

CONSTRUCCIÓN DE UN SISTEMA DE ALTA DISPONIBILIDAD PARA
MANEJO DE DOCUMENTOS.

PAULA ANDREA GUZMÁN MOYA
20151678006

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD TECNOLÓGICA
INGENIERÍA EN TELEMÁTICA
BOGOTÁ D.C
2018

CONSTRUCCIÓN DE UN SISTEMA DE ALTA DISPONIBILIDAD PARA
MANEJO DE DOCUMENTOS.

PAULA ANDREA GUZMÁN MOYA
20151678006

LUIS FELIPE WANUMEN SILVA
DIRECTOR INTERNO

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS
FACULTAD TECNOLÓGICA
INGENIERIA EN TELEMÁTICA
BOGOTÁ D.C
2018

DEDICATORIA

Deseo dedicar este proyecto a Dios, por darme la fuerza necesaria cuando todo anda mal, por ser siempre el motivo de mis alegrías y mis triunfos en toda etapa de mi vida, colocando a mi alrededor personas muy especiales, que siempre creyeron en mi capacidad y esfuerzo por ser una mejor persona cada día, ya sea a nivel personal como profesional.

A mi familia, por ser los cimientos que me sostienen, me guían, están conmigo siempre, sin importar la situación buena o mala. Doy infinitas gracias por todas las cosas y valores inculcados, por sus palabras, energía e incondicional acompañamiento.

Al ejemplo más claro de esfuerzo y dedicación, como guía a tener en cuenta para ser una buena profesional. Profesor Luis Felipe Wanumen Silva, gracias a sus clases, disposición, apoyo, incluso charlas que me permitieron aprender cosas muy valiosas, deseo expresar mi gratitud hacia usted deseándole éxito en todo momento.

Por último, a todas aquellas personas que conocí y en algún momento compartí, me llevo siempre presente todo los buenos momentos, gracias por permitirme conocerlos, a todo ellos éxitos en su trayectoria profesional.

Paula Andrea Guzmán Moya

TABLA DE CONTENIDO

	pág.
LISTA DE ANEXOS	8
GLOSARIO	9
RESUMEN	10
INTRODUCCIÓN	11
1. PLANTEAMIENTO DEL PROBLEMA	12
1.1 Descripción del Problema	12
2. FORMULACIÓN DEL PROBLEMA	13
3. JUSTIFICACIÓN	14
4. OBJETIVOS	15
4.1 Objetivo General	15
5. ALCANCE	16
6. MARCO REFERENCIAL	17
6.1 Marco Histórico	17
6.1.1 Historia de Sistema de Alta Disponibilidad	17
6.1.2 Historia Servicios Web	21
6.2 Marco Teórico	23
6.2.1 Arquitectura de las Aplicaciones Web	23
6.2.2 Protocolos de Aplicación más Utilizados	26
6.2.3 Sistema de Alta Disponibilidad	28
6.2.4 PHP	29
6.2.5 Laravel	30
6.2.6 Servidor Web	31
6.2.7 Servicios	32
6.2.8 Servidores más Usados	33
6.2.9 Aplicaciones en el Lado del Servidor	34
6.2.10 Ambiente Virtual de Aprendizaje	35
6.3 Marco Tecnológico	36
7. IMPACTOS	37
8. ENFOQUE DE LA ARQUITECTURA – DISEÑO METODOLÓGICO	38
9. OBJETIVOS ARQUITECTÓNICOS PARA LOGRAR ALTA DISPONIBILIDAD	39
9.1 Meta arquitectónica	39
9.2 Lineamientos arquitectónicos	40

10.	SERVICIOS ARQUITECTONICOS.....	41
11.	IMPLEMENTACIÓN SERVICIOS ARQUITECTONICOS	47
11.1	Descripción Servicio de Aplicación Seguridad HTTP.....	47
11.2	Descripción Servicio de Aplicación Seguridad en el Envío de Datos.....	49
11.3	Descripción Servicio de Aplicación Reutilización de Frameworks Probados en el Mercado.....	50
11.4	Descripción Servicio de Aplicación Configuración del Entorno.....	51
11.5	Descripción Servicio de Aplicación Bases de Datos.....	52
11.6	Descripción Servicio de Hosting en las Aplicaciones	53
11.7	Descripción Servicio de Recursos	54
11.8	Descripción Servicio de Almacenamiento	55
11.9	Descripción Servicio de Testeo.....	56
11.10	Descripción Servicio de Actualización.....	57
12.	VISUALIZACIÓN DE LA IMPLEMENTACION DEL SISTEMA.....	59
13.	PRUEBAS TAM HECHAS AL SISTEMA	65
	CONCLUSIONES	71
	BIBLIOGRAFIA.....	72

TABLA DE ILUSTRACIONES

	pág.
Ilustración 1: Fuente Autora: Subsistema que soporta la arquitectura.....	13
Ilustración 2: Fuente Autora: Arquitectura de las aplicaciones web.	23
Ilustración 3: Fuente Autora: Capa de datos (servidor de datos)	26
Ilustración 4: Fuente Autora: Enfoque de la arquitectura, diseño lógico.	38
Ilustración 5: Fuente Autora: Meta arquitectónica	39
Ilustración 6: Fuente Autora: Lineamientos arquitectónicos.....	40
Ilustración 7: Fuente Autora: Servicios arquitectónicos	41
Ilustración 8: Fuente Autora: Descripción de aplicación seguridad.....	42
Ilustración 9: Fuente Autora: Descripción de la aplicación seguridad en el envío de datos.....	42
Ilustración 10: Fuente Autora: Descripción servicio de aplicación reutilización de Frameworks.....	43
Ilustración 11: Fuente Autora: Descripción servicio de aplicación configuración del entorno.	43
Ilustración 12: Fuente Autora: Descripción servicio de aplicación bases de datos	44
Ilustración 13: Fuente Autora: Descripción servicio de hosting en las aplicaciones.....	44
Ilustración 14: Fuente Autora: Descripción servicio de recursos.....	45
Ilustración 15: Fuente Autora: Descripción servicio de almacenamiento	45
Ilustración 16: Fuente Autora: Descripción servicio de testeo.	46
Ilustración 17: Fuente Autora: Descripción servicio de actualización.	46
Ilustración 18: Fuente Autora: Descripción servicio de aplicación seguridad HTTP	47
Ilustración 19: Fuente Autora: Paquete middlewere.....	48
Ilustración 20: Fuente Autora: Paquete request	48
Ilustración 21: Fuente Autora: Descripción servicio de aplicación seguridad en el envío de datos.	49
Ilustración 22: Fuente Autora: Soporte para seguridad.....	49
Ilustración 23: Fuente Autora: Soporte para interoperabilidad en mensajes.	50
Ilustración 24: Fuente Autora: Soporte para recuperabilidad.	50
Ilustración 25: Fuente Autora: Descripción servicio aplicación reutilización de Frameworks probados en el mercado.	50
Ilustración 26: Fuente Autora: Soporte de escalabilidad en el ambiente.....	51
Ilustración 27: Fuente Autora: Descripción del servicio aplicación para configuración del entorno.	51
Ilustración 28: Fuente Autora: Soporte a configuraciones.	52
Ilustración 29: Fuente Autora: Descripción servicio de aplicación bases de datos.	52
Ilustración 30: Fuente Autora: Soporte a escalabilidad en bases de datos.	53
Ilustración 31: Fuente Autora: Descripción servicio hosting en las aplicaciones.....	53
Ilustración 32: Fuente Autora: Soporte para almacenamiento de recursos.....	54
Ilustración 33: Fuente Autora: Descripción servicio de recursos.....	54
Ilustración 34: Fuente Autora: Soporte a recursos internos con los que trabaja el framework.....	55
Ilustración 35: Fuente Autora: Descripción servicio de almacenamiento.	55
Ilustración 36: Fuente Autora: Configuración en el acceso a archivos.	56
Ilustración 37: Fuente Autora: Configuración en el envío de correos.	56
Ilustración 38: Fuente Autora: Descripción servicio de testeo.	56
Ilustración 39: Fuente Autora: Soporte a disponibilidad.	57
Ilustración 40: Fuente Autora: Soporte a pruebas.....	57
Ilustración 41: Fuente Autora: Descripción servicio de actualización.	57

Ilustración 42: Fuente Autora: Soporte actualizaciones de servidor.	58
Ilustración 43: Fuente Autora: Soporte actualizaciones de plataforma.	58
Ilustración 44: Fuente Autora: Soporte actualización de Framework.	58
Ilustración 45: Fuente Autora: Interfaz principal de acceso al sistema.	59
Ilustración 46: Fuente Autora: Interfaz usuario estudiante.	60
Ilustración 47: Fuente Autora: Interfaz temario estudiante.	60
Ilustración 48: Fuente Autora: Interfaz evaluación estudiante.	61
Ilustración 49: Fuente Autora: Interfaz usuario administrador.	61
Ilustración 50: Fuente Autora: Interfaz usuarios.	62
Ilustración 51: Fuente Autora: Interfaz temas.	62
Ilustración 52: Fuente Autora: Interfaz subtemas.	63
Ilustración 53: Fuente Autora: Interfaz de preguntas.	63
Ilustración 54: Fuente Autora: Interfaz de habilitar preguntas.	64
Ilustración 55: Fuente Autora: Interfaz de respuestas.	64
Ilustración 56: Fuente Autora: Interfaz resultados de los alumnos.	65
Ilustración 57: Fuente: Davis [2]: Modelo de Aceptación de la Tecnología.	66

LISTA DE ANEXOS

Anexo 1: Manual Usuario

GLOSARIO

AMBIENTE VIRTUAL: Es una herramienta tecnológica o plataforma educativa que, valiéndose de las propiedades de computabilidad, conectividad y virtualidad, permite la edición y desarrollo de cursos no presenciales o experiencias educativas para apoyar la enseñanza presencial.

SUBSISTEMA: Es un conjunto de elemento interrelacionados que, en sí mismo, es un sistema, pero a la vez es parte de un sistema superior.

APLICACIONES WEB: Son un tipo de software que se codifica en un lenguaje soportado por los navegadores web y cuya ejecución es llevada a cabo por el navegador en Internet o de una intranet (de ahí que reciban el nombre de App web).

DISPONIBILIDAD: Se refiere a la habilidad de la comunidad de usuarios para acceder al sistema, someter nuevos trabajos, actualizar o alterar trabajos existentes o recoger los resultados de trabajos previos. Si un usuario no puede acceder al sistema se dice que está no disponible. El término tiempo de inactividad (downtime) es usado para definir cuándo el sistema no está disponible.

FRAMEWORKS: Es un esquema (un esqueleto, un patrón) para el desarrollo y/o la implementación de una aplicación. Sí, es una definición muy genérica, pero también puede serlo un framework: sin ir más lejos, el paradigma MVC (Model-View-Controller) dice poco más que “separa en tu aplicación la gestión de los datos, las operaciones, y la presentación”.

FIABILIDAD: Se define como la probabilidad de que un bien funcione adecuadamente durante un período determinado bajo condiciones operativas específicas

RESUMEN

EL desarrollo de este proyecto toca como tema el diseño de un sistema de alta disponibilidad, implementado en un software prototipo, como una herramienta de manejo de documentos para cualquier organización. En virtud de lo anterior se hace necesario comprender los conceptos de alta disponibilidad, ambiente virtual, sistemas modulares y seguridad.

ABSTRACT

The development of this project touches on the design of a high availability system, implemented in prototype software, as a document management tool for any organization. In view of the above, it is necessary to understand the concepts of high availability, virtual environment, modular systems and security.

INTRODUCCIÓN

El presente proyecto de grado tiene como objetivo desarrollar un software de alta disponibilidad que permita almacenar información para ser consultada en la web, ofreciendo a los usuarios la posibilidad de registrarse, visualizar contenido y realizar una evaluación, dicho sistema contiene una serie de subsistemas como: de fiabilidad, de recuperación, de detección de errores y de operaciones continuas que garantizan la arquitectura adecuada para avalar la alta disponibilidad, utilizando lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML como lo es PHP.

Para llevar a cabo este proyecto se realizó una investigación amplia sobre arquitecturas de aplicaciones web, protocolos, servidores web, estructuras de un sistema de alta disponibilidad y lenguajes de programación que implementen framework con funcionalidades que garanticen desarrollar este tipo de sistemas como el propuesto de forma modular y segura.

Los resultados de esta construcción permitirán al grupo de investigación METIS contar con un prototipo que permita implementar mejores y más robustos sistemas de alta disponibilidad en cualquier área o enfoque que se necesite.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 Descripción del Problema

Actualmente uno de los problemas principales de toda empresa es no contar siempre con sus aplicaciones, las cuales apoyan sus procesos por la falta de disponibilidad, implicando costos, tiempos y pérdidas que se ven reflejadas en sus operaciones además de la insatisfacción de sus clientes. Es por ello que se ve la importancia de contar con aplicaciones web con una arquitectura que permita garantizar escalabilidad, alta disponibilidad, alto rendimiento seguido de retos como velocidad, seguridad y compatibilidad en sistemas web. Se deben investigar los elementos claves que debe tener una arquitectura de alta disponibilidad.

Una vez extractados estos elementos, surge la pregunta de investigación: ¿Es posible desarrollar un sistema de alta disponibilidad sobre tecnologías PHP para el manejo de documentos de texto?

Se propone un prototipo de diseño para sistemas de alta disponibilidad para el manejo de documentos de texto, en base al estudio de las diferentes y principales alternativas tecnológicas existentes en el mercado.

2. FORMULACIÓN DEL PROBLEMA

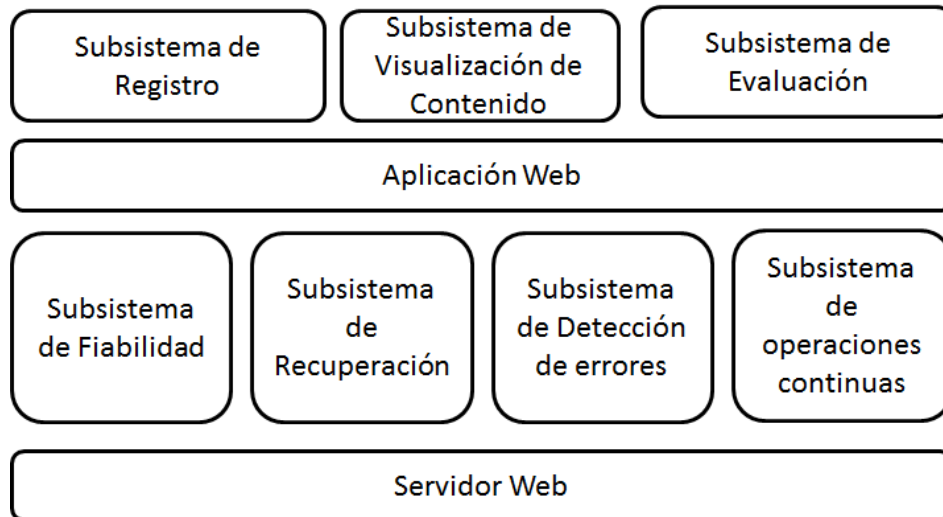


Ilustración 1: Fuente Autora: Subsistema que soporta la arquitectura.

Se plantea el desarrollo de un software web que se ejecute sobre un servidor web. Dicho software debe implementar un subsistema de fiabilidad, de recuperación, de detección de errores y de operaciones continuas tal como se muestra en la figura anterior. Sobre este sistema de alta disponibilidad se plantea la construcción de una aplicación web de gestión documental que incluya un subsistema de registro, de visualización de contenido y de evaluación.

3. JUSTIFICACIÓN

Actualmente, se evidencia la importancia de contar con sistemas informáticos seguros y fiables, siendo fundamentales para todo modelo de negocio. Es una escena habitual que las organizaciones trabajen y produzcan en base a los servidores y datos de los que disponen, y que el éxito final de su modelo de negocio dependa de cómo sus servicios son informatizados. Las tecnologías de la información se han convertido, con el paso de los años, en un elemento clave para la competitividad de las organizaciones, razón por la cual se ve la importancia de contar con sistemas web de alta disponibilidad.

Lo que se quiere lograr con el proyecto es mostrar una alternativa al uso de la tecnología y sobre todo brindar una solución tecnológica a un problema común del uso de aplicaciones web, servicios e información en la nube para que sean accesibles, de alta disponibilidad y seguros.

4. OBJETIVOS

4.1 Objetivo General

Diseñar e implementar un modelo de alta disponibilidad bajo plataforma PHP para manejo de documentos con subsistemas de fiabilidad, de recuperación, de detección de errores y de operaciones continuas.

4.2 Objetivo Específico

- ✓ Analizar un subsistema de fiabilidad que permita la ejecución de un software prototipo bajo unas condiciones definidas a la largo del proyecto de grado.
- ✓ Diseñar un subsistema de recuperación que permita a los usuarios reestablecer el acceso cuando han perdido los datos de ingreso al software prototipo.
- ✓ Implementación de un sistema de detección de errores que permita encapsular y gestionar los errores por medio de logs.
- ✓ Validar la arquitectura propuesta mediante la construcción de un sistema de gestión de documentos de texto e imágenes que implemente un módulo de registro de usuario, de visualización de contenido y de evaluación.

5. ALCANCE

Con el desarrollo de este sistema de alta disponibilidad se espera:

- ✓ Conocer los puntos claves que se deben tener en cuenta al momento de diseñar un sistema de alta disponibilidad.
- ✓ Partiendo del procesos investigativo, lograr el desarrollo de un software web que contenga una arquitectura de alta disponibilidad, utilizando software libre.
- ✓ Cumplir con el proceso del diseño del subsistema de recuperación que permita a los usuarios reestablecer el acceso cuando han perdido los datos de ingreso al software prototipo.
- ✓ Garantizar el funcionamiento de la aplicación web de gestión documental que incluya un subsistema de registro, de visualización de contenido y de evaluación.

6. MARCO REFERENCIAL

6.1 Marco Histórico

6.1.1 Historia de Sistema de Alta Disponibilidad

Martin (1999, p.107) define la disponibilidad como “la cantidad o porcentaje de tiempo que un sistema se encuentra disponible para la realización de una determinada misión”. Sin embargo algunos autores consideran que no es solo un problema de aplicación sino del entorno que se afecta al no contar con el mismo, por ejemplo para Farley, Stearns y Hsu (1999, p.215) Los sistemas de Alta disponibilidad se deben entender como” sistemas que permiten a las aplicaciones seguir operando a pesar que el hardware o software falle. Estos sistemas deben proteger a los usuarios de fallos de software así como de fallas presentes en las unidades de procesamiento, disco o en los componentes de red”. Por lo que es razonable pensar que de ser posible eliminar los principales puntos o causas de falla en cualquiera de estos aspectos (software, hardware, red) mayores serán las probabilidades de contar con la disponibilidad de los sistemas críticos de la organización, más aun si se implantan componentes redundantes o tolerantes a fallos quienes junto a una alternativa de software puedan entrar en funcionamiento y asumir la responsabilidad en el caso que un componente falle.

Ahora bien, si bien es cierto que la redundancia asegura un alto grado de la disponibilidad de los sistemas, también es cierto que debido a los altos costos en los que se incurriría por la duplicación de cada una de las partes sería conveniente primeramente definir los requerimientos de disponibilidad, el tiempo tolerante a fallos ó los períodos de inoperabilidad aceptados.

Una vez identificadas realmente la criticidad de las aplicaciones y conocidos los niveles de tolerancia permitida, ya entonces se puede pensar en la corrección de puntos de fallo y en la redundancia de componentes. En este sentido, es necesario mencionar todos aquellos puntos de fallos en los cuales se debe centrar la atención

Falla de las Unidades de sistemas de Procesamiento (SPU): El SPU consiste en un grupo de elementos, cada uno de los cuales es propenso a fallar provocando la pérdida del servicio. Entre los más importantes de este grupo cabe la pena mencionar:

La unidad central de procesamiento, Los controladores de entrada y salida y la Tarjeta Principal.

Para eliminar los downtime asociados a la falla de estos componentes, se puede optar por el uso de una tecnología cluster, la cual permite reparar o reemplazar los mismos sin tener pérdida en el sistema. En lo que respecta particularmente a los fallos en la unidad central de procesamiento se podría quizás pensar en un esquema de multiprocesamiento simétrico,

Como otro punto de falla sobre el cual hay que centrar la atención es la falla en la red de área local (LAN). Las fallas en la red pueden ocurrir ya sea por el tipo de conexión utilizada, por las tarjetas de interfaz, por los routers, concentradores o incluso por las velocidades de transferencias utilizadas en la misma. Este tipo de problemas puede eliminarse bien sea proporcionando una redundancia total en las conexiones o configurando el switching de las tarjetas de interfaz.

Otro punto de fallo de interés radica en los subsistemas de almacenamiento, en este caso podría pensarse en la técnica de espejos, donde hay dos unidades independientes para cada disco de datos, siendo la unidad principal la encargada de manejar las peticiones de los usuarios y utilizándose la segunda unidad como un disco imagen de la primera, es decir, en este esquema los dos componentes realizan exactamente el mismo trabajo de forma entrelazada, teniéndose de esta manera dos sistemas equivalentes, de forma tal que si uno de ellos falla el otro continua trabajando.¹

Como otra alternativa podría ser la duplexación de discos o duplicación como también se le conoce, quien virtualmente opera de la misma manera que la técnica de espejo, excepto que añade un nuevo controlador, de allí que el rendimiento se incremente debido a la menor contención que ocurre en el bus de entrada y salida.

En lo que respecta a fallos por los subsistemas de almacenamiento, hoy día se cuenta con una serie de sistemas o metodologías de combinación de discos, conocidas como Arreglos de Discos Redundantes (RAID) que son más que un grupo de discos en los cuales se almacenan además de la información normal una serie de información redundante que permite, que en el caso que alguno de los discos falle, la información pueda regenerarse de forma automática". Este esquema puede ser implementado en diferentes formas dependiendo de la distribución que se haga en el disco, del tipo de rendimiento y del coste.²

El método **RAID nivel 0** fue el primero que apareció. En realidad este nivel RAID no incorpora tolerancia a fallos, por lo que un fallo en alguno de sus dispositivos provoca la pérdida de la información almacenada en todas las unidades. Este esquema a pesar de no ofrecer tolerancia es considerado, por

¹ Computer Associates. "Computer Associates anuncia una solución que confiere una alta disponibilidad a los servidores Windows NT para aplicaciones de misión crítica".[En línea]
<<http://www.cwordl.cl/archivos/ed179/reportaje.html>> [Citado en: 13 de agosto de 2017].

² Farley, M., &Stearns, T., & Hsu, J., Seguridad e Integridad de datos. España. Osborne/McGraw-Hill.

ser el primer método que apareció para la conjunción de varios discos en una única partición.

Un **RAID nivel 1** son discos en espejos, de esta forma se tiene duplicada la información en dos discos distintos, de manera que si uno de ellos falla, se continúa trabajando con el otro, sin por ello perder ni eficiencia en el servidor ni por supuesto la información. Este esquema consigue una fiabilidad en el sistema muy mejorada ya que aunque fallaran ambos discos simultáneamente (situación poco probable) se ahorraría tiempo de recuperación, adicionalmente las operaciones de lectura de un disco que forma parte de un espejo son más rápidas, ya que la lectura puede realizarse de forma simultánea en ambos discos. Sin embargo, en términos de almacenamiento solo se cuenta con el 50% de la capacidad total del disco, en operaciones de escritura se afecta el rendimiento ya que la información debe guardarse en dos sitios y en términos monetarios los costos de este esquema son sumamente elevados.

El RAID 5, método de tolerancia a fallos más utilizado en la actualidad, tiene la capacidad de acceder a tantos discos como necesite al mismo tiempo pero en peticiones independientes, por lo que su tasa de entrada y salida es mucho mayor si se compara con los esquemas predecesores. En este esquema la información se reparte a nivel de bloque e incluso de registro. Aquí desaparece la necesidad de un disco de paridad, sin embargo su implementación requiere el manejo de un mínimo de 3 discos. En este esquema uno de los discos conocido como Spare es usado como backup de todos los volúmenes raid que han sido configurados en el arreglo, de forma que si cualquiera de los volúmenes falla, el spare es utilizado para restablecer la información.

Entre las principales ventajas que este esquema ofrece se pueden mencionar: Mejor utilización del disco, buena velocidad de lectura y disminución significativa de los costos. Sin embargo y a pesar de ser un muy buen esquema para el manejo de tolerancia a fallos, este esquema tiene como punto en contra, el bajo rendimiento que presenta para las operaciones de escritura.

Adicional a todos los esquemas raid ya mencionados, también se cuenta con los arreglos de discos EMC o arreglos simétricos que permiten conexiones desde el mismo disco de datos a múltiples nodos del cluster a través de diferentes buses.

El RAID, originó una nueva técnica conocida como intercambio en caliente, que no es más que la capacidad de sacar o introducir discos, mientras este continúa funcionando y suministrando entradas y salidas al sistema al que está conectado. Por supuesto, esto requiere un diseño que permita manejar las sobrecargas de energía y las interrupciones que se producen en el bus interno de entrada y salida como resultado de las extracciones o inserciones realizadas. En este ámbito se cuenta con una técnica en particular conocida como Disk Storage Enclosure, quien permite realizar un intercambio en caliente de los discos espejos, sin perder el servicio, mientras el sistema operativo continúa corriendo y el dispositivo funcionando.

Como se ve existe una extensa variedad de alternativas de hardware para ofrecer alta disponibilidad de las aplicaciones, cada una de las cuales se adapta a un escenario en particular. Sin embargo como un esquema general, que actualmente asegura la alta disponibilidad, donde actualmente se enmarcan la mayoría de las soluciones y que quizás es el esquema más implementado es la tecnología clusters, definida por Weygant (2000 p.73) como un “grupo de servidores que poseen suficiente redundancia de software y hardware de forma que la falla de alguno de sus componentes no impida la disponibilidad del servicio “.Consiguiéndose así potenciar las principales características de las aplicaciones de misión crítica.

Esta tecnología puede ser implementada de tres (3) maneras básicamente.

- ✓ Configuración Activa/Standby:
- ✓ Configuración Activa/Activa
- ✓ Configuración de Base de Datos Paralela

Como última opción no puede dejar de mencionarse la tecnología de Red de Area de Almacenamiento (Storage Area Network o SAN) pues es allí donde se dirigen las tendencias del mercado y donde se están invirtiendo grandes esfuerzos.

SAN es una red dedicada en exclusiva a funciones de almacenamiento, la cual se construye sobre una red de fibra óptica o Fibre Channel a la que se conectan todos los servidores que utilizan el almacenamiento, ofreciendo un ancho de banda potente. SAN es una tecnología fiable, económica y que ha desarrollado una potencia hasta hace años desconocida. Básicamente presenta dos opciones de implantación: Bucle Arbitrado y Fabric Switched.

En ambos casos se pueden integrar soluciones que eviten utilizar el ancho de banda de la red de área local para la realización de Backups, estas soluciones se integran en la SAN a través de controladoras específicas permitiendo así un altísimo rendimiento así como una operación desatendida.

Entre las ventajas más importantes que puede ofrecer la red de área de almacenamiento están:

- ✓ Superior ancho de banda.
- ✓ Administración centralizada de todo el almacenamiento.
- ✓ Estandarización en recursos de almacenamiento y configuración de servidores.
- ✓ Mayor flexibilidad en la utilización del espacio disponible.
- ✓ Mayor tolerancia de fallas en el sistema.
- ✓ Al tratarse de dispositivos de ámbito corporativo, el rendimiento de los sistemas de almacenamiento puede llegar a ofrecer decenas de miles de I/O (operaciones de lectura/escritura) por segundo.
- ✓ Fácil integración de soluciones de backup de ámbito corporativo de simple administración, alto rendimiento y fiabilidad.

- ✓ Simplificación del proceso de adquisición de ampliaciones al sistema de almacenamiento, pues no estará basado en modelos ni marcas de servidores.
- ✓ Mejora de la facilidad de planificación de las necesidades globales.

Ahora bien, la alta disponibilidad puede conseguirse tanto con alternativas de hardware como de software, En el mercado tecnológico actual, son muchas las empresas que se han dedicado, al desarrollo de herramientas de software orientadas a la alta disponibilidad.

En primer punto se puede mencionar un esquema propuesto por la empresa HP, conocido como Journaled File System (JFS). Esta alternativa orientada al sistema operativo Unix, utiliza una conexión especial que le permite registrar información acerca de los cambios hechos en el file system de metadatos. Dicha conexión permite mejorar la disponibilidad reduciendo a solo unos segundos el tiempo necesario para reiniciar un file system luego que el mismo ha fallado.

Como una alternativa opcional para expandir las funciones de JFS se puede considerar OnlineJFS, quien elimina los downtime planificados asociados con las actividades de mantenimiento de los file system pues actividades como desfragmentación, reorganización y expansión de los file system pueden ser ejecutadas mientras las aplicaciones acceden la data de los mismos. En lo que respecta a las actividades de backup las mismas se ejecutan utilizando la filosofía snapshot, lo que permite que las aplicaciones accedan la data más reciente mientras que el proceso de backup accede la copia.

Otro esquema de software que igualmente apoya la disponibilidad de los sistemas es la herramienta Transaction Processing Monitors (TPM). Este esquema asegura la disponibilidad en segundos cuando es usado en conjunto con la tecnología clusters, pues se reenvían las transacciones a otro nodo cuando el nodo original presenta alguna falla.³

6.1.2 Historia Servicios Web

Los Servicios Web surgieron ante una necesidad de estandarizar la comunicación entre distintas plataformas (PC, Mainframe, Mac, etc.) y lenguajes de programación (PHP, C, Java, etc.). Anteriormente se habían realizado intentos de crear estándares pero fracasaron o no tuvieron el suficiente éxito, algunos de ellos son DCOM y CORBA, por ser dependientes de la implementación del vendedor DCOM – Microsoft, y CORBA – ORB (a pesar que CORBA de múltiples vendedores pueden operar entre si, hay ciertas limitaciones para aplicaciones de niveles más altos en los cuales se necesite seguridad o administración de transacciones).

³TELEMATIQUE, «TELEMATIQUE, VOLUMEN 2,» [En línea].
 <<http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/758/1817>.> [Citado en: 13 de agosto de 2017].

Otro gran problema es que se hacía uso de RPC (Remote Procedure Call) para realizar la comunicación entre diferentes nodos. Esto, además de presentar ciertos problemas de seguridad, tiene la desventaja de que su implementación en un ambiente como es Internet, es casi imposible (muchos Firewalls bloquean este tipo de mensajes, lo que hace prácticamente imposible a dos computadoras conectadas por Internet comunicarse). Los Servicios Web surgieron para finalmente poder lograr la tan esperada comunicación entre diferentes plataformas. En la actualidad muchos sistemas legacy están pasando a ser servicios web. Es por esto que en 1999 se comenzó a plantear un nuevo estándar, el cual terminaría utilizando XML, SOAP, WSDL, y UDDI.

A pesar de mucho limitar el uso de los servicios Web al Protocolo http, los servicios Web no fueron pensados para un protocolo en particular, es decir, nada impidió utilizar SOAP sobre algún otro protocolo de Internet (SMTP, FTP, etc.). Se utiliza principalmente HTTP por ser un protocolo ampliamente difundido y que se encuentra menos restringido por firewalls (generalmente se bloquean puertos como el FTP, pero el HTTP es muy probable que no esté bloqueado).

Primer Servidor Web

La década de los 80's fue marcada por el surgimiento de la PC y de la interfase gráfica. Entre 1988 y 1993, NeXT fabricó una estación de trabajo de altas prestaciones para la época de la que hablamos que impulsó con el Sistema operativo de la casa, el NeXTSTEP. Contaba con un micro de la serie (68040) de Motorola capaz de trabajar a 25 MHz, una memoria de 8 MB ampliables a 64 MB y un monitor de 17". La máquina, que costaba 6500 dólares, recibió el nombre de NeXT Computer, aunque se le acabó conociendo como NeXTcube o, simplemente, "The Cube". Más allá de su Hardware y sus posibilidades técnicas, la NeXT Computer ha pasado a formar parte de la pequeña gran historia de la informática por ser el ordenador que Tim Berners-Lee, el "inventor" de Internet, se utilizó por primera vez como un Servidor Web.

En la década de los 90's Internet permitió conectar computadoras en una escala global. En principio la conexión fue entre PCs y servidores por medio del explorador de Internet. A comienzos de este siglo es clara la necesidad de permitir a las computadoras conectadas a Internet comunicarse entre ellas. Desde entonces se va dando forma al nuevo modelo de computación distribuida llamado servicios Web basados en XML. El objetivo es permitir comunicarse entre sí a sistemas heterogéneos dentro y fuera de la empresa. Esta comunicación es independiente del Sistema Operativo, lenguaje o modelo de programación. Para conseguir esto se desarrollaron estándares. El consorcio de Internet <http://www.w3c.org> fue el encargado de crear y mantener estos estándares.

Desde los inicios de Internet, fueron surgiendo diferentes demandas por los usuarios y se dieron soluciones mediante lenguajes estáticos. A medida que paso el tiempo, las tecnologías fueron desarrollándose y surgieron nuevos

problemas a dar solución. Esto dio lugar a desarrollar lenguajes de programación para la Web dinámicos, que permitieran interactuar con los usuarios y utilizaran sistemas de Bases de Datos.

Antes de la adopción del modelo de Servicios Web basados en XML los datos eran “islas” que se encontraban dentro de las aplicaciones. Era muy difícil y costoso implementar soluciones para acceder a la información desde afuera de la aplicación. Las aplicaciones pueden ahora, comunicarse entre sí y con los sistemas de sus socios, proveedores y clientes gracias a los Servicios Web y XML.

6.2 Marco Teórico

6.2.1 Arquitectura de las Aplicaciones Web

Una aplicación Web es proporcionada por un servidor Web y utilizada por usuarios que se Conectan desde cualquier punto vía clientes Web (browsers o navegadores). La arquitectura de un Sitio Web tiene tres componentes principales:

- ✓ Un servidor Web
- ✓ Una conexión de red
- ✓ Uno o más clientes

El servidor Web distribuye páginas de información formateada a los clientes que las solicitan. Los requerimientos son hechos a través de una conexión de red, y para ello se usa el protocolo HTTP. Una vez que se solicita esta petición mediante el protocolo HTTP y la recibe el servidor Web, éste localiza la página Web en su sistema de archivos y la envía de vuelta al navegador que la solicitó.

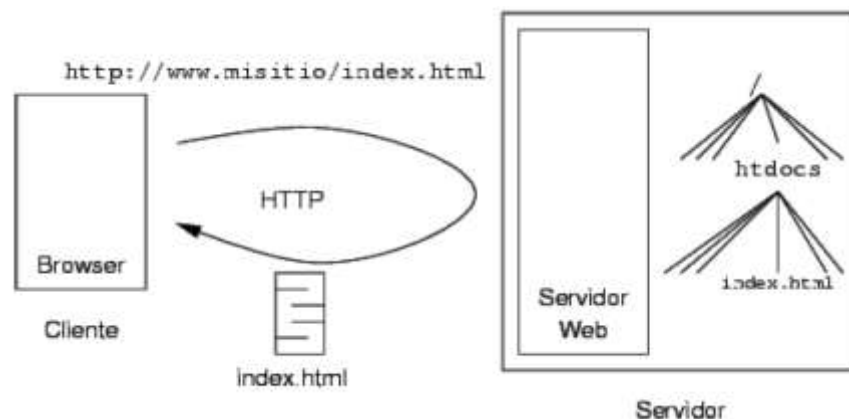


Ilustración 2: Fuente Autora: Arquitectura de las aplicaciones web.

Las aplicaciones Web están basadas en el modelo Cliente/Servidor que gestionan servidores web, y que utilizan como interfaz páginas web.

Las páginas Web son el componente principal de una aplicación o sitio Web. Los browsers piden páginas (almacenadas o creadas dinámicamente) con información a los servidores Web. En algunos ambientes de desarrollo de aplicaciones Web, las páginas contienen código HTML y scripts dinámicos, que son ejecutados por el servidor antes de entregar la página.

Una vez que se entrega una página, la conexión entre el browser y el servidor Web se rompe, es decir que la lógica del negocio en el servidor solamente se activa por la ejecución de los scripts de las páginas solicitadas por el browser (en el servidor, no en el cliente). Cuando el browser ejecuta un script en el cliente, éste no tiene acceso directo a los recursos del servidor. Hay otros componentes que no son scripts, como los applets (una aplicación especial que se ejecuta dentro de un navegador) o los componentes ActiveX. Los scripts del cliente son por lo general código JavaScript o VBScript, mezclados con código HTML.

La colección de páginas son en una buena parte dinámicas (ASP, PHP, etc.), y están agrupadas lógicamente para dar un servicio al usuario. El acceso a las páginas está agrupado también en el tiempo (sesión). Los componentes de una aplicación Web son:

1. Lógica de negocio

- ✓ Parte más importante de la aplicación.
- ✓ Define los procesos que involucran a la aplicación.
- ✓ Conjunto de operaciones requeridas para proveer el servicio.

2. Administración de los datos

- ✓ Manipulación de BD y archivos.

3. Interfaz

- ✓ Los usuarios acceden a través de navegadores, móviles, PDAs, etc.
- ✓ Funcionalidad accesible a través del navegador.
- ✓ Limitada y dirigida por la aplicación.

Las aplicaciones web se modelan mediante lo que se conoce como modelo de capas, Una capa representa un elemento que procesa o trata información. Los tipos son:

- ✓ **Modelo de dos capas:** La información atraviesa dos capas entre la interfaz y la administración de los datos.
- ✓ **Modelo de n-capas:** La información atraviesa varias capas, el más habitual es el modelo de tres capas.

Modelo de dos Capas

Gran parte de la aplicación corre en el lado del cliente (fat client).

Las capas son:

- ✓ **Cliente (fat client):** La lógica de negocio está inmersa dentro de la aplicación que realiza el interfaz de usuario, en el lado del cliente.
- ✓ **Servidor:** Administra los datos.

Modelo de tres Capas

Está diseñada para superar las limitaciones de las arquitecturas ajustadas al modelo de dos capas, introduce una capa intermedia (la capa de proceso) Entre presentación y los datos, los procesos pueden ser manejados de forma separada a la interfaz de usuario y a los datos, esta capa intermedia centraliza la lógica de negocio, haciendo la administración más sencilla, los datos se pueden integrar de múltiples fuentes, las aplicaciones web actuales se ajustan a este modelo.⁴

Las capas de este modelo son:

- ✓ **Capa de presentación (parte en el cliente y parte en el servidor)**
 - Recoge la información del usuario y la envía al servidor (cliente)
 - Manda información a la capa de proceso para su procesamiento
 - Recibe los resultados de la capa de proceso
 - Generan la presentación
 - Visualizan la presentación al usuario (cliente)
- ✓ **Capa de proceso (servidor web)**
 - Recibe la entrada de datos de la capa de presentación
 - Interactúa con la capa de datos para realizar operaciones
 - Manda los resultados procesados a la capa de presentación
- ✓ **Capa de datos (servidor de datos)**
 - Almacena los datos
 - Recupera datos
 - Mantiene los datos
 - Segura la integridad de los datos

⁴ I. T. d. Matehuala, «PROGRAMACION WEB,» 05 06 2015. [En línea]. <<https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>> [Citado en: 11 de agosto de 2017].

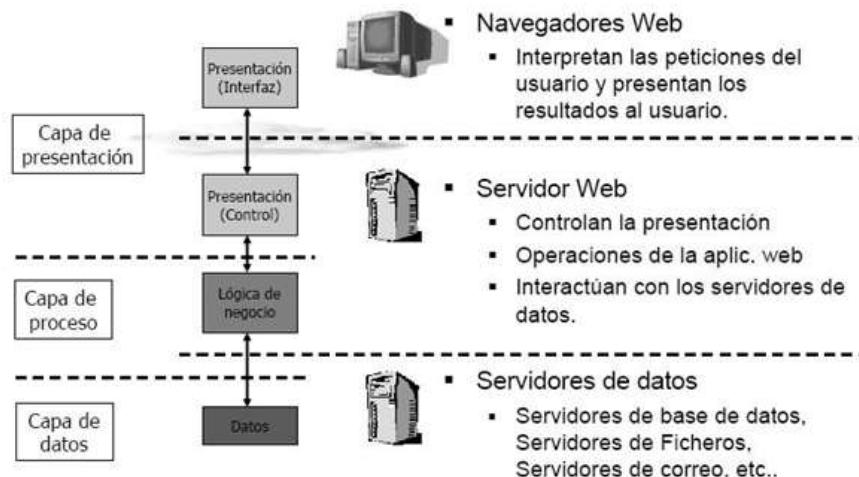


Ilustración 3: Fuente Autora: Capa de datos (servidor de datos)

6.2.2 Protocolos de Aplicación más Utilizados

Un protocolo es un método estándar que permite la comunicación entre procesos (que potencialmente se ejecutan en diferentes equipos), es decir, es un conjunto de reglas y procedimientos que deben respetarse para el envío y la recepción de datos a través de una red. Existen diversos protocolos de acuerdo a cómo se espera que sea la comunicación. Algunos protocolos, por ejemplo, se especializarán en el intercambio de archivos, como el FTP (File Transfer Protocol, Protocolo de transferencia de ficheros); otros pueden utilizarse simplemente para administrar el estado de la transmisión y los errores (como es el caso de ICMP), etc.

En Internet, los protocolos utilizados pertenecen a una sucesión de protocolos o a un conjunto de protocolos relacionados entre sí. Este conjunto de protocolos se denomina TCP/IP. Entre otros, contiene los siguientes protocolos:

El protocolo HTTP

Desde 1990, el protocolo HTTP (Hiper Text Transfer Protocol, Protocolo de transferencia de hipertexto) es el protocolo más utilizado en Internet. La versión 0.9 solo tenía la finalidad de transferir los datos a través de Internet (en particular páginas web escritas en HTML). La versión 1.0 del protocolo (la más utilizada) permite la transferencia de mensajes con encabezados que describen el contenido de los mensajes mediante la codificación MIME.

El propósito del protocolo HTTP es permitir la transferencia de archivos (principalmente, en formato HTML) entre un navegador (el cliente) y un servidor web localizado mediante una cadena de caracteres denominada dirección URL (Uniform Resource Locator, localizador uniforme de recursos).

La comunicación entre el navegador y el servidor se lleva a cabo en dos etapas:

- ✓ El navegador realiza una solicitud HTTP.
- ✓ El servidor procesa la solicitud y después envía una respuesta HTTP.

El Protocolo HTTPS

El protocolo seguro de Transferencia de hipertexto (HTTPS, Hiper Text Transfer Protocol Secure) es la versión segura del protocolo HTTP. La diferencia es que HTTPS permite realizar transacciones de forma segura. Por lo tanto, podremos desarrollar actividades de tipo e-commerce, acceso a cuentas bancarias on line, tramites con la administración pública, etc.

En los navegadores comunes como Firefox, Explorer o Chrome, cuando estamos empleando un protocolo HTTPS podemos ver el icono de un candado que aparece en la barra principal de nuestro navegador. Además, en la barra de direcciones podremos ver que “http://” será sustituido por “https://”. Y, ¿cómo funciona la conexión exactamente? ¿Por qué es más segura? Básicamente, lo que ocurre es que la página web codifica la sesión con certificado digital. De este modo, el usuario tiene ciertas garantías de que la información que envíe desde dicha página no podrá ser interceptada y utilizada por terceros.

Estos certificados de seguridad son conocidos como SSL. Cuando estos están instalados en la página web veremos el candado del que hablábamos anteriormente. Por otro lado, si hay instalados Certificados de Validación Extendida, además del candado, los usuarios podremos ver que la barra de URL del navegador toma un fondo verdoso.

El Protocolo FTP

El protocolo FTP (File Transfer Protocol, Protocolo de transferencia de archivos) es, como su propio nombre indica, un protocolo para transferir archivos.

La implementación del FTP se remonta a 1971, cuando se desarrolló un sistema de transferencia de archivos (descrito en RFC141) entre equipos del Instituto Tecnológico de Massachusetts (MIT, Massachusetts Institute of Technology). Desde entonces, diversos documentos de RFC han mejorado el protocolo básico, pero las innovaciones más importantes se llevaron a cabo en julio de 1973.

El protocolo FTP define la manera en que los datos deben ser transferidos a través de una red TCP/IP. El objetivo del protocolo FTP es:

- ✓ Permitir que equipos remotos puedan compartir archivos.
- ✓ Permitir la independencia entre los sistemas de archivo del equipo del cliente y del equipo del servidor.

- ✓ Permitir una transferencia de datos eficaz.

El Protocolo SMTP

El protocolo SMTP (Simple Mail Transfer Protocol, Protocolo simple de transferencia de correo) es el protocolo estándar que permite la transferencia de correo de un servidor a otro mediante una conexión punto a punto.

SMTP se basa en el modelo cliente-servidor, donde un cliente envía un mensaje a uno o varios receptores. La comunicación entre el cliente y el servidor consiste enteramente en líneas de texto compuestas por caracteres ASCII.

El tamaño máximo permitido para estas líneas es de 1.000 caracteres.⁵

6.2.3 Sistema de Alta Disponibilidad

La disponibilidad es una de las características de las arquitecturas empresariales que mide el grado con el que los recursos del sistema están disponibles para su uso por el usuario final a lo largo de un tiempo dado. Ésta no sólo se relaciona con la prevención de caídas del sistema (también llamadas tiempos fuera de línea, downtime u offline), sino incluso con la percepción de “caída” desde el punto de vista del usuario: cualquier circunstancia que nos impida trabajar productivamente con el sistema – desde tiempos de respuesta prolongados, escasa asistencia técnica o falta de estaciones de trabajo disponibles – es considerada como un factor de baja disponibilidad.

Las bases de datos y el auge de Internet han permitido la colaboración y el intercambio de información desde cualquier parte del mundo, ampliando el alcance de las aplicaciones de bases de datos en todas las organizaciones y comunidades.

Este alcance nos hace resaltar la importancia de la alta disponibilidad (HA, del inglés High Availability) en soluciones de gestión de datos. Tanto las pequeñas empresas y las empresas mundiales tienen usuarios de todo el mundo que necesitan acceso a los datos las 24 horas del día. Sin este acceso a datos, las operaciones pueden detenerse, y perder ciertos ingresos. Los usuarios se han vuelto más dependientes de sus soluciones, que ahora exigen acuerdos de nivel de servicio de sus departamentos de Tecnología de la Información (IT, del inglés Information Technology) y proveedores.

La disponibilidad es el grado en que una aplicación o servicio está disponible cuándo y cómo los usuarios esperan. La disponibilidad se mide por la percepción de una aplicación del usuario final. Los usuarios finales

⁵ RA-M, «Conceptos generales de la arquitectura de aplicaciones web,» de Implantación de aplicaciones web, p. 11.

experimentan frustración cuando sus datos no están disponibles, y ellos no entienden o son capaces de diferenciar los complejos componentes de una solución global. Fiabilidad, valorización, continuas operaciones y detección de errores son características de una solución de alta disponibilidad.

Fiabilidad: Los componentes hardware fiables de una solución de HA, el software fiable, incluida la base de datos, servidores web y aplicaciones, es la parte crítica de una implementación de una solución de alta disponibilidad.

Recuperación: Puede haber muchas opciones para recuperarse de un fracaso si ocurre alguno. Es importante determinar qué tipo de fallos pueden ocurrir en su entorno de alta disponibilidad y la forma de recuperarse de estos fallos en el tiempo que satisface las necesidades comerciales.

Detección de errores: Si un componente en su arquitectura falla, entonces la rápida detección, de dicho componente es esencial en la recuperación de un posible fracaso inesperado. Si bien es posible que pueda recuperarse rápidamente de un corte de luz, si se lleva a otros 90 minutos para descubrir el problema, entonces usted no puede satisfacer su SLA. La monitorización del estado del entorno de trabajo requiere un software fiable, para ver de forma rápida y notificar al administrador de bases de datos (DBA) un problema.⁶

6.2.4 PHP

PHP (acrónimo recursivo de PHP: Hypertext Preprocessor) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

En lugar de usar muchos comandos para mostrar HTML (como en C o en Perl), las páginas de PHP contienen HTML con código incrustado que hace "algo" (en este caso, mostrar "¡Hola, soy un script de PHP!"). El código de PHP está encerrado entre las etiquetas especiales de comienzo y final `<?php` y `?>` que permiten entrar y salir del "modo PHP".

Lo que distingue a PHP de algo del lado del cliente como Javascript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente. El cliente recibirá el resultado de ejecutar el script, aunque no se sabrá el código subyacente que era. El servidor web puede ser configurado incluso para que procese todos los ficheros HTML con PHP, por lo que no hay manera de que los usuarios puedan saber qué se tiene debajo de la manga.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores

⁶ S. Mehdi and N. E. Zarour, "Composition of web services using multi agent based planning with high availability of web services," 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, 2016, pp. 10-15.

profesionales. No sienta miedo de leer la larga lista de características de PHP. En unas pocas horas podrá empezar a escribir sus primeros scripts.

Aunque el desarrollo de PHP está centrado en la programación de scripts del lado del servidor, se puede utilizar para muchas otras cosas.

PHP está enfocado principalmente a la programación de scripts del lado del servidor, por lo que se puede hacer cualquier cosa que pueda hacer otro programa CGI, como recopilar datos de formularios, generar páginas con contenidos dinámicos, o enviar y recibir cookies. Aunque PHP puede hacer mucho más.

Existen principalmente tres campos principales donde se usan scripts de PHP.

- ✓ **Scripts del lado del servidor.** Este es el campo más tradicional y el foco principal. Son necesarias tres cosas para que esto funcione. El analizador de PHP (módulo CGI o servidor), un servidor web y un navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del programa de PHP con un navegador, viendo la página de PHP a través del servidor. Todo esto se puede ejecutar en su máquina si está experimentado con la programación de PHP.
- ✓ **Scripts desde la línea de comandos.** Se puede crear un script de PHP y ejecutarlo sin necesidad de un servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo de esta manera. Este tipo de uso es ideal para scripts que se ejecuten con regularidad empleando cron (en *nix o Linux) o el Planificador de tareas (en Windows). Estos scripts también pueden usarse para tareas simples de procesamiento de texto.
- ✓ **Escribir aplicaciones de escritorio.** Probablemente PHP no sea el lenguaje más apropiado para crear aplicaciones de escritorio con una interfaz gráfica de usuario, pero si se conoce bien PHP, y se quisiera utilizar algunas características avanzadas de PHP en aplicaciones del lado del cliente, se puede utilizar PHP-GTK para escribir dichos programas. También es posible de esta manera escribir aplicaciones independientes de una plataforma. PHP-GTK es una extensión de PHP, no disponible en la distribución principal.⁷

6.2.5 Laravel

Es un framework joven con un gran futuro. Ofrece las funcionalidades necesarias para desarrollar aplicaciones modernas de manera fácil y segura. En el 2011 su creador, Taylor Otwell, lo mostró por primera vez al mundo. Por lo tanto, se trata de un framework con un enfoque fresco y moderno; ya que es bastante joven. Está hecho para arquitectura MVC y resuelve necesidades

⁷ PHP, «PHP,» [En línea]. <<http://php.net/manual/es/intro-whatcando.php>> [Citado en: 11 de agosto de 2017].

actuales como manejo de eventos y autenticación de usuarios. Además, cuenta con un código modular y extensible por medio de un administrador de paquetes y un soporte robusto para manejo de bases de datos. No importa si eres un experto en PHP o si son tus primeros pasos; cuando lo conozcas, sabrás que Laravel es el framework que estabas buscando para tus proyectos PHP.

Laravel es modular y extensible. Esto quiere decir que te permite agregar todo lo que necesitas a través de su directorio Packalyst que cuenta con más de 5,500 paquetes. Esto con el objetivo de que siempre encuentres lo que necesitas.

Cuenta con un sistema de enrutamiento rápido y eficiente, similar al que se usa en Ruby on Rails. Este nos permite relacionar las partes de nuestra aplicación con las rutas que ingresa el usuario en el navegador. Las aplicaciones estarán blindadas usando Middleware; ya que se encarga de analizar y filtrar las llamadas HTTP en tu servidor. Puedes instalarlo para que se encargue de verificar que se trate de un usuario registrado, de evitar problemas de tipo Cross-Site-Scripting (XSS) y otras medidas de seguridad. La seguridad es muy importante, Laravel viene listo para implementar autenticación de usuarios de forma nativa e incluye la opción de “recordar” al usuario. Además, te permite incluir parámetros adicionales, lo que nos asegurará, por ejemplo, si se trata de un usuario activo. Una aplicación segura necesita ser capaz de encriptar sus datos. Con Laravel tienes todo lo necesario para empezar a usar seguridad OpenSSL y cifrado AES-256-CBC. Adicionalmente, todos los valores encriptados están firmados por un código de autenticación de mensaje que detecta si el mensaje encriptado fue alterado.⁸

6.2.6 Servidor Web

La principal función de un servidor Web es almacenar los archivos de un sitio y emitirlos por Internet para poder ser visitado por los usuarios. Básicamente, un servidor Web es una gran computadora que guarda y transmite datos vía Internet. Cuando un usuario entra en una página de Internet su navegador se comunica con el servidor enviando y recibiendo datos que determinan qué es lo que ve en la pantalla. Por eso decimos que los servidores Web están para almacenar y transmitir datos de un sitio según lo que pida el navegador de un visitante.

Gestiona cualquier aplicación en el lado del servidor realizando conexiones bidireccionales y/o unidireccionales y síncronas o asíncronas con el cliente generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un Navegador Web. Para la transmisión de todos estos datos se utiliza algún

⁸ REICEK, «Platzi,» 2015. [En línea]. <<https://platzi.com/blog/laravel-framework-php/>> [Citado en: 11 de agosto de 2017].

protocolo. Generalmente se utiliza el protocolo HTTP para estas comunicaciones, perteneciente a la capa de aplicación del Modelo OSI. El término también se emplea para referirse al ordenador que ejecuta el programa.

6.2.7 Servicios

- ✓ **Internet:** Internet es una Red de Ordenadores conectados en toda la extensión del Globo Terráqueo, que ofrece diversos servicios a sus usuarios como pueden ser el Correo electrónico, el Chat o la Web. Todos los servicios que ofrece Internet son llevados a cabo por miles de ordenadores que están permanentemente encendidos y conectados a Internet, esperando que los usuarios les soliciten los servicios y sirviéndolos una vez son solicitados.
- ✓ **Página Web:** Documento o fuente de información, generalmente en formato HTML y que puede contener hiperenlaces a otras Páginas Web. Dicha página, podrá ser accesible desde un dispositivo físico, una intranet, o Internet.
- ✓ **Navegador Web:** Para establecer conexiones con los servidores Web, y obtener la información y los servicios que estos prestan, el usuario necesita tener instalado en su equipo un programa cliente capaz de comunicarse con ellos. Estos programas son los llamados Navegadores Web. Los Navegadores Web, también llamados Visores de Web o Browsers, son aplicaciones que permiten ver en pantalla texto con formato (con palabras en negrita, y con distintas fuentes tipográficas, tamaños y colores) y presentar imágenes en línea. También permiten visualizar secuencias de vídeo y escuchar ficheros de sonido.
- ✓ **Servidor:** Un Servidor es un tipo de Software que suministra servicios a los usuarios o terminales que lo solicitan. Por ejemplo, en una típica Arquitectura Cliente-servidor, el cliente podría ser un ordenador que realiza peticiones de información a través de un programa de correo (Outlook Express por ejemplo) y, el servidor le entrega los datos en forma de correos electrónicos en respuesta a su solicitud. Un servidor no es necesariamente una máquina de última generación de grandes proporciones, no es necesariamente un superordenador; un servidor puede ser desde una computadora vieja, hasta una máquina sumamente potente (ej. Servidores web, bases de datos grandes, etc., procesadores especiales y hasta varios gigabytes de memoria).

Todo esto depende del uso que se le dé al servidor. Si usted lo desea, puede convertir al equipo desde el cual se está trabajando en un servidor instalando un programa que trabaje por la Red y a la que los usuarios de su red ingresen a través de un programa de servidor web como Apache. Por lo cual podemos

llegar a la conclusión de que un servidor también puede ser un proceso que entrega información o sirve a otro proceso.

El modelo Cliente-Servidor no necesariamente implica tener dos ordenadores, ya que un proceso cliente puede solicitar algo como una impresión a un proceso servidor en un mismo ordenador. Cliente/servidor: Esta Arquitectura consiste básicamente en un programa cliente que realiza peticiones a otro programa el servidor que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un Sistema Operativo multiusuario distribuido a través de una red de computadoras.

En esta Arquitectura la capacidad de proceso está repartida entre los Clientes y los Servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema. La separación entre Cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa.

Los tipos específicos de servidores que incluyen son los Servidores Web, los Servidores de archivo, los Servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma. Una disposición muy común son los Sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema.

6.2.8 Servidores más Usados

Apache

Está diseñado para ser un Servidor Web potente y flexible que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diferentes características o funcionalidades. Apache se ha adaptado siempre a una gran variedad de entornos a través de su diseño modular.

Este diseño permite a los administradores de Sitios Web elegir qué características van a ser incluidas en el servidor seleccionando que módulos se van a cargar, ya sea al compilar o al ejecutar el servidor. Este es el más común y más utilizado en todo el mundo.

Además, es gratuito, y de Código abierto, así que podríamos decir que corre sobre cualquier plataforma. Apache es una muestra, al igual que el Sistema Operativo Linux (un Unix desarrollado inicialmente para PC), de que el trabajo voluntario y cooperativo dentro de Internet es capaz de producir aplicaciones de calidad profesional difíciles de igualar.

Nginx

Nginx (se pronuncia “engin x”) es un Servidor http y proxy inverso gratuito, de Código abierto y de alto rendimiento, además de ser Servidor proxy para IMAP y POP3. Este servidor está actualmente manejando entre el 1% y el 4% de todos los dominios globales. A pesar de ser todavía una versión “beta”, Nginx es conocido por su estabilidad, su gran conjunto de características, una configuración sencilla y por consumir pocos recursos. Como este servidor no provee actualmente de un adaptador directo para Merb, usted requerirá utilizar un proxy reverso en Nginx con el fin de direccionar peticiones hacia uno o varios procesos Merb distintos. Esto puede ejecutarse con cualquier Servidor de aplicaciones basado en Rack, como Mongrel, Thin, Ebb o Glassfish.

Lighttpd

Es un servidor Web para los Sistemas operativos Unix/Linux y Microsoft Windows. Este servidor también conocido como Lighty, es una alternativa para el Servidor de páginas Web Apache. Está diseñado para ser seguro, rápido (muy rápido a decir verdad), compatible con los estandars y flexible a la vez que esta optimizado para entornos en los cuales la velocidad es crítica.

Su huella de memoria es muy pequeña (en comparacion a otros servidores Web), una ligera carga en el CPU y su enfoque en velocidad hacen de lighttpd perfecto para servidores con demasiada carga. Este servidor Web es otro de los más ligeros que hay en el mercado. Está especialmente pensado para hacer cargas pesadas sin perder balance, utilizando poca RAM y poca de CPU. Algunas páginas populares que lo usan son Youtube, Wikipedia y otras que soportan gran tráfico diariamente. También es gratuito y se distribuye bajo Licencia BSD.⁹

6.2.9 Aplicaciones en el Lado del Servidor

Una aplicación en el lado del servidor es cualquier programa o conjunto de instrucciones diseñadas con la finalidad de que un servidor Web las procese para realizar alguna acción. Las aplicaciones del lado del servidor están escritas mediante un lenguaje de programación, entre los que más se utilizan están los siguientes:

- ✓ PHP
- ✓ ASP
- ✓ Perl
- ✓ Python
- ✓ Ruby

⁹ EcuRed, «ecured,» [En línea]. <https://www.ecured.cu/Servidor_Web> [Citado en: 13 de agosto de 2017].

6.2.10 Ambiente Virtual de Aprendizaje

Según la UNESCO “Los entornos de aprendizaje virtuales constituyen una forma totalmente nueva de Tecnología Educativa y ofrece una compleja serie de oportunidades y tareas a las instituciones de enseñanza de todo el mundo, el entorno de aprendizaje virtual lo define como un programa informático interactivo de carácter pedagógico que posee una capacidad de esta comunicación integrada, es decir, que está asociado a nuevas tecnologías”.

Para lograr un apropiado ambiente virtual de aprendizaje, internamente existen varios roles que son necesarios para desarrollar este tipo de proyectos entre los que encontramos:

- ✓ **Director del Proyecto:** Encargado de orientar al resto del grupo para un apropiado desarrollo del ambiente virtual. Además tiene la responsabilidad en todas las fases del proyecto.
- ✓ **Diseñador de Instrucción:** Encargado de determinar las necesidades educativas, escoger la mejor forma de análisis pedagógico, diseñar los módulos de aprendizaje, y el desarrollo inicial de los esquemas del sistema de educación web.
- ✓ **Administrador del Sistema:** Encargados de proveer el mantenimiento al sistema. . A medida que el diseñador de instrucción crea lecciones y esquemas de diseño, el administrador del sistema los revisa para asegurar que la red y el software existente sean compatibles con el diseño.
- ✓ **Administradores del proceso de aprendizaje:** Encargados de llenar encuestas y responder cuestionarios, participan en entrevistas y en sesiones de observación. Cuando está completo el piloto, los aprendices y sus administradores se encargan de revisar el programa y proveer retroalimentación.
- ✓ **Editores:** Estas personas se responsabilizan de la gramática, consistencia y claridad de los textos usados en las páginas del sitio. Hacen una revisión del esquema antes de entregarlo a los programadores y artistas gráficos, reducen la cantidad de trabajo de corrección durante el desarrollo del sitio.
- ✓ **Programadores:** Poseen un rol activo durante las últimas etapas del desarrollo. A medida que se construye el esquema de aprendizaje revisan el diseño y hacen recomendaciones técnicas: por ejemplo, si el diseñador de instrucción desea crear un ejercicio que hace seguimiento a las respuestas del aprendiz y provee retroalimentación dinámica, el programador puede entregar recomendaciones acerca de la manera en la cual se podría llevar a cabo.
- ✓ **Artistas Gráficos:** Encargados de traducir los diseños de las lecciones en páginas de web. Los artistas gráficos proveen dirección y estilo creativo: las imágenes, navegación y layout (diseño) delineados en la etapa de diseño se crean durante la fase de desarrollo. Los artistas trabajan en conjunto con

los programadores para crear imágenes de formatos utilizables en páginas web como son los JPEGy los GIF.

- ✓ **Webmaster:** Encargado del mantenimiento del servidor de web y el sitio de la unidad de aprendizaje. Durante la etapa de diseño, el Webmaster hace un estimado de los requerimientos de capacidad de servidor y de espacio de almacenamiento para el soporte al módulo de educación en línea.¹⁰

Existen diferentes entornos que debe garantizar los ambientes virtuales de aprendizaje, para nuestro proyecto son fundamentales y que logran el progreso y el adecuado uso de la plataforma que deseamos plantear:

- ✓ **Entorno de Conocimiento:** Mediante los contenidos digitales se invita al estudiante a buscar y manipular la información en forma atractiva, interactiva y creativa.
- ✓ **Entorno de Colaboración:** Interacción entre alumnos y profesor para la retroalimentación de los conocimientos.
- ✓ **Entorno de Asesoría:** Espacio para la solución de dudas y retroalimentación
- ✓ **Entorno de Experimentación:** Espacio de complemento al contenido programado en el entorno.
- ✓ **Entorno de Gestión:** Espacio para trámites de estudiantes o profesores.¹¹

6.3 Marco Tecnológico

El objetivo de implementar un sistema de alta disponibilidad, utilizan herramientas como los frameworks, es hacer que nos centremos en el verdadero problema, y no preocuparnos por implementar funcionalidades que son de uso común en muchas aplicaciones, como podría ser el proceso de login de usuarios o establecer la conexión con la base de datos.

Los frameworks nos guían en el proceso de desarrollar nuestra aplicación siguiendo el patrón de diseño MVC. La capa modelo es la que se encarga de trabajar con los datos, habitualmente es la que se encarga de almacenar los datos. Por su parte, la vista es la capa de presentación, cómo presentamos los datos al usuario final, al cliente de nuestra aplicación. Y, por último, el controlador es el mediador entre el modelo y la vista. Normalmente la vista requiere de datos del modelo, pero la vista no tiene comunicación directa con el modelo. Por tanto, la vista pide esos datos al controlador y éste a su vez al modelo, el modelo maneja los datos en la forma que defina nuestra lógica de la aplicación y devuelve la respuesta de nuevo al controlador, quién, por último, entrega esta respuesta a la vista para poder ser renderizada.

¹⁰ P. Mendoza B-A. Galvis P AMBIENTES VIRTUALES DE APRENDIZAJE: UNA METODOLOGÍA PARA SU CREACIÓN http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-106223_archivo.pdf

¹¹ Y. Hui and H. Haiyan, "A Study on Data Storage System Based on High-Availability Open Virtual Experiment Platform," 2016 International Conference on Robots & Intelligent System (ICRIS), Zhangjiajie, 2016, pp. 55-58.

7. IMPACTOS

Durante la ejecución del sistema, se realizara un monitoreo constante en caso de que alguna de las variables definidas en los procesos establecidos se cambie o modifique. Es posible que en el desarrollo del proyecto surjan nuevos riesgos u oportunidades que no estaban presentes al principio del mismo, y otros pueden tienden a desaparecer por su poca probabilidad de ocurrencia por lo que es necesaria una continua labor de gestión periódica, haciendo de esta actividad un ciclo que soporta los demás procesos.

La implementación de este sistema de alta disponibilidad para el manejo de documentos por medio de un aplicativo web genera un impacto positivo, ya que permite continuar con la investigación en el desarrollo de aplicaciones de este tipo para el grupo de investigación METIS de la Universidad Distrital Francisco José de Caldas – Facultad Tecnológica; puesto que al implementar un entorno altamente disponible garantiza a toda organización cumplir con sus funciones y ser competitivos. Este tipo de desarrollos promueven la creación de diversos software con la misma arquitectura de alta disponibilidad implementada en este proyecto.

8. ENFOQUE DE LA ARQUITECTURA – DISEÑO METODOLÓGICO

Existen enfoques Top-Down, Bottom-Up, e híbridos que han sido utilizados en la ingeniería del software para validar arquitecturas. La literatura sobre estos enfoques reconoce ventajas y desventajas de éstos a nivel general en el caso de la arquitectura planteada se trabaja un enfoque Top Down, el cual parte de lo general a lo particular, tal como muestra la siguiente gráfica:

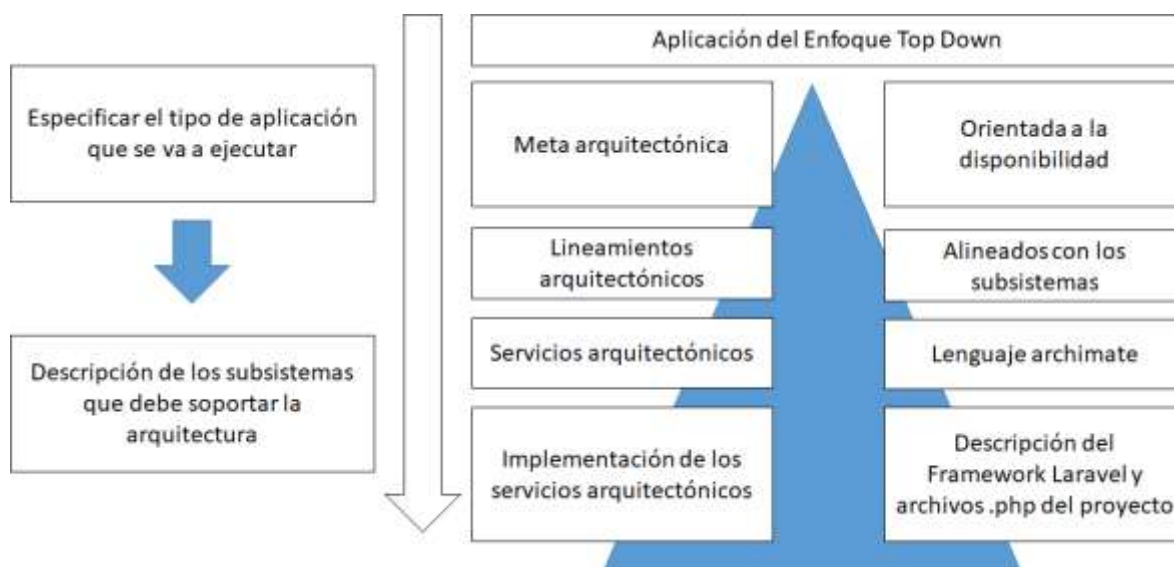


Ilustración 4: Fuente Autora: Enfoque de la arquitectura, diseño lógico.

En donde se aprecia que partiendo del tipo de aplicación a realizar y de los subsistemas que debe tener esta aplicación, se genera una meta arquitectónica de disponibilidad que es refinada mediante unos lineamientos arquitectónicos que se encuentran alineados con cada uno de los subsistemas a desarrollar.

Los lineamientos arquitectónicos se especifican mediante unos servicios arquitectónicos, los cuales son mostrados en el lenguaje notacional “Archimate”.

Finalmente la implementación de estos servicios es mostrada a la luz de la funcionalidad que tienen los diferentes archivos tipo php que fueron desarrollados para comprobar la arquitectura.

9. OBJETIVOS ARQUITECTÓNICOS PARA LOGRAR ALTA DISPONIBILIDAD

9.1 Meta arquitectónica

Toda arquitectura debe tener clara una meta arquitectónica, dependiendo esta meta arquitectónica se detallan los elementos necesarios para la construcción de la arquitectura. En el caso del sistema a desarrollar se tiene una meta arquitectónica, seguida de unos lineamientos arquitectónicos, tal como se detalla a continuación:

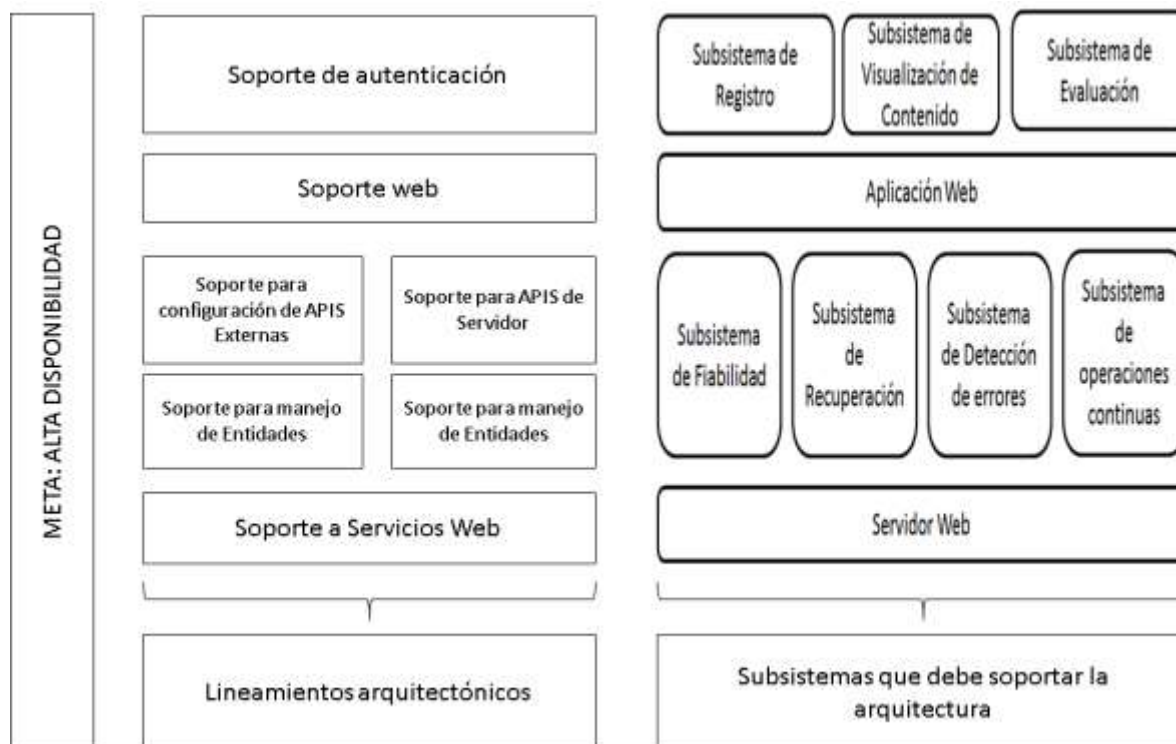


Ilustración 5: Fuente Autora: Meta arquitectónica

En la figura anterior se puede apreciar los lineamientos que deben regir la construcción de la arquitectura. De otra parte en la figura anterior también se aprecia cómo dichos lineamientos se producen a partir de unas necesidades concretas que debe implementar un sistema a través de cada uno de sus subsistemas.

9.2 Lineamientos arquitectónicos

En la sección anterior teníamos unos lineamientos que salían a partir de unas necesidades concretas expuestas por el tipo de sistema que se desea construir. En así como en esta sección se detallan cada uno de estos lineamientos y se muestran los elementos de bajo nivel que permiten su implementación.

En la siguiente gráfica se describen los servicios que permitirían implementar cada uno de los lineamientos arquitectónicos.

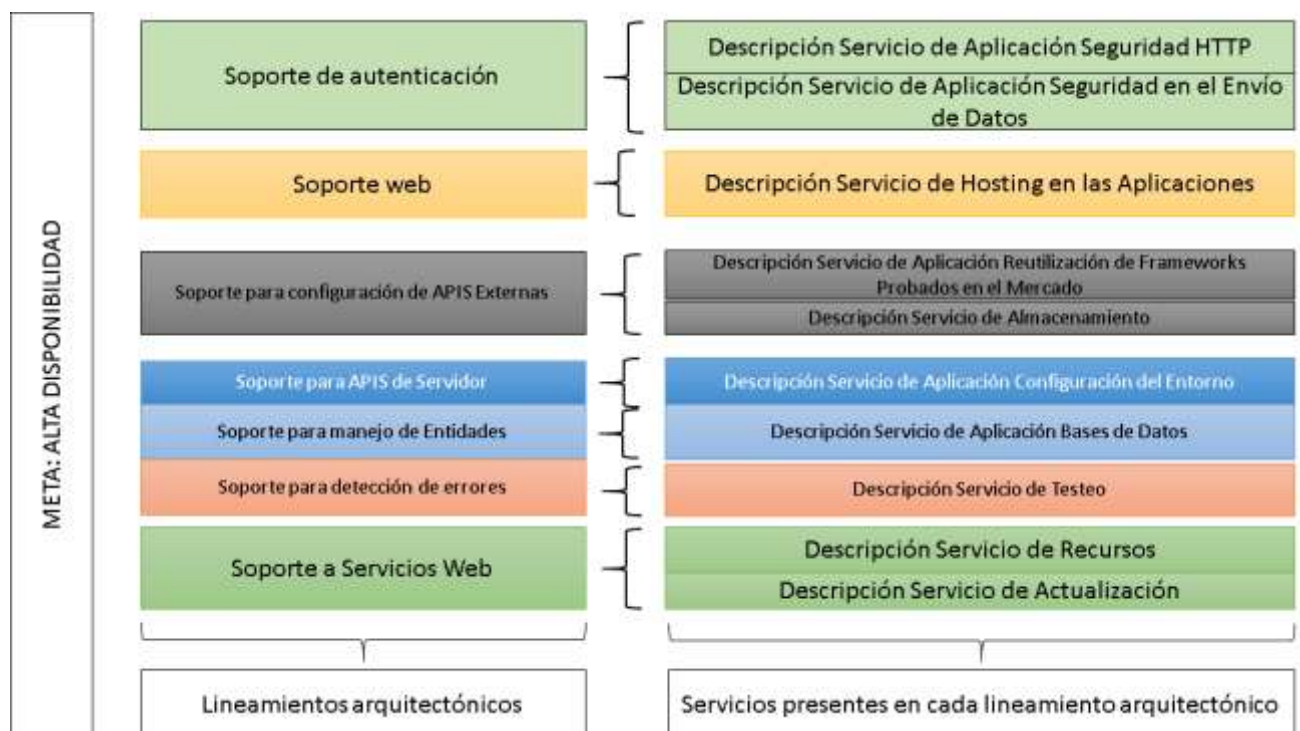


Ilustración 6: Fuente Autora: Lineamientos arquitectónicos.

10. SERVICIOS ARQUITECTONICOS

Vista Introductoria General

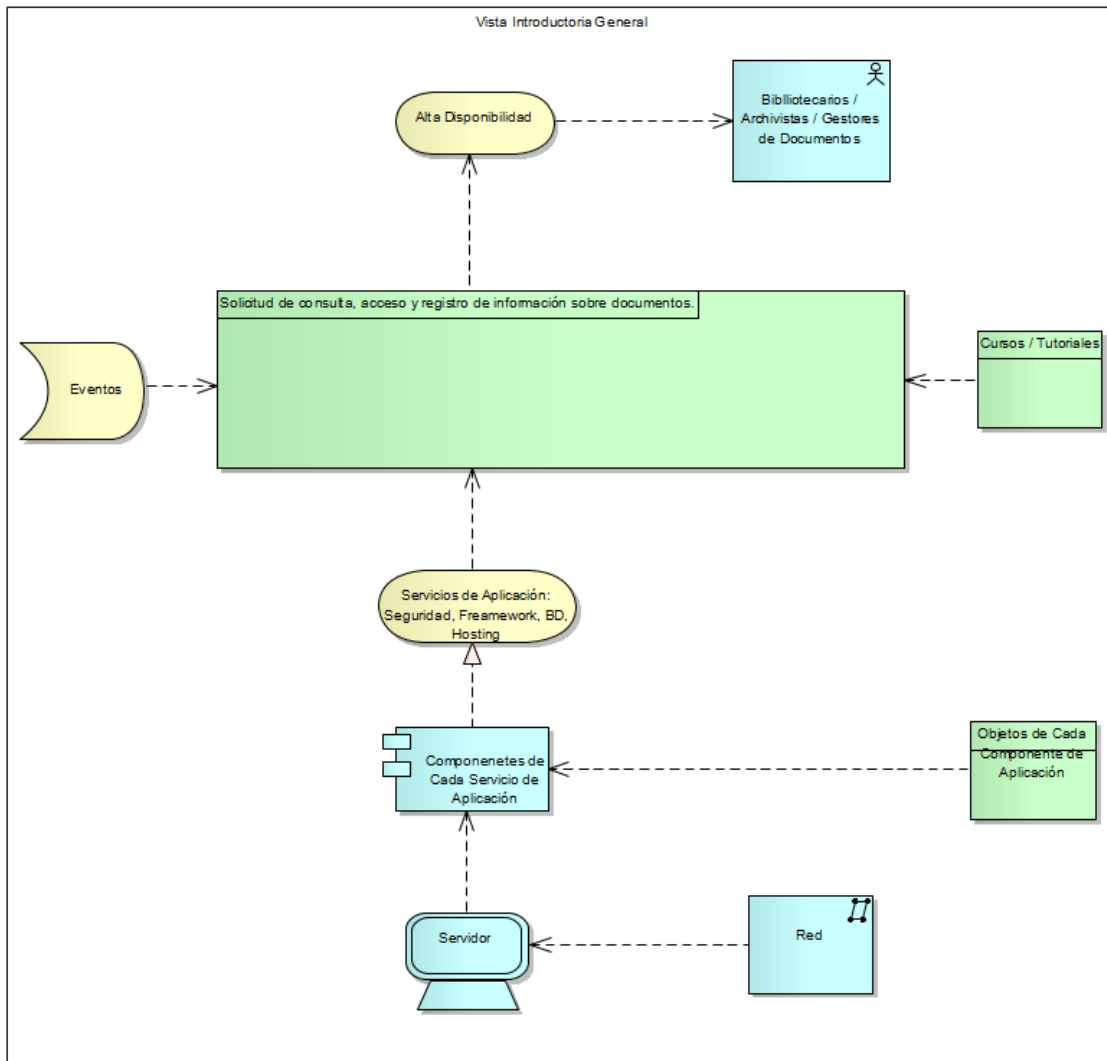


Ilustración 7: Fuente Autora: Servicios arquitectónicos

Descripción Servicio de Aplicación Seguridad HTTP

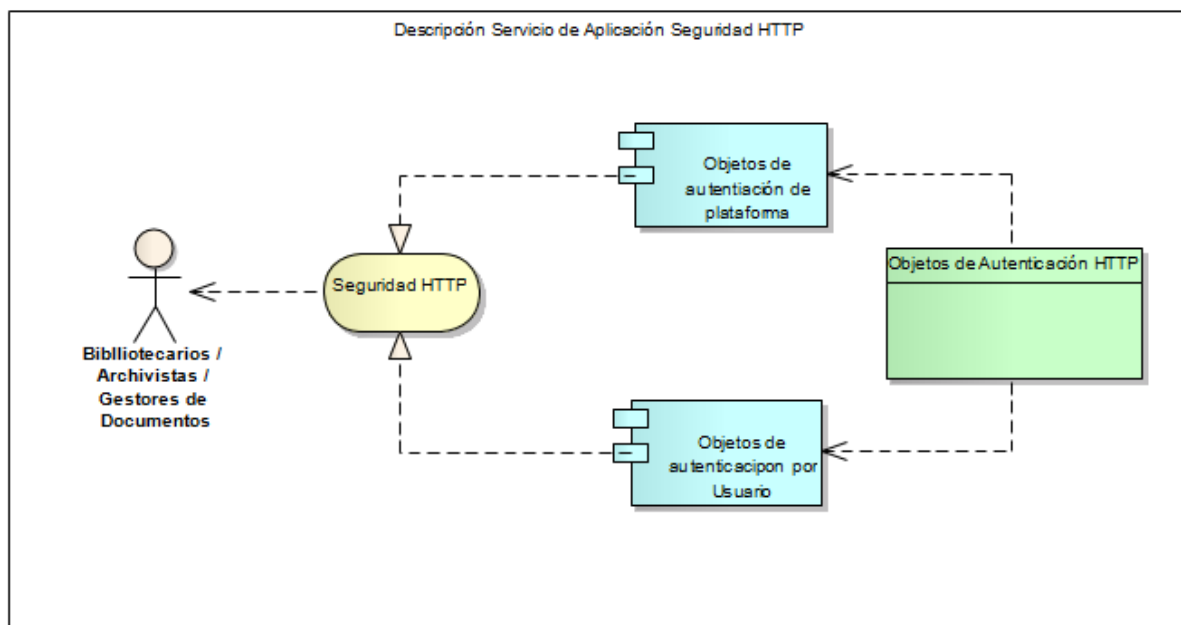


Ilustración 8Fuente Autora: Descripción de aplicación seguridad.

Descripción Servicio de Aplicación Seguridad en el Envío de Datos

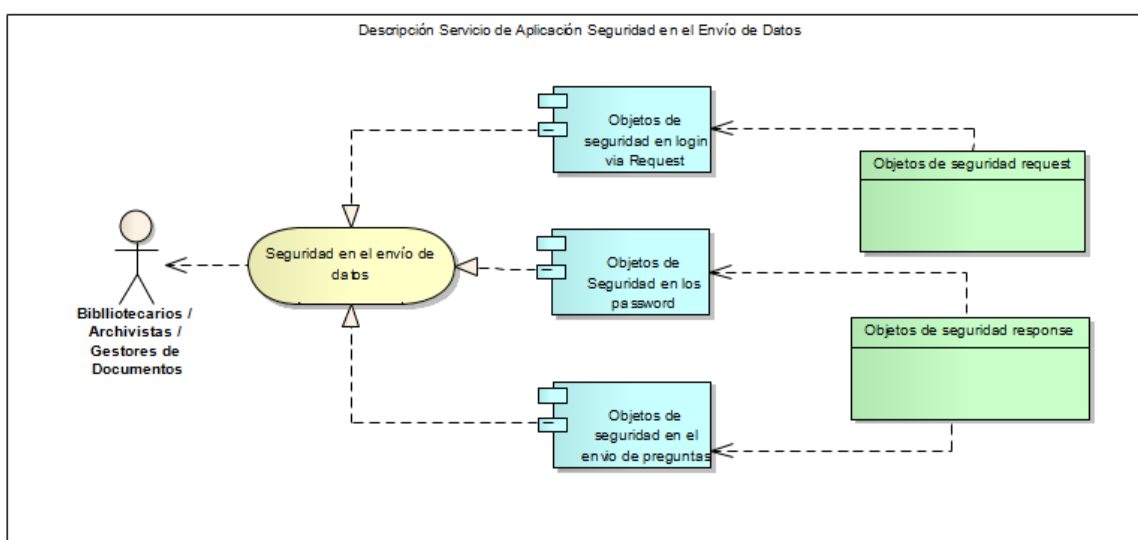


Ilustración 9: Fuente Autora: Descripción de la aplicación seguridad en el envío de datos.

Descripción Servicio de Aplicación Reutilización de Frameworks Probados en el Mercado

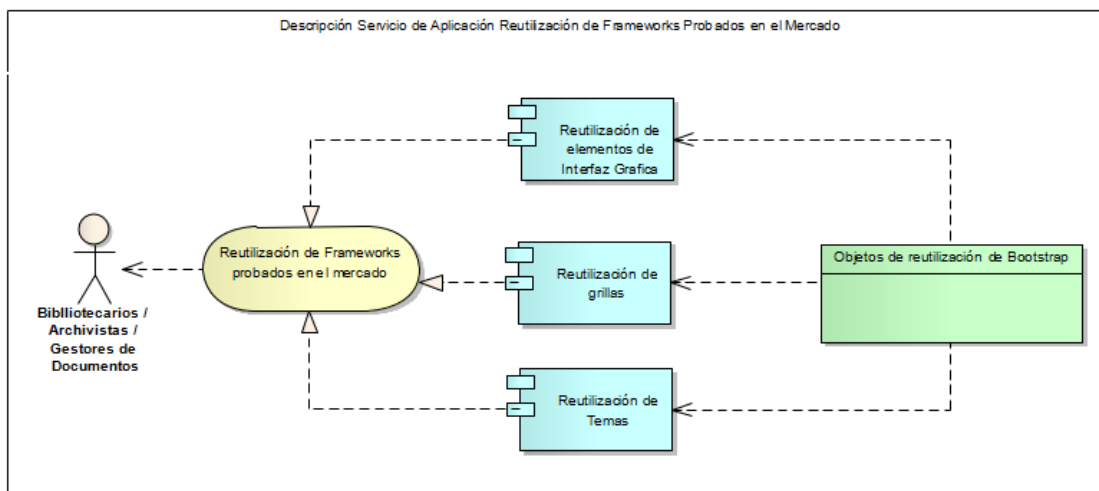


Ilustración 10: Fuente Autora: Descripción servicio de aplicación reutilización de Frameworks

Descripción Servicio de Aplicación Configuración del Entorno

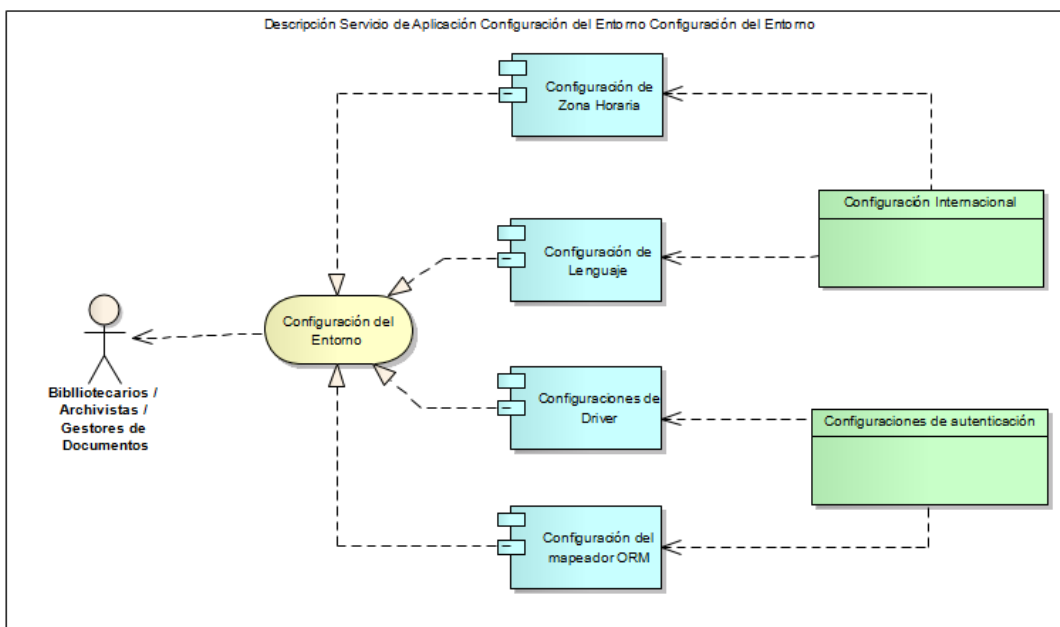


Ilustración 11: Fuente Autora: Descripción servicio de aplicación configuración del entorno.

Descripción Servicio de Aplicación Bases de Datos

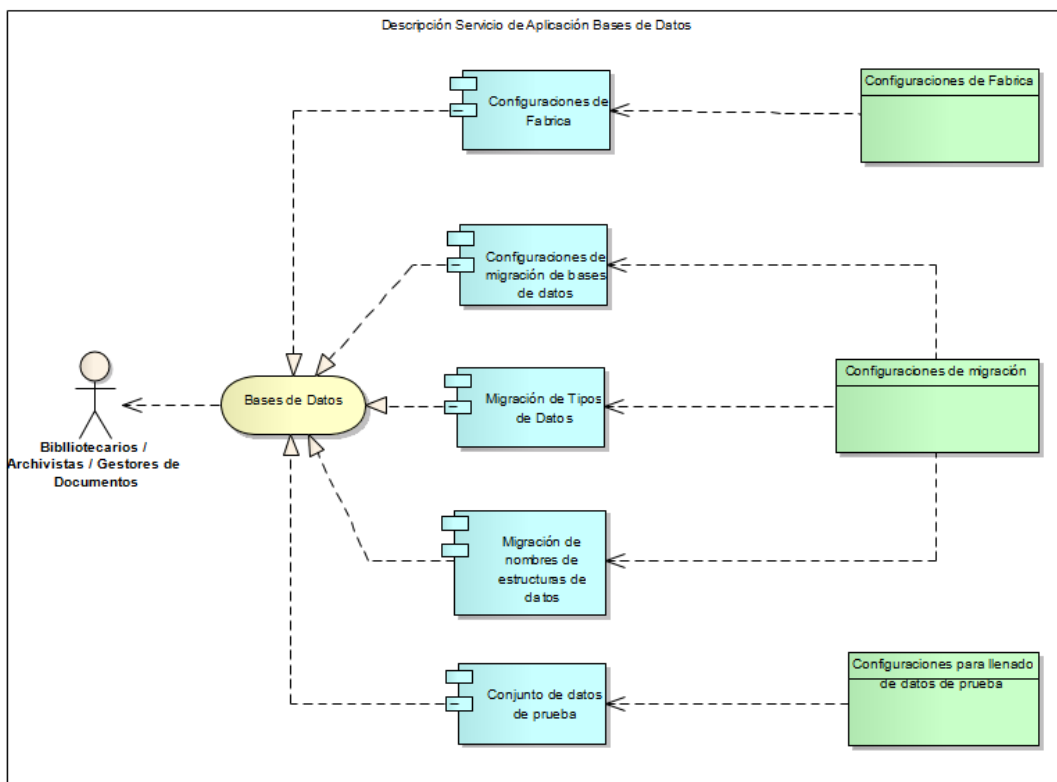


Ilustración 12: Fuente Autora: Descripción servicio de aplicación bases de datos

Descripción Servicio de Hosting en las Aplicaciones

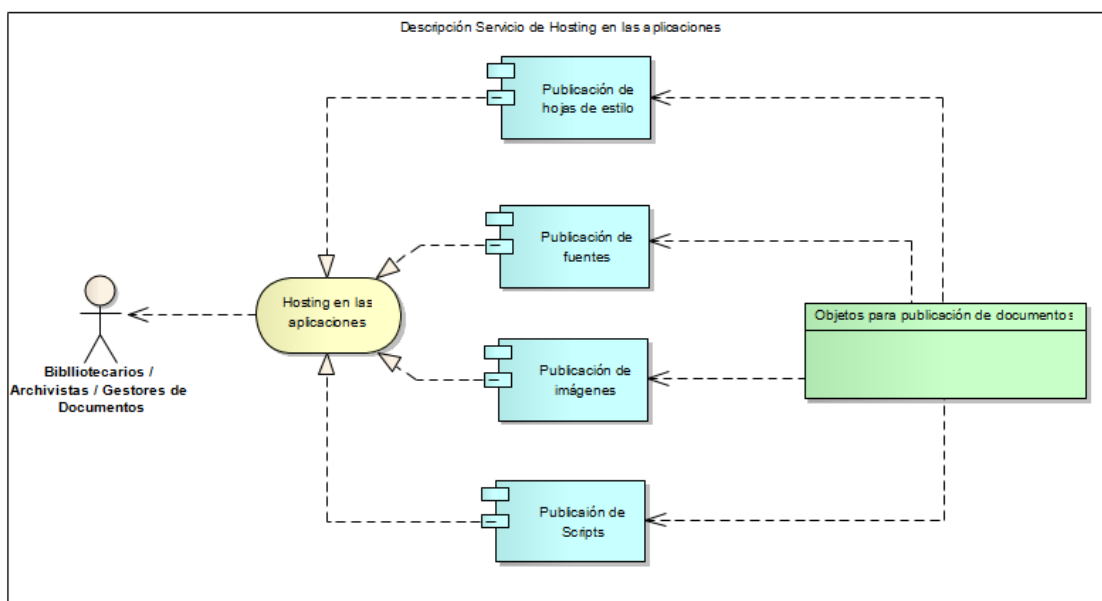


Ilustración 13: Fuente Autora: Descripción servicio de hosting en las aplicaciones

Descripción Servicio de Recursos

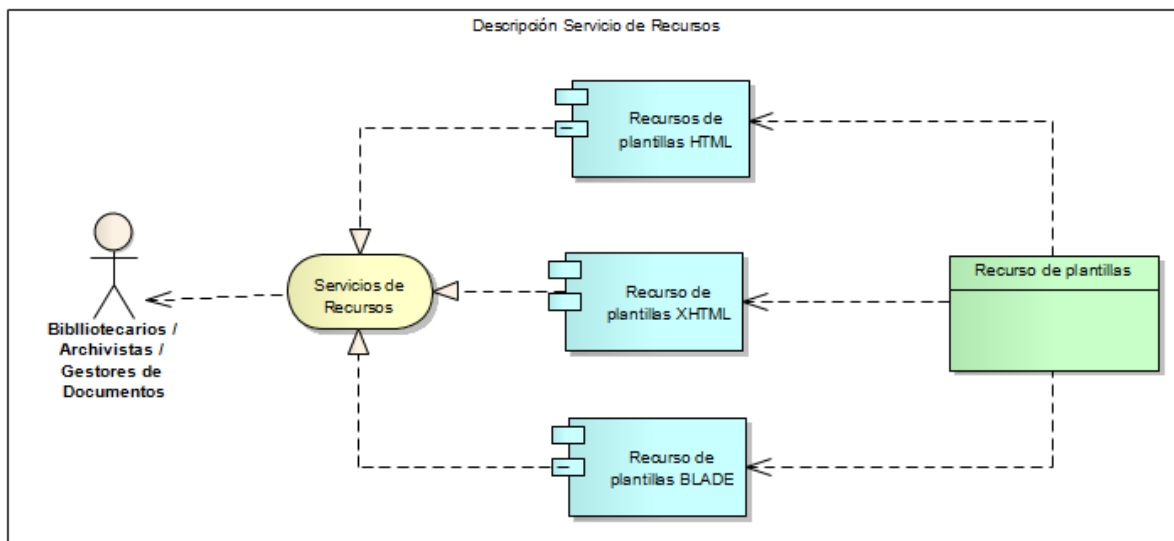


Ilustración 14: Fuente Autora: Descripción servicio de recursos

Descripción Servicio de Almacenamiento

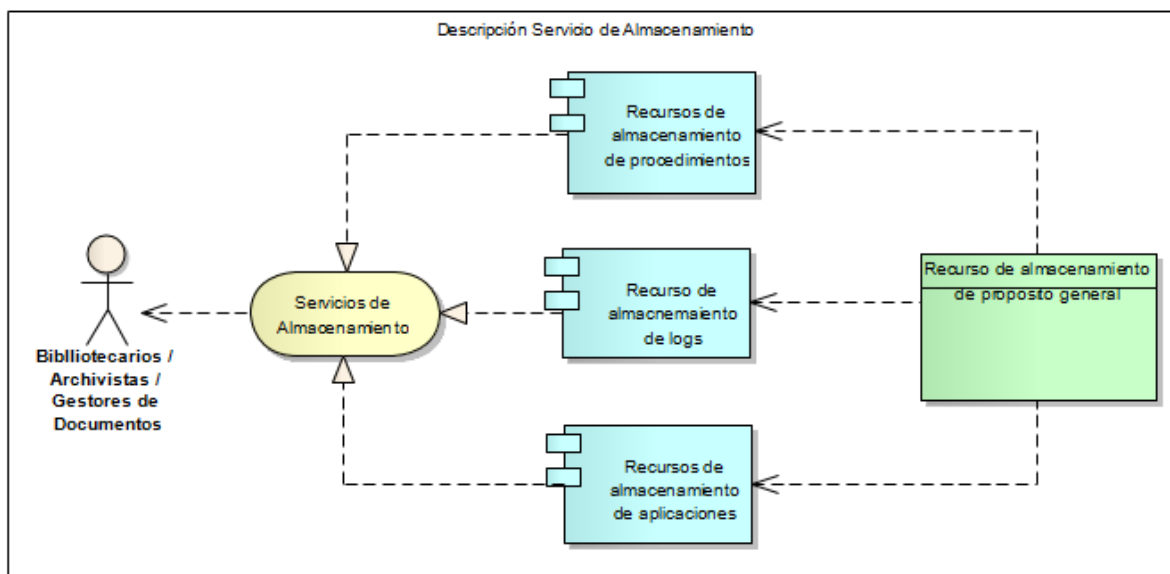


Ilustración 15: Fuente Autora: Descripción servicio de almacenamiento

Descripción Servicio de Testeo

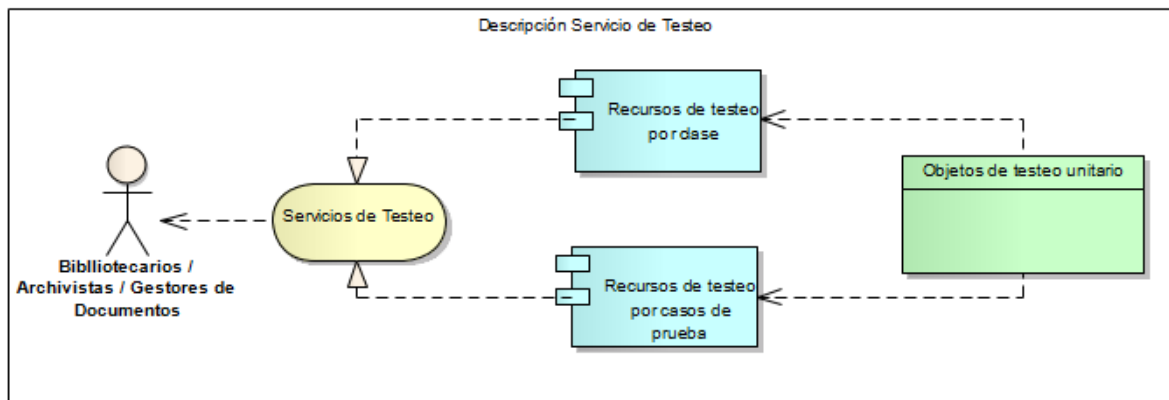


Ilustración 16: Fuente Autora: Descripción servicio de testeo.

Descripción Servicio de Actualización

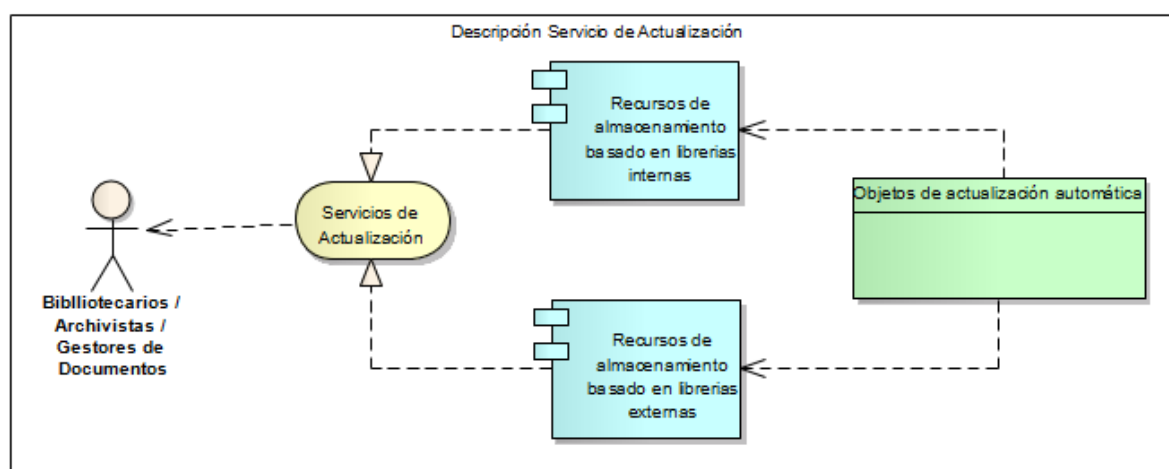


Ilustración 17: Fuente Autora: Descripción servicio de actualización.

11.IMPLEMENTACIÓN SERVICIOS ARQUITECTONICOS

En esta sección se relacionan cada uno de los servicios arquitectónicos expuestos en el capítulo 10 con los mecanismos y los archivos que fueron necesarios desarrollar, descargar o configurar para cumplir con la funcionalidad plasmada por cada servicio arquitectónico.

11.1 Descripción Servicio de Aplicación Seguridad HTTP

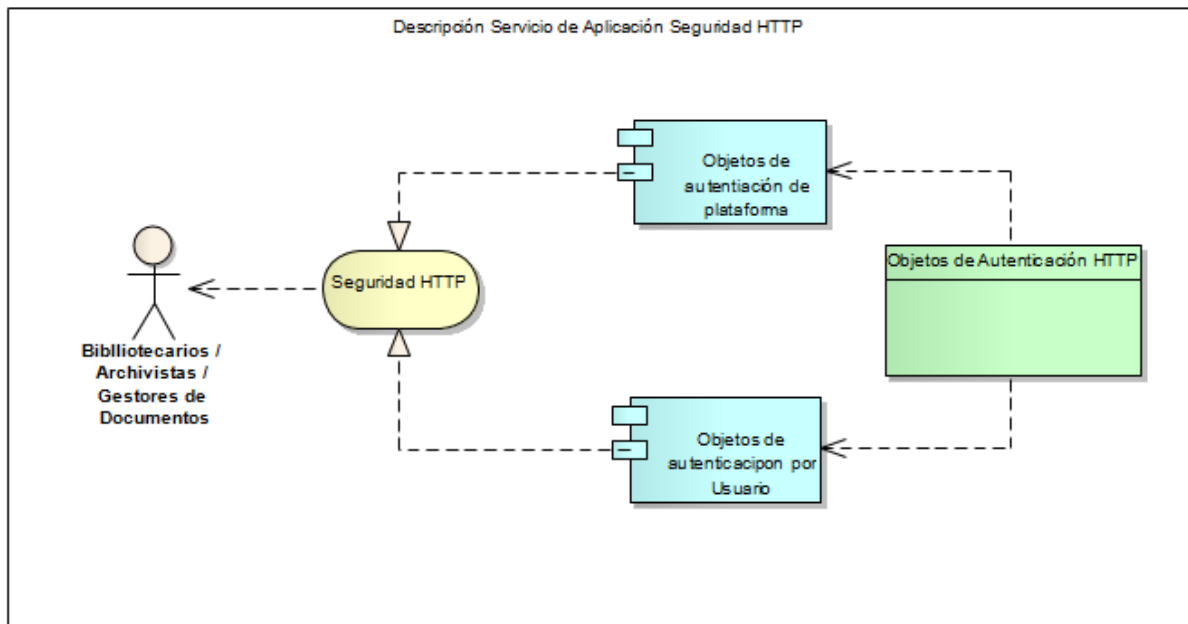


Ilustración 18: Fuente Autora: Descripción servicio de aplicación seguridad HTTP

Para el caso del servicio de aplicación basada en seguridad HTTP se implementaron validaciones tipo request sobre HTTP, que permitieron hacer de este servicio una realidad.



Ilustración 19: Fuente Autora: Paquete middleware.

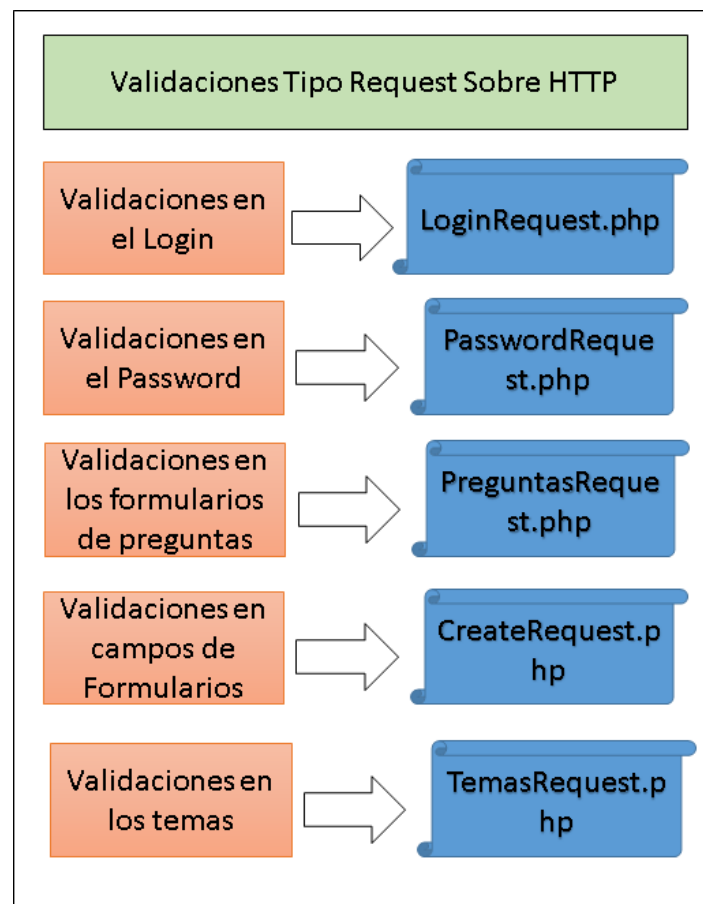


Ilustración 20: Fuente Autora: Paquete request

11.2 Descripción Servicio de Aplicación Seguridad en el Envío de Datos

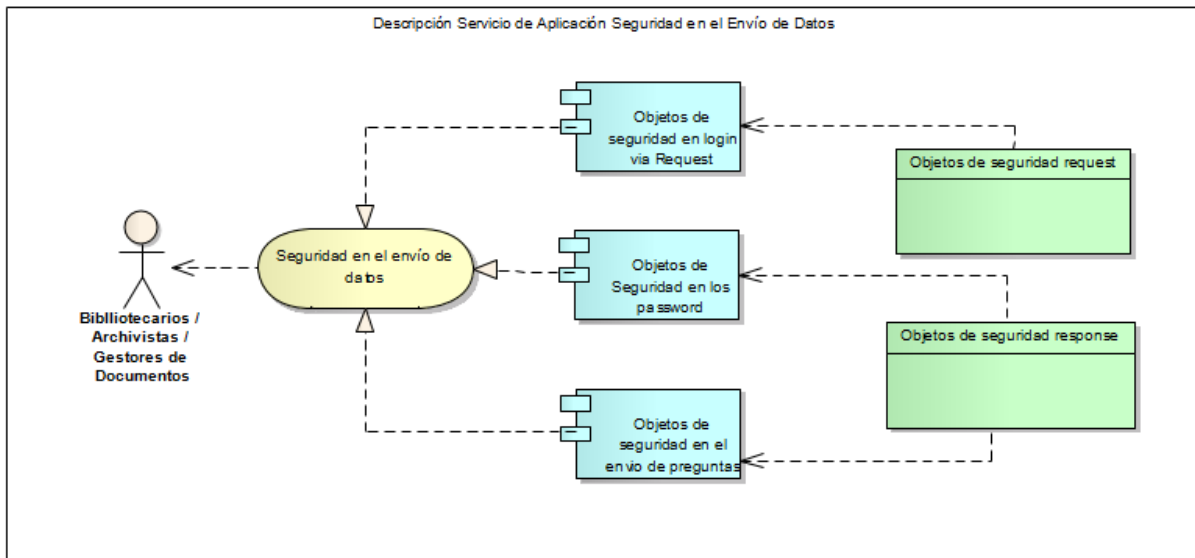


Ilustración 21: Fuente Autora: Descripción servicio de aplicación seguridad en el envío de datos

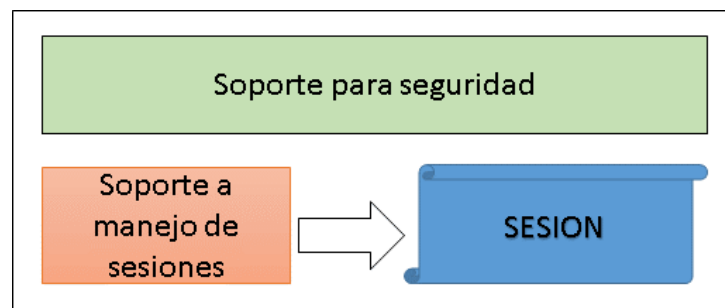


Ilustración 22: Fuente Autora: Soporte para seguridad.

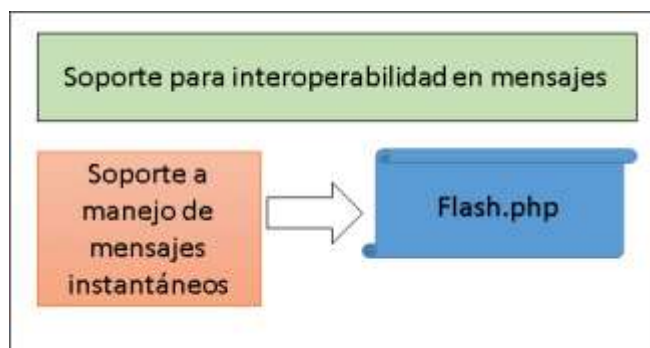


Ilustración 23: Fuente Autora: Soporte para interoperabilidad en mensajes.

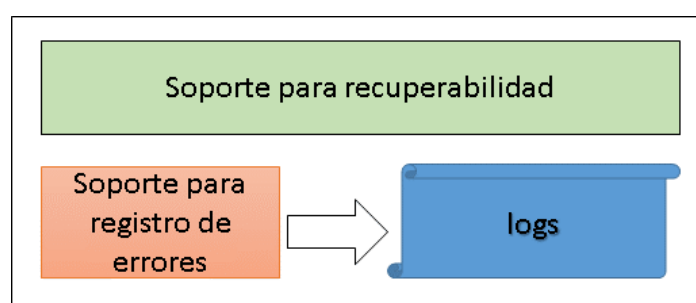


Ilustración 24: Fuente Autora: Soporte para recuperabilidad.

11.3 Descripción Servicio de Aplicación Reutilización de Frameworks Probados en el Mercado

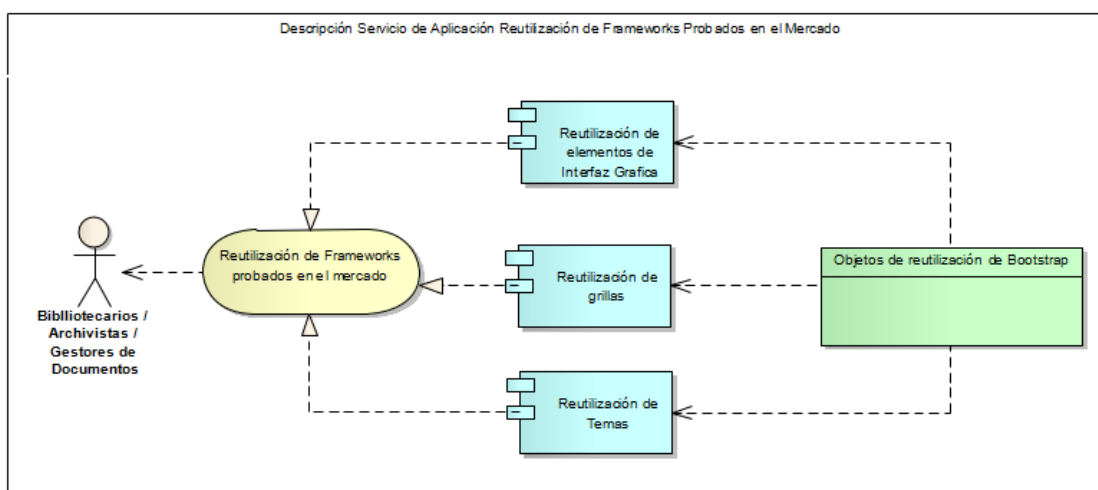


Ilustración 25: Fuente Autora: Descripción servicio aplicación reutilización de Frameworks probados en el mercado.

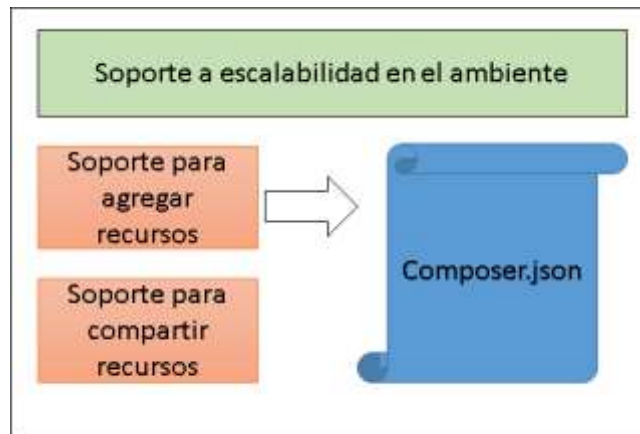


Ilustración 26: Fuente Autora: Soporte de escalabilidad en el ambiente.

11.4 Descripción Servicio de Aplicación Configuración del Entorno

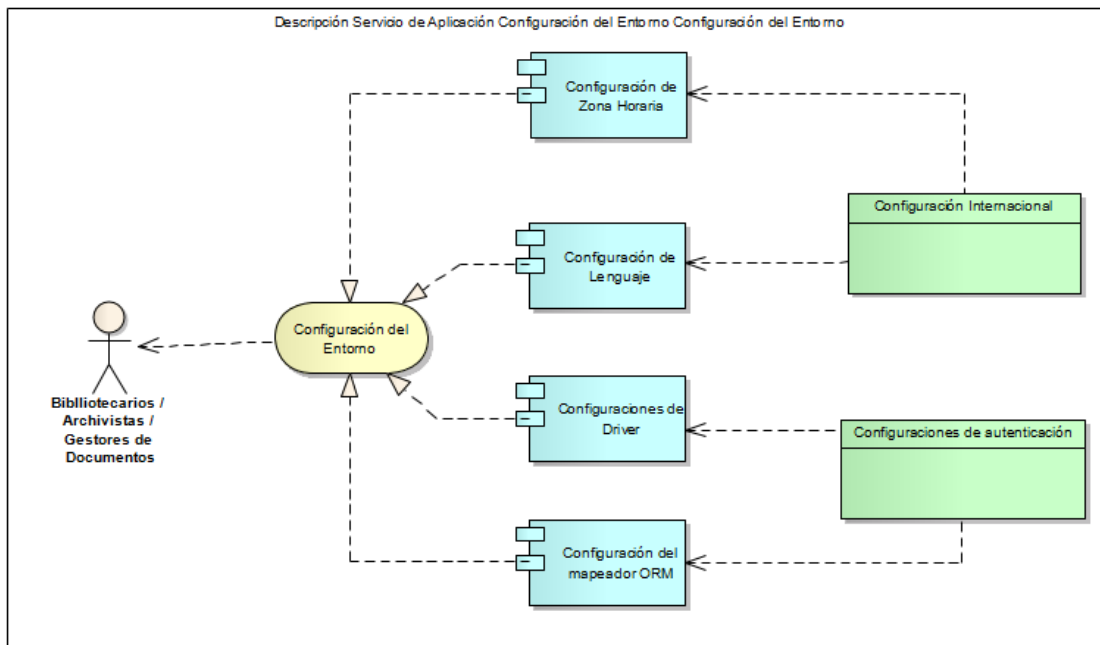


Ilustración 27: Fuente Autora: Descripción del servicio aplicación para configuración del entorno.

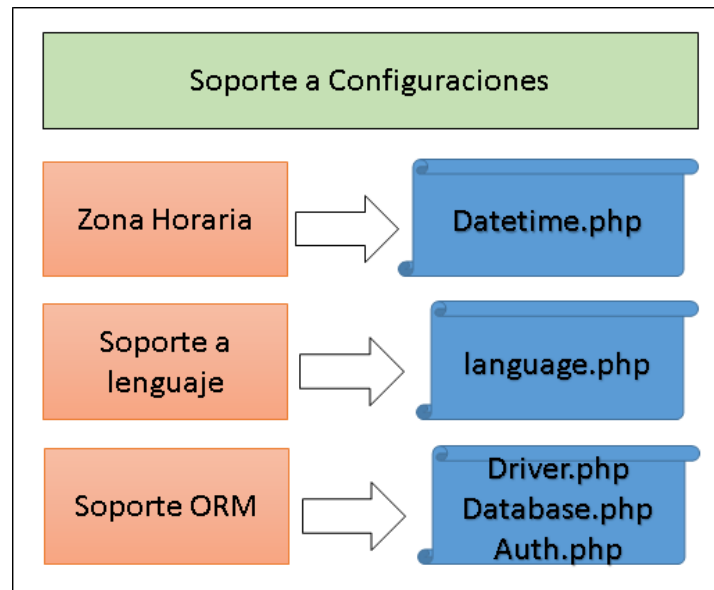


Ilustración 28: Fuente Autora: Soporte a configuraciones.

11.5 Descripción Servicio de Aplicación Bases de Datos

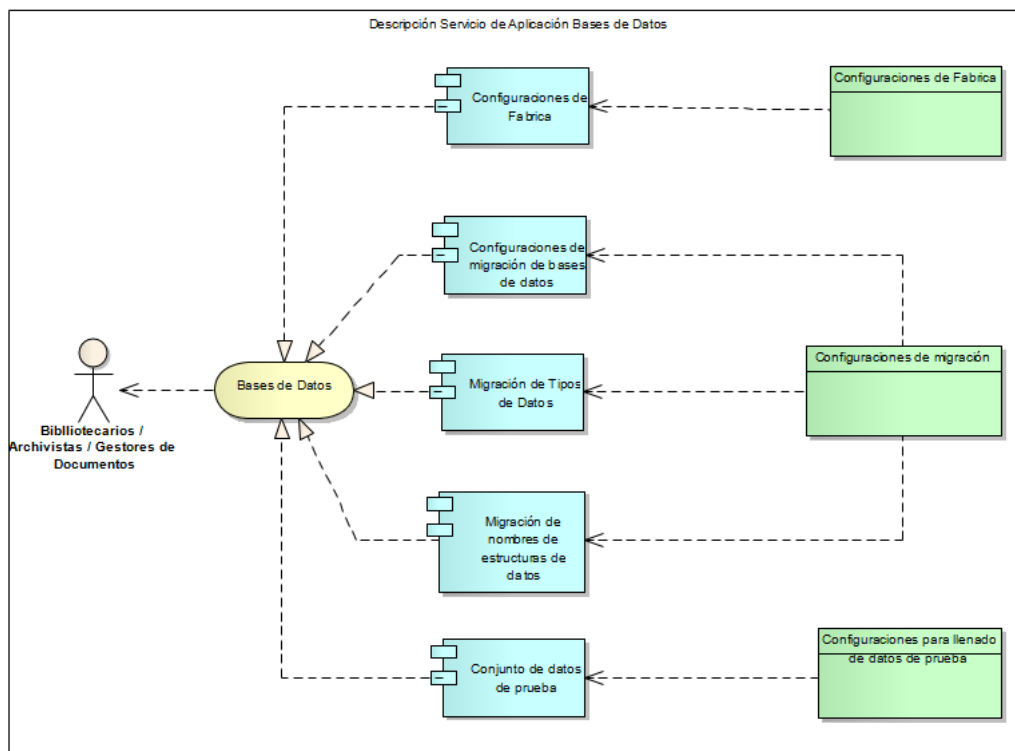


Ilustración 29: Fuente Autora: Descripción servicio de aplicación bases de datos.

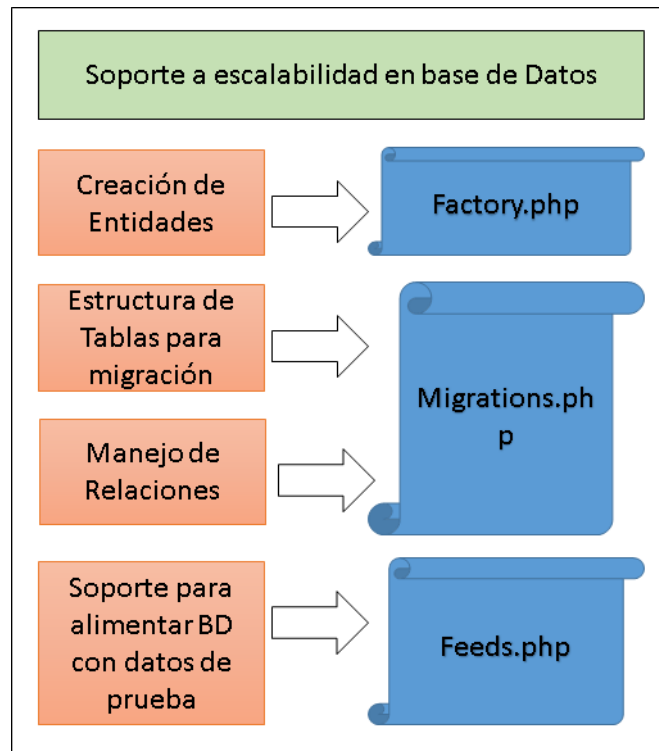


Ilustración 30: Fuente Autora: Soporte a escalabilidad en bases de datos.

11.6 Descripción Servicio de Hosting en las Aplicaciones

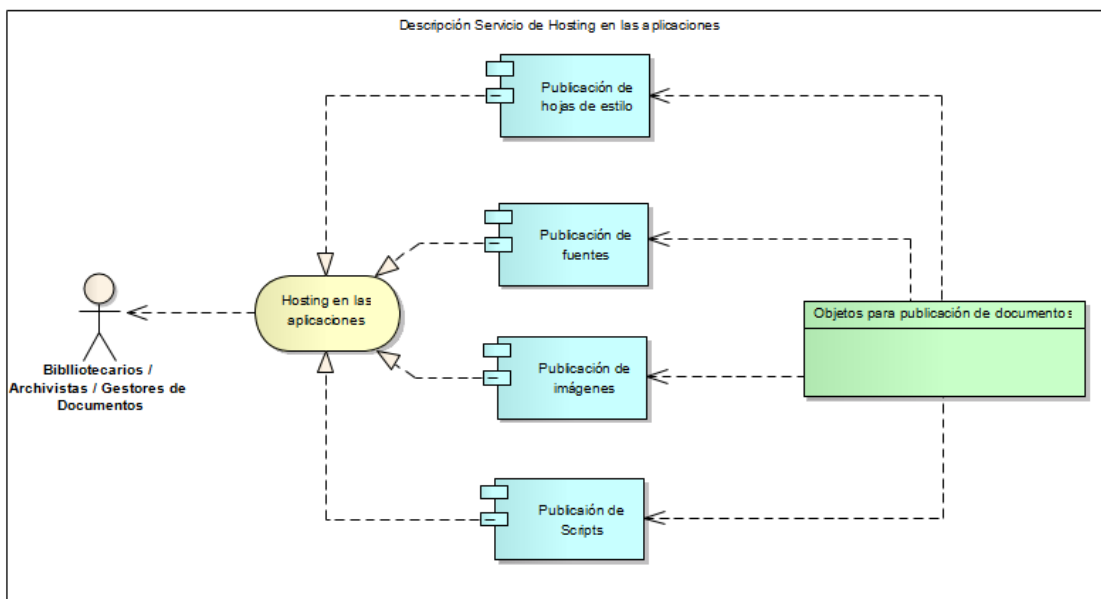


Ilustración 31: Fuente Autora: Descripción servicio hosting en las aplicaciones.



Ilustración 32: Fuente Autora: Soporte para almacenamiento de recursos.

11.7 Descripción Servicio de Recursos

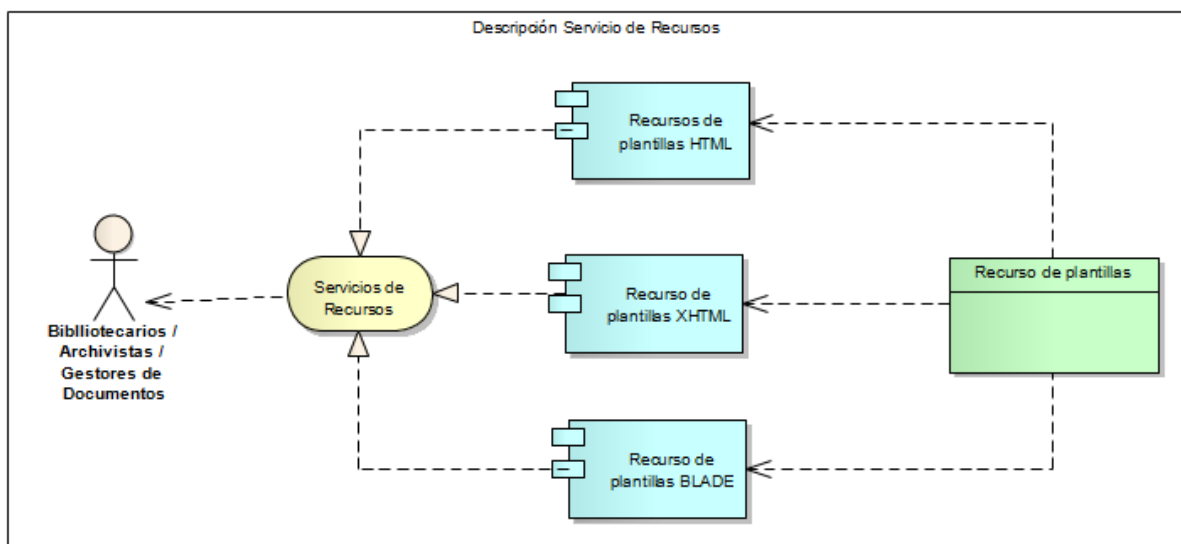


Ilustración 33: Fuente Autora: Descripción servicio de recursos.

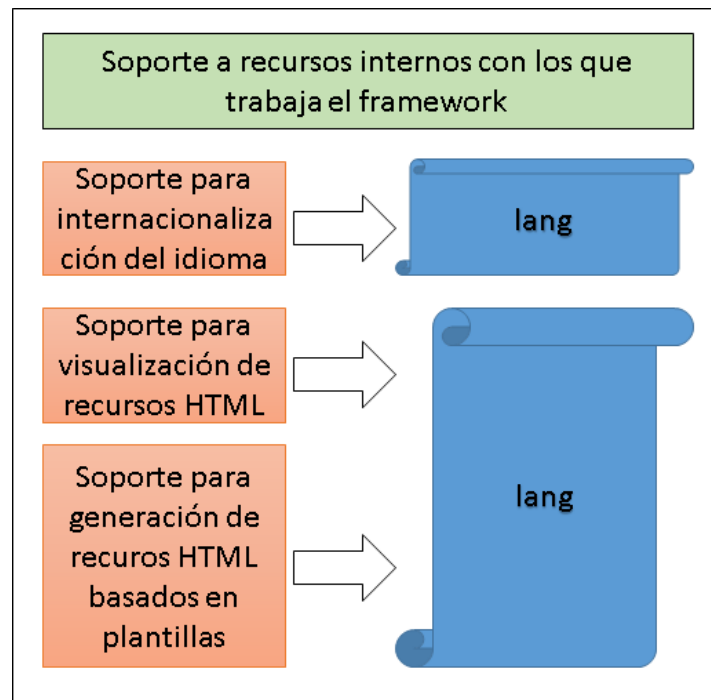


Ilustración 34: Fuente Autora: Soporte a recursos internos con los que trabaja el framework.

11.8 Descripción Servicio de Almacenamiento

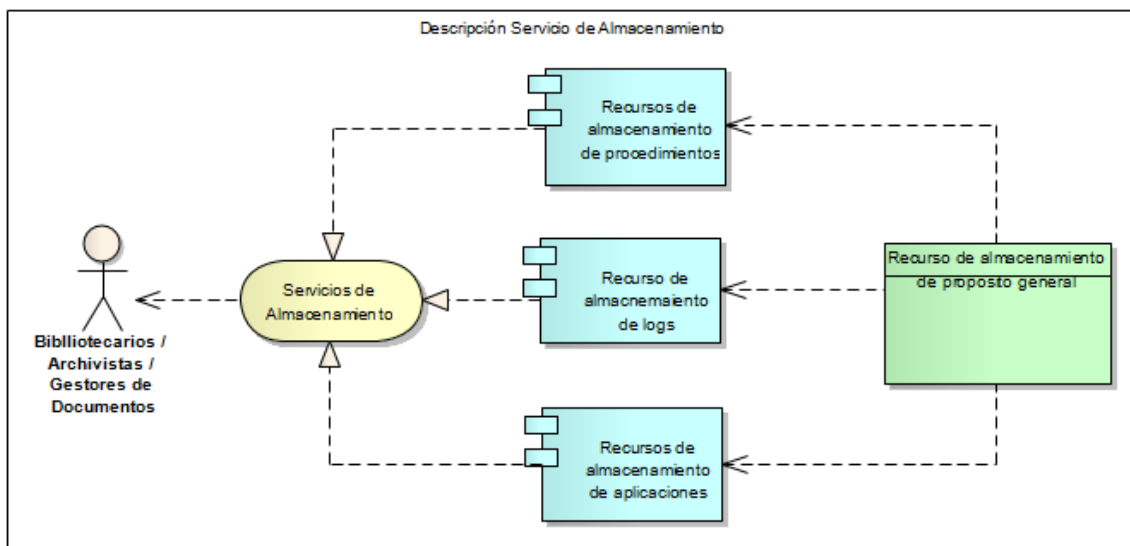


Ilustración 35: Fuente Autora: Descripción servicio de almacenamiento.

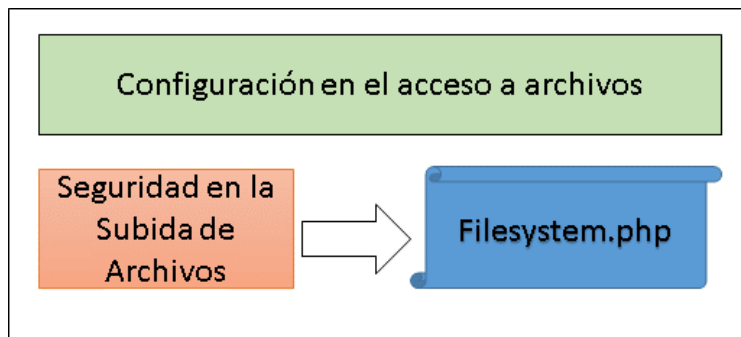


Ilustración 36: Fuente Autora: Configuración en el acceso a archivos.

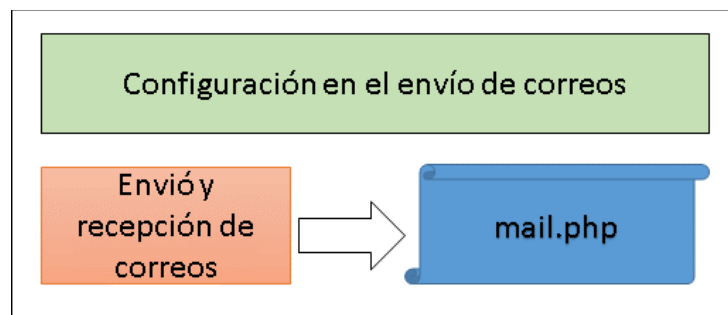


Ilustración 37: Fuente Autora: Configuración en el envío de correos.

11.9 Descripción Servicio de Testeo

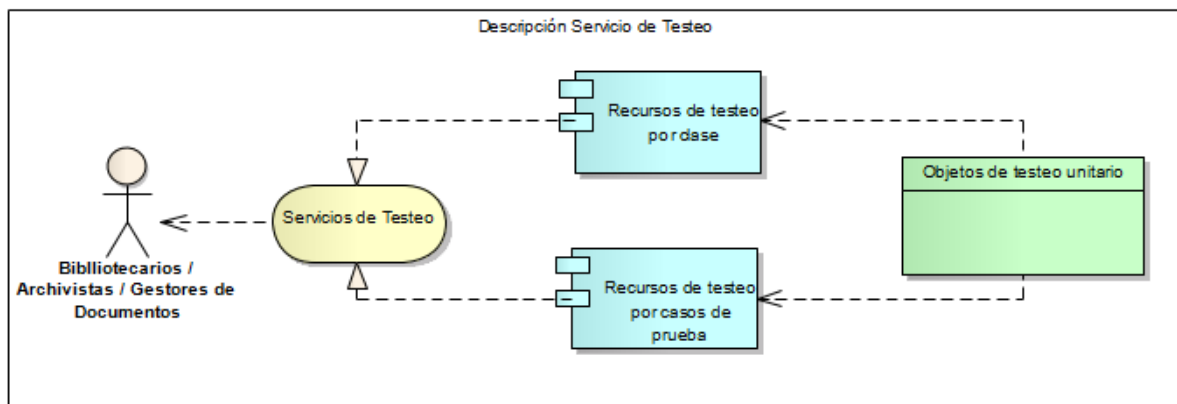


Ilustración 38: Fuente Autora: Descripción servicio de testeo.

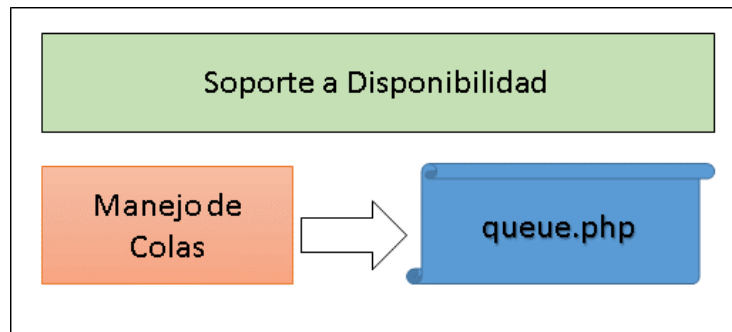


Ilustración 39: Fuente Autora: Soporte a disponibilidad.

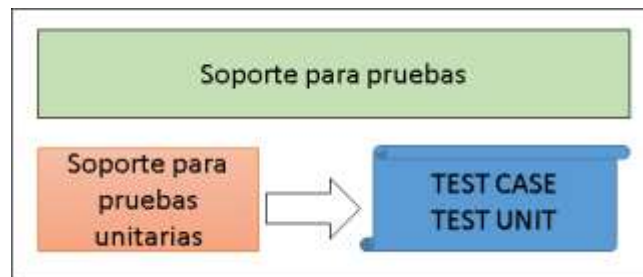


Ilustración 40: Fuente Autora: Soporte a pruebas.

11.10 Descripción Servicio de Actualización

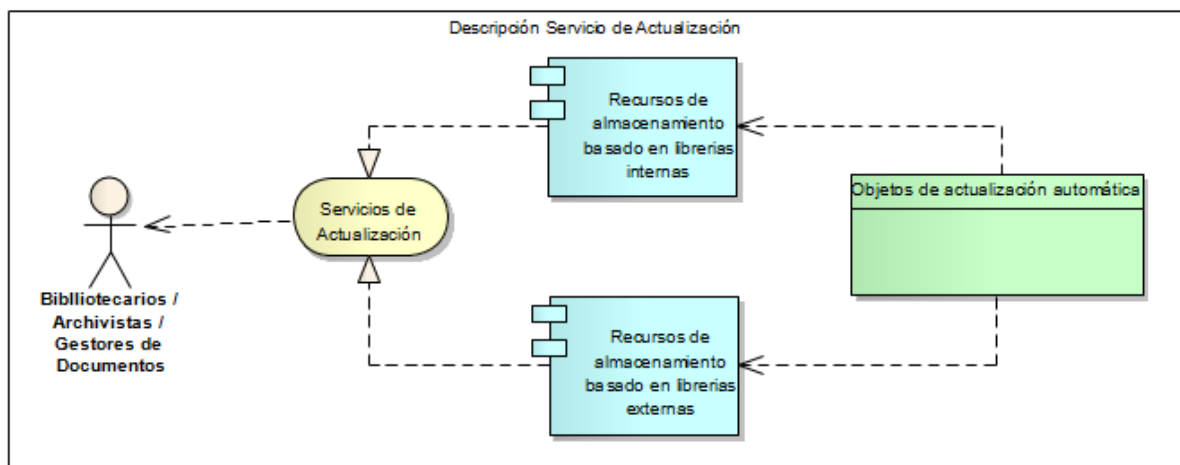


Ilustración 41: Fuente Autora: Descripción servicio de actualización.

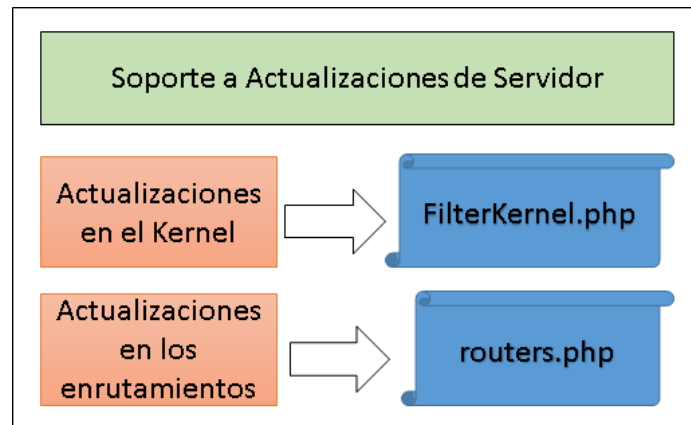


Ilustración 42: Fuente Autora: Soporte actualizaciones de servidor.

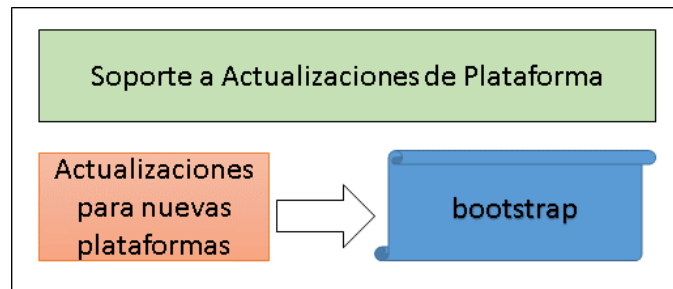


Ilustración 43: Fuente Autora: Soporte actualizaciones de plataforma.

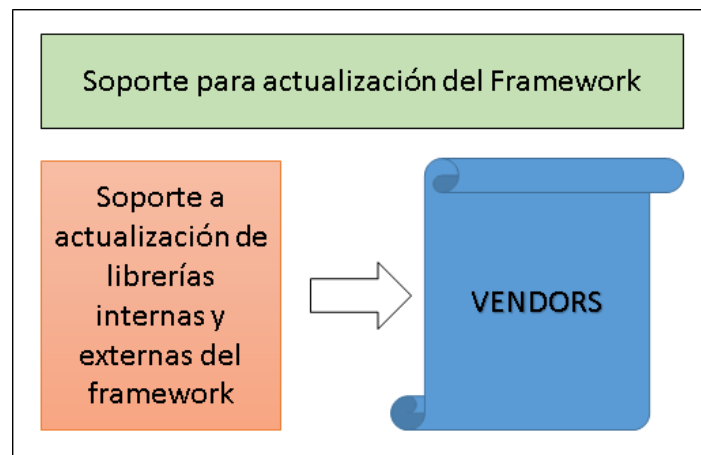


Ilustración 44: Fuente Autora: Soporte actualización de Framework.

12. VISUALIZACIÓN DE LA IMPLEMENTACION DEL SISTEMA

A continuación se muestra la interfaz gráfica del sistema para el manejo de documentos, prototipo diseñado como un ambiente virtual, que lo compone de:

Actor	Descripción
Administrador	Actor encargado de realizar consultas, crear, modificar y eliminar los usuarios como Administradores, Estudiantes.
Estudiante	Actor que interactúa con el ambiente virtual. Realiza ejercicios de práctica, de acuerdo a la información que almacena la plataforma.

Interfaz Principal de acceso al sistema

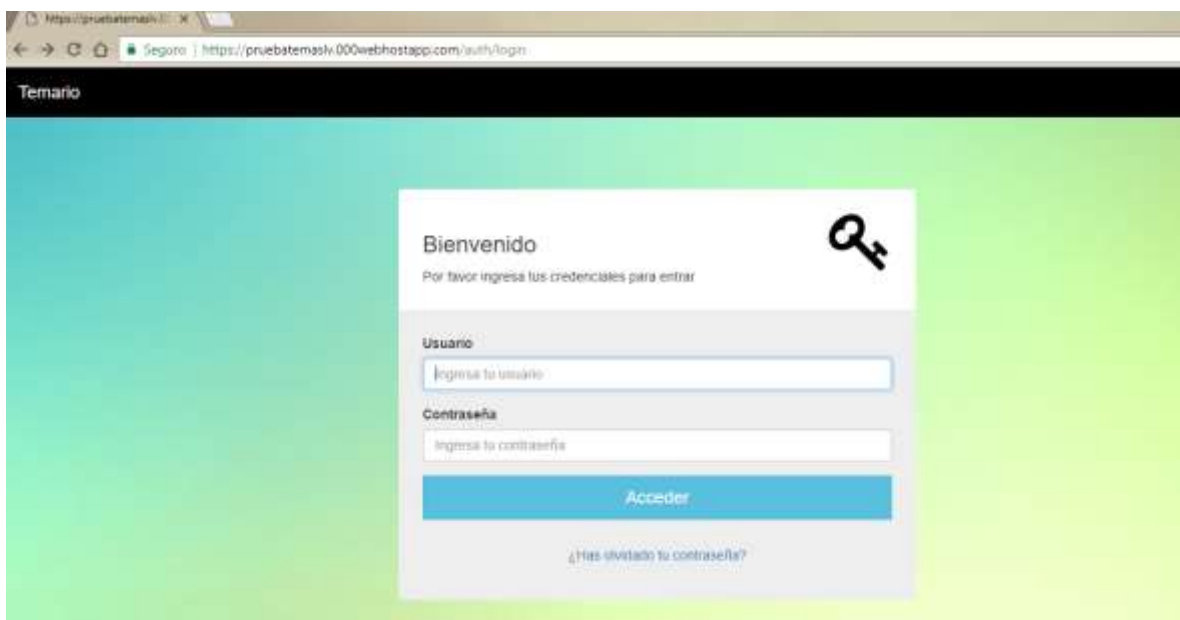


Ilustración 45: Fuente Autora: Interfaz principal de acceso al sistema.

Interfaz usuario Estudiante

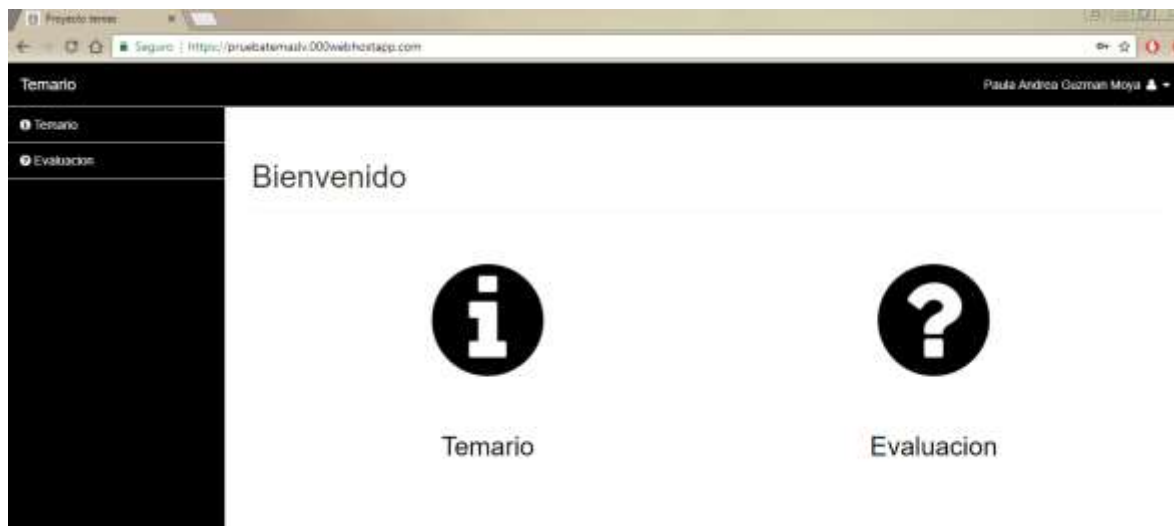


Ilustración 46: Fuente Autora: Interfaz usuario estudiante.

Interfaz Temario - Estudiante

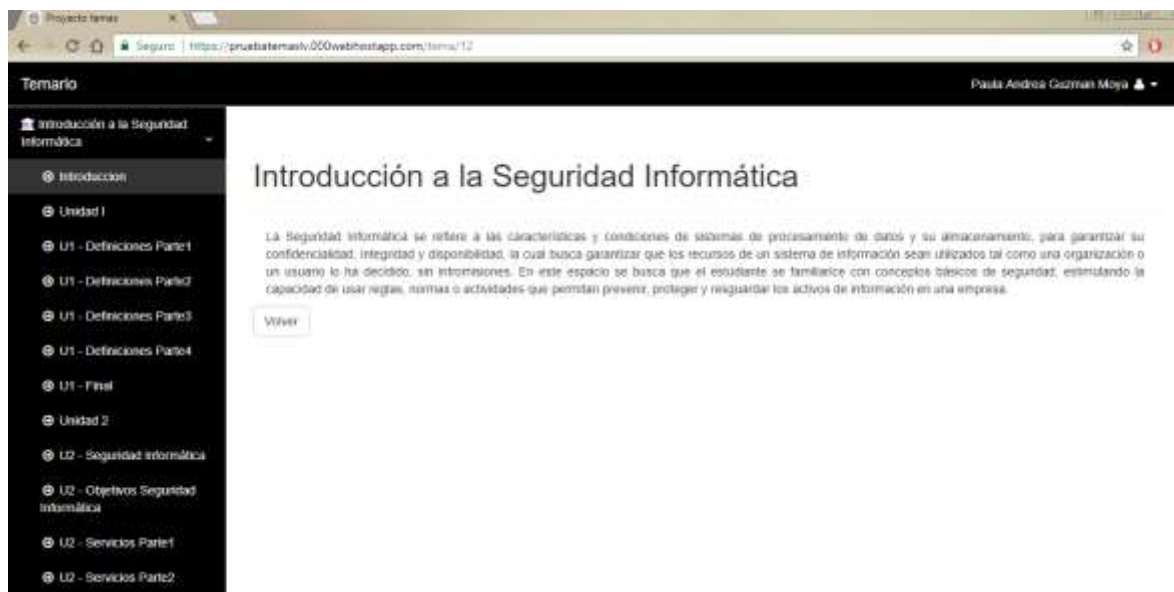


Ilustración 47: Fuente Autora: Interfaz temario estudiante.

Interfaz Evaluación - Estudiante



Ilustración 48: Fuente Autora: Interfaz evaluación estudiante.

Interfaz usuario Administrador

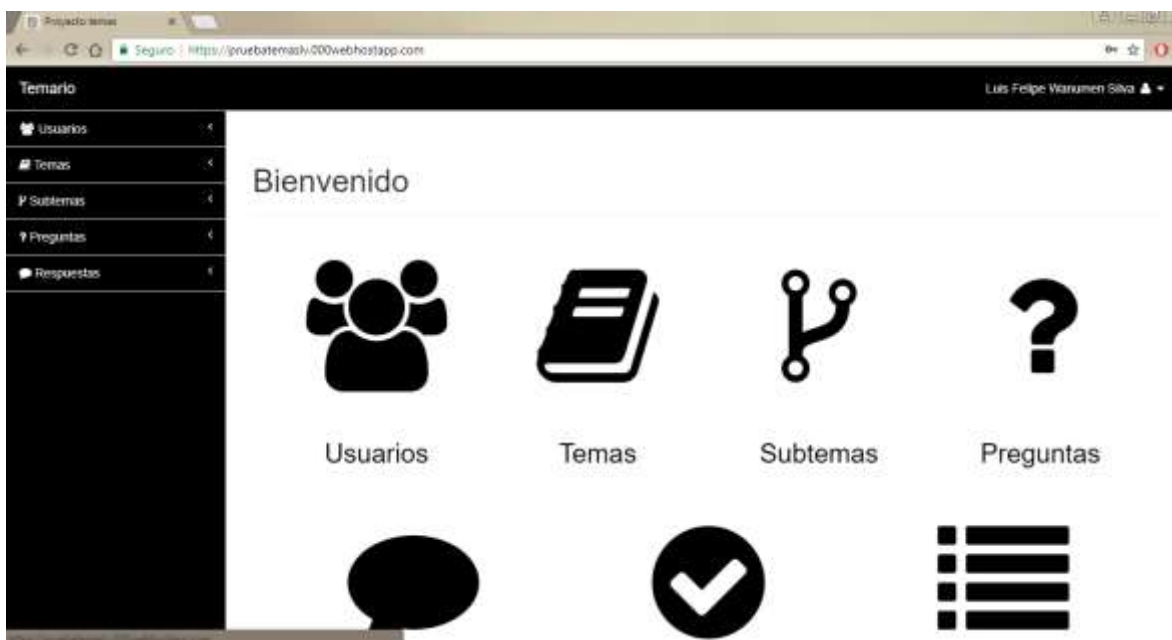


Ilustración 49: Fuente Autora: Interfaz usuario administrador.

Interfaz Usuarios



Ilustración 50: Fuente Autora: Interfaz usuarios.

Interfaz Temas



Ilustración 51: Fuente Autora: Interfaz temas.

Interfaz Subtemas



Ilustración 52: Fuente Autora: Interfaz subtemas.

Interfaz de Preguntas

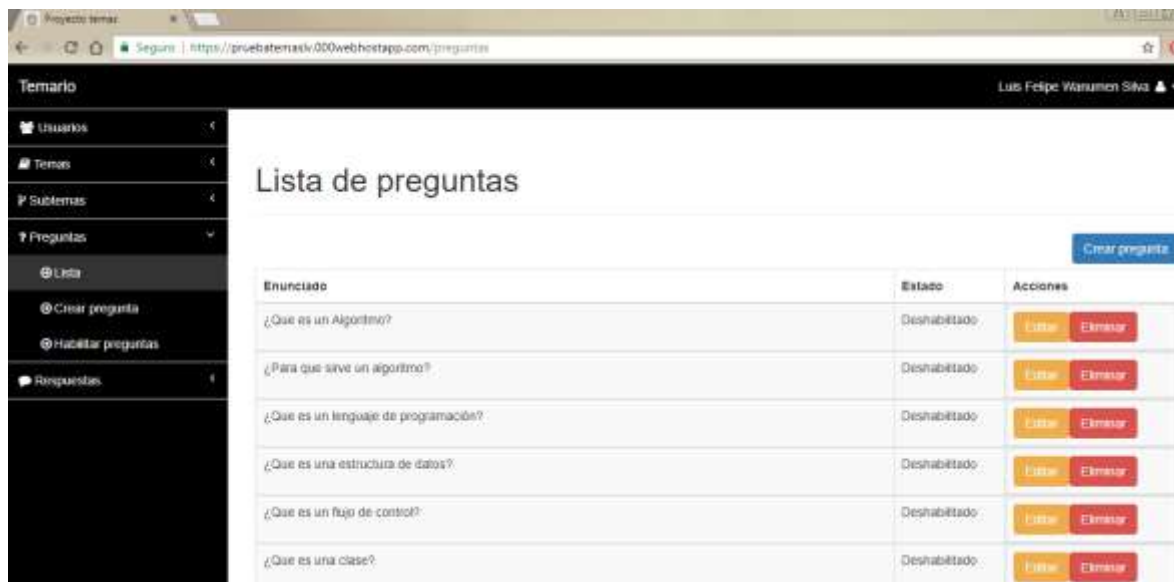


Ilustración 53: Fuente Autora: Interfaz de preguntas.

Interfaz de Habilitar Preguntas



Ilustración 54: Fuente Autora: Interfaz de habilitar preguntas.

Interfaz de Respuestas



Ilustración 55: Fuente Autora: Interfaz de respuestas.

Interfaz Resultado de los Alumnos



Usuario	Respuestas correctas	Respuestas totales	Calificación	Acciones
20121078007	3	20	0.75	Agregar registro

Ilustración 56: Fuente Autora: Interfaz resultados de los alumnos.

13.PRUEBAS TAM HECHAS AL SISTEMA

Si bien es cierto que existen formas para hacer pruebas a los sistemas, también es cierto que las mejores pruebas casi siempre las hacen los usuarios de los sistemas, en el sentido que son ellos, los que evalúan el desempeño, el consumo de recursos.

Para ello se han elaborado para el actual sistema una serie de pruebas basadas en el modelo TAM Modelo de Aceptación de la Tecnología (Technology Acceptance Model - TAM) de Davis¹² resumido en la figura 11, es una teoría de los sistemas de información que modela cómo los usuarios llegan a aceptar y utilizar una tecnología. El modelo sugiere que cuando a los usuarios se les presenta una nueva tecnología, una serie de factores influyen en su decisión sobre cómo y cuándo la van a utilizar, dicho de otra forma, sirven de base para determinar las actitudes enfocadas al uso del sistema. Estos factores son:

PU (Perceived usefulness, Utilidad Percibida): “el grado en que una persona cree que el uso de un determinado sistema mejora su rendimiento en el trabajo”.

FUP (Perceived ease-of-use, Percepción de facilidad de uso): “el grado en que una persona cree que utilizando un sistema en particular, podrá liberarse del esfuerzo que le conlleva realizar un trabajo”.

¹² SG, «SG, VOLUMEN 2,» [En línea]. <<https://sg.com.mx/content/view/265>> [Citado en: 18 de enero de 2018].

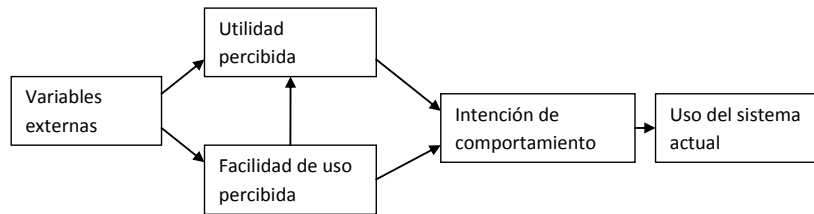


Ilustración 57: Fuente: Davis [2]: Modelo de Aceptación de la Tecnología.

La aplicación de la validación usando TAM se dio mediante el siguiente formulario de encuesta que se presentó a estudiantes y profesores de ingeniería que ven y explican temas de termodinámica. Se tomó como guía el tipo de encuesta realizado por Orantes [4], con respectivos ajustes para el entorno en el cual se va a aplicar, además de controlar las opciones de votación de 7 opciones como maneja orantes a 5 (totalmente de acuerdo - 5, de acuerdo - 4, ni acuerdo ni desacuerdo - 3, en desacuerdo - 2, totalmente en desacuerdo - 1):

Facilidad de Uso

Afirmación.1: Es fácil aprender a operar el nuevo sistema.

Afirmación.2: Es fácil acceder al sistema para hacer lo que deseo.

Afirmación.3: Es fácil aumentar mi experiencia gracias al uso del nuevo sistema.

Afirmación.4: El nuevo sistema es fácil de utilizar.

Afirmación.5: Sería fácil llegar a ser un experto en el manejo del sistema.

Utilidad Percibida

Afirmación.6: El uso del nuevo sistema me ayudaría a atender mejor el tema de la seguridad informática.

Afirmación.7: El uso del nuevo sistema mejoraría el funcionamiento de la seguridad informática en las empresas.

Afirmación.8: El uso del nuevo sistema aumenta mi productividad en el momento de implementar políticas de seguridad informática.

Afirmación.9: El uso del nuevo sistema incrementa la efectividad con que los estudiantes realizan implementaciones de aplicaciones con seguridad informática.

Afirmación.10: El nuevo sistema es de utilidad dentro de estudiantes de seguridad informática.

Actitud hacia el Uso

Afirmación.11: Usar el nuevo sistema es una buena idea.

Afirmación.12: Usar el nuevo sistema es una inteligente idea.

Afirmación.13: Me gusta la idea de usar el nuevo sistema.

Afirmación.14: Usar el nuevo sistema me parece (placentero/no-placentero).

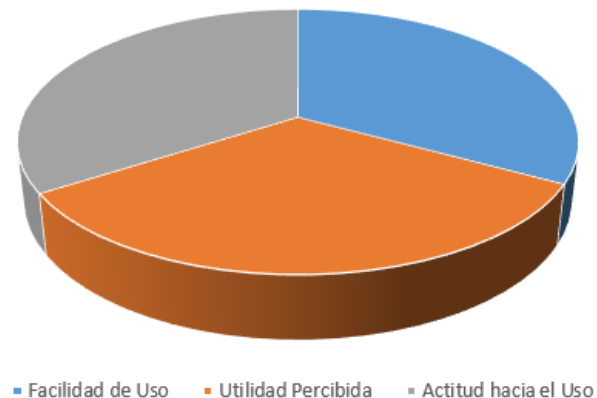
Análisis de resultados

Usuarios	1	2	3	4	5	6	7	8	9	10	11	12	13	Ponderado
Facilidad de Uso														
Afirmación.1: Es fácil aprender a operar el nuevo sistema	5	4	3	4	4	4	4	5	5	5	4	5	5	4,38461538
Afirmación.2: Es fácil acceder al sistema para hacer lo que deseo	4	5	5	5	4	5	3	4	5	5	4	5	5	4,53846154
Afirmación.3: Es fácil aumentar mi experiencia gracias al uso del nuevo sistema	5	5	5	4	3	4	4	4	5	4	3	4	3	4,07692308
Afirmación.4: El nuevo sistema es fácil de utilizar	3	4	5	5	5	5	4	5	5	5	5	5		4,30769231
Afirmación.5: Sería fácil llegar a ser un experto en el manejo del sistema	5	5	5	4	3	4	5	3	4	4	3	4	4	4,07692308
Utilidad Percibida														
Afirmación.6: El uso del nuevo sistema me ayudaría a atender mejor el tema de seguridad informática	5	3	4	5	5	4	5	3	5	5	5	5	4	4,46153846
Afirmación.7: El uso del nuevo sistema mejoraría mi forma de aprender la seguridad informática	4	4	4	4	4	4	5	3	4	4	4	4	4	4
Afirmación.8: El uso del nuevo sistema aumenta mi productividad en el momento de implementar sistemas con seguridad	5	4	5	5	4	5	5	5	5	4	5	3	4	4,53846154

informática															
Afirmación.9: El uso del nuevo sistema incrementa la efectividad con que los estudiantes aprenden el tema de la seguridad informática	5	3	4	4	5	3	4	5	5	5	5	5	5	4	4,38461538
Afirmación.10: El nuevo sistema es de utilidad para los estudiantes que estudian el tema de la seguridad informática	4	4	4	4	4	5	5	4	3	4	4	4	4	4	4,07692308
Actitud hacia el Uso															
Afirmación.11: Usar el nuevo sistema es una buena idea	5	5	7	8	5	4	4	3	4	4	4	5	4	4,76923077	
Afirmación.12: Usar el nuevo sistema es una inteligente idea	3	4	5	3	4	4	4	5	5	4	5	5	5	4,30769231	
Afirmación.13: Me gusta la idea de usar el nuevo sistema	4	4	4	5	5	5	4	3	4	5	3	4	4	4,15384615	
Afirmación.14: Usar el nuevo sistema me parece (placentero/no-placentero)	4	5	3	4	5	5	4	5	4	4	5	5	5	4,46153846	

A nivel general se puede apreciar que la evaluación de aceptación tecnológica es alta en el sentido que tanto la facilidad de uso, la utilidad percibida y la actitud de uso del sistema están por encima del valor 4, el cual corresponde con una evaluación "Muy de acuerdo". Esto permite reflejar una potencial aplicación del sistema en distintos ámbitos. Sin embargo teniendo en cuenta que la evaluación fué aplicada a profesionales que trabajan en área relacionadas con las tecnologías de la información, se puede afirmar que en primera instancia el potencial uso se haría en empresas de desarrollo tecnológico.

En forma particular se hace necesario desglosar cada uno de los aspectos de la evaluación ATAM. En primera medida, de los tres aspectos relevantes del modelo ATAM, el aspecto que tuvo la mejor evaluación fue el de actitud de uso, con una evaluación de 4.42, esto permite observar que aunque los usuarios la primera vez que interactuaron con el sistema algunas cosas no las comprendieron y en un principio no vieron la necesidad de usar el sistema, reconocen que es necesario impulsar este tipo de ideas y tienen una actitud de uso tal alta que está por encima de la facilidad de uso y de la utilidad percibida en este momento.



Es importante anotar que la utilidad percibida es una evaluación inmediata, en donde el usuario entrega su opinión sobre la utilidad que tendría el sistema en el corto plazo. Dado que esta evaluación estuvo por debajo de la evaluación de actitud hacia el uso, se puede afirmar categóricamente que los problemas percibidos por el usuario para la implementación inmediata del sistema, no quitan la visión de necesidad que ven los usuarios en el largo plazo de implementar el sistema.

Adicionalmente, se tiene que la evaluación entre la facilidad de uso estuvo muy parecida a la evaluación de utilidad percibida, lo cual indica que potencialmente el grado con el que se construya un sistema usable, afecta en forma directa la forma como el usuario ve su potencial aplicación en el corto plazo. Esto es totalmente coherente en el sentido que en el grado en que los usuarios vean que el sistema se puede usar, en este grado es que se puede aplicar en un momento dado. Dado que las dos evaluaciones en cuestión estuvieron por debajo de la evaluación de actitud hacia el uso, demuestra que los usuarios tienen muchas expectativas de mejora del sistema que pueden proponer a fin de mejorar su opinión de facilidad de uso y de utilidad percibida.

Otro aspecto importante es la relación entre los tres aspectos más relevantes del modelo ATAM y los resultados obtenidos en el caso del sistema desarrollado, se puede concluir que los usuarios estuvieron al frente de un sistema que en general les pareció usable y necesario en el corto plazo, pero que ven que no se puede implementar a menos que se hagan pequeños ajustes al sistema. Se puede

concluir que los ajustes que exige la comunidad de usuarios del sistema son pequeños, en el sentido que tanto la facilidad de uso y la utilidad percibida están por encima de cuatro, pero no llegan a ser cinco.

Finalmente, por encima de los problemas encontrados, los usuarios esperan que este tipo de desarrollos no se dejen de lado y no se abandonen y esto se refleja cuando entregan una opinión que está por encima de la actitud hacia el uso, en donde se percibe que ellos reconocen que al implementar este tipo de sistemas, las empresas de tecnología podrían ganar valor agregado y sería una buena idea que haría en un futuro el desarrollo de aplicaciones un trabajo más placentero.

CONCLUSIONES

La mayoría de usuarios son conscientes de la importancia de implementar sistemas que cumplan requerimientos no funcionales como por ejemplo el de disponibilidad, sin embargo al evaluar el software, los usuarios se guían más por el cumplimiento de los requerimientos funcionales.

Desarrollar un sistema que cumpla unas metas arquitectónicas permite concentrarse en mejorar la arquitectura de las aplicaciones web y ayuda mucho a la visualización de problemas a priori al desarrollo del proyecto ya a priori a su construcción. Pero lo más interesante es que hasta la misma selección de los frameworks de desarrollo se ve beneficiada por el enfoque top down basado en el refinamiento de una meta arquitectónica.

BIBLIOGRAFIA

I. T. d. Matehuala, «PROGRAMACION WEB,» 05 06 2015. [En línea]. <<https://programacionwebisc.wordpress.com/2-1-arquitectura-de-las-aplicaciones-web/>> [Citado en: 11 de agosto de 2017].

RA-M, «Conceptos generales de la arquitectura de aplicaciones web,» de Implantación de aplicaciones web, p. 11.

S. Mehdi and N. E. Zarour, "Composition of web services using multi agent based planning with high availability of web services," 2016 2nd International Conference on Advanced Technologies for Signal and Image Processing (ATSIP), Monastir, 2016, pp. 10-15.

PHP, «PHP,» [En línea]. <<http://php.net/manual/es/intro-whatcando.php>> [Citado en: 11 de agosto de 2017].

REICEK, «Platzi,» 2015. [En línea]. <<https://platzi.com/blog/laravel-framework-php/>> [Citado en: 11 de agosto de 2017].

EcuRed, «ecured,» [En línea]. <https://www.ecured.cu/Servidor_Web> [Citado en: 13 de agosto de 2017].

P. Mendoza B-A. Galvis P AMBIENTES VIRTUALES DE APRENDIZAJE: UNA METODOLOGÍA PARA SU CREACIÓN
http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-106223_archivo.pdf

Y. Hui and H. Haiyan, "A Study on Data Storage System Based on High-Availability Open Virtual Experiment Platform," 2016 International Conference on Robots & Intelligent System (ICRIS), Zhangjiajie, 2016, pp. 55-58.

Computer Associates. "Computer Associates anuncia una solución que confiere una alta disponibilidad a los servidores Windows NT para aplicaciones de misión crítica".[En línea]

<<http://www.cwordl.cl/archivos/ed179/reportaje.html>> [Citado en: 13 de agosto de 2017].

Farley, M., &Stearns, T., & Hsu, J., Seguridad e Integridad de datos. España. Osborne/McGraw-Hill.

TELEMATIQUE, «TELEMATIQUE, VOLUMEN 2,» [En línea]. <<http://publicaciones.urbe.edu/index.php/telematique/article/viewArticle/758/1817>> [Citado en: 13 de agosto de 2017].

ProcesosdeSoftware, «procesosdesoftware.wikispaces.com,» [En línea]. <<https://procesosdesoftware.wikispaces.com/METODOLOGIA+RUP>.> [Citado en: 13 de agosto de 2017].

Á. Araujo, J. Lázaro, A. Astarloa, A. Zuloaga and J. I. Gárate, "PRP and HSR for High Availability Networks in Power Utility Automation: A Method for Redundant Frames Discarding," in IEEE Transactions on Smart Grid, vol. 6, no. 5, pp. 2325-2332, [Citado en: 13 de agosto de 2017].

R. K. Litvyak, S. P. Vorobyev and A. A. Katsupeev, "The statement and features of phase space of reliability research problem of high availability information system based on architecture with servers redundancy," 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Chelyabinsk, 2016, pp. 1-6. [Citado en: 08 de agosto de 2017].