DOCUMENTO INGENIERÍA DEL SOFTWARE

DEL PROYECTO: “Web MPI”

DANIEL DIAZ GIRALDO

ANDERSON ALBERTO OCHOA

UNIVERSIDAD TECNOLÓGICA DE PEREIRA

INGENIERÍA DE SISTEMAS Y COMPUTACIÓN

PEREIRA

01/02/2016

Tabla de contenido

[1. Introducción 4](#_Toc442668507)

[1.1. Propósito 4](#_Toc442668508)

[1.2. Alcance 4](#_Toc442668509)

[1.3. Definiciones, siglas y abreviaturas 5](#_Toc442668510)

[1.4. Referencias 8](#_Toc442668511)

[1.5. Visión General 8](#_Toc442668512)

[2. Vista global del producto 9](#_Toc442668513)

[2.1 Objetivos del Negocio 9](#_Toc442668514)

[2.2. Perspectiva del producto 9](#_Toc442668515)

[2.3. Supuestos y dependencias 9](#_Toc442668516)

[3. Características del producto 11](#_Toc442668517)

[3.2. Características principales 11](#_Toc442668518)

[3.3. Otras características 12](#_Toc442668519)

[4. Usuarios (perfiles de interesados y perfiles de involucrados) 13](#_Toc442668520)

[4.2. Limitaciones 13](#_Toc442668521)

[4.3. Alcance 13](#_Toc442668522)

[5. Fase de requerimientos y análisis del desarrollo 14](#_Toc442668523)

[5.2. Historias de usuario 14](#_Toc442668524)

[5.3. Diagramas de casos de uso (detallado) 18](#_Toc442668525)

[5.4. Diagrama de secuencia o actividades detallado (por cada escenario) 19](#_Toc442668526)

[5.5. Fases de diseño 36](#_Toc442668527)

[5.5.1. Diagrama de base de datos No relacional 36](#_Toc442668528)

[5.5.2. Diagrama de componentes 37](#_Toc442668529)

[5.5.3. Diagrama de despliegue 38](#_Toc442668530)

[5.6. Fase de implementación 39](#_Toc442668531)

[5.6.1. Plantilla de estándar de codificación: 39](#_Toc442668532)

[Guía de estilo. Nomenclatura en archivos Javascript 39](#_Toc442668533)

[“1. General 39](#_Toc442668534)

[2. Nombres de Fichero 39](#_Toc442668535)

[3. Objeto Global. 40](#_Toc442668536)

[4. Sub-objetos globales. 40](#_Toc442668537)

[5. Variables. 40](#_Toc442668538)

[6. Constructores. 41](#_Toc442668539)

[7. Constantes. 41](#_Toc442668540)

[8. Propiedades. 41](#_Toc442668541)

[9. Métodos. 42](#_Toc442668542)

[5.6.2. Lenguaje de programación 42](#_Toc442668543)

[5.6.3. IDE de desarrollo 43](#_Toc442668544)

[5.6.4. Motor de bases de datos 43](#_Toc442668545)

[5.6.5. Framework 44](#_Toc442668546)

[5.7. Fase de pruebas 44](#_Toc442668547)

[5.7.1. Herramientas para pruebas 45](#_Toc442668548)

[5.7.2. Análisis de los resultados de pruebas. 45](#_Toc442668549)

[6. Conclusiones. 46](#_Toc442668550)

[7. Bibliografía. 47](#_Toc442668551)

Lista de ilustraciones

[Ilustración 1 18](#_Toc442668552)

[Ilustración 2 20](#_Toc442668553)

[Ilustración 3 21](#_Toc442668554)

[Ilustración 4 23](#_Toc442668555)

[Ilustración 5 24](#_Toc442668556)

[Ilustración 6 26](#_Toc442668557)

[Ilustración 7 28](#_Toc442668558)

[Ilustración 8 29](#_Toc442668559)

[Ilustración 9 30](#_Toc442668560)

[Ilustración 10 32](#_Toc442668561)

[Ilustración 11 33](#_Toc442668562)

[Ilustración 12 35](#_Toc442668563)

[Ilustración 13 36](#_Toc442668564)

[Ilustración 14 37](#_Toc442668565)

[Ilustración 15 38](#_Toc442668566)

Lista de Anexos

* Anexo documento “FormatoPruebasWebmpi.xls”

# Introducción

En cada proyecto de software es de suma importancia tener un sustento conceptual de lo que el software representa a nivel lógico y arquitectónico, para que el cliente sepa de manera temprana cuáles son los cimientos de todo el esquema de software.

En el presente documento se pondrá a modo de exposición los componentes esenciales comprendidos por la ingeniería de software dentro del proyecto de grado denominado “implementación de un prototipo de plataforma tecnológica tipo cluster, para la computación de algoritmos realizados en mpi presentados en el campus universitario.”, indispensable para la elaboración de un software con la calidad y compromiso planeado.

## Propósito

El propósito del contenido aquí expuesto es consolidar las técnicas que propone el área de la ingeniería de software, intentando plasmar de manera ilustrativa y organizada los puntos claves del desarrollo del proyecto de grado respectivo.

Se tienen en cuenta los aspectos funcionales detallados que son importantes en la elaboración del producto. El documento se generó gracias al proceso de calidad que se viene presentando a lo largo del desarrollo de la investigación presentada en el grupo de investigación Sirius.

Siendo más puntual se podrá ver de manera más cómoda los esquemas que acompañarán a la elaboración de la web del proyecto.

## Alcance

Con el desarrollo de esta aplicación se pretende lograr que los usuarios tengan una herramienta a la mano que les permita tener una interacción agradable referente a la plataforma, ocultando la capa de asegurando así el principio de transparencia usuario-sistema distribuido.

Mostrar información ordenada tanto del proceso de compilación como el de ejecución, haciendo más fácil la programación a nivel distribuido.

Además esta aplicación web servirá para administrar usuarios, ver sus procesos activos, ver los recursos empleados, supervisar las ejecuciones de código, a su vez se podrá autorizar a profesores y estudiantes para acceder a la plataforma.

El sistema estará diseñado bajo ciertas restricciones de visualización de contenido y ejecución de acciones de acuerdo con el rol que desempeñe el usuario, esto con el fin de brindarle al usuario la información necesaria que debe saber y permitirle realizar solo las acciones que debe realizar.

Con el constante avance en tecnología y el aumento evidente del uso de smartphones, también se podrá acceder al sistema mediante tabletas, con el fin de facilitar al usuario el acceso al sistema, además de fomentar y difundir el uso de las tecnologías.

## Definiciones, siglas y abreviaturas

**Alojamiento web:** (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

**Arquitectura de software:** Las técnicas metodológicas desarrolladas con el fin de facilitar la programación se engloban dentro de la llamada Arquitectura de Software o Arquitectura lógica. Se refiere a un grupo de abstracciones y patrones que nos brindan un esquema de referencia útil para guiarnos en el desarrollo de software dentro de un sistema informático. Así, los programadores, diseñadores, ingenieros y analistas pueden trabajar bajo una línea común que les posibilite la compatibilidad necesaria para lograr el objetivo deseado. Algunos objetivos dentro de un esquema de Arquitectura de Software pueden ser: el software debe ser mantenible, esto es, fácilmente analizable, modificable, corregible; también puede ser un objetivo el nivel de interacción con otros sistemas informáticos, o su escalabilidad.

**Framework**:La palabra inglesa "framework" (marco de trabajo) define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular que sirve como referencia, para enfrentar y resolver nuevos problemas de índole similar. En el desarrollo de software, un framework o infraestructura digital, es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software. Típicamente, puede incluir soporte de programas, bibliotecas, y un lenguaje interpretado, entre otras herramientas, para así ayudar a desarrollar y unir los diferentes componentes de un proyecto.

**Hosting:** El alojamiento web (en inglés web hosting) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web. Es una analogía de "hospedaje o alojamiento en hoteles o habitaciones" donde uno ocupa un lugar específico, en este caso la analogía alojamiento web o alojamiento de páginas web, se refiere al lugar que ocupa una página web, sitio web, sistema, correo electrónico, archivos etc. en internet o más específicamente en un servidor que por lo general hospeda varias aplicaciones o páginas web.

**Ingeniería de software:** es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques, es decir, la aplicación de la ingeniería al software.1 Es la aplicación de la ingeniería al software, ya que integra matemáticas, ciencias de la computación y prácticas cuyos orígenes se encuentran en la ingeniería.

**Navegador:** Un navegador o navegador web, o browser, es un software que permite el acceso a Internet, interpretando la información de archivos y sitios web para que éstos puedan ser leídos.

**Owasp**: El proyecto OWASP (Open Web Application Security Project) es un proyecto abierto dedicado a la seguridad en aplicaciones web. Consiste en una comunidad a nivel mundial enfocada en la mejora de la seguridad en las aplicaciones de software. OWASP está organizada en proyectos y capítulos locales repartidos por todo el mundo dedicados al desarrollo de documentación, herramientas y estándares de código libre, además, está abierta a la participación de cualquier persona y/o patrocinador.

**Smartphone:** Un teléfono inteligente (smartphone en inglés) es un teléfono móvil construido sobre una plataforma informática móvil, con una mayor capacidad de almacenar datos y realizar actividades semejantes a una minicomputadora y conectividad que un teléfono móvil convencional.

**Servidor web:** Se trata de un programa que implementa el protocolo HTTP (HyperText Transfer Protocol). Este protocolo está diseñado para transferir lo que llamamos hipertextos, páginas web o páginas HTML: textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de música.

**Tablet:** en muchos lugares también llamada tablet (del inglés: tablet o tabletcomputer),1 2 es una computadora portátil de mayor tamaño que un teléfono inteligente o una PDA, integrada en una pantalla táctil (sencilla o multitáctil) con la que se interactúa primariamente con los dedos o un estilete (pasivo o activo), sin necesidad de teclado físico ni ratón.

**TIC:** Conjunto de recursos, procedimientos y técnicas usadas en el procesamiento, almacenamiento y transmisión de información, esta definición se ha matizado de la mano de las TIC, pues en la actualidad no basta con hablar de una computadora cuando se hace referencia al procesamiento de la información.

**W3C:** El World Wide Web Consortium, abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web.Fue creado en octubre de 19941 y está dirigida por Tim Berners-Lee, el creador original de URL (UniformResourceLocator, Localizador Uniforme de Recursos), HTTP (HyperText Transfer Protocol, Protocolo de Transferencia de HiperTexto) y HTML (Hyper Text MarkupLanguage,Lenguaje de Marcado de HiperTexto) queson las principales tecnologías sobre las que se basa la Web.

**URL:** Son las siglas de Localizador de Recurso Uniforme (en inglés UniformResourceLocator), la dirección global de documentos y de otros recursos en la World Wide Web. La primera parte de la dirección indica qué protocolo utilizar, la segunda parte especifica la dirección IP o nombre de dominio donde se localiza el recurso.

## Referencias

|  |  |  |
| --- | --- | --- |
| **Documento** | **Quien lo elabora** | **Fecha de elaboración** |
| Implementación de un prototipo de plataforma tecnológica tipo cluster, para la computación de algoritmos realizados en mpi presentados en el campus universitario. | Daniel diaz giraldo  Anderson alberto ochoa | 2015 |
| Web Cuda UTP  <http://judge.utp.edu.co:3000/> | Alejandro Suarez | 2015 |
| FAQ MongoDB  https://docs.mongodb.org/manual/ | Comunidad opensource | 2007 - Presente |
| <http://www.w3.org/People/Berners-Lee/> | Tim Berners | 1989 |
| NodeJs Manual and training | Comunidad Nodejs | 2009 - Presente |
| Manuales y formatos, laboratorio de software UTP. | Paula Andrea Villa | 2013 |

## Visión General

El documento de ingeniería de software será el documento de referencia para los entendidos en la materia, dará a conocer los modelos estructurados del proyecto, para que ningún dato se escape. Integrará todos los modelos ya hechos con su mayor nivel de detalle, facilitando la vida al lector y los desarrolladores del proyecto.

# Vista global del producto

### 2.1 Objetivos del Negocio

* ON-1: Hacer uso de las herramientas tecnológicas para desarrollar una interfaz web para la plataforma tipo cluster sustentado en el proyecto de grado referenciado.
* ON-2: Hacer una implementación que maneje múltiples usuarios, con diferentes interfaces.
* ON-3: Realizar una implementación web de bajo costo y buena optimización, que permita la ejecución paralela de varios programas lanzados por diferentes usuarios sin que esto altere el resultado.
* ON-4: Realizar una interfaz web capaz de mostrar los resultados de la plataforma a nivel distribuido en una sola vista.
* ON-5: Hacer una implementación que contenga un panel administrativo para el control eficaz de las tareas lanzadas en la plataforma y la autorización de usuarios.

## Perspectiva del producto

Web MPI es un proyecto de interés académico que guiará y facilitará los procesos para la compilación y ejecución de código, en especial, los realizados con la librería mpi, se visualiza que el proceso sea transparente en el ingresó a la página web, identificando a cada tipo de usuario (Administrador, Usuario). Se espera más que una página web, un sitio del que los usuarios puedan ejecutar algoritmos (c++), con facilidad y de manera organizada, supervisada por los administradores.

La principal función que hará el brindar una utilidad a la hora de ejecutar código distribuido o normal. Tendrá fines educativos e investigativos, todo con el fin de integrar, orientar y facilitar el acceso a toda la comunidad universitaria.

## Supuestos y dependencias

La plataforma a desarrollar trabajará conjuntamente de herramientas de verificación de funcionamiento y optimización establecido por la W3C.

El desarrollo normal del proyecto dependerá de:

* Alcances referente al proyecto de grado anteriormente mencionado.
* Pruebas profundas en un entorno mucho más complejo.
* De los estudiantes para hacer pruebas en la web.
* A los administradores de red, encargados en abrir puertos necesarios para
* El hardware y los métodos que maneje el hosting así como el paquete y alternativas que ofrezca.

Por otro lado, también se deberán hacer los debidos supuestos para tratar de complementar vacíos pertinentes o para complementar puntos del proyecto,

Estos son:

* Suponemos que el proyecto será viable en un entorno más complejo al mencionado en la investigación derivada.
* Suponemos que el sistema no será violentado por informáticos ajenos al proyecto.
* Suponemos que los usuarios además de contar con una computadora también tendrán dispositivos móviles (Smartphone y/o Tablet) para el acceso a la página web.
* Suponemos que los usuarios serán del campus universitario y tendrán conocimientos en programación lógica, c++ y mpi.

Se asumirá por otro lado:

* La elaboración de la plataforma que deberá ser de manera profesional y cumpliendo los requerimientos más básicos de usabilidad y optimización.
* La disponibilidad web será limitada dado los puntos anteriores referentes al hardware y permisos de red.
* Que los participantes de ésta sepan manejar correctamente, sin ningún tipo de problemas.
* Todo será vía web.

# Características del producto

El desarrollo del proyecto Web MPI se llevara a cabo mediante la metodología de desarrollo ágil XP utilizando todos los conocimientos necesarios en Ingeniería del Software, Comunicaciones y Bases de Datos.

Para poder hablar sobre las características proyecto se dará una breve definición del mismo. Con ayuda de las TICs se realizará un sistema que sea capaz de compilar código del famoso lenguaje c++ por medio de un aplicativo de tipo cluster que se podrá acceder a través de una página web que mostrara un canvas básico para escribir código fuente y obtener dichas salidas. Los códigos de cada usuario serán procesados de manera individual y concurrentemente llegado el caso, esta plataforma correrá dentro del hardware facilitado por el grupo de investigación Siruis de la Universidad Tecnológica de Pereira esto permitirá a los estudiantes y académicos computar soluciones distribuidas brindando información relevante para sus proyectos investigativos

## Características principales

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descripción** | **Prioridad**  **(0-5)** | **Objetivo de negocio asociado** |
| CAR-01 | Los usuarios podrán registrarse en el sistema para poder obtenerlos beneficios que esta presenta. | 4 | ON – I  ON – II  ON – III |
| CAR-02 | El sistema registrará a los usuarios con la siguiente información: Nombre de usuario, Email, Apellido, Contraseña, Verificación de Contraseña (dicha contraseña tendrá mínimo 4 caracteres alfanuméricos). | 3 | ON- II |
| CAR-03 | El sistema registra a los usuarios con 2 tipos de rol: Administrador (uno solo), Usuario normal. | 4 | ON – II |
| CAR-04 | Solo el administrador podrá eliminar usuarios de la base de datos. Y será el único que podrá autorizar el uso de cualquier usuario en la plataforma. | 4 | ON – V |
| CAR-05 | Cada usuario podrá recuperar su contraseña, para poder recuperarla debe haber confirmación de correo electrónico. | 2 | ON – II |
| CAR-06 | El usuario normal podrá realizar PQR a través de la plataforma. | 2 | ON – II |
| CAR-07 | El sistema será de código abierto, permitiendo la colaboración para la optimización del código. | 3 | ON – I |
| CAR-08 | El sistema tendrá una reacción de fallos por debajo de los 3 minutos. | 4 | ON – III |
| CAR-09 | El sistema cumplirá con las normas del habeas data. | 4 | ON – III |
| CAR-10 | El sistema seguirá el estándar W3C de accesibilidad y usabilidad. | 2 | ON – III |
| CAR-11 | El sistema cumplirá con los estándares mínimos de seguridad (nivel 1 con el método OWAS). | 2 | ON – III |
| CAR-12 | El sistema no superará un tiempo de respuesta de 4 segundos, por otro lado la respuesta al compilar y ejecutar un código dependerá de la complejidad de este, se establecerá un tiempo de espera en ejecución de 10 horas de top para cada ejecución. | 5 | ON – I  ON – IV |
| CAR-13 | El sistema tendrá un buen manejo de errores y excepciones. | 2 | ON – IV |

## Otras características

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descripción** | **Prioridad** | **Objetivo de negocio asociado** |
| CAR-01 | La plataforma web debe ser flexible y soportar múltiples navegadores | 3 | ON-V |

# Usuarios (perfiles de interesados y perfiles de involucrados)

## Limitaciones

Las limitantes referentes al documento van desde código hasta usabilidad.

El documento contempla la parte arquitectónica del proyecto, para consultar el código fuente acceder a: <https://github.com/aaochoa/webmpi/tree/MPI>. También deja de lado la parte comercial, para entrar a una estructuración del proyecto.

## Alcance

Con del desarrollo del presente documento se pretende afianzar lo que se venía ejecutando en el proyecto de investigación mencionado anteriormente. Sirviendo de base conceptual para cualquier lector.

Dentro del alcance del documento de ingeniería de software se define las acciones capaces de ser desarrolladas por el proyecto en cuestión, gracias a los diagramas esquemáticos que se verán sustentados a lo largo de este documento.

También en las definiciones no limitantes del documento encontramos desde la asignación de perfiles de usuario hasta la lista de actividades en el proceso de codificación.

Por último contendrá modelos de 4+1 y modelos relacionales del producto de software para tener un entendimiento más preciso del sistema.

1. Fase de requerimientos y análisis del desarrollo

## Historias de usuario

La toma de requisitos en contexto fueron acordados con el docente director del proyecto y a base de algunas especulaciones lógicas.

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 1** | **Nombre:** Navegadores | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: única** |
| **Descripción:** Debe ser un software de arquitectura web y los usuarios deben poder acceder a él por Internet, usando navegadores Internet Explorer 10 o superior, Firefox 23 o superior, Google Chrome 28.0 o superior. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 2** | **Nombre:** Interfaz de usuario | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: única** |
| **Descripción:** Cumplir con requisitos mínimos de usabilidad de la W3C. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 3** | **Nombre:** Desempeño | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: única** |
| **Descripción:** Cada petición que realice el usuario debe tener una respuesta del sistema en máximo 4 segundos y debe soportar al menos 40 usuarios conectados de forma simultánea. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 4** | **Nombre:** Disponibilidad | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 1** |
| **Descripción:** Yo como administrador y propietario del programa deseo que el sistema cumpla con una interfaz administrativa limpia y fácil de usar. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 5** | **Nombre:** Seguridad | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 1** |
| **Descripción:** Yo como usuario del producto espero contar con una seguridad básica en formularios y acceso por directorio transversal. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 6** | **Nombre:** Asignación de roles | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 1** |
| **Descripción:** El sistema debe permitir la administración de 3 diferentes tipos de usuarios.  ·         Administrativo (Cuenta única).  ·         Usuario normal.  . Invitado | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 7** | **Nombre:** Registro de usuarios | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 1** |
| **Descripción:** Yo como usuario me puedo registrar con un nombre de usuario, una contraseña y un email para recuperación. Los datos del usuario son: ID(generado por el motor de base de datos), Nombres y apellidos, Correo Electrónico, Contraseña (mínimo 4 caracteres alfanuméricos), Verificar Contraseña. El administrador no requiere registro y está configurado por defecto. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 8** | **Nombre:** Gestión de usuarios | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 1** |
| **Descripción:** El administrador posee la capacidad para atender las PQR´s, Detener ejecuciones de los usuarios. Así como eliminar usuarios. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 9** | **Nombre**: Perfiles de usuario. | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 1** |
| **Descripción:** Cada usuario podrá realizar PQR´s, cambiar su contraseña y recuperarla, petición de eliminación de cuenta. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 10** | **Nombre:** Autenticación. | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 1** |
| **Descripción:** Yo como usuario (Administrador, normal) deseo hacer login por medio de autentificación de nombre de usuario y contraseña para acceder a la aplicación “Web MPI”. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 11** | **Nombre:** Perfil Administrativo. | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 2** |
| **Descripción:** Yo como Administrador, podré ver los recursos consumidos por cada proceso de ejecución, ver los usuarios que quieren ser parte de la plataforma, ver las pqrs y resolverlas, entre otras vistas. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 12** | **Nombre:** Perfil Normal | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 2** |
| **Descripción:** El perfil normal dispone de un cuadro de texto para introducir su código de programación y otro cuadro de información donde saldrá el resultado de su cómputo, podrá computar algoritmos escritos en c++. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 13** | **Nombre:** visitante | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 2** |
| **Descripción:** Un visitante solo podrá ver la página principal de la web, pero podrá pedir petición para registrarse. | | |
| **Observaciones:** | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 14** | **Nombre:** Códigos en mpi | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 2** |
| **Descripción:** implementa el uso de códigos en mpi que puedan ser interpretados por la plataforma de forma que el usuario pueda ver la el resultado de manera transparente. | | |
| **Observaciones:** Otorgara transparencia y practicidad al usuario. | | |

|  |  |  |
| --- | --- | --- |
| **Historia de Usuario** | | |
| **Número: 15** | **Nombre:** Re direccionamiento a sitio | |
| **Prioridad en Negocio: 1** | | **Iteración Asignada: 2** |
| **Descripción:** Yo como usuario deseo poder acceder a la aplicación “web mpi” desde cualquier dispositivo que cuente con un navegador web actual he internet. | | |
| **Observaciones:** | | |

## Diagramas de casos de uso (detallado)

**Caso de los usuarios Administrador, Normal e invitado:**

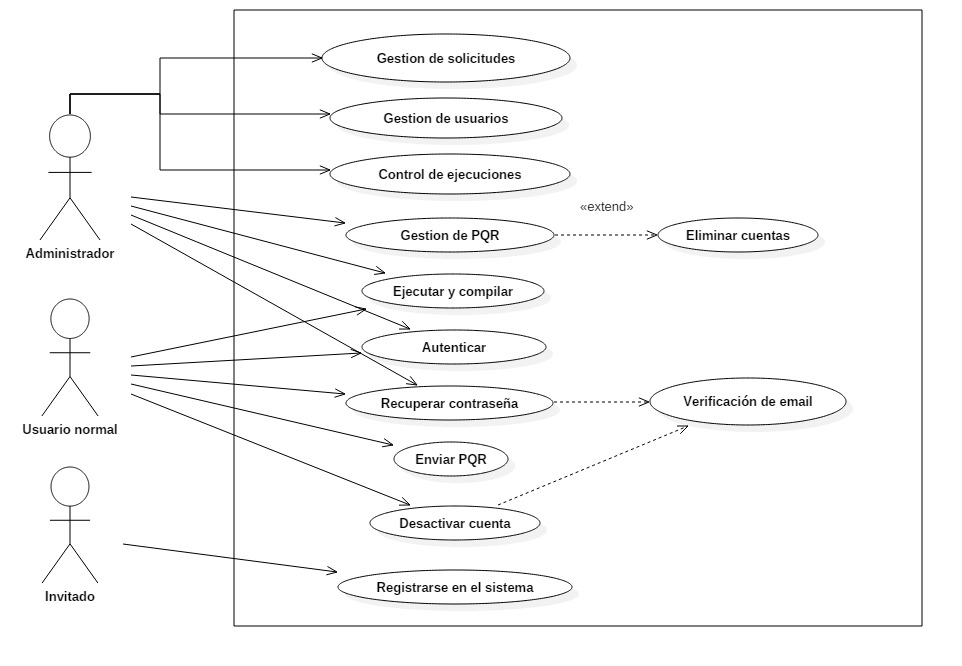


Ilustración 1

Fuente: Los autores

En el diagrama anterior se muestra los casos de uso que puede desarrollar el usuario administrador, el normal e invitado. Hay que tener en cuenta que el usuario administrador no puede hacer procesos como el de registrarse debido a que es un usuario por defecto.

## Diagrama de secuencia o actividades detallado (por cada escenario)

|  |  |
| --- | --- |
| **Caso de uso** | Autenticar |
| **Actores** | Usuarios |
| **Propósito** | En este caso de uso el usuario ingresa el dominio indicado por medio de un navegador web. |
| **Resumen** | Este caso de uso comienza cuando el usuario ingresa a la web y termina cuando el usuario se autentica |
| **Tipo** | Esencial |
| **Referencias** | 1,2,10,15 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. Este caso de uso inicia cuando el usuario ingresa al sistema por medio de un navegador. | 2. El sistema pide verificación de usuario para acceder a la información por medio de dos campos, uno designado para el Nombre de usuario y otro para la contraseña. |
| 3. El usuario se identifica con su Nombre de usuario y contraseña. | 4. El sistema verifica que los datos ingresados en los campos cumplan con los requisitos mínimos. |
|  | 5. El sistema verifica que el Nombre de usuario y la contraseña estén registradas y que coincidan. |
|  | 6. El sistema permite el acceso al sistema y a toda la información correspondiente al rol que desempeñe el usuario. |
| 7. El usuario tiene acceso a la información correspondiente a su rol. |  |
| **Curso Alterno** | |
| **Acción 2:** Permitir al usuario registrarse al sistema en caso de que no esté registrado **IR AL CASO DE USO REGISTRARSE AL SISTEMA.** | |
| **Acción 4:** Si el nombre de usuario o contraseña no cumplen con los requisitos mínimos avisar al usuario. | |
| **Acción 5:** Si el nombre de usuario está registrados, el sistema muestra un mensaje de no autorizado. | |

**Diagrama de actividad Autenticar:**

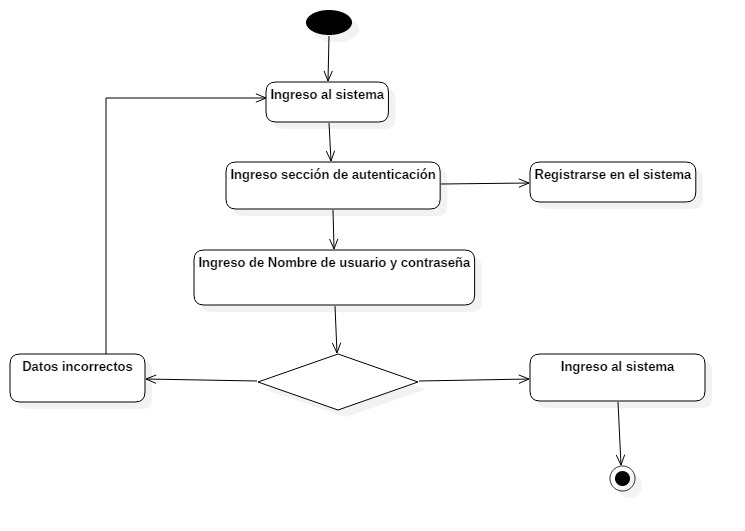
****

Ilustración 2

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Registrarse al sistema |
| **Actores** | Usuarios |
| **Propósito** | En este caso de uso el usuario se registra en el sistema. |
| **Resumen** | Este caso de uso comienza cuando el usuario se quiere registrar en el sistema y termina cuando dicho registro es permitido. |
| **Tipo** | Esencial |
| **Referencias** | 3,5,7,13 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. Este caso de uso inicia cuando el usuario quiere registrarse en el sistema. | 2. El sistema genera un formulario de registro con los siguientes campos: Nombre de usuario, contraseña (mínimo 4 caracteres alfanuméricos), nombres y apellidos, correo electrónico. |
| 3. El usuario diligencia el formulario y lo envía. | 4. El sistema pasa a verificar la información de cada campo. **DIRIGIRSE AL CASO DE USO VERIFICAR DATOS** |
|  | 5. El sistema genera el registro y pasa a guardarlo en la base de datos. |
|  | 6. El sistema muestra al usuario que su registro se ha hecho con éxito. |
| 7. El usuario recibe la confirmación del registro al sistema. |  |
| **Curso Alterno** | |

**Diagrama de actividad registrarse al sistema:**

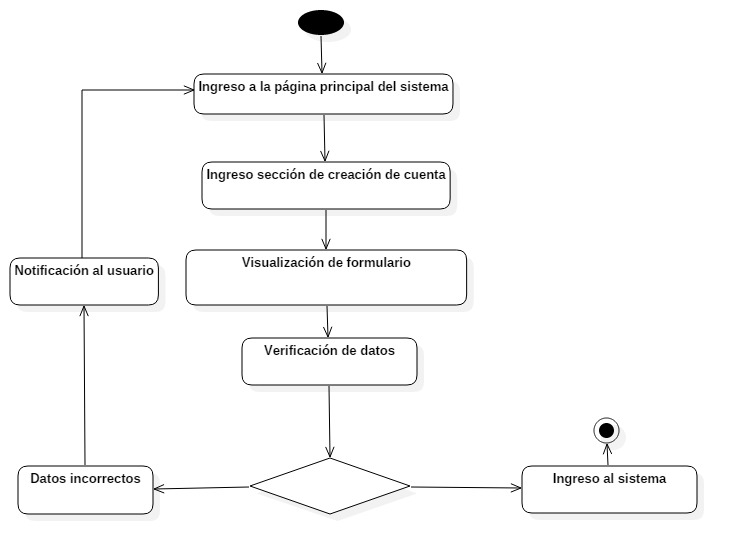


Ilustración 3

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Verificar Datos |
| **Actores** | Ninguno |
| **Propósito** | En este caso de uso el sistema verifica la consistencia y veracidad de los datos suministrados por el usuario. |
| **Resumen** | Este caso de uso comienza cuando se debe realizar una verificación de los datos y termina cuando se notifica si los datos son correctos o incorrectos. |
| **Tipo** | Esencial |
| **Referencias** |  |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
|  | 1. El sistema pasa a verificar que cada uno de los campos que el usuario ha diligenciado sean consistentes cumpliendo los requerimientos de cada campo. |
|  | 2. El sistema verifica si el Nombre de usuario ya se encuentra registrado. |
|  | 3. El sistema verifica que la contraseña cumpla con los requerimientos y que sea consistente con la verificación. |
| 4. Avisar al sistema que todos los datos son consistentes. **CONTINUAR EN EL SIGUIENTE DEL CASO DE USO QUE HIZO EL LLAMADO** |  |
| **Curso Alterno** | |
| **Acción 1:** Si el usuario omite alguno de los campos obligatorios mostrar un mensaje que avise al usuario de dicha omisión. | |
| **Acción 2:** Si el Nombre de usuario está registrado avisar al usuario que ya existe un registro. | |
| **Acción 3:** Si la contraseña ingresada por el usuario no cumple con los requisitos mínimos mostrar un mensaje al usuario de aviso y recordando dichos requisitos. | |

**Diagrama de actividad verificar datos:**

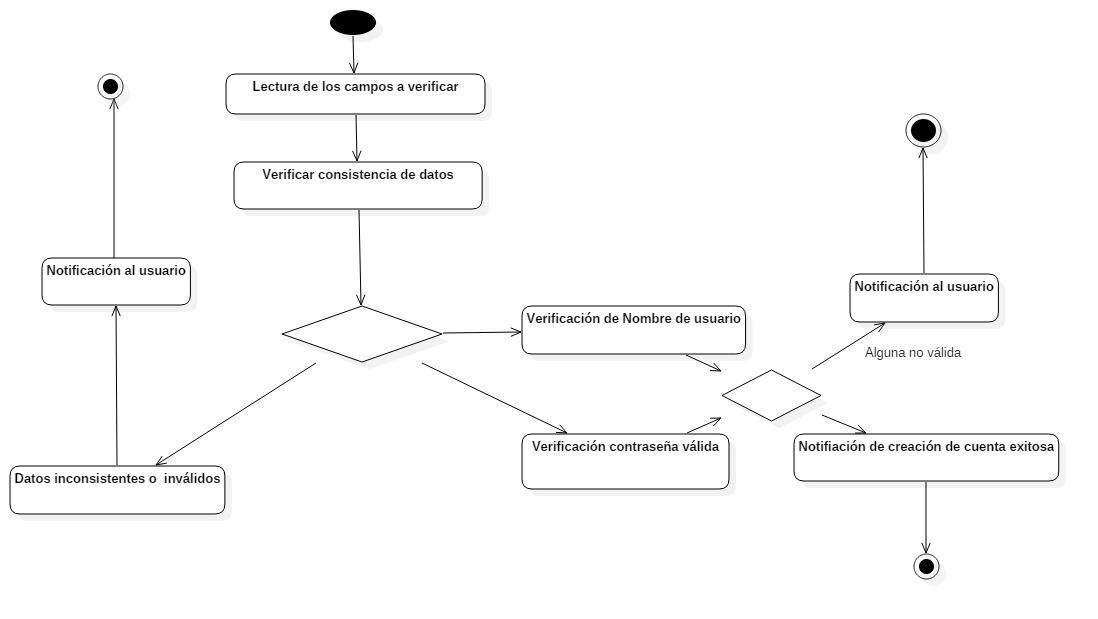


Ilustración 4

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Habilitar cuenta de usuario |
| **Actores** | Usuario Administrador |
| **Propósito** | En este caso de uso el Administrador usuario podrá el estado de una cuenta nueva. |
| **Resumen** | Este caso de uso comienza cuando un usuario se registra y termina en el momento que se habilita una cuenta. |
| **Tipo** | Esencial |
| **Referencias** | 1,4,7,11 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. Este caso de uso inicia cuando un invitado se registra de manera exitosa en la plataforma. | 2. el sistema almacena la petición con el nombre de usuario y un id de petición. |
| 3. El administrador accede a la web y se dirige a su vista administrativa. |  |
| 4. El Administrador autoriza o no al usuario deseado. | 5. El sistema cambia el estado del usuario. |
| 6. El usuario deberá iniciar sesión en la web. |  |
| 7. El usuario podrá acceder al compilador online. |  |
| **Curso Alterno** | |
| 7. De no ser admitido, el usuario recibirá una notificación. | |

**Diagrama de actividad Habilitar cuenta de usuario:**

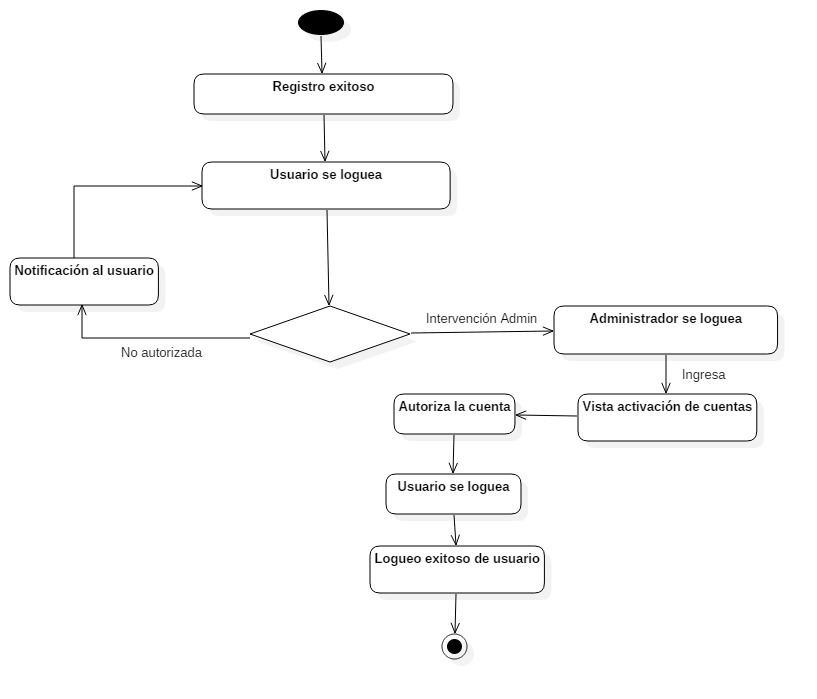


Ilustración 5

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Gestión de usuarios |
| **Actores** | Usuario Administrador |
| **Propósito** | En este caso de uso el usuario que desempeñe el rol de administrador podrá eliminar a los usuarios de la base de datos. |
| **Resumen** | Este caso de uso comienza el administrador va a eliminar un usuario de la base de datos y termina cuando dicho usuario sea borrado de la base de datos. |
| **Tipo** | Esencial |
| **Referencias** | 4,6,8,9 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. El caso de uso inicia cuando el administrador va a la opción de eliminar usuarios. | 2. El sistema genera la lista de usuarios, nombre de usuario y su id. |
| 3. El administrador llena el campo con el nombre de usuario correspondiente de la persona a modificar. | 4. El sistema realiza una lectura del campo y verifica su consistencia. |
|  | 5. El sistema elimina el usuario pertinente. |
| **Curso Alterno** | |
| **Acción 4:** Si la información no es consistente, la lista de usuarios no se alterará. | |

**Diagrama de actividad Gestión de usuario:**

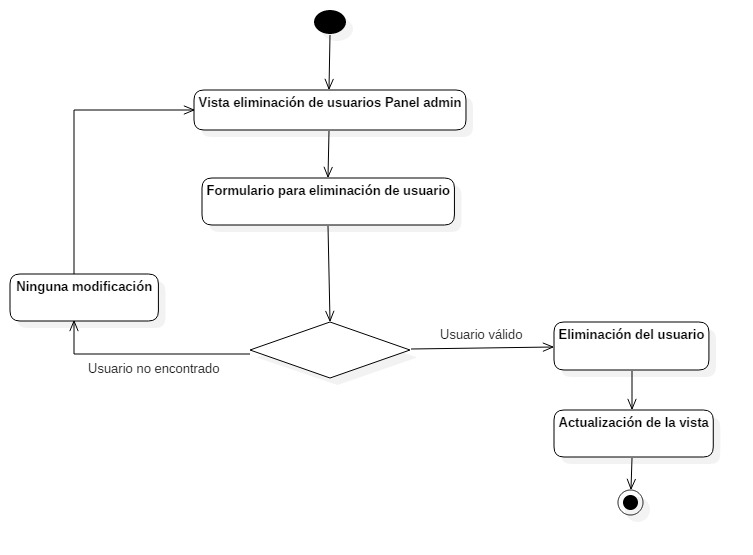


Ilustración 6

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Petición de eliminación de cuentas |
| **Actores** | Usuarios |
| **Propósito** | En este caso de uso el usuario que desempeñe el rol de usuario normal envía una petición al administrador para darse de baja en el sistema. |
| **Resumen** | Este caso de uso comienza cuando el usuario normal hace una petición darse de baja en el sistema y termina en el momento que la eliminación de cuenta por parte del administrador es realizada. |
| **Tipo** | Esencial |
| **Referencias** | 4,5,8,11 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. Este caso de uso inicia cuando el usuario realiza una petición para darse de baja en el sistema. **DIRIGIRSE AL CASO DE USO ENVÍO DE PETICIONES.** Nota: el usuario modifica el asunto del formulario con la intensión de eliminar su cuenta del sistema. |  |
| 2. El administrador envía un email de confirmación explicando el motivo del deceso del usuario. |  |
| 3. El usuario debe responder el email enviado por el administrador con su nombre de usuario, apellido e email. | 4. El sistema de email por defecto notifica al administrador que tiene un correo nuevo. |
| 5. El administrador revisará el correo y verificará la consistencia de la información. |  |
| 6. El administrador accede al panel administrativo. |  |
| 7. **DIRIGIRSE AL CASO DE USO GESTIÓN DE USUARIOS.** |  |
| **Curso Alterno** | |
| **Acción 3:** Si el usuario nunca responde el email, se hará caso omiso del contenido de la petición. | |
| **Acción 5:** Si el administrador nota inconsistencias en la información del mail enviado se hará caso omiso del contenido de la petición. | |

**Diagrama de actividad de Modificación de Información Petición de eliminación de cuentas:**

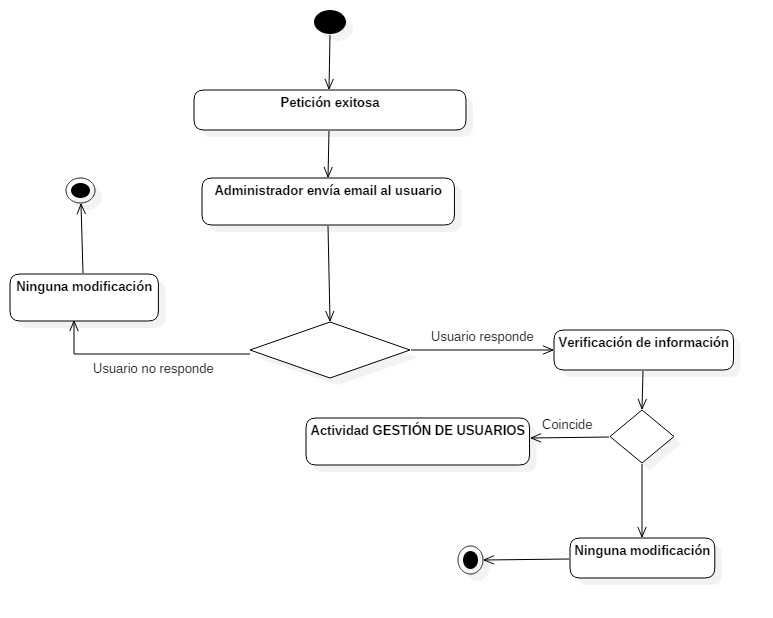


Ilustración 7

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Envío de peticiones |
| **Actores** | Usuario Administrador, normal |
| **Propósito** | En este caso de uso los usuarios que deseen enviar PQR´S al sistema deberán llenar un simple formulario. |
| **Resumen** | Este caso de uso comienza cuando un usuario autorizado envía una petición al administrador y termina cuando el sistema confirma que la petición fue enviada satisfactoriamente. |
| **Tipo** | Esencial |
| **Referencias** | 2,3,4,5,8,9,11 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. El caso de uso inicia cuando el usuario logueado se dirige a la opción de diligenciar petición. | 2. El sistema muestra los siguientes campos: date, Subject, body. |
| 3. El usuario llena todos los campos incluyendo la petición. | 4. El sistema recibe la petición y avisa al usuario que su envío fue exitoso. |
| 5. El usuario recibe el mensaje de que el envío fue exitoso. |  |
| **Curso Alterno** | |

**Diagrama de actividad de Envío de Peticiones:**

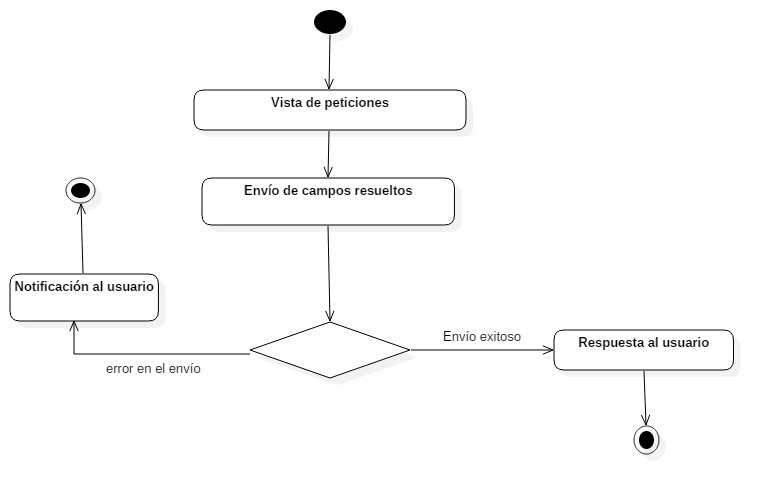


Ilustración 8

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Resolver peticiones |
| **Actores** | Usuarios |
| **Propósito** | En este caso de uso el usuario que desempeñe el rol de Administrador podrá ver y resolver todas las peticiones que llegan al sistema. |
| **Resumen** | Este caso de uso comienza cuando un usuario realiza una petición y termina cuando el administrador la da como resuelta. |
| **Tipo** | Esencial |
| **Referencias** | 8,9,11 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. El caso de uso inicia cuando un usuario realiza una petición. **DIRIGIRSE AL CASO DE USO ENVÍO DE PETICIONES** |  |
| 2. El administrador atiende la petición deseada marcando ingresando el id de la petición. | 3. El sistema da como resuelta la petición eliminándola de la base de datos. |
| **Curso Alterno** | |

**Diagrama de actividad de Resolver Peticiones:**

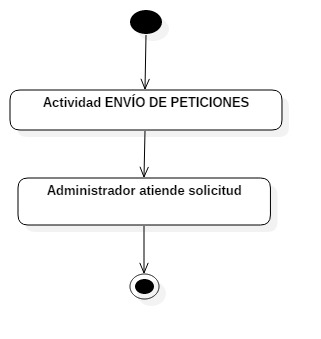


Ilustración 9

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Compilar y ejecutar |
| **Actores** | Usuario Estudiante Nuevo o Visitante |
| **Propósito** | En este caso de uso el usuario normal o administrador desea ejecutar algún algoritmo dentro de la plataforma. |
| **Resumen** | Este caso de uso comienza cuando un usuario ejecuta un algoritmo y termina cuando el algoritmo termine su ejecución. |
| **Tipo** | Esencial |
| **Referencias** | 2,3,14 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. El caso de uso inicia cuando un usuario desea compilar y ejecutar algún algoritmo dentro del sistema. |  |
| 2. El usuario escribe su código dentro de la vista del compilador online. | 3. El sistema almacena de manera parcial el código escrito dentro de esta vista y llama a rutinas del sistema para poder compilar el código. |
|  | 4. El sistema ejecuta el código. |
|  | 5. El sistema almacena las salidas del programa en ejecución para mostrarlas al usuario. |
| 6. El usuario observa las salidas cuando el proceso de ejecución ha terminado. |  |
| **Curso Alterno** | |
| **Acción 3:** Si se produce un error en la compilación, se le notificará al usuario el porqué del error, abortando la ejecución. | |
| **Acción 5:** Si se produce un error en la ejecución, se le notificará al usuario el porqué del error. | |
| **Acción 6:** Si la ejecución de algún programa se prolonga por mucho tiempo (variable, preestablecido por el administrador) el sistema notifica al usuario que no se puedo completar la ejecución. | |

**Diagrama de actividad de Acceso a Compilar y ejecutar:**

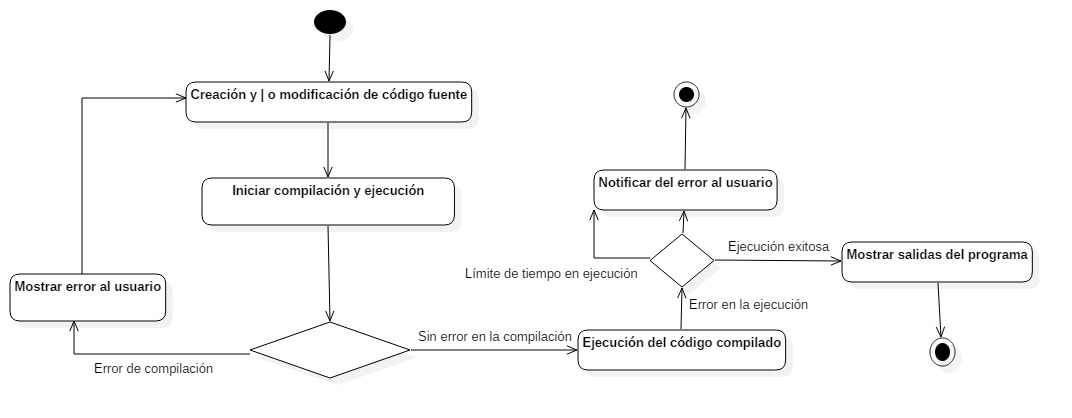


Ilustración 10

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Administración de recursos y ejecuciones. |
| **Actores** | Administrador |
| **Propósito** | En este caso de uso el usuario de rol administrador podrá monitorear los recursos y tareas lanzadas en la plataforma por medio de una terminal web, también podrá eliminar las ejecuciones de tipo maliciosas o que nunca terminarán. |
| **Resumen** | Este caso de uso comienza el usuario Administrador accede a una terminal web en el panel administrativo y termina cuando se acabe una sesión en dicha terminal. |
| **Tipo** | Esencial |
| **Referencias** | 2,3,4,11 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. El caso de uso inicia cuando el usuario Administrador se dirige al panel administrativo. | 2. El sistema muestra una terminal de tipo tty vía web. |
|  | 3. El sistema pide contraseña de usuario para confirmación. |
| 4. El usuario Administrador se autentica en dicha terminal. | 5. El sistema verifica el nombre de usuario y contraseña. |
| 6. El usuario Administrador accede a la terminal. | 7. El sistema coordina cada comando en tiempo real. |
| 8. El .usuario administrador ejecuta los comandos deseados |  |
| 9. El usuario termina la sesión en la terminal |  |
|  | 10. El sistema termina la sesión activa. |
| **Curso Alterno** | |
| **Acción 5:** El usuario no acierta en el proceso de autenticación la terminal no se abrirá y se seguirá preguntando por nombre de usuario y contraseña. | |
| **Acción 5:** El nombre de usuario y contraseña son diferentes a las que se le pide al administrador para autenticarse. | |
| **Acción 7:** La limitante de los comando está determinada por el sistema operativo nativo del servidor. | |

**Diagrama de actividad Administración de recursos y ejecuciones**:

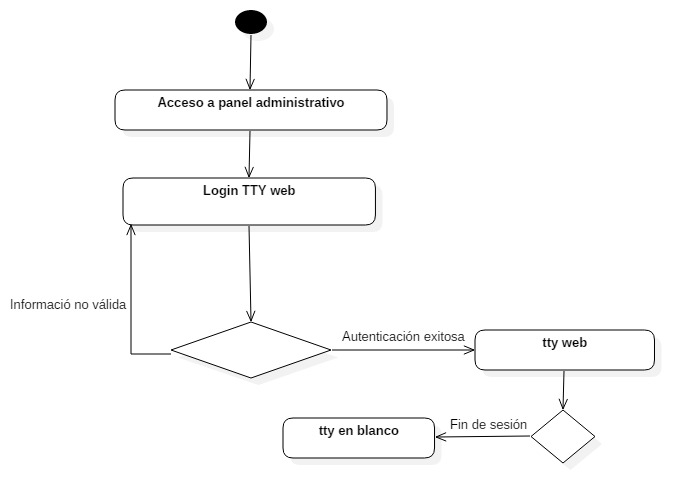


Ilustración 11

Fuente: Los autores

|  |  |
| --- | --- |
| **Caso de uso** | Recuperar Contraseña |
| **Actores** | Usuarios |
| **Propósito** | En este caso de uso los usuarios pueden recuperar su contraseña de usuario. |
| **Resumen** | Este caso de uso comienza el usuario registrado en el sistema quiere recuperar la contraseña y termina cuando logra la recuperación de su cuenta con una nueva contraseña. |
| **Tipo** | Esencial |
| **Referencias** | 9 |
| **Curso Normal de los Eventos** | |
| **Acción de los actores** | **Respuesta del sistema** |
| 1. El caso de uso inicia cuando el usuario quiere recuperar su contraseña. Y va a la opción olvide contraseña. | 2. El sistema genera la opción de recuperar contraseña mediante de pregunta de su correo y nombre de usuario. |
| 3. El usuario ingresa la información pertinente. | 4. El sistema verifica que el nombre de usuario y correo estén asociados a la misma cuenta. |
|  | 5. El sistema envía un correo con los pasos a seguir para recuperar la contraseña. |
| 6. El usuario sigue las instrucciones del correo electrónico. |  |
| 7. El usuario se dirige a la vista indicada en el correo electrónico. | 8. El sistema genera la vista. |
| 10. El usuario ingresa el id enviado por el correo electrónico, nueva contraseña y correo electrónico. | 11. El sistema verificará la información solicitada. |
|  | 12. El sistema guarda la nueva contraseña en la base de datos y confirma la acción exitosa. |
| 13. El usuario recibe la confirmación de éxito. |  |
| 14. El usuario se autentica con la nueva contraseña. |  |
| **Curso Alterno** | |
| **Acción 4:** Si el usuario suministra información no verídica el sistema hace caso omiso sin aviso alguno. | |
| **Acción 6:** Si el usuario proporciona un correo inválido el proceso de recuperación de contraseña no se podrá realizar. | |
| **Acción 11:** Si el usuario suministra información no verídica el sistema hace caso omiso al proceso sin aviso alguno. | |

**Diagrama de actividad de Recuperar Contraseña:**

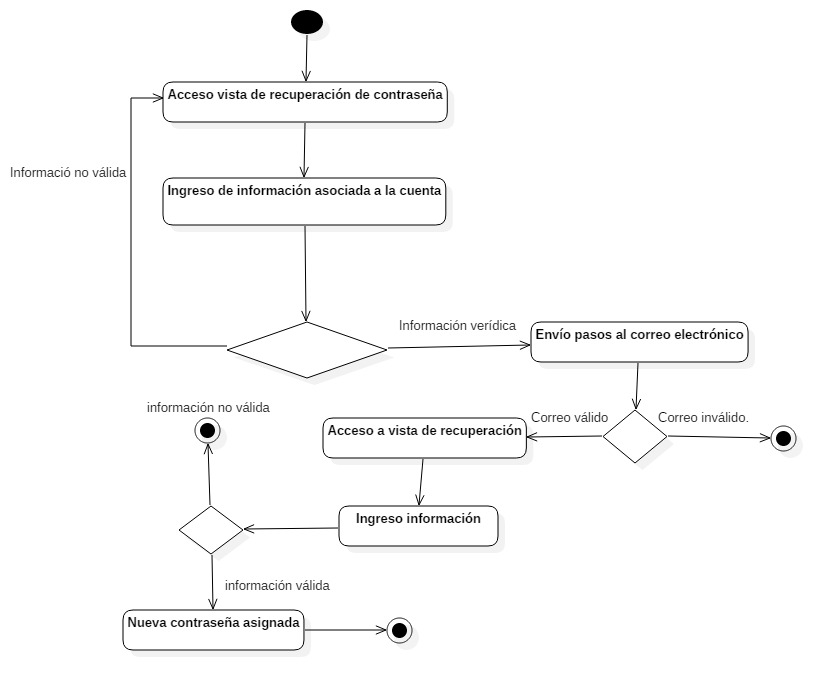


Ilustración 12

Fuente: Los autores

## Fases de diseño

### Diagrama de base de datos No relacional (Document view)

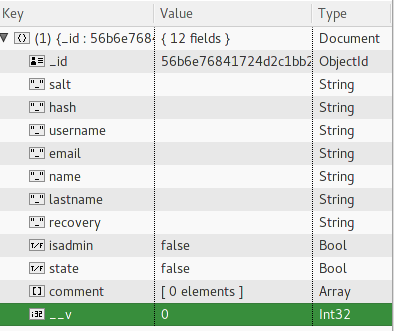


Ilustración 13

Fuente: Los autores

### Diagrama de componentes

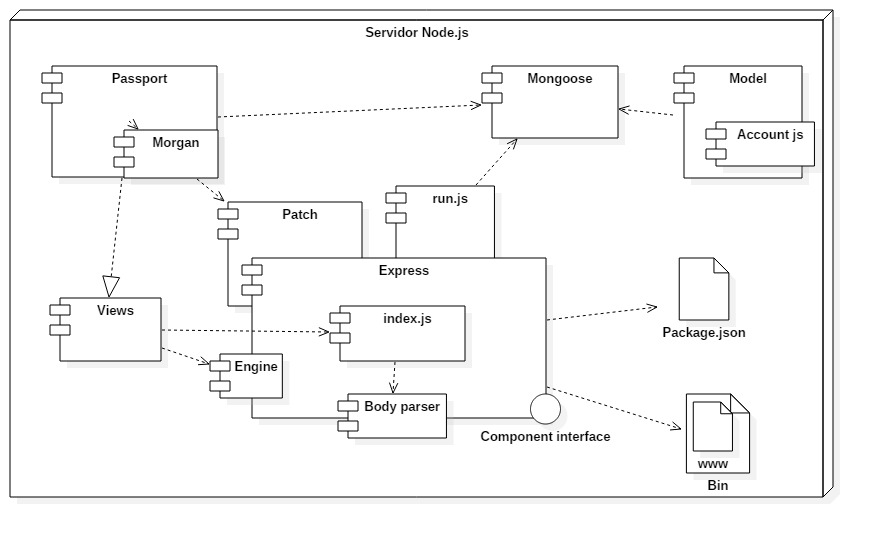


Ilustración 14

Fuente: Los autores

### Diagrama de despliegue

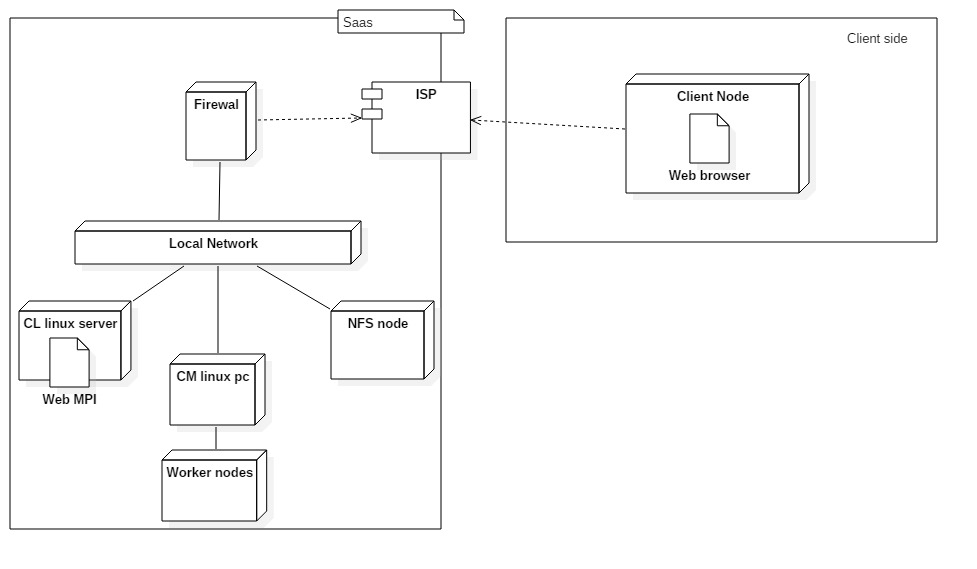


Ilustración 15

Fuente: Los autores

## Fase de implementación

### Plantilla de estándar de codificación:

## Guía de estilo. Nomenclatura en archivos Javascript[[1]](#footnote-1)

## “1. General

* Los ficheros tienen que ser nombrados en minúsculas utilizando la forma ‘[underscore](http://en.wikipedia.org/wiki/Underscore)‘.
* El objeto global se declara según la forma ‘UPPER\_CASE’ con guión bajo como separador y precedido del signo de dolar ($).
* Los subojetos principales que categorizan las funcionalidades, se declaran según la forma ‘UPPER\_CASE’ con guión bajo como separador.
* Las variables tienen que ser nombradas utilizando la forma ‘[camelCase](http://en.wikipedia.org/wiki/CamelCase)‘.
* Los constructores tienen que ser nombrados utilizando la forma ‘[StudlyCaps](http://en.wikipedia.org/wiki/StudlyCaps)‘.
* Las constantes, de haberlas, tienen que ser nombradas en ‘UPPER\_CASE’ con guión bajo como separador.
* Las propiedades tienen que ser nombradas utilizando ‘[camelCase](http://en.wikipedia.org/wiki/CamelCase)‘.
* Los métodos tienen que ser nombrados utilizando ‘[camelCase](http://en.wikipedia.org/wiki/CamelCase)‘.

## 2. Nombres de Fichero

Los ficheros se nombrarán siempre en minúsculas para evitar errores con aquellas plataformas ‘sensitive-case’. Los ficheros tienen que concluir con la extensión .js, y no deben incluir signos de puntuación excepto – (guión medio) o \_ (guión bajo), siendo preferible este último a la hora de separar palabras.

Ficheros correctamente nombrados:

|  |
| --- |
| jquery.js  betacompany\_utils.js  namespace\_example.js |

Ficheros mal nombrados:

|  |
| --- |
| ConfirmDialog.js  finalconf.js  contactStatus.js |

En los ejemplos anteriores, se está utilizando indistintamente los criterios ‘[StudlyCaps](http://en.wikipedia.org/wiki/StudlyCaps)‘, ‘nonseparation‘ y ‘[camelCase](http://en.wikipedia.org/wiki/CamelCase)‘ respectivamente.

## 3. Objeto Global.

El objeto global es aquel que envuelve a toda la aplicación y que evita la contaminación del entorno global. Para denotar su jerarquía, se declara en mayúsculas precedido del signo de dólar ($) y su nombre suele coincidir con el de la aplicación.

|  |
| --- |
| **var** $UDO = {}; |
|  |

## 4. Sub-objetos globales.

Los sub-objetos globales son aquellas subdivisiones principales del proyecto que actúan como categorías superiores. Suelen segmentar el código en bibliotecas o bloques más manejables.

|  |
| --- |
| $UDO.UI = {}; *// Jerarquía superior para todos los elementos gráficos.*  $UDO.TICKETS = {}; *// Jerarquía superior para todas las acciones de los tickets.*  $UDO.WINDOW = {}; *// Jerarquía superior para todas las acciones que engloben ventanas.*  $UDO.UTILS = {}; *// Jerarquía superior para utilidades. Aquí se suelen incluir bibliotecas varias.* |

## 5. Variables.

Las variables tienen que declararse utilizando la forma ‘camelCase’. No deben ir precedidas de un guion bajo para indicar que se tratan de variables privadas.

Cuando se declaren más de una variable de forma consecutiva, solo se utilizará una única sentencia ‘var‘.

|  |
| --- |
| **var** myVar = foo; |
| **var** myVar = 'foo',  myVar2 = 'bar',  myVar3 = 'foobar'; | |

Las variables que vayan a contener una instancia de jQuery, se identificarán **anteponiéndoles un signo de dólar** ($).

|  |
| --- |
| **var** myVar = 'foo',  $myLayer = $('#layer-one'),  $myLayer2 = $('#layer-two'); |

## 6. Constructores.

Los constructores (también llamados clases) tienen que nombrarse siguiendo la forma ‘StudlyCaps‘ para diferenciarlos a simple vista del resto de estructuras.

|  |
| --- |
| **var** WindowFrame = **function**(col) {  **var** that = **this**,  color = col;    **this**.getColor = **function**() {  **return** color;  };  }    **var** blueWindow = **new** WindowFrame("blue"); |

## 7. Constantes.

Las constantes se nombran en mayúsculas y separando palabras con un guión bajo. Evitamos el uso del comando @const para evitar errores en Internet Explorer.

|  |
| --- |
| $UDO.UI.window.defaults = **function**() {  **var** WINDOW\_WIDTH = 400px,  WINDOW\_HEIGHT = 400px;    */\* ... \*/*  } |

## 8. Propiedades.

Las propiedades se declaran utilizando ‘camelCase‘ al igual que las variables.

|  |
| --- |
| $UI.TICKETS.createNewTicket = **function**( params ) {  **var** ticketType,  ticketCreator = params.currentUser,  startDate = params.currentDate,  finishDate = **undefined**,  ticketDetails;  } |

## 9. Métodos.

Los métodos de los objetos tienen que ser nombrados utilizando ‘camelCase‘, al igual que las variables y propiedades.

|  |
| --- |
| $UDO.UI.printWindow = **function**( currentWindow ) {    **var** retrievingData = **function**() { */\* ... \*/* },    populatingWindow = **function**() { */\* ... \*/* },  printingWindow = **function**() { */\* ... \*/* }    } |

“

### Lenguaje de programación

JavaScript[[2]](#footnote-2) (abreviado comúnmente JS) es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

### IDE de desarrollo

Sublime[[3]](#footnote-3) Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode.

Se distribuye de forma gratuita, sin embargo no es software libre o de código abierto, se puede obtener una licencia para su uso ilimitado, pero el no disponer de ésta no genera ninguna limitación más allá de una alerta cada cierto tiempo.

### Motor de bases de datos

MongoDB[[4]](#footnote-4) (de la palabra en inglés “humongous” que significa enorme) es un sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

MongoDB forma parte de la nueva familia de sistemas de base de datos NoSQL. En vez de guardar los datos en tablas como se hace en las base de datos relacionales, MongoDB guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (MongoDB llama ese formato BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida.

El desarrollo de MongoDB empezó en octubre de 2007 por la compañía de software 10gen.[cita requerida] Ahora MongoDB es una base de datos lista para su uso en producción y con muchas características (features). Esta base de datos se utiliza mucho en la industria1 y MTV Network,2 Craiglist3 y Foursquare4 son algunas de las empresas que utilizan esta base de datos.

El código binario está disponible para los sistemas operativos Windows, Linux, OS X y Solaris.

### Framework

Node.js [[5]](#footnote-5)es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla.

Node.js es similar en su propósito a Twisted o Tornado de Python, Perl Object Environment de Perl, React de PHP, libevent o libev de C, EventMachine de Ruby, vibe.d de D y de Java existe Apache MINA, Netty, Akka, Vert.x, Grizzly o Xsocket. Al contrario que la mayoría del código JavaScript, no se ejecuta en un navegador, sino en el servidor. Node.js implementa algunas especificaciones de CommonJS. Node.js incluye un entorno REPL para depuración interactiva.

## Fase de pruebas

Para realizar una buena implementación se diseñaron casos de prueba para verificar el funcionamiento adecuado del sistema en todos los aspectos básicos. Dentro de estos aspectos se pueden identificar la funcionalidad, la seguridad, el rendimiento, la usabilidad y accesibilidad. A continuación se podrá observar en el documento FormatoPruebasWebmpi.xls el diseño de los casos de pruebas en cada aspecto de los que hablamos anteriormente.

### Herramientas para pruebas

Para realizar las pruebas de funcionamiento de la totalidad del sistema se utilizaran diferentes herramientas para ejecutar las verificaciones. Cuando se habla de la totalidad del sistema se hace referencia a la funcionalidad general, seguridad, rendimiento, accesibilidad y usabilidad. Para este sprint se harán pruebas solamente de funcionalidad general del sistema. A continuación se nombraran las herramientas que serán de utilidad en la verificación de todos los aspectos del sistema:

* **Websiteoptimization:** Prueba la velocidad del sitio web para mejorar el rendimiento. Calcula el tamaño de página, la composición y el tiempo de descarga. La secuencia de comandos calcula el tamaño de los elementos individuales y resume cada tipo de componente de página de web.
* **Tawdis:** Comprobar el nivel de accesibilidad alcanzado en el diseño y desarrollo de páginas web con el fin de permitir el acceso a todas las personas independientemente de sus características diferenciadoras.
* **Validator.w3:** Este validador comprueba la validez de marcado de documentos Web en HTML, XHTML, SMIL, MathML, etc.

### Análisis de los resultados de pruebas.

Después de realizar las pruebas de funcionalidad del sistema con la codificación que hay hasta el momento, se puede decir que un funcionamiento básico sólido. El desarrollo que se ha hecho hasta ahora tiene algunas fallas que se pueden mejorar. Unas de los fallas del sistema es la ineficiente verificación de los campos a la hora de crearse una cuenta, también se presentan falencias a la hora de manejar errores y la pertinentes notificaciones al usuario.

# Conclusiones.

Primero se estableció como hacer un plan general de un proyecto, sus fases y variantes, a saber cómo distribuir cargas de trabajo. Cabe resaltar el uso de las comunidades virtuales que también formó parte importante a la hora de la elaboración del este documento.

El presente documento sirvió de base guía para la obtención del diseño concreto de la fase de inicio del proyecto, gracias a la elaboración estructural, detallada y elocuente de actividades contenidas en el campo de diseño como lo es la ingeniería de software.

Se interiorizó como se debe de formar e integrar la parte de ingeniería del software a un proyecto y como, a partir de estos elementos de diseño, llegar a concretar de manera más tangible el proyecto. También se hizo el análisis correspondiente al primer bosquejo de la plataforma arrojando resultados como una buena página con lo que se pidió en los requerimientos, cumpliéndolos de buena manera, sin dejar ningún detalle aparte.

También se le asocia la fase de buen maquetamiento y documentación que hace gala de un despliegue organizacional de lo que contiene el proyecto haciendo que el lector del documento en su preferencia el product-owner se sienta guiado y sienta que tiene el control general de su producto.

# Bibliografía.

1.G. Rudolph, Federico. Programación. 1 ed. Mar del plata, Argentina.:   
Introducción a Visual Studio .NET, 2004, 35 p.

2. Villa, Paula Andrea. Versión 1. Conoce Nuestra U. UTP. 2014, 5 p.

3. Web performace powered by google, {en línea} 2016.Disponible en: <https://developers.google.com/speed/pagespeed/insights/>

4. Philippe Kruchten, Architectural Blueprints—The “4+1” ViewModel of Software Architecture, en: IEEE software 12(6) ,November 1995, pp. 42-50.

5. ICONTEC, Normas para la elaboración de trabajos y documentos,{en línea} 2014 ,48pDisponible en: <http://www.icontec.org>

6. Validador clasificatorio de la w3c, {en línea}, Disponible en: <http://validator.w3.org/>

1. Estilos de codificación en javasript, {en línea} , { 4 febrero 2016 }Disponible en <http://www.etnassoft.com/2012/10/23/guia-de-estilo-nomenclatura-en-archivos-javascript/> [↑](#footnote-ref-1)
2. Qué es javasript ?, {en línea} , { 4 febrero 2016 }, Disponible en <https://es.wikipedia.org/wiki/JavaScript> [↑](#footnote-ref-2)
3. Sublime text , {en línea} , { 4 febrero 2016 }, Disponible en https://en.wikipedia.org/wiki/Sublime\_Text [↑](#footnote-ref-3)
4. Definición MongoDB , {en línea} , { 4 febrero 2016 }, Disponible en <https://es.wikipedia.org/wiki/MongoDB> [↑](#footnote-ref-4)
5. Definición Node.js , {en línea} , { 4 febrero 2016 }, Disponible en <https://es.wikipedia.org/wiki/Node.js> [↑](#footnote-ref-5)