



UNIVERSIDAD DE VALPARAÍSO
Facultad de Ingeniería
Escuela de Ingeniería Civil en Informática
Ingeniería Civil Informática

Sistema de Inventarios de Aplicaciones y Servidores

Informe 3

Felipe Guillermo Aguayo Cortés

felipe.aguayoc@alumnos.uv.cl

7 de junio de 2012

Profesor Guía: Marco Aravena

Resumen

Este artículo presenta el desarrollo del marco conceptual, la fase de análisis y la fase de Diseño para la implementación de un sistema de inventario para servidores y aplicaciones en DISICO de la Universidad de Valparaíso, con la finalidad de complementar de mejor manera el registro actual de inventario realizado a los servidores pertenecientes a dicha dirección. El objetivo es entregar una herramienta que sea de fácil manipulación para cada uno de los perfiles de Usuarios que interactúan con el proceso de inventario de servidores y sus aplicaciones asociadas, con la finalidad de poseer orden, claridad y registro de la información almacenada. La aplicación a desarrollar, a diferencia de las ya existentes, será de fácil manipulación y adecuada a las necesidades planteadas por DISICO, mejorando de esta manera el proceso actual de inventario.

Índice

1. Introducción	4
2. Diseño	6
2.1. Diseño Arquitectónico	6
2.1.1. Estructuración del Sistema	6
2.1.2. Modelado de Control	11
2.1.3. Descomposición Modular	12
2.2. Diseño de Plan de Pruebas	15
2.2.1. Pruebas de Diseño:	16
2.2.2. Pruebas Unitarias:	17
Bibliografía	18

Índice de Tablas

1.	Planilla para prueba de Diseño.	16
2.	Detalle de Módulos del Sistema según su leyenda	16
3.	Detalle de Requerimientos del Sistema según su leyenda	16
4.	Plantilla para Pruebas Unitarias.	17

Índice de Figuras

1.	Modelo Cliente-Servidor.	6
2.	Arquitectura de un sitio web.	8
3.	Arquitectura del sistema dividido en 3 capas.	8
4.	Modelo perteneciente a la Capa de Negocios.	9
5.	Modelo de control del sistema.	11
6.	Descomposición modular del Sistema.	12
7.	Diagrama de Ciclo de vida de Pruebas.	15

1. Introducción

Hoy en día es muy importante para una empresa mantener un registro ordenado y claro de los bienes y servicios que pertenecen a la empresa. Esto es porque, al tener la posibilidad de conocer todo lo que se tiene como empresa, se genera una ventaja al momento de tomar decisiones sobre la finalidad de sus bienes. Por lo tanto es aconsejable que en una empresa exista algún proceso que tenga por finalidad otorgar un inventariado de los bienes y servicios que posea.

Según la rae, se define inventario como *"Asiento de los bienes y demás cosas pertenecientes a una persona o comunidad, hecho con orden y precisión"*[2]. Por lo tanto, además de poseer la información de mis bienes, es necesario que dicha información sea presentada de forma ordenada y con lujo de detalle, para que en el caso de que la empresa desee realizar alguna decisión sobre los bienes que posee, esta información sea la justificación de una buena decisión. Por lo tanto, podríamos decir que *"Saber que es lo que tengo determina que es lo que soy y ayuda de gran manera a decir que es lo que puedo ser y hacer"*.

Ahora bien, la tarea de realizar un proceso de inventario no es para nada sencillo, inclusive, al momento de disponer de muchos bienes y servicios se hace complicado obtener un registro de todo de manera clara, ordenada y precisa, y casi imposible asegurar además la disponibilidad y confiabilidad de la información. Por lo mismo, a medida que avanza el tiempo, las empresas sienten la necesidad de adquirir tecnología para el mejoramiento de sus sistemas y de sus procedimientos, con el fin de garantizar un eficaz funcionamiento.

Controlar la información de los diferentes bienes y servicios a través de un sistema computarizado, puede asegurar la disponibilidad, la confiabilidad, el orden y la claridad que se requiera en un momento determinado. Es por esto que se ha pensado en el diseño de un sistema computarizado para el inventario de aplicaciones y servidores para la Dirección de Servicios de Información y Comunicación (DISICO) de la Universidad de Valparaíso, la cual permitirá controlar de manera eficiente dicho proceso, estableciendo la mayor calidad de las actividades relacionadas con el inventariado de los servidores y sus aplicaciones instaladas, evitando de esta manera problemas como la redundancia de la información o falta de información.

Es por esto que se a planificado realizar distintas fases en el ciclo de vida del desarrollo de este sistema, primero la fase de Definición de Requerimientos y análisis, luego la fase de Diseño de la solución propuesta, posteriormente la implementación y finalmente la prueba e implantación del sistema.

Para poder llevar acabo dicha planificación, es que se optó por trabajar con una metodología incremental, determinando con esto, la forma de ejecutar dichas fases del

ciclo de vida de este sistema. Con respecto a la fase que se completará en el documento a continuación, se encuentra la fase de Diseño, detallando con esto, un bosquejo de como será el funcionamiento del sistema, para luego tomar esto como la base de la implementación del sistema.

Para exponer las temáticas que se definen en este documento, es que se a decidido estructurarlo en 2 capítulos que se definirán a continuación y el detalle de la Bibliografía utilizada para el trabajo. Por lo tanto la estructura se compone de la manera siguiente:

- Introducción.
- Diseño.
 - Diseño Arquitectónico.
 - Diseño de Pruebas.
- Bibliografía.

2. Diseño

A continuación se procederá a describir las decisiones que se tomaron y los modelos que se realizaron para el sistema propuesto en la fase de diseño, presentando en cada uno de éstos una justificación y un resguardo de la trazabilidad y consistencia entre dichos modelos.

2.1. Diseño Arquitectónico

El diseño de la arquitectura del sistema es una parte importante al momento de definir el funcionamiento del sistema, ya que define nociones de abstracción y múltiples vistas.

Es por esto que entenderemos el concepto de arquitectura de un programa o sistema computacional como *“la estructura o estructuras del sistema, la cual comprende elementos de software, las propiedades de estos elementos visibles externamente y las relaciones entre ellos”*[1].

Para definir el diseño arquitectónico del sistema se realizarán las siguientes actividades:

- Estructuración del Sistema (Modelo de Estructura).
- Modelado de Control.
- Descomposición modular.

2.1.1. Estructuración del Sistema

La arquitectura seleccionada para la implementación del sistema es una arquitectura de tipo Cliente-Servidor compuesta por 3 capas (capa de presentación, capa de negocio y capa de acceso a datos) localizadas en una misma máquina, esto se debe a que el sistema no posee una mayor complejidad como para distribuirlo en mas equipos. A continuación se presentará un modelo que muestra como se estructura una arquitectura Cliente-Servidor.

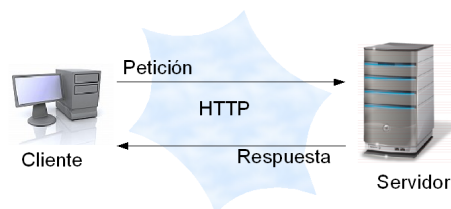


Figura 1: Modelo Cliente-Servidor.

La arquitectura cliente-servidor es un modelo de aplicación distribuida, en la cual, las tareas se reparten entre los proveedores de recursos o servicios, determinados como servidores, y los demandantes, establecidos como clientes. En otras palabras, un cliente solicita un servicio y un servidor se lo proporciona. La definición otorgada por IBM sobre Cliente-Servidor es la siguiente:

“Es la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o, a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o clientes, resultan en un trabajo realizado por otros computadores llamados servidores”.

Ahora bien, los componentes de la arquitectura cliente-servidor los entenderemos como se presenta a continuación:

- **Cliente:** Es el componente encargado de solicitar servicios de internet o intranet. Una aplicación consta de una parte determinada como cliente, los cuales pueden ser ejecutados en diferentes sistemas. Los usuarios finales invocan la parte cliente de la aplicación, que construye una solicitud, y se la envía al servidor.

Las funciones que lleva a cabo el cliente se resumen en lo siguiente:

- Administrar la interfaz de usuario.
 - Interactuar con el usuario.
 - Generar requerimientos de Bases de Datos.
 - Interpretar los resultados del servidor.
 - validar datos a nivel local.
- **Servidor:** Es el componente que ofrece los servicios solicitados por los clientes. El servidor es un programa que recibe una solicitud, realiza la acción requerida y devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo.

Las funciones que lleva a cabo el servidor se resumen en lo siguiente:

- Gestionar los requerimientos de Bases de Datos que hacen los clientes.
- Formatear datos para transmitirlos a los clientes
- Procesar la lógica de la aplicación y realizar validaciones a nivel de bases de datos.

Los servidores web utilizan la arquitectura Cliente-Servidor adaptando los aspectos y tecnologías por nuevas y propias a la misma, detallando de esta manera tres componentes principales: un servidor web, una conexión de red y uno o más clientes (Browsers). En la imagen a continuación se presenta la forma en que se relacionan dichos componentes en la arquitectura de un sitio web.

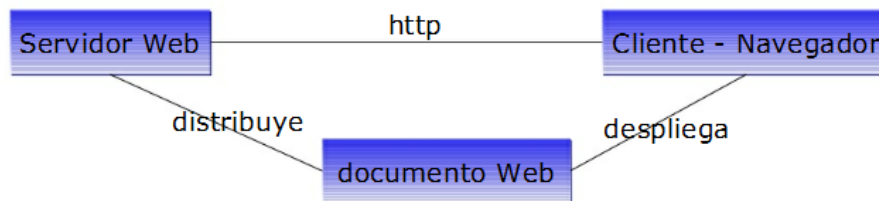


Figura 2: Arquitectura de un sitio web.

Ahora bien, una de las justificaciones de porqué se seleccionó esta arquitectura es que presenta la capacidad de poder realizar el desarrollo de su funcionamiento en niveles, en donde cada una de estas capas se desentiende de lo que ocurre en las capas superiores y solo se dedica a trabajar en la finalidad que le corresponde interactuando con sus capas vecinas. También es necesario determinar que la búsqueda de errores se simplifica, ya que al poseer el sistema dividido en capas separadas, no es necesario indagar sobre código mezclado. En la figura que se presenta a continuación se presenta la forma en que se definen las capas y como estas interactúan, determinando así, la arquitectura del sistema.



Figura 3: Arquitectura del sistema dividido en 3 capas.

Se observa en la Figura 3 el sistema dividido en 3 capas: capa de presentación, capa de negocios y capa de acceso a datos. A continuación se detallara el enfoque tomado en ellas.

- **Capa de Presentación:**

Esta capa, también conocida como capa de acceso o de usuario, es la que se encarga de que el sistema interactúe con el usuario y viceversa, ya que muestra el sistema al usuario, le presenta la información y obtiene información del usuario en un algún proceso del sistema. En una aplicación corresponde a la interfaz gráfica y debe tener la característica de ser entendible y fácil de usar para el usuario. Además esta capa no se debe de ocupar para el procesamiento de la información, al menos que sean validaciones de forma local. Esta capa se comunica únicamente con la capa intermedia o de negocio.

- **Capa de Negocios:**

Esta capa, también conocida como capa de procesos o lógica del negocio, es donde residen las funciones que se ejecutan en el sistema, se procesa la información y se envían las respuestas tras el proceso. En esta capa es donde se estableces todas las reglas que deben cumplirse. Esta capa se comunica con la capa de presentación, para recibir las solicitudes de los usuarios y presentar los resultados de dichas solicitudes, y con la capa de acceso a datos, para solicitar al gestor de base de datos almacenar, modificar o recuperar datos de él.

Los módulos que componen ésta capa corresponden a los presentados en el siguiente diagrama:



Figura 4: Modelo perteneciente a la Capa de Negocios.

Como se puede observar en la figura 4, la *capa de Negocios* del sistema se encuentra dividida en 6 módulos, los cuales se encuentran descritos a continuación:

- **Identificación:** Éste módulo se encuentra encargado de la verificación de los datos para el ingreso de los distintos usuario (administrador, Encargado de Datacenter, Encargado de Área, Encargado de Servidor, Encargado

Aplicación) al sistema, además es posible iniciar o detener el sistema.

- **Administración Usuarios:** Éste módulo está enfocado a realizar la administración de los usuarios que pertenecen al sistema, y otorga todas las herramientas necesarias para la gestión de éstos.
- **Administración Servidores:** Éste módulo se encuentra destinado para la administración de los servidores pertenecientes al sistema, además otorga toda las herramientas que se necesitan para su gestión.
- **Administración Aplicaciones:** Éste módulo se encuentra destinado para la administración de las Aplicaciones pertenecientes al sistema, además otorga toda las herramientas que se necesitan para su gestión.
- **Administración UPS:** Éste módulo se encuentra destinado para la administración de las UPS pertenecientes al sistema, además otorga toda las herramientas que se necesitan para su gestión.
- **Gestión Información:** Éste módulo se encuentra enfocado en la gestión de la información perteneciente al sistema, esto es, generación de estadísticas e informes, entregando con esto las herramientas necesarias para su gestión.
- **Conexiones:** Este módulo se encarga específicamente de las conexiones a la base de datos del sistema. Es el interprete para la base de datos, ya que se conecta directamente con la capa de Datos.

Para acceder a los datos pertenecientes al Sistema, cada uno de los módulos identificados anteriormente interactúan con el módulo de conexiones, que se comunica con la capa de datos.

■ **Capa de Acceso a Datos:**

Esta capa, también conocida como Capa de Datos, es la encargada de almacenar los datos del sistema y de los usuarios. La principal función de ésta es almacenar, modificar y retornar datos a la capa de negocio, aunque para esto también es necesario en algunos casos, que tenga procedimientos almacenados y funciones dentro de la capa. En una arquitectura de tres capas es la única que puede acceder a los datos. Está formada por uno o varios sistemas gestores de bases de datos (DBMS), localizados en un mismo servidor o en varios. Solo se comunica con la capa de negocios para enviar o recibir información.

Todas estas capas se encontrarán ubicadas de forma física en un único equipo (servidor), debido a que la complejidad del sistema no es alta, aunque si en algún momento el sistema aumentara en complejidad, la forma en que se encuentra estructurado el sistema, permitirá situar alguna o todas las capas en distintos equipos.

2.1.2. Modelado de Control

Para que cada uno de los componentes que se integran en el sistema trabajen en conjunto e interactúen con los usuarios, es que se debe de determinar un estilo de control para que los servicios, que se proveen dentro del sistema, sean entregados y respondidos en el lugar y tiempo correcto.

Es por esto que se a decidido optar por un estilo de control basado en Eventos de tipo invocación implícita (broadcast), ya que los modelos de control basado en eventos se rigen por eventos generados de forma externa al sistema, esto es, el timing del evento está fuera del control del componente que maneja ese evento.

Ahora cabe señalar que se escogió que fuera de tipo invocación implícita debido a que cumple con las condiciones que el sistema requiere, ya que en el caso de que un evento del sistema sea producido de forma externa por un usuario, se debe transmitir a todos los componentes, por lo cual, cualquier componente que pueda manejar ese evento debe poder hacerlo, situación que es satisfecha por este modelo.

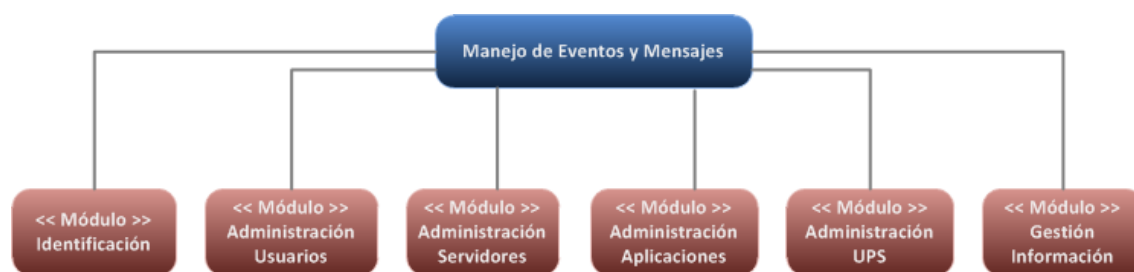


Figura 5: Modelo de control del sistema.

En la figura 5 se identifican los módulos del sistema que interactúan con los distintos usuarios pertenecientes al sistema, en la sección 2.1.3 se detallará la descomposición modular de cada uno de éstos. Además se puede observar que cada uno de los módulos identificados pueden y deben responder los eventos que son generados por el usuario de forma externa, entregando de la misma manera un mensaje con la respuesta a dicho evento.

2.1.3. Descomposición Modular

En esta sección se procederá a descomponer cada uno de los módulos identificados en la sección 2.1.1 de la capa de Negocios. Se puede observar en la Figura 6 la composición de dichos módulos, identificando como se relacionan los componentes que les pertenecen y la implementación del patrón DAO (Módulo Conexiones) para lograr orden en el diseño y la independencia de la capa de Datos de la capa de negocios.

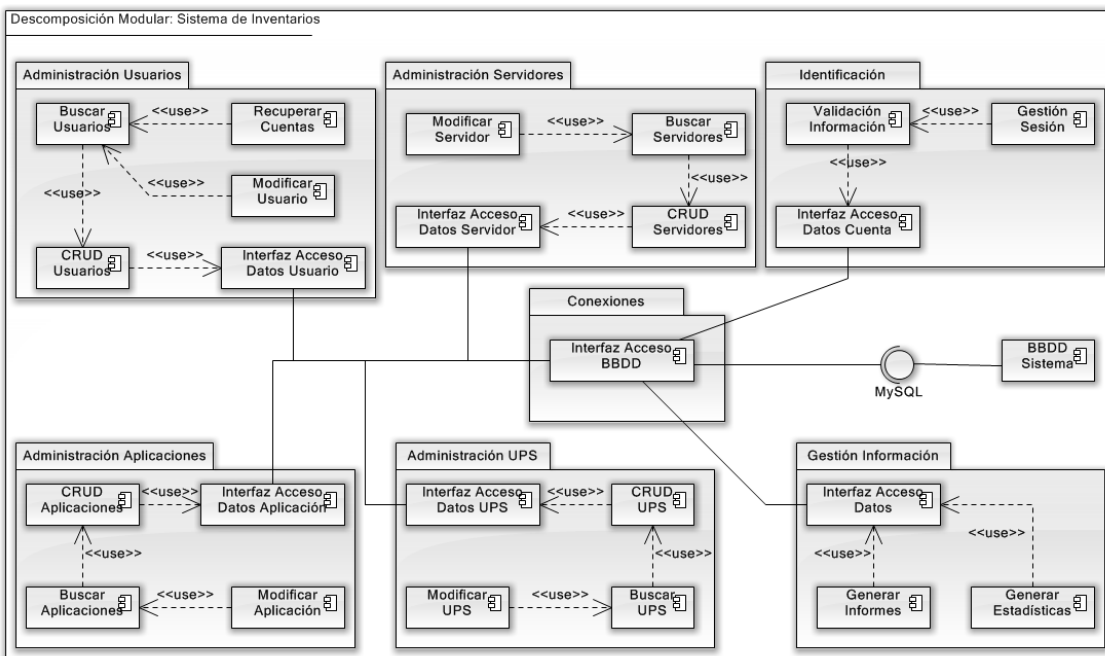


Figura 6: Descomposición modular del Sistema.

En base al diagrama presentado en la Figura 6, se realizará la identificación y descripción (breve) de cada uno de los componentes pertenecientes a cada módulo.

■ Identificación:

- **Gestión Sesión:** Componente encargado de gestionar la sesión del usuario, ya sea para iniciar sesión dentro del sistema, como para cerrarla.
- **Validación Información:** Componente encargado de validar la información ingresada con la información del sistema. Se comunica con el componente Interfaz Acceso Datos Cuenta para obtener los datos del sistema.
- **Interfaz Acceso Datos Cuenta:** Componente que permite la conexión del módulo Identificación con el módulo Conexión para obtener los datos de la base de datos del sistema.

■ **Administración Usuarios:**

- **Modificar Usuario:** Componente encargado de modificar los datos de un Usuario perteneciente al sistema, el cual es obtenido del componente Buscar Usuario.
- **Recuperar Cuentas:** Componente encargado de Recuperar la cuentas de Usuario que han sido desactivadas o han perdido su contraseña utilizando para esto el componente Buscar Usuario.
- **Buscar Usuario:** Componente encargado de Buscar usuarios en el sistema, ya sea para listarlos todos o para encontrar a uno o varios en el sistema.
- **CRUD Usuario:** Componente encargado de crear (C), leer (R), modificar (U) y desactivar (D) usuarios en el sistema.
- **Interfaz Acceso Datos Usuario:** Componente que permite la conexión del módulo Administración Usuarios con el módulo Conexión para obtener los datos de la base de Datos del Sistema.

■ **Administración Servidores:**

- **Modificar Servidor:** Componente encargado de gestionar la modificación de un servidor en el sistema.
- **Buscar Servidores:** Componente encargado de Buscar usuarios en el sistema, ya sea para listarlos todos o para encontrar a uno o varios en el sistema.
- **CRUD Servidores:** Componente encargado de crear (C), leer (R), modificar (U) y desactivar (D) Servidores en el sistema.
- **Interfaz Acceso Datos Servidor:** Componente que permite la conexión del módulo Administración Servidores con el módulo Conexión para obtener los datos de la base de Datos del Sistema.

■ **Administración Aplicaciones:**

- **Modificar Aplicación:** Componente encargado de gestionar la modificación de una aplicación en el sistema.
- **Buscar Aplicaciones:** Componente encargado de Buscar aplicaciones en el sistema, ya sea para listarlas todas o para encontrar a una o varias en el sistema.
- **CRUD Aplicaciones:** Componente encargado de crear (C), leer (R), modificar (U) y desactivar (D) Aplicaciones en el sistema.
- **Interfaz Acceso Datos Servidor:** Componente que permite la conexión del módulo Administración Aplicaciones con el módulo Conexión para obtener los datos de la base de Datos del Sistema.

■ **Administración UPS:**

- Modificar UPS: Componente encargado de gestionar la modificación de una UPS en el sistema.
- Buscar UPS: Componente encargado de Buscar UPS en el sistema, ya sea para listarlas todas o para encontrar a una o varias en el sistema.
- CRUD UPS: Componente encargado de crear (C), leer (R), modificar (U) y desactivar (D) UPS en el sistema.
- Interfaz Acceso Datos UPS: Componente que permite la conexión del módulo Administración UPS con el módulo Conexión para obtener los datos de la base de Datos del Sistema.

■ **Gestión Información:**

- Generar Estadísticas: Componente encargado de generar las estadísticas del sistema.
- Generar Informe: Componente encargado de generar los informes del sistema.
- Interfaz Acceso Datos: Componente que permite la conexión del módulo Gestión Información con el módulo Conexión para obtener los datos de la base de Datos del Sistema.

■ **Conexiones:**

- Interfaz Acceso BBDD: Componente encargado de realizar consultas y operaciones en la base de datos del sistema, utilizando el lenguaje SQL de MySQL.

■ **Bases de Datos:**

- BBDD Sistema: Base de datos del sistema Administrada con MySQL.

2.2. Diseño de Plan de Pruebas

El objetivo de ésta sección es presentar en detalle el Diseño de Pruebas que se ejecutarán para evaluar que tan defectuoso se encuentra el sistema luego de su implementación. Las pruebas comprendidas en detalle serán las Pruebas de Diseño, Pruebas Unitarias, Pruebas de Integración, Pruebas de Sistema y Pruebas de Aceptación.

En el diagrama que se presenta en la figura 7 se muestra las faces del Ciclo de vida del proceso de Diseño de pruebas que se deberá llevar a cabo o tomar en cuenta, dependiendo del caso, al momento de generar alguna de las pruebas en el sistema.



Figura 7: Diagrama de Ciclo de vida de Pruebas.

Cabe destacar que en este diseño de pruebas no se detallaran los casos de prueba de forma específica (esto es dar el dato en duro, ej: rut: 12.345.678-9), esto debido a que como aun no se implementa el sistema ni tampoco se cubren los requerimientos, ni se definen con certeza los datos que manejará el sistema, no es posible realizar dicho nivel de detalle. Por lo tanto en ésta sección se expondrá y determinará los casos de pruebas como modelos a realizar al momento de probar más adelante.

2.2.1. Pruebas de Diseño:

El principal objetivo de esta prueba es el de satisfacer los requerimientos en el diseño del sistema, por lo que se hace importante destacar que cada requerimiento debe estar implementado en algún módulo del sistema en específico, para poder lograr esto se hace un cruzamiento entre los requerimientos y módulos del sistema de la sección 2.1.3 para asegurar de esta manera el cumplimiento de éstos en la arquitectura. La tabla 1 presentada a continuación presenta el formato de esta prueba, para que sea utilizado al momento de realizar esta prueba. La persona encargada de realizar esta tarea es el arquitecto de Software, responsabilidad que será tomada por el autor de este trabajo.

Prueba de Diseño						
Requerimiento \ Módulo	Mod_1	Mod_2	Mod_3	...	Mod_{n-1}	Mod_n
RF_1	X	X				
RF_2			X			
RF_3	X					X
...				...		
RF_{n-1}					X	
RF_n						X

Tabla 1: Planilla para prueba de Diseño.

Detalle de Módulos		
Leyenda	Nombre	Descripción
Mod_1		
Mod_2		
...		
Mod_n		

Tabla 2: Detalle de Módulos del Sistema según su leyenda

Detalle de Requerimientos	
Leyenda	Descripción
RF_1	
RF_2	
...	
RF_n	

Tabla 3: Detalle de Requerimientos del Sistema según su leyenda

2.2.2. Pruebas Unitarias:

El objetivo principal de este tipo de pruebas es encontrar errores basándose en la estructura o especificaciones. Es útil principalmente para verificar que cada módulo del sistema funcione de manera correcta por separado. El formato del detalle de la información de esta prueba debe de responder al presentado en la Tabla 4, el cual debe de ser ocupado al momento de ejecutar la prueba. Los encargados de realizar estas pruebas son en un primer lugar el programador y luego el tester, uno en la fase de programación y el otro en la fase de testing, respectivamente. En el caso de este Sistema, dichos cargos serán cubiertos por el creador de este trabajo de título.

Datos Iniciales - Prueba Unitaria		
Fecha:	N° Caso de Prueba:	ID Prueba:
Característica:	Subcaracterística:	
Módulo de Chequeo		
Informe del Caso de Prueba		
Descripción:	Enfoque:	
Datos Entrada:	Datos Esperados:	
Procedimiento del Caso de Prueba		
1. Pasos a Seguir:		
2. Condiciones Externas:		
Resultados de Ejecución		

Tabla 4: Plantilla para Pruebas Unitarias.

Bibliografía

- [1] Len Bass, Paul Clements and Rick Kazman. *Software Architecture in Practice*. Addison Wesley, 2003. Segunda Edición.
- [2] Real Academia Española. Definición de inventario. <http://buscon.rae.es/draeI/SrvltGUIBusUsual?LEMA=inventario&origen=RAE>. Última vez visitado, 7 de junio de 2012.