Active SRS

*Versión 2.1.0*

Proyecto de Ingeniería de software

Entrega 2

Active

Juan Miguel Gómez

Carlos Eduardo Camacho

Natalia Sofía Otero

Luis Felipe Urdaneta

Bogotá D.C

17/10/2017

Versión 1.0

# Historial de Cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Sección | Nombre | Responsable |
| 16/9/2017 | 6.1 | Propósito del documento | Carlos Camacho |
| 16/9/2017 | 6.2 | Alcance del Documento | Luis Urdaneta |
| 16/9/2017 | 6.3 | Glosario | Natalia Otero |
| 16/9/2017 | 7.1 | Perspectiva del Producto | Natalia Otero |
| 16/9/2017 | 7.1 | Interfaces con el Sistema | Juan Gómez |
| 16/9/2017 | 7.2 | Funciones del Producto | Carlos Camacho |
| 16/9/2017 | 7.1.2 | Restricciones de memoria | Carlos Camacho |
| 16/9/2017 | 7.1.3 | Operaciones | Carlos Camacho |
| 16/9/2017 | 7.4 | Restricciones | Carlos Camacho |
| 16/9/2017 | 7.5 | Modelo de Dominio | Natalia Otero |
| 16/9/2017 | 7.6 | Suposiciones y dependencias | Juan Gómez |
| 16/9/2017 | 8.1.1 | Interfaces con el Usuario | Luis Urdaneta |
| 16/9/2017 | 8.1.2 | Interfaces con el Hardware | Juan Gómez |
| 16/9/2017 | 8.1.3 | Interfaces con el software | Luis Urdaneta |
| 16/9/2017 | 8.1.4 | Interfaces de comunicación | Luis Urdaneta |
| 20/9/2017 | 7.3 | Características del usuario | Luis Urdaneta |
| 20/9/2017 | 7.7 | Distribución de requisitos | Carlos Camacho |
| 27/9/2017 | 8.1 | Requisitos de Interfaces externas | Juan Gómez |
| 27/9/2017 | 8.2 | Características del Producto de Software | Natalia Otero |
| 27/9/2017 | 8.3 | Requisitos de Desempeño | Luis Urdaneta |
| 27/9/2017 | 8.4 | Restricciones de Diseño | Carlos Camacho |
| 27/9/2017 | 8.5 | Atributos de Software | Luis Urdaneta, Juan Gómez |
| 27/9/2017 | 8.6 | Requisitos de la Base de Datos | Luis Urdaneta, Juan Gómez |
| 2/10/2017 | 9.1 | Planeación | Carlos Camacho |
| 2/10/2017 | 9.2 | Levantamiento | Natalia Otero |
| 2/10/2017 | 9.3 | Especificación | Luis Urdaneta |
| 2/10/2017 | 10 | Proceso de Verificación y Validación | Natalia Otero, Juan Gómez |
| 16/10/2017 | 11 | Anexos | Natalia Otero |
| 16/10/2017 | 12 | Referencias | Natalia Otero |

Tabla 1 Historial de Cambios

# Prefacio

En el presente documento se expone el desarrollo del Software Requirements Specification (SRS) aplicado para el desarrollo del aplicativo web Campus Finder, este aplicativo busca facilitar el desplazamiento de la comunidad educativa de la Pontificia Universidad Javeriana en la institución.

Los temas que se abordan de manera general en este documento corresponden a: la descripción global del proyecto, en el cual se especifican las principales interfaces para la funcionalidad y distintos servicios para promover el uso de este aplicativo web en la comunidad educativa de la universidad; también, se presentan las restricciones del proyecto, que en otras palabras significa, lo que abarca el proyecto y con que necesita contar. Otro de los temas que encontramos en el SRS es la especificación de requerimientos específicos de la plataforma web, donde se aborda de una manera más concreta cada requerimiento de las distintas interfaces, también, los requerimientos de desempeño y de las bases de datos, así como también los atributos y características que hacen parte de Campus Finder. Por último, encontramos el proceso de ingeniería de requisitos que se desarrolló para la elaboración de Campus Finder, la cual consta de tres grandes factores que son la planeación, el levantamiento y la especificación de los requisitos, estos factores se explican con mayor detalle más adelante.

El documento está dirigido a todos aquellos que estén interesados en conocer el desarrollo del SRS elaborado por el grupo ACTIVE para la elaboración del aplicativo Campus Finder, esto es con el fin de que los lectores puedan ver las características que este proceso de desarrollo lleva.



# Contenido

[1 Historial de Cambios 2](#_Toc496471613)

[2 Prefacio 4](#_Toc496471614)

[3 Contenido 5](#_Toc496471617)

[4 Lista de Figuras 7](#_Toc496471618)

[5 Lista de Tablas 7](#_Toc496471619)

[6 Introducción 8](#_Toc496471620)

[6.1 Propósito del documento 8](#_Toc496471621)

[6.2 Alcance del producto 8](#_Toc496471622)

[6.2.1 Funcionalidades 8](#_Toc496471623)

[6.2.2 Beneficios 9](#_Toc496471624)

[6.3 Glosario 10](#_Toc496471625)

[7 Descripción global 11](#_Toc496471626)

[7.1 Perspectiva de producto 11](#_Toc496471627)

[7.1.1 Interfaces con el sistema 11](#_Toc496471628)

[7.1.2 Restricciones de memoria 11](#_Toc496471629)

[7.1.3 Operaciones 12](#_Toc496471630)

[7.2 Funciones del producto 13](#_Toc496471631)

[7.3 Características del usuario 15](#_Toc496471632)

[7.4 Restricciones 16](#_Toc496471633)

[7.5 Modelo de dominio 16](#_Toc496471634)

[7.6 Suposiciones y dependencias 22](#_Toc496471635)

[7.7 Distribución de requisitos 22](#_Toc496471636)

[8 Requisitos específicos 23](#_Toc496471637)

[8.1 Requisitos de interfaces externas 23](#_Toc496471638)

[8.1.1 Interfaces con el usuario 23](#_Toc496471639)

[8.1.2 Interfaces con el hardware 23](#_Toc496471640)

[8.1.3 Interfaces con el software 24](#_Toc496471641)

[8.1.4 Interfaces de comunicación 26](#_Toc496471642)

[8.2 Características del producto software 26](#_Toc496471643)

[8.3 Requisitos de desempeño 26](#_Toc496471644)

[8.4 Restricciones de diseño 27](#_Toc496471645)

[8.5 Atributos del sistema software 28](#_Toc496471646)

[8.5.1 Usabilidad 32](#_Toc496471647)

[8.5.2 Rendimiento 35](#_Toc496471648)

[8.5.3 Soporte 36](#_Toc496471649)

[8.5.4 Implementación (+) 36](#_Toc496471650)

[8.6 Requisitos de la base de datos 37](#_Toc496471651)

[9 Proceso de ingeniería de requisitos 38](#_Toc496471652)

[9.1 Planeación 38](#_Toc496471653)

[9.1.1 Plan de administración de requisitos de software 38](#_Toc496471654)

[9.1.2 Frecuencia de planeación 38](#_Toc496471655)

[9.1.3 Recolección de requisitos 41](#_Toc496471656)

[9.2 Levantamiento 43](#_Toc496471657)

[9.3 Especificación 43](#_Toc496471658)

[10 Proceso de verificación y validación 44](#_Toc496471659)

[10.1 Proceso de validación 44](#_Toc496471660)

[10.2 Proceso de verificación 45](#_Toc496471661)

[11 Prototipo 45](#_Toc496471662)

[11.1 Diseño 45](#_Toc496471663)

[11.2 Justificación según priorización 45](#_Toc496471664)

[12 Anexos 45](#_Toc496471665)

[13 Referencias 46](#_Toc496471666)

# Lista de Figuras

[Ilustración 1 Diagrama modelo de dominio 17](#_Toc496471667)

[Ilustración 2Recolección de requisitos PMBOOK 38](#_Toc496471668)

[Ilustración 3 Levantamiento de requisitos 39](#_Toc496471669)

[Ilustración 4 Tipos de requerimientos 42](#_Toc496471670)

[Ilustración 5 Priorización de requerimientos 42](#_Toc496471671)

# Lista de Tablas

[Tabla 1 Historial de Cambios 3](#_Toc496471672)

[Tabla 2 Plantilla para los casos de uso 13](#_Toc496471673)

[Tabla 3 Usuarios registrados 15](#_Toc496471674)

[Tabla 4 Usuario no registrado 15](#_Toc496471675)

[Tabla 5 Usuario administrador 16](#_Toc496471676)

[Tabla 6 Plantilla para descripción de entidades 17](#_Toc496471677)

[Tabla 7 Descripción entidad cuenta 18](#_Toc496471678)

[Tabla 8 Descripción entidad preferencia 18](#_Toc496471679)

[Tabla 9 Descripción entidad Notificación/Alarma 19](#_Toc496471680)

[Tabla 10 Descripción entidad ruta 19](#_Toc496471681)

[Tabla 11 Descripción entidad Asignatura 19](#_Toc496471682)

[Tabla 12 Descripción entidad puntos de interés 20](#_Toc496471683)

[Tabla 13 Descripción entidad edificio 21](#_Toc496471684)

[Tabla 14 Restricciones de diseño 28](#_Toc496471685)

[Tabla 15 Plantilla documentación requerimientos 44](#_Toc496471686)

# Introducción

## Propósito del documento

El propósito de este documento es determinar y aplicar de forma adecuada el levantamiento de todos los requerimientos que son necesarios para las funcionalidades de App web “Campus Finder”, para esto se sigue el Software Requirements Specification (SRS) las cuales son un conjunto de recomendaciones para la especificación de requerimientos, que además cubre todos los aspectos importantes para hacer una adecuada gestión de requerimientos

El desarrollo de este documento facilita la definición y análisis de las funcionalidades de el aplicativo, de esta forma se determina el alcance del proyecto y se puede hacer una mejor especificación de este.

Como aspecto importante de este documento encontramos:

1. Determinar los requerimientos.
2. Determinar funcionalidad.
3. Definir el alcance de software
4. Determinar restricciones
5. Determinar las interfaces del aplicativo
6. Definir los atributos y características de aplicativos

## Alcance del producto

El equipo “ACTIVE”, desarrollará el SRS del proyecto titulado “Campus Finder”, el cual es una App web en la cual la comunidad de la Pontificia Universidad Javeriana pueda utilizar para facilitar sus desplazamientos en el campus universitario.

Para cada usuario que se registre en nuestra App web “Campus Finder” tendrá el rol de usuario normal mediante el cual podrá ver su horario, gestionar notificaciones, calcular rutas desde un edificio a otro; también, se posee el rol de administrador, el cual es el encargado de mantener actualizado los datos correspondientes a cada edificio. Aparte de estos dos roles fundamentales encontramos el rol de usuario externo el cual tendrá funcionalidades limitadas. Los usuarios externos/no registrados podrá ver el mapa del campus universitario y calcular notas, sin embargo, los datos correspondientes de estas consultas no serán guardados en el sistema.

### Funcionalidades

Con el proyecto “Campus Finder”, se quiere establecer un sistema en el cual cualquier miembro de la comunidad educativa de la Pontificia Universidad Javeriana pueda registrarse mediante su correo electrónico y de este modo, puedan buscar una ruta de un edificio a otro, guardar su horario, guardar preferencias e incluso calcular notas.

Para el cálculo de una ruta el usuario debe estar registrado en la App web y debe especificar el número y/o nombre del edificio de inicio y llegada, esto con el fin de que la App calcule la ruta más adecuada según las preferencias de cada usuario, es importante mencionar que se tienen en cuenta las limitaciones físicas que puede tener un miembro de la comunidad educativa (las limitaciones están especificadas en las preferencias de usuario), una vez el sistema muestra la ruta obtenida para el desplazamiento en el campus universitario se le da la opción al usuario de guardar la ruta y/o el edificio en la lista de favoritos, al finalizar la ruta el usuario tiene la opción de calificar los edificios y puntos de interés dados por el sistema.

En la funcionalidad del cálculo de notas existen dos tipos de procedimientos según el usuario que ingrese en esta funcionalidad

1. **Usuario registrado:** El usuario ingresa a la opción de calcular notas, se mostrarán las asignaturas que está cursando en este semestre (según el horario que ingrese el usuario), el usuario selecciona la materia para ingresar las notas y el sistema mostrará una tabla por asignatura donde el usuario ingresa las notas y porcentajes de la materia escogida; es importante destacar que el usuario únicamente podrá guardar sus datos cuando llene los porcentajes de las notas y estos completen un 100%.
2. **Usuario no registrado:** El usuario ingresa a la opción de calcular notas, se mostrarán una tabla para que complete las notas con su respectivo porcentaje, una vez se complete estos datos se realizara el cálculo de la nota, importante destacar que estos usuarios no tendrán la opción de guardar sus notas.

Otra de las funcionalidades de Campus Finder, corresponde a guardar preferencias de los usuarios a través de las opciones de su perfil, esto con el fin de que los usuarios del aplicativo puedan tener una personalización de las rutas suministradas de acuerdo con sus necesidades, es decir, un usuario que se encuentre con alguna limitación física se le mostrará una ruta en la cual se reduzcan al mínimo el número de obstáculos que dificulten su desplazamiento en el campus universitario.

Por último, la función de notificaciones y alarmas, las notificaciones corresponden a eventos que se estén realizando en el medio universitario. Las notificaciones y las alarmas se tienen en cuenta en las preferencias del usuario.

### Beneficios

El proyecto de Campus Finder está desarrollado con el fin de cumplir con los siguientes objetivos y brindando beneficios como:

* Obtener experiencia en el desarrollo de la ingeniería de requisitos.
* Desarrollar una app web que facilite la movilidad, acceso a la información de interés de la comunidad javeriana.
* Brindar rutas optimas de acuerdo con las necesidades del usuario.
* Notificar a los usuarios de los eventos que se estén realizando en los edificios del campus universitario.
* Dar una mejor experiencia a los miembros de la comunidad que no conocen el campus universitario

## Glosario

1. JavaScript: JavaScript es un lenguaje de programación interpretador, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativos débilmente tipados y dinámicos
2. TCP: Transmission Control Protocol (TCP) o protocolo de control de transmisión, es uno de los protocolos fundamentales para internet
3. IP: Dirección IP, el número que identifica a cada dispositivo de una red
4. SRS: Una especificación de requerimientos de software (SRS)es una descripción completa del propósito y el entorno previsto par el software de desarrollo. El SRS describe completamente lo que el software hara y como se espera que se realice.
5. Ram: Sigla de Random Access Memory (memoria de acceso aleatorio), memoria principal de la computadora, donde residen programa y datos, sobre la que se puede efectuar operaciones de lectura y escritura
6. Servidor Web: Un servidor web es un programa que utiliza protocolo de transferencia de hipertexto, HTTP(Hipertext Transfer Protocol), para servir los archivos que forman páginas web a los usuarios, en respuesta a sus solicitudes, que son reenviados por los clientes HTTP de sus computadoras
7. Protocolo: En informática, un protocolo es un conjunto de reglas usadas por computadoras para comunicarse unas con otras a través de una red. Un protocolo es una convención o estándar que controla o permite la conexión, comunicación, y transferencia de datos entre dos puntos finales
8. Open source: La Open Source Initiative (OSI, en español Iniciativa para el Código Abierto) es una organización dedicada a la promoción del código abierto
9. Safari: Safari es un navegador web de código cerrado desarrollado por Apple Inc. Está disponible para macOS, iOS (el sistema usado por el iPhone, el iPod touch y iPad).
10. Firefox: Mozilla Firefox es un navegador web libre y de código abierto desarrollado para Linux, Android, IOS OS X y Microsoft Windows coordinado por la Corporación Mozilla y la Fundación Mozilla.
11. Google Chrome: Google Chrome es un navegador web desarrollado por Google y compilado con base en varios componentes e infraestructuras de desarrollo de aplicaciones (frameworks) de código abierto.

# Descripción global

## Perspectiva de producto

El producto Campus Finder propuesto por el grupo “ACTIVE” es un producto complementario para la aplicación JaveMovil donde se le da al cliente la opción de ubicar puntos de interés en el mapa y posteriormente crear rutas desde donde se encuentra hasta ese punto. Adicionalmente, se le proporcionará información de los edificios en cuanto a los pisos donde podrá encontrar los baños, las entradas y salidas, las oficinas, las facultades entre otros *(*Tabla 13 Descripción entidad edificio*)*.

Esta idea nació por la necesidad de reconocer la ubicación de los edificios que muchas veces son llamados por los números o nombres que lo representan no solo para los estudiantes de primeros semestres sino también para todas las personas nacionales o extranjeras que necesiten ubicarse dentro de las instalaciones de la Pontificia Universidad Javeriana.

Es por lo anterior, que el grupo “ACTIVE” llegó a la conclusión de que era necesario crear un mapa de la universidad de manera que el usuario desde su móvil pueda encontrar la ruta más adecuada para sus preferencias conociendo información de los puntos de interés.

Adicionalmente, el grupo vio la necesidad de tener una calculadora de notas para poder llevar el proceso de las asignaturas en el semestre, llevando así una mejor organización en cuanto a los trabajos. Para esto, el aplicativo contará con la opción de ingresar el horario, de tal manera que el usuario pueda activar alarmas como notificación de clase o de otros compromisos.

Como proyección a futuro, se busca que las aplicaciones CampusFinder y Javemovil se integren para dar mayores funcionales al usuario, brindándole servicios sobre los sistemas de información ya establecidos en la universidad como el XIE o el MAX.

### Interfaces con el sistema

El aplicativo web Campus Finder no contará con sistemas externos ya que sólo necesita el servidor brindado por DigitalOcean especificado en la siguiente sección (Restricciones de memoria) y una base de datos creada únicamente para este fin.

### Restricciones de memoria

Los componentes de este proyecto son el servidor brindado por DigitalOcean el cual cuenta con especificaciones como: 1GB de memoria RAM, 1 procesador, 20 GB de unidad de estado sólido, y 1TB de persistencia (DigitalOcean, s.f.). Estas características son más que suficiente para lograr ejecutar el aplicativo ya que la demanda no es muy grande y la duración de mantenibilidad no es prolongada.

En cuanto al dispositivo que utilice el usuario debe tener las siguientes características: 1 GB de RAM, acceso a internet con soporte a HTML5 con velocidad mínima de 2 Mbps, con resolución mínima de 320X512 pixeles para el funcionamiento correcto del aplicativo y se tenga disponibilidad total de las funcionalidades del sistema.

### Operaciones

En esta sección se explicarán las operaciones más representativas de la aplicación y las características de las operaciones que interactúan tanto con el usuario como con el sistema.

#### Modos de operación

Existen diferentes tipos de usuario, cada uno con características que se encuentran especificadas a profundidad en la sección *7.3 Características del usuario*, y por este motivo la aplicación cuenta con diferentes modos de operación que le presentan a cada tipo de usuario las acciones y la información que le son pertinentes:

1. **Modo invitado/Usuario no registrado**

Este modo presenta el usuario la interfaz principal de la aplicación en la cual puede ver el mapa, calcular notas (con restricciones explicadas en la sección *6.2.1 Funcionalidades*) y el acceso a las interfaces donde puede iniciar sesión o crear una cuenta. Cuando crea una cuenta o inicia sesión pasa al modo usuario o al modo administrador.

1. **Modo usuario/Usuarios registrados**

Este modo le presenta al usuario un conjunto de operaciones adicionales que no están disponibles en el modo invitado, tales como la capacidad de agregar su horario, utilizar la calculadora de notas(sin restricciones explicadas en la sección *6.2.1 Funcionalidades*); guardar, calcular y compartir sus rutas, modificar sus datos de usuario.

1. **Modo administrador**

En este modo la plataforma le ofrece al usuario que ingreso operaciones que no están disponibles para los invitados ni para los usuarios normales, ya que estas operaciones están relacionadas a la información de la base de datos y le ofrece a este usuario la capacidad para modificar esta información.

#### Periodo de actividad e inactividad

La aplicación Campus Finder estará disponible para los usuarios y los desarrolladores las 24 horas del día y solo se presentarán 2 periodos de inactividad.

1. **Primer caso**: La aplicación no estará disponible para los usuarios que se presentara cada 2 semanas en donde los desarrolladores bloquearan el acceso al servidor para realizar operaciones de mantenimiento, respaldar la información de la base de datos, verificar la integridad de la base de datos y del sistema.
2. **Segundo caso:** La aplicación no estará disponible cuando el proveedor del hosting (Digital Ocean) realice operaciones de mantenimiento o se apaguen los servidores por situaciones que se encuentren fuera de su control.

#### Proceso de recuperación

De acuerdo lo especificado en la sección anterior *(7.1.3.2Periodo de actividad e inactividad*), los desarrolladores realizaran operaciones de mantenimiento y respaldo de la información de la base de datos cada 2 semanas. Gracias a la plataforma que se está utilizando para la base de datos (MongoDB), respaldar la información de la misma es fácil y rápido gracias a su esquema de guardar la información en documentos en formato, JSON; por este motivo el respaldo de la información se encontrara en un disco duro externo que se encontrar en posesión del desarrollador líder.

## Funciones del producto

En la presente sección expondrá todas aquellas funciones que la aplicación contará, con el fin de especificar de una mejor forma el alcance que este tendrá y las características con las que dispondrá, además de especificar los servicios que se darán a los usuarios en “Campus Finder”

Para presentar el software de una forma clara y concisa se mostrarán los casos de uso que se determinaron para el desarrollo de la plataforma, un caso de uso representa los requisitos funcionales, especificando su funcionamiento y la forma en que el usuario podrá interactuar con el sistema.

A continuación, se presenta las funciones que el software brindara, para una descripción de su funcionalidad y sus respectivas características, se usara la plantilla descrita en la Tabla 2 Plantilla para los casos de uso.

|  |  |
| --- | --- |
| ***ID*** | Código del caso de uso asociado |
| ***Nombre*** | Nombre del caso de uso |
| ***Actor*** | Actores involucrados |
| ***Precondición*** | Condiciones necesarias para efectuar el caso de uso |
| ***CU Asociado*** | Caso de uso asociados |
| ***Flujo*** | Paso a paso para la ejecución del caso de uso |
| ***Curso alterno*** | Paso a paso alternativo al flujo |
| ***Requerimientos asociados*** | Requisito relacionado al caso de uso |

Tabla 2 Plantilla para los casos de uso

En esta table se especifican las condiciones necesarias para que se realice el caso de uso, además de los procesos conjuntos o alternos que pueden llegar a necesitar para su ejecución y los pasos necesario para desarrollar dicha ejecución, los casos de uso determinados en base a la descripción y alcance del producto detallados en la sección *6.2 Alcance del producto*, los casos de uso se encuentran completamente detallados en la documentación de casos de uso (anexo: **Casos de uso**)

A continuación, se detallan los principales escenarios del proyecto y las funcionalidades generales se pueden evidenciar en cada uno:

* **Login de la página**

El usuario podrá registrarse en la plataforma y se solicitará el nombre, un nombre de usuario, una contraseña y un correo para realizar el registro, si el usuario ya se encuentra registrado este solamente ingresa al sistema

Casos de uso asociados:

* + 1. Ingresar al sistema
    2. Calcular promedio
    3. Ver mapa

Los casos de uso de calcular promedio y ver mapa están disponibles para los usuarios no registrados, de tal forma que puedan acceder a estas funcionalidades sin embargo poseen restricciones ya explicadas en la sección *6.2.1 Funcionalidades*

* **Página inicio de sesión**

Se le solicitan al usuario la autenticación para el inicio de sesión, el usuario también puede ingresar por medio de un botón de registro, para crear un nombre de usuario único con su respectiva contraseña por medio de un correo electrónico en Campus Finder

Casos de uso asociados:

* + 1. Registrarse
    2. Recuperar contraseña
* **Perfil de usuario**

El usuario contara con un perfil en el que se registra información personal como lo es su nombre, datos de contacto, foto, su horario con sus correspondientes materias; también encontramos la posibilidad de eliminar su cuenta.

Casos de uso asociados:

* + 1. Actualizar perfil
    2. Ingresar horario
    3. Eliminar cuenta
* **Menú principal**

En esta página se encuentran las funcionalidades del sistema permitiendo al usuario ingresar en cada una de estas

Casos de uso asociados:

* + 1. Calcular promedio
    2. Ver mapa
    3. Generar ruta
    4. Actualizar perfil

Estos casos de uso disponen de toda la funcionalidad, es decir, los usuarios registrados pueden guardar datos para el uso posterior, aparte de esto pueden generar rutas desde un edificio a otro, en comparación de un usuario no registrado los cuales solo pueden ver mapa y calcular promedios sin la posibilidad de guardar información para su posterior uso de la aplicación.

* **Buscar Edificio**

Se le solicita al usuario que ingrese información básica del edificio, como es el número del edificio y/o su nombre para comenzar la búsqueda en el sistema, para luego retornar la información correspondiente del edificio.

Casos de uso asociados:

* + 1. Consultar edificios
* **Calcular ruta**

Se le solicita al usuario el ingreso de un edificio destino y un edificio de origen, luego de que el usuario ingresa estos datos, se le muestra la ruta optima al usuario según sus preferencias de perfil, cabe destacar que la ruta generada será mostrada en un mapa representativo del campus universitario y se contara con la opción en la cual el usuario puede ver la ruta como texto.

Casos de uso asociados:

* + 1. Generar ruta
* **Calcular notas**

Se le solicita al usuario que ingrese sus notas y porcentajes correspondientes para cada una de acuerdo cada materia cursada, una vez ingresadas las notas y el porcentaje total corresponda a un 100% se procede a calcular las notas de acuerdo con los porcentajes dados.

Casos de uso asociados:

* + 1. Calcular promedio

## Características del usuario

“Campus Finder “posee tres tipos de usuarios, el usuario registrado, el usuario no registrado, el usuario registrado y el administrador. En la *Tabla 3 Usuarios registrados, Tabla 4 Usuario no registrado* y en la *Tabla 5 Usuario administrador se describen las características particulares de cada uno de los usuarios.*

|  |  |
| --- | --- |
| ***Características de usuario*** | ***Usuario registrado*** |
| **Nivel de seguridad o de privilegios** | * Eliminar cuenta * Actualizar perfil * Calcular ruta * Ajustar notificación * Ver mapa * Compartir ubicación * Ajustar notificación * Ingresar notas * Consultar edificio |
| **Roles** | Puede guardar su información para el uso posterior de la aplicación |
| **Nivel de estudios o experiencia técnica** | Conocimiento básico de uso de computadores y/o dispositivos móviles.  Lectura del manual de usuario |
| **Frecuencia de uso** | Cada vez que el usuario quera hacer uso de la aplicación |

Tabla 3 Usuarios registrados

|  |  |
| --- | --- |
| ***Características del usuario*** | ***Usuario no registrado*** |
| ***Nivel de seguridad o de privilegios*** | * Registrar usuario * Ver mapa * Ingresar notas (limitada) |
| ***Roles*** | Este tipo de usuario podrá acceder a funcionalidades básicas del sistema y no podrá guardar sus datos para posteriores usos de la aplicación |
| ***Nivel de estudio o experiencia técnica*** | Conocimiento básico de uso de computadores y/o dispositivos móviles.  Lectura del manual de usuario |
| ***Frecuencia de uso*** | Cada vez que el usuario quera hacer uso de la aplicación |

Tabla 4 Usuario no registrado

|  |  |
| --- | --- |
| ***Características del usuario*** | ***Administrador*** |
| ***Nivel de seguridad o de privilegios*** | * Gestionar información |
| ***Roles*** | Este usuario es el encargado de vigilar continuamente el correcto funcionamiento de la aplicación, es el encargado de resolver cualquier conflicto y de gestionar el sistema |
| ***Nivel de estudio o experiencia técnica*** | Dominio en la administración de servidores bases de datos. Conocimiento hacer del funcionamiento del sistema |
| ***Frecuencia de uso*** | Monitoreo continuo para verificar el funcionamiento del sistema, resolución de conflictos y gestión del sistema. |

Tabla 5 Usuario administrador

## Restricciones

En esta sección se hace referencia a tres tipos de restricciones entre las cuales encontramos: restricciones generales, restricciones de software y restricciones de hardware. A continuación, se describe cada una de las restricciones para cada tipo.

1. **Restricciones generales**
   * El sistema debe ser bilingüe. Debe estar en español en ingles
   * El sistema debe contar con persistencia de datos.
   * El sistema debe contar con una arquitectura de cliente / servidor.
2. **Restricciones de software**
   * El servidor de la aplicación debe soportar NoSQL.
   * Los navegadores de los usuarios deben soportar HTML5.
   * Las licencias de los softwares utilizados son de tipo libre.
3. **Restricciones de hardware**
   * La resolución de pantalla mínima para utilizar la aplicación en modo móvil es de 320x512 pixeles. Y en cuanto a las restricciones de memoria referirse a la sección 7.1.2 Restricciones de memoria

## Modelo de dominio

En esta sección se presenta el diagrama del modelo de dominio en el cual se basará el sistema. En este se plantea la información/conceptos que se logra abstraer para permitir el entendimiento del sistema.

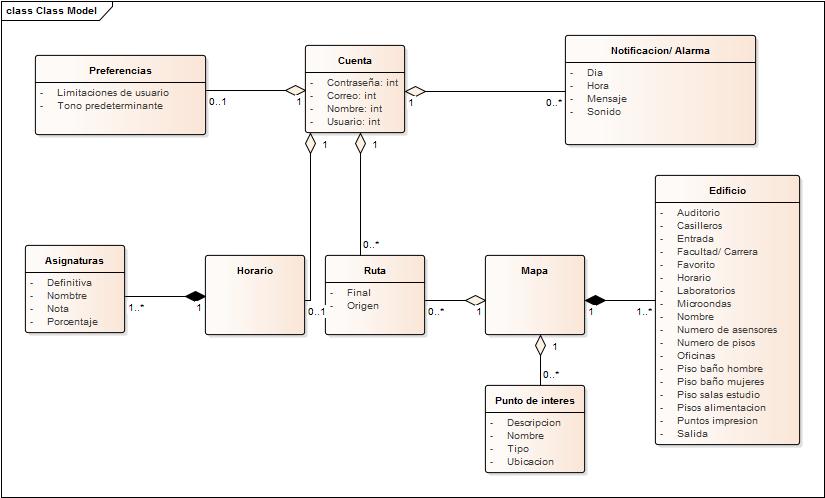


Ilustración 1 Diagrama modelo de dominio

Con base con Ilustración 1 Diagrama modelo de dominio se procede a describir cada entidad/concepto usando Tabla 6 Plantilla para descripción de entidades

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | *identificador único del elemento del dominio del problema.* | | ***Elemento del dominio*** | *indica el nombre del elemento del dominio del problema que será documentado.* | |
| ***Descripción*** | *Contiene una breve descripción del elemento, se debe indicar el por qué se debe crear este elemento dentro del dominio.* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Nombre del atributo a documentar* | | *información del objetivo del atributo dentro del elemento* | | | *De acuerdo con el lenguaje planeado para la implementación del sistema, establecer el tipo de dato que se le asignará al atributo.* |
| ***Objetivo*** | *descripción global acerca del elemento documentado con el fin de exponer su funcionalidad en el sistema* | | | | |

Tabla 6 Plantilla para descripción de entidades

Teniendo en cuenta la tabla anterior se explica cada una de las entidades a continuación:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***1*** | | ***Elemento del dominio*** | *Cuenta* | |
| ***Descripción*** | *Este elemento representa a los usuarios registrados de la aplicación* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Usuario* | | *Nombre único de usuario en el sistema* | | | *Texto* |
| *Contraseña* | | *Contraseña del usuario en el sistema* | | | *Texto* |
| *Correo* | | *Correo de registro para el sistema* | | | *Texto* |
| *Nombre* | | *Nombre real de usuario* | | | *Texto* |
| ***Objetivo*** | *Dar unicidad a cada usuario, evitar dobles registros, y permitir la personalización de la app web* | | | | |

Tabla 7 Descripción entidad cuenta

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***2*** | | ***Elemento del dominio*** | *Preferencia* | |
| ***Descripción*** | *Este elemento contiene las preferencias del usuario para la App web* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Limitaciones de usuario* | | *Representa las limitaciones físicas del usuario, tales como discapacidad visual, discapacidad auditiva, restricciones de movimiento* | | | *Texto* |
| *Tono predeterminante* | | *Representa el sonido de las alarmas y/o notificaciones de la app web hacia el usuario* | | | *Texto* |
| ***Objetivo*** | *Guardar las preferencias del usuario, y sus necesidades para el que se calcule una ruta de un edificio a otro* | | | | |

Tabla 8 Descripción entidad preferencia

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***3*** | | ***Elemento del dominio*** | *Notificación / alarma* | |
| ***Descripción*** | *Este elemento representa las características de una notificación/notificación del sistema hacia el usuario* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Dia* | | *Es el día el cual la notificación/alarma debe ser mostrada al usuario* | | | *Fecha* |
| *Hora* | | *Es la hora a la cual la notificación/alarma debe ser mostrada al usuario* | | | *Fecha* |
| *Mensaje* | | *Representa el motivo de la notificación/alarma* | | | *Texto* |
| *Sonido* | | *Representa el tono predeterminado para esa notificación/alarma* | | | *Texto* |
| ***Objetivo*** | *Guardar las preferencias del usuario, y sus necesidades para el que se calcule una ruta de un edificio a otro* | | | | |

Tabla 9 Descripción entidad Notificación/Alarma

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***4*** | | ***Elemento del dominio*** | *Ruta* | |
| ***Descripción*** | *Este elemento representa la ruta que el usuario desea* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Final* | | *Representa el destino al cual el usuario desea llegar* | | | *Texto* |
| *Origen* | | *Representa el punto de partida de la ruta* | | | *Texto* |
| ***Objetivo*** | *Mostrar la ruta la cual cumple las necesidades del usuario, desde un origen hasta un fin.* | | | | |

Tabla 10 Descripción entidad ruta

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***5*** | | ***Elemento del dominio*** | *Asignatura* | |
| ***Descripción*** | *Representa la asignatura que está cursando el usuario, estas hacen parte de un horario que es único por usuario* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Nombre* | | *Representa el nombre de la asignatura que el usuario está cursando* | | | *Texto* |
| *Nota* | | *Representa la nota obtenida por el usuario* | | | *Entero* |
| *Porcentaje* | | *Representa el porcentaje de la definitiva de la asignatura por cada una de las notas* | | | *Entero* |
| *Definitiva* | | *Representa la definitiva de la asignatura de acuerdo con las notas y porcentajes suministrado por el usuario* | | | *Entero* |
| ***Objetivo*** | *Guardar los datos de la(s) asignatura(s) que está cursando el usuario según su horario* | | | | |

Tabla 11 Descripción entidad Asignatura

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***6*** | | ***Elemento del dominio*** | *Punto de interés* | |
| ***Descripción*** | *Este elemento representa las características de una notificación/notificación del sistema hacia el usuario* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Descripción* | | *Descripción general del punto de interés* | | | *Texto* |
| *Nombre* | | *Nombre del punto de interés* | | | *Texto* |
| *Tipo* | | *Tipo de punto de interés en el campus universitario* | | | *Texto* |
| *Ubicación* | | *Es el punto donde se encuentra el punto de interés* | | | *Texto* |
| ***Objetivo*** | *Representar los puntos de interés en el campus universitario* | | | | |

Tabla 12 Descripción entidad puntos de interés

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***7*** | | ***Elemento del dominio*** | *Edificio* | |
| ***Descripción*** | *Este elemento representa los edificios que conforman el campus universitario* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Auditorio* | | *Representa los auditorios que pueden contener los edificios* | | | *Texto* |
| *Casillero* | | *Representa los casilleros que pueden contener los edificios* | | | *Texto* |
| *Entrada* | | *Representa el piso donde se encuentra la entra que posee el edificio* | | | *Texto* |
| *Salida* | | *Representa el piso donde se encuentra la salida que posee el edificio* | | | *Texto* |
| *Facultad/Carrera* | | *Es la facultad/carrera la cual se encuentra en el edificio* | | | *Texto* |
| *Horario* | | *Es el horario de apertura y cierre del edificio* | | | *Texto* |
| *Laboratorios* | | *Son los laboratorios que puede contener el edificio* | | | *Texto* |
| *Microondas* | | *Representa los pisos donde se encuentran los microondas.* | | | *Texto* |
| *Nombre* | | *Nombre del edificio* | | | *Texto* |
| *Numero de ascensores* | | *Representa la cantidad de ascensores que contiene un edificio* | | | *Entero* |
| *Número de pisos* | | *Representa la cantidad máxima de pisos que posee un ascensor* | | | *Entero* |
| *Oficinas* | | *Representa la ubicación y el tipo de oficina que se encuentra en ese edificio* | | | *Texto* |
| *Favorito* | | *Especifica si es un edificio favorito para el usuario* | | | *booleano* |
| *Piso baño hombre* | | *Ubicación de los baños masculinos en un edificio* | | | *Texto* |
| *Piso baño mujeres* | | *Ubicación de los baños femeninos en un edificio* | | | *Texto* |
| *Piso sala estudio* | | *Ubicación de las salas destinadas para el estudiar de la comunidad educativa* | | | *Texto* |
| *Piso alimentación* | | *Ubicación de los puntos de alimentación del edificio* | | | *Texto* |
| *Puntos de impresión* | | *Ubicación de los puntos de impresión del edificio* | | | *Texto* |
| ***Objetivo*** | *Busca representar la información relevante para los usuarios de un edificio* | | | | |

Tabla 13 Descripción entidad edificio

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***8*** | | ***Elemento del dominio*** | *Mapa* | |
| ***Descripción*** | *Este elemento corresponde al mapa que se muestra en la aplicación* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Rutas* | | *Rutas solicitadas por el usuario para llegar de un edificio a otro* | | | *Texto* |
| *Puntos de interés* | | *Representa los puntos de interés que se encuentran en el campus universitario* | | | *Texto* |
| *Edificio* | | *Representa los edificios del campus universitario* | | | *Texto* |
| ***Objetivo*** | *Represente el mapa del campus universitario, en el cual se le muestra al usuario la ruta de un edificio a otro, puntos de interés* | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ***ID*** | ***9*** | | ***Elemento del dominio*** | *Horario* | |
| ***Descripción*** | *Este elemento corresponde al horario del usuario registrado* | | | | |
| ***Atributos*** | | | | | |
| ***Nombre*** | | ***Descripción*** | | | ***Tipo de dato*** |
| *Asignaturas* | | *Representa las asignaturas que está cursando el usuario* | | | *Texto* |
| ***Objetivo*** | *Representar los puntos de interés en el campus universitario* | | | | |

## Suposiciones y dependencias

Todo sistema posee factores y/o circunstancias fuera de su entorno que pueden afectar el correcto funcionamiento de este y no se encuentra bajo el control del equipo de desarrollo. A continuación, se definen las consideraciones necesarias para evitar el mal funcionamiento del sistema.

* El sistema proporcionara sus funcionalidades completas para los usuarios registrados, a los usuarios no registrados solo les dejara hacer un cálculo de notas y ver el mapa (sin posibilidad de guardar datos para futuro uso de la aplicación).
* El sistema operara en dispositivos con acceso a internet.
* El sistema funcionara bajo los parámetros de interfaces de hardware descritos en la sección *8.1.2 Interfaces con el hardware.*
* El sistema funcionara en los equipos de las salas de computo de la Facultad de Ingeniería de la Pontificia Universidad Javeriana.
* El sistema se realiza con el fin de ayudar a la movilidad y de la comunidad educativa y sin ningún fin comercial.
* El cliente será fundamental para la recolección de requisitos que se encuentran en más detalle en la sección *7.4 Restricciones* .
* Los servidores funcionan sin contratiempo durante el desarrollo y entrega final del sistema.
* La app web será intuitiva y de rápido aprendizaje para el manejo de la misma
* Los usuarios del sistema no estarán en la facultad de agregar nuevas funcionalidades al sistema.
* Los navegadores soportados por el sistema deberán operar sin ningún inconveniente en los dispositivos del usuario.
* El sistema necesitara una conexión a internet sin interrupción.

## Distribución de requisitos

En esta sección se realiza la relación entre los casos de uso establecidos con los requerimientos asociados y la distribución de los casos de uso, de modo que se pueda identificar fácilmente cuales módulos del proyecto corresponde a cada uno.

En la distribución de casos de uso se encuentra tres aspectos importantes: funcionalidad, involucrado y el caso de uso. La funcionalidad se determina de acuerdo con lo que el grupo ACTIVE quiere desarrollar con el proyecto, mientras que el involucrado está determinado por los siguientes aspectos:

* **Usuario:** Rol que desempeñan los miembros de la comunidad educativa, y aquellos individuos que quieren acceder al sistema.
* **Administrador**: Rol que desempeñan los miembros de ACTIVE con el fin de mantener el correcto funcionamiento de App.
* **Base de datos**: Colección de datos que permite almacenar la información de los usuarios de la App.
* **Plataforma**: Sistema por medio del cual los usuarios ingresan al sistema.
* **Servidor**: Programa que permite el funcionamiento.

Los casos de uso se especifican en el anexo: **Casos de uso**, en el cual se encuentra el respectivo diagrama de casos de uso con su respectiva documentación. De acuerdo con lo anterior se realiza la relación entre casos de uso con sus respectivos requerimientos, lo cual también se encuentra explicado en el *anexo:* **Casos de uso**

# Requisitos específicos

## Requisitos de interfaces externas

### Interfaces con el usuario

“Campus Finder” es una app web que permitirá al usuario de forma fácil y rápida buscar rutas entre diferentes edificios del campus universitario; para ello se establecen que las interfaces de comunicación del usuario con el software deben ser de la manera más sencilla posible, para que usuario pueda acceder a través de un computador, laptop o incluso un dispositivo móvil. Con esto, se planea que la interfaz GUI del sistema sea adaptada a cualquier dispositivo, dado así dos interacciones con el sistema según el dispositivo:

* **Interacción táctil:**

Se encuentra en dispositivos tales como tablets, celulares e incluso en algunos portátiles con pantalla táctil, este tipo de interacción está ligada al toque a través de la pantalla.

* **Interacción por hardware:**

Este tipo de interacciones se encuentran en dispositivos como computadores de escritorio, portátiles sin pantalla táctil, por lo cual es necesario que la interacción se dé por medio de un teclado, touch pad y un mouse/ratón(periféricos).

Teniendo definidas las formas que el usuario interactuara con el sistema, cabe resaltar que la interfaz GUI del sistema le ofrece usuario será muy importante, por lo cual se le atribuye un requerimiento específicamente para lograr que la interfaz se adapte a el tipo de pantalla, a esto se le conoce como interfaz responsive la cual consiste en redimensionar y colocar los elementos de la app web de forma que se adapten al ancho y alto de la pantalla de la cual se esté visualizando para dar comodidad al usuario al momento de utilizar la app web, por otro lado se realiza la planeación de las interfaces por medio de mockups los cuales se encuentran en el anexo: **Mockups**

### Interfaces con el hardware

Las interfaces de hardware para el uso de Campus Finder se dividen en dos dependiendo del tipo de dispositivo, ya sea Pc o un dispositivo móvil.

* **PC:** Para PC Campus Finder se puede utilizar e cualquier sistema operativo sea Windows, Linux o Mac, es imprescindible que este tenga una conexión continua a internet, y cuente con un navegador web el cual se encuentra especificado en la sección *8.1.3 Interfaces con el software*, además, que este implemente el protocolo HTTP/IP para la comunicación con el servidor tal como está descrito en la sección *8.1.4* Interfaces de comunicación. Además, el PC debe contar con los medios de interacción descritos en la sección *8.1.1* Interfaces con el usuario*.*
* **Dispositivos móviles:** Para el caso de dispositivos móviles Campus Finder posee independencia del sistema operativo que contenga el dispositivo móvil, y al igual que en PC es necesario que el dispositivo cuente con una conexión continua a internet y un navegador web descrito en la sección *8.1.3 Interfaces con el software*; igualmente, estos dispositivos deben contar con una pantalla táctil ya que no cuentan con periféricos para la interacción como es el caso de PC

### Interfaces con el software

A continuación, se presentan todas las interfaces de software necesarias para la implementación, funcionamiento e interacción con la aplicación, con una pequeña descripción y el propósito que cumple la interfaz dentro de la aplicación:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Nombre | Descripción | Versión | Propósito | Fuente |
| DigitalOcean cloud server | DigitalOcean es un proveedor de infraestructura para desarrollo de aplicaciones en la nube, que cuenta con herramientas para crear servidores rápidamente sobre un disco de estado sólido que monta una máquina virtual. | No disponible | Servicio que nos provee la infraestructura donde se alojara la aplicación. | https://www.digitalocean.com/help/ |
| Ubuntu server OS | Ubuntu server es una variante de Ubuntu OS, sistema operativo GNU/Linux basado en la arquitectura de Debian. Ubuntu es una de las distribuciones de Linux más populares para computadores de escritorio y servidores gracias a la extensa documentación que se encuentra disponible y el amplio soporte que las empresas y la comunidad ofrece. | 16.04 | Sistema operativo en el que se encuentran la interfaz, la base de datos y todos los componentes que son necesarios para el funcionamiento de la aplicación. | https://www.ubuntu.com/server |
| MongoDB | MongoDB es una base de datos NoSQL orientada a documentos que no requiere de un esquema fijo como las tablas que se utilizan en las bases de datos relacionales. Gracias a su flexibilidad es fácil de implementar para proyectos de cualquier complejidad. | 3.4 | Es la base de datos en la cual guardara la información de todos los componentes que necesiten de persistencia en la aplicación. | https://docs.mongodb.com/manual/ |
| 1. Fi   Firefox  (Navegador) | Firefox es un navegador web de código abierto multiplataforma, programado pensando en la seguridad del usuario y en la velocidad de navegación. Firefox cuenta con una amplia gama de opciones para desarrolladores web y posee una gran cantidad de complementos que ayudan al usuario | 56.0.1 | Se utiliza como una interfaz mediante la cual los usuarios del sistema podrán acceder a la plataforma y utilizar la aplicación. | https://www.mozilla.org/en-US/firefox/products/?v=b |
| Safari  (Navegador) | Safari es el navegador web por defecto para los productos de Apple. Muchos usuarios utilizan computadores de escritorio Mac, IPhone y IPad; dispositivos que son utilizados por una parte significativa de los usuarios objetivo de la aplicación. | 8.0.2 | Se utiliza como una interfaz mediante la cual los usuarios del sistema podrán acceder a la plataforma y utilizar la aplicación. | https://www.apple.com/safari/ |
| Google Chrome  (Navegador) | Navegador web multiplataforma más popular del mundo, desarrollado por Google y cuenta con herramientas que facilitan el desarrollo web. Se caracteriza por ser rápido, intuitivo y cuenta con facilidades para los usuarios que utilicen los demás servicios de Google | 63.0 | Se utiliza como una interfaz mediante la cual los usuarios del sistema podrán acceder a la plataforma y utilizar la aplicación. | https://www.google.com/chrome/browser/features.html |
| JavaScript | Es un lenguaje de programación interpretado, utilizado principalmente para definir los eventos y los resultados de la interacción de un usuario con una página web. JavaScript interactúa con los elementos escritos en HTML y CSS para hacer una página web dinámica e interactiva. JavaScript puede interactuar con otras aplicaciones como base de datos. | 1.8.5 - ECMAScript 2015 | Se utiliza como el lenguaje principal de programación para implementar la mayoría de la lógica de la aplicación. | https://developer.mozilla.org/en-US/docs/Learn/Getting\_started\_with\_the\_web/JavaScript\_basics |
| Node.js | **Node.js** 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. | 6.11.4 | Librería necesaria para manejar las conexiones de los clientes con el servidor. | https://nodejs.org/en/about/ |

Para más información acerca de todas las herramientas que se utilizaran para el desarrollo del proyecto, remítase al documento SPMP, sección 6.2 Lenguajes y Herramientas.

### Interfaces de comunicación

Para las conexiones de los usuarios al servidor de Campus Finder, se le dio prioridad al momento de diseñar las conexiones que las comunicaciones entre ambas partes se realicen lo más rápidamente posible sin necesidad de transmitir más información de la que se está pidiendo. Por este motivo se usarán solamente los protocolos TCP/IP (Transmission Control Protocol / Internet Protocol) y HTTP (Hyper Text Transfer Protocol).

## Características del producto software

Las características del software están definidas por los requerimientos funcionales de este. Estos son los que determinan las características útiles que el sistema debe poseer y cumplir, los cuales, se encuentran definidos en el anexo 4: Requisitos, sin embargo, se muestran a continuación los requisitos más importantes del sistema:

## Requisitos de desempeño

Los requerimientos de desempeños descritos en esta sección hacen referencia a todos aquellos requisitos relacionados con números de solicitudes, cantidades de usuarios soportadas entre otros, entre los cuales encontramos los siguientes:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF37 | **Involucrado** | Servidor |  |
| **Descripción** | El sistema debe permitir el acceso simultaneo a mínimo 200 usuario | | | |
| **Razón** | Permitir el acceso a la plataforma a un gran número de usuarios al mismo tiempo | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Se realizan solicitudes al servidor y se verifican las respuestas por parte de este | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF38 | **Involucrado** | Base de datos |  |
| **Descripción** | La base de datos debe tener la capacidad de registrar a 1000 usuarios | | | |
| **Razón** | Soportar un gran número de usuarios para el uso de la plataforma | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Se realizan solicitudes al servidor y se verifican las respuestas por parte de este | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF40 | **Involucrado** | Plataforma |  |
| **Descripción** | El sistema debe estar disponible 24 horas al día | | | |
| **Razón** | Brindar acceso al sistema en cualquier momento que usuario quiera acceder a este | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Acceder a cualquier momento del día al sistema y verificar su adecuado funcionamiento | | | |
| **Versión** | 1.1 | | | |

## Restricciones de diseño

En la *Tabla 14 Restricciones de diseño* se describen las restricciones generales, de hardware y software para el desarrollo de la app Campus Finder, la cual fueron determinados por el scrum team y permiten cumplir con el alcance del proyecto descrito en la sección *6.2* Alcance del producto

|  |  |
| --- | --- |
| ***Elementos*** | ***Restricciones*** |
| ***Generales*** | * El sistema debe desarrollarse dentro de las fechas estipuladas por el product Owner * El sistema funciona únicamente con conexión continua a internet * Las funcionalidades establecidas en el SRS se deben cumplir * El sistema debe ser desarrollado en español |
| ***Hardware*** | * El sistema debe soportar entrada por mouse y por pantalla táctil. * Los servidores del sistema deben tener conexión a internet |
| ***Software*** | * Los lenguajes de programación en los cuales la app está desarrollado son: HTML, CSS, JavaScript * Los navegadores en los cuales está disponible la app son: Safari, Firefox y Google Chrome (para más información remitirse a la sección 8.1.3 Interfaces con el software) * El sistema debe tener arquitectura cliente-servidor |
| ***Herramientas CASE*** | * Todas las herramientas que se utilicen no deben generar gastos para el grupo ACTIVE * Las herramientas utilizadas deben ser softwares libres |

Tabla 14 Restricciones de diseño

Las restricciones previamente mencionadas se complementan con las restricciones establecidas en el documento SPMP, sección *7.4*Restricciones, además de las restricciones establecidas en la sección *8.1* Requisitos de interfaces externas

## Atributos del sistema software

En la siguiente sección se detallará los requerimientos no funcionales que le otorgan al sistema calidad y garantizan la funcionalidad del sistema. Adicionalmente, los requerimientos a continuación otorgan facilidad de uso, calidad de la experiencia del usuario y estabilidad al sistema y establecen políticas de implementación del sistema. Estos son listados a continuación:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF1 | **Involucrado** | Usuario | **CU1** |
| **Descripción** | El sistema debe registrar clientes | | | |
| **Razón** | El sistema necesita conocer la información de los clientes para poder brindarle todos sus servicios | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Registrando cuentas y verificando su existencia en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF2 | **Involucrado** | Base de datos | **CU2** |
| **Descripción** | El sistema debe verificar clientes | | | |
| **Razón** | Para mantener la integridad de la base de datos es necesario verificar que la información de los clientes sea consistente | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar datos de prueba erróneos para que el sistema lo reconozca y según eso se evalúa cuantos reconoce. Por lo tanto, se verifica si está correcto o no | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF5 | **Involucrado** | Administrador | **CU21** |
| **Descripción** | El sistema debe permitir gestionar la información guardada de los puntos de interés | | | |
| **Razón** | La planta física de la universidad y los puntos de interés (edificios, cafeterías, maquinas, etc.) pueden cambiar, lo que modifica la información que se tiene de estos en la base de datos | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Con versiona miento según la fecha de actualización del punto | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF6 | **Involucrado** | Usuario | **CU12, CU23** |
| **Descripción** | El sistema debe calcular las rutas solicitadas | | | |
| **Razón** | El objetivo principal de la aplicación web es proporciona rutas al cliente para trasladarse dentro del campus | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Solicitar una ruta y verificar si la retorna o no | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF8 | **Involucrado** | Usuario | **CU3** |
| **Descripción** | El sistema debe calcular el promedio de las notas ingresadas | | | |
| **Razón** | El cliente podrá verificar sus notas y calcular su promedio para que lleve un seguimiento de su situación académica | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar notas y calcular el promedio verificando si el resultado es correcto o no. A manera que si es incorrecto deberá ser corregido | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF9 | **Involucrado** | Base de datos | **CU12, CU13, CU17,** |
| **Descripción** | El sistema debe proveer la información especificada a los usuarios | | | |
| **Razón** | El usuario debe haber hecho alguna búsqueda | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Hacer una búsqueda y verificar la veracidad de la información | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF10 | **Involucrado** | Plataforma | **CU6, CU12** |
| **Descripción** | El sistema debe mostrar gráficamente la ruta solicitada | | | |
| **Razón** | Es más llamativo y fácil de comprender para el usuario ver la ruta que debe seguir a manera de gráfico que en texto | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Solicitar ruta y verificar si concuerda lo graficado con lo informado | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF11 | **Involucrado** | Plataforma | **CU6** |
| **Descripción** | El sistema debe mostrar gráficamente un mapa de la universidad | | | |
| **Razón** | Se considera una interfaz más amigable con el usuario en cuanto a la facilidad de entender la ubicación y las distancias | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Verificar si se muestra el mapa al momento de accionar el botón de Ver mapa | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF13 | **Involucrado** | Usuario | **CU4** |
| **Descripción** | El sistema debe permitir ingresar notas | | | |
| **Razón** | El cliente podrá verificar sus notas y hacer seguimiento de la materia vista. esto es necesario para posteriormente calcular el promedio del estudiante | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Guardar notas de prueba y verificar la persistencia en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF14 | **Involucrado** | Usuario | **CU8** |
| **Descripción** | El sistema debe permitir ingresar el horario | | | |
| **Razón** | El cliente debe haberse registrado antes y debe tener por lo menos una asignatura | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar un horario y verificarlo en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF15 | **Involucrado** | Usuario | **CU12, CU23** |
| **Descripción** | El sistema debe permitir calcular la ruta entre dos sitios de interés o desde la ubicación actual | | | |
| **Razón** | El cliente haber escrito por lo menos el punto de destino | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Solicitar ruta y verificar la veracidad de la información dada | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF16 | **Involucrado** | Usuario | **CU16, C14** |
| **Descripción** | El sistema debe permitir modificar el horario guardado | | | |
| **Razón** | El cliente debe haberse registrado antes y haber guardado un horario. Con el fin de que el usuario pueda mantener actualizado su horario de semestre en semestre | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Modificar un horario y verificar la actualización en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF18 | **Involucrado** | Base de Datos | **CU1, CU2, CU8, CU7,CU15** |
| **Descripción** | El sistema debe diferenciar entre los clientes registrados y los no registrados | | | |
| **Razón** | Esto es para proporcionar las funcionalidades dependiendo el usuario | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Verificar el usuario que ingresa si está en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF19 | **Involucrado** | Usuario | **CU17, CU20, CU18** |
| **Descripción** | El sistema debe guardar los edificios destacados por el cliente | | | |
| **Razón** | El usuario debe haber buscado algún edificio | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar 10 edificios y verificar en la base de datos su existencia | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF22 | **Involucrado** | Base de Datos | **CU21** |
| **Descripción** | El sistema no debe permitir el registro de administradores | | | |
| **Razón** | Sólo debe existir un administrador de manera que sea el único que pueda modificar la información almacenada de los edificios y puntos de interés | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar puntos de interés/ edificios y revisar si existen en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF27 | **Involucrado** | Usuario | **CU14, CU16** |
| **Descripción** | El sistema debe permitir que el usuario ingrese un horario | | | |
| **Razón** | El usuario debe estar registrado de tal manera que pueda llevar un control de las asignaturas que está cursando y de esa manera las notas | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar un horario y verificar su existencia en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RE28 | **Involucrado** | Base de Datos | **CU1, CU5, CU11, CU19, CU20** |
| **Descripción** | El sistema debe comunicarse con la base de datos creada | | | |
| **Razón** | Para poder guardar la información y mantener la persistencia | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Verificar la información que retorna una solicitud hacia la base de datos, si genera error significa que no hay base de datos conectada | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RE29 | **Involucrado** | Base de Datos | **CU1, CU5, CU11, CU19, CU20** |
| **Descripción** | El sistema debe actualizar la base de datos cada vez que el usuario realiza acciones de almacenamiento | | | |
| **Razón** | Es importante mantener la consistencia de la información que ingresa el usuario y lo que se recupera una vez guardada la información | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Realizar cambios en la base | | | |
| **Versión** | 1.1 | | | |

### Usabilidad

Son los atributos que miden la facilidad de uso y la experiencia del usuario con el sistema. Para determinar la funcionalidad se analizan los aspectos del sistema con los que interactúa el usuario con las interfaces, o la forma como el propio sistema maneja sus propias características para lograr involucrarse con el usuario

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF3 | **Involucrado** | Usuario | **CU14** |
| **Descripción** | El sistema debe permitir a los clientes la modificación de su perfil | | | |
| **Razón** | La información de los clientes puede cambiar en cualquier momento y la aplicación usa la información más reciente del cliente en la base de datos | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Por medio de casos de prueba sacando el promedio entre fallos y éxitos de actualización | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF4 | **Involucrado** | Usuario | **CU22** |
| **Descripción** | El sistema debe permitir configurar alarmas o notificaciones | | | |
| **Razón** | Los clientes quieren llevar un seguimiento de sus rutas y ser notificados cuando deban ir a clase | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Configurar diferentes alarmas y promediar cuántos suenan en el tiempo configurado y cuáles no. Con eso es posible medir la veracidad de lo solicitado por el requerimiento | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF7 | **Involucrado** | Usuario | **CU10** |
| **Descripción** | El sistema debe permitir compartir rutas | | | |
| **Razón** | Ya que muchos clientes no conocen en su totalidad la universidad. y para mejor ubicación y traslado dentro del campus otro cliente podrá compartir su ruta. así se beneficiarán mutuamente | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Compartiendo 10 rutas y verificando cuántas llegan correctamente y calificar la veracidad de lo solicitado. Si es menos de 8 se considera incorrecto y deberá ser nuevamente elaborado | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF12 | **Involucrado** | Base de Datos | **CU15** |
| **Descripción** | El sistema debe permitir que el usuario elimine su cuenta | | | |
| **Razón** | En caso de que el usuario no desee seguir en el sistema | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Borrar 5 perfiles y verificar cuántos de ellos se borraron y según eso revisar la veracidad de la acción | | | |
| **Versión** | 1.1 | | | |
| **Id requerimiento** | RF20 | **Involucrado** | Usuario | **CU17, CU20** |
| **Descripción** | El sistema debe permitir que el usuario destaque edificios | | | |
| **Razón** | El usuario puede mantener registro de su edificio más frecuentados e incluso los favoritos | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Poner casos de prueba y verificar que en la base de datos aparezca el edificio destacado | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF21 | **Involucrado** | Base de Datos | **CU11** |
| **Descripción** | El sistema debe permitir guardar las rutas favoritas del cliente | | | |
| **Razón** | El sistema puede mantener registro de sus rutas favoritas de desplazamiento | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Poner casos de prueba y verificar que en la base de datos aparezca las rutas destacadas | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF26 | **Involucrado** | Plataforma | **CU1, CU3, CU4, CU5, CU9** |
| **Descripción** | El sistema debe proporcionar mensajes de error comprensibles para el usuario final en base a las acciones realizadas por el mismo. | | | |
| **Razón** | Mantener al usuario notificado brindándole mejor experiencia de usuario | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Realizar pruebas de error para que se muestren los mensajes y dependiendo de la respuesta que tenga el sistema | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF30 | **Involucrado** | Plataforma | **Caso de uso asociado** |
| **Descripción** | El sistema debe notificar al usuario sobre los cambios que se haga a su información o actividades | | | |
| **Razón** | El usuario debe estar registrado para recibir estas notificaciones | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Modificar algo en la base de datos y esperar a que notifique al usuario para comprobar que este requerimiento se cumple | | | |
| **Versión** | 1.1 | | | |

### Rendimiento

Es un factor que determina la capacidad del sistema de responder a las peticiones que se le puedan realizar a este, se puede tomar como marco de referencia la velocidad de procesamiento, el tiempo de respuesta, el consumo de recursos. Estos requerimientos también son clasificados como requerimientos de desempeño los cuales se pueden evidenciar en la sección 8.3 Requisitos de desempeño

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF25 | **Involucrado** | Plataforma |  |
| **Descripción** | El sistema debe establecer que el usuario tenga conexión a internet | | | |
| **Razón** | El sistema hará uso de un servidor y de una base de datos las cuales solo tiene conexión por medio de internet | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Hacer ping al dispositivo del usuario que se intentó conectar | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF32 | **Involucrado** | Plataforma | **CU1, CU5, CU11, CU19, CU20, CU21** |
| **Descripción** | El sistema debe operar en dispositivos con una conexión mínima de 1 Mbps | | | |
| **Razón** | Para asegurar que la interacción entre la aplicación y el usuario no se vea comprometida | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar desde los dispositivos móviles de los miembros del grupo ACTIVE verificando que en todos funciona la aplicación | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF33 | **Involucrado** | Servidor | **CU1, CU5, CU11, CU19, CU20, CU21** |
| **Descripción** | El servidor debe contar mínimo con 1GB de RAM | | | |
| **Razón** | Para asegurar que la interacción entre la aplicación y el usuario no se vea comprometida y que la información de la base de datos no se vea afectada | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar desde los dispositivos móviles de los miembros del grupo ACTIVE verificando que en todos funciona la aplicación y se almacenan correctamente los datos en la base de datos | | | |
| **Versión** | 1.1 | | | |

### Soporte

Este factor determina lo que es necesario para hacer uso del sistema, específicamente los requerimientos que requieran una instalación y/o configuración, así como facilidades para mantener y administrar las operaciones del sistema

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF34 | **Involucrado** | Plataforma |  |
| **Descripción** | El sistema debe ser versionado mediante Git | | | |
| **Razón** | Mantener un control del código para facilitar la configuración y desarrollo del mismo | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Evidenciar el versiona miento en el repositorio del proyecto | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF35 | **Involucrado** | Plataforma |  |
| **Descripción** | El sistema debe ser implementado para navegadores web únicamente con HTML5 y JavaScript | | | |
| **Razón** | Permitir el uso del sistema en los navegadores actuales | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Probar el sistema en un antiguo y desactualizado | | | |
| **Versión** | 1.1 | | | |

### Implementación (+)

La implementación de FURPS+ es un factor que determina los requisitos necesarios para ejecutar el sistema que se desarrolló, tomando como factor fundamentar la construcción o codificación de un sistema con estándares, lenguajes, políticas de integración a la base de datos, limitaciones de recurso y ambientes de operación:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF17 | **Involucrado** | Plataforma | **Caso de uso asociado** |
| **Descripción** | El sistema debe acceder a todas las funcionalidades de la aplicación en no más de 5 clics. | | | |
| **Razón** | Es una recomendación para la buena experiencia de usuario | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Con casos de prueba verificando cuántos clics costó de ir de una funcionalidad a otra | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF24 | **Involucrado** | Plataforma |  |
| **Descripción** | El sistema debe estar configurado en español Latam y en inglés americano | | | |
| **Razón** | El idioma de la población objetivo es el español Latam y como una buena opción para los estudiantes extranjeros es el inglés americano | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Cambiar de idioma dentro de la aplicación verificando que tanto comandos como mensajes estén en el idioma ajustado | | | |
| **Versión** | 1.1 | | | |

## Requisitos de la base de datos

En la siguiente sección se especificará los requisitos para la base de datos. Estos están definidos en el *Anexo* **Requisitos**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF2 | **Involucrado** | Base de datos | **CU2** |
| **Descripción** | El sistema debe verificar clientes | | | |
| **Razón** | Para mantener la integridad de la base de datos es necesario verificar que la información de los clientes sea consistente | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar datos de prueba erróneos para que el sistema lo reconozca y según eso se evalúa cuantos reconoce. Por lo tanto, se verifica si está correcto o no | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF9 | **Involucrado** | Base de datos | **CU12, CU13, CU17,** |
| **Descripción** | El sistema debe proveer la información especificada a los usuarios | | | |
| **Razón** | El usuario debe haber hecho alguna búsqueda | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Hacer una búsqueda y verificar la veracidad de la información | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF12 | **Involucrado** | Base de Datos | **CU15** |
| **Descripción** | El sistema debe permitir que el usuario elimine su cuenta | | | |
| **Razón** | En caso de que el usuario no desee seguir en el sistema | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Borrar 5 perfiles y verificar cuántos de ellos se borraron y según eso revisar la veracidad de la acción | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF18 | **Involucrado** | Base de Datos | **CU1, CU2, CU8, CU7,CU15** |
| **Descripción** | El sistema debe diferenciar entre los clientes registrados y los no registrados | | | |
| **Razón** | Esto es para proporcionar las funcionalidades dependiendo el usuario | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Verificar el usuario que ingresa si está en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF22 | **Involucrado** | Base de Datos | **CU21** |
| **Descripción** | El sistema no debe permitir el registro de administradores | | | |
| **Razón** | Sólo debe existir un administrador de manera que sea el único que pueda modificar la información almacenada de los edificios y puntos de interés | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Ingresar puntos de interés/ edificios y revisar si existen en la base de datos | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RF21 | **Involucrado** | Base de Datos | **CU11** |
| **Descripción** | El sistema debe permitir guardar las rutas favoritas del cliente | | | |
| **Razón** | El sistema puede mantener registro de sus rutas favoritas de desplazamiento | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Poner casos de prueba y verificar que en la base de datos aparezca las rutas destacadas | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RE28 | **Involucrado** | Base de Datos | **CU1, CU5, CU11, CU19, CU20** |
| **Descripción** | El sistema debe comunicarse con la base de datos creada | | | |
| **Razón** | Para poder guardar la información y mantener la persistencia | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Verificar la información que retorna una solicitud hacia la base de datos, si genera error significa que no hay base de datos conectada | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RE29 | **Involucrado** | Base de Datos | **CU1, CU5, CU11, CU19, CU20** |
| **Descripción** | El sistema debe actualizar la base de datos cada vez que el usuario realiza acciones de almacenamiento | | | |
| **Razón** | Es importante mantener la consistencia de la información que ingresa el usuario y lo que se recupera una vez guardada la información | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Realizar cambios en la base | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF38 | **Involucrado** | Base de datos |  |
| **Descripción** | La base de datos debe tener la capacidad de registrar a 1000 usuarios | | | |
| **Razón** | Soportar un gran número de usuarios para el uso de la plataforma | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Se realizan solicitudes al servidor y se verifican las respuestas por parte de este | | | |
| **Versión** | 1.1 | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** | RNF41 | **Involucrado** | Base de datos |  |
| **Descripción** | El sistema debe actualizar la información en menos de 10 segundos | | | |
| **Razón** | Brindar a los usuarios información actualizada | | | |
| **Autor** | Active | | | |
| **Criterio de medición** | Toma de tiempo de la actualización de la información | | | |
| **Versión** | 1.1 | | | |

Adicionalmente, el sistema cuenta con un API REST para lograr la comunicación entre el front y el back end. Este esta especificado en la documentación adjunta al prototipo “Finder API”

# Proceso de ingeniería de requisitos

## Planeación

En esta sección se presenta el plan de administración de requerimientos utilizados para el proyecto “Campus Finder”, además de exponer los mecanismos utilizados para desarrollar el proceso de ingeniería de requisitos.

### Plan de administración de requisitos de software

En esta sección se describe como se analizara, documentara y gestionar los requisitos. Para cumplir con estos aspectos, el grupo “ACTIVE” considera oportuno seguir el esquema planteado para la recolección de requerimientos que ofrece el PMBOOK, el cual se muestra en la *Ilustración 2Recolección de requisitos PMBOOK*  en la cual se especifica el flujo que gestiona el levantamiento de requisitos con sus respectivas entradas, herramientas y salidas después de realizar el levantamiento de requisitos

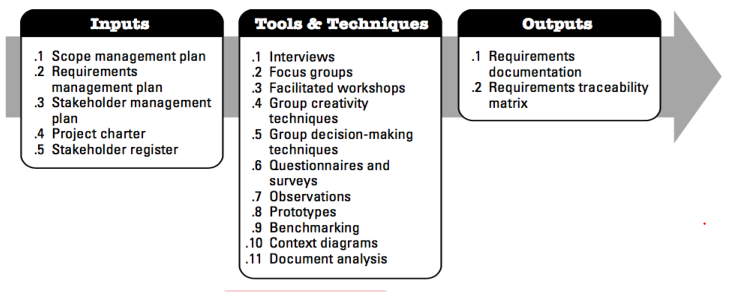


Ilustración 2Recolección de requisitos PMBOOK

### Frecuencia de planeación

Puesto que el desarrollo del sistema se trabaja mediante la metodología Scrum, y esta es una metodología incrementa, la planeación de los requisitos se hace de acuerdo a la actividades desarrolladas en cada una de las metas del sprint, las cuales es posible ver en mayor detallen en el reporte gerencial, en el cual se describen las actividades realizadas en cada sprint, la prioridad, el responsable, el tiempo (horas) empleados para este y una estimación del esfuerzo de estas actividades.

Sin embargo los requerimientos para el proyecto se levantaron durante el desarrollo del SRS, por lo que se determinan los requerimientos necesarios para cada iteración en lugar de especificarlos en un momento inicial de esta forma se tiene una visión mas realista al momento de realizar el levantamiento de los requerimientos y da cabida a un mejor versiona miento de los mismos

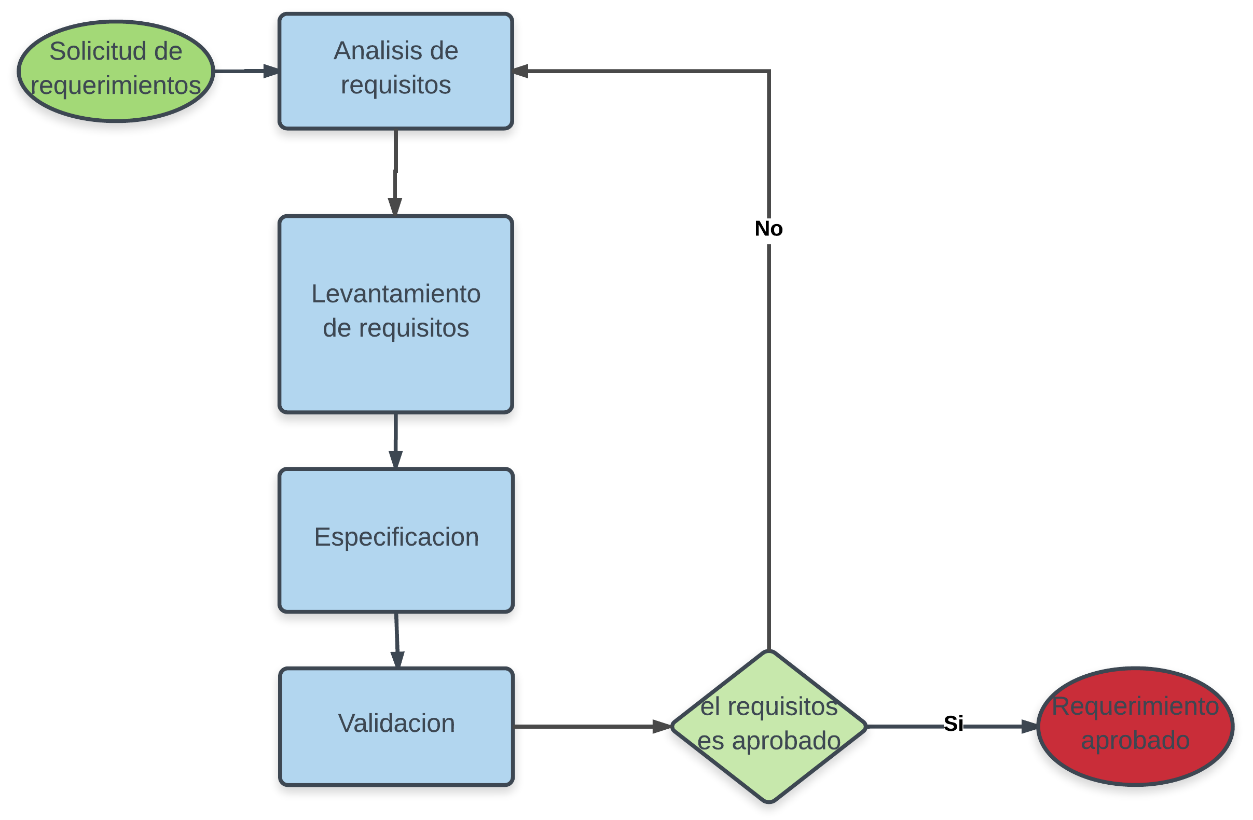


Ilustración 3 Levantamiento de requisitos

En la *Ilustración 3 Levantamiento de requisitos* se expone el ciclo empleado para el levantamiento y especificación de los requisitos. A continuación, se realiza la explicación de cada una de las actividades expuestas en la *Ilustración 3 Levantamiento de requisitos:*

* Análisis de requerimientos: el análisis de requisitos se desarrolla en la sección 8.1 Requisitos de interfaces externas, esta describe los aspectos del sistema, hardware, usuario, comunicación y memoria que la app web “Campus Finder “requiere según el grupo “ACTIVE”
* Levantamiento de requisitos: el levantamiento de requisitos hace referencia a la identificación y documentación de los requisitos del sistema, a partir de los usuarios, clientes o interesados; en la sección *9.2 Recolección* de requisitos

La recolección de requisitos se define en el PMBOK Guide como se muestra en la Ilustración 2Recolección de requisitos PMBOOK, sin embargo, solo es necesario desarrollar los siguientes:

* Plan de gestión de alcance: proporciona claridad en cuanto a cómo el equipo “ACTIVE” determina qué tipo de requisitos necesitan ser recolectados para el proyecto de “Campus Finder”. Por ende, el alcance para esta entrega se encuentra especificado en la sección 6.2 Alcance del producto
* Plan de gestión de requerimientos:
  + Rastreo
  + Priorización
  + Métricas del producto implementadas

#### Rastreo

A la hora de determinar los requerimientos de software se analizan las funcionalidades que el sistema debe poseer, y todos aquellos factores extras con los que debe contar para cumplir con su propósito. Para esto se clasifican los requerimientos según las características que brindan al sistema, el esquema implementado para la clasificación se encuentra a continuación, el esquema implementado es el brindado por FURPS, como se muestra en la Ilustración 4 Tipos de requerimientos

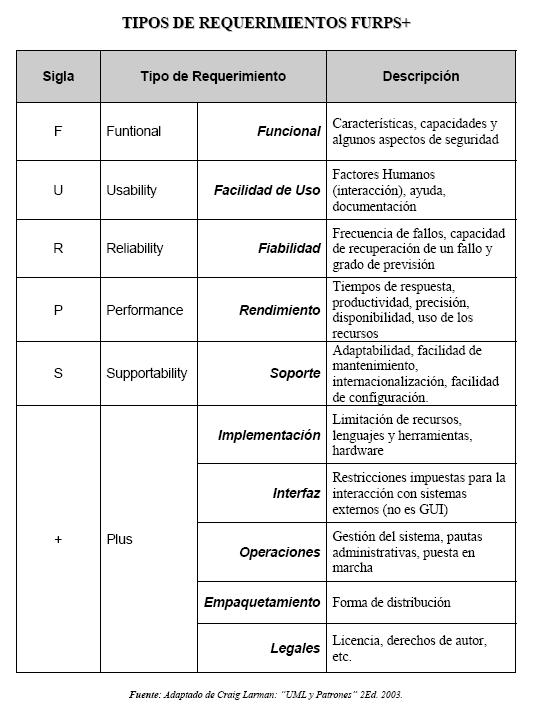


Ilustración 4 Tipos de requerimientos

#### Priorización de requisitos

Una vez se han determinado los requerimientos del sistema, estos deben ser clasificados según su importación para dar un orden de realizarlos según su prioridad con respecto al resto. Para priorizar los requerimientos se establecen tres niveles de prioridad, Alta, Media, Baja, siendo Alta para aquellos requerimientos que aportan funcionalidad al sistema y deben ser realizados antes que los demás; la priorización Media corresponde a aquellos requerimientos que su elaboración brinda un extra al sistema y por último encontramos la prioridad Baja la cual corresponde a los requisitos que no afectan el funcionamiento o el objetivo del proyecto.

Los niveles de prioridad para cada requerimiento se establecen para cada requerimiento según la clasificación que obtengan según el anexo **Requisitos**, de acuerdo a las calificaciones de cada requerimiento se asignó un rango de prioridad como se muestra en:

|  |  |
| --- | --- |
| **Nivel** | **Rango** |
| **Alto** | 4-5 |
| **Medio** | 1.8-3.9 |
| **Bajo** | 0-1.7 |

Ilustración 5 Priorización de requerimientos

#### Métricas

Las métricas que se utilizaron para el proyecto y el software se encuentran especificadas en el documento SPMP en la sección 11.4 Medidas y procesos de medición. La medida utilizada con los requerimientos se encuentra en la sección 10 Proceso de verificación y validación

* Levantamiento donde se especifica el levantamiento y herramientas implementadas para el levantamiento de los requisitos
* Especificación: hace referencia a la descripción de los requerimientos de una forma más detallada para realizar el diseño y realizar pruebas al sistema; En la sección 9.3 Para el proceso de levantamiento de requerimientos y considerando los 5 componentes clasificatorios de los requerimientos, los cuales son: servidor, plataforma, base de datos, usuario y administrador; el grupo ACTIVE decidió separar en 3 técnicas diferentes el levantamiento de estos requerimientos con el fin de abarcar la mayor cantidad de necesidades cumpliendo con el alcance establecido y las metas propuestas.

Las técnicas utilizadas son:

* **Workshops:** Esta técnica consiste en pequeñas reuniones organizadas en donde el objetivo es que teniendo la misma información todos los integrantes del grupo, puedan llegar a necesidades y requisitos puntuales, que no se repitan y que, al momento de llevarlas a cabo en las siguientes fases, todos los miembros sepan de qué se trata el requerimiento y no haya cupo para malas interpretaciones o diferencias con el producto prometido al usuario.
* **BrainStorming**: Esta técnica consiste, como su nombre lo dice en que todos los miembros del grupo expondrán muchas ideas para al finalizar ser evaluadas obteniendo así una gran cantidad de ideas adecuadas para la puesta en marcha del proyecto. Para esta técnica, se requiere de dos partes. En la primera fase se deberán identificar la mayor cantidad de ideas sin comentar o justificar ninguna, luego, se revisan todas las ideas y se agrupan por similares para pasar a la segunda fase, donde es la calificación de todas las ideas siguiendo unos parámetros establecidos. Consiguiendo gran cantidad de ideas que cumplen con los criterios necesarios para ser parte de los requerimientos.
* **UserStories**: Esta técnica, aunque consiste en conversar con los usuarios finales de tal manera que se puedan recopilar sus necesidades enfatizando en las funcionalidades del sistema intentando que sea lo más concreto y claro posible. Sin embargo, como para este proyecto el grupo ACTIVE es su propio usuario, se generó la modificación que cada uno proponía diferentes requerimientos poniéndose en el rol de usuario y al finalizar una persona externa los leía y comentaba de acuerdo con lo que se propuso, de tal manera que pudiese conocer qué era lo que se estaba esperando y adicionalmente el grupo ACTIVE podría contar con otra perspectiva frente a las necesidades y funcionalidades del aplicativo CampusFinder.

Por lo tanto, se decidió que basándose en las técnicas propuestas por PMOinformatica se realizaría el levantamiento de los requerimientos del servidor y de la base de datos por medio de la técnica de “workshops”, para los de usuario “UserStories” y para los requerimientos de plataforma y de administrador, la técnica “Brainstorming” utilizando como criterios de evaluación: la dificultad de implementación, la conveniencia, número de componentes involucrados.

* Especificación, se determina la forma en que se especifican los requerimientos.
* Verificación: Los parámetros y procesos para la verificación y validación de los requerimientos se encuentran especificados en la sección 10 Proceso de verificación y validación.

### Recolección de requisitos

La recolección de requisitos se define en el PMBOK Guide como se muestra en la Ilustración 2Recolección de requisitos PMBOOK, sin embargo, solo es necesario desarrollar los siguientes:

* Plan de gestión de alcance: proporciona claridad en cuanto a como el equipo “ACTIVE” determina qué tipo de requisitos necesitan ser recolectados para el proyecto de “Campus Finder”. Por ende, el alcance para esta entrega se encuentra especificado en la sección 6.2 Alcance del producto
* Plan de gestión de requerimientos:
  + Rastreo
  + Priorización
  + Métricas del producto implementadas

#### Rastreo

A la hora de determinar los requerimientos de software se analizan las funcionalidades que el sistema debe poseer, y todos aquellos factores extras con los que debe contar para cumplir con su propósito. Para esto se clasifican los requerimientos según las características que brindan al sistema, el esquema implementado para la clasificación se encuentra a continuación, el esquema implementado es el brindado por FURPS, como se muestra en la Ilustración 4 Tipos de requerimientos

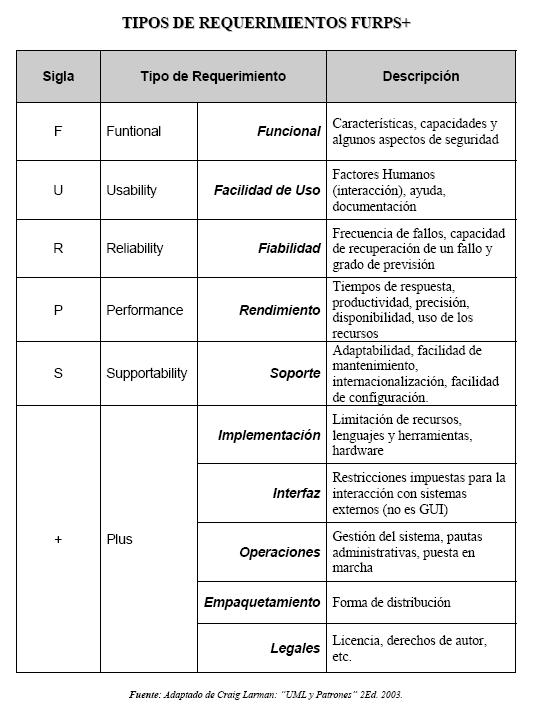


Ilustración 4 Tipos de requerimientos

#### Priorización de requisitos

Una vez se han determinado los requerimientos del sistema, estos deben ser clasificados según su importación para dar un orden de realizarlos según su prioridad con respecto al resto. Para priorizar los requerimientos se establecen tres niveles de prioridad, Alta, Media, Baja, siendo Alta para aquellos requerimientos que aportan funcionalidad al sistema y deben ser realizados antes que los demás; la priorización Media corresponde a aquellos requerimientos que su elaboración brinda un extra al sistema y por último encontramos la prioridad Baja la cual corresponde a los requisitos que no afectan el funcionamiento o el objetivo del proyecto.

Los niveles de prioridad para cada requerimiento se establecen para cada requerimiento según la clasificación que obtengan según el anexo **Requisitos**, de acuerdo a las calificaciones de cada requerimiento se asignó un rango de prioridad como se muestra en:

|  |  |
| --- | --- |
| **Nivel** | **Rango** |
| **Alto** | 4-5 |
| **Medio** | 1.8-3.9 |
| **Bajo** | 0-1.7 |

Ilustración 5 Priorización de requerimientos

#### Métricas

Las métricas que se utilizaron para el proyecto y el software se encuentran especificadas en el documento SPMP en la sección 11.4 Medidas y procesos de medición. La medida utilizada con los requerimientos se encuentra en la sección 10 Proceso de verificación y validación

## Levantamiento

Para el proceso de levantamiento de requerimientos y considerando los 5 componentes clasificatorios de los requerimientos, los cuales son: servidor, plataforma, base de datos, usuario y administrador; el grupo ACTIVE decidió separar en 3 técnicas diferentes el levantamiento de estos requerimientos con el fin de abarcar la mayor cantidad de necesidades cumpliendo con el alcance establecido y las metas propuestas.

Las técnicas utilizadas son:

* **Workshops:** Esta técnica consiste en pequeñas reuniones organizadas en donde el objetivo es que teniendo la misma información todos los integrantes del grupo, puedan llegar a necesidades y requisitos puntuales, que no se repitan y que, al momento de llevarlas a cabo en las siguientes fases, todos los miembros sepan de qué se trata el requerimiento y no haya cupo para malas interpretaciones o diferencias con el producto prometido al usuario.
* **BrainStorming**: Esta técnica consiste, como su nombre lo dice en que todos los miembros del grupo expondrán muchas ideas para al finalizar ser evaluadas obteniendo así una gran cantidad de ideas adecuadas para la puesta en marcha del proyecto. Para esta técnica, se requiere de dos partes. En la primera fase se deberán identificar la mayor cantidad de ideas sin comentar o justificar ninguna, luego, se revisan todas las ideas y se agrupan por similares para pasar a la segunda fase, donde es la calificación de todas las ideas siguiendo unos parámetros establecidos. Consiguiendo gran cantidad de ideas que cumplen con los criterios necesarios para ser parte de los requerimientos.
* **UserStories**: Esta técnica, aunque consiste en conversar con los usuarios finales de tal manera que se puedan recopilar sus necesidades enfatizando en las funcionalidades del sistema intentando que sea lo más concreto y claro posible. Sin embargo, como para este proyecto el grupo ACTIVE es su propio usuario, se generó la modificación que cada uno proponía diferentes requerimientos poniéndose en el rol de usuario y al finalizar una persona externa los leía y comentaba de acuerdo con lo que se propuso, de tal manera que pudiese conocer qué era lo que se estaba esperando y adicionalmente el grupo ACTIVE podría contar con otra perspectiva frente a las necesidades y funcionalidades del aplicativo CampusFinder.

Por lo tanto, se decidió que basándose en las técnicas propuestas por PMOinformatica se realizaría el levantamiento de los requerimientos del servidor y de la base de datos por medio de la técnica de “workshops”, para los de usuario “UserStories” y para los requerimientos de plataforma y de administrador, la técnica “Brainstorming” utilizando como criterios de evaluación: la dificultad de implementación, la conveniencia, número de componentes involucrados.

## Especificación

La especificación de requerimientos se llevó a cabo, durante la fase de diseño de “Campus Finder” , por el grupo ACTIVE donde todos los integrantes realizaron aportes para la captura de requerimientos, sin embargo, se asignaron a 2 integrantes del equipo (Natalia Sofia Otero y Carlos Eduardo Camacho) para formalizarlos en las tablas expuestas anteriormente( sección *Funciones del producto* y sección *Requisitos específicos)*

Los integrantes escogidos para la formalización de los requerimientos capturados poseen la responsabilidad de realizar las respectivas correcciones a los requerimientos si se llega a considerar necesario, esta decisión es dada por todo el Scrum team.

1. **Características de los requerimientos:** las características que debe cumplir los requerimientos se encuentran especificadas en el anexo **Características de los requisitos**. Cada una de estas características fue validad para cada requisito como se especifica en la sección Proceso de verificación y validación
2. **Control y seguimiento:** El control y seguimiento de los requerimientos se especifica en la sección Planeación como se especifica en los ítems Planificación y Gestión de requerimientos
3. **Artefactos de desarrollo:** Los artefactos que se implementaron con el fin de especificar los requerimientos, fue la implementación de la plantilla de documentación de requerimientos

En la Tabla 15 Plantilla documentación requerimientos se muestra el formato implementado para la documentación de cada requerimiento obtenido en la captura de requerimientos

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id requerimiento** |  | **Involucrado** |  |  |
| **Descripción** |  | | | |
| **Razón** |  | | | |
| **Autor** |  | | | |
| **Criterio de medición** |  | | | |
| **Versión** |  | | | |

Tabla 15 Plantilla documentación requerimientos

En los anexos se realiza la justificación de la priorización de cada uno de los requerimientos, donde cada requerimiento recibe una calificación acorde a su importancia, costo en tiempo, riesgo y factibilidad, para consiguiente realizar el promedio entre estos parámetros, donde el resultado de su promedio corresponde a la prioridad que se le dará.

# Proceso de verificación y validación

Los criterios que se tomaran en cuenta para la verificación y validación de los requisitos se encuentran definidos en la *Anexo* **Requisitos**

## Proceso de validación

Para que un requisito sea aprobado, el equipo de diseño debe verificar que cumpla con las características mencionadas anteriormente. Aquel requisito que no cumpla con alguna característica debe ser reevaluado para decidir qué sucederá. En caso tal que se puede reestructuras para que sea acorde con lo estipulado, se harán los cambios necesarios. Si se decide que el requisito no va acorde con lo que se propone en el proyecto, este procederá a ser descartado.

Como se mencionó anteriormente, el equipo de diseño debe pasar por cada requisito y evaluar su valor e importancia y eliminar aquellos que no pasen la revisión.

## Proceso de verificación

El proceso de verificación se lleva a cabo con cada sprint. A medida que se dan iteraciones del diseño y se agregan funcionalidades, se debe verificar que todos los requisitos aproados se estén cumpliendo. Esto es que el diseño refleje estos requisitos y el sistema los cumpla.

# Prototipo

## Diseño

El diseño del software se ve reflejado en este documento y en los diferentes casos de uso que se encuentran definidos en el *anexo* **Casos de uso**, los conceptos base se ven reflejados en el Modelo de dominio *cómo es posible observarlos en la* Ilustración 1 Diagrama modelo de dominio El diseño de la interfaz gráfica con el usuario se hizo por medio de mockups en Balsamiq Muckups que se encuentran en el anexo **Mockups**

## Justificación según priorización

|  |  |  |
| --- | --- | --- |
| **ID** | **Nombre** | **Estado** |
| **CU08** | Ingresar al sistema | Funcionalidad Parcial |
| **CU12** | Generar Ruta | Funcionalidad Completa |
| **CU06** | Cerrar Sesión | Funcionalidad Completa |

# Anexos

1. **Casos de uso**

Archivo comprimido ZIP, en el que se encuentra el documento original con el que se trabajaron los casos de uso usando Enterprise Architect, los diagramas finales de los casos de uso en formato JPG, y la documentación de los casos de uso en PDF y Word

1. **Requisitos**

Archivo comprimido ZIP, en el que se encuentra el análisis de requisitos en un documento Excel, procesos de validación y verificación de requisitos en formato PNG.

1. **Mockups**

Archivo comprimido ZIP, en el que se encuentra los mockups del proyecto, realizados en la herramienta Balsamiq Muckups; en este se encuentra las imágenes de cada Muchkups que se tuvieron en cuenta para el desarrollo del prototipo

1. **Control de trabajo**

Documento en Excel en el cual se lleva un seguimiento del trabajo asignado para cada uno de los miembros del Scrum Team

1. **Características de los requisitos**

Documento PDF donde se encuentra la respectiva explicación de las características de los requisitos, además se encuentra la tabla de documentación de los mismos

# Referencias

(s.f.). Obtenido de W3.CSS Home: https://www.w3schools.com/w3css/

*Administracion de Requerimientos*. (s.f.). Recuperado el 25 de 8 de 2017, de http://clases3gingsof.wikifoundry.com/page/Administraci%C3%B3n+de+Requerimientos

Agile Alliance. (s.f.). *What is Extreme Programming (XP)? | Agile Alliance*. Obtenido de Agile Alliance: https://www.agilealliance.org/glossary/xp/#q=~(filters~(postType~(~'post~'aa\_book~'aa\_event\_session~'aa\_experience\_report~'aa\_glossary~'aa\_research\_paper~'aa\_video)~tags~(~'xp))~searchTerm~'~sort~false~sortDirection~'asc~page~1)

*Amazon Web Services*. (s.f.). Obtenido de https://aws.amazon.com/es/nosql/

Bizmanualz. (s.f.). *Document Manager Job Description*. Recuperado el 17 de 8 de 2017, de https://www.bizmanualz.com/job-descriptions/aerospace-job-descriptions/document-manager-job-description

Bustamante Barajas Juan Manuel, C. O. (28 de Junio de 2015). *Gestión de Proyectos de software SCG-1009 SC9A.* Recuperado el 21 de Agosto de 2017, de https://docs.google.com/viewer?a=v&pid=sites&srcid=ZGVmYXVsdGRvbWFpbnwyMDE1MDZncHN1bGxvYXJvZG9sZm98Z3g6M2ViMGY2ZGJmNzY0MjljYQ

*Ciclo de vida en "V" - INGENIERIA DE SOFTWARE*. (s.f.). (Weebly) Recuperado el 20 de 8 de 2017, de http://ingsoftware.weebly.com/ciclo-de-vida-en-v.html

DigitalOcean. (s.f.). *DigitalOcean: Cloud computing designed for developers*. Obtenido de https://www.digitalocean.com/

International Scrum Institute. (s.f.). *Scrum Roles – The Scrum Team - International Scrum Institute*. (International Scrum Institute) Recuperado el 16 de 8 de 2017, de http://www.scrum-institute.org/Scrum\_Roles\_The\_Scrum\_Team.php

*Introduction to HTML*. (s.f.). Obtenido de https://developer.mozilla.org/en-US/docs/Learn/HTML/Introduction\_to\_HTML

ISQTB. (s.f.). *What is Spiral model- advantages, disadvantages and when to use it?* Recuperado el 19 de 8 de 2017, de http://istqbexamcertification.com/what-is-spiral-model-advantages-disadvantages-and-when-to-use-it/

iZenBridge. (s.f.). *What Is Configuration Management?* Recuperado el 19 de 8 de 2017, de https://www.izenbridge.com/blog/what-is-configuration-management-a-software-management-study/

Project Management Institute Staff. (2013). *A Guide to the Projecto Management Body of Knowledge(PMBOK Guide)-Fifth Edition.* Newtown square,Pa: Project Management Institute Inc.

Scrum Alliance. (s.f.). *Scrum Roles Demystified - Scrum Alliance*. (Scrum Alliance) Recuperado el 16 de 8 de 2017, de https://www.scrumalliance.org/agile-resources/scrum-roles-demystified

Scrum.Org. (s.f.). *The Scrum Guide™*. Recuperado el 21 de 8 de 2017, de https://www.scrumguides.org/scrum-guide.html

SCRUMstudy. (2016). *A Guide to the Scrum Body of Knowledge (SBOK™ Guide) – 2016 edition .* VMEdu Inc.

Software Engineering Institute. (s.f.). *Duties, Skills, & Knowledge of a Software Architect*. Recuperado el 19 de 8 de 2017, de https://www.sei.cmu.edu/architecture/research/previousresearch/duties.cfm

*Software Life Cycle Models*. (s.f.). (Scribd) Recuperado el 20 de 8 de 2017, de https://www.scribd.com/document/73647276/Software-Life-Cycle-Models

SOMMERVILLE, I. (2011). *Software Engineering (Ninth Edition).* Adison Wesley.

The Interantional Scrum Institute. (s.f.). *Scrum Revealed.*

Tusalario. (08 de 2017). *Tusalario.org/Colombia*. Obtenido de http://www.tusalario.org/colombia/Portada/tusalario/compara-tu-salario#/

Unified Process for Education. (s.f.). *Role: Configuration Manager*. Recuperado el 19 de 8 de 2017, de http://www.upedu.org/process/workers/wk\_cmmgr.htm

Weese, S. (9 de 4 de 2009). *Identify Projecto Assuptions and Constraints | Learning Tree*. Recuperado el 9 de 4 de 2009, de https://blog.learningtree.com/identifying-projecto-assumptions-and-constraints/