**Documento de Arquitectura**

PROYECTO: “Sistema adaptativo de formación educativa (SAFE)”

Integrantes – Año 2016

|  |  |  |
| --- | --- | --- |
| Legajo | Nombre | E-Mail |
| 1173388 | Santiago, Peralta | [speralta83@gmail.com](mailto:speralta83@gmail.com) |
| 1202388 | Obregon, Juan Facundo | [facundo.obregon@safabox.com](mailto:facundo.obregon@safabox.com) |
| 1223161 | Montañez, Cinthia | [cinthiamontaez@gmail.com](mailto:cintiamontaez@gmail.com) |
| [1](mailto:cintiamontaez@gmail.com)237937 | Caro, Jonatan | [joni1087@gmail.com](mailto:joni1087@gmail.com) |
| 1134115 | Matsui, Gerardo | [gerardo.matsui@gmail.com](mailto:gerardo.matsui@gmail.com) |

Profesores:

***Director de Cátedra:*** *Dra. Inés Casanovas*

***Profesor a cargo del curso:*** *Ing.*

***Profesor a cargo del proyecto:*** *Lic. Silvia Balduzzi - Ing. Pablo Abramowicz*

***Controller:*** *Mag.Ing. Gabriela Salem*

Contenido

[Historial de Revisión 3](#_Toc454554485)

[1. Objetivo 4](#_Toc454554486)

[2. Diagrama de Contexto 4](#_Toc454554487)

[3. Diagrama de Arquitectura 5](#_Toc454554488)

[**3.1.** **Arquitectura General** 5](#_Toc454554489)

[**3.2.** **Arquitectura de Infraestructura** 6](#_Toc454554490)

[Tecnologías 6](#_Toc454554491)

[4. Atributos de calidad – Requisitos No Funcionales 7](#_Toc454554492)

[5. Tácticas para garantizar RNF 8](#_Toc454554493)

[6. Aprobaciones 11](#_Toc454554494)

Historial de Revisión

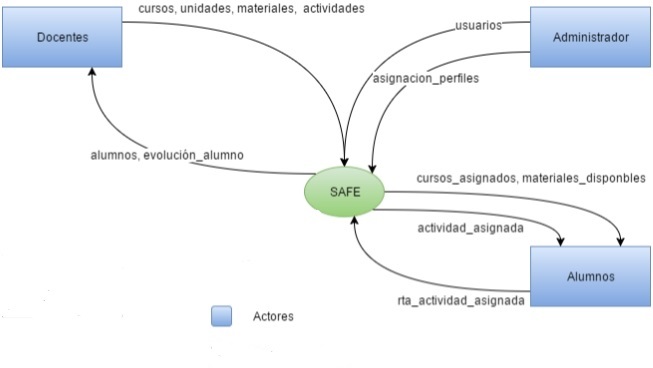
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Fecha | Versión | Descripción | Rol | Autor |
| 03/06/2016 | 1.0 | Creación del documento. | Arquitecto de Software | Gerardo Matsui |
| 10/06/2016 | 1.1 | Corrección del diagrama de contexto, diagrama de arquitectura y patrones de diseño.  Definición de la arquitectura de infraestructura.  Agregado de RNF, tácticas de RNF, métodos de integración y patrones. | Arquitecto de Software | Gerardo Matsui |
| 24/06/2016 | 1.2 | Correcciones al documento. | Arquitecto de Software | Gerardo Matsui |
| 26/06/2016 | 1.3 | Correcciones en diagramas y descripciones | Arquitecto de Software | Gerardo Matsui |
|  |  |  |  |  |

# Objetivo

Este documento describe la arquitectura general de la solución implementada en el proyecto, los componentes individuales que componen la solución y la forma en que interactúan.

# Diagrama de Contexto

El siguiente diagrama muestra en un alto nivel de detalle, los entes y entidades que interactúan con el sistema, determinando los límites y el ambiente del mismo.



# Diagrama de Arquitectura

## **Arquitectura General**

El sistema esta comprendido por un grupo de módulo, cada uno con su respectivas funcionalidades y responsabilidades.



**Módulo de Perfiles:** Gestiona a los usuarios del sistema.

* Alta, baja y modificación de los usuarios del sistema.
* Asignación del perfil del usuario: define el tipo de perfil que tendrá un usuario pueden ser del tipo Alumno, Docente o Administrador.
* Autorización: Permite/deniega determinadas operaciones de acuerdo al perfil del usuario.

**Módulo de Alumnos:** Gestiona la información referente a los cursos y actividades que realizan los alumnos

* Cursos en los que se encuentra asignado el alumno.
* Las actividades a realizar por el alumno.
* El avance en las actividades realizado por el alumno.

**Módulo de Docentes:** Administra la información de los cursos que imparte y el avance de los alumnos en cada curso.

* Cursos impartidos por el docente.
* Alumnos asignados en cada curso.
* El avance de cada alumno del curso.

**Módulo de Cursos:** Gestiona la información referente a los cursos.

* Alta, baja y modificación de los cursos.
* Definición del temario del curso.
* Dependencias entre cada tema.

**Módulo de Actividades:** Gestiona la información referente a las actividades, la misma pueden ser de carácter teórico, práctico o ambas.

* Alta, baja y modificación de actividades.

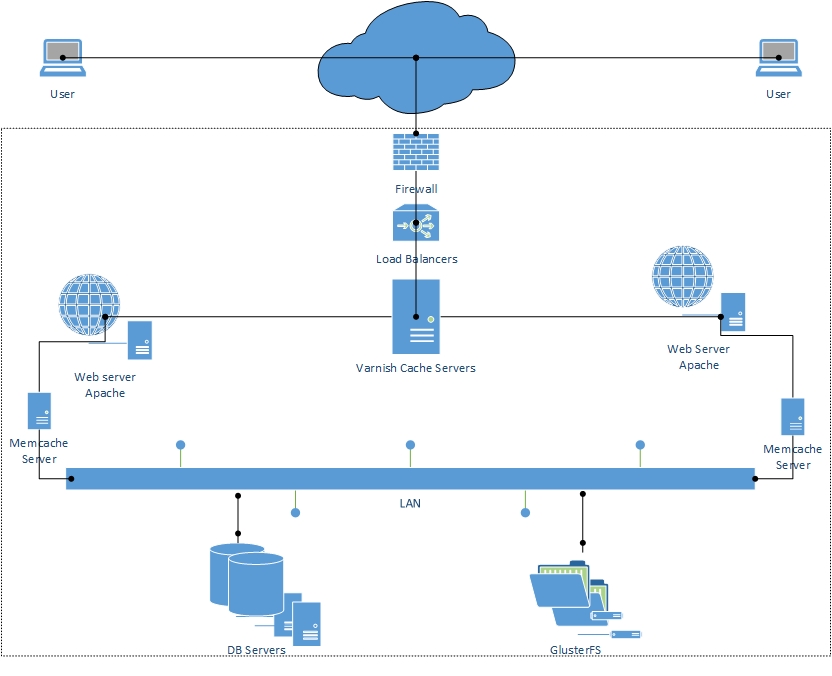
**Módulo de Autoaprendizaje:** Es el encargado de registrar los progresos de cada alumno y determinar en base a la información recolectada la siguiente actividad a realizar por el alumno.

## **Arquitectura de Infraestructura**

La arquitectura es la de una plataforma web la cual dispone de un firewall para limitar el acceso a la plataforma desde ip´s identificadas y conocidas. A su vez posee un balanceador de carga que permite balancear la carga entre diferentes nodos para garantizar la capacidad de procesamiento y confiabilidad. También asegura la disponibilidad de las aplicaciones monitorizando el estado de las misma.

Para mejorar las prestaciones de la aplicación se provee dos niveles de caché, uno a nivel de aplicación (Varnish) y otro a nivel acceso a datos (MemCached).

A nivel de persistencia de archivos, la aplicación utiliza un sistema de archivo en red (Gluster) lo que nos permite mantener las instancias de la aplicación clusterizada.



### Tecnologías

|  |  |
| --- | --- |
| Tecnología | Descripción |
| Web Server | Servidores web apache. |
| Data Base Servers | Servidores de bases de datos. |
| File Servers | GlusterFS |
| Cache Server | Varnish cache servers, Memcache servers |
| Symfony 2.8 | Framework de aplicaciones PHP |
| Bootstrap | Framework html/css/js. |
| AngularJs | Framework JS |

# Atributos de calidad – Requisitos No Funcionales

|  |  |
| --- | --- |
| ***Compatibilidad*** | |
| ***Estímulo*** | *Origen del Estímulo* | | *Alumno/Docente/Administrador* |
| *Estímulo* | | *Acceso al Sitio con navegadores soportados*  [http://www.henrykszeryng.net/graphx/logo_firefox.gif](http://www.mozilla-europe.org/fr/firefox/)**Firefox 31** o superior  [http://www.henrykszeryng.net/graphx/logo_safari.gif](http://www.apple.com/es/safari/download/)**Safari 5.0.6** o superior  [http://www.henrykszeryng.net/graphx/logo_chrome.gif](http://www.google.com/chrome/?hl=es)**Google Chrome 31** o superior  [http://www.henrykszeryng.net/graphx/logo_internet_explorer.gif](http://www.microsoft.com/downloads/Search.aspx?displaylang=fr)**Internet Explorer 9** o superior |
| ***Ambiente*** | *Ambiente* | | *-* |
| *Componente* | | *Capa de presentación* |
| ***Respuesta*** | *Respuesta* | | *El sitio debe informar que el navegador utilizado no es soportado por el sistema* |
| *Medida de Respuesta* | | *Disponible siempre que se detecte el evento* |

|  |  |
| --- | --- |
| ***Portabilidad*** | |
| ***Estímulo*** | *Origen del Estímulo* | | *Implementador* |
| *Estímulo* | | *Implementación del sistema* |
| ***Ambiente*** | *Ambiente* | | *Al momento de instalar el sistema* |
| *Componente* | | *Sistema* |
| ***Respuesta*** | *Respuesta* | | *El aplicativo debe permitir su instalación en plataforma Windows y GNU/Linux* |
| *Medida de Respuesta* | | *Cada implementación* |

|  |  |
| --- | --- |
| ***Escalabilidad Horizontal*** | |
| ***Estímulo*** | *Origen del Estímulo* | | *Alto volumen de carga* |
| *Estímulo* | | *Mayor capacidad de procesamiento* |
| ***Ambiente*** | *Ambiente* | | *Crecimiento en el volumen de usuarios en el sistema* |
| *Componente* | | *Sistema* |
| ***Respuesta*** | *Respuesta* | | *El aplicativo debe permitir funcionar con otras instancias en paralelo* |
| *Medida de Respuesta* | | *420 request por segundo* |

# Tácticas para garantizar RNF

A continuación, se detallan las tácticas para lograr los atributos de calidad.

* **Táctica para garantizar la compatibilidad**: Identificar el tipo y versión del navegador en el Request al sistema. Se agregará un algoritmo que obtendrá el tipo y versión del navegador que está siendo utilizado por el usuario. En caso de que el navegador no sea soportado por el sistema se le mostrará un cartel con el listado de navegadores soportados.
* **Táctica para portabilidad**: Utilizar recursos disponibles en ambas plataformas.

Para el desarrollo del sistema solo se utilizarán recursos (lenguaje, intérpretes, librerías, etc) disponibles en las plataformas GNU/Linux y Windows.

* **Escalabilidad horizontal**: Soporte para integrar un balanceador de carga.

Se desarrollará el sistema con capacidad de anteponerle un balanceador de carga y ejecutar varias instancias del sistema.

Mecanismos de integración

Las integraciones entre los sistemas se realizarán mediante HTTP utilizando un API REST y TCP/IP para los accesos a la BBDD.

Patrones de diseño y de Arquitectura

Los patrones que se utilizarán serán:

* **MVC (Modelo – Vista - Controlador):** Para la interfaz del usuario.
* **Arquitectura por capas (Interfaz, Servicio y Persistencia):** Para la aplicación montada en el servidor de aplicaciones.



# Aprobaciones

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Fecha:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Aprobación Técnica de la Arquitectura

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** Fecha:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Profesor a cargo del Proyecto