

Universidad de las Ciencias Informáticas

Facultad 6



**Título: “Sistema de Planificación y Control de la
Dirección General de Alimentos de la UCI:
Módulo de administración”**

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autor: Vladimir Campos Kindelan

Tutores: Ing. María de Los Angeles Garcia Montero
Ing. Elvis Vázquez Aragón

Co-tutor: Ing. Alberto Tamayo Ramos

Consultante: Ing. Abel Meneses Abad

Asesor: Ms. Sandry Hernández Torres

Junio 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Dirección de Informatización de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Vladimir Campos Kindelan

Elvis Vázquez Aragón

María de Los Angeles Garcia Montero

DATOS DE CONTACTO

Nombre y Apellidos del **Tutor**: Elvis Vázquez Aragón

Email: evazqueza@uci.cu

Curriculum: Graduado de Ingeniero en Ciencias Informáticas, 2007, UCI. Instructor recién graduado en adiestramiento. Especialista general de la Dirección de Informatización.

Nombre y Apellidos del **Tutor(a)**: María de Los Angeles Garcia Montero

Email: magarcia@uci.cu

Curriculum: Graduada de Ingeniera en Ciencias Informáticas, 2008, UCI. Instructora recién graduada en adiestramiento. Especialista general de la Dirección de Informatización.

Nombre y Apellidos del **Asesor**: Sandry Hernández Torres

Email: sandry@uci.cu

Curriculum: Graduado de Ingeniero Industrial, 2004 en la CUJAE. Graduado de Máster en Ciencias en Logística y Gestión de la Producción, 2009 en la CUJAE. Actualmente, Especialista Superior de la Vicerrectoría de Logística de la UCI.

Nombre y Apellidos del **Consultante**: Abel Meneses Abad.

Email: abelma@uci.cu

Curriculum: Graduado de Ingeniero en Telecomunicaciones y Electrónica, 2004, CUJAE. Posee categoría docente de Instructor. Es líder del Proyecto Unicornios (Servicios Especializados para la Migración a SWL de la UCI). Miembro organizador del Taller de Software Libre de Informática Habana. Actualmente se desempeña como Asesor de Investigaciones del Vicedecano de Producción de la Facultad X.

DEDICATORIA

Este trabajo va dedicado especialmente a cuatro personas que representan cada célula de mi ser, a mi adorada e inolvidable madre, aunque no me distes el tiempo en vida de regalarte este triunfo, donde quiera que estés te llegue el placer de yo haber realizado tu sueño y que te recuerdo siempre, papi gracias por ser tan especial, por haber cumplido también con mami, conmigo y haberme aguantado tanto, a los dos de verás que hacer esto ha sido una de las pocas cosas con que les puedo pagar tanto amor.

A mi bella e incondicional esposa, a quien le ha tocado vivir las buenas y malas, por haberme dado esos momentos de felicidad que son únicos y la fuerza para vencer obstáculos en la vida, te dedico este triunfo. Deseo también que algún día se te cumpla tu sueño y seas una buena fisioterapeuta que bien te lo mereces.

A mi princesa, la más hermosa y buena de todas las niñas, a través de este regalo que es tuyo también, gracias por haberme hecho el padre más feliz y quiero que sigas el ejemplo de papi, quien te ama sin fin.

A mi familia sin que se me quede uno, esto es una victoria de ustedes también.

AGRADECIMIENTOS

Agradezco especialmente a todas las personas que a lo largo de mi carrera han contribuido a lograr este sueño. Súper especialmente agradezco a mis tutores Elvis, Maria, Sandry que siempre estuvieron corriendo conmigo y no me abandonaron nunca, a Gladys, Abel por su incondicional ayuda. A todo el equipo del Proyecto AlimentosUCI a Jorgito, Alberto, Deibys, Joel Michel, Dayron, Raulito, Ariel, Ingrid, Guille, a la Dirección de Informatización en general. Agradezco a mis compañeros de la Dirección de Laboratorios, a Michael y Landy, a Alain, Daisy, Yadelis, Yuriem por la compañía y ayuda, a mis profesores de la Facultad 6, a mis compañeros de trabajo de la Dirección General de Alimentos y los complejos comedores, amigos y a todos los que, aunque no se vean aquí de corazón les agradezco todo el apoyo.



“Es ley maravillosa de la naturaleza que solo esté completo el que da: y no se empieza a poseer la vida hasta que no vaciamos sin reparo y sin tasa, en bien de los demás, la nuestra”.

RESUMEN

La presente investigación está enfocada a solucionar la problemática que presenta el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI, evidenciada en el aislamiento de la configuración de las aplicaciones que lo componen, lo que provocó consecuencias desfavorables en su explotación. El hecho de no contar con una herramienta administrativa que englobe una serie de funcionalidades de autenticación, accesibilidad, autorización, entre otras, ha sido el objeto de motivación para desarrollar esta aplicación, en favor de garantizar de forma flexible y segura el flujo informativo que se procesa en este sistema. La composición de esta aplicación web se basa en una interfaz de usuario provista de un menú de funcionalidades, que permite interactuar con un servicio web, donde están declarados varios métodos consumidos por estas aplicaciones. Las ventajas de haberlo desarrollado sobre la tecnología web, mejora la usabilidad múltiple de usuarios y evitará hacer cambios engorrosos en la arquitectura durante el proceso de integración con las demás aplicaciones del Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI u otros sistemas con negocios similares. Para implementar este módulo se contempló el desarrollo de algunas aplicaciones de gestión existentes en el mundo, haciendo énfasis en el funcionamiento y características fundamentales de las mismas. Además se tuvieron en cuenta las herramientas y tecnologías factibles para desarrollarlas, así como las ventajas de haber usado la metodología ágil SXP como guía de orientación en el ciclo completo de desarrollo de este software.

PALABRAS CLAVE

Aplicación, Autorización, Accesibilidad, Configuración, Servicio web, Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI, SXP

TABLA DE CONTENIDOS

DEDICATORIA	I
AGRADECIMIENTOS	II
RESUMEN	IV
INTRODUCCION	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	6
1.1 Los Sistemas de Información (SI) en el proceso de negocio	6
1.2 La administración en los sistemas informáticos	9
1.3 Soluciones para la administración de sistemas informáticos. El BackOffice	12
1.4 Tecnologías, tendencias y evolución de la Web.....	14
1.5 Lenguajes y herramientas para el desarrollo de la aplicación.....	17
1.6 Metodologías de desarrollo de software. Herramienta y lenguaje de modelado	24
CAPÍTULO 2: FUNDAMENTOS DE LA SOLUCIÓN PROPUESTA	31
2.1 Objetivos estratégicos. Análisis crítico de la ejecución de los procesos	31
2.2 Propuesta de integración	33
2.3 Propuesta de la aplicación	34
2.4 Caracterización de la arquitectura. Modelo del diseño.....	46
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA	58
3.1 Tareas de ingeniería	58
3.2 Plan de release	58
3.3 Estándar de programación	60
3.4 Pruebas de aceptación.....	61
CONCLUSIONES	67
RECOMENDACIONES	68
TRABAJOS CITADOS	69
BIBLIOGRAFIA	71
ANEXOS	72
GLOSARIO	85

INTRODUCCION

En todo el entorno de las buenas prácticas del desarrollo de una organización, las Tecnologías de la Información y las Comunicaciones (TIC) aplicadas a los sistemas de información han manifestado su latente potencial como herramientas claves para el control y administración de los procesos de la empresa. Estos sistemas automatizados permiten agilizar la gestión del negocio y organizar la forma en que la información, las funciones, reglas, roles, normas y procedimientos en general se asocian a cada trabajador de la empresa.

En todo entorno empresarial es imprescindible contar con sistemas informáticos que se ajusten a las características del proceso y hagan fluir su objeto principal: la información. En todo sistema informático puede evaluarse la administración como el entorno general que comprende la seguridad y la configuración, indicadores administrativos claves para lograr el procesamiento de forma segura dentro del entorno de negocio garantizando la integridad, confidencialidad y disponibilidad de la información en el sistema.

Los sistemas operativos fueron los pioneros en establecer indicadores de configuración y seguridad. Por lo que muchos de ellos fueron diseñados como software propietarios y simulaban ser seguros a través del aislamiento de los usuarios a su código. Los mismos implementaban métodos que pretendían ocultar los recursos de alta sensibilidad, propiciando que el usuario no detectara agujeros de seguridad e impidiendo el acceso a repararlos. Esto tuvo un efecto desfavorable para los clientes pues los hizo depender de los creadores de dichos sistemas. Asimismo, para cualquier tipo de cambios o mantenimiento se generó una dependencia administrativa absoluta.

La interpretación asumida por muchas comunidades de clientes acerca de la seguridad del software, concluyó siendo una falsa creencia que respaldaba los llamados sistemas oscuros. Este concepto, que comenzó a cambiar paralelamente con la integridad de la información, se convirtió de inmediato en una debilidad de los sistemas informáticos y trajo consigo algunos problemas que se agudizaron con el surgimiento de las redes de Internet.

Con el arribo de las grandes comunidades de ordenadores, se ha hecho vital establecer indicadores administrativos a los sistemas en dependencia de su naturaleza. Inicialmente, existían redes digitales que pretendían responder a servicios y funcionalidades públicas absolutamente vinculadas al compartimiento de algunos recursos de hardware, como impresoras, faxes y otros. La evolución gradual del compartimiento de este tipo de recursos, abrió una brecha que permitió posteriormente compartir información entre usuarios de una o varias redes, lo cual tuvo un desarrollo a gran escala

representando un paso de avance en el procesamiento de información en la era digital. Las empresas capitalistas generalmente con mayor alcance a estas tecnologías, comenzaron a dirigir sus negocios usando sistemas informáticos basados en flujos financieros e informativos, los cuales, a pesar de constituir un gran progreso tecnológico, no contaban con entornos flexibles y seguros.

En la actualidad, a pesar del desarrollo tecnológico alcanzado por la industria del software a nivel mundial, resulta difícil obtener sistemas configurables que dispongan a su vez de altos niveles de seguridad, debido a las contradicciones que se generan al integrar estos indicadores. En los sistemas informáticos, es muy importante la conducción segura de la información, a partir de la configuración de todos los componentes del sistema. La autenticación, autorización, configuración y la seguridad añaden un valor a las funcionalidades del sistema, que lo hacen una herramienta flexible y segura.

Nuestro país, a pesar de todos los contratiempos generados por el embargo económico impuesto por las potencias capitalistas, quienes han monopolizado las grandes industrias del software, ha concentrado sus esfuerzos en el desarrollo y uso de las TIC, experimentando nuevas estrategias en varios campos de la ciencia y la tecnología, orientados a insertar a Cuba en el mercado del software.

La Universidad de las Ciencias Informáticas (UCI), idea forjada en la visión del compañero Fidel Castro Ruz, juega un papel protagónico y decisivo en el desarrollo de la informatización de la sociedad cubana y algunos rincones de nuestra América. Sus miembros están representados en casi 18 000 estudiantes, profesores y trabajadores vinculados de conjunto al proceso de formación docente y productivo.

En cuanto a los procesos de apoyo más importantes de la UCI, se encuentra el servicio de comedores, el cual está orientado a garantizar la alimentación a la comunidad universitaria. Este servicio ejecutado por la Dirección General de Alimentos, recientemente sufrió cambios en su diseño lo que ha fomentado algunas mejoras en su proceso de planificación y control.

Como resultado del rediseño de este proceso un equipo conformado por la Dirección General de Alimentos (DGA) y la Dirección de Informatización de la UCI, han desarrollado un grupo de aplicaciones para la automatización de gran parte de las actividades del proceso. Estas aplicaciones se describen a continuación:

El Sistema de Asignación, permite asignar grupos de clientes y usuarios pertenecientes a las diferentes estructuras organizativas de la UCI, a las unidades de producción y servicio de la DGA, a partir de la evaluación de los indicadores de carga y capacidad. En esta aplicación están contenidos

los perfiles, distribuciones y estructuras jerárquicas que representan las unidades de servicios, comedores y puntos de acceso respectivamente, así como la caracterización de los eventos que rigen dicho servicio.

El Sistema de Reservación y Cancelación brinda la posibilidad de definir quiénes de los clientes y usuarios asignados, serán candidatos o excluidos a recibir el servicio de alimentación.

El Sistema de Planificación del Menú garantiza la conformación de platos para la planificación y conformación del menú en diferentes períodos, a partir de un listado de productos y teniendo en cuenta la demanda de comensales para los eventos.

El Sistema de Control de Acceso permite registrar el acceso de los clientes y usuarios asignados, a los puntos de acceso de cada unidad de producción y servicios, teniendo en cuenta los datos de los mismos. Por último se cita el Sistema de Reportes visualiza el resultado estadístico a partir del comportamiento de los eventos.

Estas aplicaciones fueron desarrolladas de manera independiente, priorizando en cada una sus principales funcionalidades, y considerando que todas formarían parte de un sistema único. A pesar de haber obtenido un gran avance en la automatización del proceso, el aislamiento entre las aplicaciones que lo conforman provocó una serie de problemáticas que generaron consecuencias desfavorables en el flujo informativo del mismo, como son: demora en la configuración y bajos niveles de seguridad.

A partir de estas problemáticas surge el siguiente problema científico: ¿Cómo integrar la configuración en el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI?

Con relación al problema científico planteado, se define como objeto de estudio: El proceso de administración en los sistemas informáticos, adentrado en el siguiente campo de acción: La configuración en el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI.

Atendiendo a lo anterior se define como objetivo general: Desarrollar una aplicación informática, que permita configurar el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI, desglosado en los siguientes objetivos específicos:

- Definir las funcionalidades de la aplicación
- Modelar el diseño

- Implementar el código fuente
- Validar los resultados de la implementación

Como fundamento de los objetivos específicos se planifican para su ejecución las siguientes tareas de investigación:

- Valoración del estado del arte para la configuración en los sistemas informáticos
 - Evaluación de las tendencias, tecnologías, metodologías
 - Selección de las herramientas utilizadas para desarrollar la solución
- Definición de la Lista de Reserva del Producto (LRP) para identificar las funcionalidades de la aplicación
- Identificación y especificación de las historias de usuario
- Modelado de las clases del diseño
- Definición de las tareas de ingeniería
- Elaboración del plan de release
- Definición del estándar de código y desarrollo del código fuente de la aplicación
- Validación de las pruebas de aceptación

El presente trabajo consta de tres capítulos y varios anexos, aborda el ciclo completo de desarrollo de la aplicación que permitirá administrar el Sistema de Planificación y Control de la Dirección General de Alimentos.

Capítulo 1: Fundamentación Teórica: abarca conceptos relacionados con el proceso de integración de indicadores administrativos en sistemas informáticos. Se analizan conceptos relacionados con el objeto de estudio y el campo de acción. Asimismo se incluye una valoración del estado del arte para estos sistemas. Por último, se evalúan algunas tendencias, tecnologías y se brinda una apreciación de las plataformas, metodologías, lenguajes y herramientas utilizadas en el desarrollo de la aplicación.

Capitulo 2: Fundamentos de la solución propuesta: se expresa el desglose de la problemática existente en el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI a partir de los objetivos estratégicos, así como el análisis del flujo actual de los procesos enmarcados en el entorno donde se desarrolla la solución. Se describe la propuesta para la integración y se realiza una descripción general de las funcionalidades acorde con los requerimientos funcionales y no funcionales identificados. Además se diseñaron los modelos de clases teniendo en cuenta los patrones de arquitectura y diseño y se obtuvo modelo de persistencia para realizar el modelo físico de datos.

Capitulo 3: Implementación y prueba: se visualiza la interacción entre componentes, así como el despliegue de los nodos que conforman la aplicación desarrollada a través de los diagramas correspondientes. Se muestran además, los artefactos referentes a la planificación de la fase de desarrollo como son el plan de release y tareas de ingeniería. La implementación revela los artefactos referentes al estándar de código y código fuente donde se definen las clases, métodos y el servicio que sustentan esta aplicación. En cuanto a la validación del correcto funcionamiento de los requerimientos definidos en las Historias de Usuario (HU) se realizan las pruebas de aceptación como premisa de conformidad de los usuarios finales de la aplicación.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El reciente avance de las TIC ha inducido un nuevo modelo de satisfacer las exigencias tecnológicas a favor de brindar mejores prestaciones mediante el diseño de sistemas con interfaces amigables, de alta disponibilidad.

Como referencia de los requerimientos de configuración de todo sistema informático y a partir de la necesidad de integrar todas las aplicaciones que conforman el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI, este capítulo abarca conceptos relacionados con el proceso de integración de indicadores administrativos en sistemas informáticos. Se definen algunos criterios a partir de estos conceptos. Incluye una valoración del estado del arte para estos sistemas. Se evalúan algunas tendencias, tecnologías y una apreciación de las plataformas, metodologías, lenguajes y herramientas utilizadas en el desarrollo de la aplicación.

1.1 Los Sistemas de Información (SI) en el proceso de negocio

Antes de indagar acerca de los SI, sería prudente referirnos a su objeto central por el cual se le atribuye ese nombre. La información está concebida por varios estudiosos como un fenómeno que proporciona significado o sentido a las cosas, e indica mediante códigos y conjuntos de datos, los modelos del pensamiento humano. La información por tanto, procesa y genera el conocimiento humano. Aunque muchos seres vivos se comunican transmitiendo información para su supervivencia, la diferencia de los seres humanos radica en su capacidad de generar y perfeccionar tanto códigos como símbolos con significados que conformaron lenguajes comunes útiles para la convivencia en sociedad, a partir del establecimiento de sistemas de señales y lenguajes para la comunicación.

Algunas definiciones citadas afirman que: *“Los datos están constituidos por los registros de los hechos, acontecimientos, transacciones, etc. Por el contrario, la información implica que los datos estén procesados de tal manera que resulten útiles o significativos para el receptor de los mismos, por lo que en cierto modo, los datos se pueden considerar la materia prima para obtener información”* (1).

Otros autores definen que *“...Son datos ordenados de manera tal que permita tomar buenas decisiones”* (2).

Atendiendo a los conceptos referenciados anteriormente se interpreta que la información representa el resultado lógico de la intercepción de varios elementos y criterios que contienen un valor agregado y son utilizados en distintos objetivos, que pueden ser en su mayoría de decisión.

Normalmente el término Sistema de Información es usado de manera errónea como sinónimo de sistema informático, estos precisamente ubicados en el campo de estudio de la tecnología de la información, aún cuando puedan formar parte de un SI (como recurso material), por sí solos no se pueden considerar como sistemas de información, ya que ese concepto es más amplio que el de sistema informático.

Un SI comprende un conjunto formal de procesos que, operando sobre una colección de datos estructurada según las necesidades de la empresa, recopilan elaboran y distribuyen la información necesaria, para las operaciones de dicha empresa y para las actividades de dirección y control correspondientes (decisiones) de acuerdo a su estrategia de negocio.

Algunos gurús que se han dedicado al estudio de este campo han definido los SI como herramientas para: "... procesar entradas, mantener archivos de datos relacionados con la organización y producir información, reportes y otras salidas...". (3)

Los SI están basados en un conjunto organizado de elementos, principalmente relacionados entre sí (Anexo 1). Estos elementos aparecen en 4 clasificaciones que pueden estar expresadas en:

- Personas
- Datos
- Actividades o técnicas de trabajo

Generalmente los SI utilizan recursos materiales en general, típicamente recursos informáticos y de comunicación, aunque no tienen por qué ser de este tipo obligatoriamente. Todo SI se clasifica en base a su función (Anexo 2) a partir de esto ellos responden a un determinado resultado, en dependencia del tipo de usuario final del mismo:

- Sistema de procesamiento de transacciones (TPS).- Gestiona la información referente a las transacciones producidas en una empresa u organización
- Sistemas de información gerencial (MIS).- Orientados a solucionar problemas empresariales en general
- Sistemas de soporte a decisiones (DSS).- Herramienta para realizar el análisis de las diferentes variables de negocio con la finalidad de apoyar el proceso de toma de decisiones

- Sistemas de información ejecutiva (EIS).- Herramienta orientada a usuarios de nivel gerencial, que permite monitorizar el estado de las variables de un área o unidad de la empresa a partir de información interna y externa a la misma
- Sistemas de automatización de oficinas (OAS).- Aplicaciones destinadas a ayudar al trabajo diario del administrativo de una empresa u organización
- Sistema experto (SE).- Emulan el comportamiento de un experto en un dominio concreto
- Sistema Planificación de Recursos (ERP).- Integran la información y los procesos de una organización en un solo sistema

Todas estas modalidades de SI pudieran estar basadas en el uso de computadoras, a partir del uso de un software. En la actualidad los más evolucionados son los de clasificación ERP entre los que se han desarrollado se encuentran ASSET, SAP, siendo este ultimo vanguardia del entorno empresarial. Estos software han logrado integrar en varios módulos la gran mayoría de las funcionalidades de los anteriores, procesando un conjunto de datos de entrada y proporcionando como resultado otro conjunto de datos de salida expresados en reportes, manejando diversos criterios que finalmente apoyarán la toma de decisiones.

La implementación de esta modalidad de sistemas informáticos para automatizar los procesos de negocio ha sido el paradigma del éxito en muchas empresas del mundo, principalmente las que están orientadas a la producción y los servicios. Con estos sistemas aumenta la garantía en la toma de decisiones, disminuyendo los tiempos de discusión de las propuestas de solución a problemas e incrementando a su vez la productividad y la calidad de los servicios.

El desarrollo de la tecnología opuesto a la resistencia al cambio, ha generado que muchas empresas que persisten en utilizar procedimientos manuales y no implementen sistemas informáticos, que abran paso a una mejor variante en la toma de decisiones mediante el uso de nuevas tecnologías, resulten derrotadas quebrando su objeto fundamental, la producción y los servicios ocasionando un efecto negativo en la competitividad, indicador clave del actual desarrollo empresarial.

En todo sistema informático, a pesar de las numerosas ventajas que estos ofrecen al gigantesco entorno empresarial, existen riesgos asociados al funcionamiento de los mismos. Estos generalmente están vinculados a la manera en que se organiza y manipula el flujo informativo, teniendo en cuenta los indicadores administrativos que lo hagan flexible, usable y seguro.

Apoyados en algunos elementos del objeto de estudio y vinculados con el campo de acción se describen importantes parámetros a tener en cuenta para el cumplimiento del objetivo general de esta investigación.

1.2 La administración en los sistemas informáticos

La administración es un concepto que abarca numerosas responsabilidades relacionadas entre sí con el fin de centralizar el manejo de varios elementos de un entorno único. Este término es utilizado por lo general para hacer referencia a los negocios de la empresa y a pesar de que en el campo de la informática se conserva su uso, en el campo empresarial a partir de su evolución se le define como gestión. Varios autores catalogados como los gurús del campo empresarial, definen la administración como: *"... el proceso de planear, organizar, dirigir y controlar el uso de recursos para lograr objetivos..."* (4)

Otra definición es la de Harold Koontz y Heinz Weihrich quienes aportaron el siguiente concepto *"...la administración es el proceso de diseñar y mantener un ambiente en el cual las personas, trabajando juntas en grupos, alcanzan con eficiencia metas seleccionadas..."* (5)

Partiendo de los elementos que arrojan las definiciones anteriores, sí se hace una similitud con el campo de la informática, se puede describir la administración en los sistemas informáticos, como la conducción integrada mediante un software para gestionar un conjunto de funcionalidades de configuración y seguridad a partir del uso de recursos de hardware y que interactúan armónicamente en el funcionamiento de un sistema logrando la integridad, confidencialidad y disponibilidad de la información.

La administración en los sistemas informáticos esta llamada a responder generalmente al tratamiento de requerimientos no funcionales que a su vez están implícitos en todas las funcionalidades del negocio que automatizan. El propósito de las herramientas administrativas en la gran mayoría de los sistemas es integrar a través de indicadores comunes de sus piezas. Entre los indicadores fundamentales se destacan la configuración y la seguridad mediante los cuales se garantiza la gestión de cambios y la restricción a través de políticas y reglas, garantizando el flujo informativo cuando los usuarios del sistema lo necesiten.

Luego de establecer los controles de acceso sobre los sistemas y la aplicación, es necesario administrar eficientemente todas las medidas de seguridad lógica, implicando la implementación, seguimientos, pruebas y modificaciones sobre los accesos de los usuarios de los sistemas.

La configuración y la seguridad en los sistemas informáticos

La configuración es un conjunto de datos que determina el valor de algunas variables de un programa o sistema de software, estas opciones generalmente son declaradas y definidas por el usuario administrador. Al abordar este indicador en los sistemas informáticos, debe hacerse de la forma más flexible, lo que sugiere personalizarla a favor de lograr adaptarla a las necesidades de uso en el mismo. De esta forma estará definida especialmente por el usuario y además almacenada generalmente en un archivo o en una base de datos. También puede estar cifrada para que sólo se pueda modificar por el programa administrativo que permita hacerlo, o puede ser texto plano para que se pueda modificar sin depender del programa. La configuración de un software es la actividad del proceso administrativo que representa uno de los requisitos no funcionales, está centrada en garantizar el nivel de dinámica o gestión del cambio, en otras palabras una flexibilidad controlada y constituye una de las responsabilidades del administrador del sistema mediante las cuales se organiza, planifica y controla el funcionamiento de los componentes del sistema.

Asociada a la configuración se incluye otro indicador muy importante que se encarga de contrarrestar la flexibilidad en los sistemas informáticos mediante la custodia de todos sus elementos, esto se refiere a la seguridad. De acuerdo a los niveles para los cuales se haya definido la seguridad, consiste en el reconocimiento de los usuarios que deberán identificarse para acceder a las aplicaciones y las operaciones que pudieran realizar acorde con sus privilegios, esto evita el acceso y/o manipulación indebida de los datos y en su caso quedaría registrada. Dentro de la seguridad, la integridad garantiza que todas las partes funcionen de forma correcta en su totalidad. El aislamiento permite que los datos manipulados por los usuarios deben ser independizados uno de otro, físico y lógicamente, usando técnicas de ocultación y/o compartimiento, también deberán aislarse los datos públicos y los de carácter crítico. La auditabilidad es el procedimiento utilizado para verificar, comprobar y demostrar el uso adecuado del sistema, el registro de todas las transacciones y la información en cuanto a, tiempo, lugar, autoridad y recurso empleado. Estableciendo este indicador quedarían asignados los usuarios de las aplicaciones mediante la configuración, se declararían los diversos roles que agruparían estos usuarios de acuerdo con las responsabilidades en el negocio, permitirá la auditabilidad del software para garantizar el control y la custodia de la información.

Autenticación y autorización. Roles

Autenticarse responde evidentemente a los indicadores de seguridad en el proceso administrativo de los sistemas informáticos, es la función mediante la cual se citan las verificaciones de que alguien o

algo es quien o lo que dice ser. En los sistemas informáticos o redes de computadoras la autenticación se lleva a cabo comúnmente a través de una interfaz de inicio de sesión que exige como elementos de validación de acceso, que los clientes escriban su nombre de usuario y contraseña.

En algunas tecnologías como la referente a cliente–servidor, la seguridad se maneja del lado del servidor. En el momento en que los usuarios intentan autenticarse para conectarse a las páginas de un servidor se genera una petición de accesibilidad. Una clase controladora se encargará de que a través de sus métodos, la clase modelo valide la entrada del usuario en la base de datos. De coincidir éstos, el cliente no sólo accederá con éxito, sino será considerado confiable en el sistema cada vez que ejecute alguna acción relacionada con sus funciones en el negocio.

Finalmente el rol del usuario es chequeado contra una política específica, donde cada uno de los controles y acciones son permitidos o denegados. Condicionalmente el usuario permanecerá autenticado tanto tiempo como dure la conexión. Una vez que el usuario se autentica, el sistema tendrá previsto como controlar lo que puede hacer en una aplicación, las interfaces donde puede agregar, modificar, eliminar información o si sólo puede leerla.

La autorización suele considerarse como una fusión de los aspectos de configuración y seguridad, aunque en realidad no es más que la forma en que se respaldan las propias reglas del negocio (políticas). En resumen es la representación jurisdiccional de la Empresa para determinar qué usuarios tienen acceso a determinadas funcionalidades, dentro del sistema relacionadas a su vez con la función de cada uno de ellos en el proceso.

El acceso a la información debe estar controlado por la función que desempeñan en el negocio los usuarios para los que se requiere dicho acceso. Los roles responden a los privilegios de lectura, escritura, ejecución, borrado, insertar entre otras. Algunos ejemplos de roles son: administrador de sistema, planificador, supervisor, consultores etc.

En cierta forma los usuarios, funcionalidades, roles y reglas específicamente definidas en cada una de las partes de un sistema informático se integraran por lo general para conducir de manera centralizada los indicadores administrativos, en este caso la configuración y la seguridad, dándole una mayor desenvolvura a los usuarios administradores, quienes están responsabilizados con varias actividades como la instalación, configuración, soporte, mantenimiento, planeación de respuesta a contingencias, la supervisión, el entrenamiento de operadores de cómputo y ser el consultor para los problemas que se encuentran más allá del conocimiento técnico del personal de soporte.

Se puede afirmar que la información como ya se ha mencionado es fundamental para realizar las operaciones y actividades en las empresas, pero es necesario manejarla de una forma adecuada y a la vez rápida, siendo necesario que los sistemas informáticos estén bien administrados para así tener un mayor control de los mismos.

1.3 Soluciones para la administración de sistemas informáticos. El BackOffice

En la actualidad mundial el negocio de la administración tiene un carácter integrador y se enfoca de diferentes maneras, estas generalmente responden a muchas de las particularidades de cada sistema en dependencia de cómo se gestionan, pero no por eso dejan de cumplir con sus propósitos generales, es decir expresan un modelo objetivo común. Pudiera existir una coincidencia para adaptar soluciones de este tipo ya desarrolladas, pero tendrían no solo estar acorde con el esquema de negocio para luego establecerlos, sino contar con un modelo de implementación enfocado al uso de herramientas de código accesible y determinada flexibilidad. De todos modos se propone estudiarlas en favor de encontrar aportes que ayuden a elaborar la solución la problemática.

Algunas de las herramientas más novedosas en este ámbito son los llamados BackOffice ó trastiendas que son interfaces que tienen la funcionalidad de configurar las redes y otros recursos vinculados a la administración de los sistemas informáticos. Los indicios de desarrollar aplicaciones para gestionar estos procesos se remontan en el desarrollo de los sistemas operativos, como lo es el caso de Microsoft Windows Xp, NT, que cuentan con paneles de control donde aparecen funcionalidades de configuración de carácter público y otras de carácter exclusivo, accesibles solo por un administrador.

La herramienta BackOffice para Microsoft Windows Server 4.5 (Anexo 3) desarrollada por el este consorcio, incluye un conjunto de aplicaciones para servidores de red Windows NT Server 4.0, sistema operativo que le permite la instalación, configuración y administración de la red; Exchange Server, para la administración de correo interno y correo externo, que se relaciona al Outlook y posibilita la utilización al máximo de este producto. (6)

Aunque esta solución se basa en el proceso administrativo, está diseñado como un traje a la medida para gestionar la integración de estos procesos, específicamente en plataformas de sistemas operativos, que a pesar de pertenecer al entorno del objeto de estudio no constituye una solución recomendable, para el desarrollo de la propuesta. Sin embargo no se deja de reconocer la lógica de su negocio que pudiera aportar algunos elementos a la solución de la problemática expresada. Otro de los más relevantes impedimentos es ser un recurso bajo licencia propietaria y al igual que las plataformas, lenguajes y herramientas utilizadas para su desarrollo, lo que implica no poder adaptarlo a una

solución para gestionar la configuración en el Sistema de Planificación y Control de la Dirección General de la UCI.

Otras empresas como Antares s.a (Anexo 4) actualmente desarrolla aplicaciones de comercio electrónico en la que los usuarios realizan sus compras a través de Internet y se les ofrecen una serie de servicios que se configuran por los administradores, para que a su vez estén asociados con su compra. También el mantenimiento por parte de estos gestores y administradores de la web cuenta con utilidades y aplicaciones de uso intuitivo y personalizado en función de los requisitos del comercio. Tanto el proceso de compra, el panel de control de cliente, así como la herramienta que gestiona el BackOffice están preparadas para una comunicación encriptada entre el cliente y el servidor para ello se describen algunas de las características del BackOffice de la herramienta e-commerce (comercio electrónico) de Antares s.a (7):

- Formulario de acceso restringido
- Panel de control y seguimiento de actividad del sitio
- Registro de alertas y sucesos
- Herramienta para gestión de catálogo de productos, clientes, usuarios y contraseñas, proveedores, pedidos, suscriptores
- Información estadística de visitas, productos, productos más buscados, visitados y comprados

La herramienta de e-commerce según sus programadores está desarrollada en XHTML y PHP. Está pensada para funcionar en cualquier entorno y en cualquier servidor con soporte para Apache y PHP, con base de datos MySQL, de igual manera como se planteaba anteriormente, esta solución sugiere un modelo más próximo a la aplicación que se pretende construir, principalmente por la tecnología usada cliente – servidor lo cual propone una aplicación web. Un aspecto adicional a tener en cuenta es el uso de algunas herramientas y lenguajes de programación de código abierto, disponible sin licencias propietarias, aunque en el caso específico del gestor de base de datos utilizado en esta solución, pudiera ser sustituido por algún otro que no fuese de carácter propietario como PostgreSQL por mencionar alguno. Aún así, la forma en que fue desarrollada esta aplicación, así como su semejanza con la tecnología que implementa se ajustan un poco más la idea que se propone desarrollar en este trabajo.

En nuestro país, específicamente en la UCI se han desarrollado varias soluciones de esta naturaleza en favor de responder a las necesidades de administración en los sistemas. Entre estas puede citar el modulo de administración para el Sistema Integrado de Transportación de la UCI. Otro ejemplo con similar naturaleza es el proyecto del MINFAR de la facultad 4 de la UCI en el que se está implementando un Sistema ERP, que cuenta con un módulo de administración encargado de gestionar la seguridad de cada una de las funcionalidades que este pone a disposición de los usuarios. En este último el nivel de acceso está concebido para muchos usuarios que pueden estar asociados a uno o muchos roles y estos a su vez con un número determinado de funcionalidades del sistema. Para lograr la autenticación y configuración del sistema en dependencia del usuario que se autentique se realiza el siguiente proceso, el sistema identifica el usuario y los roles a los cuales este está vinculado para obtener la lista de funcionalidades y mostrarlas al usuario que se ha autenticado. Además de estos beneficios también cuenta con otras funcionalidades, pero no pudiera usarse ya que habría que realizar numerosas transformaciones. Otras razones implican que estos módulos no integran varios subsistemas, sino que están concebidos dentro del propio sistema y desde su interior manipulan funcionalidades de configuración.

Teniendo en cuenta la evaluación del estado del arte mediante un breve análisis de las aplicaciones existentes en el ámbito internacional, así como en la UCI, se llego a la conclusión de que estas soluciones no cumplen en su totalidad con el objetivo que se pretende desarrollar, siendo esta la motivación para implementar una aplicación informática de gestión de la configuración o BackOffice para el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI. Con el propósito de agrupar más elementos se propone evaluar algunas tendencias y tecnologías actuales que aporten algunos elementos a tener en cuenta, a lo largo de esta investigación.

1.4 Tecnologías, tendencias y evolución de la Web

La Web fue diseñada originalmente como un medio puramente informativo pero ha evolucionado hasta convertirse en un medio de aplicación, pero antes de analizar cuál ha sido su historia y evolución, es importante definir qué se entiende como tal.

Una aplicación Web es un software basado en tecnologías y estándares de W3C que provee recursos específicos tales como contenidos y servicios a través de de una interfaz de usuario (UI) a la que puede accederse utilizando un navegador Web (8).

El primer paso para diferenciar las aplicaciones Web de los tradicionales sitios Web centrados en el contenido consiste, según Bob Baxley (9), en enfocarse en el significado de la palabra aplicación. A

partir de este concepto una aplicación Web puede definirse de forma sencilla como un software al que se accede mediante un navegador Web y cuyo propósito es la realización de una o varias tareas solicitadas por un usuario. La siguiente definición aporta otros elementos importantes a tener en cuenta:

La evolución de las aplicaciones Web ha transformado los sistemas informáticos, rompiendo barreras geográficas, económicas y lógicas debido a los diferentes sistemas operativos abriendo un abanico de nuevas posibilidades. En este proceso ha existido una retroalimentación entre el desarrollo de la Web y las aplicaciones que en la misma se pueden construir. Se puede enmarcar su desarrollo en las últimas de sus cuatro grandes etapas o generaciones. (10)

El uso extendido de CGI beneficio la creación de páginas dinámicas, pero debido a sus limitantes surgieron otras tecnologías. La primera solución fue Internet Database Connector (IDC) de Microsoft, le siguió Active Server Page (ASP) que resulto ser la verdadera revolución de las paginas dinámicas. Después de esto aparecieron otras como: ColdFusion, PHP y Java Server Page (JSP). Aún así la Web permaneció en su versión 1.5 pero tuvo a su favor una serie de herramientas que hacían de las aplicaciones Web un adversario frente a las tradicionales aplicaciones de escritorio. Las páginas pertenecientes a esta generación son las más comunes en la actualidad y se caracterizan por:

- Uso de Cascading Style Sheets (CSS) y optimización del código HyperText Markup Language (HTML)
- Se orientan al usuario final, ofreciendo mayor contenido de servicios, información y comercio
- Se tienen en cuenta principios tipográficos y de organización visual de la información a través del empleo de forma coherente de colores, imágenes, tipos de letras, símbolos e iconos
- Se incorporan los principios de usabilidad y accesibilidad

Esta etapa forjo bases muy sólidas para la próxima generación de 1999 hasta nuestros días, donde la información es almacenada a partir de las páginas Web en una base de datos. Las características de las páginas actuales son:

- Se emplean nuevamente los recursos gráficos
- HTML evoluciona: se extiende el uso de tecnologías poco empleadas hasta ese momento, como CSS y surge Dynamic HTML (DHTML). Estas tecnologías generaron incompatibilidad entre distintos navegadores, a pesar de introducir considerables ventajas

- Uso de nuevas tecnologías multimedia: Se puede crear un sitio Web sin emplear HTML

Aunque el desarrollo actual llega hasta la Web 2.0, se tienen aspiraciones en las cuales ya se incursiona la Web 3.0 pretende añadirle significado a la Web y promete transformar la experiencia de los usuarios desde todos los puntos de vista para ello componentes de inteligencia artificial serán los encomendados de sentar las bases de esta etapa de desarrollo, asociándose en gran medida al término de Web semántica y aunque no es lo mismo, será una característica notable que se agregue.

La Web 4.0 (WebOS) puede que suene a ciencia ficción, pero Raymond Kurzweil¹ afirma que para el 2029 las computadoras tendrán la potencia de proceso equivalente al de un cerebro humano. La WebOS como se le atribuye, pretende convertir la Web en un sistema operativo virtual, los usuarios podrán conectarse usando un navegador y trabajar en línea del mismo modo que lo hacen hoy en sus puestos de trabajo.

La Web 2.0 es la representación de la evolución de las aplicaciones tradicionales hacia aplicaciones Web enfocadas al usuario final. Se trata de aplicaciones que generen colaboración y servicios que reemplacen las aplicaciones de escritorio. El uso de diferentes tecnologías y estándares como XHTML, la separación del contenido y el diseño a través de las hojas de estilo CSS y el uso de Ajax agregando mucha más interactividad con los usuarios.

A modo de conclusión de estas tendencias se puede expresar que el aumento de la productividad que generan las TIC en las empresas depende en gran medida del aprovechamiento que éstas hagan de ellas.

Los Servicios web

Un Servicio Web es un recurso programable y direccionable mediante una URL. La implementación de servicios web ha resultado una variante que flexibiliza el flujo de datos entre los sistemas que invocan o envían datos entre sí, sin tener en cuenta la arquitectura de los mismos, evitando complejizar las consultas a sus bases de datos así como la sobrecarga de los gestores. Es significativo recalcar que esta tecnología opera asincrónicamente, es muy utilizado para el envío y recepción de información, por su nivel de optimización, no requiere de conocimiento acerca de cómo están implementados los sistemas o aplicaciones y proporcionan un nivel alto de operatividad. Desde el punto de vista del programador el servicio no es más que una clase que se auto-define usando XML para ofrecer al

¹ Músico, empresario, inventor, escritor y científico de la computación, nacido en algún lugar del estado de Massachusetts

exterior una interfaz estándar con el que utilizar los métodos que proporciona, y accesible mediante HTTP por ser este el protocolo de aplicación utilizado por los navegadores web.

En la UCI la utilización de servicios web se hace casi imprescindible por el gran número de aplicaciones con plataformas diversas de acuerdo con las particularidades que proponen los desarrolladores. La mayor responsabilidad de estos es la integración, tiene presente diferentes vías de transporte o los diferentes protocolos como son HTTP, HTTPS, HTTP-R, BEEP, JABBER, IIOP, SMTP o FTP. Los servicios web usan muchos tipos de tecnologías con las cuales se trabajó en el desarrollo de esta aplicación, ellas son:

- XML de sus siglas en inglés (Extensible Markup Language)
- SOAP de sus siglas en inglés (Simple Object Access Protocol)
- WSDL de sus siglas en inglés (Web Service Definition Language)
- UDDI de sus siglas en inglés (Universal Description, Discovery and Integration)

Para facilitar la creación y el formateo de los mensajes SOAP que deben intercambiar las aplicaciones, se dispone de distintas utilidades y aplicaciones. Estas implementaciones incluyen utilidades para generar los mensajes a partir de componentes. Por otra parte WSDL facilita la descripción de los distintos métodos o funciones que están disponibles en el servicio web, así como su signatura, es decir, el número de argumentos o parámetros que se les debe pasar, y el tipo de dato que devolverá la función como resultado, estableciendo una equivalencia entre el documento WSDL y un contrato que especifica los servicios que el servidor se compromete a ofrecer al cliente, siempre que éste los solicite de la forma adecuada

Finalmente como su creación resulta compleja, las distintas implementaciones de SOAP permiten generar estos archivos de forma sencilla, sin necesidad de conocer los elementos y la estructura del esquema XML en el que se basan. XML es un lenguaje de etiquetas (tags) para describir datos estructurados que separa los datos y su propia definición (XSD, DTD) de la representación (XSL).

1.5 Lenguajes y herramientas para el desarrollo de la aplicación

Para desarrollar aplicaciones Web existen diferentes lenguajes de programación, tales como: C#, Perl, Java, ASP, PHP, Ruby y Python. La elección de uno de ellos debe sustentarse principalmente en las particularidades de la solución que se proponga y los conocimientos de los desarrolladores. A partir de

los lineamientos establecidos por la Dirección de Informatización en la UCI (11) documento que establece las pautas para desarrollar los sistemas en la UCI, por el cual además fueron diseñadas las aplicaciones que conforman el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI, se utiliza el lenguaje de programación PHP, el framework CodeIgniter y el modelo de datos existente soportado en el sistema gestor de base de datos (SGBD) PostgreSQL y la tecnología servidor Apache.

Apache

Apache es una de las tecnologías de servidor web más utilizado del mundo, encontrándose muy por encima de sus competidores, tanto gratuitos como comerciales. Es un software de código abierto que funciona sobre cualquier plataforma. Desde su origen ha evolucionado hasta convertirse en uno de los mejores servidores en términos de eficiencia, funcionalidad y velocidad. Apache presenta entre otras características, mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido.

¿Por qué Apache?

- Es gratuito, distribuido bajo la licencia Apache Software License, lo que permite la modificación y adaptación de su código
- Es flexible y extensible lo que facilita la posibilidad de, ampliar sus capacidades y bondades mediante módulos
- Es extremadamente popular, por lo que resulta muy fácil conseguir documentación o ayuda para su uso
- Es altamente fiable pues cerca de un 90% de los servidores con más alta disponibilidad funcionan con él
- Su eficiencia se destaca por la alta velocidad, característica fundamental para la gestión en el proceso de configuración y seguridad

PHP

Un lenguaje de programación no es más que el protocolo establecido por los humanos para ejecutar acciones en los ordenadores que generan una estructura secuencial denominada programa. Estos programas están formados por cadenas de instrucciones o códigos fuente que al ser interpretados por el ordenador emiten una respuesta.

Este lenguaje de programación en sus inicios denominado según sus siglas Personal Home Page hoy más conocido por Hipertext Pre-processor es una variante muy usada para el diseño de páginas web. La rapidez de su motor y la integración con diversos SGBD como Oracle, Informix, MySQL, PostgreSQL, Microsoft SQL Server, entre otros además de su adaptabilidad a múltiples plataformas lo han convertido en un estándar para este propósito. Entre algunas de estas características están (12):

- Simplicidad: usuarios con experiencia en Perl, C ó C++ podrán utilizarlo rápidamente
- Velocidad: alta velocidad de ejecución, bajo consumo de recursos y muy buena integración con Apache
- Estabilidad: ninguna aplicación está completamente libre de errores (bugs), pero con una amplia comunidad de programadores y usuarios corregirlos es mucho más fácil
- Posee un sofisticado manejo de variables que lo hacen muy robusto y estable
- Seguridad: permite la protección contra diversos ataques a través de diferentes niveles de seguridad que pueden ser configurados desde el archivo.ini

PHP es un lenguaje de programación de código abierto que actualmente se puede ejecutar sobre tecnologías de servidor web como Apache, IIS, AOLServer, Roxen y THTTPD. PHP corre en casi cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en aproximadamente 25 plataformas, incluyendo diferentes versiones de Unix, Windows (95,98,NT,ME,2000,XP,etc) y Macs.

Contiene soporte integrado para SOAP, lo que resulta muy provechoso cuando se requiere trabajar con servicios Web, como es en este caso. Aunque PHP no obliga a seguir una determinada metodología a la hora de programar, existen formas de obtener código ordenado, estructurado y manejable. Una de ellas es el uso de un framework.

Los entornos integrados de desarrollo (IDEs)

Dentro del desarrollo de aplicaciones Web, es muy importante, elegir dos aspectos, por una parte un lenguaje y por otra un IDE (Entorno de Desarrollo Integrado por sus siglas en ingles), esta de mas decir de que PHP es un sólido lenguaje de programación web, que ofrece muchísimas características que se expresadas anteriormente, pero estas pueden ser más notables si se utilizan estas

herramientas que favorecen considerablemente al desarrollador a la hora de escribir el código de una aplicación.

Sin embargo en el segundo aspecto es muy complicado decidirse por uno o por otro, puesto que en el mundo existen diversos IDEs que brindan soporte para PHP, algunos son Libres otros propietarios, entre ellos los más populares para PHP.

Eclipse PDT

Es un excelente IDE, para PHP, que está basado en el popular IDE Eclipse, Este es uno de los primeros IDEs, de código abierto (Open Source) para PHP, cuya comunidad de desarrolladores crece diariamente y tiene apoyo de Zend. Este cuenta con un coloreado de sintaxis, autocompletado de código, es multiplataforma, es extensible a través de plugins, posee un navegador interno para ver los resultados a medida que se desarrollan los proyectos, permite editar documentos XHTML, CSS y JavaScript, generación de getters y setters automatizado, pero ha resultado complicado al configurar, debido a que existe muy poca documentación sobre este IDE y consume mucho recurso del sistema puesto que está desarrollado en Java.

Zend

Zend Studio, es uno de los IDEs, más completos para desarrollar en PHP, dirigido a la elaboración de aplicaciones web en este lenguaje, muy utilizado por la comunidad del PHP a nivel mundial, ofrece depuración de código, además soporta lenguajes utilizados en la web como son javascript, css, html, entre otros. Es un editor multiplataforma elaborado en software libre, el editor ofrece herramientas de programación, ofrece ayuda contextual para las sentencias del lenguaje y otras muchas ventajas. Su última versión, ya no está basada en la plataforma Java, su interfaz es casi similar al de Eclipse PDT, existe dos variantes de este IDE, Zend Studio, y otro que está disponible como plugin para Eclipse (Zend Studio para Eclipse).

Gedit

Gedit es el editor de texto libre oficial del escritorio GNOME. Este editor de software libre, de código abierto y gratuito está publicado bajo la licencia GPL. Además de las funcionalidades básicas que son habituales en un editor de texto (copiar, cortar, pegar, imprimir, etc.) Gedit incorpora, entre otras, las siguientes funcionalidades:

- Compatibilidad con textos internacionalizados, usando la codificación UTF-8
- Coloreado del texto según la sintaxis de varios lenguajes de programación

- Incorporación de plugins para ampliar las funcionalidades básicas del programa
- Numeración de líneas
- Búsqueda y reemplazo de texto
- Edición de archivos remotamente
- Copia de seguridad de los archivos sobre los que se trabaja

CodeIgniter

Un framework no es más que una herramienta para simplificar y acelerar el proceso de desarrollo de una aplicación automatizando patrones que solucionan tareas comunes mediante el encapsulado de operaciones complejas en instrucciones sencillas. Las características que fundamentan el éxito de su uso (Anexo 5) y el flujo de datos del framework (Anexo 6), son algunas de las buenas razones que sugieren usarlo. De acuerdo con lo anterior se muestra la descripción de cada componente (13):

- El index.php sirve como controlador frontal, inicializando los recursos básicos necesarios para correr CodeIgniter
- El Router examina la petición HTTP para determinar que debe ser hecho con él
- Si un archivo de caché existe, es enviado directamente al explorador, sobrepasando el sistema de ejecución normal
- Seguridad. Antes que el controlador sea cargado, la petición HTTP y cualquier dato suministrado por el usuario son filtrados por seguridad
- El controlador carga los modelos, librerías, plugins, asistentes y cualquier otro recurso necesario para procesar la petición específica
- La Vista finalizada es presentada entonces enviada al explorador web para ser vista. Si el cacheo está habilitado, la vista es cacheada primero para que las peticiones subsecuentes puedan ser servidas

A pesar de que existen otros frameworks similares como Symfony o Cake herramientas no propietarias, de uso respaldado en los lineamientos, CodeIgniter ha resultado la elección adecuada a esta aplicación. La variante de explotar el patrón MVC en forma pasiva, la estandarización con algunas

aplicaciones web como el Sistema de reportes, Sistema de reservaciones y el Sistema de planificación del menú, y su fácil aprendizaje son razones evidentes para usarlo.

Sistemas Gestores de Bases de datos (SGBD). PostgreSQL

Los SGBD son aquellos sistemas que están encargados de almacenar los datos y contienen interfaces claves para la definición manipulación y consulta de datos de forma práctica y eficiente. Estos sistemas están destinados al cumplimiento de distintos objetivos que deben cumplir los SGBD:

- **Abstracción de la información:** Ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Si una base de datos ocupa uno o cientos de archivos, este hecho se hace transparente al usuario de forma tal que existan *niveles de abstracción*.
- **Consistencia:** En aquellos casos en los que no se ha logrado eliminar la redundancia, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- **Seguridad:** Se debe garantizar que esta información se encuentra segura de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías de permisos.
- **Manejo de Transacciones:** Están provistos de mecanismos para programar las modificaciones de los datos de una forma más simple que las aplicaciones que lo usan.
- **Tiempo de respuesta:** Tiempo que el SGBD tarda en darnos la información solicitada y en almacenar los cambios realizados.

Entre los SGBD más robustos están Oracle, Microsoft SQL Server, MySQL, SQLite, PostgreSQL, ect, aunque en el momento de seleccionar uno de ellos debe tenerse en cuenta ventajas y licencias de uso. Para el desarrollo de esta aplicación fundamentado en que ya existe un gestor para todo el Sistema de Planificación y Control de la UCI, se hace referencia a las características y ventajas del mismo.

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. Como muchos otros proyectos de software libre, el desarrollo de PostgreSQL es manejado por una comunidad de desarrolladores y organizaciones comerciales las cuales trabajan en su desarrollo, dicha comunidad es denominada el PostgreSQL Global Development Group (PGDG) (14). Algunas de sus principales características son, entre otras: la alta concurrencia

permitiendo que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos y amplia variedad de tipos nativos brindando soporte para:

- Números de precisión arbitraria
- Texto de largo ilimitado
- Figuras geométricas (con una variedad de funciones asociadas)
- Direcciones IP (IPv4 e IPv6)
- Bloques de direcciones estilo CIDR
- Direcciones MAC
- Arrays

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indizables gracias a la infraestructura de PostgreSQL. Algunos ejemplos son los tipos de datos creados por el proyecto PostGIS.

Los disparadores (triggers en inglés): se define en una acción específica basada en algo ocurriente dentro de la base de datos. En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:

- El nombre del trigger o disparador
- La frecuencia de la ejecución
- El momento en que el disparador debe arrancar
- La función que podría ser llamada
- El evento del disparador deberá activarse en la tabla donde el disparador se activara

Entonces combinando estas seis características, PostgreSQL le permitirá crear una amplia funcionalidad a través de su sistema de activación de disparadores:

- Vistas
- Integridad transaccional
- Herencia de tablas
- Tipos de datos y operaciones geométricas

Patrón arquitectónico MVC

El patrón arquitectónico conocido como Modelo-Vista-Controlador (MVC) separa el modelado del dominio, la presentación y las acciones basadas en datos ingresados por el usuario en tres clases diferentes:

- Modelo: administra el comportamiento y los datos del dominio de aplicación, responde a requerimientos de información sobre su estado (usualmente formulados desde la vista) y responde a instrucciones de cambiar el estado (habitualmente desde el controlador)
- Vista: Maneja la visualización de la información
- Controlador: Interpreta las acciones del ratón y el teclado, informando al modelo y/o a la vista para que cambien según resulte apropiado
- Tanto la vista como el controlador dependen del modelo, el cual no depende de las otras clases

Esta separación permite construir y probar el modelo independientemente de la representación visual. La separación entre vista y controlador puede ser secundaria en aplicaciones de clientes ricos y, de hecho, muchos frameworks de interfaz implementan ambos roles en un solo objeto (15).

1.6 Metodologías de desarrollo de software. Herramienta y lenguaje de modelado

El desarrollo de software, desde sus inicios está basado en varios métodos, procedimientos y normas que comprenden las actividades que se llevan a cabo, para obtener un exitoso producto de software. Este conjunto de elementos conforman las llamadas metodologías de desarrollo de software. La misión fundamental de estas metodologías es el seguimiento del ciclo de vida durante cada etapa en el proceso de creación de cualquier aplicación.

Todo desarrollo de software es riesgoso y difícil de controlar. Sí no se aplica una metodología adecuada, solo se obtienen clientes insatisfechos y desarrolladores aún más insatisfechos, en síntesis

el fracaso. Sin embargo, en ocasiones no se evalúa como utilizar una metodología, sobre todo cuando se trata de proyectos pequeños de dos o tres meses. Este tipo de proyectos separa rápidamente lo aplicativo a cada proceso, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo (16).

Las metodologías ágiles forman parte del movimiento de desarrollo ágil de software conocido anteriormente como metodologías livianas, que se basan en la adaptabilidad de cualquier cambio como medio para aumentar las posibilidades de éxito de un proyecto. Se le denomina ágil como la habilidad de responder de forma versátil al cambio para maximizar los beneficios. Intentan evitar los tortuosos y burocráticos caminos de las metodologías tradicionales enfocándose en la gente y los resultados. Las metodologías ágiles se basan en los siguientes principios (17):

- Realizar entregas cortas en el tiempo y continuas
- Dar la bienvenida a los cambios
- Entregas periódicas y frecuentes que funcionen
- Los clientes forman parte del equipo de desarrollo
- Equipo con individuos motivados. Darles para ello el ambiente, apoyo y confianza

La comunicación directa es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo, cualquier otra vía entorpecerá la comprensión entre sus integrantes. La regla o medida principal de progreso es construir un software que funcione, teniendo como premisa un desarrollo sostenible y como elemento vital debe existir paz y armonía en el equipo para lograr el éxito. Entre las más conocidas están:

XP (Extreme Programming)	SCRUM
Methodology Crystal	Dynamic Systems Development Method (DSDM)
Feature -Driven Development (FDD)	Adaptive Software Development (ASD)
Lean Development (LD)	AUP

Existen otras metodologías de desarrollo de software que no precisamente tienen un enfoque ágil, pero si forman parte del modelo ideal para desarrollar cualquier proyecto de desarrollo de sistemas, un

ejemplo clásico es Rational Unified Process (RUP) la cual a través de un flujo de trabajo donde se llevan a cabo la ejecución de cuatro fases que son Inicio, Elaboración, Construcción y Transición logra obtener un producto basado en calidad. Esta metodología es usada en muchos proyectos de gran envergadura y que a su vez cuentan con grandes equipos que ejecutan diversas actividades en cada fase. A pesar de su modelo ideal para procesos de software, además de constituir para muchos desarrolladores la madre de todas las metodologías, no es recomendable usarla en soluciones como la que se pretende desarrollar en esta investigación, por algunas desventajas como son el exceso de documentación que no aporta valores a la solución y retrasa la entrega del producto. Arraigado a esto se propone valorar las metodologías Scrum y XP a favor de obtener lo más favorable de cada una de ellas.

El proceso de software con Scrum

Define un marco para la gestión de proyectos, que se ha utilizado con éxito durante los últimos 10 años. Está especialmente indicada para proyectos con un rápido cambio de requisitos. Su enfoque principal es desarrollar el software mediante iteraciones, denominadas sprints, con una duración de 30 días. El resultado de cada sprint es un incremento ejecutable que se muestra al cliente. La otra perspectiva son las reuniones a lo largo proyecto, entre ellas destaca la reunión diaria de 15 minutos del equipo de desarrollo para coordinación e integración.

Esta metodología no se concibe independiente, sino como complemento a otras metodologías como XP y MSF. Como método, Scrum enfatiza valores y prácticas de gestión, sin pronunciarse sobre requerimientos, implementación y demás cuestiones técnicas; expresando su deliberada insuficiencia y complementariedad. Scrum promueve además algunos valores como equipos auto-dirigidos y auto-organizados.

No hay Director que decida; la excepción es el Scrum Master que debe ser 50% programador y que resuelve problemas, pero no manda. Una vez elegida una tarea, no se agrega trabajo extra. En caso que se agregue algo, se recomienda quitar alguna otra cosa. Encuentros diarios con las tres preguntas siguientes:

- ¿Qué es lo que se hizo el día anterior?
- ¿Qué es lo que se va a hacer hoy?
- ¿Qué impedimentos tengo para realizar mi trabajo?

Como conclusión las ventajas más notables consisten en realizar iteraciones de treinta días, aunque se admite que sean más frecuentes, otra sería la demostración a participantes externos al fin de cada iteración y por último realizar al principio de cada iteración, el planeamiento adaptativo guiado por el cliente.

El proceso de software utilizando Programación Extrema (XP)

Es una de las metodologías de desarrollo de software más exitosas en la actualidad, utilizadas para proyectos de corto plazo y equipos pequeños. La metodología consiste en una programación ágil y cuya particularidad es tener como parte del equipo, al usuario final. Está basada en:

- Pruebas Unitarias: pruebas realizadas a los requerimientos críticos, de tal modo que se puedan prever algunas fallas que pudieran ocurrir. Es como si se adelantara a obtener los posibles errores
- Refactorización: se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio
- Programación en parejas: una particularidad de esta metodología es que propone la programación en parejas, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. Es como el chofer y el copiloto: mientras uno conduce, el otro consulta el mapa

Las ventajas más notables de XP son:

- Empieza en pequeño y añade funcionalidad con retroalimentación continua
- El manejo del cambio se convierte en objetivo del proceso
- El costo del cambio no depende de la fase o etapa
- No introduce funcionalidades antes que sean necesarias

Para el desarrollo de este trabajo se propone la fusión de dos metodologías ágiles enunciadas anteriormente, SCRUM y XP para la planificación y el desarrollo respectivamente. Esta metodología puesta en práctica en muchos proyectos de élite mundial como Google, ORACLE, Yahoo, SUN Microsystems y Microsoft ha demostrado grandes ventajas en la orientación a cambios, entregas

continuas y rápidas, así como otros elementos que justifican su implementación para un proyecto como en el que se enmarca la solución que propone el presente trabajo. Con esto se logra llegar a la entrega de satisfactorios resultados para los clientes, un problema que consume a otros equipos de desarrollo de la UCI.

La fusión Scrum y XP

SXP es una metodología compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo. SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, constituyendo esto una guía para supervisar eficientemente un proyecto. La metodología XP que prioritariamente está encaminada para la programación; consiste en una implementación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final. Esto constituye uno de los requisitos más relevantes para llegar al éxito, experiencia que fue probada en el proyecto que desarrollo la gran mayoría de las aplicaciones del Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI.

La propuesta de elaborar un híbrido entre las metodologías Scrum y XP fue avalada por el departamento de calidad de la UCI, a partir de su explotación en desarrollo de aplicaciones como SISCLON de proyectos Agiles Unicornios en la facultad 10 de la UCI, entre otros. Como se expresa anteriormente, la propia experiencia del proyecto Alimentación donde se enmarca la solución que se pretende implementar en este caso ha tenido resultados significativos en el propósito de entregar productos de software funcionales en tiempo record y con escasos recursos. Para entrar en detalles a continuación se abordan las 4 fases principales de la metodología SXP:

- Planificación-Definición: donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto
- Desarrollo: es donde se realiza la implementación del sistema hasta que esté listo para ser entregado
- Entrega: es la puesta en marcha; y por último
- Mantenimiento: es la fase donde se realiza el soporte para el cliente

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, el diseño, la implementación, pruebas, entre otras, generan artefactos para documentar todo el proceso. Otra ventaja son las entregas frecuentes y la refactorización continua, lo que permite mejorar el diseño cada vez que se le añada una nueva funcionalidad. SXP está especialmente indicada para proyectos de pequeños equipos de trabajo, donde existe un alto riesgo técnico, aunque se orienta a una entrega rápida de resultados y una alta flexibilidad.

Lenguaje unificado de modelado UML. La herramienta Visual Paradigm

El Lenguaje Unificado de Modelado es un protocolo que proporciona un vocabulario y reglas para permitir una comunicación entre los entornos de clientes y desarrolladores de software; en este caso este lenguaje se ajusta en la representación gráfica de un sistema. UML contribuye a la comunicación de las ideas entre clientes, analistas y desarrolladores además de servir de apoyo en los procesos de análisis de un problema. UML se ha convertido en ese estándar tan ansiado para representar y modelar la información con la que se trabaja en las fases de planificación y definición; se indica cómo crear y leer los modelos que posteriormente serán interpretados por los arquitectos y programadores del software para generar el código fuente de los sistemas informáticos. Los objetivos de UML son muchos, pero se pueden sintetizar sus funciones:

- Visualizar: permite expresar de una forma gráfica un sistema de forma que se pueda entender
- Especificar: permite especificar cuáles son las características de un sistema antes de su construcción
- Construir a partir de los modelos especificados se pueden construir los sistemas diseñados
- Documentar: los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión

Visual Paradigm es una herramienta Case con un enfoque de negocio que genera un software de calidad, tiene la particularidad de ser un lenguaje estándar común para todo el equipo de desarrollo que facilita la comunicación entre todos sus integrantes. Posee capacidades de ingeniería directa (versión profesional) e inversa, modelo y código que permanece sincronizado en todo el ciclo de desarrollo, disponibilidad de múltiples versiones, disponibilidad en múltiples plataformas. Como ventaja fundamental es una herramienta no propietaria y de fácil adquisición todo lo contrario de Rational Rose.

Existen varias versiones de Visual Paradigm para UML. La gratuita no permite realizar ingeniería inversa, pero permite crear diagramas y generar código a partir de ellos. Las versiones comerciales difieren entre ellas por su funcionalidad y su capacidad de integración con otras herramientas; también hay disponible un programa para acceder a las herramientas a un precio bajo si se desea emplear con fines académicos. En el presente trabajo se propone utilizar esta herramienta apoyados en las facilidades y ventajas expresadas anteriormente.

Conclusiones del capítulo

En este capítulo se expuso de forma general los aspectos teóricos a tener en cuenta para el desarrollo de un sistema de administración. Con el propósito de situarlo en el universo de la configuración y la seguridad, se mostraron diferentes sistemas existentes, sus características, funcionamiento y los métodos administrativos empleados.

Además se evaluaron y valoraron algunas tendencias y tecnologías empleadas en los sistemas de informáticos. Se realizó una evaluación de las herramientas y lenguajes propuestos a utilizar para el diseño, implementación y prueba, así como la metodología de desarrollo de software utilizada.

CAPÍTULO 2: FUNDAMENTOS DE LA SOLUCIÓN PROPUESTA

La descripción de la problemática, los objetivos estratégicos así como el estudio del flujo actual de los procesos enmarcados en el entorno donde se desarrollara; la solución propuesta, incide en el cumplimiento de las necesidades expresadas por los clientes. El reconocimiento e identificación de los procesos propuestos para la automatización, respaldados en la captura de los requisitos con los que contará el software, forman parte de las bases fundamentales, para garantizar un producto con rapidez y la calidad requerida.

Este capítulo, expresa un análisis profundo de la problemática existente en el Sistema de planificación y control de la Dirección General de Alimentos de la UCI, donde además se realiza una descripción general de la propuesta para desarrollar una aplicación informática que cumpla con el detallado levantamiento de los requerimientos funcionales y no funcionales identificados por el analista en coordinación con el cliente, el cual forma parte del equipo de desarrollo. En este capítulo quedan expresados los modelos de clases para el diseño, modelo de persistencia y estructura de la base de datos.

2.1 Objetivos estratégicos. Análisis crítico de la ejecución de los procesos

El proyecto UCIA Alimentación surge en el año 2005, donde se detectan un grupo de deficiencias en la planificación y el control del servicio de comedores de la UCI. Estos problemas fundamentan la necesidad de automatizar un modelo de procesos que permita aumentar los niveles de eficiencia de este servicio.

Como parte del desarrollo de esta etapa inicial en Enero del 2008 se desplegaron cuatro soluciones informáticas que constituyen la base del desarrollo futuro del sistema. Este sistema constituye una herramienta de gestión integral para planificar y controlar el servicio de comedores de la UCI. A través de su uso pueden realizarse las asignaciones de clientes a las distintas unidades de servicio, se controlan los consumos de la producción elaborada y de las materias primas, además permite consultar el estado de los indicadores de control con el uso de reportes automatizados.

A pesar de haber logrado automatizar un gran porcentaje de las actividades del proceso expuestas en cada una de las aplicaciones que conforman el Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI estas aplicaciones gestionan su funcionamiento de forma independiente, lo que representa un entorpecimiento para su configuración.

Con el propósito de recoger criterios acerca del funcionamiento interno de las aplicaciones, se toma como referencia los encuentros realizados en el equipo de desarrollo donde se entrevistaron individualmente todos sus miembros y los principales clientes de cada una de las aplicaciones. Los criterios recogidos permitieron definir los requerimientos comunes de configuración y establecer el nivel de seguridad que permita integrar a partir de una aplicación informática cumpliendo con los procedimientos fundamentales que siguen las aplicaciones para gestionar estos indicadores. La recopilación de estos criterios concluyó que de forma general, los procesos fluyen de la siguiente manera:

Para el registro de los usuarios las aplicaciones necesitan tener definido en su código de fuente quiénes pueden acceder a ellas, de esta forma no es fundamental identificar los privilegios que tendrán dentro de la misma, pues la autenticación absorbe en gran medida la autorización. Este comportamiento es similar en todos los casos, sin embargo, la manera en que se gestiona el registro de usuarios y el proceso de permitir y/o denegar permisos a los mismos, no está estandarizado en todas las aplicaciones del sistema, a pesar de que es posible hacerlo. Por tal razón no está delimitado un modelo general, donde se describan los procedimientos que actualmente siguen dichas aplicaciones, para registrar a los usuarios y definir sus privilegios.

Respecto a la autenticación, el usuario que intenta acceder a una aplicación es verificado para comprobar si es quién dice ser según su nombre de usuario y contraseña, pero no existe una forma en que los usuarios comunes entre aplicaciones hereden la accesibilidad de manera tal que teniendo el mismo rol en una de las aplicaciones, pueda acceder a otra sin tener que autenticarse nuevamente.

A su vez no existe la forma de restringir la accesibilidad a estas aplicaciones teniendo en cuenta desde donde un usuario intenta hacerlo, poniendo en riesgo el resultado del flujo informativo en aplicaciones como es el caso de Control de Acceso, donde se identificó el mayor riesgo para este aspecto. Esto sería un parámetro de gran importancia, que además de su aporte al control de trazabilidad de acciones sobre las aplicaciones, pudiera disminuir el flujo de peticiones de accesibilidad si un usuario intenta acceder desde una estación de trabajo no registrada. Esta última indica que la mayoría de las aplicaciones no gestionan la trazabilidad de las acciones que ejecutan los usuarios al no contar con historiales de accesos y cambios.

La falta de esta información genera inadecuación al proceder en caso de ataques o violaciones. Unido a ello, tampoco existe un control centralizado y actualizado de las personas que tienen acceso a una

determinada aplicación, ni la cantidad de estas que pudiesen acceder con un usuario determinado, considerándose ésta, información sumamente valiosa para la seguridad informática.

En conclusión la ejecución actual de los procesos ratifica que la configuración de las aplicaciones se realiza accediendo a cada una de ellas aisladamente lo que impide agrupar los usuarios en roles, asignar dichos roles a funcionalidades en las aplicaciones, asociar puestos a aplicaciones, como tampoco es posible auditar y conocer todo tipo de evidencia en caso que se requiera.

2.2 Propuesta de integración

Insertar una aplicación informática para gestionar la configuración en el Sistema de Planificación y Control de la Dirección General de Alimentos, conllevaría a una adaptación en el código de las aplicaciones existentes, mediante la creación de una instancia del servicio AdministracionWS. Esta instancia permitirá consultar los métodos declarados en el servicio web, validando los datos enviados desde que el usuario inicia la interacción en una de las aplicaciones. Antes de continuar con la descripción de este flujo, se debe tener en cuenta que la instancia de AdministracionWS declarada en cada una de las aplicaciones debe contener como parámetro un identificador único, que le servirá para ser reconocida por el servicio al invocar cualquiera de los métodos del mismo, este identificador funcionara como una llave para poder acceder a él, lo que evitaría al servicio atender a llamadas no deseadas.

La primera validación cuando el usuario intenta acceder a la aplicación, se realiza por parte de la aplicación enviando como parámetros su identificador, conjuntamente con el IP desde donde se intenta ejecutarla. Como efecto el servicio AdministraciónWS recibirá estos parámetros a través del método ObtenerAplicacionDadold, y validará en la base de datos si está siendo encuestado por una aplicación conocida. Con el mismo parámetro IdAplicacion el servicio validará en la base de datos, si la dirección de IP está asociada con el mismo. Como respuesta al mensaje el servicio AdministraciónWS enviará una confirmación de autorización, para que la aplicación pueda construir la interfaz de autenticación de usuario.

Posteriormente una vez validada la dirección de IP, desde donde se intenta acceder y construida por parte de la aplicación la interfaz de autenticación, nuevamente la misma envía como parámetros su identificador, el nombre de usuario y la contraseña, esta ultima en el formato de encriptación md5. El servicio AdministraciónWS recibe este mensaje a través del método ObtenerUsuariodadoNombre y retorna a través del método ObtenerRolDeUsuarios el rol que tiene asociado el usuario autenticado.

Finalmente conocido el rol se obtienen a partir del método `ObtenerFuncionalidadesDadoldRol`, las funcionalidades o métodos que pudiera ejecutar el usuario según su rol se construirá en la aplicación la interfaz que contiene un menú y el usuario realizara las distintas acciones.

Como conclusión de esta propuesta de integración se evaluó que la misma no sólo responde a solucionar los problemas analizados en el flujo actual de los procesos, sino que será muy provechoso para la configuración de las aplicaciones logrando agilizar la configuración de las aplicaciones y reduciendo considerablemente los riesgos de seguridad.

La importancia del flujo informativo, los recursos que se gestionan y los riesgos a que están sometidas todas estas aplicaciones, hace necesario que sean soberanamente protegidas. Debido a la ausencia de una aplicación informática que gestione la configuración como un todo se hace la siguiente propuesta

2.3 Propuesta de la aplicación

La solución informática que se propone en la presente investigación sugiere la construcción de una aplicación y un servicio web. La aplicación web funciona, permite establecer una estandarización en el control de la configuración y restringe la seguridad. El servicio web es parte de la clase de acceso a datos de la aplicación de configuración, por lo que contiene métodos muy particulares de la misma y los que son consumidos por las demás aplicaciones, para conocer las configuraciones declaradas para cada una de ellas. La implementación de esta variante utilizando un servicio web como parte de la capa de acceso a datos, representa una novedad en la concepción arquitectónica de este tipo de aplicaciones para gestionar las configuraciones de varios módulos de un sistema, por su carácter flexible y adaptable al cambio y como ventaja evitará transformaciones más engorrosas para alcanzar el objetivo propuesto.

La aplicación cuenta con una interfaz de presentación que describe genéricamente todas sus funcionalidades, antes de interactuar con estas, los usuarios con el privilegio de administración se autentican y en caso de ser satisfactorios los datos del usuario y contraseña, se desplegara en otra vista el menú de funcionalidades, desde la cual se puede acceder a las actividades de configuración. Primeramente se gestionan las aplicaciones que formaran parte del esquema administrativo, lo que implica poder insertarlas, editarlas y eliminarlas, de igual forma sucede para las funcionalidades solo que en este caso se le asocia la aplicación a la cual pertenecen. La funcionalidad gestionar puesto además de declarar todas las direcciones físicas de red, facilita la asociación de dichas direcciones con las aplicaciones, estableciendo un nivel de accesibilidad no solo enfocada al usuario que accede,

sino a la estación de trabajo desde donde lo hace. El menú además permite a través de formularios gestionar los roles y asociarlos al crear usuarios locales o del dominio UCI, también cuenta con una funcionalidad que permitirá en una vista mostrar un informe de trazas facilitando la auditabilidad de las aplicaciones.

El desarrollo de esta aplicación no constituye en principio ser parte del negocio del Sistema de Planificación y Control de la Dirección General de Alimentos, por lo que implementarlo en estos momentos conlleva a generar un modelo que nos permita reconocer el entorno en que se desenvuelven las aplicaciones y de esta forma definir algunos conceptos, este artefacto se conoce como Modelo de Dominio.

Modelo de Dominio

Debido a la ausencia de un procedimiento detallado, que exprese la forma en que debiera ejecutarse este proceso y teniendo en cuenta que no representa una alta complejidad, se propone realizar un modelo de dominio en el que se definen los conceptos relevantes que existen y los eventos que suceden en el entorno en el que trabajará el sistema. Esto facilita agrupar concretamente las funcionalidades que conlleven al cumplimiento del objetivo general planteado.

Conceptos y entidades principales:

Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI: Solución informática que abarca un conjunto de aplicaciones que automatizan el proceso de Planificación y Control de la Dirección General de Alimentos de la UCI.

Aplicación: Aplicaciones que conforman el Sistema de Planificación y Control de la Dirección General de Alimentos y que responden a las diferentes actividades del negocio.

Usuario: persona que juega un determinado rol para ejecutar una determinada funcionalidad en una aplicación que pertenece a la solución informática.

Usuario Local: usuario propio de una aplicación que pertenece a la solución informática cuyos permisos, están limitados estrictamente a las funcionalidades de dicha aplicación.

Usuario Directorio: usuario que pertenece al Directorio de Personas (LDAP).

Rol: entidad o forma que agrupa a los usuarios con características semejantes según su desempeño dentro de una aplicación, atendiendo a los privilegios de acceso a los diferentes recursos de la misma.

Directorio de Personas (LDAP): entidad encargada de la gestión de los usuarios de dominio que se encuentran en un directorio activo. Contiene las funcionalidades necesarias para brindar los servicios

de autenticación. La comunicación es mediante LDAP v3 ("Lightweight Directory Access Protocol "). En la figura se muestra el modelo de dominio que correspondiente.

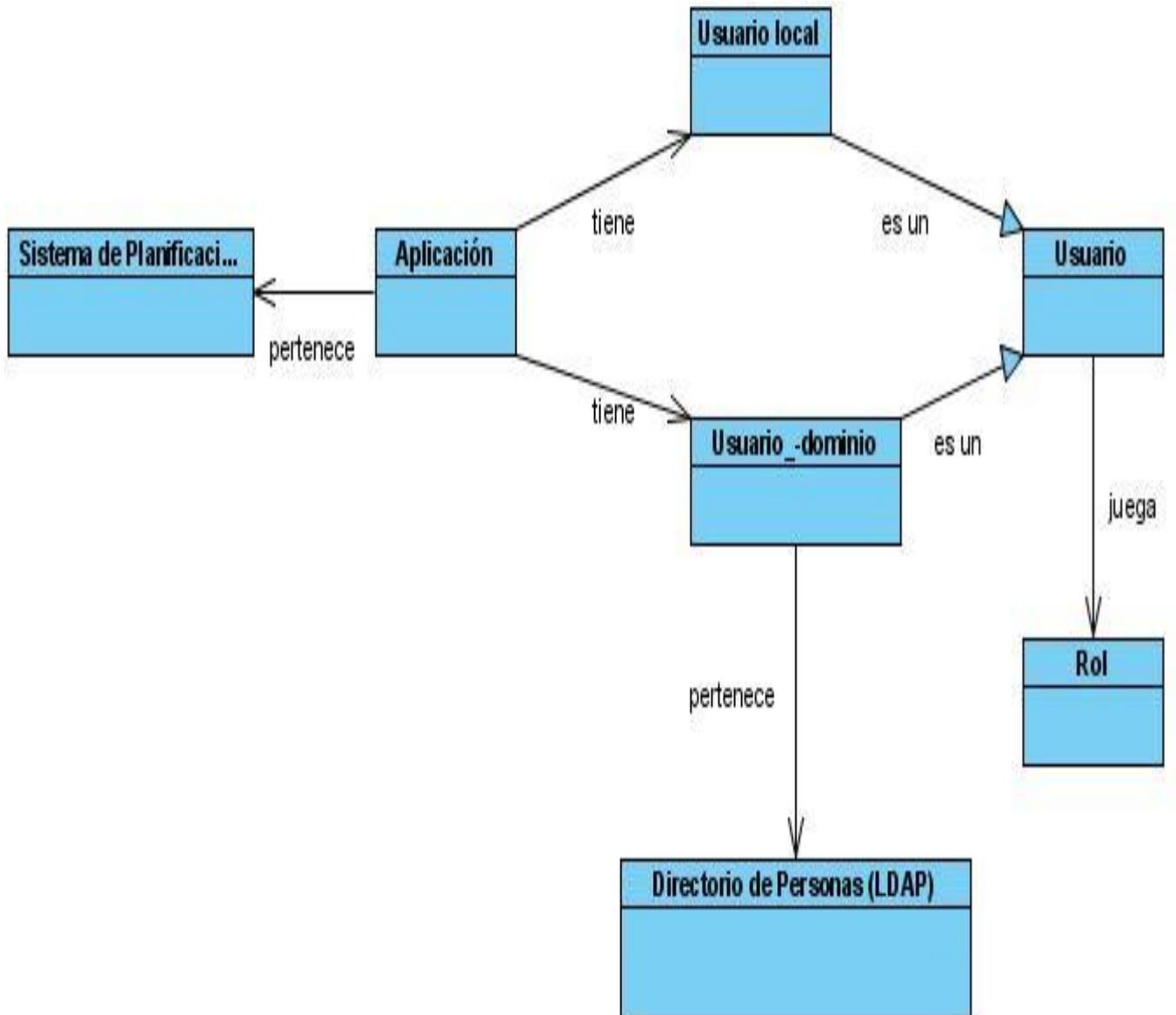


Figura 2.1 Diagrama del Modelo de Dominio

Lista de Reserva del producto (LRP)

Teniendo en cuenta la previsión de esta propuesta se procede a la elaboración de la (LRP), lista priorizada que define el trabajo que se va a realizar en el proyecto. "Refleja los requisitos en lenguaje del cliente y sugiere los más importantes que casi siempre son más que suficientes para un Sprint (Iteración)" (18). Esta lista puede crecer y modificarse a medida que se obtiene más conocimiento acerca del producto y del cliente. Con la restricción de que solo puede cambiarse entre Sprint. Para el Sprint inicial la LRP cuenta con los siguientes elementos:

Asignado a	Ítem *	Descripción	Estimación	Estimado por
Prioridad Muy alta				
Vladimir	1	Gestionar las aplicaciones del sistema		Vladimir
	2	Gestionar las funcionalidades de aplicaciones para grupos de usuarios		
	3	Declarar los grupos de usuarios con distintos niveles y privilegios para desempeñar funcionalidades en las aplicaciones del sistema		
	4	Declarar los usuarios que de acuerdo al grupo al que estén asignados realizaran diferentes acciones en una o varias aplicaciones del sistema		
	5	Restringir el acceso a las aplicaciones desde las estaciones de trabajo		
	6	El sistema debe permitir visualizar todas las acciones que realizan los usuarios en las aplicaciones.		
	7	El sistema debe permitir autenticarse		

Tabla 2.1 Lista de Reserva del producto

Historias de usuario

Las historias de usuario en lo adelante (HU) “son la técnica utilizada en la metodología SXP para especificar la LRP, lo que equivaldría a los casos de uso en RUP, aunque esto no implica que las HU sean casos de uso”². Para la propuesta de esta aplicación se han definido a partir de la misma las siguientes HU:

² Idem 18

Historia de Usuario

Número: 1	Nombre Historia de Usuario: Gestionar aplicación
Usuario: Vladimir	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto

Descripción: Presenta un formulario que permite insertar una aplicación en el sistema y hacer una breve descripción. Permite además mostrar, modificar y eliminar una aplicación que se haya creado.

Prototipo de interface:

The first screenshot shows the 'Registrar' form. The second screenshot shows the following table of registered applications:

Nombre	Activo	Acción
dgreguas	<input checked="" type="checkbox"/>	editar eliminar
app1	<input checked="" type="checkbox"/>	editar eliminar
nombre1	<input checked="" type="checkbox"/>	editar eliminar
yanet	<input checked="" type="checkbox"/>	editar eliminar
control de acceso	<input checked="" type="checkbox"/>	editar eliminar
ejemplo	<input checked="" type="checkbox"/>	editar eliminar
frf	<input checked="" type="checkbox"/>	editar eliminar
oiuygiu	<input checked="" type="checkbox"/>	editar eliminar

Tabla 2.2 Historia de usuario Gestionar aplicación

Historia de Usuario	
Número: 2	Nombre Historia de Usuario: Gestionar funcionalidad
Usuario: Vladimir	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Descripción: Presenta formulario que permite insertar una funcionalidad asociada a una aplicación creada y hacer una breve descripción. Permite además mostrar, modificar y eliminar una funcionalidad que se haya creado.	
Prototipo de interface:	

The top screenshot shows a web application interface titled 'Administración'. The date is '18 de Junio del 2009 [[127.0.0.1]]' and the user is 'Vladimiro'. A sidebar on the left contains navigation links: 'Gestionar Aplicación' (Registrar, Listar), 'Gestionar Funcionalidad' (Crear, Listar), 'Gestionar Rol' (Crear, Funcionalidad, Listar), 'Gestionar Usuario' (Crear, Listar), and 'Gestionar Puesto' (Crear, Listar). The main content area displays a 'Funcionalidad' form with a dropdown for 'Aplicación' (set to '[---Seleccione---]'), a text input for 'Nombre De Funcionalidad', and a larger text area for 'Descripción De Funcionalidad'. 'Cancelar' and 'Registrar' buttons are at the bottom right.

The bottom screenshot shows the same interface with the date '25 de Marzo de 2009'. The sidebar is identical. The main content area displays a table of existing functionalities:

Funcionalidad		
APP1		
Nombre	Activo	Acción
funcionalidad1	<input checked="" type="checkbox"/>	editar eliminar
FRRF		
Nombre	Activo	Acción
ww	<input type="checkbox"/>	editar eliminar

Copyright © 2008. Universidad de las Ciencias Informáticas. Todos los derechos reservados. | cargado en 0.2803 seg

Tabla 2.3 Historia de usuario Gestionar funcionalidad

Historia de Usuario	
Número: 3	Nombre Historia de Usuario: Gestionar rol
Usuario: Vladimir	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Descripción: Presenta un formulario que permite insertar un rol en el sistema, hacer una descripción de uso. Permite además mostrar, modificar y eliminar el rol que se haya creado.	
Prototipo de interface:	

The top screenshot shows a web application interface for role management. The page title is 'Administración'. The date is '18 de Junio del 2009' and the version is '[1.27.0.0.1]'. The user is 'Vladimir'. The left sidebar contains a menu with options: 'Gestionar Aplicación' (Registrar, Listar), 'Gestionar Funcionalidad' (Crear, Listar), 'Gestionar Rol' (Crear, Funcionalidad, Listar), 'Gestionar Usuario' (Crear, Listar), 'Gestionar Puesto' (Crear, Listar), and 'Informe Trazas' (Informe). The main content area shows a 'Rol' form with two input fields: 'Nombre De Rol' and 'Descripción De Rol'. There are 'Cancelar' and 'Registrar' buttons at the bottom right of the form. The footer contains the copyright notice: 'Copyright © 2008. Universidad de las Ciencias Informáticas. Todos los derechos reservados.' and the load time: 'cargado en 0.2497 seg'.

The bottom screenshot shows the same web application interface. The date is '25 de Marzo del 2009'. The left sidebar is the same as in the top screenshot. The main content area shows a table titled 'Rol' with the following data:

Nombre	Activo	Acción
administrador	<input checked="" type="checkbox"/>	editar eliminar
usuario avanzado	<input checked="" type="checkbox"/>	editar eliminar
usuario	<input checked="" type="checkbox"/>	editar eliminar

The footer contains the same copyright notice: 'Copyright © 2008. Universidad de las Ciencias Informáticas. Todos los derechos reservados.'.

Tabla 2.4 Historia de usuario Gestionar rol

Historia de Usuario	
Número:4	Nombre Historia de Usuario: Gestionar usuario
Usuario: Vladimir	Iteración Asignada:2
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Descripción: Permite insertar un usuario local, registrar en caso de usuarios de DNS. Se le asigna un rol en una o varias aplicaciones se podrán eliminar además.	
Observaciones: Para modificar un usuario local es necesario eliminarlo y volverlo a crear	
Prototipo de interface:	
<p>The screenshot shows a web application interface titled 'Administración'. On the left is a navigation menu with options like 'Gestionar Aplicación', 'Gestionar Funcionalidad', 'Gestionar Rol', and 'Gestionar Usuario'. The main content area displays a 'Usuario' form with the following elements: a text input for 'Nombre Usuario', a checkbox for 'Usuario Local' (which is unchecked), a dropdown menu for 'Aplicación' (set to '[---Seleccione---]'), and another dropdown menu for 'Rol' (also set to '[---Seleccione---]'). At the bottom right of the form are 'Cancelar' and 'Registrar' buttons. The top right corner of the page shows the date '25 de Marzo del 2009'.</p>	
<p>This screenshot shows the same 'Administración' web application interface. The 'Usuario' form now includes a 'Contraseña' field. The 'Usuario Local' checkbox is now checked. The date in the top right corner is '28 de Abril del 2009' with the version number '[[127.0.0.1]]' in brackets. The rest of the interface, including the navigation menu and form fields, remains the same as in the previous screenshot.</p>	

Tabla 2.5 Historia de usuario Gestionar usuario

Historia de Usuario	
Número:5	Nombre Historia de Usuario: Gestionar puesto
Usuario: Vladimir	Iteración Asignada: 1
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Descripción: Permite insertar un puesto de trabajo, eliminar, mostrar y asignarlo a una aplicación	
Prototipo de interface:	

Administración

17 de Junio del 2009 [[10.12.69.35]] Elvis:Salir

<ul style="list-style-type: none"> > Gestionar Aplicación <ul style="list-style-type: none"> Registrar Listar > Gestionar Funcionalidad <ul style="list-style-type: none"> Crear Listar > Gestionar Rol <ul style="list-style-type: none"> Crear Funcionalidad <ul style="list-style-type: none"> Listar > Gestionar Usuario <ul style="list-style-type: none"> Crear Listar > Gestionar Puesto <ul style="list-style-type: none"> Crear Listar > Informe Trazas <ul style="list-style-type: none"> Informe 	<p>Puestos de Aplicaciones</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #003366; color: white;"> <th colspan="2">ASIGNACION</th> <th style="text-align: right;">Acción</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">IP</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.12.23.234</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.23.33.34</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.9.12.23</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.12.34.56</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr style="background-color: #003366; color: white;"> <th colspan="2">NOMBREAPP1</th> <th style="text-align: right;">Acción</th> </tr> <tr> <td style="text-align: center;">IP</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.12.23.234</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.23.33.34</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr style="background-color: #003366; color: white;"> <th colspan="2">ADMINISTRACION</th> <th style="text-align: right;">Acción</th> </tr> <tr> <td style="text-align: center;">IP</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.0.0.1</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.12.69.35</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr style="background-color: #003366; color: white;"> <th colspan="2">EJEMPLO</th> <th style="text-align: right;">Acción</th> </tr> <tr> <td style="text-align: center;">IP</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> <tr> <td>10.0.0.31</td> <td></td> <td style="text-align: right;">eliminar</td> </tr> </tbody> </table>	ASIGNACION		Acción	IP		eliminar	10.12.23.234		eliminar	10.23.33.34		eliminar	10.9.12.23		eliminar	10.12.34.56		eliminar	NOMBREAPP1		Acción	IP		eliminar	10.12.23.234		eliminar	10.23.33.34		eliminar	ADMINISTRACION		Acción	IP		eliminar	10.0.0.1		eliminar	10.12.69.35		eliminar	EJEMPLO		Acción	IP		eliminar	10.0.0.31		eliminar
ASIGNACION		Acción																																																		
IP		eliminar																																																		
10.12.23.234		eliminar																																																		
10.23.33.34		eliminar																																																		
10.9.12.23		eliminar																																																		
10.12.34.56		eliminar																																																		
NOMBREAPP1		Acción																																																		
IP		eliminar																																																		
10.12.23.234		eliminar																																																		
10.23.33.34		eliminar																																																		
ADMINISTRACION		Acción																																																		
IP		eliminar																																																		
10.0.0.1		eliminar																																																		
10.12.69.35		eliminar																																																		
EJEMPLO		Acción																																																		
IP		eliminar																																																		
10.0.0.31		eliminar																																																		

Administración

12 de Junio del 2009 [[127.0.0.1]] Vladimir:Salir

<ul style="list-style-type: none"> > Gestionar Aplicación <ul style="list-style-type: none"> Registrar Listar > Gestionar Funcionalidad <ul style="list-style-type: none"> Crear Listar > Gestionar Rol <ul style="list-style-type: none"> Crear Funcionalidad <ul style="list-style-type: none"> Listar > Gestionar Usuario <ul style="list-style-type: none"> Crear > Gestionar Puesto <ul style="list-style-type: none"> Crear Listar > Informe Trazas <ul style="list-style-type: none"> Informe 	<p>Asignar Aplicación A Puesto</p> <p>Puesto</p> <input style="width: 100%;" type="text"/> <p>Aplicaciones</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr style="background-color: #003366; color: white;"> <th style="width: 15%;">Seleccione</th> <th>Nombre</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>CONTROL DE ACCESO</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>ASIGNACION</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td>NOMBREAPP1</td> </tr> </tbody> </table> <p style="text-align: right;"> <input type="button" value="Cancelar"/> <input type="button" value="Registrar"/> </p>	Seleccione	Nombre	<input type="checkbox"/>	CONTROL DE ACCESO	<input type="checkbox"/>	ASIGNACION	<input type="checkbox"/>	NOMBREAPP1
Seleccione	Nombre								
<input type="checkbox"/>	CONTROL DE ACCESO								
<input type="checkbox"/>	ASIGNACION								
<input type="checkbox"/>	NOMBREAPP1								

|Copyright © 2008. Universidad de las Ciencias Informáticas. Todos los derechos reservados. | cargado en 1.5432 seg

Tabla 2.6 Historia de usuario Gestionar puesto

Historia de Usuario	
Número: 6	Nombre Historia de Usuario: Informe de trazas
Usuario: Vladimir	Iteración Asignada:2
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Descripción: Permite visualizar todas las acciones realizadas por un usuario en una aplicación, desde un puesto, en una fecha y hora determinada	
Prototipo de interface:	

Administración

17 de Junio del 2009 [(10.12.69.182)] Elvis :

- > Gestionar Aplicación
 - Registrar
 - Listar
- > Gestionar Funcionalidad
 - Crear
 - Listar
- > Gestionar Rol
 - Crear
 - Funcionalidad
 - Listar
- > Gestionar Usuario
 - Crear
 - Listar
- > Gestionar Puesto
 - Crear
 - Listar
- > Informe Trazas
 - Informe

Traza De Aplicación

Aplicación

Administracion

Traza

Fecha	Hora	Nombre Usuario	Descripcion
2009-06-17	16:24:37	Elvis	Accedio A 'http://localhost/admin/index.php/aplicacion_c/mostrar_traza'
2009-06-17	16:24:42	Elvis	Accedio A 'http://localhost/admin/index.php/aplicacion_c/mostrar_funcionalidad'
2009-06-17	16:24:42	Elvis	Accedio A 'http://localhost/admin/index.php/aplicacion_c/mostrar_funcionalidad'
2009-06-17	16:24:45	Elvis	Accedio A 'http://localhost/admin/index.php/rol_c'
2009-06-17	16:24:46	Elvis	Accedio A 'http://localhost/admin/index.php/rol_c/funcionalidad'
2009-06-17	16:25:40	Elvis	Accedio A 'http://localhost/admin/index.php/aplicacion_c/mostrar_traza'
2009-06-17	16:25:43	Elvis	Accedio A 'http://localhost/admin/index.php/aplicacion_c/mostrar_traza'
2009-06-17	16:26:03	Elvis	Accedio A 'http://localhost/admin/index.php/aplicacion_c/mostrar_traza'

Administración

17 de Junio del 2009 [(10.12.69.182)] Elvis :

- > Gestionar Aplicación
 - Registrar
 - Listar
- > Gestionar Funcionalidad
 - Crear
 - Listar
- > Gestionar Rol
 - Crear
 - Funcionalidad
 - Listar
- > Gestionar Usuario
 - Crear
 - Listar
- > Gestionar Puesto
 - Crear
 - Listar
- > Informe Trazas
 - Informe

Traza De Aplicación

Aplicación

[---Seleccione---]

Traza

No Existen Datos A Mostrar

|Copyright © 2008. Universidad de las Ciencias Informáticas. Todos los derechos reservados. | cargado en 0.3065 seg

Tabla 2.6 Historia de usuario Informe de trazas

Historia de Usuario	
Número: 7	Nombre Historia de Usuario: Autenticar Usuario
Usuario: Vladimir	Iteración Asignada:2
Prioridad en Negocio: Alta	Riesgo en Desarrollo: Alto
Descripción: Permite que los usuarios de la aplicación se autenticuen	
Prototipo de interface:	

Tabla 2.7 Historia de usuario AutenticarUsuario

Requisitos no funcionales

Los requerimientos no funcionales de un sistema forman parte de las restricciones u obligaciones impuestas a las prestaciones del mismo. Las limitaciones impuestas al software, así como las restricciones de diseño deben estar reflejadas y estrechamente vinculadas a los elementos de la LRP. Con este fin se definen a continuación cada uno de estos:

- Apariencia o interfaz externa
 - Diseño sencillo y con alta navegabilidad, que permita un flujo viable y utilización
 - Estará diseñado para resolución de 800x600, aunque deberá verse en cualquier resolución superior a esta

- Usabilidad
 - Deberá visualizarse en los navegadores (Internet Explorer, Mozilla Firefox, Opera, Maxthon)
 - El sistema será imprescindible para la configuración y acceso a las aplicaciones implantadas, por lo que debe ser eficiente, preciso, con tiempos de respuestas cortos y alta velocidad de procesamiento de la información
- Seguridad
 - El sistema debe garantizar la confidencialidad e integridad del flujo informativo
 - La información manejada por el sistema deberá estar protegida de acceso no autorizado
 - Garantizar que la información sea vista únicamente por quien tiene derecho a verla
 - Protección contra acciones no autorizadas o que puedan afectar la integridad de los datos
 - Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario
- Hardware
 - Estación servidor Core Duo a 2GHZ, RAM DDR-2 bus 800 2Gb, HDD Scosi de 250 Gb
 - Estación cliente Dual Pentium a 2.4GHz, RAM DDR2 bus 667 512 MB, HDD SATA 80Gb

Restricciones en el diseño y la implementación

- Diseño con lenguaje UML, representado a través de la herramienta Case Visual Paradigm
- Implementación sobre plataforma libre PHP, framework CodeIgniter, patrón arquitectónico Modelo-Vista-Controlador, arquitectura en capas y como gestor de base de datos PostgreSQL

- Utilizar los estándares de diseño y codificación establecidos en los lineamientos de arquitectura³

2.4 Caracterización de la arquitectura. Modelo del diseño

“La arquitectura de software, tiene que ver con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.” (19)

Partiendo de lo anterior pudiera expresarse que 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. Es la encargada de definir una vista estructural de alto nivel, define estilos o combinación de estilos para construir la solución, se concentra en requerimientos no funcionales y marca el éxito o fracaso de un proyecto, de ahí la importancia que tiene determinar una arquitectura robusta, pero a su vez flexible sobre todo para el cambio.

La arquitectura de software, también denominada arquitectura lógica, consiste en un conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software.

Una arquitectura de software se selecciona y diseña enfocada en objetivos y restricciones. Los objetivos son aquellos prefijados para el sistema informático, pero no solamente los de tipo funcional, también otros objetivos como la mantenibilidad, auditabilidad, flexibilidad e interacción con otros sistemas de este tipo y las restricciones son aquellas limitaciones derivadas de las tecnologías disponibles para implementar el sistema.

La concepción de esta aplicación introduce algunos cambios en el modelo arquitectónico del Sistema de Planificación y Control de la Dirección General de Alimentos de la UCI teniendo en cuenta la problemática expresada en el análisis crítico de la ejecución de los procesos y la propuesta de integración (*ver en la figura 2.2*).

³ Idem 11



Figura 2.2 Arquitectura de integración

Existen además, numerosos estilos arquitectónicos (Anexo 7) (20) introducidos equitativamente desde el punto de vista que se prefiera observar dicha arquitectura. Haciendo referencia a la misma, en este caso particular se tiene en cuenta algunos aspectos como la tecnología empleada cliente-servidor y principalmente el uso de el framework CodeIgniter el cual está basado en uno de los estilos arquitectónicos de llamada y retorno, MVC.

A su vez la implementación de este patrón de arquitectura mediante el uso del framework, engloba patrones muy importantes y que están mayormente referidos al diseño dentro de los cuales están el patrón de diseño *Observer* para la vista y el modelo, *Composite* para anidar las vistas, *Strategy* para la vista y el controlador dentro de este *Factory Method*, para el ultimo por defecto y *Adapter* para adaptar el modelo a la vista y al controlador. Existen además otros patrones, que serán descritos posteriormente en la descripción de uno de los diagramas del diseño.

Con el objetivo de describir cómo el sistema será realizado a partir de las HU previstas e indicando con precisión lo que se debe programar se procede a la elaboración de los modelos de clases de diseño por funcionalidad, este modelo quedara expresado en la plantilla del modelo de diseño.

La plantilla del modelo de diseño, es el documento que se genera del Diseño con las metáforas, donde se debe diseñar la solución más simple que pueda funcionar y ser implementada en un momento determinado del proyecto. En XP no se enfatiza la definición temprana de una arquitectura estable

para el sistema. Dicha arquitectura se asume de forma evolutiva y los posibles inconvenientes que se generarían por no contar con ella explícitamente en el comienzo del proyecto se solventan con la existencia de una metáfora. Teniendo en cuenta las características anteriores, se define en esta plantilla, un esbozo inicial del diseño del sistema, sin entrar en especificaciones, ni detalles, solo lo que el diseñador necesita para hacer un primer entregable del sistema.

Estos modelos revelan el comportamiento, estructura y responsabilidades de las clases del diseño a partir de patrones que aportan a la solución una arquitectura bien definida y robusta (Anexo 8). Para un mejor entendimiento se describen todas las clases (Anexo 9) y en específico se detalla el diagrama de clases del diseño Gestionar usuario (*ver figura 2.6*).

La descripción de este diagrama se refleja los detalles del flujo entre clases a través de las relaciones, el uso del patrón MVC y los patrones de diseño aplicados. Las clases de la vista están integradas por las clases clientes `cp_in_usuario` y `cp_mostrar_usuario`, la primera asociada por una relación de composición con el formulario `frm_in_usuario` que permite insertar un usuario que puede ser del dominio UCI o local y la segunda visualiza los usuarios insertados y permite a través de un vínculo eliminarlos.

Se cuenta además con dos objetos de tipo `button` (insertar y cancelar). En la función `action` del botón insertar antes que los datos sean enviados, son validados en la clase `javascript` que se encuentra asociada por relaciones de agregación al formulario `frm_in_usuario`. Esto significa que este código está embebido en el código HTML.

Antes de continuar se debe tener en cuenta que estas vistas según lo referido acerca del patrón MVC⁴ son construidas por otras clases que se encuentran del lado del servidor, conocidas como controladoras. En estas clases se pone de manifiesto el patrón de diseño `Factory Method` clasificado en la categoría de patrones creacionales y consiste en utilizar una clase constructora abstracta con unos cuantos métodos definidos y otros abstractos. Este está dedicado a la construcción de objetos de un subtipo de un tipo determinado y funciona como una simplificación del `Abstract Factory`, en la que la clase abstracta tiene métodos concretos que usan algunos de los abstractos; según usemos una u otra hija de esta clase abstracta, tendremos uno u otro comportamiento.

Como se puede observar en el diagrama, la clase controladora `Sp_usuario_c` está diseñada a partir de otros patrones de asignación de responsabilidades (GRASP) como por ejemplo, el patrón `Experto` y

⁴ Idem 15

Controlador los que le atribuyen la virtud de poder reconocer las clases que contienen la información a guardar o solicitada y quien debería de atender un evento en el sistema definiendo además el método de para su ejecución. También por las facilidades que brinda el framework CodeIgniter se crea esta clase que hereda del controlador principal del framework a través de relaciones de dependencia, aplicando patrones de herencia y polimorfismo.

Cuando se envían los datos desde el formulario a través de relaciones de asociación (submit), estos son recepcionados por el controlador frontal (Index) y son enviados por este a la clase de acceso a datos o mejor conocida como clase modelo Esta clase modelo llamada usuario_modelo de igual forma que la clase controladora está asociada por relaciones de dependencia con la clase modelo del framework de la cual hereda su funcionamiento. Como parte de esta clase se encuentra asociado a través de una relación de composición el servicio web AdministracionWS el cual contiene los métodos para esta funcionalidad.

En la clase modelo se evidencian además algunos patrones GOF, específicamente el patrón Proxy encargado de que la clase de acceso a datos pueda ocultar los objetos reales, como lo es para las clases Rol y Usuario, evitando el contacto directo de la vista con dichas clases. Para esto mediante este patrón se crean dentro de ella objetos virtuales que son parte de los métodos declarados en la clase y que pudieran ser accesibles por el usuario. Las clases Usuario y Rol están vinculadas al servicio mediante relaciones de agregación las cuales permiten a este acceder a ellas.

Asociados a la capa de acceso a datos, se utilizan los métodos ObtenerusuariodadoCadenabusqueda y ObtenerUsuario (pertenecientes al componente externo representado por el servicio de autenticación) para insertar un usuario del dominio. A través de este servicio se logra insertar en la aplicación los usuarios que forman parte del dominio UCI, guardados en la clase Usuario, la cual posee dos clases (Usuariodominio y UsuarioLocal) que se asocian por relaciones de herencia con la misma.

Por último, el flujo de respuesta a la petición del usuario se aprecia a través de la relación de asociación (redirect) mediante la cual el controlador frontal (index) envía el resultado a la controladora (Usuario_c) para que construya la vista mostrar.

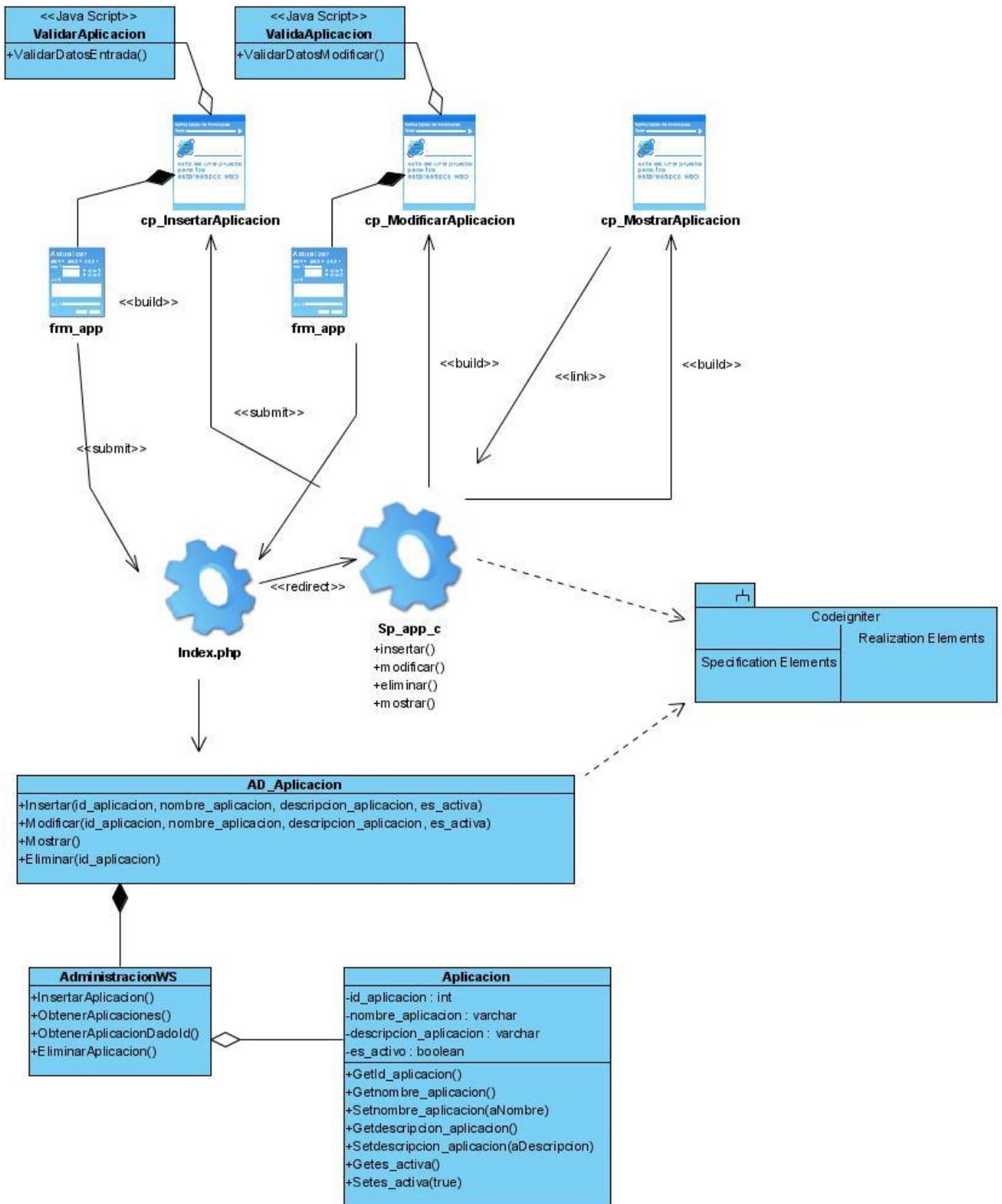


Figura 2.3 Diagrama de clases de diseño para la HU Gestionar aplicación

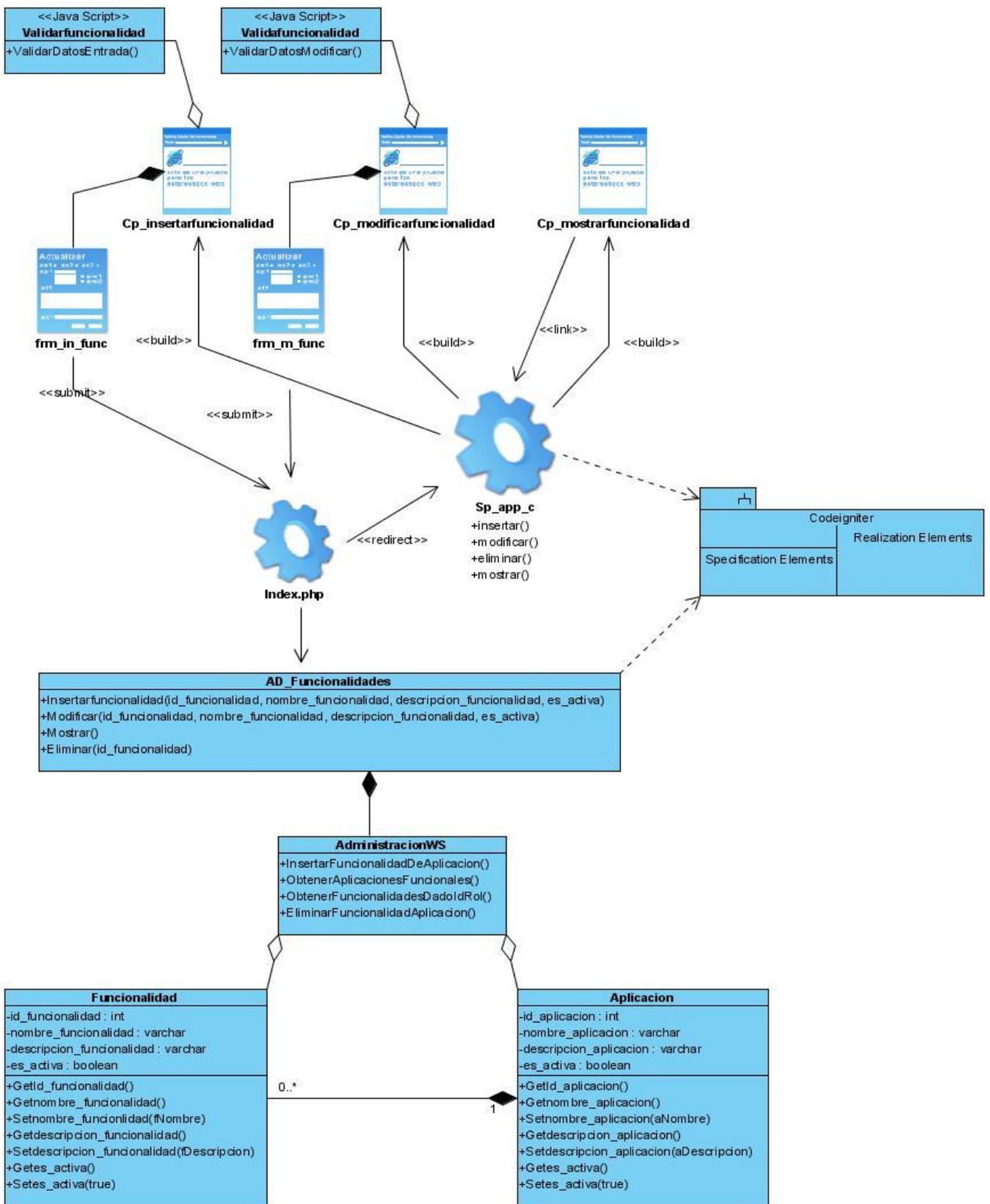


Figura 2.4 Clases de diseño para la HU Gestionar funcionalidad

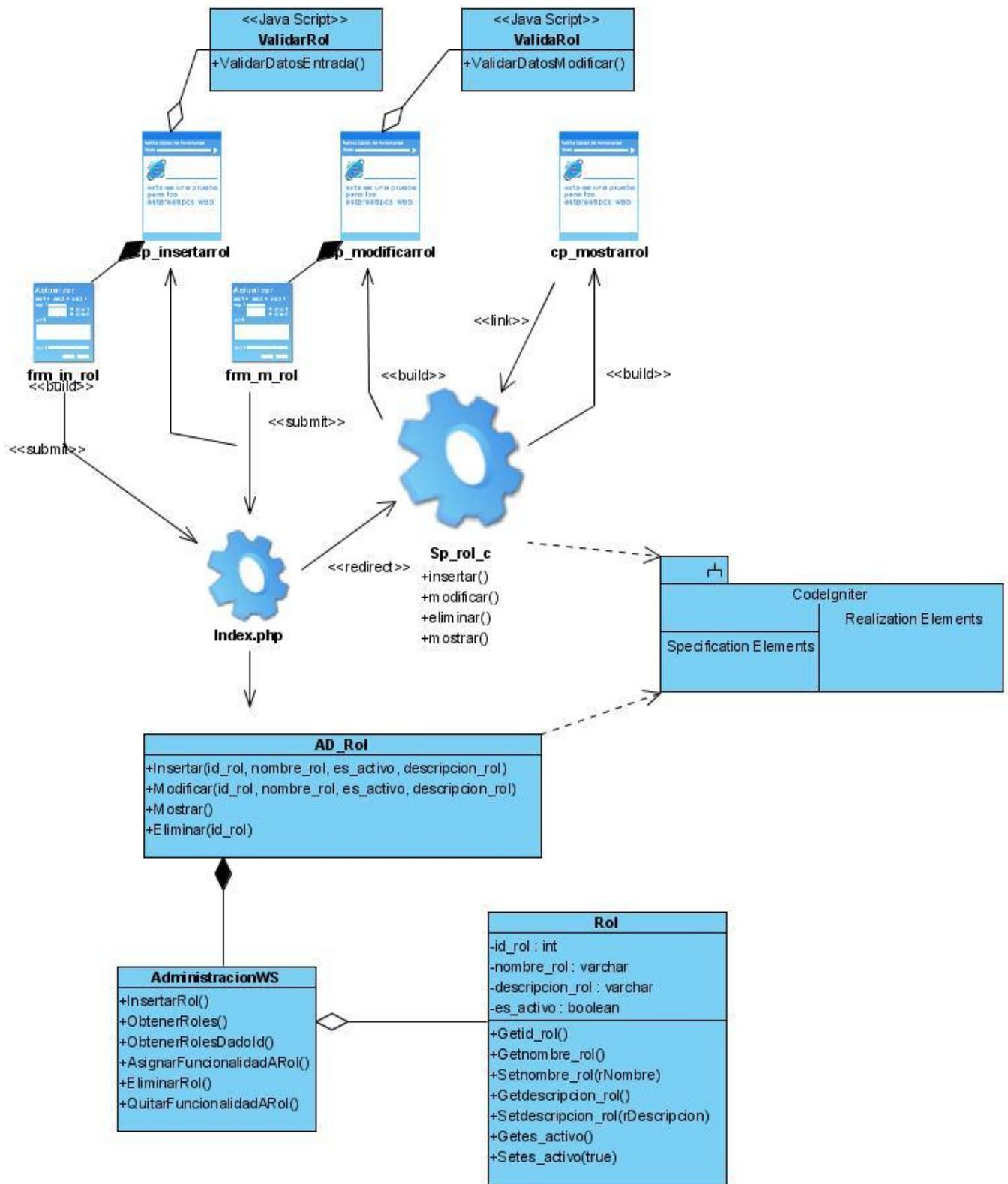


Figura 2.5 Clases del diseño para la HU Gestionar rol

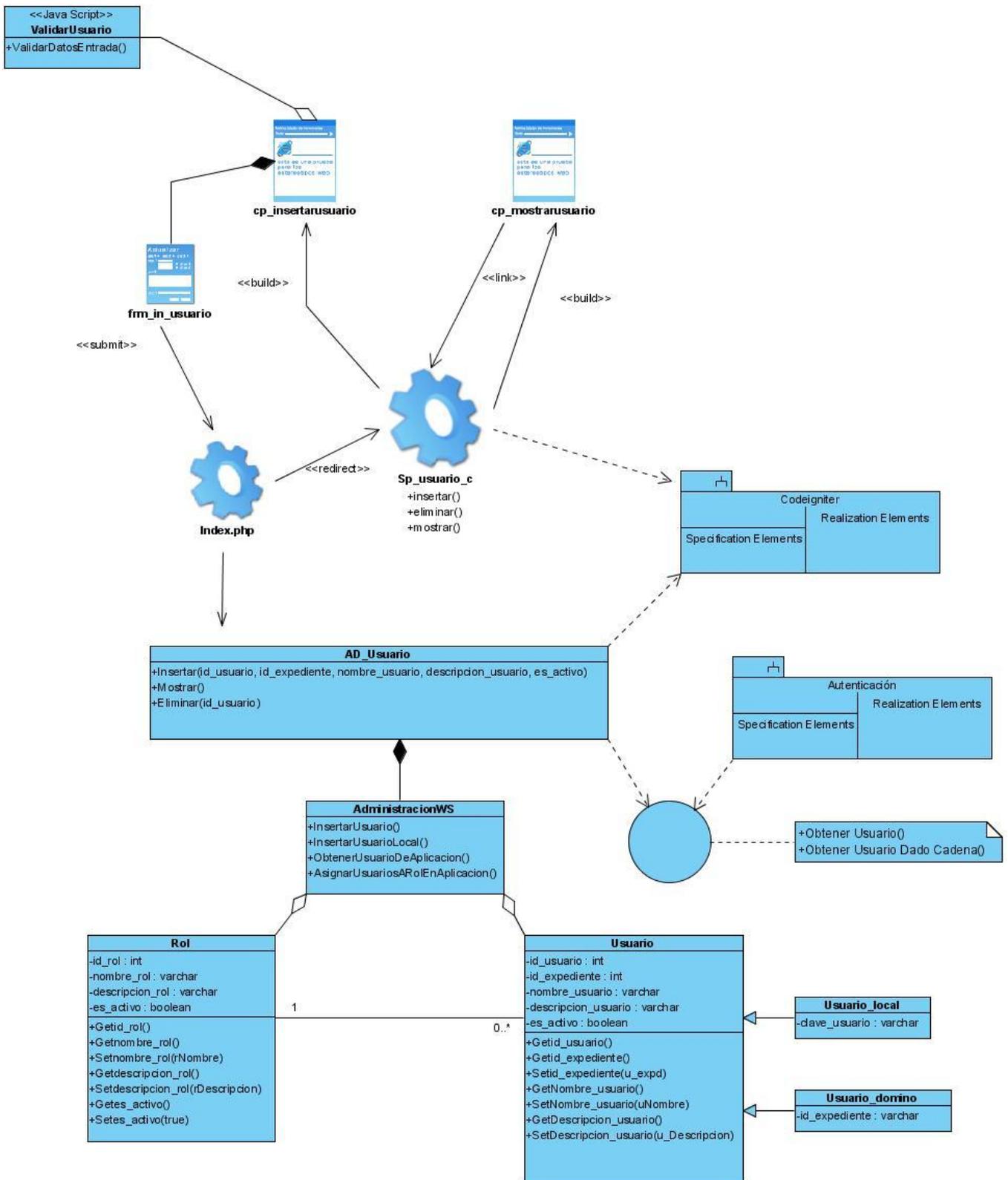


Figura 2.6 Clases de diseño para la HU Gestionar usuario

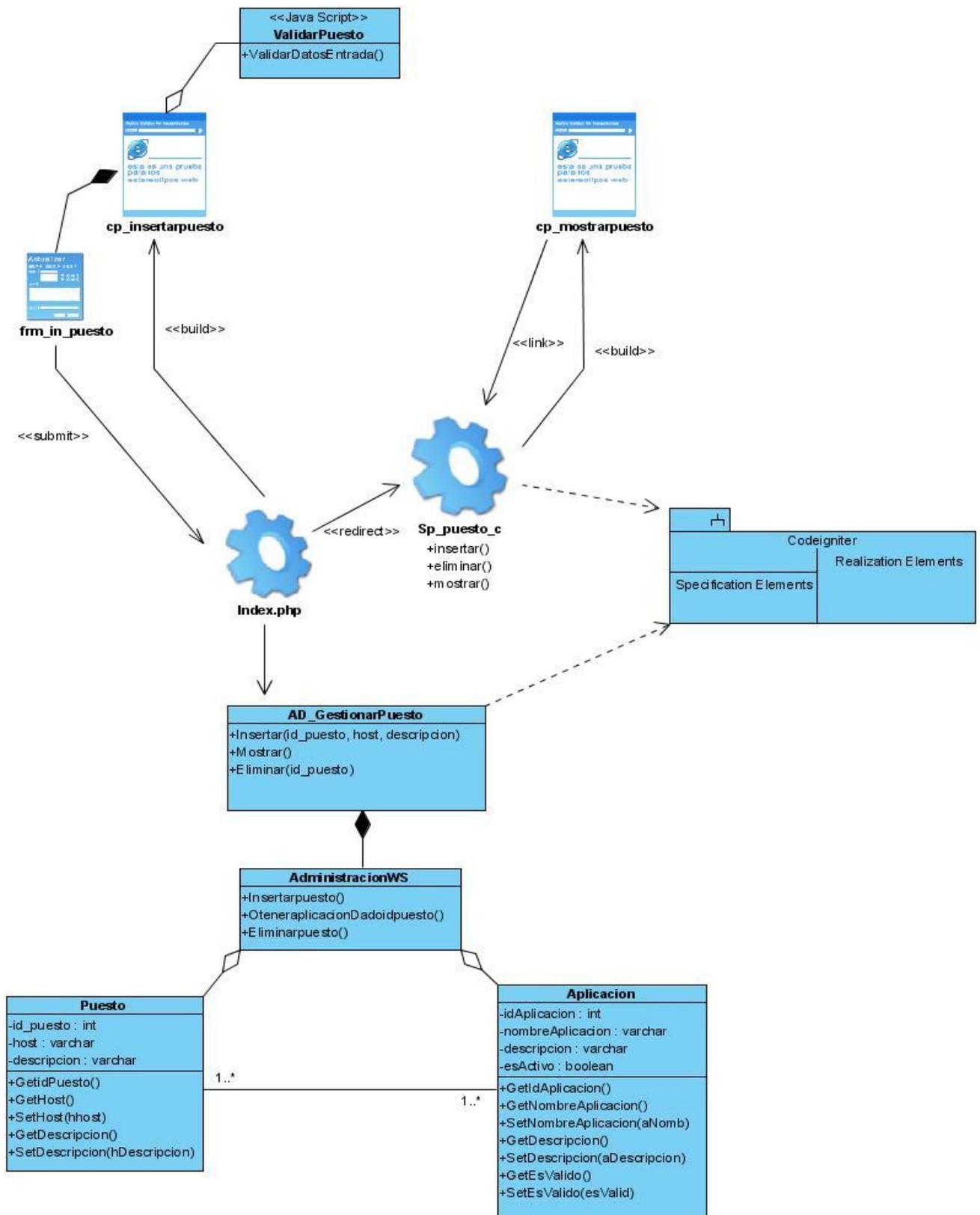


Figura 2.7 Clases de diseño para la HU Gestionar puesto

2.6.1 Diseño de la base de datos

Las bases de datos son estructuras que permiten almacenar los datos, reconocer el contenido y recuperar la información en las aplicaciones. Está se define a partir de la necesidad de guardar los datos que persisten y constituyen a su vez la primera abstracción de la vista de datos.

La persistencia es la propiedad que permite a las clases conservar su valor en el espacio y el tiempo estás requieren de conservar información en un medio físico permanente de almacenamiento, además de mantener su estado. A continuación se muestra el diagrama de clases persistentes (ver figura 2.7) del sistema que se propone:

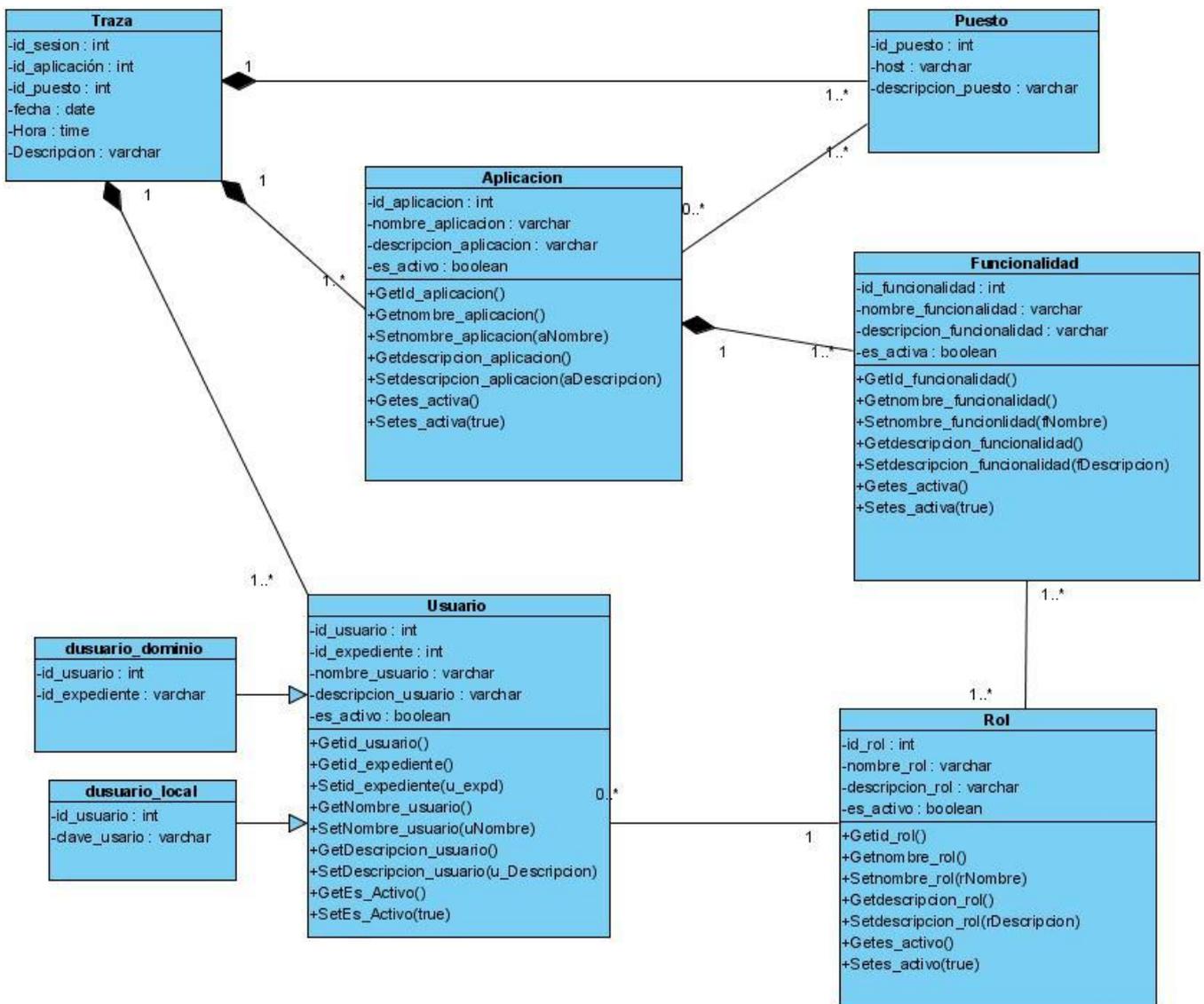


Figura 2.7 Modelo de persistencia

El modelo físico datos mostrado a continuación (ver figura 2.8) es un componente que describe la representación lógica y física de datos persistentes en el sistema.

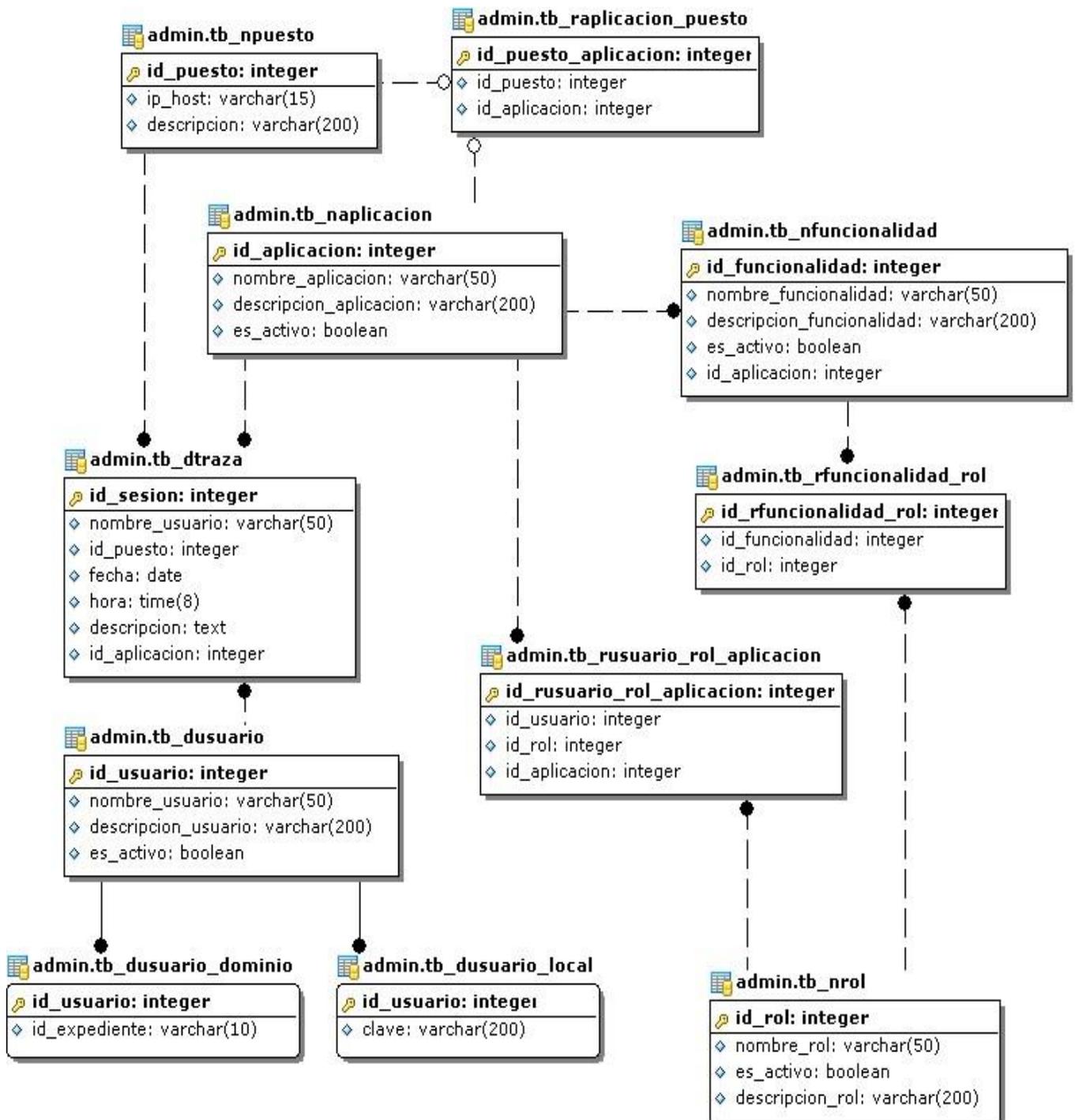


Figura 2.8 Esquema del modelo físico de datos

Conclusiones del capítulo

Basados en el modelo de dominio representado en este capítulo, se logro mostrar el entorno en que se solicita el sistema, proporcionando una forma rápida y concreta de captar la necesidad del mismo. A partir del dominio del flujo actual de los procesos y su análisis crítico se obtuvo la LRP definida en función de las necesidades del cliente y que a su vez, respaldaron la base para el diseño del software.

Se llevó a cabo la construcción del modelo de diseño para cada HU basado en la utilización de los patrones de diseño y estilos arquitectónicos, así como los demás elementos necesarios para la implementación. Se definieron, a partir del mismo, el modelo lógico a través del diagrama de las clases persistentes y la estructura que define el modelo físico de datos.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

La implementación de la aplicación está contenida en la etapa de desarrollo de la metodología y como tarea fundamental pretende expresar los artefactos referidos a la plantilla de glosario de términos, las tareas de ingeniería, plan de release, estándar de código de fuente, así como visualizar los diagramas de componentes y despliegue. También se realiza la planificación de pruebas y las plantillas de pruebas de aceptación, artefactos generados para la validación de cada funcionalidad identificada en las HU.

3.1 Tareas de ingeniería

La plantilla de tarea de ingeniería (Anexo 10), posee una gran importancia, pues permite definir cada una de las actividades que estarán asociadas a las HU y que permitirán su implementación. También posibilita conocer que programador está asignado a cada tarea, así como el tiempo que se necesita para su realización, lo que facilita la estimación del tiempo consumido en cada HU a implementar, de acuerdo a su complejidad. Permite organizar el proceso de implementación, pues las tareas se van implementando de acuerdo a su prioridad y viabiliza conocer el grado de complejidad de cada HU, teniendo en cuenta la cantidad de tareas asociadas⁵.

3.2 Plan de release

La plantilla del plan de release, es un documento que define cuales son las HU más significativas, divide el proceso de desarrollo de software en iteraciones, asocia las HU a las iteraciones según la prioridad. Además, se planifica el trabajo a realizar en cada una de ellas. Las HU de mayor riesgo y mayor prioridad se incluyen en las primeras iteraciones. El lanzamiento y las iteraciones tienen fechas fijas para su consecución, así como un alcance variable⁶.

Release	Orden de la HU a implementar	Duración total (en semanas)
1	1,2,5,3	4
2	4,6,7	4

⁵ Idem 18

⁶ Idem 18

Diagrama de componentes

Un diagrama de componente describe los elementos físicos de un sistema y sus relaciones de interacción. En él se muestran los elementos del software que sustentaron la manufactura de la aplicación de administración para el Sistema de Planificación y Control de la Dirección General de Alimentos. Por la estructura de organización que proporciona el patrón de arquitectura utilizado MVC a través del framework CodeIgniter, se separan los componentes en los paquetes Modelo, Vista y Controlador, con el fin de lograr una mejor comprensión. Además se representa la base de datos Alimnetacion_UCI, el servicio web AdministracionWS y las librerías utilizadas, considerándose éstos últimos componentes sin estados.

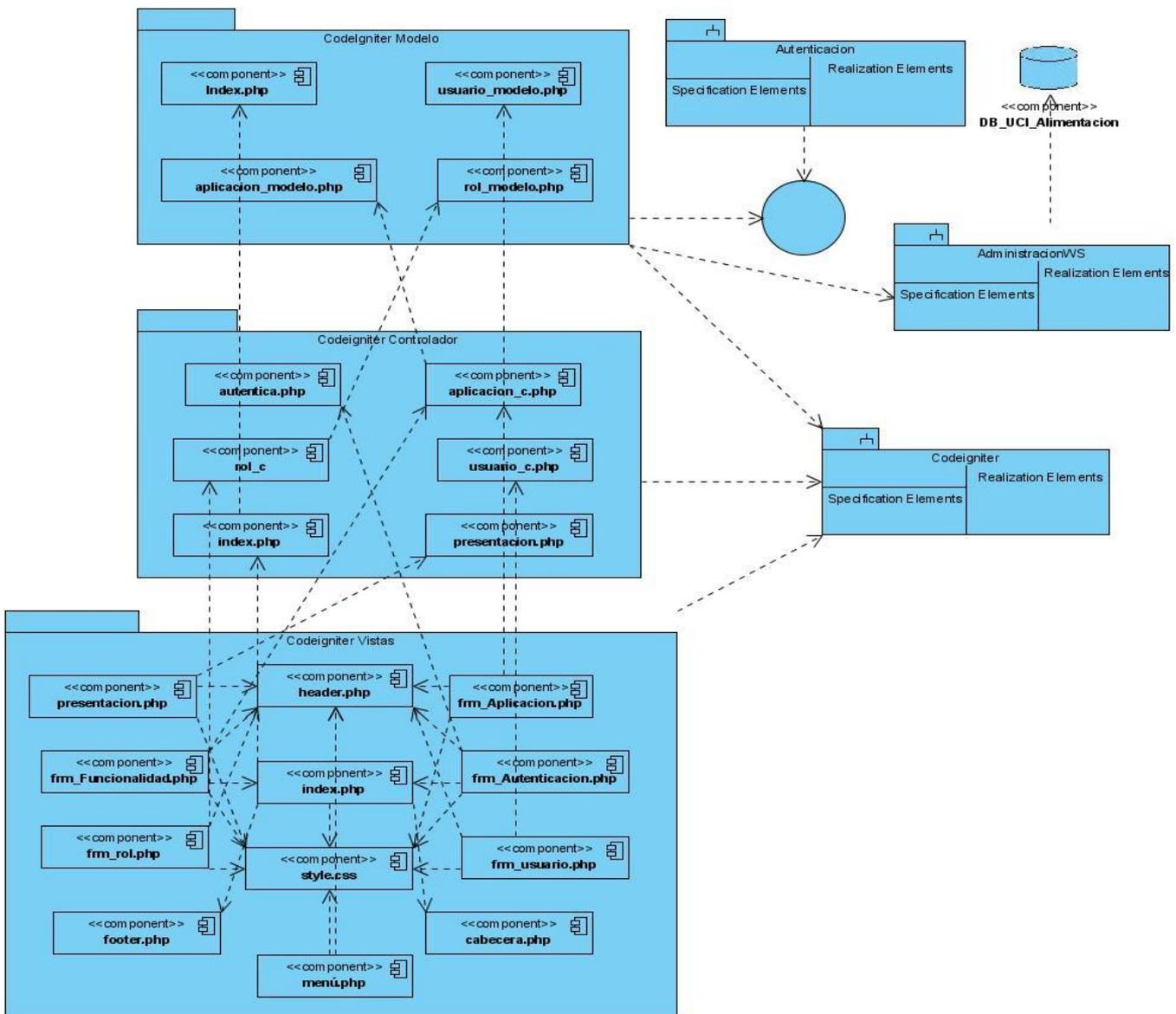


Figura 3.2 Diagrama de componentes

3.3 Estándar de programación

El documento Estándar de programación, se genera de la etapa de implementación. En él se recoge el estándar utilizado y su explicación. Facilita una posible reutilización del código, conociendo el estándar utilizado para su programación y posibilita una mejor descripción de los elementos fundamentales del proceso implementación⁷.

Despliegue

El diagrama de despliegue es el artefacto generado en UML para mostrar las relaciones físicas entre los componentes hardware y software de un sistema informático. Este diagrama está compuesto por los diferentes nodos que pueden ser representaciones de estaciones clientes, servidores de aplicaciones, servidores de bases de datos, servicios web y los protocolos de comunicación entre ellos que pueden ser drivers de conexión ejemplo PDO, ADO, etc. Entre los servidores y clientes se encuentran componentes de protocolo TCP/IP como son HTTP, HTTPS, FTP, entre otros. En el siguiente diagrama (Figura 3.1) se modela la distribución de las diferentes partes que conforman el Sistema de Gestión de Accesos.

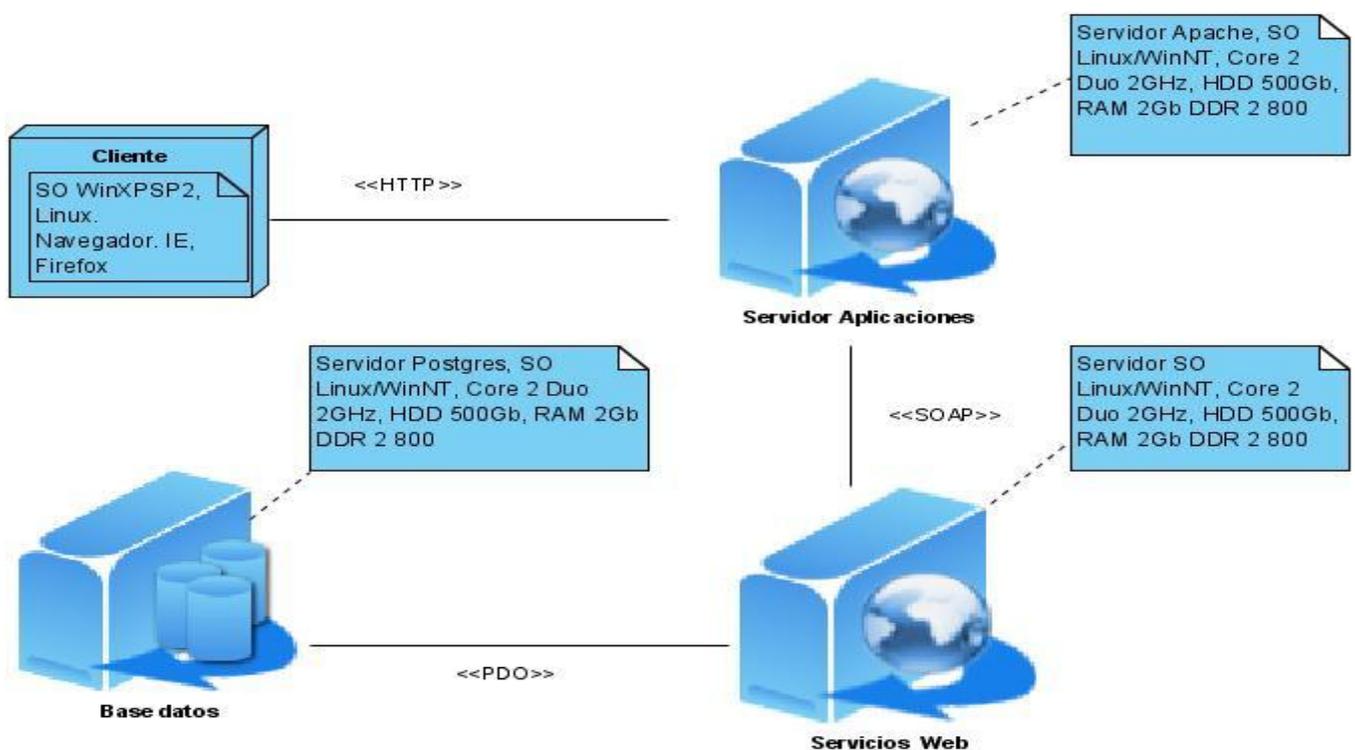


Figura 3.1 Diagrama de despliegue

⁷ Idem 18

Finalmente después de haber implementado todas las tareas de ingeniería y desarrollado el código fuente de la aplicación, se procede a la validación de los resultados a través de la ejecución de pruebas que certifiquen la correcta implementación de todas las necesidades del cliente.

3.4 Pruebas de aceptación

La plantilla de Caso de prueba de aceptación, se genera de la etapa de pruebas. El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad. Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al cliente. La utilización de estas constituye el termómetro de los desarrolladores, fundamentalmente de los programadores a la hora de medir la calidad de su trabajo y garantizan la entrega de un producto con calidad, que responde a las necesidades del cliente⁸. Para constancia de esto se muestran las siguientes pruebas:

Casos de Prueba de Aceptación

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-1-1	Nombre Historia de Usuario: Gestionar Aplicación
Nombre de la persona que realiza la prueba: Vladimir Campos	
Descripción de la Prueba: Al ejecutar el vinculo Gestionar aplicación del menú principal se muestra la interfaz de usuario (formulario frmAplicacion.php). (Ver tabla 2.2 HU Gestionar Aplicación). Para la funcionalidad (insertar) se comprueba si los datos de cada objeto del formulario son validados e insertados.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Conexión con el servidor • Ser usuario con privilegios de administrador • El usuario debe estar autenticado en la aplicación 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. El usuario oprime el botón cancelar sin escribir datos 	

⁸ Idem 18

2. El usuario oprime el botón cancelar con datos escritos
3. El usuario escribe datos incorrectos y oprime el botón insertar
4. El usuario omite campos requeridos y oprime el botón insertar
5. El usuario inserta una aplicación con el mismo nombre
6. El usuario inserta una aplicación nueva

Resultado Esperado:

1. Retorna al menú principal
2. Reinicia el formulario
3. Se muestra mensaje (Datos incorrectos) y ubica el cursor en el campo erróneo
4. Se muestra mensaje (Datos inválidos)
5. Se muestra un mensaje (Ya existe una aplicación con ese nombre)
6. Se visualizan en la interfaz (mostrar) el listado de aplicaciones

Evaluación de la Prueba: Insatisfactoria

Observación: La prueba no cumplió con el requisito de salida 5

Caso de Prueba de Aceptación

Código Caso de Prueba: A-1-2

Nombre Historia de Usuario: Gestionar Aplicación

Nombre de la persona que realiza la prueba: Vladimir Campos

Descripción de la Prueba: Al ejecutar el vínculo Gestionar aplicación del menú principal se muestra la interfaz de usuario (formulario frmAplicacion.php). (Ver tabla 2.2 HU Gestionar Aplicación). Para la funcionalidad (insertar) se comprueba si los datos de cada objeto del formulario son validados e insertados.

Condiciones de Ejecución:

- Conexión con el servidor
- Ser usuario con privilegios de administrador
- El usuario debe estar autenticado en la aplicación

<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El usuario oprime el botón cancelar sin escribir datos 2. El usuario oprime el botón cancelar con datos escritos 3. El usuario escribe datos incorrectos y oprime el botón insertar 4. El usuario omite campos requeridos y oprime el botón insertar 5. El usuario inserta una aplicación con el mismo nombre 6. El usuario inserta una aplicación nueva
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. Retorna al menú principal 2. Reinicia el formulario 3. Se muestra mensaje (Datos incorrectos) y ubica el cursor en el campo erróneo 4. Se muestra mensaje (Datos inválidos) 5. Se muestra un mensaje (Ya existe una aplicación con ese nombre) 6. Se visualizan en la interfaz (mostrar) el listado de aplicaciones
<p>Evaluación de la Prueba: Satisfactoria</p>

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-3-1	Nombre Historia de Usuario: Gestionar Rol
Nombre de la persona que realiza la prueba: Vladimir Campos	
Descripción de la Prueba: Al ejecutar el vinculo Gestionar rol del menú principal se muestra la interfaz de usuario (formulario frmrol.php). (Ver tabla 2.4 HU Gestionar Rol). Para la funcionalidad (modificar) se comprueba si se actualizaron los datos.	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> • Conexión con el servidor • Ser usuario con privilegios de administrador • El usuario debe estar autenticado en la aplicación 	

<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El usuario selecciona un rol en la vista (mostrar) 2. El usuario inserta nuevos los nuevos datos
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. Se muestra el formulario con los datos de rol (frmRol) 2. Se observa en la vista (mostrar) los datos actualizados
<p>Evaluación de la Prueba: Satisfactoria</p>

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-5-1	Nombre Historia de Usuario: Gestionar puesto
Nombre de la persona que realiza la prueba: Vladimir Campos	
Descripción de la Prueba: Si el puesto está asignado a la aplicación se muestra la interfaz de autenticación, donde el usuario intenta autenticarse.	
<p>Condiciones de Ejecución:</p> <ul style="list-style-type: none"> • Conexión con el servidor • El puesto debe estar asignado a la aplicación • El usuario debe estar asignado a un rol en la aplicación • El usuario debe autenticarse en una aplicación 	
<p>Entrada / Pasos de ejecución:</p> <ol style="list-style-type: none"> 1. El usuario intenta acceder a la aplicación 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. Se muestra la interfaz de autenticación 	
<p>Evaluación de la Prueba: Satisfactoria</p>	

Caso de Prueba de Aceptación

Código Caso de Prueba: A-7-1	Nombre Historia de Usuario: Autenticar Usuario
Nombre de la persona que realiza la prueba: Vladimir Campos	
Descripción de la Prueba: Si el usuario está asignado a la aplicación accede a ella.	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Conexión con el servidor • El puesto debe estar asignado a la aplicación • El usuario debe estar asignado a un rol en la aplicación • El usuario debe autenticarse en una aplicación 	
Entrada / Pasos de ejecución: <ol style="list-style-type: none"> 1. El usuario se autentica en la aplicación 	
Resultado Esperado: <ol style="list-style-type: none"> 1. Se muestra la interfaz de usuario de la aplicación 	
Evaluación de la Prueba: Satisfactoria	

Caso de Prueba de Aceptación	
Código Caso de Prueba: A-6-1	Nombre Historia de Usuario: Informe de trazas
Nombre de la persona que realiza la prueba: Vladimir Campos	
Descripción de la Prueba: El usuario realizara una serie de acciones en la aplicación y se comprobara si estas son registradas en el informe de trazas	
Condiciones de Ejecución: <ul style="list-style-type: none"> • Conexión con el servidor • El puesto debe estar asignado a la aplicación • El usuario debe estar asignado a un rol en la aplicación • El usuario debe autenticarse en una aplicación 	

Entrada / Pasos de ejecución: 1. El usuario realiza ejecuta varias funcionalidades en la aplicación
Resultado Esperado: 1. Las funcionalidades ejecutadas por el usuario se muestran en la interfaz de usuario Trazas
Evaluación de la Prueba: Satisfactoria

Finalmente a través de estas pruebas han quedado probadas algunas funcionalidades básicas de la aplicación y a su vez el funcionamiento del servicio AdministraciónWS, no obstante fueron señaladas algunas inconformidades respecto al diseño gráfico de las interfaces de usuario, lo cual quedará reflejado en las recomendaciones de la investigación, para darle cumplimiento en las próximas iteraciones que se le realicen a la misma.

Conclusiones del capítulo

En este capítulo se dio cumplimiento a la realización de los artefactos referidos a la plantilla de glosario de términos, las tareas de ingeniería, plan de release, estándar de código de fuente, así como visualizar los diagramas de componentes y despliegue. También se validó a partir de los requisitos identificados el cumplimiento con las funcionalidades expuestas por el cliente en la LRP y se señalaron algunas inconformidades con respecto al requisito no funcional que responde al diseño gráfico de las interfaces de usuario.

CONCLUSIONES

- A partir de las necesidades del cliente reflejadas en la LRP, se identificaron 7 historias de usuario que incluyen los requerimientos necesarios para la implementación de la aplicación
- Se elaboró el modelo de diseño donde se obtuvo una vista arquitectónica del flujo entre clases y facilitó la comprensión de las relaciones entre las mismas
- Se obtuvieron 11 tablas en la base de datos a partir del modelo de persistencia de clases
- La implementación del código fuente permitió obtener una solución para gestionar la configuración del Sistema de Planificación y Control de la Dirección General de Alimento de la UCI
- Las pruebas realizadas certifican que la aplicación obtenida cumple con las necesidades expresadas por el cliente

RECOMENDACIONES

- Las aplicaciones del Sistema de Planificación y Control de la Dirección General de Alimento de la UCI deben acoplarse a este módulo, de manera que se logre la gestión administrativa
- El presente trabajo debe seguir desarrollándose con el objetivo de enriquecer el diseño gráfico de las interfaces de usuario
- Los artefactos generados en el trabajo deben incluirse en el expediente del proyecto UCIAlimentación
- Se debe evaluar la generalización de este tipo de aplicación a otros sistemas con características similares dentro o fuera de la UCI

TRABAJOS CITADOS

1. **CARRERA, R.** *Apuntes de la materia de sistemas integrales de información.* 2002.
2. **CIV.** Centro de la industria virtual. *Sitio del CIV de la UTEM.* [En línea] Noviembre de 2001. [Citado el: 16 de Febrero de 2009.] http://www.civ.cl/academico/aedil/asignaturas/Sistemas_Administracion/sia.doc.
3. **J, SENN.** *Análisis y diseño de Sistemas de Información.* Mexico : s.n., 1995.
4. **CHIAVENATO, I.** *Introducción a la Teoría General de la Administración.* 7ma Edición. s.l. : McGraw Hill, 2005. ISBN: 9701055004.
5. **KOONTZ, H. y WEINRICHY, H.** *Elementos de la Administración.* 7ma Edición. México : McGraw Hill, 2007. ISBN: 970106058X. ISBN-13: 9789701060582.
6. **Varni, Dario Angel Gonzalez y Marcos.** *BackOffice Small Business Server 4.5.* s.l. : Mp Ediciones Corp, Oct, 2000. Paperback & CD-ROM. 9789875260429.
7. **Antares.** www.antares.com. [En línea] [Citado el: 29 de Enero de 2009.] <http://www.antares.com>.
8. **Kappel, Gerti.** *Web Engineering: The Discipline of Systematic Development of Web Applications*, chapter An Introduction to Web Engineering. [aut. libro] Gerti Kappel et al. s.l. : John Wiley & Sons Ltd, 2006, págs. 1–22.
9. **Baxley, Bob.** <http://www.boxesandarrows.com>. [En línea] 2003. [Citado el: 20 de Enero de 2009.] http://www.boxesandarrows.com/view/what_is_a_web_application.
10. **Mora, Sergio Luján.** *Programación de aplicaciones Web: historia, principios básicos y clientes Web.* s.l. : Editorial Club Universitario, 2002, págs. 25–37.
11. **Dirección de Informatización UCI.** <http://www.uddi.uci.cu>. *Sitio web de la arquitectura para los sistemas que conforman la intranet de la UCI.* [En línea] Mayo de 2007. [Citado el: 20 de Marzo de 2009.] <http://uddi.uci.cu/files/arquitectura.2007.5.9.pdf>.
12. **Dondo, Agustín.** <http://www.programacion.com>. [En línea] [Citado el: 12 de Febrero de 2009.] <http://www.programacion.com/php/articulo/porquephp>.
13. **Codeigniter.** Codeigniter corp. [En línea] [Citado el: 2 de Febrero de 2009.] Manual disponible en Inglés, traducido al español. <http://www.codeigniter.com>.
14. **Equipo de Desarrollo de PostgreSQL.** PostgreSQL corp. [En línea] Septiembre de 2007. [Citado el: 15 de Mayo de 2009.] <http://es.tldp.org/Postgresql-es/web/navegable/tutorial/tutorial.html>. 1996-9.
15. **Burbeck, Steve.** <http://st-www.cs.uiuc.edu>. *Application programming in Smalltalk-80: How to use Model-View-Controller (MVC).* [En línea] [Citado el: 2 de Febrero de 2009.] University of Illinois in Urbana-Champaign, Smalltalk Archive. <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>.
16. **Sánchez, M. A. M.** <http://www.informatizate.net>. [En línea] 2002. [Citado el: 2 de Febrero de 2009.] http://www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_07062004.pdf.
17. **José H. Canós, Patricio Letelier, y Msc Carmen Penadés.** <http://www.willydev.net>. [En línea] 2005. [Citado el: 4 de Febrero de 2009.] <http://www.willydev.net/descargas/prev/ToDoAgil.Pdf>.
18. **Peñalver Romero, Gladys Marsi.** *MA-GMPR-UR2 Metodología ágil para proyectos de software libre.* s.l. : Facultad 10, 2008.

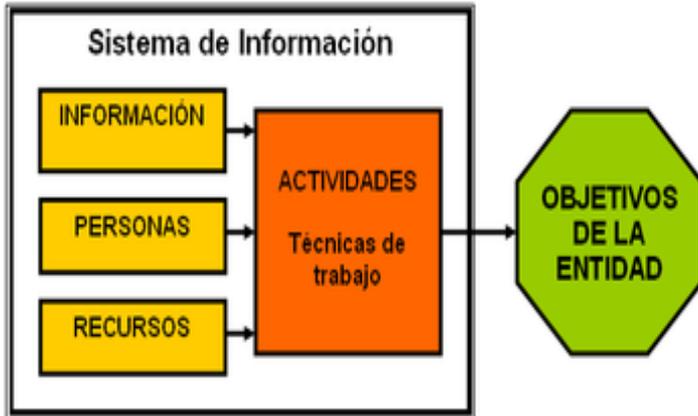
19. **Kruchten, Philippe.** "Architectural Blueprints--The 4+1 View Model of Software Architecture". 1995, págs. 42-50.
20. **Reynoso, Billy.** MSDN. Microsoft.corp, Billy Reynoso, Billyr@microsoft.com.ar. *Microsoft.corp.* [En línea] 2006. [Citado el: 3 de Abril de 2009.] <http://www.microsoft.com/spanish/msdn/arquitectura>.
21. **soeduc.** <http://www.soeduc.cl>. [En línea] [Citado el: 17 de Febreo de 2009.] <http://www.soeduc.cl/apuntes/basededatos.doc>.
22. **Universidad de Ciencias Informáticas.** *Sitio Web de la Direccion de Televisión Universitaria para la transmisión de tele clases.* [En línea] DTU-UCITeVe, 2004. [Citado el: 16 de Mayo de 2009.] http://inter-nos.uci.cu/Teleclases/Teleclases.asp?id_as=12.
23. **Grupo de PHP.** *The PHP Group.* [En línea] PHP, 2001. [Citado el: 25 de 05 de 2009.] <http://www.php.net/md5>.

BIBLIOGRAFIA

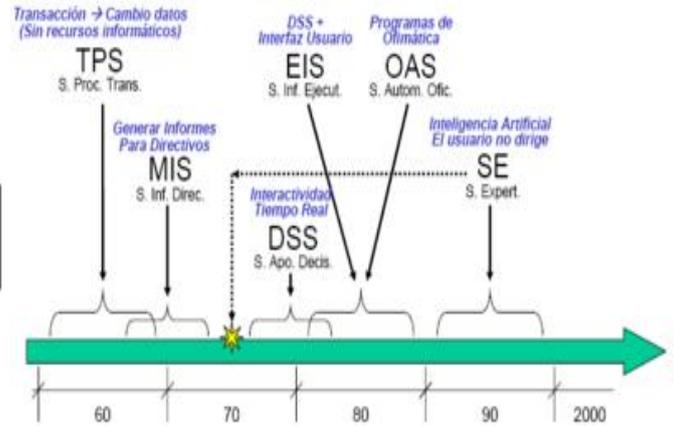
- [Aguilar and López, 2003] Aguilar, A. and López, C. (2003). El juego de planeación en el proyecto buscador de colecciones xml. En preparación.
- [Beck, 1999] Beck, K. (1999). *Extreme Programming Explained*. Addison- Wesley.
- [Beck, 2002] Beck, K. (2002). *Test-Driven Development By Example*. Addison Wesley.
- [Gamma et al., 1995] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. (1995). *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.
- [Humphrey, 2002] Humphrey, W. (2002). Comments on extreme programming. <http://www.computer.org/SEweb/Dynabook/HumphreyCom.htm>.
- [Kerievsky, 2000] Kerievsky, J. (2000). Patterns and XP. In *Extreme Programming Examined*.
- [Martin, 2002] Martin, R. (2002). RUP/XP guidelines. <http://www.rational.com/>.
- [McDowell et al., 2003] McDowell, C., Werner, L., Bullock, H., and Fernald, J. (2003). The impact of pair programming on student performance, perception and persistence.
- [Michele Marchesi, 2002] Michele Marchesi, Giancarlo Succi, D. W. L. W. (2002). *Extreme Programming Perspectives*. Addison Wesley.

ANEXOS

Anexos 1 y 2 Sistemas de información

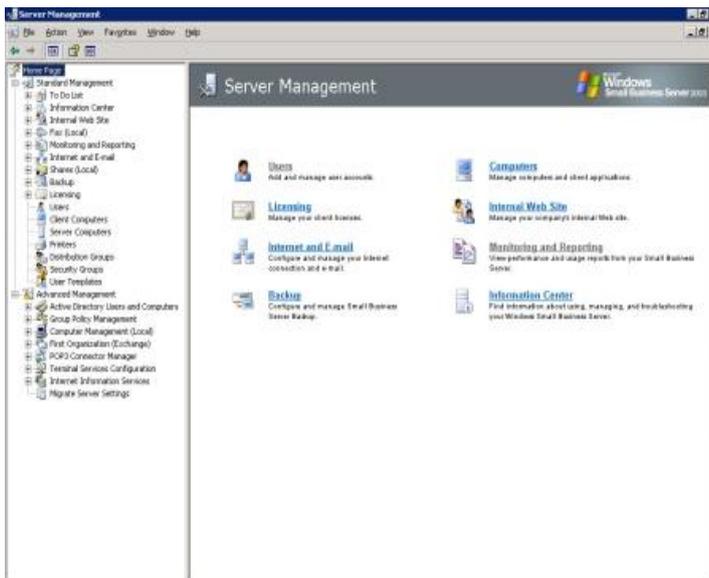


Anexo 1



Anexo 2

Anexos 3 y 4. Soluciones BackOffice para sistemas informáticos



Anexo 3

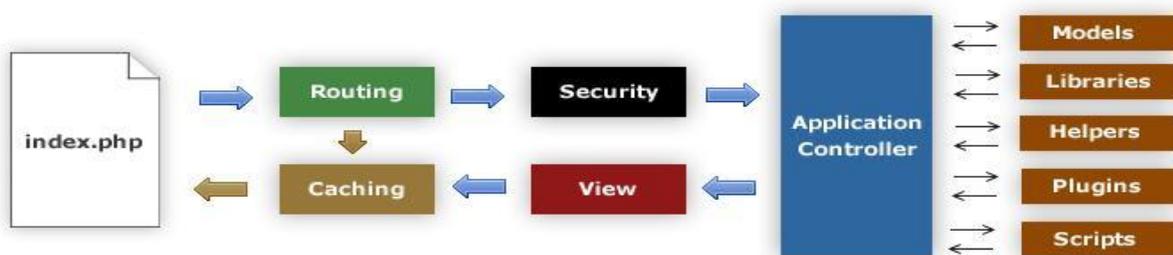


Anexo 4

Anexo 5 Funcionalidades del framework CodeIgniter

Características y funcionalidades del framework CodeIgniter				
GD, ImageMagick, y NetPBM	Seguridad y Filtro XSS	Puntos de referencia	Perfilando la Aplicación	Encriptación de Datos
Paginación	Manejo de Sesión	Clase de FTP	Scaffolding	Historial de Errores
Extremadamente Liviano	Librería XML-RPC	Compatible con PHP	Cacheo de páginas enteras	Ruteo de URI Flexible
Clases de base de datos llenas de características con soporte para varias plataformas	Librería de Manipulación de Imagen (cortar, redimensionar ó rotar). Soporta	Soporte para Ganchos, Extensiones de Clase y Plugins	Soporte de Active Record para Base de Datos	Sistema Basado en Modelo-Vista-Controlador
Clase de Motor de Plantillas	Clase de Prueba de Unidad	Clase de Trackback	Clase de Agente del Usuario	Clase de Carga (upload) de Archivo
Clase de Codificación Zip	Clase de Calendario	Larga librería de funciones "asistentes"	URLs amigables a motores de búsqueda	Formulario y Validación de Datos

Anexo 6 Flujo del Framework CodeIgniter



Anexo 7 Patrones arquitectónicos

Estilos	Flujo de datos	Centrado en datos	Llamada y retorno	Código móvil	Heterogéneos	Peer to Peer
Patrones	Tubería y filtros	Arquitecturas de Pizarra o Repositorio	Modelo Vista Controlador Arquitecturas en Capas Arquitecturas Orientadas a Objetos Arquitecturas Basadas en Componentes	Arquitectura de Máquinas Virtuales	Sistemas de control de procesos Arquitecturas Basadas en Atributos	Arquitecturas Basadas en Eventos Arquitecturas Basadas en Recursos Arquitecturas Orientadas a Servicios

Anexo 8 Patrones de diseño

Patrones de diseño utilizados en la solución	
GOF, Proxy: Estos son patrones creacionales, de comportamiento y estructurales (Adapter, Composite, Bridge). Definen las clases, problemas de herencia y polimorfismo.	Patrones GRASP: Experto, Creador (Abstract factory, Singleton), Bajo acoplamiento. Estos patrones son de asignación de responsabilidades, definen las clases que contienen la información necesaria, además de quien las crea, ambos soportan el patrón Bajo acoplamiento.

Anexo 9 Descripción de clases:

Nombre:	Nomenclador		
Tipo de clase:	entidad		
Atributo:			Tipo:
Id			integer
Nombre			varchar
Descripción			varchar
Para cada responsabilidad:			
Nombre:	Id		
Descripción:	Propiedad para identificar la identidad		

Nombre:	Nombre
Descripción:	Propiedad que refiere el nombre de la entidad
Nombre:	Descripción
Descripción:	Propiedad para caracterizar la entidad

Nombre:	Aplicación
Tipo de clase:	Entidad
Atributo:	Tipo:
Funcionalidad	Array
Descripción:	
Esta es una clase extendida de la clase Nomenclador	

Nombre:	Rol
Tipo de clase:	Entidad
Atributo:	Tipo:
Funcionalidad	Array
Descripción:	
Esta es una clase extendida de la clase Nomenclador	

Nombre:	Funcionalidad
Tipo de clase:	Entidad
Descripción:	
Esta es una clase extendida de la clase Nomenclador	
Nombre:	Usuario
Tipo de clase:	Entidad
Atributo:	Tipo:
Id	integer
Nombre	varchar
IdAplicacion	integer
IdRol	integer
Descripción	varchar

Para cada responsabilidad:	
Nombre:	Id
Descripción:	Propiedad para identificar el usuario
Nombre:	Nombre
Descripción:	Propiedad que refiere el nombre de usuario
Nombre:	IdAplicacion
Descripción:	Propiedad para identificar aplicación
Nombre:	IdRol
Descripción:	Propiedad para identificar el rol
Nombre:	Descripción
Descripción:	Describe especificaciones del usuario

Nombre:	UsuarioDominio	
Tipo de clase:	Entidad	
Atributo:	Tipo:	
IdExpediente	varchar	
Descripción:		
Esta es una clase extendida de la clase Usuario		

Nombre:	UsuarioLocal	
Tipo de clase:	Entidad	
Atributo:	Tipo:	
Clave	varchar	
Descripción:		
Esta es una clase extendida de la clase Usuario		

Clases controladoras:

Nombre:	Aplicación_c.php	
Tipo de clase:	Controladora	
Atributo:	Tipo:	

aplicacion_modelo	Aplicacion_modelo
Para cada responsabilidad:	
Nombre:	crear()
Descripción:	Método para crear una nueva aplicación
Nombre:	crear_funcionalidad()
Descripción:	Método para crear una funcionalidad
Nombre:	editar(IdAplicacion)
Descripción:	Método para editar los datos de una aplicación
Nombre:	editar_funcionalidad(IdAplicacion, IdFuncionalidad)
Descripción:	Método para editar los datos de una funcionalidad
Nombre:	actualizar()
Descripción:	Método para actualizar los datos de una aplicación
Nombre:	actualizar_funcionalidad()
Descripción:	Método para actualizar los datos de una funcionalidad
Nombre:	eliminar(IdAplicacion)
Descripción:	Método para eliminar una aplicación
Nombre:	eliminar_funcionalidad_aplicacion(IdAplicacion, IdFuncionalidad)
Descripción:	Método para eliminar una funcionalidad
Nombre:	mostrar()
Descripción:	Método para listar
Nombre:	mostrar_funcionalidad()
Descripción:	Método para listar funcionalidades

Nombre:	rol_c.php	
Tipo de clase:	Controladora	
Atributo:	Tipo:	
rol_modelo	rol_modelo	
aplicacion_modelo	Aplicacion_modelo	
Para cada responsabilidad:		
Nombre:	crear()	
Descripción:	Método para crear un rol	
Nombre:	editar(IdRol)	

Descripción:	Método para editar los datos de un rol
Nombre:	asignar_funcionalidad(IdRol, IdFuncionalidad)
Descripción:	Método para asociar funcionalidades a un rol
Nombre:	Actualizar()
Descripción:	Método para actualizar los datos de un rol
Nombre:	Eliminar(IdRol)
Descripción:	Método para eliminar un rol
Nombre:	Eliminar_funcionalidad (IdRol,IdFuncionalidad)
Descripción:	Método para eliminar una funcionalidad del rol
Nombre:	Mostrar()
Descripción:	Método para listar

Nombre:	Usuario_c.php	
Tipo de clase:	Controladora	
Atributo:	Tipo:	
aplicacion_modelo	aplicacion_modelo	
rol_modelo	rol_modelo	
usuario_modelo	usuario_modelo	
Para cada responsabilidad:		
Nombre:	crear()	
Descripción:	Método para crear un usuario	
Nombre:	editar(IdUsuario, IdAplicacion)	
Descripción:	Método para editar los datos de un usuario local	
Nombre:	Eliminar(IdUsuario)	
Descripción:	Método para eliminar un usuario local	
Nombre:	buscar(StrCadena)	
Descripción:	Método para buscar un usuario	
Nombre:	Mostrar()	
Descripción:	Método para listar	

Anexo 10 Tareas de ingeniería

Tarea de Ingeniería	
Número Tarea: 1	Número Historia de Usuario: 1
Nombre Tarea: Definir funcionalidades para aplicación en el fichero AdministracionWS.php	
Tipo de Tarea : Desarrollo	
Programador Responsable: Vladimir Campos	
Descripción: Se crea la clase entidad Nomenclador y la clase extendida (aplicación), en el fichero AdministraciónWS.php, aquí también se incluye el constructor del servicio web que establece la conexión Se declara el método InsertarAplicacion () que se le pasa un parámetro Aplicación, lo inserta en la base de datos, ObtenerAplicaciones() el cual obtiene un arreglo de aplicaciones, ObtenerAplicacionesFuncionales(), al igual que el anterior devuelve un arreglo de las aplicaciones asociadas, ObtenerAplicacionDadold(IdAplicacion).se utiliza para editar una aplicación, ActualizarAplicacion(Aplicacion), EliminarAplicacion(IdAplicacion).	

Tarea de Ingeniería	
Número Tarea: 2	Número Historia de Usuario: 1
Nombre Tarea: Crear interfaces de usuario para Aplicación	
Tipo de Tarea : Desarrollo	
Programador Responsable: Vladimir Campos	
Descripción: Se crea el formulario (frmAplicacion.php) que permite introducir los datos de la aplicación. Se valida mediante la función java script, que los datos que entra el usuario se ajusten al contenido del formulario. Se crea además una vista para visualizar, editar los datos insertados y eliminar una aplicación.	

Tarea de Ingeniería

Número Tarea: 3	Número Historia de Usuario: 1
Nombre Tarea: Definir la clase controladora aplicacion_c.php	
Tipo de Tarea : Desarrollo	
Programador Responsable: Vladimir Campos	
Descripción: Se declara la clase controladora aplicacion_c, compuesta por los métodos crear(), editar(IdAplicacion), actualizar(), eliminar(IdAplicacion), mostrar().	

Tarea de Ingeniería	
Número Tarea: 4	Número Historia de Usuario: 2
Nombre Tarea: Definir funcionalidades para Funcionalidad en el fichero AdministracionWS.php	
Tipo de Tarea : Modificación	
Programador Responsable: Vladimir Campos	
<p>Descripción: Se crea la clase funcionalidad, extendida de la clase Nomenclador en el fichero AdministraciónWS.php. Se declara el método InsertarFuncionalidadDeAplicacion(IdAplicacion, Funcionalidad) que se le pasa un parámetro IdAplicacion y el objeto Funcionalidad, para insertar en la base de datos, ObtenerFuncionalidadAplicacion(IdAplicacion, IdFuncionalidad) el cual obtiene un arreglo de funcionalidades con la aplicación asociada, ObtenerFuncionalidadesAplicacionDadoIdUsuario(IdAplicacion, IdUsuario), al igual que el anterior devuelve un arreglo de las funcionalidades pasándole los parámetros de aplicación y usuario, ObtenerFuncionalidadesDadoIdRol(IdRol) devuelve un arreglo de funcionalidades asociadas a un rol, AsignarFuncionalidadARol(IdRol, IdFuncionalidad), ActualizarFuncionalidadDeAplicacion(IdAplicacion, Funcionalidad), QuitarFuncionalidadARol(IdRol, IdFuncionalidad) EliminarFuncionalidadAplicacion(IdAplicacion, IdFuncionalidad).</p>	

Tarea de Ingeniería

Número Tarea: 5	Número Historia de Usuario: 2
Nombre Tarea: Crear interfaces de usuario para Funcionalidad	
Tipo de Tarea : Desarrollo	
Programador Responsable: Vladimir Campos	
Descripción: Se crea el formulario (frmFuncionalidad.php) que permite introducir los datos de la funcionalidad, se le asocia además una aplicación. Se valida mediante la función java script, que los datos que entra el usuario se ajusten al contenido del formulario. Se crea además una vista para visualizar, editar los datos insertados y eliminar una funcionalidad.	

Tarea de Ingeniería	
Número Tarea: 6	Número Historia de Usuario: 2
Nombre Tarea: Definir los métodos de funcionalidad en la clase controladora aplicacion_c.php	
Tipo de Tarea : Modificación	
Programador Responsable: Vladimir Campos	
Descripción: Se agregan a la clase controladora aplicacion_c, los métodos crear_funcionalidad(), editar_fucionalidad(IdAplicacion, IdFuncionalidad), actualizar_funcionalidad(), mostrar_funcionalidad(), eliminar_funcionalidad_aplicacion(IdAplicacion, IdFuncionalidad)	

Tarea de Ingeniería	
Número Tarea: 7	Número Historia de Usuario: 3
Nombre Tarea: Definir funcionalidades para rol en el fichero AdministracionWS.php	
Tipo de Tarea : Modificación	

Programador Responsable: Vladimir Campos
Descripción: Se crea la clase rol, extendida de la clase Nomenclador en el fichero AdministraciónWS.php. Se declara el método InsertarRol () se le pasa un objeto Rol, lo inserta en la base de datos. ObtenerRoles () el cual obtiene un arreglo de roles, ObtenerRolesDadold (IdRol) devuelve un rol de para editarlo, ActualizarRol (Rol).se utiliza para actualizar después de editado el rol, EliminarRol (IdRol).

Tarea de Ingeniería	
Número Tarea: 8	Número Historia de Usuario: 3
Nombre Tarea: Definir interfaces de usuario para Rol	
Tipo de Tarea : Desarrollo	
Programador Responsable: Vladimir Campos	
Descripción: Se crea el formulario (frmRol.php) que permite introducir los datos del rol, se valida mediante la función java script, que los datos que entra el usuario se ajusten al contenido del formulario. Se crea además una vista para visualizar, editar los datos insertados y eliminar un rol.	

Tarea de Ingeniería	
Número Tarea: 9	Número Historia de Usuario: 3
Nombre Tarea: Definir la clase controladora rol_c.php	
Tipo de Tarea : Desarrollo	
Programador Responsable: Vladimir Campos	
Descripción: Se agregan a la clase controladora rol_c.php, los métodos crear(), editar (IdRol), actualizar(), mostrar(), asignar_funcionalidad(IdRol, IdFuncionalidad), quitar_funcionalidad(IdRol, IdFuncionalidad), eliminar(IdRol).	

Tarea de Ingeniería	
Número Tarea: 10	Número Historia de Usuario: 5
Nombre Tarea: Definir funcionalidades para puesto en el fichero AdministracionWS.php	
Tipo de Tarea : Modificación	
Programador Responsable: Vladimir Campos	
<p>Descripción: Se crea la clase puesto, extendida de la clase Nomenclador en el fichero AdministraciónWS.php. Se declara el método InsertarPuesto() se le pasa un objeto puesto, lo inserta en la base de datos. ObtenerPuesto() el cual obtiene un arreglo de puestos, ObtenerPuestoDadold(IdPuesto) devuelve un puesto para verificar que existe. EliminarPuesto(IdPuesto) para eliminarlo.</p>	

Tarea de Ingeniería	
Número Tarea: 11	Número Historia de Usuario: 5
Nombre Tarea: Crear interfaces de usuario para puesto	
Tipo de Tarea : Desarrollo	
Programador Responsable: Vladimir Campos	
<p>Descripción: Se crea el formulario (frmPuesto.php) que permite introducir los datos del puesto, se valida mediante la función java script, que los datos que entra el usuario se ajusten al contenido del formulario. Se crea además una vista para visualizar y eliminar un puesto.</p>	

Tarea de Ingeniería	
Número Tarea: 12	Número Historia de Usuario: 3

Nombre Tarea: Definir las funcionalidades de Puesto dentro de la clase controladora aplicación_c.php
Tipo de Tarea : Modificación
Programador Responsable: Vladimir Campos
Descripción: Se agregan a la clase controladora aplicación_c.php, los métodos crear (), mostrar(), asignar_aplicación(IdPuesto, IdAplicacion), quitar_aplicación(IdPuesto, IdAplicacion), eliminar(IdPuesto).

GLOSARIO

Administrador de sistemas: Persona encargada de un sistema informático, tanto de su instalación como de su puesta a punto y mantenimiento, debiendo velar porque todo funcione de acuerdo con lo esperado. Además, el administrador tiene que hacer frente a cuantas incidencias se produzcan en el sistema, y establecer mecanismos de seguridad.

Seguridad: Asegurar que los recursos del sistema de una organización sean utilizados de la manera que se decidió y que la información que se considera importante no sea fácil de acceder por cualquier persona que no se encuentre acreditada.

URL: Del inglés Uniform Resource Locator. Localizador uniforme de recurso Es una secuencia de caracteres, de acuerdo a un formato estándar, que se usa para nombrar recursos, por su localización.

Apache: Servidor de páginas web de código abierto para diferentes plataformas (UNIX, Windows, etc.)
Aplicación: Programa con el que el usuario final interactúa a través de una interfaz, realizando tareas útiles para éste.

Arquitectura software: Conjunto de patrones y abstracciones coherentes que proporcionan el marco de referencia necesario para guiar la construcción del software para un sistema de información. Establece los fundamentos para que los desarrolladores trabajen en una línea común que permita alcanzar los objetivos y necesidades del sistema.

Capa de Datos: Es donde residen los datos. Está formada por uno o más gestores de bases de datos que realizan todo el almacenamiento de información, reciben solicitudes de almacenamiento o recuperación de información desde la capa de negocio.

Capa de Negocio: Es donde residen los programas que se ejecutan, recibiendo las peticiones del usuario y enviando las respuestas tras el proceso. Se establecen todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes y presentar los resultados, y con la capa de datos, para almacenar o recuperar datos.

Capa de Presentación: Presenta el sistema al usuario, le comunica información y captura información de éste dando un mínimo de proceso (realiza un filtrado previo para comprobar que no hay errores de formato). Esta capa se comunica únicamente con la capa de negocio.

Componente: Parte física y reemplazable del sistema que cumple y proporciona la realización de un conjunto de interfaces, ejemplo: ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

Despliegue: Cuando varios trabajos relativamente independientes (flujos de control, procesos) se distribuyen entre diferentes dispositivos hardware (procesadores).

UCI: Son las siglas de Universidad de las Ciencias Informáticas la cual fue creada, por el compañero Fidel Castro Ruz, al calor de la batalla de ideas; en el año 2002.

IDE: Entorno de Desarrollo Integrado

CSS: Cascading Style Sheets. Tecnología empleada en la creación de páginas Web, que permite un mayor control sobre el lenguaje HTML. Permite hojas de estilo que definen como cada elemento, se tiene que mostrar. CSS ha sido desarrollado por W3C.

Herramienta: Software que se utiliza para automatizar las actividades definidas en el proceso.

Herramienta CASE: Computer Aided Software Engineering, Ingeniería de Software Asistida por Ordenador. Diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero.

Open Source: Software que distribuye de forma libre su código fuente, de forma que los desarrolladores pueden hacer variaciones, mejoras o reutilización en otras aplicaciones. También conocido como software libre.

PHP: Lenguaje de programación para el desarrollo de webs dinámicas, originalmente se conocía como Personal Home Page tools, herramientas para páginas personales.

HTML: Del inglés HyperText Markup Language: Lenguaje compuesto de una serie de etiquetas o marcas que permiten definir el contenido y la apariencia de las páginas Web. Aunque se basa en el estándar SGML, no se puede considerar que sea un subconjunto de él. Existen cientos de etiquetas con diferentes atributos. W3C se encarga de su estandarización. El futuro sustituto de HTML es XHTML.

DHTML Dynamic HTML. Conjunto de extensiones a HTML que permiten modificar el contenido de una página Web en el cliente sin necesidad de establecer una conexión con el servidor. Se basa en el uso de DOM para acceder al contenido de la página.

SGML Del inglés Standard Generalized Markup Language. Lenguaje que permite organizar y etiquetar los distintos elementos que componen un documento. Se emplea para manejar grandes documentos que sufren constantes revisiones y se imprimen en distintos formatos de idioma.

XHTML Del inglés Extensible Hypertext Markup Language. HTML escrito según las normas que marca XML. Por tanto, se trata de una aplicación concreta de XML y no tienen que confundirse entre sí.

RUP Del inglés Rational Unified Process. Proceso racional unificado en el que se han agrupado técnicas de desarrollo y metodologías, está preparado para desarrollar grandes y complejos proyectos y es orientado a objetos.

XML del inglés Extensible Markup Language (Lenguaje de Marcas Extensible), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que

HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

Framework Estructura de soporte definida en la cual otro proyecto de software puede ser organizado y desarrollado. Puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros softwares para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

MVC Del inglés Model View Controller Modelo Vista Controlador, patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos.

W3C Del inglés World Wide Web Consortium. Consorcio internacional de compañías y organizaciones involucradas en el desarrollo de Internet y en especial de la WWW. Su propósito es desarrollar estándares y poner orden en Internet.

HTTP Del inglés HyperText Transfer Protocol. Es el protocolo que emplea la WWW. Define como se tienen que crear y enviar los mensajes y que opciones debe tener el servidor y el navegador en respuesta a un comando.

Sprint: Término utilizado en la metodología de desarrollo de software XP que se refiere a una iteración

Release: Término utilizado en la metodología de desarrollo de software XP que se refiere a un conjunto de sprint.