todos los revisores tuvieron interpretaciones idénticas (**Nui**) entre la cantidad de requisitos de *software* (**Rt**).

$$Q1 = Nui/Rt, Rt = Rf + Fnf$$

Donde **Rf** es la cantidad de requisitos funcionales y **Fnf** es la cantidad de requisitos no funcionales.

Cuanto más cerca de uno esté el valor de **Q1** menor será la ambigüedad de la especificación

(Pressman, 2002).

$$Rt = 38 + 20, Rt = 58$$

$$Q1 = 55 / 58, Q1 = 0.94$$

Los miembros del equipo de inspección se presentan a continuación:

Tabla 1. Miembros del equipo de inspección.

Nombre y apellidos	Rol desempeñado
Iviannis Cañizares Rivera	Jefe de proyecto
Sergio René Vázquez	Jefe de línea
Anisley Martínez Romero	Analista

A continuación se muestran los resultados obtenidos al aplicar la métrica:

## Capítulo II. Propuesta de solución

Tabla 2. Resultados obtenidos al aplicar la métrica.

Atributo de calidad	Tipo de requisito	Interpretaciones	
		Iguales	Desiguales
Especificidad	Funcionales: 38	36	2
	No funcionales: 20	19	1
	Total: 58	55	3

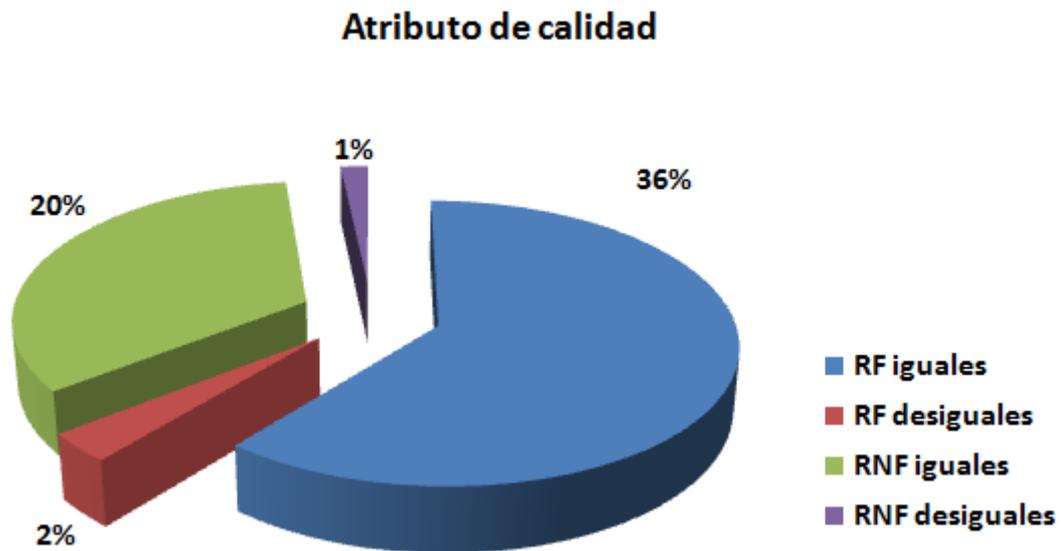


Figura 2. Resultados de la métrica para la calidad de la especificación de los requisitos de *software*.

El valor de  $Q1 = 0.94$ , resultado que demuestra que los requisitos del subsistema de asignación de comensales se encuentran con un alto nivel de especificidad, es decir que la ambigüedad de los requisitos es baja. Es importante destacar que un requisito es ambiguo si tiene múltiples interpretaciones.

### 2.5 Diccionario de datos.

A continuación se describen las entidades que persisten en el subsistema.

Tabla 3. Entidad: Acceso.

Nombre de la entidad	Acceso				
Descripción de la entidad	En esta entidad se registran los accesos a los diferentes puntos de control durante los eventos.				
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones	Criterio de selección

## Capítulo II. Propuesta de solución

				Clases válidas	Clases no válidas	Múltiple	Única
id_punto	Identificador del punto al que se va a tener acceso.	Texto	No	0112	Vacío Caracteres Letras	-	SI
id_evento	Identificador del evento al que se va a tener acceso.	Entero	No	2	Vacío Caracteres Letras	-	SI
id_persona	Identificador de la persona que va a tener dicho acceso.	Texto	No	43789	Vacío Caracteres Letras	-	SI
fecha	Fecha en la que se registró el acceso de una persona.	Texto	No	04/02/12	-	-	SI

Tabla 4. Entidad: Punto de control.

Nombre de la entidad		Punto de control					
Descripción de la entidad		Esta entidad almacena los datos de los puntos de control de los comedores.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_punto_control	Identificador del punto de control.	Texto	No	0122	-	-	Si
nombre_punto_control	Nombre del punto de control.	Texto	No	Puerta 322	Vacío	-	Si
descripcion	Descripción del punto de control.	Texto	Si	Esta es la puerta número 2 que pertenece al comedor 2 del complejo 3.		-	Si
activo	Indica si el evento está activo.	Booleano	No	Verdadero/ Falso	-	-	Si
capacidad	Número máximo de personas que deben tener acceso al punto de control.	Entero	No	600	-	-	Si
ip	Dirección ip de la PC del punto de control.	Texto	Si	10.8.15.12	-	-	Si

## Capítulo II. Propuesta de solución

Tabla 5. Entidad: Grupo.

Nombre de la entidad		Grupo						
Descripción de la entidad		Esta entidad almacena los grupos creados en el subsistema para organizar las personas.						
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección		
				Clases válidas	Clases no válidas	Múltiple	Única	
id_grupo	Identificador del grupo.	Texto	No	01123			Si	
nombre_grupo	Nombre del grupo.	Texto	No	Facultad 1	Vacío	-	SI	
descripcion	Descripción del grupo.	Texto	Si	En este grupo se encuentran todos los estudiantes que pertenecen a la facultad 1.	Vacío	-	SI	
id_grupo_padre	Identificador del grupo padre.	Texto	No	0112	Vacío		Si	
asignado	Indica si el grupo fue asignado al punto de control.	Booleano	No	Verdadero/Falso	Vacío		Si	

Tabla 6. Entidad: Evento.

Nombre de la entidad		Evento						
Descripción de la entidad		Esta entidad contiene los eventos que se corresponden a los comedores, especificando las características propias de cada uno de ellos.						
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección		
				Clases válidas	Clases no válidas	Múltiple	Única	
nombre_evento	Nombre que identifica al evento.	Texto	No	Comida	Vacío/caracteres	-	Si	
numero_acceso	Cantidad de veces que un comensal puede pasar por un punto de control en el evento.	Entero	No	1	Vacío/caracteres	-	Si	
categoria	Se clasifica el evento.	Texto	No	Regular	-	-	Si	
hora_inicio	Indica la hora de comienzo del evento.	Texto	No	06:30	-	-	Si	
hora_fin	Hora en la que termina el evento.	Texto	No	10:00	-	-	Si	
activo	Indica si el evento está activo.	Booleano	No	Verdadero/Falso	-	-	Si	

## Capítulo II. Propuesta de solución

Tabla 7. Entidad: Usuario.

Nombre de la entidad		Usuario					
Descripción de la entidad		En esta entidad se guardan los datos de cada usuario, teniendo en cuenta la aplicación a la que pertenecen y el rol que desempeñan en la misma.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
usuario	Usuario del sistema.	Texto	No	dcobiella	Vacío		Si
pnombre	Primer nombre del usuario.	Texto	No	Dayami	Vacío/Números	-	Si
snombre	Segundo nombre del usuario.	Texto	Si	-	Vacío/Números	-	Si
papellido	Primer apellido del usuario.	Texto	No	Alvarez	Vacío/Números	-	Si
sapellido	Segundo apellido del usuario.	Texto	No	Cobiella	Vacío/Números	-	Si
contrasenna	Contraseña de acceso del usuario.	Texto	No	Dayi123**	Vacío	-	Si
activo	Indica si el usuario está activo.	Booleano	No	Verdadero/Falso	-	-	Si
id_aplicacion	Identificador de la aplicación a la que pertenece el usuario.	Entero	No	2	Vacío	-	Si

Tabla 8. Entidad: Distribución.

Nombre de la entidad		Distribución					
Descripción de la entidad		Esta entidad registra las distribuciones que se configuran en el subsistema.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_distribucion	Identificador de la distribución.	Entero	No	2			Si
nombre_distribucion	Nombre de la distribución.	Texto	No	Distribución actual	Vacío	-	-
activa	Indica el estado de la distribución (Solo puede haber una distribución activa).	Booleano	No	Verdadero/Falso	-	-	Si

## Capítulo II. Propuesta de solución

Tabla 9. Entidad: Persona autorizada.

Nombre de la entidad		Persona autorizada					
Descripción de la entidad		Esta entidad almacena los datos de las personas existentes en la institución con derecho a los eventos.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_persona_autorizada	Identificador de la persona.	Texto	No	100127	Vacío	-	Si
activa	Indica si una persona está activa en la institución.	Booleano	No	Verdadero/Falso	Nulo	-	Si
codigo_barra	Código de barra asignado a una persona.	Texto	No	H77SI2H	Vacío	-	Si
foto	Identificador de la foto de la persona.	Texto	No	9201f47d-b3b2-ada	Vacío	-	Si
asignada	Indica si una persona está asignada a un grupo.	Booleano	No	Verdadero/Falso	Vacío	-	Si

Tabla 10. Entidad: Recurso foto.

Nombre de la entidad		Recurso de foto					
Descripción de la entidad		Esta la entidad registra la dirección del servidor de fotos.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
direccion_lectura	Dirección del servidor de fotos.	Texto	No	-		-	SI

Tabla 11. Entidad: Estructura.

Nombre de la entidad		Estructura					
Descripción de la entidad		En esta entidad se guardan las estructuras que son las edificaciones que componen la institución. Ejemplo: Comedor 1.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_estructura	Identificador de la estructura	Texto	No	01121	Vacío		Si
nombre_estructura	Nombre de la estructura.	Texto	No	Complejo 1	Vacío	-	Si

## Capítulo II. Propuesta de solución

descripcion	Información que describe la estructura.	Texto	No	Este es la estructura del complejo número1.	Vacío	-	Si
activo	Se activa si desea que la estructura se ponga en función, sino no se activa.	Boolea no	No	Verdadero o Falso	Vacío	No	Si

Tabla 12. Entidad: Reservación.

<b>Nombre de la entidad</b>		Reservación					
<b>Descripción de la entidad</b>		Es la entidad donde se recogen todos los datos para realizar una reservación.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_reservacion	Identificador de la reservación.	Entero	No	1	Vacío		Si
activo	Indica si la reservación está activa.	Boolea no	No	Verdadero o Falso	Vacío		Si
fecha	Fecha de la reservación.	Date	No	06/02/12	Vacío		Si
solapin	Identificador del expediente de la persona que realizó la reservación.	Fecha	No	7/02/12	Vacío		Si

Tabla 13. Entidad: Sede.

<b>Nombre de la entidad</b>		Sede					
<b>Descripción de la entidad</b>		En esta entidad registra la sede donde se implantará el subsistema.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_sede	Identificador de la sede.	Entero	No	1	Vacío		Si
nombre_sede	Nombre de la sede.	Texto	No	UCI	Vacío	-	Si
activa	Indica si la sede está activa.	Booleano	No	Verdadero/ Falso	Vacío	No	Si

## Capítulo II. Propuesta de solución

Tabla 14. Entidad: Área.

Nombre de la entidad		Área					
Descripción de la entidad		Es donde se encuentran todas las áreas que pertenecen a la institución donde se implanta el subsistema y a las cuales pertenecen todas las personas.					
Nombre del atributo	Descripción	Tipo	Puede ser nulo	Restricciones		Criterio de Selección	
				Clases válidas	Clases no válidas	Múltiple	Única
id_area	Identificador del área.	Texto	No	3	Vacío	-	SI
nombre_area	Nombre del área	Texto	No	Docente 3	Vacío Caracteres	-	SI

### 2.6 Modelo lógico y físico.

Una base de datos es una colección de información organizada y almacenada en forma sistemática, de manera tal que los datos que la conforman puedan ser seleccionados rápidamente en forma fragmentada cuando sea necesario. Es posible considerarla como una especie de armario electrónico para archivar.

El modelo de datos del Subsistema de asignación de comensales fue modelado con la herramienta Visual Paradigm 8.0, creando primeramente el modelo lógico (Ver figura 3) y posteriormente el modelo físico (Ver figura 4), donde se muestran las diferentes tablas con sus atributos y sus respectivas relaciones. Este modelo de datos cuenta con diecinueve tablas; doce de datos, una nomencladora y seis de relación. La nomenclatura de dichas entidades es de la siguiente forma: para las tablas de datos tbd\_<nombre>, para las nomencladoras tbn\_<nombre> y para las que almacenan la relaciones entre entidades tbr\_<nombre1>\_<nombre2>.

# Capítulo II. Propuesta de solución

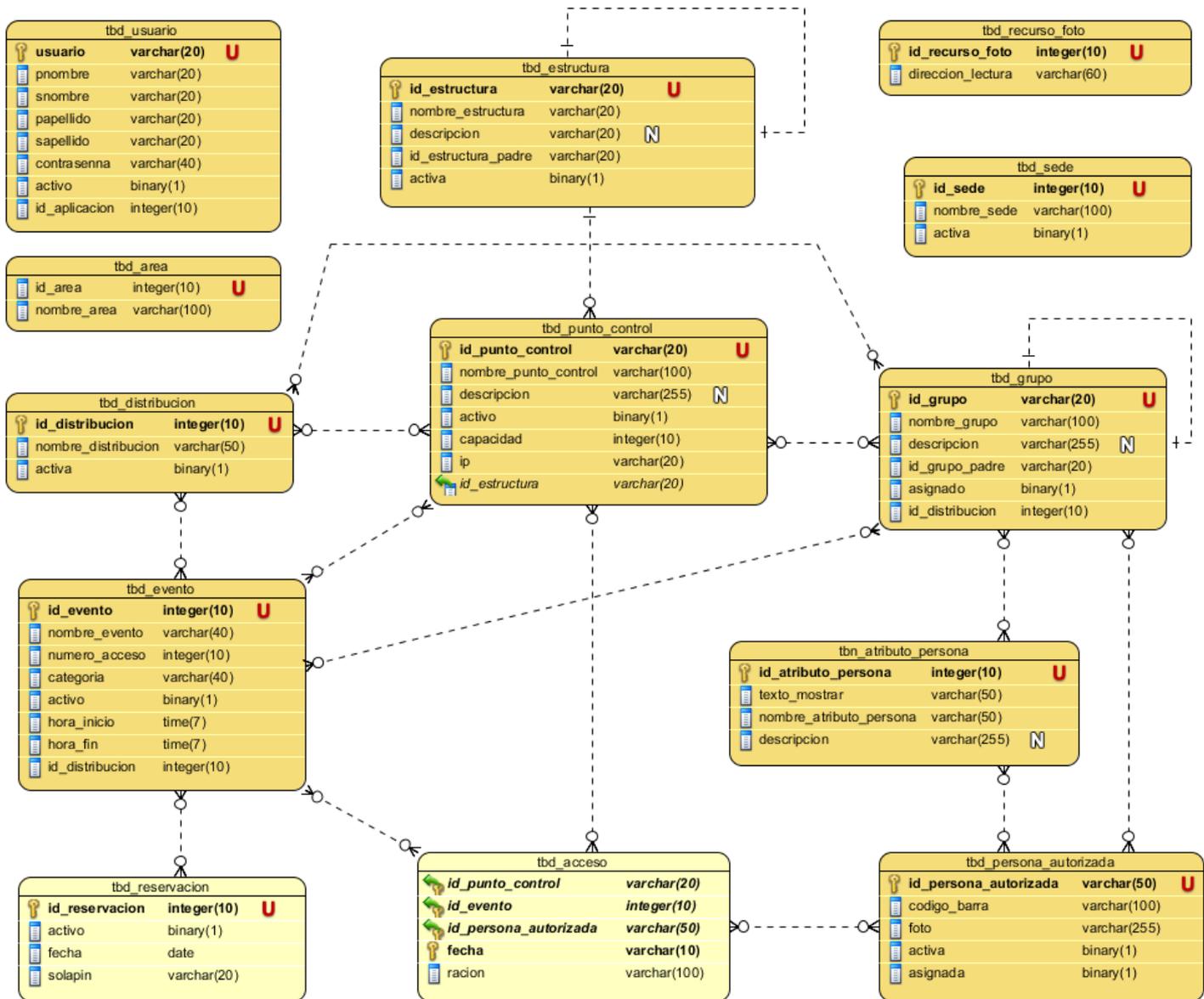


Figura 3. Modelo lógico del Subsistema de asignación de comensales.



### 2.7 Patrones de diseño.

Los patrones son principios generales basados en la experiencia que aplican ciertos conocimientos y estilos que guían la creación de un *software*. Cada patrón especifica un problema recurrente en el diseño y la implementación de un determinado programa.

#### **Patrones para asignar responsabilidades (GRASP).**

Estos patrones constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objetos. A continuación se mencionan los utilizados:

- ✓ **Bajo acoplamiento:** uno de los principios para proteger al *software* frente al cambio es mantener bajo el acoplamiento entre clases. El acoplamiento de una clase es el conjunto de dependencias que tiene con otras clases, cuanto menor sea el acoplamiento entre clases, menor influencia tendrán los cambios. En la solución que se ofrece, cada clase se relaciona solo con quien lo necesita para realizar sus procedimientos (o métodos).
- ✓ **Alta cohesión:** al asignar responsabilidades en el diseño, se buscan soluciones que asignen los métodos a las clases de forma coherente, completa y relacionada. De esta forma, se obtienen clases cohesionadas. Las ventajas son evidentes. Una clase cohesionada facilita el cambio, al realizar un cambio en una clase muy cohesionada, todos los métodos que pueden verse afectados, toda la información que se necesita controlar, estará a la vista, en el mismo fichero. Este patrón se relaciona con el de bajo acoplamiento, porque un diseño cohesionado tendrá un bajo acoplamiento entre clases. Ejemplos de este patrón se pueden encontrar en todas las clases de las capas de negocio y acceso a datos, donde cada clase realiza los métodos que le competen, según su concepción y finalidad.
- ✓ **Experto:** se tiene en consideración a qué clase debe pertenecer un método, este principio sugiere que se asigne a la clase que más sepa del método (es decir al experto). Esto es una consecuencia del principio de alta cohesión, ya que si se asignan los métodos a las clases que tienen la información necesaria para ejecutarlos, se están creando clases altamente cohesionadas. Al igual que el patrón anterior, un ejemplo de la utilización de este patrón se puede encontrar en la clase **PersonaDao** la cual se encarga de obtener específicamente los datos referentes a las personas que se manejan en el subsistema.
- ✓ **Creador:** permite decidir cuáles serán las clases creadoras de otras clases. Este patrón se tiene muy en cuenta a la hora de estructurar las clases según la arquitectura, donde cada clase de una capa superior crea su similar en la inferior (en conjunto con el marco de trabajo, ya que *Spring* es quien maneja los objetos de negocio) (Larman, 2004).

## Capítulo II. Propuesta de solución

---

Así mismo se tiene de ejemplo que la clase **GrupoNeg** es quien concibe y en conjunto con el marco de trabajo crea la clase **GrupoDao**, la cual utilizará para sus operaciones.

### **Patrones GOF**

#### **Patrones de creación:**

✓ **Singleton:** garantiza que una clase sólo tenga una instancia, y proporciona un punto de acceso global a ella. Esto se logra haciendo que sea la propia clase la responsable de controlar la existencia de una única instancia. Este patrón se utiliza en la clase **FachadaDao** la cual se utiliza en la creación del fichero DB4O.

✓ **Factory:** este patrón consiste en utilizar una clase constructora con unos cuantos métodos dedicados a la construcción de objetos de un tipo determinado. Su utilización puede verse en la clase **Loader** que es la encargada de implementar estos métodos de construcción utilizados para crear los objetos.

### **2.8 Arquitectura**

Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de *Software* o Arquitectura lógica. Se refiere a un grupo de abstracciones y patrones que brindan un esquema de referencia útil para guiar el desarrollo de *software* dentro de un sistema informático. Así, los programadores, diseñadores, ingenieros y analistas pueden trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo deseado (Booch et al, 2004).

El sistema está basado en una arquitectura por capas, la cual es una especificación de la arquitectura cliente-servidor. Esta arquitectura brinda un estilo de programación donde el objetivo primordial es la separación de la lógica de negocio, de la lógica de diseño. En el caso específico del subsistema desarrollado, la arquitectura está compuesta por tres capas (Ver figura 5): capa de presentación, capa de negocio y capa de acceso a datos. El uso de *Spring* aporta el patrón de inyección de dependencia con el cual se delega la responsabilidad de instanciar los objetos en archivos de configuración XML. Esto resulta muy útil puesto que evita cualquier dependencia que pueda existir entre los objetos, garantizando la comunicación entre las capas horizontales y también entre los módulos y permitiendo el desacoplamiento entre las capas. A continuación se detallan cada una de las capas y los componentes que conforman la arquitectura:

## Capítulo II. Propuesta de solución

---

### ➤ **Capa de presentación**

La capa de presentación es la que ve el usuario (también se la denomina "capa de usuario"), presenta el sistema al usuario, le comunica la información y captura la información que este introduce en 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.

#### **Formularios de interfaz de usuario**

Todos los módulos tienen un diseño y comportamiento estándar con vistas a facilitar el trabajo con ellos y la estandarización del subsistema completo. Su uso se basa en la explotación de los formularios de interfaz de usuario que provee el IDE facilitando el desarrollo del subsistema.

#### **Gestión de formularios de interfaz de usuario**

Se basa en acciones contempladas en la gestión de formularios de interfaz de usuario donde cada acción se corresponde con funcionalidades de captura o visualización de los datos que tiene asociado un formulario, cuya forma depende de la funcionalidad específica para la cual fue concebida dicha acción.

### ➤ **Capa lógica de negocio**

El diseño de esta capa depende directamente del negocio específico al que se refiera cada módulo. El negocio recibe datos y/o información capturada en las interfaces de usuario, gestiona o procesa la misma, de ser necesario solicitándola a la capa de acceso a datos para finalmente enviársela a la presentación nuevamente para que esta la muestre al usuario en el punto donde se inició la petición.

#### **Interfaces de negocio**

El pilar fundamental de la arquitectura radica en el uso de las interfaces como recurso que se utiliza para brindar funcionalidades que representan un nivel de abstracción. Este nivel es precisamente el que asegura el patrón bajo acoplamiento conjuntamente con otros elementos.

#### **Lógica de negocio**

Componentes encargados de implementar las funcionalidades descritas en las interfaces de negocio, definiendo las reglas de negocio específicas de cada acción.

### ➤ **Capa acceso a datos**

Definir la estrategia de persistencia de una aplicación es una de las decisiones de arquitectura más importantes. En una aplicación estándar más del 50% del código generado está relacionado con lógica de persistencia (Ebersole, 2010).

Por tanto, el acceso a datos es la capa más crítica y sensible a cambios de la arquitectura, pues controla todo lo concerniente a la información que se encuentra en la fuente de almacenamiento (Origen de datos).

# Capítulo II. Propuesta de solución

Al ser la capa inferior no conoce los niveles superiores, únicamente se limita al manejo de la información, ya sea para persistirla o proporcionarla para su procesamiento y propagación por la aplicación. Todo este manejo es responsabilidad de los Objetos de Acceso a Datos (DAOs por sus siglas en inglés).

## Interfaces de acceso a datos

Representan las funcionalidades que brinda este nivel, es decir las referidas a los DAOs. Su uso y funcionamiento es el mismo que las de la capa lógica de negocio.

## Lógica de acceso a datos

Todas las funcionalidades del acceso a datos radican en los DAOs, es decir este ensamblado implementa la totalidad de las operaciones de persistencia y obtención de datos explotando los recursos que brinda *Hibernate*, que cumple perfectamente con el objetivo de este nivel, dígase trabajo con procedimientos almacenados y métodos de persistencia o consultas.

## Origen de datos

Corresponde a la fuente de datos del subsistema y es común a todos los módulos del subsistema en general.

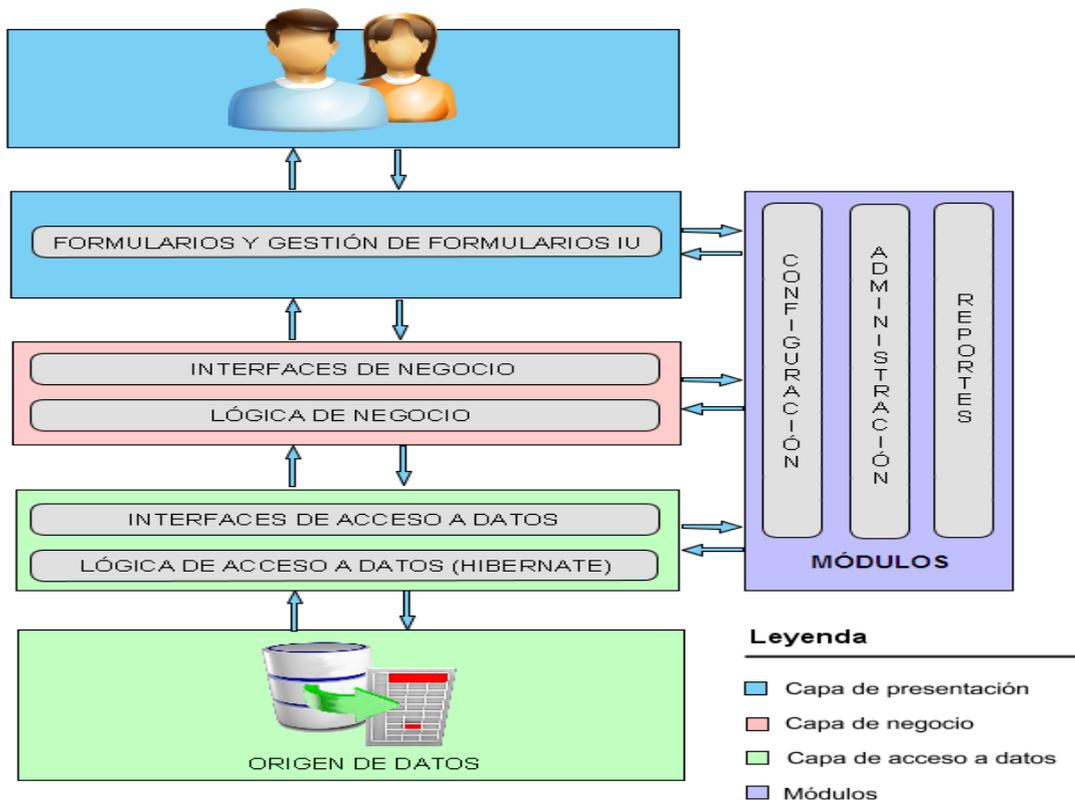


Figura 5. Arquitectura del Subsistema de asignación de comensales.

## Capítulo II. Propuesta de solución

---

### 2.9 Descripción del subsistema

Como propuesta de solución se obtuvo una aplicación de escritorio, funcional tanto en Linux como en *Windows*, desarrollada totalmente con herramientas libres y que cumple con las políticas del centro CENIA en el cual se desarrolló. La misma cuenta con treinta y ocho funcionalidades distribuidas en tres módulos: **Configuración**, **Administración** y **Reportes**.

En el módulo **Configuración** se definen las características y elementos que permiten el correcto funcionamiento de la aplicación. Esta configuración contempla la gestión de usuarios, la gestión de la sede, los atributos referentes a las personas con que opera el subsistema y la dirección del servidor de fotos de la sede o institución donde se encuentre funcionando el subsistema.

El módulo **Administración** es el módulo fundamental del subsistema y en él se encuentran las funcionalidades relacionadas con la asignación de comensales. Las principales agrupaciones funcionales con que cuenta son:

- ✓ Gestión de eventos: se gestionan los eventos (Desayuno, Almuerzo, Comida, entre otros) que se efectúan en la universidad.
- ✓ Estructuras y puntos: se diseña en forma de árbol la estructura de los comedores de la universidad así como los puntos de control que pertenecen a las mismas.
- ✓ Grupos y personas: Se crean grupos para organizar a las personas según criterios definidos por el administrador de la aplicación.
- ✓ Asignar grupos a puntos de control: los grupos anteriormente creados y que contienen a un conjunto de personas, son asignados a los puntos de control por donde dichas personas tendrán derecho a consumir los platos que se oferten durante un determinado evento.
- ✓ Asignar eventos a puntos: se establecen los eventos que tendrán lugar en cada una de las puertas, permitiendo especificar para cada grupo asignado a dicha puerta los eventos particulares a los que estos tendrán acceso.

En el módulo **Reportes** se obtiene la información y los datos que se manejan en el subsistema, como por ejemplo los puntos de control que están activos en un momento dado en la universidad, la cantidad de personas que pasaron por un punto de control en un evento y un día determinado, saber si una persona en específico pasó por un punto de control y a qué punto de control está asignada una persona. El módulo también brinda la posibilidad de exportar la información mostrada en cada reporte en formato .xls y .pdf para un mejor análisis de los resultados.

Con el fin de mejorar la gestión de las personas en el subsistema, se integra a la base de datos centralizada del Sistema de Gestión Universitaria, donde se encuentran todos los trabajadores y estudiantes de la institución. Con esta conexión se elimina el antiguo concepto de los servicios web

## Capítulo II. Propuesta de solución

usado en la aplicación que está actualmente en explotación. Esto permitió un aumento en la velocidad de ejecución y actualización de los datos.

Se obtuvo un subsistema con una interfaz de usuario sencilla y fácil de utilizar con estructura semejante a la anterior para facilitar el trabajo de los usuarios que ya estaban familiarizados con la misma. Las funcionalidades de asignación están enfocadas a una correcta organización de las personas, restringiendo el agrupamiento de las mismas de forma tal que una persona solamente pueda pertenecer a un grupo. Este nuevo concepto posibilita conocer exactamente las personas asignadas y sin asignar en todo momento y trae como resultado una mejor distribución final, pues no se da el caso de que un comensal tenga acceso a más de un punto de control durante un mismo evento.

El Subsistema de asignación de comensales se integrará también al Sistema de Planificación y Control de la Alimentación existente en la universidad. Este sistema está compuesto además por el Subsistema de control de acceso, el cual se utiliza en las puertas de los comedores para restringir y controlar el acceso de los comensales y el Subsistema de reservación de alimentación y planificación de menú, el cual brinda la posibilidad a los comensales de la universidad de reservar y cancelar el servicio de alimentación para los distintos eventos.

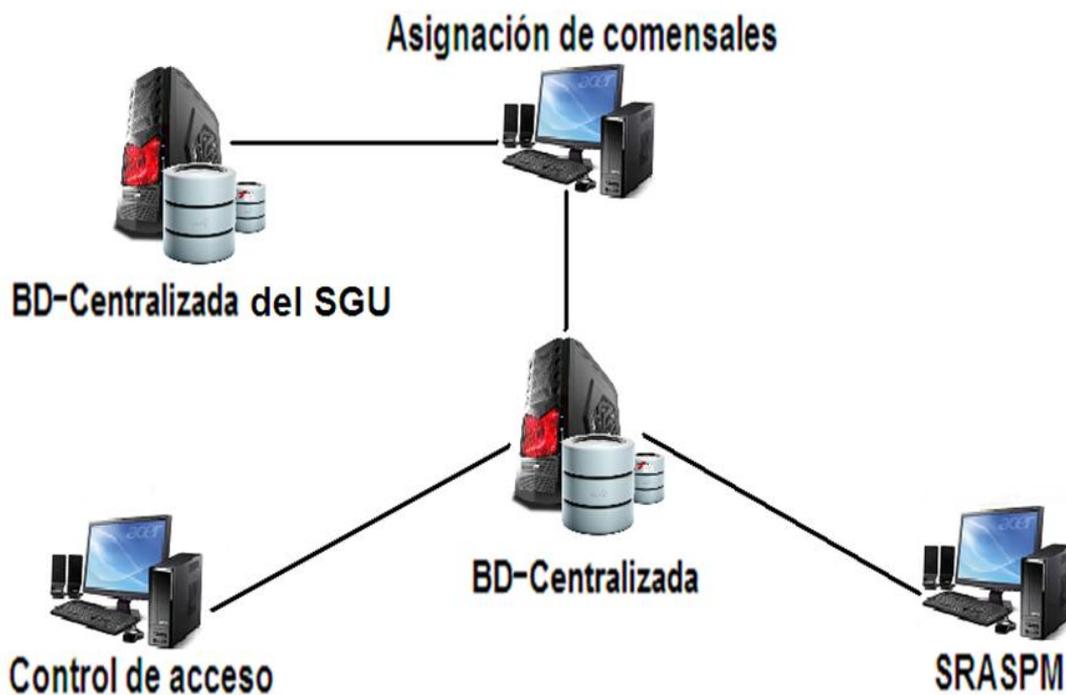


Figura 6. Integración del Subsistema de asignación de comensales.

## *Capítulo II. Propuesta de solución*

---

El subsistema propuesto podrá ser utilizado en la Dirección de Alimentación de la universidad, y los usuarios del mismo realizarán los procesos de asignación en menos tiempo, permitiendo realizar cambios de última hora con un tiempo de ejecución menor. Los datos de los comensales al ser actualizados de una fuente de datos centralizada, serán más confiables y estarán sujetos a menos problemas de falta de integridad. La mejor gestión de personas, impedirá la asignación de las mismas a más de un punto de control durante un evento y facilitará el trabajo de los usuarios de la aplicación, los cuales tendrán la información exacta del personal asignado y sin asignar en todo momento. Así mismo, la posibilidad de exportar reportes facilitará un mejor estudio de los procesos de asignación y acceso del personal a las puertas y permitirá contar con un registro de los datos más importantes que se manejan en el subsistema.

### **2.10 Conclusiones parciales.**

En este capítulo se desarrolló la propuesta de solución del subsistema atendiendo a las características que debe tener el mismo. Se describieron las características del subsistema para un mejor entendimiento de cómo está formado. Se describió la arquitectura utilizada, así como los patrones de diseño empleados durante todo el desarrollo del subsistema de asignación de comensales. Se realizó también el modelo de dominio y la especificación de los requisitos funcionales y no funcionales. Se describió además el diseño de la base de datos mediante el modelo lógico y físico. Cada uno de estos elementos contribuyó a una correcta implementación del subsistema.

## Capítulo III: Validación de la solución.

### Introducción

En un proyecto de desarrollo de *software*, los errores pueden presentarse en cualquiera de las etapas del ciclo de vida aunque se intente detectarlos después de cada fase, utilizando técnicas como la inspección, algunos errores permanecen sin ser descubiertos. Con el objetivo de reducir al máximo la aparición de estos errores, así como la satisfacción de los requisitos, se desarrollan las pruebas de *software*.

Este capítulo se enfoca principalmente en la descripción de las pruebas realizadas al subsistema y en el diseño de los casos de prueba para validar la calidad del mismo y se definen los estándares de código a tener en cuenta en toda la implementación del subsistema.

### 3.1 Estándares de codificación

En el caso de las declaraciones, para declarar una clase se escribió el nombre de la clase con la primera letra en mayúscula y el resto del nombre en minúscula. Los métodos de una clase tienen una forma encamellada. Las variables se declararon explícitamente sin usar abreviaturas y las constantes se declararon en mayúscula.

Los comentarios se definen comenzando con los caracteres `/*` y terminando con `*/`, siempre que se quiera comentar grandes instrucciones de códigos. Si solo se desea comentar una sola instrucción se utiliza `//`. Cuando se implementa un método se pone un comentario en el encabezado que permita visualizar al mismo de una forma más rápida.

La indentación es a cuatro espacios en blanco.

Entre los operadores aritméticos y sus operandos así como entre los operadores lógicos aritméticos y sus operandos se dejó un espacio en blanco ejemplo:

```
if (personas.get(i).getIdPersona == 15 &&personas.get(i).getGrupo.equals( "15510"))
```

Las llaves para abrir y cerrar una clase, un método o un bloque de control de flujo se indentan al mismo nivel del bloque al que pertenecen, ejemplo:

```
if (personas.get(i).getIdPersona == 15 &&personas.get(i).getGrupo.equals( "15510"))  
{  
    return true;
```

}

## 3.2 Diagrama despliegue

El diagrama de despliegue es utilizado para capturar los elementos de configuración del procesamiento y las conexiones entre esos elementos. Describen la arquitectura física del sistema durante la ejecución, en términos de procesadores, dispositivos y componentes de *software*.

### Componentes del diagrama de despliegue

- **Nodos:** elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos (impresoras, lectores de códigos de barras o dispositivos de comunicación).
- **Dispositivos:** nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela. Generalmente se usan para identificar el tipo de dispositivo. Los nodos se conectan mediante asociaciones de comunicación que indican:
  - Algún tipo de ruta de comunicación entre los nodos.
  - Los nodos intercambian objetos o envían mensajes a través de esta ruta.
- **Conectores:** expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.



Figura 7. Diagrama de despliegue.

- **PC\_Administración:** computadora en la cual se utilizará la aplicación. Debe tener instalada la Máquina Virtual JAVA 7.0.

- **BD\_Centralizada:** representa el servidor donde estará el SGBD *PostgreSQL* que dará respuesta a las peticiones hechas por la aplicación y donde se almacenarán los datos actualizados de las personas así como las configuraciones de la asignación.
- **BD\_Sistema de Gestión Universitaria:** representa el servidor de base de datos centralizado al cual se integrará el subsistema para actualizar los datos de las personas.

### 3.3 Pruebas

Deben realizarse pruebas a todos los artefactos generados durante la construcción de un producto, el código fuente y el resto de productos que forman parte de la aplicación. Obviamente, se aplican diferentes técnicas de prueba a cada tipo de producto *software*. Se desarrollaron pruebas al subsistema en 2 niveles:

- **Pruebas al sistema:** se realiza cuando el *software* funciona en su totalidad para verificar que se han integrado correctamente todos los elementos del mismo.

**Tipo de prueba empleada en este nivel:**

- **Prueba de funcionalidad**

Este tipo de prueba inspecciona si la aplicación cubre sus necesidades de funcionamiento, acorde a las especificaciones de diseño. En ellas se debe verificar si el subsistema lleva acabo correctamente todas las funciones requeridas, se debe además verificar el trabajo apropiado de los requisitos funcionales, incluyendo la navegación, entrada de datos, procesamiento y obtención de resultados.

- **Pruebas de integración:** son ejecutadas para para verificar que un gran conjunto de partes de *software* funcionan juntos. Es una técnica sistemática para construir la estructura del programa mientras que al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción (Pressman, 2005).

### 3.4 Método de prueba

Existen diferentes métodos de pruebas, entre ellos se encuentra prueba de caja negra, el cual se centra principalmente en los requisitos funcionales del *software* y consideran la función específica para la cual fue creado el producto.

Método de caja negra: se lleva a cabo sobre la interfaz del *software*, se concentra en los requisitos funcionales del sistema probándolos y ejercitándolos. Los casos de prueba intentan demostrar que las

# Capítulo III. Validación de la solución

funciones del *software* son operativas. Quien realiza la prueba solo conoce las entradas apropiadas que deberá recibir la aplicación, así como las correspondientes salidas, sin llegar a saber cómo es que se realiza este proceso (Katz-Lichtenstein, 2003).

Los métodos de caja negra poseen diferentes características dentro de las que es importante citar (Rodríguez, 2011):

- ✓ Verifican el correcto manejo de funciones externas provistas o soportadas por el *software*.
- ✓ Verifican que el comportamiento observado se apegue a las especificaciones del producto y a las expectativas del usuario.
- ✓ Los casos de prueba se construyen a partir de las especificaciones del subsistema.

## 3.5 Técnica empleada mediante el método caja negra

La técnica escogida para realizar las pruebas fue la de Particiones de equivalencia para examinar los valores válidos e inválidos de las entradas existentes en el subsistema, que va dirigida y enfocada a la definición de casos de prueba que verificar si el producto satisface los requerimientos del usuario, tal como se describe en las especificaciones de requerimientos.

Un caso de prueba es un conjunto de entradas, condiciones de ejecución y resultados esperados desarrollados con un determinado objetivo y que permiten validar los requerimientos funcionales del subsistema.

Estos casos de prueba deben cumplir la condición de ejecución de que el usuario debe estar registrado en la base de datos del subsistema para poder autenticarse y acceder a las funcionalidades del mismo. A continuación se muestran algunos ejemplos de los casos de pruebas desarrollados al subsistema de asignación de comensales y que permitieron validar cada uno de los requerimientos funcionales del mismo. Para ver los restantes, (Ver anexo 2).

Tabla 15. CP: Asignar grupos a puntos.

Escenario	Descripción	Nombre	Respuesta del sistema	Flujo central
EC 1.1 Nueva	Muestra una ventana con todos	V	Adiciona el evento y luego muestra el	Desplegar el menú Administración situado en la parte superior izquierda.-Seleccionar

## Capítulo III. Validación de la solución

	los aspectos necesarios para que usted adicione una nueva distribución.	Normal	mensaje siguiente: Se ha creado la distribución “nombre distribución”.	la opción Asignar grupos a puntos. Clic en el botón Nueva situado en la parte derecha. Llenar todos los campos correctamente. La aplicación muestra un mensaje de confirmación.
EC 1.2 Activar	Muestra un ventana con todas las distribuciones existentes en la aplicación, donde se debe seleccionar la que desea activar y dar clic en el botón Activar situado en la parte derecha.	NA	Se activa la distribución que usted seleccionó y se muestra de primera en el listado de distribuciones indicando que está activa.	Desplegar el menú Administración situado en la parte superior izquierda, seleccionar la opción Asignar grupos a puntos.-Marcar la distribución que desea activar y finalmente dar clic en el botón Activar situado en la parte derecha.

Tabla 16. CP: Gestionar sede.

Escenario	Descripción	Nombre	Respuesta del sistema	Flujo central
EC 1.0 Adicionar	Muestra una ventana con todas las sedes existentes en la aplicación, en la cual aparece el botón Adicionar.	V	Se adiciona la nueva sede y se muestra en el listado de sedes existentes en la aplicación.	Desplegar el menú Configuración situado en la parte superior izquierda. Seleccionar la opción Gestionar Sede. Hacer clic en el botón Adicionar situado en la parte derecha. - En la ventana que se muestra, introducir el nombre de la nueva sede. Clic en el botón Aceptar situado en la parte derecha. - El subsistema muestra en el listado de sedes, situado debajo en la propia ventana, la nueva sede.
		UCI		
		I	No se adiciona la sede y muestra el mensaje: Debe introducir un nombre para la sede.	
		(vacío)		
EC 1.1 Eliminar	Muestra una ventana con todos las sedes existentes en la aplicación, donde aparece el botón Eliminar. Selecciona la sede que desea eliminar y pulsa el botón.	NA	Muestra el mensaje de confirmación: Seguro que desea eliminar la sede “nombre_sede”?, si da clic en Si, se elimina la sede del listado de sedes.	Desplegar el menú Configuración situado en la parte superior izquierda, seleccionar la opción Gestionar Sede. Marcar la sede que desea eliminar y dar clic en el botón Eliminar situado en la parte derecha. El subsistema muestra un mensaje de confirmación.

## Capítulo III. Validación de la solución

EC 1.1 Activar	Muestra una ventana con todas las sedes existentes en el subsistema donde aparece el botón Activar. Selecciona la sede que desea poner en función y pulsa el botón.	NA	Se activa la sede que usted seleccionó y se muestra de primera en el listado de sedes, indicando que está activa.	Desplegar el menú Configuración situado en la parte superior izquierda, seleccionar la opción Gestionar Sede. Marcar la sede que desea activar y finalmente dar clic en el botón Activar situado en la parte derecha.
-------------------	---	----	---	---

Tabla 17. CP: Balance de puntos.

Escenario	Descripción	Capacidad	Respuesta del sistema	Flujo central
EC 1.1 Balance de puntos	Muestra una ventana que donde permite introducir o cambiar la capacidad a un punto de control determinado, así como la capacidad del comedor al que pertenece y la capacidad total del complejo.	V 500	Se muestra la capacidad que usted introdujo, una vez que seleccione el punto de control correspondiente	Desplegar el menú Administración situado en la parte superior izquierda. Seleccionar la opción Balance de puntos. - Desplegar la estructura deseada para seleccionar el punto de control deseado. Una vez marcada la puerta, introducir la capacidad correspondiente a la misma. - Clic en el botón Aceptar situado en la parte derecha. El subsistema guarda el dato introducido.
		V 100	Se muestra la capacidad que tenía el punto de control inicialmente.	Desplegar el menú Administración situado en la parte superior izquierda. -Seleccionar la opción Balance de puntos.-Desplegar la estructura deseada para seleccionar el punto de control. Una vez marcada la puerta, introducir la capacidad correspondiente a la misma. - Clic en el botón Aceptar situado en la parte derecha. - Clic en el botón Restaurar. El subsistema muestra la capacidad inicial del punto de control.

Tabla 18. CP: Puntos de control.

Opciones	Descripción	Respuesta del sistema	Flujo central
Exportar a PDF	Muestra una ventana con las características más	El subsistema guarda el documento PDF en el lugar que se seleccionó.	Desplegar el menú Reportes situado en la parte superior izquierda. -Seleccionar la opción, Puntos de control. - Clic en el botón Exportar a PDF, situado en la parte superior derecha de la ventana. - El subsistema muestra una ventana para seleccionar donde desea guardar el documento. - Clic en el

## *Capítulo III. Validación de la solución*

	importantes de cada punto de control.		botón Guardar.
Exportar a Word		El subsistema guarda el documento Excel, en el lugar que se seleccionó.	Desplegar el menú Reportes situado en la parte superior izquierda.-Seleccionar la opción Puntos de control.-Clic en el botón Exportar a Excel, situado en la parte superior derecha de la ventana.-El subsistema muestra una ventana para seleccionar donde desea guardar el documento.- Clic en el botón Guardar.

**Tabla 19. CP: Sin asignar.**

Opciones	Descripción	Respuesta del sistema	Flujo central
Sin asignar	Muestra en una ventana todas las personas que no han sido asignadas a ningún punto de control. Brindando además en la propia ventana las opciones de exportar en Excel y PDF dicho reporte.	La aplicación busca las personas que no han sido asignadas a un punto de control y luego las muestra en una tabla en la propia ventana.	Desplegar el menú Reportes situado en la parte superior izquierda. Seleccionar la opción Sin asignar. El subsistema muestra una ventana con todas las personas que no han sido asignadas, así como sus datos correspondientes. Si desea realizar la búsqueda teniendo en cuenta algún criterio, entonces debe introducirlo en el cuadro de texto que aparece en la propia ventana en la parte superior izquierda.
Exportar a Excel		La aplicación guarda el documento en Excel en el lugar que seleccionó.	Desplegar el menú Reportes situado en la parte superior izquierda. Seleccionar la opción Sin asignar. Clic en el botón Exportar a Excel, situado en la parte superior derecha de la ventana. El subsistema muestra una ventana para seleccionar donde desea guardar el documento. Clic en el botón Guardar.
Exportar a PDF		La aplicación guarda el documento en PDF en el lugar que seleccionó.	Desplegar el menú Reportes situado en la parte superior izquierda. Seleccionar la opción Sin asignar. Clic en el botón Exportar a PDF, situado en la parte superior derecha de la ventana. El subsistema muestra una ventana para seleccionar donde desea guardar el documento. Clic en el botón Guardar.

### 3.6 Resultados de las pruebas

Como resultado de las pruebas se obtuvieron un conjunto de no conformidades las cuales fueron solucionándose en cada iteración realizada, lográndose un producto que cumple con los requisitos definidos. En la figura 8 se pueden apreciar los resultados obtenidos en cada iteración.

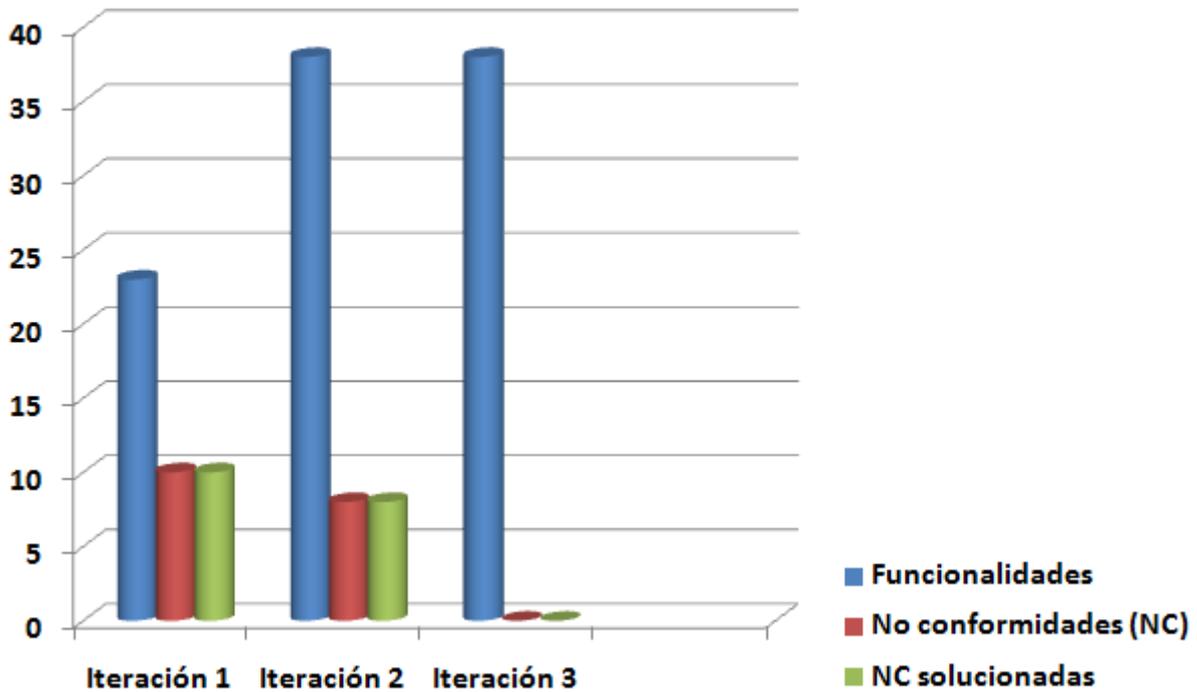


Figura 8. Resultado de las pruebas.

El módulo presenta un total de treinta y ocho funcionalidades. Se realizaron 3 iteraciones con el objetivo de identificar y solucionar los posibles errores existentes. En la primera iteración de un total de veintitrés funcionalidades analizadas, se detectaron diez no conformidades de las cuales 8 estaban relacionadas con errores ortográficos y de concordancia en los mensajes de confirmación, los 2 restantes tenían que ver con el funcionamiento interno de la aplicación. Estas fueron solucionadas en su totalidad antes de realizar la segunda iteración.

En la segunda iteración se probaron todas las funcionalidades detectándose 8 no conformidades de las cuales 4 se relacionaban con la falta de correspondencia en el nombre de algunas funcionalidades entre la aplicación y los casos de prueba, 3 de ortografía y 1 de funcionamiento interno de la

## Capítulo III. Validación de la solución

aplicación. Estas 8 no conformidades fueron también corregidas en su totalidad antes de realizarse la tercera iteración, en la cual no se encontraron no conformidades, lo que indicó que no era necesario realizar otra iteración.

En las 3 iteraciones efectuadas se detectaron un total de dieciocho no conformidades, las cuales en su mayoría respondían a errores de bajo impacto en el correcto funcionamiento de la aplicación y todas tuvieron solución un tiempo máximo de tres días, lo que indica que el subsistema desarrollado presenta buena calidad.

### Resultado de las pruebas de integración

Esta prueba se aplicó para comprobar la construcción de la estructura general del Sistema de planificación y control de la alimentación y al mismo tiempo detectar errores aislados y fáciles de corregir asociados con la interacción. Los resultados de estas pruebas fueron satisfactorios, demostrándose que el Subsistema de asignación de comensales se integra correctamente al Subsistema de control de acceso de comedores y al Subsistema de reservación de alimentación y planificación de menú.

Tabla 20. Prueba de Integración Int\_1 Subsistema de control de acceso

<b>Caso de Prueba</b>
<b>Número de caso de prueba:</b> Int_1
<b>Subsistema a integrar:</b> Subsistema de control de acceso.
<b>Condiciones de ejecución:</b> El Subsistema de asignación de comensales haya introducido los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba:</b> Comprobar que el Subsistema de control de acceso a los comedores de la UCI consulte los datos introducidos en la base de datos por el Subsistema de asignación de comensales.
<b>Entradas/Pasos de ejecución:</b> El Subsistema de asignación de comensales introduce en la base de datos central los datos y el Subsistema de control de acceso a los comedores consulta estos datos y crea el fichero DB4O.
<b>Resultado esperado:</b> Se crea el fichero DB4O con los datos.
<b>Evaluación:</b> Prueba satisfactoria.

## Capítulo III. Validación de la solución

Tabla 11. Prueba de Integración Int\_2 Subsistema de control de acceso

<b>Caso de Prueba</b>
<b>Número de caso de prueba:</b> Int_2
<b>Subsistema a integrar:</b> Subsistema de control de acceso.
<b>Condiciones de ejecución:</b> El Subsistema de asignación de comensales creó el fichero DB4O y no existe conexión con la base de datos central.
<b>Descripción de la prueba:</b> Comprobar que el Subsistema de control de acceso a los comedores de la UCI funcione correctamente con el fichero DB4O creado por el Subsistema de asignación de comensales.
<b>Entradas/Pasos de ejecución:</b> El Subsistema de asignación de comensales crea el fichero DB4O y el Subsistema de control de acceso a los comedores verifica la autenticidad del fichero y utiliza los datos que contiene el mismo.
<b>Resultado esperado:</b> El Subsistema de control de acceso a los comedores de la UCI funciona correctamente.
<b>Evaluación:</b> Prueba satisfactoria.

Tabla 22. Prueba de Integración Int\_3 Subsistema de reservación de alimentación y planificación de menú

<b>Caso de Prueba</b>
<b>Número de caso de prueba:</b> Int_3
<b>Subsistema a integrar:</b> Subsistema de reservación de alimentación y planificación de menú.
<b>Condiciones de ejecución:</b> El Subsistema de reservación de alimentación y planificación de menú introdujo los datos de las reservaciones hechas por las personas en la base de datos central.
<b>Descripción de la prueba:</b> Comprobar que el Subsistema de asignación de comensales consulte los datos introducidos en la base de datos por el Subsistema de reservación de alimentación y planificación de menú.
<b>Entradas/Pasos de ejecución:</b> El Subsistema de reservación de alimentación y planificación de menú introduce en la base de datos central los datos de las reservaciones realizadas por las personas y el Subsistema de asignación de comensales consulta estos datos y crea el fichero DB4O.
<b>Resultado esperado:</b> Se crea el fichero DB4O con los datos.
<b>Evaluación:</b> Prueba satisfactoria.

Tabla 23. Prueba de Integración Int\_4 Subsistema de reservación de alimentación y planificación de menú

## Capítulo III. Validación de la solución

---

<b>Caso de Prueba</b>
<b>Número de caso de prueba:</b> Int_4
<b>Subsistema a integrar:</b> Subsistema de reservación de alimentación y planificación de menú.
<b>Condiciones de ejecución:</b> El Subsistema de asignación de comensales haya introducido los datos en la base de datos central y exista conexión con la misma.
<b>Descripción de la prueba:</b> Comprobar que el Subsistema de reservación de alimentación y planificación de menú consulte los datos introducidos en la base de datos por el Subsistema de asignación de comensales.
<b>Entradas/Pasos de ejecución:</b> El Subsistema de asignación de comensales introduce en la base de datos central los datos y el Subsistema de reservación de alimentación y planificación de menú consulta estos para realizar sus funciones de reservación.
<b>Resultado esperado:</b> Se obtienen los datos necesarios para la reservación.
<b>Evaluación:</b> Prueba satisfactoria.

### 3.7 Conclusiones parciales.

Con el uso de la métrica para la calidad de los requisitos y las técnicas de validación de los mismos se pudo demostrar que los requisitos definidos en el subsistema están completos, son exactos y consistentes. Las pruebas fueron de gran importancia en la garantía de la calidad del subsistema, ya que permitieron identificar errores que atentaban contra el correcto funcionamiento del Subsistema de asignación de comensales mediante el método de prueba Caja negra. Además, se verifica la correcta integración de Subsistema de asignación de comensales con el Subsistema de control de acceso a comedores y el Subsistema de reservación de alimentación y planificación de menú a través de las pruebas de integración. Una vez realizadas las pruebas necesarias se concluyó que el subsistema desarrollado presenta buena calidad.

## Conclusiones.

En el presente trabajo se ha realizado una investigación de vital importancia para el desarrollo del Subsistema de asignación de comensales para la Universidad de las Ciencias Informáticas.

- ✓ En la primera etapa de la investigación se abordaron los aspectos teóricos que facilitaron la comprensión de las principales características y tendencias de sistemas similares que existen en el mundo, en el país y el que se encuentra actualmente en explotación lo que evidenció la necesidad de desarrollar una nueva aplicación que diera solución a la problemática planteada. Se hizo un estudio de las herramientas y lenguajes que posibilitaran el desarrollo teniendo en cuenta las necesidades del cliente.
- ✓ Se realizó el análisis y diseño de la aplicación teniendo en cuenta la investigación realizada previamente, lo que permitió definir todas las funcionalidades necesarias, generando los artefactos asociados al proceso utilizado para guiar el desarrollo del *software* y que facilitan el entendimiento de la solución propuesta por otras personas en el futuro.
- ✓ El subsistema desarrollado utiliza nuevos mecanismos de actualización, de agrupamiento de personas y navegabilidad, áreas en las que existen disímiles problemas en el sistema que se está utilizando hoy en día.
- ✓ Se le realizaron un conjunto de pruebas al subsistema detectando dieciocho no conformidades de bajo impacto, las cuales fueron erradicadas en un corto período de tiempo dando como resultado que el subsistema funciona de la manera en que se previó.

Al concluir el presente trabajo se puede afirmar que el cumplimiento de los objetivos específicos trajeron consigo el desarrollo de un novedoso Subsistema de asignación de comensales que permitirá un aumento en la organización y agilidad en los procesos de asignación y que puede ser utilizado en cualquier institución donde se encuentre desplegado el Sistema de Gestión Universitaria.

## **Recomendaciones.**

Al finalizar la investigación y cumplir el objetivo general de la misma, con el fin de mejorar el funcionamiento del Subsistema de asignación de comensales se recomienda:

- ✓ Desarrollar un sistema de notificaciones que permita a los usuarios de la Dirección de Alimentación conocer los cambios ocurridos diariamente en los datos de las personas y facilite determinar si se hace necesario actualizar o no los datos del sistema.

## Bibliografía referenciada

- Advanced Software. 2011.** Módulo de Gestión y Control de Comedores. [En línea]. [Citado el: 13 de octubre de 2011.] <http://www.advancedsoft.net/pdf/AdvComedor03.pdf>.
- André Ampuero, Margarita y Gulnara Baldoquí, María. 2010.** *Un sistema de soporte a la decisión para la asignación de recursos humanos a equipos de proyectos de software*. La Habana, Cuba : s.n., 2010. Vol. 31.
- Booch, Grady, Jacobson, Ivar Jacobson y Rumbaugh, Jim. 2004.** El Proceso Unificado de Desarrollo de Software. La Habana : Félix Varela.
- BuenasTareas.com. 2012.** buenas tareas. [En línea]. [Citado el: 25 de enero de 2012.]
- Carvajal Riola, Jose Carlos. 2008.** Metodologías ágiles. [pdf] Barcelona, España: s.n., 2008.
- Chrissis, Mary Beth, Konrad, Mike, Shrum, Sandy. 2009.** CMMI Guía para la integración de procesos y la mejora de productos. Pearson Educación
- CAVSI. 2009.** CAVSI Computer Audio Video System Integrator. [En línea]. [Citado el: 7 de diciembre de 2011.] <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>.
- Ebersole. 2010.** Hibernate. [En línea]. [Citado el: 15 de abril de 2012.] <http://www.hibernate.org/343.html>.
- Evolus. 2010.** Pencil Project. [En línea]. [Citado el: 20 de noviembre de 2011.] <http://pencil.evolus.vn/en-US/Home.aspx>.
- Grupo Satellite s.a. 2011.** Grupo satélite. [En línea]. [Citado el: 12 de noviembre de 2011.] [http://www.gruposatellite.net/index.php?option=com\\_content&view=article&id=66&Itemid=69](http://www.gruposatellite.net/index.php?option=com_content&view=article&id=66&Itemid=69).
- Gracia Murugarren, Joaquin. 2003.** IngenieroSoftware. [En línea] 2003. [Citado el: 5 de noviembre de 2011.] <http://www.ingenierosoftware.com/calidad/cmm-cmmi-nivel-2.php>.
- HIBERNATE-Comunity. 2010.** HIBERNATE-Community Documentation. [En línea]. [Citado el: 5 de diciembre de 2011.] <http://docs.jboss.org/hibernate/orm/3.5/reference/es-ES/html/preface.html>.
- Johnson, Rod. 2007.** TheServerSide.com. [En línea]. [Citado el: 01 de diciembre de 2011.] <http://www.theserverside.com/news/1363858/Introduction-to-the-Spring-Framework>.
- Katz-Lichtenstein, Giora. 2003.** O'Reilly on Java. *O'Reilly on Java*. [En línea]. [Citado el: 26 de Abril de 2012.] <http://onjava.com/pub/a/onjava/2003/05/07/blackboxwebtest.html>.
- LARCON-SIA S.R.L. 2009.** IDENTIFICACION Clockcard. [En línea]. [Citado el: 9 de octubre de 2011.] [http://www.larconsia.com/05\\_controldeacceso\\_accpro.asp](http://www.larconsia.com/05_controldeacceso_accpro.asp).

## Bibliografía referenciada

---

- Larman, Craig. 2004.** *Applying UML and Patterns*. s.l. : Addison Wesley Professional, 2004. ISBN: 0-13-148906-2.
- Masterhacks. 2011.** Team- masterhacks. [En línea]. [Citado el: 27 de noviembre de 2011.] <http://www.masterhacks.net/java.php>.
- netbeans.org. 2012.** NetBeans. [En línea]. [Citado el: 29 de enero de 2012.] [http://netbeans.org/index\\_es.html](http://netbeans.org/index_es.html).
- pgadmin.org. 2011.** pgAdmin PostgreSQL Tools. [En línea]. [Citado el: 8 de diciembre de 2011.] <http://www.pgadmin.org/index.php>.
- PostgreSQL. 2011.** PostgreSQL-es. [En línea]. [Citado el: 08 de diciembre de 2011.] [http://www.postgresql.org.es/sobre\\_postgresql](http://www.postgresql.org.es/sobre_postgresql).
- Lenguajes de Programación. 2009.** Lenguajes de Programación. [En línea]. [Citado el: 25 de noviembre de 2011.] <http://www.lenguajes-de-programacion.com/herramientas-de-programacion.shtml>.
- QuinStreet Inc. 2012.** DevX.com. [En línea]. [Citado el: 15 de marzo de 2012.] <http://www.devx.com/Java/Article/27583>.
- Rodríguez Tello, Eduardo A. 2011.** *Importancia de las pruebas de software*. Tamaulipas : s.n., 2011.
- Sáuco Peña, Aida María. 2010.** *Sistema de gestión de asignación de aguas disponibles*. [pdf] Holguín: Centro de Desarrollo de Software Holguín, 2010.
- Schildt, Herbert. 2008.** *Java Manual de referencia*. s.l. : MC GRAW HILL, 2008. ISBN: 9789701062883.
- Scribd Inc. 2011.** Scribd. [En línea] 2011. [Citado el: 27 de noviembre de 2011.] <http://es.scribd.com/doc/57226129/Caracteristica-Del-Java>.
- SpringSource. 2011.** springsource community. [En línea]. [Citado el: 28 de noviembre de 2011.] <http://www.springsource.org/features>.
- Terrence, W. Pratt y Marvin, V. Zelkowitz. 1998.** *Lenguajes de programación : diseño e implementación*. México : Prentice-Hall , 1998. ISBN: 970-17-0046-5.
- VERSANT CORP. 2010.** db4objects. [En línea]. [Citado el: 5 de Febrero de 2010.] <http://www.db4o.com/espanol/>.
- Visual Paradigm. 2010.** Visual Paradigm. [En línea]. [Citado el: 13 de noviembre de 2011.] <http://www.visual-paradigm.com>.
- Walls, Craig. 2008.** *Spring in Action*. s.l.: Manning.

# Bibliografía consultada

---

## Bibliografía consultada

- Blaha, Michael. 2010.** *Patterns of Data Modeling*. United States of America: CRC Press. ISBN: 978-1-4398-1989-0.
- E. Kendall, Kenneth y E. Kendall, Julie. 2005.** *Analisis Y Diseño de Sistemas*. s.l. : Pearson Educación. ISBN : 9702605776.
- Hernández Meléndrez, Edelsys. 2006.** *Cómo escribir una tesis*. Medellín : Escuela Nacional de Salud Pública.
- Johnson, Rod y Hoeller, Juergen. 2011.** *The Spring Framework - Reference Documentation*.
- King, Gavin. 2012.** *Hibernate Reference*.
- Larman, Craig. 2004.** *UML y patrones*. La Habana : Felix Varela.
- McLaughlin, Brett. 2001.** *Java & XML*. s.l. : O'Reilly. ISBN: 0-596-00197-5.
- Piattini Velthuis, Mario Gerardo. 2007.** *Análisis y diseño detallado de aplicaciones informáticas de gestión*. s.l. : Ra-Ma Editorial, S.A. ISBN: 8478977767.
- S. Pressman, Roger. 2005.** *Ingeniería del Software un Enfoque Práctico*. s.l. : MC GRAW HILL. ISBN: 9786071503145.
- Schildt, Herbert. 2008.** *Java Manual de referencia*. s.l. : MC GRAW HILL. ISBN: 9789701062883.
- Silberschatz, Abraham y F. Korth, Henry. 2006.** *Fundamentos de Bases de Datos*. s.l. : McGraw-Hill. ISBN:8448146441.
- Sommerville, Ian. 2005.** *Ingeniería de software*. Madrid : Pearson Educación. ISBN: 84-7829-074-5.
- Walls, Craig. 2008.** *Spring in Action*. s.l. : Manning Publications Co. ISBN: 1-933988-13-4.

## Glosario de términos

### A

**Asignación:** decisión por la cual se determina que una cosa le corresponde a una persona.

Es la acción de señalar, fijar o establecer lo que le corresponde a algo o alguien para un determinado objetivo.

**Ambigüedad:** posibilidad de que algo pueda entenderse de varios modos o de que admita distintas interpretaciones.

### C

**Código abierto:** *software* que pone a disposición de cualquier usuario su código fuente. A pesar de que este tipo de *software* es de buena calidad, el principal atractivo es que es gratis.

**Control de acceso:** el control de acceso es un mecanismo que en función de la identificación ya autenticada permite acceder a áreas, datos o recursos.

**Comensales:** cada una de las personas que comen en una misma mesa.

Personas que comen en un determinado lugar.

### D

**Distribución:** es la división de una cosa en partes dando a cada una de ellas un destino o

una posición según un criterio determinado.

### E

**Evento:** es aquella actividad que se desarrolla en un momento puntual en el tiempo, con un principio y un fin previamente conocido, independientemente de pueda poseer una periodicidad establecida.

### F

**Framework:** es una estructura de soporte definida, mediante la cual otro proyecto de *software* puede ser organizado y desarrollado.

### G

**Grupo:** la palabra grupo deriva del concepto italiano *gruppo* y hace referencia a la pluralidad de seres o cosas que forman un conjunto, ya sea material o mentalmente considerado.

### M

**Multiplataforma:** es un término usado para referirse a los programas, sistemas operativos, lenguajes de programación, u otra clase de *software*, que puedan funcionar en diversas plataformas.

### P

**Plataforma:** es un término de carácter genérico que designa normalmente una arquitectura de *hardware*, aunque también se usa a veces para

**Punto de control:** es un mecanismo que

verifica que se cumplen las condiciones necesarias para permitir el paso de un individuo o de un objeto cualquiera por un determinado lugar.

### R

**Recolector de basura:** es un mecanismo implícito de gestión de memoria implementado en algunos lenguajes de programación.

### S

**Sede:** del latín *sedes* (“silla”, “asiento”), la sede es un lugar donde tienen lugar determinados acontecimientos o donde se erige la base de una organización.

### X

**XML:** siglas en inglés de *eXtensible Markup Language* (lenguaje de marcas extensible), es un metalenguaje extensible de etiquetas desarrollado por el *World Wide Web Consortium* (W3C). XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades, de ahí que se le denomine metalenguaje.

Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir

la información de una manera segura, fiable y fácil. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

Anexo 1

Tabla 24. RF: Mostrar eventos.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
1	Mostrar eventos	A esta funcionalidad podrá accederse mediante la opción "Gestión de eventos" del menú "Administración". Una vez seleccionado se mostrará una interfaz con los datos de los eventos que están activos, y los botones referidos a las acciones "Adicionar", "Modificar" y "Eliminar".	Media	Alta
<b>Prototipo</b>				
				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>			

Tabla 25. RF: Adicionar evento

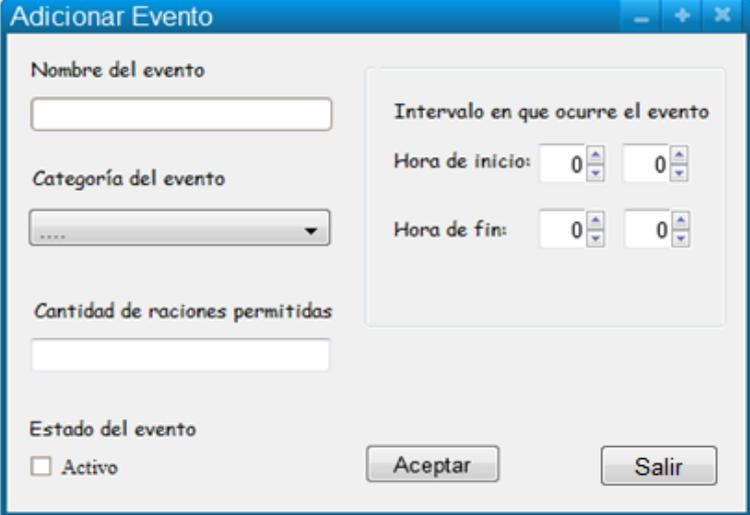
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
2	Adicionar evento	A esta funcionalidad puede accederse mediante el botón "Adicionar" de la interfaz Configuración de eventos. Brinda la posibilidad al usuario de adicionar un nuevo evento.	Media	Media
<b>Prototipo</b>				
				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	Nombre del evento	Texto	No puede ser nulo. o permite la entrada de caracteres especiales. No debe exceder los 30 caracteres.	
	Categoría del evento	Campo de selección		
	Raciones permitidas	Entero	No puede ser nulo. No permite la entrada de letras y caracteres especiales. Pueden escribirse números de hasta 2 cifras.	
	Estado del evento	Campo de selección		
	Hora de Inicio	Campo de selección	No puede ser nulo.	
	Hora de Fin	Campo de selección	No puede ser nulo.	
	<b>Observaciones</b>	<p>En caso de coincidir la hora de inicio y de fin del evento, se muestra un mensaje de error.</p> <p>Cuando se adiciona un evento se muestra un mensaje de información.</p> <p>Si desea adicionar un evento cuyo nombre ya existe, se muestra un mensaje de error.</p> <p>Si alguno de los campos queda sin llenar se muestra un mensaje de error.</p>		

Tabla 26. RF: Modificar evento.

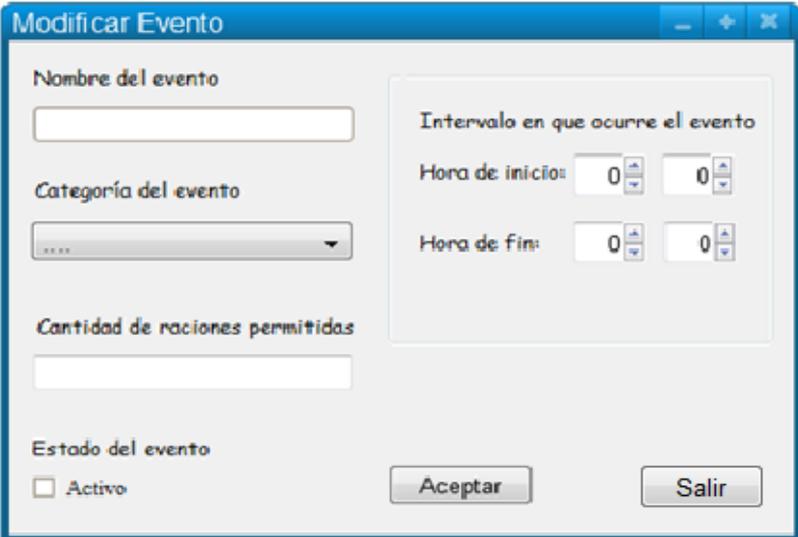
Nombre	Descripción	Complejidad	Prioridad para cliente
3	Modificar evento A esta funcionalidad se accede mediante el botón "Modificar" de la interfaz "Configuración de eventos". Brinda la posibilidad de modificar el evento seleccionado.	Media	Media
<b>Prototipo</b>			
			
Campos	Tipos de Datos	Reglas o Restricciones	
Nombre del evento	Texto	No puede ser nulo. No permite la entrada de caracteres especiales. No debe exceder los 30 caracteres.	
Categoría del evento	Campo de selección		
Raciones permitidas	Entero	No puede ser nulo. No permite la entrada de letras y caracteres especiales. Pueden escribirse números de hasta 2 cifras.	
Estado del evento	Campo de selección		
Hora de Inicio	Campo de selección	No puede ser nulo.	
Hora de Fin	Campo de selección	No puede ser nulo.	
<b>Observaciones</b>	<p>En caso de coincidir la hora de inicio y de fin del evento, se muestra un mensaje de error.</p> <p>Cuando se modifica el evento seleccionado se muestra un mensaje de información.</p> <p>Si desea modificar un evento cuyo nombre ya existe, se muestra un mensaje de error.</p> <p>Si alguno de los campos queda sin llenar se muestra un mensaje de error.</p>		

Tabla 27. RF: Eliminar evento.

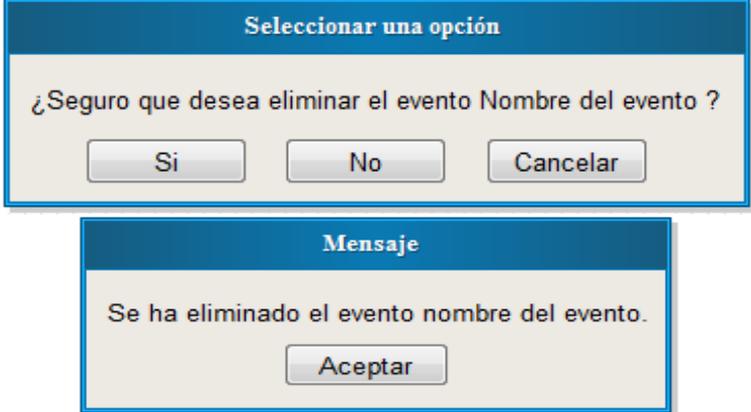
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
4	Eliminar evento	A esta funcionalidad se accede mediante el botón "Eliminar" de la interfaz "Configuración de eventos". Permite eliminar un evento del subsistema, antes debe ser seleccionado.	Baja	Media
<b>Prototipo</b>				
				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	

Tabla 28. RF Mostrar estructuras y puntos.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
5	Estructuras y puntos	A esta funcionalidad se accede mediante la opción "Estructuras y puntos" del menú "Administración". Una vez seleccionado se mostrará una interfaz con la estructura (en forma de árbol) de la Sede, y los botones referidos a las acciones "Adicionar", "Modificar" y "Eliminar".	Alta	Alta
<b>Prototipo</b>				

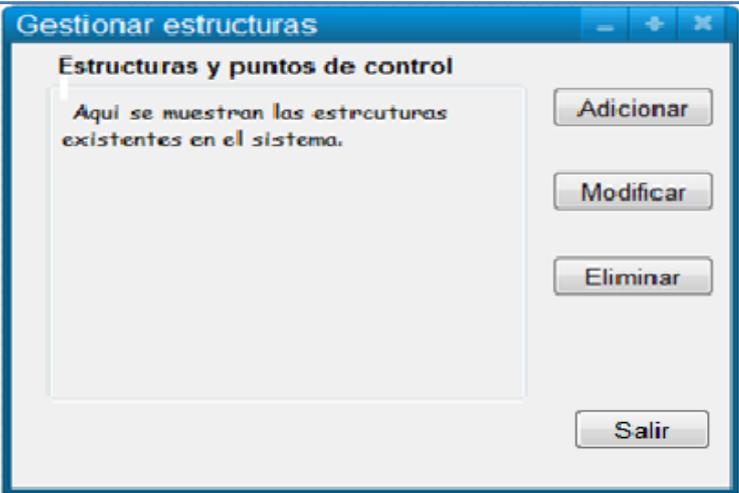
		
<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>
<b>Observaciones</b>		

Tabla 29. RF Crear estructura y punto de control.

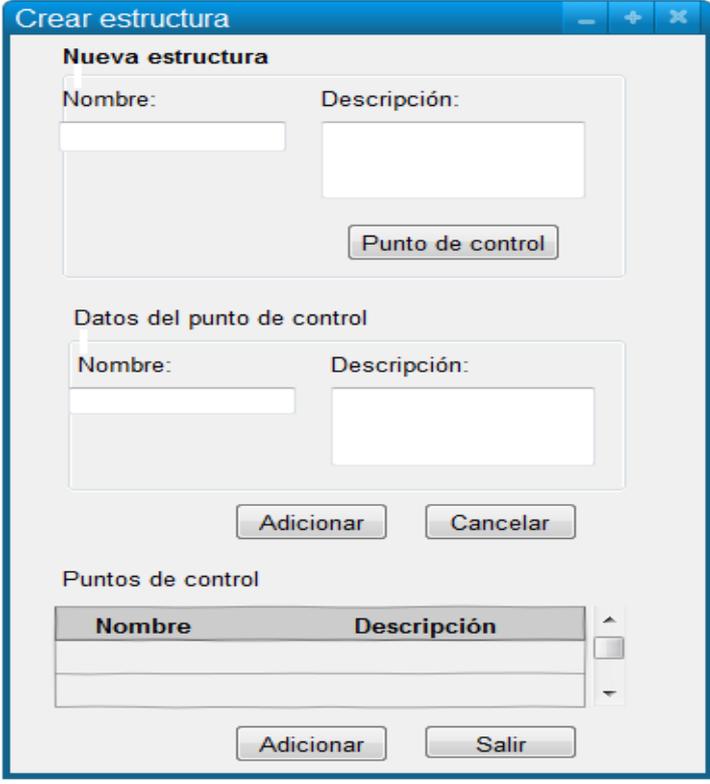
Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
6	Crear estructura y punto de control	A esta funcionalidad se accede mediante el botón "Modificar" de la interfaz "Estructuras y puntos". Brinda la posibilidad de adicionar una nueva estructura y/o los puntos de control que le pertenecerán a la misma.	Media	Media
<b>Prototipo</b>				
				
<b>Campos</b>		<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
Nombre (estructura)		Texto	No puede ser nulo. No admite caracteres especiales. No debe exceder los 30 caracteres.	
Descripción (estructura)		Texto		
Nombre (punto de control)		Texto	No puede ser nulo. No admite caracteres especiales. No debe exceder los 30 caracteres.	
Descripción (punto de control)		Texto		
<b>Observaciones</b>		Al crear una estructura y/o un punto de control, se muestra un mensaje de información. Si desea crear una estructura y/o un punto de control que ya existe, se muestra un mensaje de error.		

Tabla 30. RF Modificar estructura y punto de control.

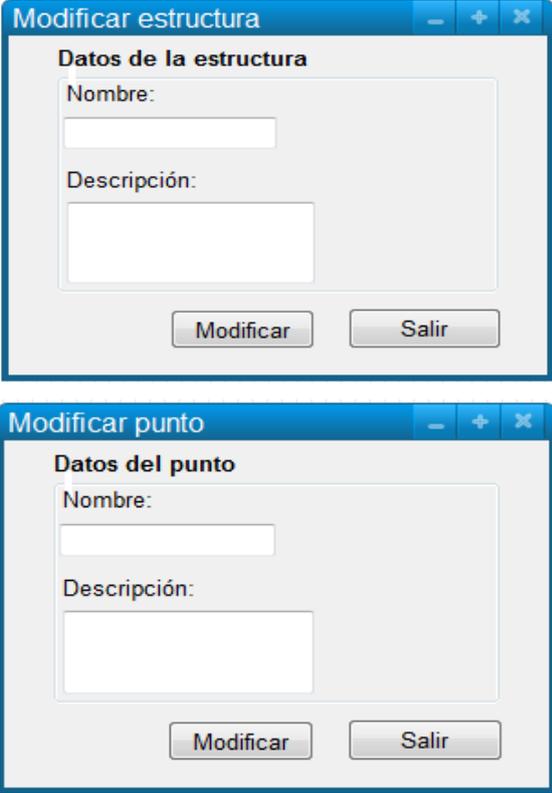
Nº	Nombre	Descripción	Complejidad	Importancia para cliente
7	Modificar estructura y punto de control	A esta funcionalidad se accede mediante el botón "Modificar" de la interfaz "Estructuras y puntos". Brinda la posibilidad de modificar una estructura o un punto de control.	Media	Media
<b>Prototipo</b>				
				
	<b>Campos</b>	<b>Tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	Nombre	Texto	No puede ser nulo. No admite caracteres especiales. No debe exceder los 15 caracteres.	
	Descripción	Texto		
	<b>Observaciones</b>	<p>Si no se selecciona la estructura o el punto de control que desea modificar el subsistema debe mostrar un mensaje de error.</p> <p>Al modificar la estructura o el punto de control seleccionado, se muestra un mensaje de información.</p> <p>Si desea modificar una estructura o un punto de control cuyo nombre ya existe, se muestra un mensaje de error.</p>		

Tabla 31. RF Eliminar estructura.

Nº	Nombre	Descripción	Complejidad	Prioridad para cliente
8	Eliminar estructura	A esta funcionalidad se accede mediante el botón "Eliminar" de la interfaz "Estructuras y puntos". Brinda la posibilidad de eliminar una estructura existente.	Media	Media
<b>Prototipo</b>				
	<b>Campos</b>	<b>tipos de Datos</b>	<b>Reglas o Restricciones</b>	
	<b>Observaciones</b>	Si no selecciona la estructura que desea eliminar se muestra un mensaje de error. Al eliminar una estructura seleccionada se muestra un mensaje de confirmación.		

## Anexo 2

Tabla 32. CP Gestionar estructuras.

Escenario	Descripción	Nombre	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Adicionar	Muestra una ventana con las estructuras existentes en el sistema y en la parte derecha el botón Adicionar.	V	NA	Adiciona la estructura y muestra el mensaje siguiente: Se adicionó la estructura: "nombre_estructura".	Desplegar el menú Administración situado en la parte superior izquierda, seleccionar la opción Estructuras y puntos. Luego dar clic en el botón Adicionar situado en la parte derecha. -Llenar correctamente los campos de la nueva estructura y dar clic en el botón Adicionar. - El sistema muestra un mensaje de confirmación.
		Complejo 1	dato		
		I	NA	No adiciona la estructura deseada y muestra el mensaje siguiente: Introduzca un nombre para la nueva estructura	
		(Vacío)	dato		
EC 1.2 Modificar	Muestra una ventana con las estructuras existentes en el sistema y en la parte derecha en botón Modificar. Selecciona la que desea modificar y pulsa en el botón.	V	NA	Modifica correctamente la estructura y muestra el mensaje siguiente: La estructura "nombre_estructura" ha sido modificada.	Desplegar el menú Administración situado en la parte superior izquierda, seleccionar la opción Estructuras y puntos, seleccionar la estructura que desea modificar – Clic en el botón Modificar situado en la parte derecha. - Llenar correctamente los campos que desea modificar y hacer clic en el botón Modificar, situado en la parte inferior de la ventana. - El sistema muestra un mensaje de confirmación.
		Complejo 2			
EC 1.3 Eliminar	Muestra una ventana con las estructuras existentes en el sistema y en la parte derecha en botón Eliminar. Selecciona la que desea eliminar y pulsa en el botón.	NA	NA	Muestra el mensaje siguiente: Seguro que desea eliminar la estructura: "nombre_estructura". Si da clic en Si, luego aparece el siguiente cartel: Se ha eliminado la estructura "nombre_estructura".	Desplegar el menú Administración situado en la parte superior izquierda, seleccionar la opción Estructuras y puntos y dar clic en el botón Eliminar situado en la parte derecha. - El sistema muestra un mensaje de confirmación.

Tabla 33. CP: Gestionar grupo.

Escenario	Descripción	Nombre del grupo	Descripción	Respuesta del sistema	Flujo central
EC 1.1 Adicionar	Una vez pulsado el botón Adicionar se muestra una ventana para que usted introduzca los datos del nuevo grupo.	V	NA	Adiciona el nuevo grupo y muestra el mensaje siguiente: Se ha adicionado el grupo Internos.	Desplegar el menú Administración situado en la parte superior izquierda. - Seleccionar la opción Gestión de grupos. -Clic en el botón Adicionar situado en la parte inferior izquierda. Se llenan todos los campos correctamente. - El sistema muestra un mensaje de confirmación.
		Internos			
		I	NA	No se adiciona el grupo y muestra el mensaje: Introduzca el nombre del grupo.	
		(Vacío)			
EC 1.1 Modificar	Una vez que seleccione el grupo que desea modificar pulsa el botón Modificar y se muestra una ventana para que usted realice los cambios al grupo seleccionado.	V	NA	Realiza los cambios y muestra el mensaje siguiente: El grupo Internos1 ha sido modificado.	Desplegar el menú Administración situado en la parte superior izquierda. - Seleccionar la opción Gestión de grupos. - Marcar el grupo que le desea realizar cambios. -Clic en el botón Modificar situado en la parte inferior derecha. - Se llenan todos los campos correctamente. - El sistema muestra un mensaje de confirmación.
		Internos1	NA		
		I	NA	No realiza los cambios y muestra el mensaje siguiente: Introduzca el nuevo nombre del grupo.	
		(Vacío)			
EC 1.1 Eliminar	Una vez que seleccione el grupo que desea eliminar del sistema, pulsa el botón Eliminar.	NA	NA	Muestra el mensaje: Seguro que desea eliminar el grupo Internos, si da clic en Si, se muestra posteriormente el mensaje: Se ha eliminado el grupo Internos.	Desplegar el menú Administración situado en la parte superior izquierda. - Seleccionar la opción Gestión de grupos. - Marcar el grupo que desea eliminar. -Clic en el botón Eliminar situado en la parte inferior derecha. - El sistema muestra un mensaje de confirmación

**Tabla 34. CP: Adicionar y eliminar personas a grupos.**

Escenario	Descripción	Concepto	Valor	Respuesta del sistema	Flujo central
EC 1.1 Adicionar	Una vez pulsado el botón Adicionar se muestra una ventana para que usted introduzca un concepto para realizar la búsqueda de la/las personas que desea adicionar.	V  Nombre	Valor  Dayami	Adiciona el nuevo grupo y muestra el mensaje siguiente: Se han adicionado "cantidad_encontrada" de personas al grupo "nomb_grupo" seleccionado.	Desplegar el menú Administración situado en la parte superior izquierda. - Seleccionar la opción Gestión de grupos. - Marcar el grupo al que desea adicionarle personas. - Clic en el botón Adicionar situado en la parte superior derecha. Seleccionar el concepto por el que desea buscar la o las personas, en la lista desplegable. Editar el campo Valor situado en la parte superior derecha de la ventana, de acuerdo al concepto antes seleccionado. Activar de acuerdo a sus necesidades el botón Asignadas o Sin asignar para realizar la búsqueda - El sistema realiza la búsqueda y muestra en la propia ventana las personas encontradas. Luego dar clic en el botón Adicionar situado en la parte inferior derecha, para adicionar la/las personas encontradas al grupo antes seleccionado.
EC 1.1 Eliminar	Una vez que seleccione las personas que desea eliminar, pulsa el botón Eliminar.	NA	NA	Muestra el mensaje: Seguro que desea eliminarla/las personas seleccionadas?, si da clic en Si, se muestra posteriormente el mensaje: Las personas han sido eliminadas.	Desplegar el menú Administración situado en la parte superior izquierda. - Seleccionar la opción Gestión de grupos. Marcar el grupo del cual desea eliminar las personas. - Clic en el botón Eliminar situado en la parte inferior derecha. - El sistema muestra un mensaje de confirmación.

Para más información referente a la especificación de requisitos y los diseños de casos de pruebas del subsistema, favor de remitirse al expediente de proyecto Sistema de planificación y control de servicios de alimentación. El cual se encuentra disponible en la siguiente dirección:

[https://repositorio.cenia.prod.uci.cu/svn/documentacion/Dpto\\_UD/Alimentacion/](https://repositorio.cenia.prod.uci.cu/svn/documentacion/Dpto_UD/Alimentacion/)