Universidad de las Ciencias Informáticas

Facultad 7

Título: Implementación del módulo Nutrición y dietética del Sistema de Información Hospitalaria alas HIS

Autores: Josué Rodríguez Ronquillo

Ramsés Fernández Martínez

Tutores: Ing. Keila García Nogueira

Ing. Juan Manuel García Orduñez

La Habana, junio de 2011

“Año 53 de la Revolución”

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Muchas son las cátedras universitarias, pero escasos los maestros sabios y nobles. Muchas y grandes son las aulas, mas no abundan los jóvenes con verdadera sed de verdad y justicia.

La Ciencia es una tentativa en el sentido de lograr que la caótica diversidad de nuestras experiencias sensoriales corresponda a un sistema de pensamiento lógicamente ordenado.

Albert Einstein

Datos de contacto

Ing. Keila García Nogueira

Graduado en el año 2007 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Posee categoría docente de Instructor. Ha impartido la asignatura Ingeniería de software. Actualmente se desempeña como analista en el proyecto Sistema de Información Hospitalaria perteneciente al Departamento de Sistemas Gestión Hospitalaria del Centro de Informática Médica (CESIM).

Correo electrónico: kgarcia@uci.cu

Ing. Juan Manuel Garcia Ordúñez

Instructor recién graduado en el año 2009 de Ingeniero en Ciencias Informáticas en la Universidad de las Ciencias Informáticas. Profesor vinculado a la Facultad 7 y miembro del Departamento de Sistema de Gestión Hospitalaria. Actualmente se desempeña como jefe de un módulo en el proyecto Sistema de Información Hospitalaria perteneciente al Departamento de Sistemas Gestión Hospitalaria del Centro de Informática Médica (CESIM).

Correo electrónico: jmgarcia@uci.cu

RESUMEN

El módulo de Nutrición y Dietética del sistema alas-HIS desarrollado por el Departamento de Gestión Hospitalaria del CESIM perteneciente a la Universidad de las Ciencias Informáticas, permite evaluar el estado nutricional de un paciente en una institución hospitalaria. Una de las razones por la cual se trabaja en este módulo es para poder facilitar las diferentes problemáticas que se encuentran en los hospitales a la hora de llevar el control de la Nutrición y Dietética. Debido a que en las instituciones hospitalarias las personas capacitadas en Nutrición y Dietética son un personal reducido, no se visitan todos los servicios de hospitalización requeridos, sino que se evalúan solamente aquellos pacientes que serán intervenidos quirúrgicamente, cardiópatas y nefrópatas, entre otros.

Para el desarrollo de este módulo se utilizaron las herramientas establecidas por el Departamento de Sistema de Información Hospitalaria alas HIS. Como lenguaje de programación orientado a objetos del lado del servidor, Java, Eclipse como Entorno de Desarrollo Integrado, PostgreSQL como Sistema Gestor de Bases de Datos. Además se empleó Hibernate como herramienta ORM para la persistencia de los datos y el framework Seam para la lógica del negocio, entre otras tecnologías.

Con la implementación de este módulo se espera agilizar la atención al paciente aprovechando las facilidades que brindan las tecnologías de la información y las comunicaciones. Además de brindarle al paciente un servicio con mayor calidad. El sistema permitirá evaluar el estado nutricional de los pacientes, así como, diagnosticarle una dieta acorde con las necesidades que presenten los mismos. Se podrán generar reportes estadísticos para estudios posteriores.

PALABRAS CLAVES:

Evaluación nutricional, Nutrición y Dietética, Patrón de dieta.

Índice

INTRODUCCIÓN

Desde la antigüedad, el hombre ha tenido la necesidad de trasmitir y procesar la información, por lo que ha creado métodos y máquinas que le faciliten esta labor. A partir del surgimiento de las computadoras se elaboraron programas informáticos con objetivo inicial de cálculo; sin embargo, los mismos fueron ganando en complejidad hasta permitir el almacenamiento de grandes volúmenes de información. En los últimos años la informática ha alcanzado un notable desarrollo en las distintas ramas de la sociedad, brindando tecnologías que agilizan la obtención de resultados y el almacenamiento de datos.

La aplicación de las Tecnologías de la Información y las Comunicaciones (TIC) se ha evidenciado durante años en la mayoría de los sectores sociales, debido a las necesidades, cada vez mayores, de simplificar y agilizar las tareas que estos realizan. Por otra parte el uso de las TIC proporciona formas más efectivas de obtener conocimiento, con el objetivo de aplicarlos en los disímiles retos que presenta hoy el desarrollo de la humanidad.

La meta fundamental de estos sistemas es informatizar la sociedad aportando grandes beneficios tales como: la reducción de errores provocados en la entrada de datos, la disminución de espacio físico destinado a su almacenamiento y la rapidez al momento de buscar algún dato en particular. Por la gran cantidad de datos que se producen a diario en las organizaciones ha sido necesario buscar alternativas más seguras para almacenarlos y gestionarlos. Los programas informatizados juegan entonces un papel importante, para dar solución a esta problemática, de forma tal que faciliten la automatización de procesos operativos y suministrando información que sirva de apoyo a la toma de decisiones.

Uno de los sectores donde se han incorporado aplicaciones informáticas progresivamente es la salud, debido al gran número de documentos clínicos que constantemente se manejan en los centros hospitalarios. Por otra parte la necesidad de obtener reportes estadísticos más específicos y que brinden información actualizada y fidedigna ha propiciado la explotación de software en el sector hospitalario. El uso de programas informáticos mejora considerablemente los servicios de planificación sanitaria, la atención a los pacientes y la gestión de los recursos o insumos en cualquier centro médico.

La incorporación de sistemas de información en la salud, apoyados en las TIC, provee oportunidades para mejorar los procesos internos y mejora la labor o desempeño por parte de los trabajadores en los diferentes niveles de atención. Cada día los procesos de registro, seguimiento y tratamiento del paciente deben mejorarse e innovarse para hacer más eficientes las actividades rutinarias hospitalarias. No basta

con tener datos e información, hay que procesarla, analizarla, interpretarla y utilizarla. Apoyándose en las tecnologías, se logra la optimización de los recursos humanos y materiales, para satisfacer las necesidades de las áreas operativas, administrativas, clínicas y de investigación en las organizaciones de salud. (1)

La masificación del uso de las microcomputadoras originó lo que se ha denominado la revolución de la informática. Con el surgimiento de los primeros sistemas de información médica, se logró un avance en las tareas administrativas, se agilizó el proceso de la admisión y alta de pacientes y los mismos facilitaron un mayor control de inventarios. Estos sistemas fueron evolucionando llegando a especializarse en los diferentes niveles de la salud, alcanzando un notable desarrollo en los hospitales, dando lugar a la creación de los Sistemas de Información Hospitalaria, conocido por sus siglas en inglés como HIS (Hospital Information System).(2)

Los Sistemas de información Hospitalaria están diseñados con el objetivo de mejorar la calidad y la eficiencia del trabajo en los centros de atención médica. Los mismos se encargan de almacenar, procesar, recopilar, recuperar y comunicar los datos obtenidos en la atención al paciente y en la labor administrativa de las instituciones hospitalarias. Estos sistemas han tenido gran impacto, ya que aumentan la productividad de los profesionales y técnicos de la salud los cuales no tienen que invertir su tiempo en elaborar grandes informes de resultados, sino que los obtienen haciendo uso de estos sistemas.

Asimismo estas aplicaciones apoyan la administración de los recursos humanos y materiales de la institución, incrementando la eficacia en la gestión de los mismos, minimizando los inconvenientes burocráticos que puede enfrentar el paciente. En la actualidad, estos sistemas informáticos gestionan la mayor parte de la información generada en un hospital, incluyen un registro histórico de la información médica del paciente, control de laboratorios y quirófanos, generación de recetas, órdenes médicas, generación de reportes estadísticos y nutrición y dietética. (3)

En Cuba, se ha llevado a cabo un seguimiento a los sistemas de gestión hospitalarios, con el objetivo de brindar un servicio de mejor calidad en los hospitales. Sin embargo, en la actualidad no existe un sistema que realice la gestión de los procesos de todas las áreas de un centro de salud. Algunos de estos sistemas realizan la gestión de algunas áreas como Admisión, Consulta Externa, Estadísticas, Hospitalización, entre otros, pero estos no constituyen una solución integral. En el año 2007 y a raíz del proceso de informatización de la sociedad cubana, la facultad 7 de la Universidad de las Ciencias Informáticas se

propuso el desarrollo de un HIS que automatizara los procesos de gestión de la información médico-administrativa; así como de los recursos humanos y materiales de las distintas áreas de las instituciones hospitalarias.

Este sistema cuenta con diferentes módulos básicos como los de Emergencias, Epidemiología, Admisión, entre otros, pero no contempla uno que se encargue de gestionar los procesos que se realizan en el área de Nutrición y Dietética en las instituciones hospitalarias. En esta área hospitalaria se realizan los procesos relacionados con la evaluación del estado nutricional de los pacientes y la gestión de dietas de los mismos. Para evaluar el estado nutricional de un paciente se utiliza una planilla confeccionada en formato papel. En el formulario no se tienen en cuenta un grupo de resultados que se calculan específicamente a partir de los datos antropométricos y durante la visita al paciente se dificulta su obtención. A causa del proceso manual se entorpece la evaluación nutricional del paciente hospitalizado.

Debido a que en las instituciones hospitalarias el personal capacitado en Nutrición y Dietética es un personal reducido, no se visitan todos los servicios de hospitalización requeridos, sino que se evalúan solamente aquellos pacientes que serán intervenidos quirúrgicamente, cardiópatas y nefrópatas, entre otros. Otra de las problemáticas existentes se relaciona con un sistema informático que cuenta con una tabla de composición de alimentos y los aportes calóricos de los mismos, sin embargo no hace uso de las evaluaciones nutricionales para sugerir la distribución de alimentos de un paciente en un día, por otra parte la base de datos de alimentos con la que cuenta es reducida. Los patrones de dieta de cada paciente también son elaborados en formato papel, lo que ocupa espacios innecesarios en locales hospitalarios.

Para la conformación de estos patrones de dieta, se utiliza el parte de dieta que emiten los diferentes servicios, el cual en ocasiones no se entrega a tiempo porque no se tiene toda la información del paciente, causando ciertos desajustes en la entrega a la cocina de la lista de alimentos que deben ser preparados. Por otra parte, existe un grupo de nutrientes para pacientes que van a presentarse a cirugía, los cuales son adquiridos por el país a altos precios. La gestión del consumo por dieta individual debe arrojar el gasto total que representa para el hospital. Debido a las limitantes abordadas anteriormente surge la necesidad de mejorar los procesos de gestión de la información en el área de Nutrición y Dietética en instituciones hospitalarias.

Después de haber realizado un amplio análisis a la situación problémica, se define el siguiente **problema a resolver:**

¿Cómo mejorar la gestión de la información de la evaluación nutricional y gestión de dietas que se realizan en el área de Nutrición y Dietética de las instituciones hospitalarias?

Este problema tiene como **objeto de estudio** el proceso de gestión de información de las instituciones hospitalarias y se enmarca en el **campo de acción** en el proceso de gestión de la información del área de Nutrición y Dietética de las instituciones hospitalarias.

Para dar solución al problema planteado se propone como **objetivo general**: Realizar la implementación de los procesos definidos en el módulo Nutrición y Dietética del Sistema de Información Hospitalaria alas HIS.

Para dar cumplimiento a estos objetivos se acordaron las siguientes **tareas a desarrollar:**

1. Evaluar las tendencias actuales en el mundo de los sistemas de información hospitalaria que cuenten con el módulo de Nutrición y Dietética.
2. Aplicar la arquitectura definida por el Departamento de Sistema de Gestión Hospitalaria para el desarrollo de sus aplicaciones.
3. Aplicar las pautas de desarrollo de los entregables y diseño definidas en el Departamento de Sistema de Gestión Hospitalaria.
4. Implementar los procesos de negocio Evaluación del estado nutricional de pacientes y Gestión de dietas.
5. Obtener los artefactos correspondientes a los flujos de trabajo “Implementación” y “Pruebas”.
6. Obtener el acta de liberación otorgada por Calidad.

Con la implementación del sistema se espera que los beneficios sean agilizar el procesamiento manual de la información, contribuyendo a la informatización de las instalaciones hospitalarias, obtener estadísticas en tiempo real que permitan controlar el estado nutricional de los pacientes, disminuir el tiempo de creación del patrón de dieta a partir del parte de dieta para lograr una mayor eficiencia y garantizar el

control centralizado de la nutrición y dietética de los pacientes que posibilite centrar la gestión de la información.

El documento está estructurado en cuatro capítulos:

Capítulo 1 “Fundamentación Teórica”: brinda un estado del arte de los Sistemas de Información Hospitalaria que gestionan los procesos asociados al área de Nutrición y Dietética.

Capítulo 2 “Descripción de la arquitectura”: se definen los requisitos no funcionales, se describe la propuesta de arquitectura y se hace un análisis de la implementación del sistema.

Capítulo 3: “Descripción y análisis de la solución propuesta”: se aborda el análisis y diseño de la aplicación. Se presentan los artefactos generados una vez analizado el modelo de datos y la vista de implementación.

Capítulo 4: “Modelo de Prueba”: se define como método de prueba a utilizar las de caja negra y se diseñan los casos de pruebas correspondientes a los Casos de Usos del Sistema seleccionados.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

En este capítulo se brinda información detallada sobre los Sistemas de Información Hospitalaria a nivel mundial, así como las tendencias actuales de los HIS que cuentan con la presencia y los servicios de un módulo de Nutrición y Dietética. Además se realiza un estudio crítico y valorativo de las metodologías, técnicas de programación, plataformas y librerías usadas para el desarrollo del módulo.

1.1 Conceptos básicos relacionados con el problema planteado

Para comprender los términos empleados en esta investigación se ha de conocer el ámbito hospitalario, por lo que a continuación se explican algunos conceptos que no forman parte del lenguaje común.

Reporte:

Documento generado por el Sistema, que muestra de forma estructurada y/o resumida datos sobresalientes, guardados o generados por la misma aplicación.

Nutrición y Dietética:

Existe un proceso relacionado con la alimentación de los pacientes al que se llama Nutrición y Dietética. Esta sección es la encargada de evaluar el estado nutricional de un paciente en la institución hospitalaria y mantener actualizados todos sus datos.

Parte de dietas:

Documento que muestra de manera detallada la alimentación que requiere cada paciente en específico.

Patrón de dietas:

Esta sección se encarga de gestionar los alimentos necesarios que se le deben asignar a cada paciente en las diferentes secciones del día.

1.2 Sistemas automatizados existentes

Existen varios sistemas en la actualidad que contienen, de una forma u otra, funcionalidades que se manejan en el área de Nutrición y Dietética de los centros hospitalarios. Algunos integran estas funcionalidades a su sistema, otros simplemente trabajan con los procesos de esta área. Después de un amplio estudio sobre algunos de los sistemas más importantes existentes en la actualidad que contemplan un módulo de Nutrición y Dietética están:

TrakCare:

Producido por InterSystems Corporation, el cual es una aplicación web que gestiona información sanitaria, que contiene módulos clínicos y administrativos los cuales serán interoperables con futuras aplicaciones. Este software avanzado, brinda mejoras exponenciales a la atención de cada paciente y a la productividad de los profesionales de la salud, además cuenta con un Registro Electrónico de Pacientes.

Se ha extendido a casi 25 países y es utilizado en las principales instituciones de los mismos. Está implantado tanto en hospitales sencillos como complejos, integrado completamente a una red de salud que sirve a 6,5 millones de pacientes en Brasilia y sus alrededores. Puede adaptar fácilmente a una amplia variedad de público y a sectores privados de la salud. Es un software propietario y multiplataforma. Maneja el estándar internacional de intercambio y acceso a datos Health Level Seven International (HL7).

Funcionalidades que ofrece:

- Permite crear una consulta de nutrición para pacientes ambulatorios.
- Contempla dietas y/o protocolos alimentarios predefinidos que después el médico/técnico de nutrición adapta a las necesidades de cada paciente.
- Permite crear cualquier tipo de formulario, haciendo posible trasladar cualquier papel a formato electrónico.
- Posee dietas predefinidas (dieta normal 1500 Kcal, dieta líquida, etc.) para los pacientes hospitalizados que pueden ser personalizadas.
- Recepción de todas las dietas y/o suplementos en una lista de trabajo disponible para el personal de la cocina.

HOSIX-V:

Desarrollado por la empresa Sivsa. Este es un software de gestión hospitalaria diseñado en una arquitectura Cliente-Servidor accesible desde clientes web, lo que permite ser ejecutado desde cualquier navegador ya sea vía Intranet o Internet. Sivsa presenta un sistema de Gestión e Información Hospitalaria de fácil uso que contempla todas las áreas de actividad del ambiente de la salud. Ha sido desarrollado utilizando como lenguaje de programación Microsoft Visual Basic y herramientas que son propiedades de la compañía Microsoft. Está constituido por diferentes módulos que pueden funcionar de forma autónoma. El de Nutrición y Dietética se encuentra en el Área de Servicios Centrales. Además, es también un Sistema Integrado, el cual se adapta con facilidad a todo tipo de organización.

Funcionalidades que ofrece:

- Ficha técnica del plato con la identificación de los valores nutricionales de los componentes.
- Permite la identificación de dietas normales, entéricas, suplementos y volumen (biberones), con identificación de componentes y número de tomas.
- Indicación del valor energético de los componentes del plato.
- Organización de las áreas de distribución, según criterios (enfermería, cocina).
- Permite atribución de menús de suplementos a pacientes no ingresados.
- Planificación de la administración de dietas por planta, piso, servicio.

Sushrut:

Elaborado por la empresa C-DAC (Centre for Development of Advanced Computing), con el objetivo de racionalizar el flujo de tratamiento de un paciente en el hospital, mientras que permite a los médicos y demás personal del hospital desarrollar su capacidad máxima, de manera óptima y eficiente. Este HIS es modular, lo que garantiza beneficios de manera sostenida a través de cambios en la tecnología, la protección y la prestación de rendimientos óptimos de la inversión. Utiliza una red de ordenadores para almacenar, procesar y recuperar la atención del paciente y la información administrativa de todas las actividades del hospital para cumplir con los requisitos funcionales de los usuarios. Se comporta como un sistema de apoyo a la decisión de las autoridades del hospital para el desarrollo de políticas de la atención integral para la salud. También contiene un módulo de cocina el cual se encarga de la gestión de los

servicios del menú de los pacientes. Este módulo depende del módulo Admisión. La dieta sólo puede ser prescrita a los pacientes ingresados.

Funcionalidades que ofrece:

- Programación de comidas
- Cancelación de comida.
- Generar informe de dietas.
- Generar informe de dietas canceladas.
- Dieta terapéutica.

SATN2009:

Sistema de Tratamiento y Análisis Nutricional. Esta aplicación de escritorio es de la última generación de herramientas de software de análisis-nutrición elaborada por el Colegio de Farmacéuticos de Madrid, que incorpora el sistema INBO-DIET. Posee una comprensiva base de información y conocimiento de los alimentos, además de poderosas herramientas. Proporciona análisis completos de dietas, recetas y menús.

La base de datos de alimentos está basada en el Archivo de Nutrientes de la Universidad de J.Liebig de Giessen (1991 y 1997 ediciones), la composición de los alimentos del Ministerio Español de Sanidad y Consumo 1999, la USDA Nutrient Database for Standard Reference SR12, 1998 y otras fuentes, proporcionando análisis para 50 nutrientes comunes.

Funcionalidades que ofrece:

- Permite una completa gestión de los pacientes.
- Para recomendar una ingesta alimentaria utiliza la ecuación de Harris Benedict.
- Contiene una base de datos de alimentos que contiene información nutritiva para más de 800 alimentos.
- Incluye la tabla de conversión de medidas.

- Permite la creación de dietas introduciendo cada uno de los alimentos que la componen, seleccionando su tipo de cocción y su cantidad justa, el horario y el día en que se van a tomar, etc.
- Permite generar informes.

Este módulo sirve para valorar el perfil y las necesidades nutricionales de cada persona, en función de sus características, estado de salud y analítica, pudiendo además personalizar y planificar sistemas de alimentación, dietas y menú, con el objeto de obtener el peso ideal, comer lo adecuado para vivir con salud, disminuir los niveles de enfermedad y aumentar en lo posible la esperanza de vida. Es un software propietario.

Birla Medisoft Quanta WebHIMS V 1.0:

Sistema de gestión información hospitalaria, basado en web, capaz de informatizar cualquier hospital a gran escala. Está totalmente interconectado a la web, lo que hace que la información esté disponible en cualquier momento y lugar. Además, maneja las principales áreas de un hospital. Tiene la portabilidad y la conectividad necesaria para funcionar en prácticamente todas las plataformas de hardware estándar, con estricta seguridad y fácil recuperación de datos en caso de falla del sistema.

Funcionalidades que ofrece:

- Prescripción diaria de una dieta por paciente.
- Instrucciones para preparar la dieta de los pacientes.
- Permite ver las dietas prescritas en un determinado período de tiempo.

Ceres:

Entre los sistemas automatizados a nivel nacional se encuentra Ceres, una aplicación de escritorio producida por el Instituto de Nutrición e Higiene de los Alimentos (INHA) para la evaluación del consumo de alimentos. Este contiene unas tablas de recomendación de calorías y nutrientes. Es comercializado por la empresa CENTERSOFT.

xHIS:

Desarrollado por la compañía ISOFT, del grupo IBA Health, es multiplataforma (WINDOWS, LINUX, UNIX), independiente de la base de datos y presenta una arquitectura cliente-servidor y en 3 capas.

Gestiona la asignación de las dietas a los pacientes ingresados en el hospital. Igualmente el facultativo podrá definir las dietas para los familiares de los enfermos y para el personal propio del hospital. La aplicación permite realizar diariamente las solicitudes de dietas a la cocina del hospital de manera que el personal de la cocina pueda elaborar todos los paltos necesarios para ese día. Incluye también la gestión y el control de los refrigerios servidos a los pacientes desde las unidades de enfermería. Incorpora unos Listados-Resúmenes que facilitan y simplifican el trabajo diario del personal hospitalario.
La información almacenada en el módulo Nutrición y Dietética será explotada por las unidades de enfermería, las gobernantas de cocina, y todo el personal de la cocina del hospital. Usuarios sin extensos conocimientos en Informática, podrán desarrollar y optimizar su trabajo de manera fácil y dinámica

Se realizó un estudio a los sistemas existentes que poseen un módulo Nutrición y Dietética teniendo en cuenta una posible reutilización de los mismos. La característica en que más se hizo énfasis fue en las funcionalidades que estos ofrecían, además de si constituyen software libres y si son aplicaciones web o de escritorio. Sin embargo, estos tienen como desventaja fundamental que son software propietario, eso significa que no son reutilizables, y de poder adquirirlos sería a altos precios. Además, algunos son aplicaciones de escritorio, lo que trae consigo dificultad en el control de acceso, que haya que instalar el programa en cada computadora que lo necesite y una difícil integración de los datos. Después de todo el análisis sobre los principales sistemas anteriormente planteados cabe la posibilidad de la creación de un sistema web propio, que no sea propietario, que sea multiplataforma e independiente de bases de datos además de que posea todas las funcionalidades necesarias para posibilitar un mejor funcionamiento en el área de Nutrición y Dietética de las instituciones hospitalarias.

1.3 Herramientas y tecnologías de desarrollo

El desempeño de un proyecto está antecedido siempre por la investigación de las tecnologías que se utilizan, así como las ventajas y desventajas que trae su aplicación. Esta investigación fue realizada, analizando las tecnologías existentes que se ajustan a la solución a desarrollar. El sistema debe estar libre de costos adicionales relativos a pago de licencias de software, debe ser adaptable a las reglas del negocio de cualquier institución hospitalaria, así como contar con la debida documentación para su mantenimiento y desarrollo. A partir de la investigación realizada se identificaron las siguientes tecnologías a utilizar en el desarrollo del sistema:

Java Server Faces (JSF)

Java Server Faces (JSF) es un framework para desarrollar aplicaciones web en el lenguaje Java, simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. JSF, usa Facelets como la tecnología que permite hacer el despliegue de las páginas, pero también se puede acomodar a otras tecnologías como XUL. (4)

El uso de JSF permite el manejo de estados y eventos, la asociación entre los datos de la interfaz y los datos de la aplicación web, la especificación de la navegación del usuario, entre otras funcionalidades. Además, posibilita la creación de nuevos componentes por el desarrollador, lo que le imprime un alto grado de flexibilidad.

RichFaces

RichFaces es una librería de componentes web enriquecidos, de código abierto y basada en el estándar JSF. Permite integrar fácilmente el código JavaScript asíncrono y XML (AJAX), utilizando para ello el framework Ajax4jsf. Provee facilidades de validación y conversión de los datos proporcionados por el usuario, administración avanzada de recursos como imágenes, código JavaScript y Hojas de Estilo en Cascada (CSS por sus siglas en inglés). Además, permite crear interfaces de usuario modernas de manera eficiente y rápida, basadas en componentes altamente configurables en cuanto a temas y esquemas de colores predefinidos por el propio framework o desarrollados a conveniencia. (5)

Ajax4jsf

Ajax4jsf es una librería de código abierto que se integra totalmente en la implementación de JSF, y extiende la funcionalidad de sus etiquetas dotándolas con tecnología Ajax sin código Javascript. Mediante este framework se puede variar el ciclo de vida de una petición JSF, recargar determinados componentes de la página sin necesidad de refrescarla en su totalidad, realizar peticiones al servidor de forma automática, controlar cualquier evento de usuario, etc. (6)

Facelets

Facelets es un framework para plantillas centrado en la tecnología Java Server Faces (JSF), permitiendo que Java Server Pages (JSP) y JSF puedan funcionar conjuntamente en una misma aplicación web, los cuales no se complementan naturalmente. Con este framework es posible diseñar de forma libre una

página web y luego asociarle los componentes JSF específicos; esto aporta mayor libertad al diseñador y mejora los informes de errores que tiene JSF entre otras cosas.

Lenguaje de Marcado de Hipertexto Extensible (XHTML)

XHTML es una versión más estricta y clara de HTML, que elimina sus limitaciones de uso con herramientas basadas en XML. XHTML combina la sintaxis de HTML 4.0, diseñado para mostrar datos, con la de XML, diseñado para describirlos. Además, permite una correcta interpretación de la información independientemente del dispositivo desde el que se accede a ella.

JBoss Seam

Es un framework de código abierto que enlaza diferentes tecnologías y estándares JAVA adicionando algunas funcionalidades no contempladas en *frameworks* anteriores. Esto garantiza la plena comunicación de los elementos de la capa de presentación, de negocio y de acceso a datos. Con *Seam* basta agregar anotaciones propias de éste a los objetos Entidad y *Session* de EJB, lo que permite escribir menos código Java y XML. Otra característica importante es que se puede hacer validaciones en los POJOs *(Plain Object Java)* y además manejar directamente la lógica de la aplicación y de negocios desde las *sessions bean.* Una de las principales ventajas del uso de *Seam* es que permite el control de sus componentes mediante anotaciones, lo cual reduce enormemente la cantidad de archivos XML de configuración. (7)

Algunos de los *framework*s con los que se integra son:

- *Java Server Faces (JSF)*: Para el desarrollo de la interfaz de usuario, en este sentido *Seam* contiene la librería *RichFaces/Ajax4JSf*, que reduce enormemente el esfuerzo de los programadores con los componentes web.
- *Hibernate*: Para el mapeo y el acceso con la base de datos.

Drools

Drools es una implementación del JSR 94 (Java Rule Engine API), especificación que define una interfaz común para un motor de reglas estándar dentro de la plataforma Java. Para definir las reglas emplea XML y permite adaptarse a la semántica de un determinado dominio definiendo un esquema que la represente. Uno de sus principales beneficios es que permite separar la lógica de negocio del código de la aplicación,

de esta manera, un cambio en el flujo del negocio no implica la realización de modificaciones en el código de la aplicación. (8)

Herramienta Hibernate

Hibernate es una capa de persistencia objeto/relacional. Permite diseñar objetos persistentes que podrán incluir polimorfismo, relaciones, colecciones, y un gran número de tipos de datos. De manera muy rápida y optimizada se podrá generar Bases de Datos en cualquiera de los entornos soportados: Oracle, DB2, MySQL, entre otras. Uno de los posibles procesos de desarrollo consiste en, una vez que se tenga el diseño de datos realizado, *mapear* este a ficheros XML a través del mapeo de Hibernate. Desde estos se puede generar el código de los objetos persistentes en las clases Java y también crear Bases de Datos, independientemente del entorno escogido. Es capaz de integrarse a cualquier tipo de aplicación.(9)

Java Persistence API (JPA)

Es la interfaz de programación de aplicaciones (API por sus siglas en inglés) de persistencia desarrollada para la plataforma Java EE y está incluida en el estándar EJB3. Esta API busca unificar la manera en que funcionan las utilidades que provee un mapeo objeto-relacional. El objetivo que persigue su diseño es no perder las ventajas de la orientación a objetos al interactuar con una base de datos, como sucedía con EJB2, y permitir usar objetos regulares conocidos como POJOs (Plain Old Java Object). (10)

Enterprise Java Bean (EJB 3)

EJB en su versión 3 proporciona un modelo distribuido y estándar de componentes que se ejecutan en el servidor. Su objetivo es dotar al programador de un modelo que le permita abstraerse de los problemas generales de una aplicación empresarial (concurrencia, transacciones, persistencia, seguridad, etc.), para centrarse en el desarrollo de la lógica de negocio en sí. El hecho de estar basado en componentes permite que éstos sean flexibles y sobre todo reutilizables. Incorpora el estándar JPA como el principal API de persistencia para aplicaciones EJB.

Java Platform Enterprise Edition 5 (JavaEE 5)

Java Platform Enterprise Edition o Java versión 5 es una plataforma de programación que parte de la Plataforma Java; para desarrollar y ejecutar software de aplicaciones en lenguaje de programación Java con arquitectura distribuida de n niveles. Se basa ampliamente en componentes de software modulares y se ejecuta sobre un servidor de aplicaciones. (11)

PosgtreSQL

PostgreSQL es un potente motor de bases de datos de código abierto que posee prestaciones y funcionalidades equivalentes a muchos gestores de bases de datos comerciales. Además de presentar una alta concurrencia que evita tener que bloquear una tabla en el momento que se está escribiendo sobre ella. También permite realizar copias de seguridad en línea, replicación asíncrona, transacciones anidadas y optimizador de consultas.

Permite la creación de procedimientos almacenados, restricciones de integridad y vistas, destacándose por su robustez, escalabilidad y cumplimiento de los estándares SQL. Cuenta con diversas versiones para sistemas operativos tales como: Linux, Windows, Mac OS X, Solaris, BSD, Tru64 entre otros, e incluye la mayor parte de los tipos de datos especificados en los estándares SQL92 y SQL99, como: entero, numérico, booleano, char, varchar, fecha, interval o timestamp.

Algunas de sus funciones incluyen replicación, respaldos pesados, optimizador avanzado, codificado de caracteres de *multibyt*e, y un tamaño de base de datos ilimitado. Del mismo modo es capaz de soportar Atomicidad, Consistencia, Aislamiento y Durabilidad (ACID acrónimo de Atomicity, Consistency, Isolation and Durability), lo que proporciona la realización de transacciones seguras, además de vistas, uniones, claves foráneas, procedimientos almacenados, *triggers*, entre otras cosas.

JBoss Application Server (JBoss AS)

JBoss AS es el servidor de aplicaciones de código abierto más ampliamente desarrollado del mercado. Se encuentra bajo la licencia GNU LGPL, por lo que puede libremente usarse sin costo alguno en cualquier aplicación comercial o ser redistribuido. Por ser una plataforma certificada JEE 5, soporta todas las especificaciones correspondientes, incluyendo servicios adicionales como clustering, caching y persistencia. JBoss es ideal para aplicaciones Java, pues está basado en este lenguaje de programación, y también es propicio para aplicaciones basadas en la web. También soporta Enterprise Java Beans (EJB) 3.0. Los componentes claves son: JBoss AS 4.2, Hibernate 3.2.4, Seam 2.0.

Eclipse

Es un entorno de desarrollo integrado el cual se caracteriza por presentar código abierto, por su portabilidad y también por ser multiplataforma. Este fue diseñado originalmente por la empresa IBM y actualmente es desarrollado por la Fundación Eclipse, una comunidad de código abierto.

Eclipse trabaja principalmente a base de módulos o *plugins*, lo cual hace posible el trabajo en variados lenguajes de programación como son Java, C++, PHP, Perl. Permite la integración con la herramienta Visual Paradigm a través del Entorno de Desarrollo Inteligente *(Smart Development Environment),* más conocido por sus siglas en inglés: SDE Enterprise Edition, lo que proporciona un mejor entendimiento entre analistas y desarrolladores.

JBoss Server

JBoss Application Server es el servidor de aplicaciones J2EE, presenta código abierto implementado completamente en Java, lo que implica que puede ser utilizado en cualquier sistema operativo que lo soporte. Por ser una plataforma certificada J2EE, soporta todas las funcionalidades de J2EE 1.4, incluyendo servicios adicionales como *clustering, caching* y persistencia. JBoss es ideal para aplicaciones web en Java. También soporta *Enterprise Java Beans* (EJB) 3.0, y esto hace que el desarrollo de las aplicaciones empresariales sea mucho más simple. Posee la ventaja de ser compatible con la versión 2.1 del *framework* Seam.

JBoss Tools

Jboss Tools que es un conjunto de herramientas para Eclipse. Entre estas herramientas, este dispone de *plugins* que proporcionan soporte en Eclipse para Hibernate, JBoss AS, Drools, JSF, (X) HTML, Seam, entre otros. (12)

Algunos de estos *plugins* son:

RichFaces VE: Editor visual para componentes HTML, JSF y RichFaces

Seam tools: Incluye soporte para la integración de los componentes del *framework* Seam.

Hibernate tools: Sirve de apoyo para la utilización de los componentes del *framework* Hibernate y para el mapeo con la base de datos.

Java Runtime Environment 6 (JRE 6)

JRE es el entorno en tiempo de ejecución Java y se corresponde con un conjunto de utilidades que permite la ejecución de programas java sobre todas las plataformas soportadas. JVM (máquina virtual Java) es una instancia de JRE en tiempo de ejecución. Este interpreta el código Java y está compuesto

además por las librerías de clases que implementan el API de Java. Ambas, JVM y API, deben ser consistentes entre sí, de ahí que sean distribuidas de modo conjunto.

JasperReport

JasperReport se conoce por ser una biblioteca de clases de generación de informes, la cual es libre y está descrita en Java. Se encarga de generar archivos XML donde se recogen las particularidades del informe La salida de este archivo puede ser un PDF, XML, HTML, CSV, XLS, RTF, TXT una vez tratada por las clases Jasper. Una de las ventajas de JasperReport es que se integra fácilmente con la librería de JFreeChart para todo tipo de gráficos

iReport

La herramienta iReport es un constructor / diseñador de informes visual, poderoso, intuitivo y fácil de usar para JasperReports escrito en Java. Este instrumento permite que los usuarios corrijan visualmente informes complejos con cartas, imágenes, subinformes, etc. iReport está además integrado con JFreeChart, una de la biblioteca gráficas OpenSource más difundida para Java. Los datos para imprimir pueden ser recuperados por varios caminos incluso múltiples uniones JDBC, TableModels, JavaBeans, XML, etc.

iReport presenta una serie de características entre las cuales se encuentra que está escrito 100% en JAVA y además OPENSOURCE y gratuito, maneja el 98% de las etiquetas de JasperReport, permite diseñar con sus propias herramientas: rectángulos, líneas, elipses, campos de los textfields, cartas, subreports (subreportes), soporta internacionalización nativamente, browser de la estructura del documento, soporta JDBC, tiene asistentes para generar los subreportes, tiene asistentes para las plantillas y facilidad de instalación. (13)

1.4 Lenguaje de programación

El lenguaje seleccionado para el desarrollo de este proyecto de software es Java, este es orientado a objetos, de una plataforma independiente, permite el desarrollo de aplicaciones bajo el esquema de Cliente Servidor, como de aplicaciones distribuidas, lo que lo hace capaz de conectar dos o más computadoras u ordenadores, ejecutando tareas simultáneamente, y de esta forma logra distribuir el trabajo a realizar. El lenguaje seleccionado, es compilado en la medida en que su código fuente se

transforma en una especie de código máquina, los bytecodes, semejantes a las instrucciones de ensamblador. (14)

Por otra parte, es interpretado, ya que los *bytecodes* se pueden ejecutar directamente sobre cualquier máquina a la cual se hayan portado el intérprete y el sistema de ejecución en tiempo real (run-time). Es indiferente de arquitectura, está diseñado para soportar aplicaciones que serán ejecutadas en los más variados entornos de red, desde Unix a Windows Nt, pasando por Mac y estaciones de trabajo, sobre arquitecturas distintas y con sistemas operativos diversos. (15)

1.5 Metodología de desarrollo

Proceso Unificado Racional (RUP)

Una buena metodología de software pretende reducir costos y retrasos de proyectos, así como mejorar la calidad del software. La metodología de desarrollo cobra gran importancia en proyectos empresariales, pues al no utilizarla adecuadamente se puede desembocar en la frustración del equipo de desarrollo y en la insatisfacción de los clientes. Por tanto el uso de una metodología es necesario para controlar el ciclo de vida de un proyecto. Una de las preocupaciones actuales de los analistas y desarrolladores de software es la necesidad de hacer productos de software con mayor flexibilidad a los continuos cambios y adaptables a diferentes entornos de desarrollo. Esta necesidad repercute muchas veces desfavorablemente en el tiempo de desarrollo, costo y esfuerzo de los equipos de trabajo.

En aras de dar solución a esta problemática y luego de un estudio de las metodologías existentes para el desarrollo de software se opta por la utilización del Proceso Unificado Racional (RUP), el cual de conjunto con el Lenguaje Unificado de Modelado (UML) constituyen una de las metodologías más utilizadas para el desarrollo de grandes productos software. RUP facilita el uso de diferentes artefactos para la representación gráfica de los procesos de negocio, requerimientos, diseño e implementación del sistema a desarrollar. Constituye una guía para el equipo de desarrollo delimitando los roles encargados de elaborar los diferentes artefactos en cada fase y disciplina, además facilita la manera de realizar cada uno de estos elementos.

A pesar de las facilidades que brinda RUP en la actualidad ha aparecido una tendencia a utilizar notaciones para la descripción y especificación de procesos de negocio. Quizás la más conocida y empleada sea la Notación para el modelado de procesos de negocio BPMN.

Notación para el modelado de procesos de Negocio

La notación para el modelado de procesos de negocio (BPMN por sus siglas en inglés) es una notación gráfica que describe la lógica de los pasos de un proceso de Negocio. Esta notación ha sido especialmente diseñada para coordinar la secuencia de los procesos y los mensajes que fluyen entre los participantes de las diferentes actividades.

BPMN es un estándar internacional de modelado de procesos aceptado por la comunidad, es independiente de cualquier metodología de modelado de procesos. Además crea un puente estandarizado para disminuir la brecha entre los procesos de negocio y la implementación de estos.

Otra de sus facilidades es que permite modelar los procesos de una manera unificada y estandarizada permitiendo un entendimiento a todas las personas de una organización, proporcionando un lenguaje común para que las partes involucradas puedan comunicar los procesos de forma clara, completa y eficiente.(16)

Lenguaje Unificado de Modelado (UML)

El Lenguaje de Modelado Unificado (UML: Unified Modeling Language) es el más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocio y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

Entre las ventajas de UML figuran: lenguaje conocido, con enfoque orientado a objetos, estándar y fácil de aprender, además de que permite el modelado de sistemas complejos, pues mientras más complejo es el sistema que se desea crear más beneficios presenta el uso de UML. Sus principales desventajas son: no ha sido diseñado para modelar procesos de negocios, no está orientado al dominio del problema, sólo lo conocen los expertos en tecnología de la Información, UML no tiene todavía una semántica formal.

En el presente capítulo se realizó un estudio de los principales sistemas que gestionan la información del área de Nutrición y Dietética, se llegó a la conclusión de que la mayoría están desarrollados sobre software propietario con altos costos de licencia. Por otra parte, no garantizan la gestión de la información relacionada con diferentes servicios de apoyo de una institución hospitalaria. De igual forma fueron analizados diferentes herramientas, metodologías y tecnologías que constituyen la propuesta tecnológica para el desarrollo del módulo Nutrición y Dietética para el Sistema de Información Hospitalaria alas HIS.

CAPÍTULO 2: DESCRIPCIÓN DE LA ARQUITECTURA

A continuación se describen los requerimientos no funcionales identificados en el transcurso de la investigación y resulta imprescindible que posea el sistema que se desarrollará. Además se explica la arquitectura definida por el Departamento de Sistemas de Gestión Hospitalaria, abordándose otros temas de interés, como la forma en que se manipulará la seguridad del sistema.

2.1. Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. A través de estos se especifican propiedades del sistema como restricciones de ambiente y desarrollo, rendimiento, dependencias de plataformas y mantenimiento. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable. (17)

Usabilidad

El sistema estará diseñado de manera que los usuarios adquieran las habilidades necesarias para explotarlo en un tiempo reducido:

- Para alcanzar un nivel Elemental asociado al dominio del sistema y el uso eficiente del mismo, serán necesarios 20 días de preparación, obteniendo la categoría de Usuarios normales.
- Para alcanzar un nivel Avanzado asociado al dominio del sistema y el uso eficiente del mismo, serán necesarios 30 días de preparación, obteniendo la categoría de Usuarios avanzados.
- La estructura concebida para la organización de la información agiliza el entendimiento del sistema por parte del usuario.
- Los usuarios serán capaces de alcanzar sus objetivos con un mínimo esfuerzo y obteniendo los resultados máximos.
- El sistema será capaz de solucionar un error cometido por el usuario o sugerir las posibles soluciones, indicándole al usuario las acciones pertinentes a seguir.
- Brindará comodidad a la hora de acceder a las diferentes funcionalidades que proporciona la aplicación mediante teclas de acceso rápido. Serán reutilizados diseños de aplicaciones comúnmente usada por los usuarios finales con vistas a aprovechar la experiencia de usuario.

Fiabilidad

- En los servidores de los hospitales y los Centro de Datos se garantizará una arquitectura de máxima disponibilidad, tanto de servidores de aplicación como de base de datos. Se garantizarán además, políticas de respaldo a toda la información, evitando pérdidas en caso de desastres ajenos al sistema.
- Las informaciones médicas relacionadas con los pacientes y que vayan a ser intercambiadas con otros hospitales por la red pública, viajarán cifradas para evitar accesos o modificaciones no autorizadas.
- Se mantendrá seguridad y control a nivel de usuario, garantizando el acceso de los mismos sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando sólo la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión.
- Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude. Los mecanismos serán capaces de informar al personal autorizado sobre posibles irregularidades que den indicios sobre la introducción de información falseada.
- El sistema implementará un mecanismo de auditoría para el registro de todos los accesos efectuados por los usuarios, proporcionando un registro de actividades (log) de cada usuario en el sistema.
- El sistema implementará un control de cambios a determinados campos de información (seleccionados por su importancia), de forma tal que sea posible determinar cuáles han sido las actualizaciones que se le han realizado.
- Ninguna información que se haya ingresado en el sistema será eliminada físicamente de la BD, independientemente de que para el sistema, este elemento ya no exista.

- El sistema permitirá la recuperación de la información de la base de datos a partir de los respaldos o salvas realizadas.

Eficiencia

- El Centro de Datos permitirá agregar recursos para aumentar el poder de procesamiento y almacenamiento sin afectar los sistemas, garantizando expansiones motivadas por futuros requerimientos.
- El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria. Para ello se potenciará como regla guardar en la memoria caché datos y recursos de alta demanda.

Soporte

- Se permitirá la creación de usuarios, otorgamiento de privilegios y roles, asignación de perfiles y activación de permisos por direcciones IP. Además, la administración remota, monitoreo del funcionamiento del sistema en los centros hospitalarios y detección de fallas de comunicación. Realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados. Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema. Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

Seguridad de acceso y administración de usuarios

- Se mantendrá seguridad y control a nivel de usuario, garantizando su acceso sólo a los niveles establecidos de acuerdo a la función que realizan. Las contraseñas podrán cambiarse solo por el propio usuario o por el administrador del sistema.
- Se mantendrá un segundo nivel de seguridad a nivel de estaciones de trabajo, garantizando únicamente la ejecución de las aplicaciones que hayan sido definidas para la estación en cuestión. Se registrarán todas las acciones que se realizan, llevando el control de las actividades de cada usuario en todo momento.
- Se establecerán mecanismos de control y verificación para los procesos susceptibles de fraude.

- El sistema proporcionará un registro de actividades de cada usuario.

Respaldo y recuperación de base de datos

- Se permitirá realizar copias de seguridad de la base de datos hacia otro dispositivo de almacenamiento externo, además de recuperar la base de datos a partir de los respaldos realizados.

Auditoría

- Se permitirá el chequeo de las operaciones y acceso de los usuarios al sistema, para esto debe existir un registro de trazas que almacene todas las transacciones realizadas en el sistema, indicando para cada caso como mínimo: usuario que realizó la transacción, tipo de operación que se realizó, fecha y hora en que se realizó la operación e información contenida en el registro modificado.

Configuración de parámetros

- Se permitirá establecer parámetros de configuración del sistema y actualización de nomencladores.

Réplica

- Se permitirá realizar réplica de la base de datos de los hospitales con el Centro de Datos. Esta réplica se podrá hacer de forma manual y automatizada a través de la red.

Documentación de usuarios en línea y ayuda del sistema.

- Se posibilitará el uso de ayudas dinámicas y tutoriales en línea sobre el funcionamiento del sistema.

Interfaz

1.1-Interfaces de usuario

Las ventanas del sistema contendrán los datos claros y bien estructurados, además de permitir la interpretación correcta de la información. La interfaz contará con teclas de función y menús desplegables

que faciliten y aceleren su utilización. La entrada de datos incorrecta será detectada claramente e informada al usuario. Todos los textos y mensajes en pantalla aparecerán en idioma español.

Rendimiento

- El sistema minimizará el volumen de datos en las peticiones y además optimizará el uso de recursos críticos como la memoria.
- El sistema respetará buenas prácticas de programación para incrementar el rendimiento en operaciones costosas para la máquina virtual como la creación de objetos.

Hardware

1.1-Estaciones de trabajo

En la solución se incluyen estaciones de trabajo para las consultas del Sistema de Información Hospitalaria alas HIS, las que necesitan capacidad de hardware que soporte un sistema operativo que cuente con un navegador actualizado y que siga los estándares web, se recomienda IE 7, Firefox 2 o versiones superiores. Por lo que se escogieron estaciones de trabajo de 256 Mb de memoria RAM y un microprocesador de 2.0 Hz con sistema operativo Linux.

1.2-Servidores

La solución estará conformada, fundamentalmente, por servidores de alta capacidad de procesamiento y redundancia, que permitan garantizar movilidad y residencia de la información y las aplicaciones bajo esquemas seguros y confiables. Servidores de base de datos: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Aplicaciones: 2 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 4GB de memoria y 2x72GB de disco y sistema operativo Linux. Servidores de Intercambio: 1 DL380 G5, Procesador Intel® Xeon® 5140 Dual-Core 2 GB de memoria y 2x72GB de disco y sistema operativo Linux.

Software

El sistema debe correr en sistemas operativos Windows, Unix y Linux, utilizando la plataforma JAVA (Java Virtual Machine, JBoss AS y PostgreSQL). El sistema deberá disponer de un navegador web, estos pueden ser IE 7, Opera 9, Google chrome 1 y Firefox 2 o versiones superiores de estos.

2.2. Descripción de la arquitectura, fundamentación

La Arquitectura de Software es la organización fundamental de un sistema encarnada en sus componentes, las relaciones entre ellos y el ambiente y los principios que orientan su diseño y evolución.

A grandes rasgos es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se le percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema.

En la creación del sistema de Nutrición y Dietética se utiliza la arquitectura basada en el patrón Modelo Vista Controlador, describe una forma, muy utilizada en la Web, de organizar el código de una aplicación. Para el diseño de aplicaciones con sofisticadas interfaces se utiliza este patrón de diseño. Si no se realiza un diseño correcto, o sea, si se realiza un diseño que mezcle los componentes de interfaz y de negocio, la consecuencia será que, cuando se necesite cambiar la interfaz, se tendrá que modificar trabajosamente los componentes de negocio. Esto conllevaría a realizar un mayor trabajo y con más riesgo de cometer errores.

Se trata de realizar un diseño que desacople la vista del modelo, con la finalidad de mejorar la reusabilidad. De esta forma las modificaciones en las vistas impactan en menor medida en la lógica de negocio o de datos.

Elementos del patrón:

Modelo: Esta sirve como representación específica de toda la información con la cual el sistema va a trabajar. La lógica de datos puede llegar a asegurar la integridad de ellos y permitir derivar nuevos datos.

Vista: Presenta el modelo con el que va a interactuar el usuario, más conocida como interfaz.

Controlador: Responde más bien a eventos, normalmente son acciones que el usuario invoca, implica cambios en el modelo y también en la vista (interfaz). (18)

La capa de presentación está desarrollada básicamente con JSF, usando la librería de componentes Richfaces 3.2.0 G.A., la cual se complementa fácilmente con el framework de integración Seam y permite generar vistas no necesariamente basadas en HTML (PDF, etc.), adiciona además controles out-of-the-box AJAX-ready y el framework de extensión AJAX para los controles JSF básicos Ajax4Jsf. Incluyendo

conversión y validación de campos, establecimiento de reglas de navegación declarativas, la internacionalización y accesibilidad de la interfaz de usuario, además de un modelo orientado a eventos y combinado con Facelets, además de que brinda la capacidad añadida de la tecnología de plantillas de Facelets.

Seam es un poderoso y moderno framework creado para unificar todas las tecnologías estándares JSF, EJB3, JPA y Drools como motor de reglas de negocio. Para el acceso a datos se usa la implementación de JPA de Hibernate 3.3, minimizando por un lado las configuraciones en XML sin chequeo de tipos y por otro lado usando los servicios del contenedor de EJB3 y/o los contextos de persistencias administrados por Seam, se elimina gran parte del código “infraestructural” en cuanto a transacciones, la transmisión del contexto de persistencia, etc., permitiendo además que se puedan establecer validaciones end-to-end gracias a los Hibernate Validators.

2.3. Análisis de posibles implementaciones, componentes o módulos que puedan ser reutilizados. Estrategias de integración

La reutilización de componentes de software es un proceso inspirado en la manera en que se producen y ensamblan componentes en la ingeniería de sistemas físicos. La aplicación de este concepto al desarrollo de software no es nueva. La misma es de gran importancia en el proceso de desarrollo de software; particularmente, en tres de las variables más importantes de la Ingeniería de Software: el costo, tiempo y esfuerzo requerido para desarrollar un producto de software. Otros beneficios importantes son el incremento de la calidad del software producido, el aumento de la productividad de los grupos de desarrollo y la reducción del riesgo global del proyecto.

El Sistema de Información Hospitalaria alas HIS, se encuentra en desarrollo, está compuesto por diferentes módulos que comparten componentes entre sí. Para la implementación de los procesos enmarcados en el campo de acción de la presente investigación, se identificaron los componentes a reutilizar a partir de un estudio realizado a las funcionalidades de los módulos relacionados con la atención al paciente.

El proceso de selección de pacientes reutilizó del módulo Hospitalización la funcionalidad Listar pacientes y camas, la cual permite listar todos aquellos pacientes que se encuentran hospitalizados. Cuando a los pacientes se les va a crear la ficha de evaluación nutricional se verifica que tengan los resultados de los

exámenes de laboratorio en caso de que no los tenga se le hace la solicitud, por lo que del módulo Laboratorio se reutilizan las funcionalidades: Gestión de la solicitud de exámenes y consultar resultados de exámenes.

Por otra parte, existen algunos componentes que se definen de manera general para que puedan utilizarlos todos los módulos del sistema alas HIS, los mismos brindan un conjunto de facilidades y simplifican el trabajo de los desarrolladores. Dentro de estos están: la clase Bitácora para el control de las trazas de todas las acciones que se realizan con la aplicación; la clase UserTools para saber qué usuario está trabajando con la aplicación en tiempo real; la clase ActiveModule para conocer qué módulo está activo y en qué entidad.

2.4. Seguridad

Si se quiere lograr la seguridad del sistema se deben garantizar los tres principios básicos de la Seguridad Informática: la confidencialidad, integridad y disponibilidad de la información. Para ello se debe llevar un control estricto de la misma, garantizar que no sea accedida ni modificada por personal no autorizado. Entre las diferentes características que debe cumplir el software para que sea seguro está la de contar con diferentes secciones, las cuales les permiten a los usuarios acceder a sus permisos establecidos, además de registrar trazas para ser verificadas por el administrador en caso de ser necesario.

Por otra parte otro de los aspectos a tener en cuenta es la fidelidad de los datos, esta se lograr mediante el cifrado de la información proveniente de la comunicación entre los componentes internos del sistema y otros sistemas que soliciten información desde otra Institución Hospitalaria. Todo esto impide la lectura o modificación de los datos confidenciales que se manejan, contribuyendo al aumento de la seguridad del sistema

2.5. Vista de Despliegue

La vista de despliegue permite representar la distribución física del sistema, así como los distintos nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución por ejemplo: servidores, computadoras cliente, además dispositivos tales como impresoras y scanners. Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema, el mismo muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del

software se trazan en esos nodos. A continuación se muestra el Diagrama de Despliegue correspondiente al módulo de Nutrición y Dietética.

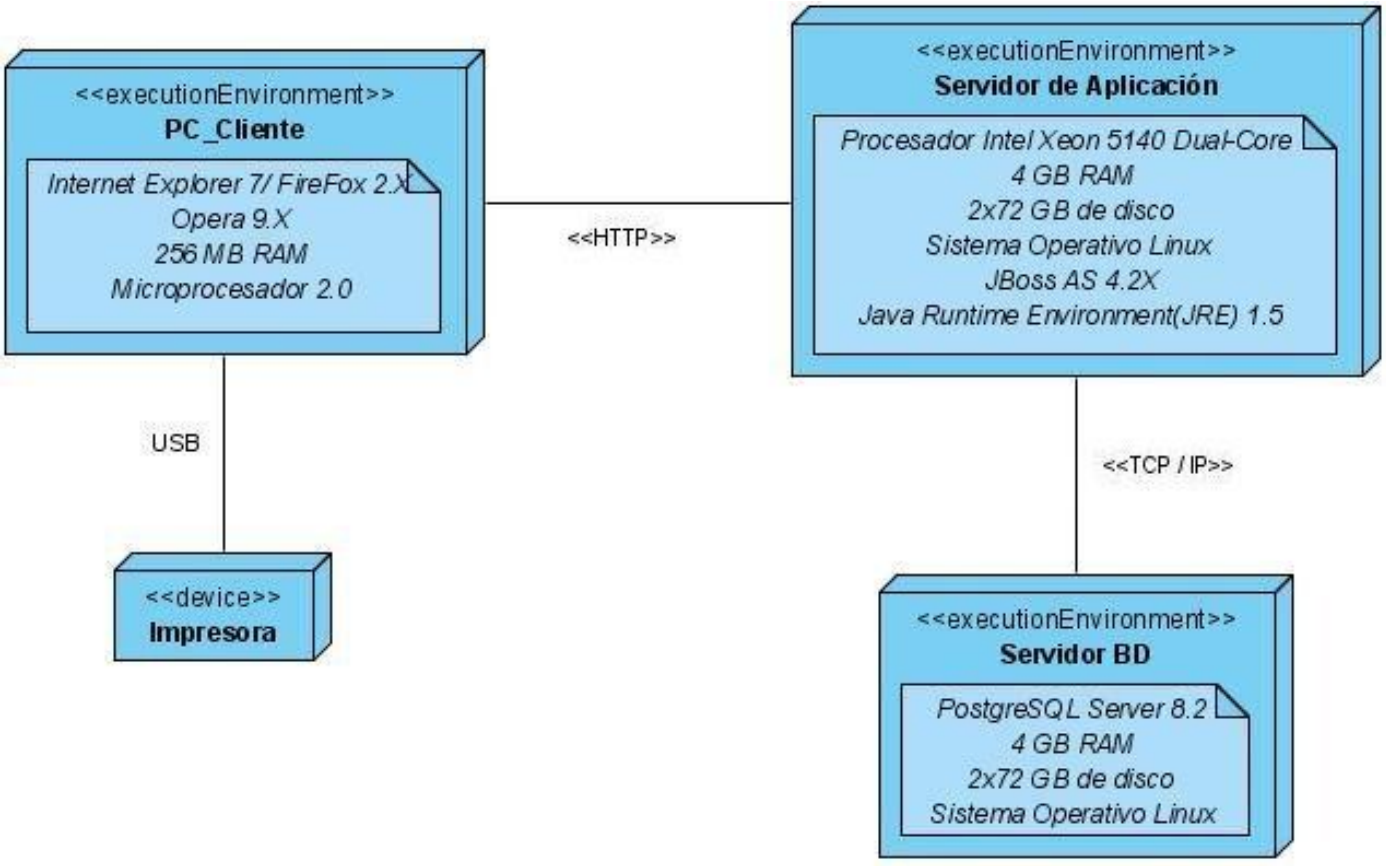


Figura 2.5 Diagrama de despliegue

2.6. Estrategias de codificación. Estándares y estilos a utilizar

Los estándares de codificación son un conjunto de directrices que especifican cómo debe escribirse el código fuente. Algunos ejemplos de directrices que se encuentran son aquellas que pautan el nombrado de variables, clases y/o paquetes; la correcta indentación del código, cómo escribir los bucles y estructuras de control o incluso qué información incluir en los comentarios. Su importancia radica en que facilitan la compresión del código y, por tanto, permiten que los diferentes desarrolladores puedan tratar cada porción del mismo como si de su propio código se tratara.

Variables y constantes

1.1-Apariencia de constantes

Todas sus letras en mayúscula: Se deben declarar las constantes con todas sus letras en mayúscula.

1.2-Aspectos generales

Nombres de las variables y constantes: El nombre empleado, debe permitir que con sólo leerlo se conozca el propósito de la misma.

Indentación

- **Inicio y fin de bloque**

  Se recomienda dejar dos espacios en blanco desde la instrucción anterior para el inicio y fin de bloque {}. Lo mismo sucede para el caso de las instrucciones if, else, for, while, do while, switch, foreach.

- **Aspectos generales**

  - ✓ El indentado debe ser de dos espacios por bloque de código. No se debe usar el tabulador; ya que este puede variar según la computadora o la configuración de dicha tecla.
  - ✓ Los inicios ({) y cierre (}) de ámbito debe estar alineados debajo de la declaración a la que pertenecen y deben evitarse si hay sólo una instrucción.
  - ✓ Nunca colocar llave ({) en la línea de un código cualquiera, esto requiere una línea propia.

Comentarios, separadores, líneas, espacios en blanco y márgenes

- **Ubicación de comentarios**

  Al inicio de cada clase o función y al final de cada bloque de código: Se recomienda comentar al inicio de la clase o función especificando el objetivo de la misma así como los parámetros que usa (especificar tipos de dato, y objetivo del parámetro) entre otras cosas.

- **Líneas en blanco**

  Se emplean antes y después de métodos, clases y estructuras: Se recomienda dejar una línea en blanco antes y después de la declaración de una clase o de una estructura y de la implementación de una función.

- **Espacios en blanco**

  Entre operadores lógicos y aritméticos: Se recomienda usar espacios en blanco entre estos operadores para lograr una mayor legibilidad en el código. Ejemplo: producto = nomproducto

- **Aspectos generales**

  - ✓ Sobre el comentario: Se debe evitar comentar cada línea de código. Cuando el comentario se aplica a un grupo de instrucciones debe estar seguido de una línea en blanco. En caso de que se necesite comentar una sola instrucción se suprime la línea en blanco o se escribe a continuación de la instrucción
  - ✓ Sobre los espacios en blanco: No se debe usar espacio en blanco: Después del corchete abierto y antes del cerrado de un arreglo. Después del paréntesis abierto y antes del cerrado. Antes de un punto y coma.

Clases y Objetos

Apariencia de clases y objetos

Primera letra en mayúscula: Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto en minúscula, en caso de que sea un nombre compuesto se empleará notación Pascal Ejemplo: MiClase (). Para el caso de las instancias se comenzará con un prefijo que identificará el tipo de dato, este se escribirá en minúscula.

Apariencia de atributos

Primera letra en minúscula: El nombre que se le da a los atributos de las clases debe comenzar con la primera letra en minúscula, la cual estará en correspondencia al tipo de dato al que se refiere, en caso de que sea un nombre compuesto se empleará notación Camello.

Declaración de parámetro en funciones

Agrupados por tipos. Poner los string 1 numéricos 2, además, agrupar según valores por defecto: Los parámetros que se le pasan a las funciones se recomienda sean declarados de forma tal que estén agrupados por el tipo de dato que contienen, especificando el tipo de datos.

Aspectos generales

Sobre las clases, los objetos, los atributos y las funciones: El nombre empleado para las clases, objetos, atributos y funciones debe permitir que con sólo leerlo se conozca el propósito de los mismos.

Bases de datos, tablas, esquemas y campos

- **Apariencia de la base de datos**

  Las 2 primeras letras representan el tipo: Los nombres de las Bases de Datos deben comenzar con el prefijo bd a continuación guión bajo y luego el nombre completamente en minúscula, en caso de que sea un nombre compuesto se empleará notación. Los nombres serán cortos y descriptivos.

- **Apariencia de las tablas**

  Las 2 primeras letras representan el tipo. Todas las letras en minúscula: El nombre a emplear para las tablas debe comenzar con el prefijo tb seguido de guión bajo y luego debe escribirse todas las letras en minúscula, en caso de que sea un nombre compuesto se utilizará guión bajo para separarlo.

- **Tablas que representen Relaciones**

  Las 2 primeras letras representan el tipo. Todas las letras en minúscula: El nombre a emplear para estas tablas de relación, debe comenzar con el prefijo tr seguido de guión bajo y el nombre de la concatenación de las dos tablas que se generaron separadas por guión bajo todo en minúscula.

- **Tablas que representen nomencladores**

  Las 2 primeras letras representan el tipo. Todas las letras en minúscula: El nombre a emplear para estas tablas de relación debe comenzar con el prefijo tn seguido de guión bajo. El nombre será corto y descriptivo, todo en minúscula.

- **Apariencia de los campos**

  Todas las letras en minúscula: El nombre a emplear para los campos debe escribirse con todas las letras en minúscula para evitar problemas con el Case Sensitive del gestor.

- **Nombre de los campos**

  En caso de identificadores: Todos los campos identificadores van a comenzar con el identificador id seguido de guión bajo y posteriormente el nombre del campo.

- **Sentencias SQL**

  Todas las letras en mayúscula: Las palabras correspondientes a las sentencias SQL y sus parámetros deben ir en mayúsculas.

- **Aspectos generales**

  Sobre las BD, vistas, tablas atributos y procedimientos: El nombre empleado para las Bases de Datos, las vistas, las tablas, los campos y los procedimientos almacenados, deben permitir que con sólo leerlos se conozca el propósito de los mismos.

Controles

- **Apariencia de los controles**

  Los controles tendrán un prefijo para el tipo de datos en minúscula: El nombre que se le da a los controles deben comenzar con las primeras letras en minúscula, las cuales identificarán el tipo de datos al que se refiere (ver tabla 2.6.2), en caso de que sea un nombre compuesto se empleará notación CamellCasing**.

Tipo Datos	Prefijo	Ejemplo
Int	i	iCantPacientes
double	d	dPesoCarro
bool	b	bPacienteActivo
string	s	sNombrePaciente
char	c	cLetra
De tipo enum	e	eSexo
sbyte	sb	sbEdadPaciente
short	sh	shVariableShort
uint	ui	uiVariableUint
long	l	lVariableLong

Tabla 2.6.1 Tipo Datos

Control	Prefijo	Ejemplo
Botón	btn	btnAceptar
Etiqueta	lbl	lblNombre

Lista/Menú	mn	mnPrincipal
Campo de Texto	txt	txtFecha
Botón de Opción	bpt	optSexo
Casilla de Verificación	chx	chxBorrar
Casilla de Selección	cbx	cbxSexo

Tabla 2.6.2 Control

En este capítulo se analizaron la concepción arquitectónica y la seguridad del sistema. Además se describieron los requerimientos no funcionales que deben cumplirse y se ha proporcionado una breve descripción de los pasos a seguir para obtener un producto que sea lo más seguro, robusto y confiable posible. De igual forma, en el diagrama de despliegue ha quedado definida la propuesta de distribución física del sistema. Al mismo tiempo se ha mostrado el estándar de codificación con el objetivo de proporcionar una mayor comprensión del software y posibilitar su futuro mantenimiento.

Capítulo 3. Descripción y análisis de la solución propuesta

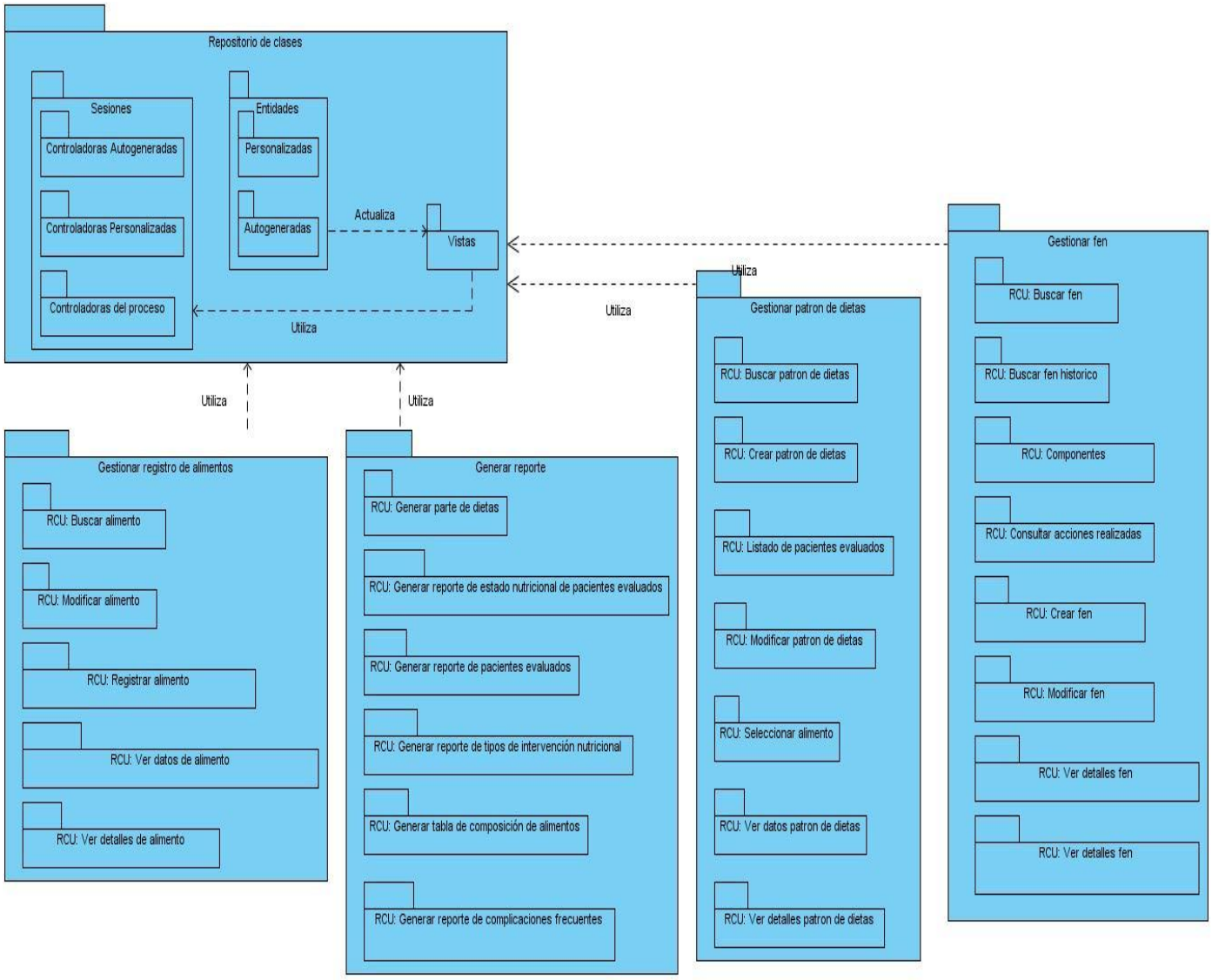
En este capítulo se profundiza de manera bien detallada el análisis y diseño del sistema. También se presentan los artefactos generados una vez analizado el modelo de datos, además de la vista de implementación.

3.1 Valoración crítica del diseño propuesto por el analista.

En esta fase se realiza el diseño de datos, a su vez, se realiza el diseño arquitectónico y se definen todas las relaciones que existen entre cada uno de los elementos estructurales del sistema. Seguidamente se realiza el diseño de la interfaz, se describe la comunicación del software consigo mismo, con los usuarios que lo emplean y a su vez con sistemas que trabajan junto con él. Se diseñan procedimientos y a su vez se transforman elementos estructurales de la arquitectura del programa.

El Diseño es imprescindible para materializar con exactitud las exigencias del cliente, lo que conlleva que se realice con buena calidad el proyecto. El proceso de diseño es una secuencia de pasos que permiten al diseñador describir todos los aspectos del sistema a construir. Esta fase es la que debe suministrar completamente la idea de lo que es el software, enfilando los dominios de datos y el comportamiento desde el punto de vista de la Implementación.

Por otra parte se encarga de modelar el sistema para que sustente todos los requisitos funcionales y no funcionales. En este modelo, los casos de uso son ejecutados por las clases del diseño, mediante los cuales se estructura el diagrama de clases del diseño. Este expone un conjunto de interfaces, colaboraciones y relaciones entre las clases definidas para dar solución a cada funcionalidad identificada, además detalla la estructura de clases del sistema y las relaciones entre las mismas.



3.1.1 Diagrama de paquetes.

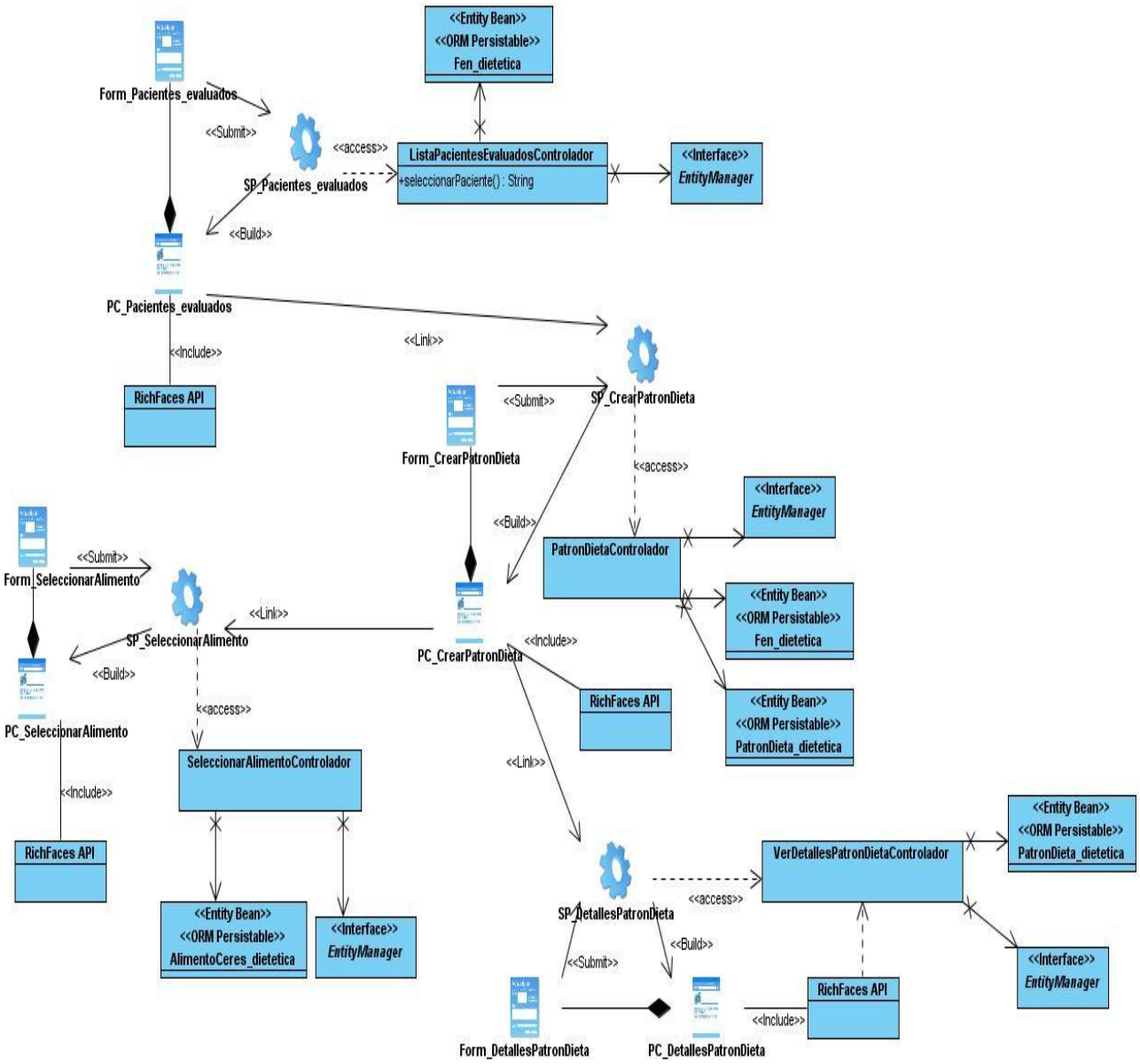


Figura 3.1.2 Diagrama de Clases del Diseño – Gestionar patrón de dietas.

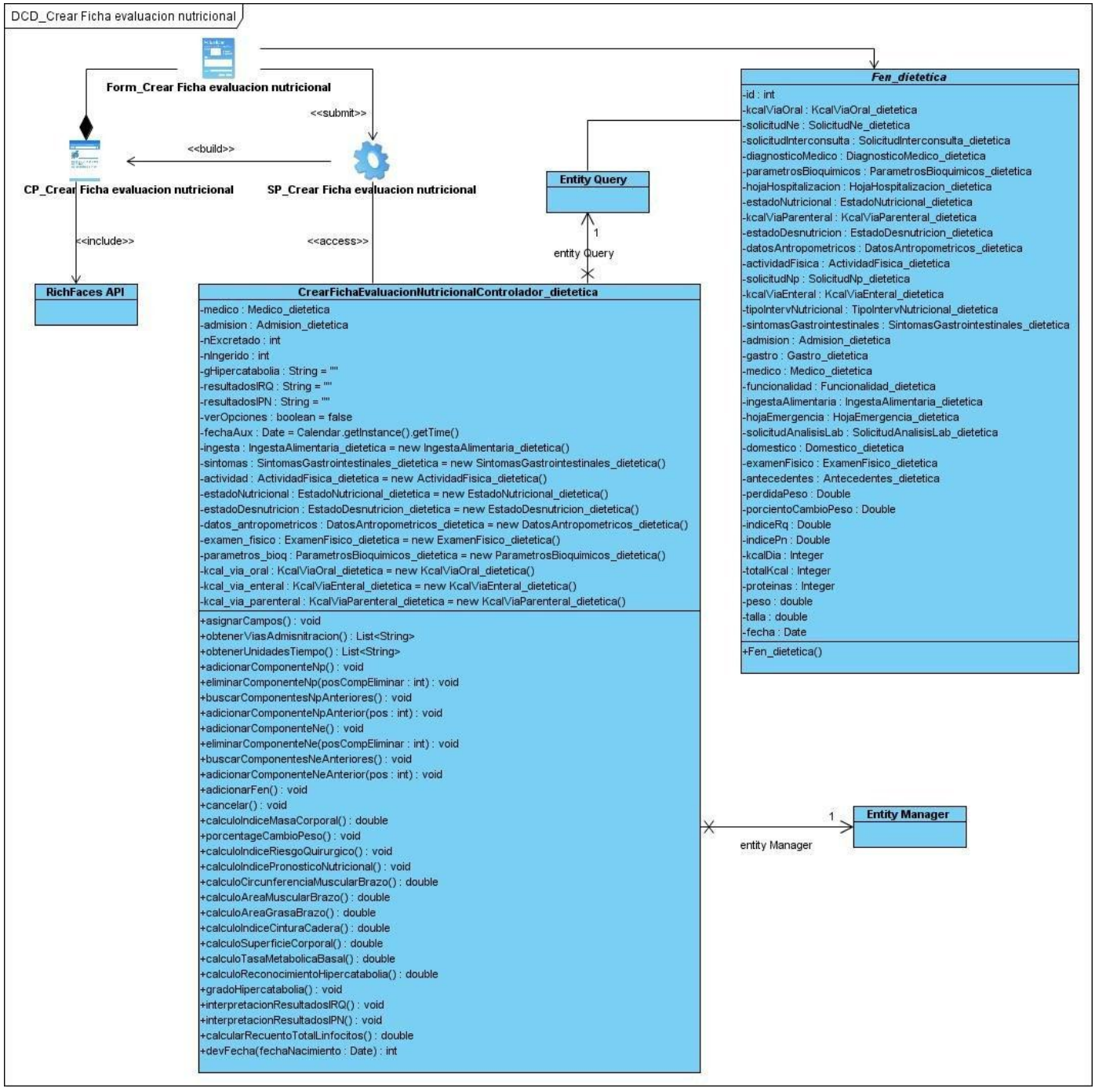


Figura 3.1.3 Diagrama de Clases del Diseño – Crear ficha de evaluación nutricional

.

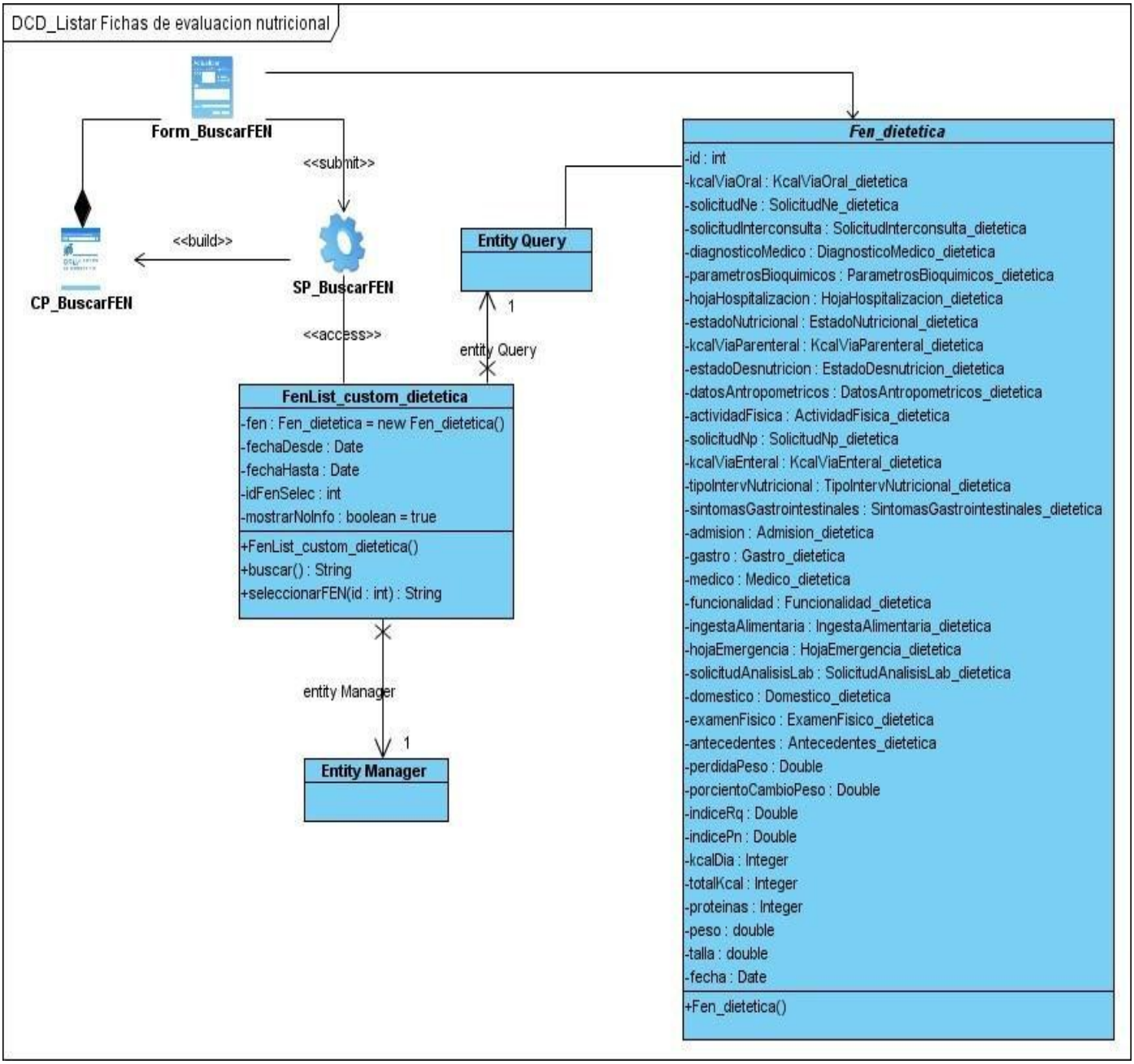


Figura 3.1.4 Diagrama de Clases del Diseño – Listar fichas de evaluación nutricional.

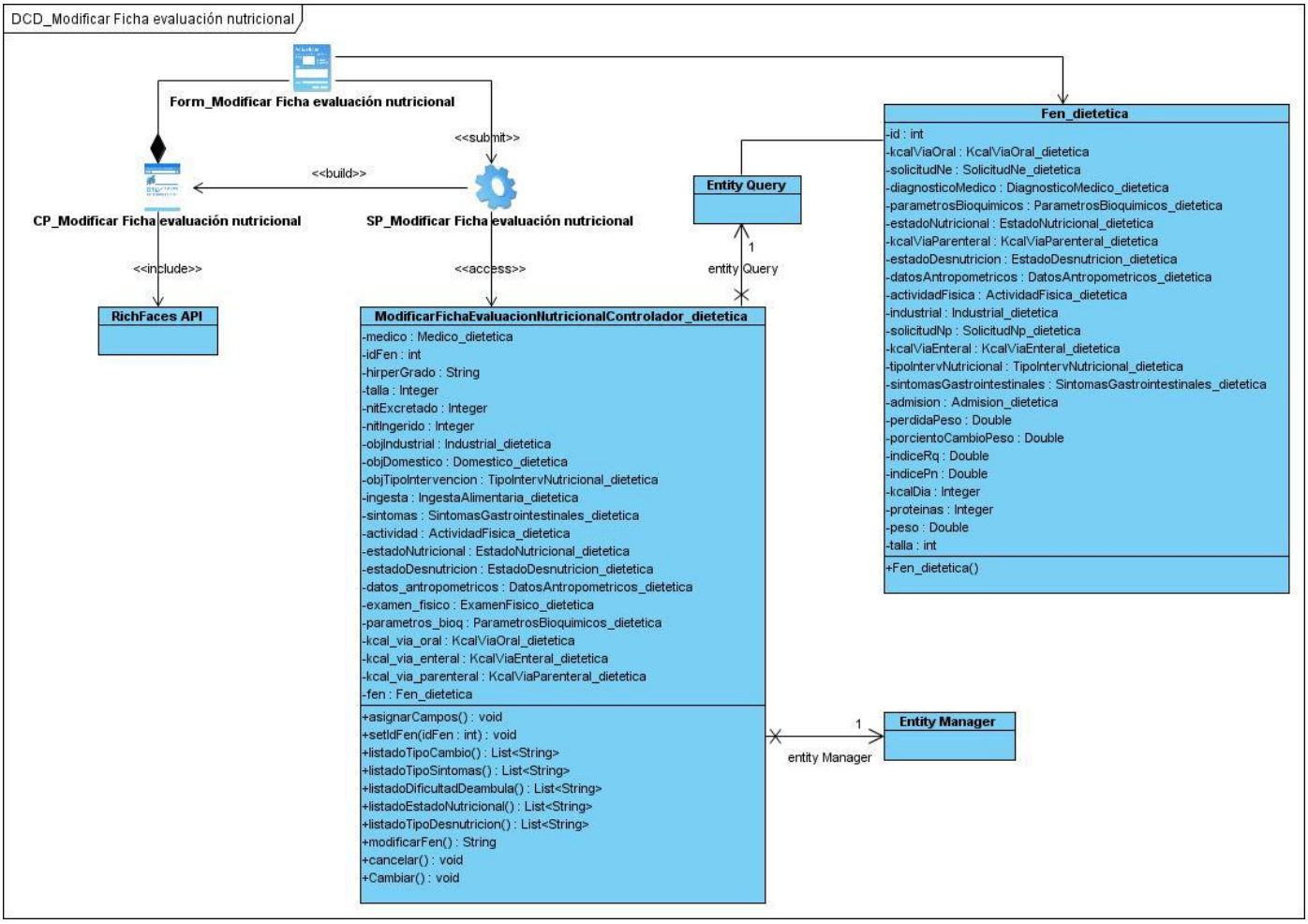


Figura 3.1.5 Diagrama de Clases del Diseño – Modificar ficha de evaluación nutricional.

(Ver artefacto Diagramas de Clases del Diseño)

3.2 Descripción de las nuevas clases u operaciones necesarias.

Nombre: ModificarPatronDietaControlador_dietetica	
Tipo de clase: Controladora	
Atributo	**Tipo**
entityManager	EntityManager
facesMessages	FacesMessages
bitacora	IBitacora
unidadMedidas	List<UnidadMedida_dietetica>
cid	Integer
idPatronDieta	Int
idModificarPaciente	Int
tipoUnidad	String
patronDieta	PatronDieta_dietetica
fichaEvaluacion	Fen_dietetica
Para cada responsabilidad:	
Nombre:	iniciar() devFecha(Date fechaNacimiento) listaUnidadMedidas() modificarPatronDieta() eliminarAlimentoDesayuno(AlimentoInComida_dietetica alim) eliminarAlimentoMeriendaUno(AlimentoInComida_dietetica alim) eliminarAlimentoAlmuerzo(AlimentoInComida_dietetica alim) eliminarAlimentoMeriendaDos(AlimentoInComida_dietetica alim)

	`eliminarAlimentoCena(AlimentoInComida_dietetica alim)`
	
	`eliminarAlimentoMeriendaTres(AlimentoInComida_dietetica alim)`

Descripción:	

(Ver Descripción de Clases)

CAPÍTULO 3. DESCRIPCIÓN Y ANÁLISIS DE LA SOLUCIÓN PROPUESTA

3.3 Descripción de las nuevas clases u operaciones necesarias

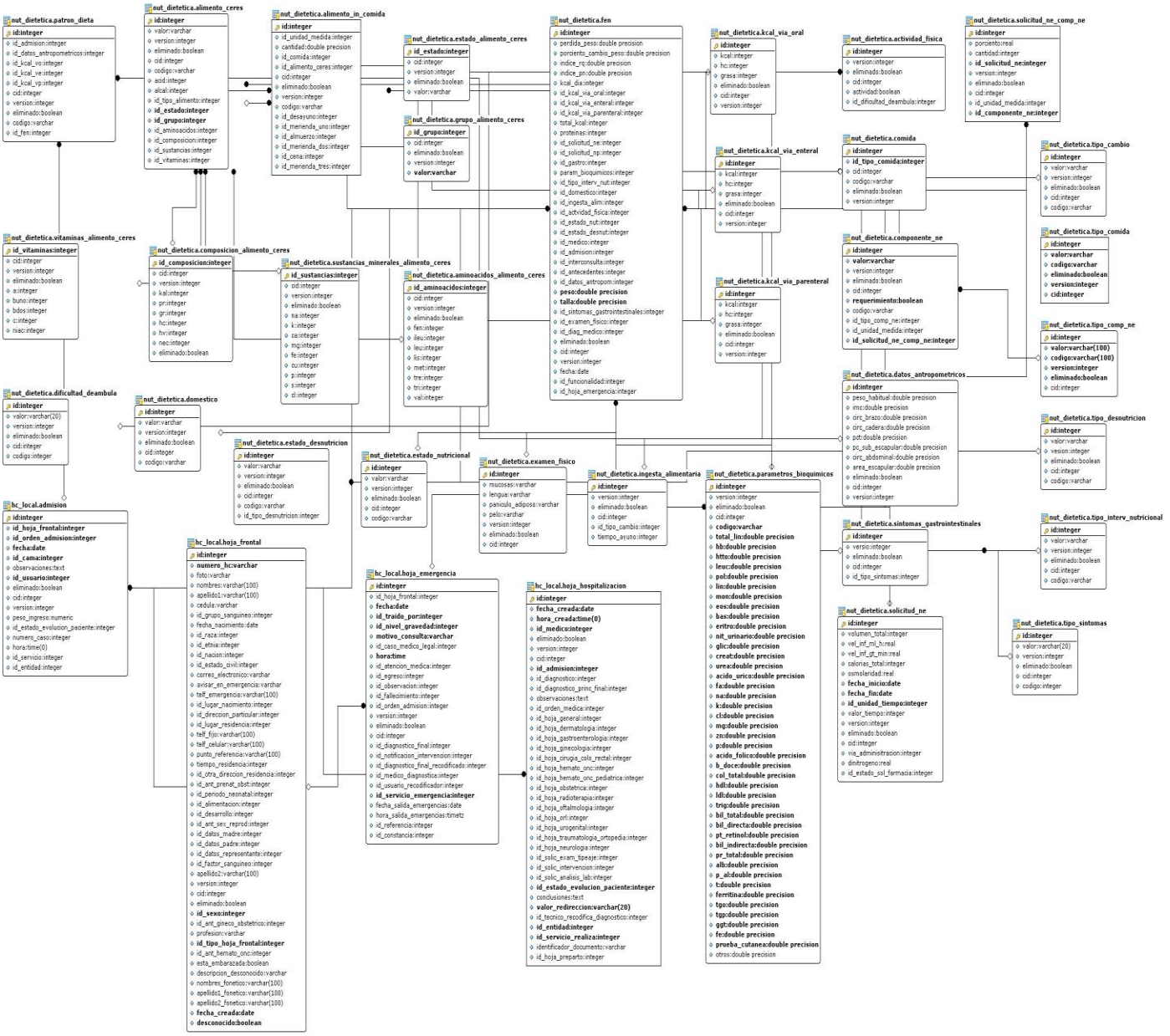


Figura 3.3.1

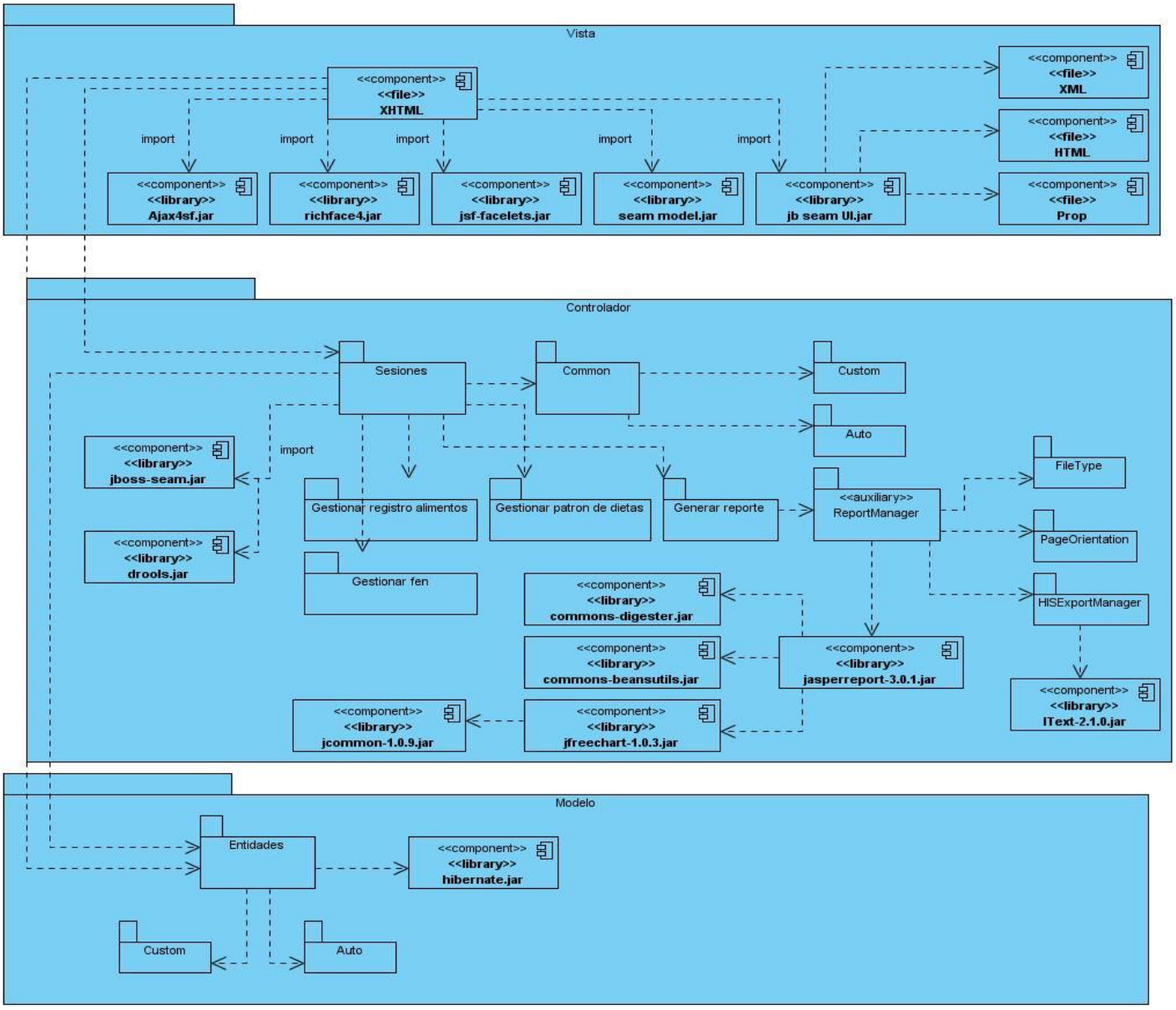
3.4 Breve valoración de las Técnicas de validación

Cuando un sistema es creado, se certifica que tenga una solución ante cualquier circunstancia posible. Esto se logra realizando diferentes técnicas de validación aunque para modelar todas las circunstancias posibles es difícil disponer de datos de entrenamiento. Existen diferentes técnicas de validación, estas deben de estar dirigidas al supuesto comportamiento erróneo que tendrá el sistema. Para impedir sucesos es que se ponen en práctica métodos y habilidades que facilitan todo este trabajo. Una de las principales condiciones a tener en cuenta en las técnicas de validación son las excepciones y el tratamiento de errores.

Las excepciones pueden surgir ante cualquier tipo de error u otro evento capaz de entorpecer la ejecución de un programa, estas se consideran eventos de ejecución que pueden causar que una funcionalidad fracase. También se encargan de trasladar el control de un programa en el lugar donde sucede la excepción, facilitando que continúe la ejecución del mismo aún en presencia de cualquier tipo de error. El registro XML nombrado *page.xml*, comprende la configuración de los diferentes mensajes que se muestran por cada tipo de excepción, así como la página a la que el sistema debe redireccionarse en caso de la presencia de un error sorpresivo.

Nombre: comida		
Descripción: (2)		
Atributo	**Tipo**	**Descripción**
id id_tipo_comida cid version eliminado valor	Integer Integer Integer Integer Boolean Varchar	(5)

3.5 Vista de Implementación. (Diagrama de Componentes)



3.5.1 Diagrama de componentes

CAPITULO 4. MODELO DE PRUEBAS

En el presente capítulo se muestra el modelo de pruebas, en el cual se efectúa una caracterización de las pruebas de caja negra, además de definirse y describirse los casos de prueba. La ejecución de las pruebas de software de un producto en cada fase de desarrollo es de suma importancia ya que son las que permiten identificar la calidad de los sistemas logrando el desarrollo de un sistema con excelente calidad y a su vez de gran aceptación en el mercado.

Las pruebas de software están divididas por dos vías:

- Pruebas de caja blanca
- Pruebas de caja negra.

A continuación se explica las características de la prueba de caja negra.

4.1. Prueba de caja negra

Para alcanzar un software en excelentes condiciones es necesaria la obtención de un buen modelo de pruebas. Con la realización de las pruebas no se puede afirmar que no existen defectos en un software, pero si prever la posible presencia de defectos. Se denomina prueba de caja negra a la acción que se realiza expresando que cada función que es completamente operativa. La prueba de caja negra se centra principalmente en los requisitos funcionales del software y permite obtener un conjunto de condiciones de entrada que ejerciten estos requisitos y se ignora la estructura de control. Estas pruebas se utilizan para demostrar que la entrada se acepta de forma adecuada, que se produce un resultado correcto y que se mantiene la integridad de la información externa.

Para llevar a cabo dicha prueba existen varias técnicas, entre las que se pueden citar las siguientes:

Técnica de la partición de equivalencia: Esta técnica fracciona el dominio de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba.

Técnica del análisis de valores límites: Mediante esta técnica se aprueba que aquellos casos de prueba que examinan las condiciones límite originan mejor resultado que aquellos que no lo hacen.

Técnica de grafos de causa-efecto: Es un método que comprende objetos tales como: objetos de datos, objetos de programa como módulos o colecciones de sentencias del lenguaje de programación que se modelan en el software.

El método a estilarse en el presente trabajo es el de **partición de equivalencia**. Mediante este método se identifican las clases de equivalencia y acto seguido los casos de pruebas. Su funcionamiento se basa en la división del dominio de entrada de un programa en clases de datos, mediante el cual se derivan los casos de prueba.

4.2. Descripción de los casos de pruebas

Escenarios del Registrar alimentos	Descripción de la funcionalidad	Flujo Central
EC1: Registrar alimentos	Registrar alimentos satisfactoriamente	Se selecciona la opción Registrar alimento. Se introducen y seleccionan los datos correspondientes. Se selecciona la opción Aceptar. Se registran los alimentos **Muestra la interfaz** Ver detalles de alimento Ver DCP: **Ver detalles de alimento.**
EC2: Cancelar operación	Cancelar la opción de registrar alimentos	Se selecciona la opción registrar alimento. Se introducen o seleccionan los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior.
EC3: Existen datos incompletos	Luego de haber introducido los	Se selecciona la opción Registrar

	datos, el sistema los verifica y valida, de haber incompletos muestra un indicador sobre los campos incompletos.	alimento. Se introducen los datos incompletos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incompletos.
EC4: Existen datos incorrectos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incorrectos muestra un indicador sobre los campos incorrectos.	Se selecciona la opción Registrar alimento. Se introducen los datos incorrectos. Se selecciona la opción Aceptar. **Muestra un indicador sobre los campos incorrectos.**

Tabla 4.1 Secciones a probar en el Caso de Uso: Registrar alimento.

Id del escenario	EC1	EC2	EC3	EC4
Escenario	Registrar alimento	Cancelar operación	Existen datos incompletos	Existen datos incorrectos
Variable 1 (Descripcion)	v			
Variable 2 (Grupo)	v			
Variable 3	v			

(Estado)				
Variable 4 (Kcal)	v		I	I
Variable 5 (Gr)	v			v
Variable 6 (N2)	v		I	
Variable 7 (Pr)	v			I
Variable 8 (HC)	v		I	v
Variable 9 (NE)	v			I
Variable 10 (A)	v			
Variable 11 (B2)	v			v
Variable 12 (B3)	v		I	
Variable 13	v			

(B1)				
Variable 14 (C)	v			
Variable 15 (Na)	v		l	
Variable 16 (Mg)	v			
Variable 17 (Cl)	v			
Variable 18 (K)	v		l	
Variable 19 (Fe)	v			
Variable 20 (S)	v			
Variable 21 (Ca)	v			
Variable 22 (Co)	v			
Variable 23	v			

(Phe)				
Variable 24 (Lys)	v			
Variable 25 (Trp)	v			
Variable 26 (Ile)	v			
Variable 27 (Met)	v			
Variable 28 (Val)	v			
Variable 29 (Leu)	v			
Variable 30 (Thr)	v			
Botón1 (Aceptar)	NA			
Botón2 (Cancelar)		NA		
Respuesta del	Registra el	Regresa a la vista	Muestra un	Muestra un

Sistema	alimento y se visualiza la interfaz ver detalles.	anterior.	indicador sobre los campos incompletos.	indicador sobre los campos incorrectos.
Resultado de la Prueba				

Escenarios del Ver detalles de alimento	Descripción de la funcionalidad	Flujo Central
EC1: Ver detalles de alimento	Ver detalles de alimento satisfactoriamente.	Se muestra la interfaz Ver detalles de alimento. Se muestran los datos de alimento Selecciona la opción Salir. Regresa a la vista de buscar alimento
EC2: Modificar alimento	Permite acceder al caso de uso modificar alimento	Se muestra la interfaz Ver detalles de alimento. Se muestran los datos de alimento Se selecciona la opción Modificar. Ver DCP Modificar alimento.
EC3: Eliminar alimento	Permite acceder al caso de uso eliminar alimento	Se muestra la interfaz Ver detalles de alimento. Se muestran los datos de

		alimento. Se selecciona la opción Eliminar. Ver DCP Eliminar alimento.

Tabla 4.2 Secciones a probar en el Caso de Uso: Ver detalles de alimento.

Id del escenario	EC 1	EC 2	EC 3
Escenario	Ver detalles de alimento satisfactoriamente	Modificar alimento	Eliminar alimento
Botón1 (Salir)	NA		
Botón2 (Modificar)		NA	
Botón 3 (Eliminar)			NA
Respuesta del Sistema	Regresa a la vista de anterior.	Muestra la interfaz para modificar.	Muestra la interfaz para eliminar.
Resultado de la Prueba			

Escenarios del modificar alimento	Descripción de la funcionalidad	Flujo Central
EC 1: Modificar alimento	Modificar un alimento	Muestra la interfaz para modificar

	satisfactoriamente.	alimento. Se modifican los datos correspondientes. Se selecciona la opción Aceptar. Se modifica el alimento. Muestra la interfaz Ver detalles del alimento. Ver DCP Ver detalles de alimento.
EC 2: Cancelar operación	Cancelar la opción de Modificar alimento.	Muestra la interfaz para modificar alimento. Se introducen los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior.
EC 3: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incompletos muestra un indicador sobre los campos incompletos.	Muestra la interfaz para modificar alimento. Se introducen los datos incompletos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incompletos.
EC 4: Existen datos incorrectos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incorrectos muestra un indicador sobre los	Muestra la interfaz para modificar alimento. Se introducen los datos

	campos incorrectos.	incorrectos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incorrectos.

Tabla 4.3 Secciones a probar en el Caso de Uso: Modificar alimento.

Id del escenario	EC 1	EC 2	EC 3	EC 4
Escenario	Modificar alimento satisfactoriamente	Cancelar operación	Existen datos incompletos	Existen datos incorrectos
(Kcal)	V		I	I
(Gr)	V			
(N2)	V		I	
(Pr)	V			V
(HC)	V		I	
(NE)	V			

(A)	V			V
(B2)	V			V
(B3)	V			
(B1)	V			
(C)	V		I	
(Na)	V			
(Mg)	V			I
(Cl)	V			
(K)	V		I	
(Fe)	V			

(S)	V			I
(Ca)	V			
(Co)	V			V
(Phe)	V			
(Lys)	V			
(Trp)	V		I	
(Ile)	V			
(Met)	V			V
(Val)	V			
(Leu)	V			

(Thr)	V			
(Aceptar)	NA			
(Cancelar)		NA		
Respuesta del Sistema	Modifica el alimento y se visualiza la interfaz ver detalles.	Regresa a la vista anterior.	Muestra un indicador sobre los campos incompletos.	campos incorrectos.
Resultado de la Prueba				

Escenarios del Crear patrón de dieta	Descripción de la funcionalidad	Flujo Central
EC1: Crear patrón de dieta	Crear patrón de dieta satisfactoriamente	Se selecciona la opción Crear patron de dieta. Se introducen y seleccionan los datos correspondientes. Se selecciona la opción Aceptar.

		Se crea el patrón de dieta **Muestra la interfaz** Ver detalles del patrón de dieta Ver DCP: **Ver detalles del patrón de dieta.**
EC2: Cancelar operación	Cancelar la opción de Crear patrón de dieta	Se selecciona la opción Crear patrón de dieta Se introducen o seleccionan los datos correspondientes. Se selecciona la opción Cancelar. **Se regresa a la vista anterior.**
EC3: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incompletos muestra un indicador sobre los campos incompletos.	Se selecciona la opción Crear patrón de dieta. Se introducen los datos incompletos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incompletos.

EC4: Existen datos incorrectos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incorrectos muestra un indicador sobre los campos incorrectos.	Se selecciona la opción Crear patrón de dieta Se introducen los datos incorrectos. Se selecciona la opción Aceptar. **Muestra un indicador sobre los campos incorrectos.**

Tabla 4.4 Secciones a probar en el Caso de Uso: Crear patrón de dieta

Id del escenario	EC1	EC2	EC3	EC4
Escenario	Crear patrón de dieta	Cancelar operación	Existen datos incompletos	Existen datos incorrectos
Botón1 (Adicionar desayuno)	NA			
Botón2 (Adicionar merienda)	NA			
Botón3 (Adicionar almuerzo)	NA			
Botón4 (Adicionar	NA			

merienda)				
Botón5 (Adicionar comida)	NA			
Botón6 (Adicionar merienda)	NA			
Boton7 (Aceptar)	NA			
Boton8 (Cancelar)		NA		
Respuesta del Sistema	Crea el patrón de dieta y se visualiza la interfaz ver detalles.	Regresa a la vista anterior.	Muestra un indicador sobre los campos incompletos.	Muestra un indicador sobre los campos incorrectos.
Resultado de la Prueba				

Escenarios del ver detalles del patrón de dieta	Descripción de la funcionalidad	Flujo Central
EC1: Ver detalles del patrón de dieta	Ver detalles del patrón de dieta satisfactoriamente.	Se muestra la interfaz Ver detalles del patrón de dieta. Se muestran los datos del patrón

		de dieta. Selecciona la opción Salir. Regresa a la vista de Listar pacientes evaluados.
EC2: Modificar patrón de dieta	Permite acceder al caso de uso modificar *patrón de dieta*	Se muestra la interfaz Ver detalles del patrón de dieta. Se muestran los datos del patrón de dieta Se selecciona la opción Modificar. Ver DCP Modificar patrón de dieta.
EC3: Eliminar patrón de dieta	Permite acceder al caso de uso eliminar *patrón de dieta*	Se muestra la interfaz Ver detalles del patrón de dieta. Se muestran los datos de alimento. Se selecciona la opción Eliminar. Ver DCP Eliminar alimento.

Tabla 4.5 Secciones a probar en el Caso de Uso: Ver detalles del patrón de dieta.

Id del escenario	EC 1	EC 2	EC 3
Escenario	Ver detalles del patrón de dieta satisfactoriamente	Modificar patrón de dieta	Eliminar patrón de dieta
Botón1	NA		

(Salir)			
Botón2 (Modificar)		NA	
Botón 3 (Eliminar)			NA
Respuesta del Sistema	Regresa a la vista de (especificar a qué vista regresa si no es la inmediata anterior).	Muestra la interfaz para modificar.	Muestra la interfaz para eliminar.
Resultado de la Prueba			

Escenarios del Modificar patrón de dieta	Descripción de la funcionalidad	Flujo Central
EC 1: Modificar patrón de dieta	Modificar un patrón de dieta satisfactoriamente.	Muestra la interfaz para modificar patrón de dieta. Se modifican los datos correspondientes. Se selecciona la opción Aceptar. Se modifica el patrón de dieta. Muestra la interfaz Ver detalles del patrón de dieta. Ver DCP Ver detalles del patrón de dieta.

EC 2: Cancelar operación	Cancelar la opción de Modificar patrón de dieta.	Muestra la interfaz para modificar patrón de dieta. Se introducen los datos correspondientes. Se selecciona la opción Cancelar. Se regresa a la vista anterior.
EC 3: Existen datos incompletos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incompletos muestra un indicador sobre los campos incompletos.	Muestra la interfaz para modificar patrón de dieta. Se introducen los datos incompletos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incompletos.
EC 4: Existen datos incorrectos	Luego de haber introducido los datos, el sistema los verifica y valida, de haber incorrectos muestra un indicador sobre los campos incorrectos.	Muestra la interfaz para modificar patrón de dieta. Se introducen los datos incorrectos. Se selecciona la opción Aceptar. Muestra un indicador sobre los campos incorrectos.

Tabla 4.6 Secciones a probar en el Caso de Uso: Modificar patrón de dieta

Id del escenario	EC 1	EC 2	EC 3	EC 4
Escenario	Modificar patrón de dieta satisfactoriamente	Cancelar operación	Existen datos incompletos	Existen datos incorrectos
Botón1 (Adicionar desayuno)	NA			
Botón2 (Adicionar merienda)	NA			
Botón3 (Adicionar almuerzo)	NA			
Botón4 (Adicionar merienda)	NA			
Botón5 (Adicionar comida)	NA			
Botón6 (Adicionar merienda)	NA			
	NA			

(Aceptar)				
(Cancelar)		NA		
Respuesta del Sistema	Modifica el alimento y se visualiza la interfaz ver detalles.	Regresa a la vista anterior.	Muestra un indicador sobre los campos incompletos.	campos incorrectos.
Resultado de la Prueba				

En el presente capítulo, se muestran las estrategias definidas para asegurar que el sistema obtenido como resultado de la presente investigación, sea robusto y confiable. Además se exponen algunos ejemplos de casos de pruebas que fueron utilizados.

CONCLUSIONES:

Una vez concluida la presente investigación se arribó a las siguientes conclusiones:

- El estudio realizado sobre los principales sistemas existentes en el mundo que gestionan información proveniente del área de Nutrición y Dietética, comprobó que, algunos son aplicaciones de escritorio, propietarios y no cumplen con todas las funcionalidades esperadas.

- Se establecieron las tecnologías, herramientas y patrones a utilizar, las cuales permitieron el desarrollo de un sistema robusto, flexible y a su vez adaptable a diferentes entornos de despliegue.

- La utilización de pautas para el proceso de desarrollo, garantizó homogeneidad en los artefactos obtenidos.

- El manejo de las funcionalidades propias a los procesos del movimiento hospitalario y planificación médica, permitirá una mejora a la realización de las mismas en el área de Nutrición y Dietética de las instituciones hospitalarias.

- La ejecución de pruebas de caja negra garantizó la calidad necesaria para su futura explotación y realización de una prueba piloto de las funcionalidades implementadas.

RECOMENDACIONES

Se recomienda:

- Analizar el funcionamiento de los equipos médicos utilizados en el área de Nutrición y Dietética con el objetivo de elaborar interfaces de comunicación con el sistema implementado.
- Establecer un mecanismo de notificación entre el Módulo de Almacén y Nutrición y Dietética para conocer los alimentos disponibles físicamente para la elaboración correcta de los Patrones de dieta.

Referencia

1. Sistema de Información Hospitalaria. [En línea] [Citado el: 24 de 2 de 2011.] http://www.facmed.unam.mx/emc/computo/ssa/HIS/hisindex.htm.

2. http://www.scribd.com/doc/46630078/Generaciones-de-Las-as

3. Sistema de Información Hospitalaria. [En línea] [Citado el: 22 de 9 de 2010.] http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf.

4. UI Develoment with JaveServer Faces. [En línea] [Citado el: 23 de 11 de 2010.] http://www.itk.ilstu.edu/faculty/bllim/itk353/j-jsf-ltr.pdf.

5. Katz, Max. *Practical Richfaces.* s.l.: Apress, 2008.

6. Ajax4jsf Developer Guide. [En línea] 20007. [Citado el: 25 de 11 de 2010.] http://www.jboss.org/file-access/default/members/jbossajax4jsf/freezone/docs/devguide/en/html_single/index.html.

7. Inside Facelets Part 1: An Introduction. [En línea] [Citado el: 3 de 12 de 2010.] http://www.jsfcentral.com/articles/facelets_1.html.

8. Seam in action. Manning Early Access Program. [En línea] [Citado el: 4 de 12 de 2010.] http://www.docstoc.com/docs/5134034/ManningSeaminActionSep2008.

9. http://www.assembla.com/wiki/show/wikicantabria/Entregable_6_-_Persistencia_de_datos

   http://petra.euitio.uniovi.es/~i9446728/persistencia.html

10. Java Hispano. [En línea] [Citado el: 5 de 12 de 2010.] http://www.javahispano.org/contenidos/es/liberado_drools_3_0/.

11. The Java Persistence API - A Simpler Programming Model for Entity Persistence. [En línea] [Citado el: 5 de 12 de 2010.] http://java.sun.com/developer/technicalArticles/J2EE/jpa/.

12. Franky, María Consuelo. Java EE 5 (sucesor de J2EE). [En línea] [Citado el: 5 de 12 de 2010.] http://www.acis.org.co/fileadmin/Conferencias/ConfConsueloFranky_Abr19.pdf.

13. JBoss releases JBoss Tools, Eclipse Plugins including Exadel. [En línea] [Citado el: 6 de 12 de 2010.] http://www.theserverside.com/news/thread.tss?thread_id=45933.

14. Informática Profesional. [En línea] [Citado el: 15 de 1 de 2011.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport.

15. Programación Java. [En línea] [Citado el: 24 de 2 de 2011.] http://www.lenguajes-de-programacion.com/programacion-java.shtml.

16. Notación para el modelado de procesos de negocio. [En línea] [Citado el: 30 de 5 de 2011.] http://www.bizagi.com/docs/BPMNbyExampleSPA.pdf

17. Rational Unified Process. [En línea] [Citado el: 25 de 1 de 2011.] http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42.

18. Especificaciones de requerimientos. [En línea] [Citado el: 18 de 1 de 2011.] http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos.

19. Neleste 2.0. [En línea] [Citado el: 24 de 2 de 2011.] http://www.neleste.com/modelo-vista-controlador/.

Bibliografía

1. Sistema de Información Hospitalaria. [En línea] [Citado el: 24 de 2 de 2011.] http://www.facmed.unam.mx/emc/computo/ssa/HIS/hisindex.htm.

2. Sistema de Información Hospitalaria. [En línea] [Citado el: 22 de 9 de 2010.] http://www.facmed.unam.mx/emc/computo/ssa/HIS/his.pdf.

3. UI Develoment with JaveServer Faces. [En línea] [Citado el: 23 de 11 de 2010.] http://www.itk.ilstu.edu/faculty/bllim/itk353/j-jsf-ltr.pdf.

4. Katz, Max. *Practical Richfaces.* s.l. : Apress, 2008.

5. Ajax4jsf Developer Guide. [En línea] 20007. [Citado el: 25 de 11 de 2010.] http://www.jboss.org/file-access/default/members/jbossajax4jsf/freezone/docs/devguide/en/html_single/index.html.

6. Inside Facelets Part 1: An Introduction. [En línea] [Citado el: 3 de 12 de 2010.] http://www.jsfcentral.com/articles/facelets_1.html.

7. Seam in action. Manning Early Access Program. [En línea] [Citado el: 4 de 12 de 2010.] http://www.docstoc.com/docs/5134034/ManningSeaminActionSep2008.

8. Java Hispano. [En línea] [Citado el: 5 de 12 de 2010.] http://www.javahispano.org/contenidos/es/liberado_drools_3_0/.

9. The Java Persistence API - A Simpler Programming Model for Entity Persistence. [En línea] [Citado el: 5 de 12 de 2010.] http://java.sun.com/developer/technicalArticles/J2EE/jpa/.

10. Franky, María Consuelo. Java EE 5 (sucesor de J2EE). [En línea] [Citado el: 5 de 12 de 2010.] http://www.acis.org.co/fileadmin/Conferencias/ConfConsueloFranky_Abr19.pdf.

11. JBoss releases JBoss Tools, Eclipse Plugins including Exadel. [En línea] [Citado el: 6 de 12 de 2010.] http://www.theserverside.com/news/thread.tss?thread_id=45933.

12. Informática Profesional. [En línea] [Citado el: 15 de 1 de 2011.] http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=ireport.

13. Programación Java. [En línea] [Citado el: 24 de 2 de 2011.] http://www.lenguajes-de-programacion.com/programacion-java.shtml.

14. Características del lenguaje Java. [En línea] [Citado el: 24 de 2 de 2011.] http://www.iec.csic.es/criptonomicon/java/quesjava.html.

15. Rational Unified Process. [En línea] [Citado el: 25 de 1 de 2011.] http://www.iteraprocess.com/index.php?option=com_content&task=view&id=18&Itemid=42.

16. Especificaciones de requerimientos. [En línea] [Citado el: 18 de 1 de 2011.] http://www.mitecnologico.com/Main/EspecificacionesDeRequerimientos.

17. Neleste 2.0. [En línea] [Citado el: 24 de 2 de 2011.] http://www.neleste.com/modelo-vista-controlador/.

18. Moreno Rojas, Rafael. *Nutrición y Dietética para tecnólogos de alimentos.* Madrid : Díaz de Santos.

19. La tecnología de Información y Comunicación (TIC) en los medios. [En línea] [Citado el: 15 de 9 de 2010.] http://www.ciberhabitat.gob.mx/medios/.

20. Diseño de Sitios web. [En línea] [Citado el: 13 de 1 de 2011.] http://www.killersites.com/translations/spanish/formatoAutomaticodelHTMLconJavascript.htm.

21. Estándares en la codificación. [En línea] [Citado el: 14 de 1 de 2011.] http://eduardodelamotta.com/2008/por-que-usar-estandares-en-la-codificacion.html.

22. Multi Stream. [En línea] [Citado el: 13 de 1 de 2011.] http://blog.multistream.tv/index.php/2009/02/12/el-estandar-de-codificacion-h264/.

23. AxiaCore. [En línea] [Citado el: 14 de 1 de 2011.] http://axiacore.com/2008/04/identar-codigo-fuente/.

24. Linux Lots. [En línea] [Citado el: 13 de 1 de 2011.] http://www.linuxlots.com/~barreiro/spanish/gnome-es/gnome-dev-info_es/codstd.html.

25. Sparx Systems. [En línea] [Citado el: 1 de 2 de 2011.] http://www.sparxsystems.com.ar/resources/tutorial/uml2_deploymentdiagram.html.

26. http://www.cad.com.mx/historia_del_lenguaje_java.htm. *Historia del lenguaje Java.* [En línea] [Citado el: 20 de 11 de 2010.] http://www.cad.com.mx/historia_del_lenguaje_java.htm.

27. Tecnologías de la Información y las Comunicaciones (TIC) en Cifras. Cuba 2009. [En línea] [Citado el: 2 de 10 de 2010.] http://www.one.cu/ticencifras2009.htm.

28. Las Tecnologías de la Información y la Comunicación (TIC) y la Sociedad. [En línea] [Citado el: 25 de 9 de 2010.] http://www.educando.edu.do/Portal.Base/Web/VerContenido.aspx?ID=110621.

29. Tecnologías de la Información y Comunicación (TIC) y Salud Pública. [En línea] [Citado el: 19 de 9 de 2010.] http://javiergarcialeon.wordpress.com/2009/11/22/tecnologias-de-la-informacion-y-comunicacion-tic-y-salud-publica/.

30. Tecnología al Instante. [En línea] [Citado el: 17 de 9 de 2010.] http://www.tecnologiahechapalabra.com/tecnologia/glosario_tecnico/articulo.asp?i=876.

31. Salud, humanización y psicología. [En línea] [Citado el: 16 de 9 de 2010.] http://medicablogs.diariomedico.com/javiercontreras/2010/01/06/nuevas-tecnologias-de-la-informacion-y-comunicacion-tic-para-la-salud/.

32. Tecnologías de la información y la comunicación (TIC's). [En línea] [Citado el: 16 de 9 de 2010.] http://www.monografias.com/trabajos37/tecnologias-comunicacion/tecnologias-comunicacion.shtml

33. Ciber Sociedad. [En línea] [Citado el: 15 de 9 de 2010.] http://www.cibersociedad.net/archivo/articulo.php?art=218.

34. LAS TIC Y SUS APORTACIONES A LA SOCIEDAD. [En línea] [Citado el: 15 de 9 de 2010.] http://peremarques.pangea.org/tic.htm.

35. de San Martín, Eloi. ProgramaciónWeb.net. [En línea] [Citado el: 12 de 1 de 2011.] http://www.programacionweb.net/articulos/articulo/?num=505.

36. García de Jalón, Javier, y otros. *Aprenda Java como si estuviera en primero.* Universidad de Navarra. España : Escuela Superior de Ingenieros Industriales.