Taller de Informática

Integrantes:

Fredy Mendivelso

Freddy Yesid Ospina

Marco Tulio Guzmán

Víctor De Hoyos

**Robinson de la Asunción**

Profesor

Paulo Coronado

UNIVERSIDAD DISTRITAL FRANCISCO JOSE DE CALDAS

FACULTAD DE INGENIERÍA

ESPECIALIZACION EN INGENIERIA DE SOFTWARE

BOGOTÁ, OCTUBRE 2 DE 2014

REGISTRATION APP

Visión

# Introducción

Dentro del mundo de las aplicaciones, es indispensable la seguridad de las apps. Se enmarca en este documento la especificación de una herramienta que expresa las vulnerabilidades que puedan existir en el mundo del software.

# Posicionamiento

Se posiciona en un ambiente de desarrollo php, con un framework symfony, lo cual posee una variedad de componente.

## Problemática

|  |  |
| --- | --- |
| El problema de la | *En la actualidad, el manejo de la seguridad en las aplicaciones web es muy vulnerable.* |
| Afecta | *Usuarios finales, administradores del sistemas, desarrolladores* |
| Cuyo impacto es | *Generar aplicaciones seguras.* |
| Una solución exitosa seria | *Generar herramientas o aplicaciones web que tengan un nivel de seguridad optimo* |

## 

# Stakeholder Descriptions

|  |  |  |
| --- | --- | --- |
| **Name** | **Description** | **Responsibilities** |
| Usuarios Finales  Desarrolladores  Administradores del sistemas | *Encargados en acceder al aplicativos*  *Son los que se encargan del desarrollo de la aplicacion web*  *Son los encargados de configurar el aplicativo y sus dependencias con el framework* | *De realizar el registro en el aplicativo*  *De desarrollar adecuadamente el aplicativo*  *De realizar los versionamientos del desarrollo* |

# Product Overview

## Needs and Features

*[Avoid design. Keep feature descriptions at a general level. Focus on capabilities needed and why (not how) they should be implemented. Capture the stakeholder priority and planned release for each feature.]*

|  |  |  |  |
| --- | --- | --- | --- |
| **Need** | **Priority** | **Features** | **Planned Release** |
|  |  |  |  |

# Other Product Requirements

*[At a high level, list applicable standards, hardware, or platform requirements; performance requirements; and environmental requirements.*

*Define the quality ranges for performance, robustness, fault tolerance, usability, and similar characteristics that are not captured in the Feature Set.*

*Note any design constraints, external constraints, assumptions or other dependencies that, if changed, will alter the* ***Vision*** *document. For example, an assumption may state that a specific operating system will be available for the hardware designated for the software product. If the operating system is not available, the* ***Vision*** *document will need to change.*

*Define any specific documentation requirements, including user manuals, online help, installation, labeling, and packaging requirements.*

*Define the priority of these other product requirements. Include, if useful, attributes such as stability, benefit, effort, and risk.]*

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Priority** | **Planned Release** |
|  |  |  |

**ARQUITECTURA**

El modelo describe la arquitectura de software del sistema a través de 5 vistas concurrentes Kruchten las agrupa estas 5 vistas por su naturaleza en 3 apartados; el conceptual donde sitúa a la vista lógica y la de procesos, el físico que lo componen las vistas de componentes y la distribuida y por último la funcional la que se refiere a la vista de casos de uso.

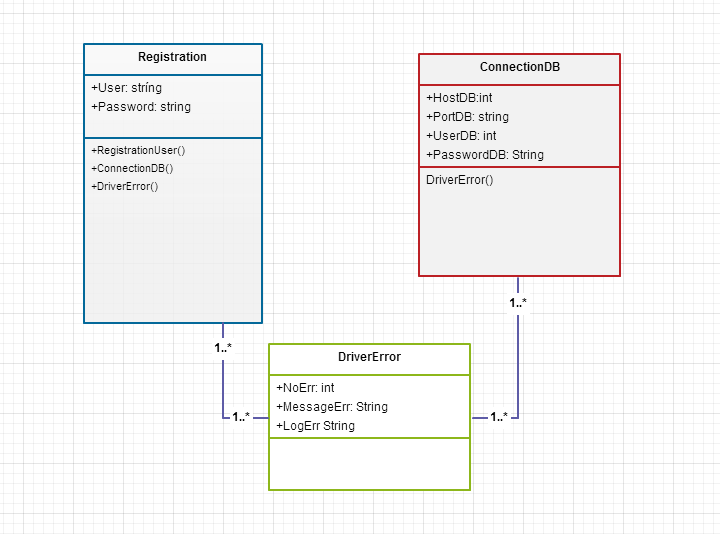
**Diagramas de Casos de Usos**

<Poner una breve descripción y su imagen del diagrama UML> Yesid Ospina

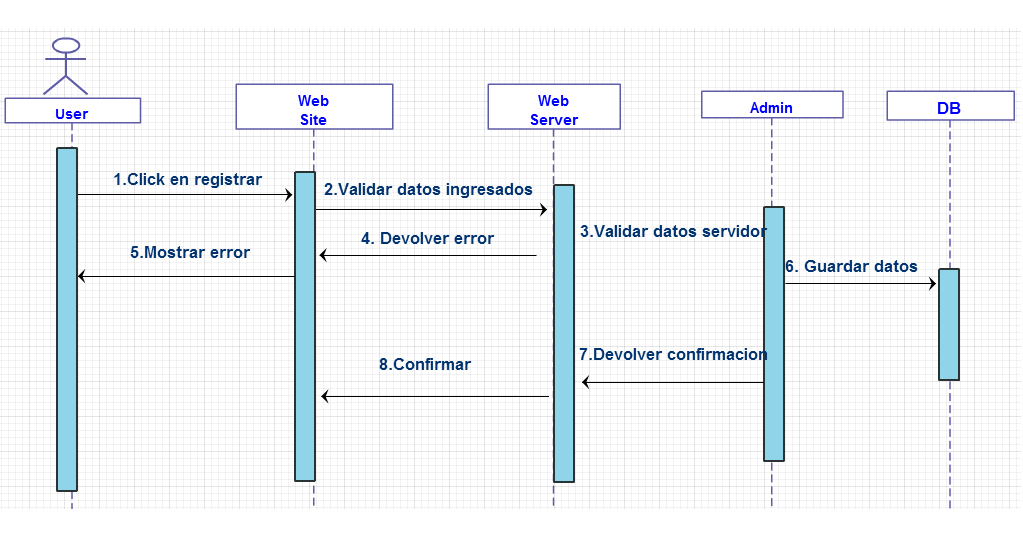
**Vista Lógica**

El usuario para el registro debe suministrar la información correspondiente a su nombre de usuario y la clave. Se debe tener en cuenta que el nombre de usuario no se puede repetir en la base de datos, la clave no puede superar los mil (1000) caracteres. Al hacer click en el botón registrar el sistema hace las validaciones del lado cliente haciendo uso de javascript, si estas validaciones son exitosas, se envía la petición al servidor para ser procesada. El servidor nuevamente hace las validaciones en casos de éxito, los datos son guardados en la base de datos. En caso de error, para ambos casos (Cliente y servidor), se devuelve un mensaje al usuario notificando la validación con errores.

**Diagrama de clases**



**Diagrama de secuencia**



**Vista Procesos**

<Poner una breve descripción y su imagen del diagrama UML> Mendivelso

**Vista de Implementación**

<Poner una breve descripción y su imagen del diagrama UML> Víctor

**Vista de Despliegues**

<Poner una breve descripción y su imagen del diagrama UML> Víctor