

Integrantes:

Juan José Afanador Ochoa

Stiven González Olaya

Santiago Roa Dueñas

Gabriel Andrés Niño Carvajal

John Jairo González Martínez

*Buen Vecino*

Pontificia Universidad Javeriana

Ingeniería de Sistemas

Ingeniería de Software



**Docente:** Carlos Andrés Parra Acevedo

Versión 1.0

# Historial de cambios

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Encargado |
| 29/09/2020 | 0.1 | Portada, resumen | Juan Afanador |
| 30/09/2020 | 0.2 | Modelo de dominio | Todos los miembros del grupo |
| 30/09/2020 | 0.3 | Perspectiva del producto | Stiven Gonzalez, John Gonzalez |
| 02/10/2020 | 0.4 | Interfaces con el sistema | Stiven Gonzalez, Santiago Roa |
| 02/10/2020 | 0.5 | Modos de operación del usuario | Gabriel Niño, John Gonzalez |
| 04/10/2020 | 0.6 | Períodos de actividad e inactividad | Gabriel Niño |
| 04/10/2020 | 0.7 | Funciones de soporte a procesamiento de datos | Gabriel Niño |
| 04/10/2020 | 0.8 | Funciones del producto | Todos los miembros del grupo |
| 04/10/2020 | 0.9 | Características del usuario | Stiven Gonzalez, Juan Afanador |
| 05/10/2020 | 0.10 | Requisitos no funcionales | Gabriel Niño, Santiago Roa |
| 05/10/2020 | 0.11 | Suposiciones y dependencias | John Gonzalez, Stiven Gonzalez, Juan Afanador |
| 05/10/2020 | 0.12 | Características del producto software | Todos los miembros del grupo |
| 07/10/2020 | 0.13 | Atributos del sistema software | Juan Afanador, Santiago Roa |
| 07/10/2020 | 0.14 | Requisitos base de datos | Gabriel Niño, Stiven Gonzalez |
| 07/10/2020 | 0.15 | Requisitos de desempeño | Juan Afanador, Santiago Roa, Gabriel Niño |
| 08/10/2020 | 0.16 | Planeación, encargados, frecuencia | Todos los miembros del grupo |
| 09/10/2020 | 0.17 | Desarrollo del proceso | Gabriel Niño |
| 09/10/2020 | 0.18 | Levantamiento | Santiago Roa |
| 09/10/2020 | 0.19 | Proceso de validación | Stiven Gonzalez |
| 09/10/2020 | 0.20 | Validación de la estructura del requisito | Juan Afanador, Stiven Gonzalez |
| 11/10/2020 | 0.21 | Validación de la estructura de la documentación | Juan Afanador, Gabriel Niño |
| 11/10/2020 | 0.22 | Modos de operación del usuario | Gabriel Niño, John Gonzalez |
| 14/10/2020 | 0.23 | Proceso de verificación | Santiago Roa |
| 15/10/2020 | 0.24 | Especificación | John Gonzalez, Gabriel Niño, Santiago Roa |
| 16/10/2020 | 0.25 | Anexos | Juan Afanador, Stiven Gonzalez |
| 16/10/2020 | 0.26 | Lista de figuras | Gabriel Niño, Juan Afanador |
| 16/10/2020 | 0.27 | Lista de tablas | John Gonzalez |
| 16/10/2020 | 0.28 | Tabla de contenidos, | John Gonzalez, Stiven Gonzalez |
| 16/10/2020 | 0.29 | Referencias | Juan Afanador, Stiven Gonzalez, John Gonzalez |
| 16/10/2020 | 1.0 | Historial de cambios | Santiago Roa, Gabriel Niño |

# Resumen

Mediante el siguiente documento se dará a conocer la especificación de requisitos y todo el proceso que se necesita para la correcta documentación de la aplicación Buen vecino.

Como introducción se presenta inicialmente una breve introducción, donde se verá el alcance del proyecto y la descripción del producto. Luego se procederá a explicar el modelo de dominio de la aplicación y posteriormente la documentación de los requisitos de la aplicación que, así como los factores generales que afectan el producto, realizando un énfasis en la especificación y los requisitos como bases de datos y atributos de software.

# Tabla de contenidos

Contenido

[1 Historial de cambios 2](#_Toc53764314)

[2 Resumen 4](#_Toc53764315)

[3 Tabla de contenidos 4](#_Toc53764316)

[4 Lista de figuras 6](#_Toc53764317)

[5 Lista de tablas 6](#_Toc53764318)

[6 Introducción 7](#_Toc53764319)

[7 Descripción global 8](#_Toc53764320)

[7.1 Modelo de dominio 8](#_Toc53764321)

[7.2 Perspectiva de producto 8](#_Toc53764322)

[7.2.1 Interfaces con el sistema 9](#_Toc53764323)

[7.2.2 Restricciones de memoria 9](#_Toc53764324)

[7.2.3 Operaciones 10](#_Toc53764325)

[7.3 Funciones del producto 12](#_Toc53764326)

[7.4 Características del usuario 12](#_Toc53764327)

[7.5 Requisitos No Funcionales 13](#_Toc53764328)

[7.5.1 Restricciones de Software 15](#_Toc53764329)

[7.5.2 Restricciones de Hardware 15](#_Toc53764330)

[7.6 Suposiciones y dependencias 15](#_Toc53764331)

[8 Requisitos específicos 16](#_Toc53764332)

[8.1 Características del producto software 16](#_Toc53764333)

[8.2 Requisitos de desempeño 17](#_Toc53764334)

[8.3 Atributos del sistema software 18](#_Toc53764335)

[8.4 Requisitos de la base de datos 21](#_Toc53764336)

[8.4.1 Tipos de datos almacenados 21](#_Toc53764337)

[8.4.2 Tipos de Consultas utilizadas 21](#_Toc53764338)

[8.4.3 Indexación de los datos 21](#_Toc53764339)

[8.4.4 Frecuencia de acceso 21](#_Toc53764340)

[9 Proceso de ingeniería de requisitos 23](#_Toc53764341)

[9.1 Planeación 23](#_Toc53764342)

[9.1.1 Encargados 23](#_Toc53764343)

[9.1.2 Frecuencia 23](#_Toc53764344)

[9.1.3 Desarrollo del proceso 23](#_Toc53764345)

[9.2 Levantamiento 24](#_Toc53764346)

[9.3 Especificación 25](#_Toc53764347)

[10 Proceso de verificación y validación 25](#_Toc53764348)

[10.1 Proceso de validación 25](#_Toc53764349)

[10.2 Validación de la estructura del requisito 25](#_Toc53764350)

[10.3 Validación de la estructura de la documentación 26](#_Toc53764351)

[10.4 Proceso de verificación 26](#_Toc53764352)

[11 Anexos 27](#_Toc53764353)

[12 Referencias 28](#_Toc53764354)

# Lista de figuras

[Ilustración 1. Desarrollo del proceso. 22](#_Toc53765760)

[Ilustración 2. Levantamiento. 22](#_Toc53765761)

# Lista de tablas

[Tabla 1. Plantilla para documentar el modelo de dominio. 8](#_Toc53765691)

[Tabla 2. Interfaces con el sitema 9](#_Toc53765692)

[Tabla 3. Restricciones de memoria navegador Google Chrome en Windows. 9](#_Toc53765693)

[Tabla 4. Restricciones de memoria navegador Google Chrome en MacOs. 9](#_Toc53765694)

[Tabla 5. Modos de operación del usuario. 11](#_Toc53765695)

[Tabla 6. Períodos de actividad e inactividad. 11](#_Toc53765696)

[Tabla 7. Características del usuario arrendador. 12](#_Toc53765697)

[Tabla 8. Características del usuario arrendatario. 13](#_Toc53765698)

[Tabla 9. Requisitos no funcionales. 15](#_Toc53765699)

[Tabla 10 Atributos del sistema software - Confiabilidad. 18](#_Toc53765700)

[Tabla 11 Atributos del sistema software - Disponibilidad. 18](#_Toc53765701)

[Tabla 12 Atributos del sistema software - Seguridad. 19](#_Toc53765702)

[Tabla 13 Atributos del sistema software - Mantenibilidad. 19](#_Toc53765703)

[Tabla 14 Atributos del sistema software - Portabilidad. 19](#_Toc53765704)

[Tabla 15 Tipo de datos almacenados 20](#_Toc53765705)

# Introducción

El escrito tiene como objetivo principal la especificación de requisitos de *software* para el desarrollo de la aplicación web Buen Vecino, por tanto, a lo largo del documento se podrán identificar puntos clave que permiten definir tantas funcionalidades como restricciones que serán sumamente importantes para desarrollar el *software* de una manera modular y funcional.

Como primera medida se abordará la descripción global, donde se extraerán los diferentes datos que permiten la correcta identificación de requisitos funcionales y no funcionales, adicional a ello los usuarios quienes accederán a la plataforma con sus credenciales y las funcionalidades y características principales del producto también estarán definidas. Adicionalmente, se podrá encontrar los diversos supuestos y restricciones para generar el *software* en el tiempo determinado y con una correcta modularidad.

El documento tiene como énfasis el explicar detalladamente la especificación de los requisitos de *software*, esto para poder definir las múltiples características del producto, los atributos del sistema y todo el proceso que conlleva tanto el levantamiento de requisitos como la planeación y posterior administración.

Como producto tenemos por tanto un aplicativo web llamado Buen Vecino, el cual funciona como un intermediario virtual entre arrendadores y arrendatarios de viviendas universitarias, por tanto, se podrán cargar a la página diversos modos de hospedaje junto a precios y datos de contacto para así generar compras dentro del aplicativo, del cual se pretende obtener un *Fee* o cuota de transacción para el mantenimiento de este. El objetivo de Buen Vecino es fortalecer la comunicación entre personas interesadas en el arriendo de inmuebles de una manera sencilla y rápida con múltiples opciones, intuitiva y de carácter estudiantil.

Para cumplir con ello el aplicativo utilizará servicios *API* junto a bases de datos para poder administrar tanto los registros de usuarios como el registro de nuevas propiedades inmobiliarias, donde luego de ello cualquier persona podrá consultar los datos y referenciarse de un mapa para poder tomar una decisión frente a costos y oportunidades de vivienda, ya que el aplicativo notificará las comodidades de cada vivienda como aire acondicionado o *Wifi*, así como puntos de interés cercanos.

El aplicativo adicional a ello, permite la edición de viviendas o solicitudes de vivienda, esto por medio del modo de programación implementado, el cual se trabaja sobre componentes, por lo cual los cambios pueden ser mantenidos de una manera rápida y efectiva, para al final poder generar una solicitud de cobro o de contacto con el arrendatario, en el cual se podrá ver una simulación de plataforma de pago interactiva.

# Descripción global

## Modelo de dominio

En esta sección se pretende presentar un modelo de domino el cual describe el sistema, en este mismo se muestra una abstracción que permite la implementación de la aplicación.

[**Ver anexo (Modelo de dominio)**](https://drive.google.com/file/d/1t37Ohlrl7Az6y9o7rCYy-VASD5ZZbpP0/view?usp=sharing)

Con base en este modelo se procede a documentar cada una de las entidades presentes en el mismo, donde se establecerán los propósitos de cada una y de igual manera los atributos tomando como referencia el siguiente formato:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***ID*** |  | | ***Elemento del dominio*** |  |
| ***Descripción*** |  | | | |
| ***Atributos*** | | | | |
| ***Nombre*** | | ***Descripción*** | | |
|  | |  | | |
|  | |  | | |

Tabla 1. Plantilla para documentar el modelo de dominio.

Dicha documentación se encuentra anexada por la longitud de esta. [Ver anexo (Entidades de dominio](https://drive.google.com/file/d/1Zu6FZiAtfnSbKEeEjig_i1V1ulwXvLie/view?usp=sharing)).

## Perspectiva de producto

Desde la llegada del internet la brecha de estudio se ha visto afectada de una manera sumamente positiva, esto es ampliamente visto cuando un estudiante de cualquier parte del país desea trasladarse a Bogotá para educarse en la universidad, Buen Vecino es una nueva alternativa la cual cuenta como gran diferenciador el ser un portal exclusivo para universitarios, donde se moderan las publicaciones manualmente y son los universitarios quienes tienen la posibilidad de previsualizar varios sitios o lugares de su preferencia, bien sea mediante un filtro en el cual puedan ver en un mapa las diversas opciones de alojamiento con sus precios así como una búsqueda por imágenes para conocer un poco más el lugar a hospedarse.

Todo esto se logra por medio de 2 módulos, la interfaz de arrendatario y la interfaz de arrendador, donde dependiendo del rol se pueden cumplir tareas específicas como publicar un inmueble o realizar una reservación.

Buen Vecino no es un sistema tradicional de hospedaje en línea ya que no cuenta con fechas específicas de arrendamiento sino con periodos preconfigurados en modalidades mensuales, donde el valor puede variar según el arrendador.

### Interfaces con el sistema

|  |  |
| --- | --- |
| **Interfaz** | **Descripción** |
| Usuario | El usuario podrá interactuar con el sistema cuando ingrese a la página web de Buen Vecino. Una vez ingrese al dominio, encontrará una interfaz web, mediante la cual el usuario podrá interactuar con las funcionalidades del sistema. Lo anterior se evidencia en el siguiente anexo: [Ver anexo (Mock-ups)](https://drive.google.com/file/d/1mJ3jzDrwvYnEA5Fwz-IzswboScXUaJmk/view?usp=sharing), dicha interfaz será modificada dependiendo del comportamiento del usuario al interactuar con la misma. |
| Software | Se utilizará las *APIs* que ofrece *Firebase* para las llamadas a la base de datos y el sistema de autenticación de usuarios para el ingreso a la plataforma. |
| Hardware | Es necesario tener un dispositivo electrónico y conexión a internet para ingresar a Buen Vecino. |

Tabla 2. Interfaces con el sistema

### Restricciones de memoria

Google Chrome

Windows:

|  |  |
| --- | --- |
| Característica | Descripción |
| Procesador | Intel Pentium 4 o posterior compatible con SSE2 |
| Espacio en disco | 100 MB + |
| Memoria RAM | 128MB |
| Sistema Operativo | Windows 7, Windows 8, Windows 8.1, Windows 10 o versiones superiores |

Tabla 3. Restricciones de memoria navegador Google Chrome en Windows.

MacOS:

|  |  |
| --- | --- |
| Característica | Descripción |
| Procesador | Intel – Based Macs (2006) + |
| Espacio en disco | 100 MB + |
| Memoria RAM | 128MB |
| Sistema Operativo | OS X versión 10.5.6 + |

Tabla 4. Restricciones de memoria navegador Google Chrome en MacOS.

### Operaciones

#### **Modos de operación del usuario**

En la tabla 5 se describen los tipos de usuario que tendrá el sistema y un listado con las funciones que cada usuario puede realizar.

|  |  |
| --- | --- |
| Tipo de usuario | Funciones |
| Arrendador | 1. Registrarse en la página web con información como: nombres y apellidos, cédula de ciudadanía o extranjería, correo electrónico, dirección y ciudad donde reside. 2. Iniciar y cerrar sesión en la página web. 3. Cargar, modificar o eliminar inmuebles a la página web con su respectiva información e imágenes. 4. Chatear en tiempo real con otro usuario. 5. Realizar una calificación sobre el comportamiento de un arrendatario durante el período de estadía. |
| Arrendatario | 1. Registrarse en la página web con información necesaria como: nombres y apellidos, cédula de ciudadanía o extranjería, correo electrónico y teléfono. 2. Iniciar y cerrar sesión en la página web. 3. Gestionar lista de favoritos (crear, modificar y eliminar). 4. Chatear en tiempo real con otro usuario. 5. Buscar un inmueble de acuerdo con sus necesidades. 6. Visualizar información adicional de un inmueble. 7. Geolocalizar un inmueble ingresando una dirección especifica. 8. Administrar métodos de pago. 9. Consultar su historial de pagos. 10. Realizar una calificación sobre el inmueble hospedado. |

Tabla 5. Modos de operación del usuario.

#### **Períodos de actividad e inactividad**

|  |  |  |
| --- | --- | --- |
| Estado | Proceso | |
| * Activo | * El aplicativo está disponible en todo momento siempre que se pueda acceder a él mediante algún navegador web que lo soporte. |
| * Inactivo | * El aplicativo estará inactivo mientras se realicen actividades de mantenimiento programado. * El aplicativo estará inactivo si existe cualquier tipo de inconsistencia con la base de datos. |

Tabla 6. Períodos de actividad e inactividad.

#### **Procesos de recuperación**

El protocolo que se va a seguir para realizar el respaldo de la información consiste en una copia de seguridad diario a las 00:00 am (GMT-5). Realizado automáticamente por el sistema de *Firebase*, el cual provee copias de seguridad automáticas de *Firebase* *Realtime* *Database*, una función de autoservicio que permite crear copias de seguridad diarias de los datos de la aplicación de *Database* y de las [reglas](https://firebase.google.com/docs/database/security/?hl=es) en formato JSON y almacenarlas en un depósito de Google Cloud Storage.[1]

#### **Funciones de soporte a procesamiento de datos**

El aplicativo debe validar el tipo de usuario que está ingresando para mostrar la interfaz correspondiente, bien sea la interfaz de arrendador o arrendatario, para ello realizará el procesamiento de los datos conectado a *Firebase*, esto ya que la base de datos a utilizar será externa al aplicativo web para mantener la integridad de los datos y su fácil acceso.

## Funciones del producto

Las funciones del producto que se ofrece en la página web de Buen Vecino se dividen en tres partes, las cuales algunas ya se han presentado en la tabla 5 de la sección *7.2.3.1 Modos de operación del usuario.* Sin embargo, en el siguiente anexo se puede encontrar la descripción de cada una de estas funcionalidades [ver anexo (Documentación casos de uso).](https://drive.google.com/file/d/1mJ3jzDrwvYnEA5Fwz-IzswboScXUaJmk/view?usp=sharing) Para visualizar una perspectiva más abstracta de las funciones del producto, consulte el diagrama de casos de uso [ver anexo (Diagrama de casos de uso).](https://drive.google.com/file/d/1DFZSlJtFrzK3f7VQFxSIez62frNv9L_1/view?usp=sharing)

## Características del usuario

**Arrendador**

|  |  |
| --- | --- |
| Características del usuario | Únicamente personas o individuos interesados en arrendar sus viviendas o propiedades deben contar con un inmueble apto para la venta y con los correspondientes papeles del inmueble en funcionamiento.  Deben proveer sus datos personales al registrarse en la aplicación y pueden ser verificados manualmente por un administrador del sitio de ser necesario. |
| Nivel de seguridad o de privilegios | Un arrendador tiene la posibilidad de crear, modificar, ver y eliminar inmuebles que pertenezcan a su propiedad, así como gestionar las peticiones de chat y notificaciones de compra por parte de un arrendatario.  El arrendador puede ver en su perfil de igual manera toda su información personal y gestionarla en cuanto a cambios se refiere. |
| Roles | Es un generador de contenido, administración de lugares propios y acceso general a la plataforma. |
| Nivel de estudios o experiencia técnica | Conocimientos básicos en manejo de plataforma web, conocimiento en subida de imágenes y configuración de métodos de pago online. |
| Frecuencia de uso | Semanalmente puede verificar el estado de sus inmuebles y dar respuesta a peticiones o inquietudes, sin embargo, dependiendo de la urgencia del arrendatario se recomienda un uso diario. |

Tabla 7. Características del usuario arrendador.

**Arrendatario**

|  |  |
| --- | --- |
| Características del usuario | Cualquier usuario universitario o público general interesado en acceder en la plataforma web Buen Vecino con el fin de buscar propiedades para hospedarse en periodos estudiantiles. |
| Nivel de seguridad o de privilegios | El arrendatario tendrá la posibilidad de acceder a la plataforma mediante un registro de usuario, esto habilitará las diversas funcionalidades como la búsqueda de inmuebles, el chat con los arrendadores, el filtrado de sitios y la visualización vía geográfica de cada lugar de interés.  Adicionalmente a ello tendrá acceso a las opciones básicas de administración de su propio perfil como edición de correo, nombre y contraseña. |
| Roles | El arrendatario únicamente tendrá la posibilidad de acceder a las funcionalidades básicas del sistema, es decir, visualización y solicitud de arrendamiento. |
| Nivel de estudios o experiencia técnica | Conocimientos básicos de manipulación de sitios web. |
| Frecuencia de uso | De acuerdo con la necesidad del usuario en específico, cada vez que requiera administrar o buscar algún lugar de hospedaje de acuerdo con su calendario universitario. Es recomendable que la plataforma sea visitada 1 vez quincenalmente mínimo. |

Tabla 8. Características del usuario arrendatario.

## Requisitos No Funcionales

Teniendo en cuenta la definición de los requisitos no funcionales, ECO pensó inicialmente en el usuario, para poder brindarle una mejor experiencia al momento de usar BuenVecino y que esto sea un diferenciador ante las páginas que ofrecen servicios similares de arrendamiento. Algunos de estos requisitos son tomados directamente de Firebase, ya que utilizando el plan gratuito genera algunas restricciones, esto lo hacemos para que las personas ajenas al proyecto pueden entender las limitaciones de este si desean implementarlo en futuros proyectos, además que puedan comprender como afectan estos requisitos al proyecto. La tabla de los requisitos se presenta a continuación.

|  |  |  |
| --- | --- | --- |
| ID-Requisito | Requisito | Clasificación FURPS+ |
| RNF-01 | El sistema debe estar disponible el 97% del tiempo. | Confiabilidad |
| RNF-02 | El sistema debe atender máximo 500 usuarios concurrentes. | Rendimiento |
| RNF-03 | El sistema soporta 1GB de almacenamiento de datos de usuarios. | + Operaciones |
| RNF-04 | El sistema soporta 5GB de almacenamiento de multimedia. | + Operaciones |
| RNF-05 | El sistema permite 50.000 lecturas de la base de datos por día. | + Operaciones |
| RNF-06 | El sistema debe funcionar sobre *desktop* y *laptops* | Usabilidad |
| RNF-07 | El sistema debe tardar como máximo 3 segundos ante la petición de un usuario. | Rendimiento |
| RNF-08 | El sistema realiza una copia de seguridadde los datos del sistema diariamente. | Confiabilidad |
| RNF-09 | El sistema se debe acoger a los lineamientos de *Firebase* | + Implementación |
| RNF-10 | El sistema debe comunicarse con una interfaz externa para realizar los pagos | +Interfaz |
| RNF-11 | El sistema esta soportado en el idioma español. | Soporte |
| RNF-12 | El sistema le permite a un arrendador tener un máximo de 10 inmuebles | Usabilidad |
| RNF-13 | Un arrendatario no puede tener dos inmuebles arrendados al mismo tiempo | Usabilidad |
| RNF-14 | El sistema le permite a un arrendador tener un máximo de 5 imágenes por inmueble | Usabilidad |
| RNF-15 | El sistema le permite a un usuario un máximo de 200 caracteres para realizar un comentario | Operaciones |
| RNF-16 | El sistema rectifica las fechas libres del inmueble | Usabilidad |
| RNF-17 | El sistema actualiza la disponibilidad del inmueble | Operaciones |
| RNF-18 | El sistema muestra mensajes solamente entre dos personas | Usabilidad |
| RNF-19 | El sistema permite eliminar un mensaje dentro del chat en los primeros 20 minutos | Soporte |

Tabla 9. Requisitos no funcionales.

### Restricciones de Software

Los lenguajes que se definieron a utilizar para el desarrollo de la aplicación web son HTML, CSS, JS, con el apoyo de la librería *React* y el *backend* estará proporcionado por *Firebase* con sus servicios de autenticación y almacenamiento de datos. También se implementarán las *API’s* proporcionada por Google como son *Maps* y OpenStreetMap *Nominatim*. Esta aplicación cuenta con una licencia de software libre.

### Restricciones de Hardware

Por parte del servidor se está utilizando *Firebase*, por lo que es necesario una conexión a internet para esta comunicación.

Ya por parte del cliente se necesita como requisitos mínimos de funcionamiento: en caso de Google Chrome un procesador Intel Pentium 4 o posterior que sea compatible con SSE2, 128 MB de RAM, más de 100MB de disco duro y una versión superior de Windows 7. En caso de MacOS un procesador Intel-Based Macs (2006+), memoria RAM 128MB, más de 100MB de disco duro y un sistema operativo OS X versión 10.5.6+.

## Suposiciones y dependencias

Para el correcto desarrollo del proyecto es necesario definir una lista de suposiciones y dependencias en el cual se permita evidenciar lo que puede afectar al desarrollo del proyecto:

* La información entregada tanto por el arrendador como por el arrendatario en cualquier etapa tanto de registro como de modificación o agregador de datos será verdadera y fiable.
* Los usuarios que desean utilizar la plataforma deben contar con un dispositivo de cómputo con Google Chrome en una versión superior a la 72.0.3626 (2019) para garantizar un correcto funcionamiento de todos los módulos del sistema.
* La base de datos estará disponible para cumplir tareas de solicitud y envío de información de manera satisfactoria, sin fallos de servicio.
* El *host* por utilizar proveerá las comunicaciones necesarias de manera adecuada sin presentar fallos.
* Los *backups* siempre tomarán como punto de generación el registro inmediatamente anterior para llevar un control adecuado.
* Cualquier usuario que desee ingresar a la plataforma deberá contar con acceso a internet y conocimiento básico de cómo usarlo.
* Cualquier cambio bien sea en documentación o código del proyecto debe ser comunicado inmediatamente a todos los miembros del grupo, realizándolo de la manera menos invasiva posible, bien sea como una versión alterna o en caso de código en un *Branch* diferente.
* La comunicación e identificación de necesidades entre el *Product* *Owner* y los S*take Holders* es precisa y clara desde el paso a documentación.

# Requisitos específicos

## Características del producto software

En esta sección se enumeran los requisitos funcionales [ver anexo (Requisitos)](https://drive.google.com/file/d/1e6_7gbha8D8ZAoZyBzo5NWukc4b4ejiR/view?usp=sharing) en donde se explican todas las características que debe poseer el sistema para poder implementarlas. De igual forma, se relacionaron estos requisitos con los casos de uso para que de esta forma se puedan cumplir con las características de trazabilidad, se realizó esto ya que facilita la ubicación del requisito en caso de realizar una modificación o implementación.

Por medio de los requisitos, se pueden detallar las características que debe poseer el sistema para que desempeñe todas las funcionalidades planteadas para el beneficio el usuario (arrendatario y arrendador), por medio de las entradas, procesando y generando salidas. Se tuvo en cuenta de los posibles cambios que se pueden ir realizando a lo largo del proyecto, es por esta razón, que se tuvo en cuenta la trazabilidad y la capacidad de modificar estos requisitos conociendo el impacto que le generaría al sistema.

## Requisitos de desempeño

Teniendo los requisitos funcionales, se empezó con el desarrollo de los requisitos de desempeño, estos se pueden dividir en requisitos estáticos y dinámicos. Los de desempeño, tienen la cantidad de usuarios concurrentes o simultáneos que el sistema debe soportar (según las especificaciones planteadas anteriormente), las terminales o equipos utilizados, etc.; los requisitos dinámicos tienen en cuenta factores como el número de transacciones que debe soportar el sistema, los tiempos de respuesta del sistema realizada por los usuarios previamente.

Se tuvo en cuenta los siguientes atributos: latencia, tasa de procesamiento, capacidad, disponibilidad. Estos atributos se pueden observar en el anexo requisitos de desempeño [ver anexo(Requisitos Desempeño)](https://drive.google.com/file/d/1oAYOblX4cuEbr_5lWqbYZKZ8J9YQ8VPq/view?usp=sharing), se realizaron para que sea más fácil comprobar el desempeño del sistema al momento de estar en las fases de verificación y validación. En el anexo de igual forma, se pueden observar las 4 categorías, que son: atención en simultaneo, capacidad de almacenamiento, actualización de base de datos, tiempo de respuesta están relacionadas con el tipo de requisito estático y dinámico, para que el usuario tenga una mejor comprensión.

## Atributos del sistema software

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***# Requisito*** |  | ***Tipo de requisito*** | | ***Confiabilidad*** | ***Casos de uso asociados*** | |  |
| ***Descripción*** | ***El mecanismo para el manejo de la información es el otorgado por Firebase con Cloud Firestore para el manejo de datos de texto, y el Storage para el manejo de archivos multimedia. Ya que Firebase se encuentra en la nube esta va a estar disponible la mayoría del tiempo en que se ejecute la aplicación.*** | | | | | | |
| ***Razón*** |  | | | | | | |
| ***Autor*** | ***ECO*** | | | | | | |
| ***Criterio de medición*** |  | | | | | | |
| ***Prioridad*** |  | | ***Módulo asociado*** | | |  | |
| ***Versión*** |  | | ***Fecha*** | | | ***14/10/2020*** | |

Tabla 10 Atributos del sistema software - Confiabilidad.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***# Requisito*** |  | ***Tipo de requisito*** | | ***Disponibilidad*** | ***Casos de uso asociados*** | |  |
| ***Descripción*** | ***El sistema va a mantener una disponibilidad del 97% semanal lo que estará disponible 163 horas a la semana. Ya esto también puede ser afectado por el llamado de API’s exteriores como lo son Maps y PSE.*** | | | | | | |
| ***Razón*** | ***Ya que Firebase se encuentra en Cloud da esta disponibilidad*** | | | | | | |
| ***Autor*** | ***ECO*** | | | | | | |
| ***Criterio de medición*** | ***La disponibilidad será medida en base a la relación del tiempo en que el sistema esté disponible y el tiempo total en el que se ejecute la medición (tiempo disponible / tiempo de medición)*** | | | | | | |
| ***Prioridad*** | ***Baja*** | | ***Módulo asociado*** | | | ***Sistema General*** | |
| ***Versión*** | ***1.0*** | | ***Fecha*** | | | ***14/10/2020*** | |

Tabla 11 Atributos del sistema software - Disponibilidad.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***# Requisito*** |  | ***Tipo de requisito*** | | ***Seguridad*** | ***Casos de uso asociados*** | |  |
| ***Descripción*** | ***El sistema va a mantener una verificación hash para las contraseñas, esto permite el manejo de información sensible de cada usuario con respecto a su cuenta. Así mismo la verificación de roles para poder desplegar las diferentes funcionalidades correspondientes a cada uno.*** | | | | | | |
| ***Razón*** | ***La información sensible de los usuarios debe viajar a través de un medio inseguro como lo es el Internet*** | | | | | | |
| ***Autor*** | ***ECO*** | | | | | | |
| ***Criterio de medición*** |  | | | | | | |
| ***Prioridad*** | ***Baja*** | | ***Módulo asociado*** | | | ***Autenticación*** | |
| ***Versión*** | ***1.0*** | | ***Fecha*** | | | ***14/10/2020*** | |

Tabla 12 Atributos del sistema software - Seguridad.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***# Requisito*** |  | ***Tipo de requisito*** | | ***Mantenibilidad*** | ***Casos de uso asociados*** | |  |
| ***Descripción*** | ***El sistema cuenta con una arquitectura MVC, aquí la vista se encuentra también divida por componentes para un mejor manejo de esta, ya el modelo cuenta con todas las funciones de lectura y escritura de Firebase cada una de las entidades, para garantizar la persistencia de sus datos. Cada uno de estos componentes se encuentra comentado por si alguna nueva persona desea conocer el funcionamiento del sistema o si desea modificarlo.*** | | | | | | |
| ***Razón*** | ***El código fuente del sistema debe ser fácilmente modificado para dar cumplimiento con los requisitos del cliente*** | | | | | | |
| ***Autor*** | ***ECO*** | | | | | | |
| ***Criterio de medición*** |  | | | | | | |
| ***Prioridad*** | ***Alta*** | | ***Módulo asociado*** | | | ***Sistema General*** | |
| ***Versión*** | ***1.0*** | | ***Fecha*** | | | ***14/10/2020*** | |

Tabla 13 Atributos del sistema software - Mantenibilidad.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ***# Requisito*** |  | ***Tipo de requisito*** | | ***Portabilidad*** | ***Casos de uso asociados*** | |  |
| ***Descripción*** | ***Nuestro alcance para el desarrollo solo nos permite un desarrollo web para dispositivos con una pantalla mínima de 1080x720, con lo cual su soporte esta dado por los principales navegadores como son Chrome y Firefox.*** | | | | | | |
| ***Razón*** | ***Establecido de esta forma debido al alcance del proyecto*** | | | | | | |
| ***Autor*** | ***ECO*** | | | | | | |
| ***Criterio de medición*** | ***La portabilidad será medida en base la relación entre los dispositivos en los cuales el sistema se despliegue correctamente y los dispositivos sobre los cuales se realicen las pruebas de portabilidad (dispositivos aceptados por el sistema / dispositivos probados)*** | | | | | | |
| ***Prioridad*** | ***Media*** | | ***Módulo asociado*** | | | ***Interfaz Grafica*** | |
| ***Versión*** | ***1.0*** | | ***Fecha*** | | | ***14/10/2020*** | |

Tabla 14 Atributos del sistema software - Portabilidad.

## Requisitos de la base de datos

### Tipos de datos almacenados

Los tipos de datos que permite *Firebase* y los cuales serán utilizados dentro del proyecto:

|  |  |  |
| --- | --- | --- |
| Tipo de Dato | Descripción | Ejemplos |
| *String* | Secuencias de caracteres (letras, símbolos y números) | Nombres, Apellidos, Direcciones |
| *Number* | Datos numéricos enteros o reales | Precios, cantidades, valoraciones |
| *Boolean* | Datos con valores falso o verdadero | Validación de contenido, |
| *timestamp* | Representa una fecha con hora | Fechas de inicio de actividades, Fecha de fin de actividades |
| *Geopoint* | Parejas de valores numéricos que representan la latitud y la longitud de una ubicación geográfica | Localización de los inmuebles |
| *Arrays* | Secuencias de datos de diferentes tipos (descritos anteriormente) | Chats, historiales, inmuebles agregados |

Tabla 15 Tipo de datos almacenados

### Tipos de Consultas utilizadas

El tipo de consulta que soporta *Firebase* se base en *Paths* o rutas, que acceden a los colecciones o documentos, de allí se extrae la información pertinente dependiendo de los filtros de cada consulta para su respectivo despliegue de información al cliente.

### Indexación de los datos

Se presenta el diagrama de bases de datos no relacionales en el siguiente Schema Json [ver anexo (Schema Json)](https://drive.google.com/file/d/1kvEl5C5IHUjku_boMOJKjHuioTkjqn90/view?usp=sharing)

### Frecuencia de acceso

El acceso a la base de datos es frecuente, ya que esta responde a las consultas realizadas por cada usuario activo en el sistema. Con lo que se cuenta con un máximo de 500 usuarios activos para poder realizar las peticiones de datos.

# Proceso de ingeniería de requisitos

## Planeación

En esta sección se presenta el plan que el equipo de trabajo planteó para llevar a cabo la administración de requisitos para el desarrollo de Buen Vecino. Se presentan los encargados y el proceso definido para manejar y controlar los cambios en los requisitos del sistema con el propósito de llevar una buena calidad del software al guiarnos por lo establecido en la sección 9.1 Administración de requisitos del anterior documento, el *Software Project Management Plan*.

### Encargados

Los encargados de realizar el proceso de planeación de requisitos son el encargado de desarrollo, el *Scrum* Máster y el *Product* *Owner*. Sin embargo, los demás miembros en del *Scrum* *Team* son libres de dar su opinión respecto a cómo se maneja la planificación de los requisitos del sistema para lograr requisitos de mayor calidad.

### Frecuencia

El equipo de trabajo de ECO determinó que la planeación de requisitos se debía hacer con un amplio margen de tiempo pues la documentación de los requisitos y el seguimiento a la planeación son puntos clave para el correcto desarrollo del proyecto. Durante el desarrollo, el equipo de trabajo estuvo abierto a hacer modificaciones sobre estos requisitos o incluir nuevos.

### Desarrollo del proceso

Para el desarrollo del proceso, se definió inicialmente, empezar con la evaluación de los requisitos, el encargado de esta tarea tendrá que verificar que los requisitos cumplan con los criterios establecidos del sistema, si el encargado de la tarea encuentra algún comentario respecto a alguno de los requisitos este pasará un estado de revisión, donde tendrá que ser revisado para que, de acuerdo al comentario planteado se puedan realizar cambios al requisito si es necesario o rechazar el requisito del sistema , una vez el requisito esté aprobado podrá pasar a la fase de documentación del mismo. El desarrollo se describe en el siguiente diagrama:

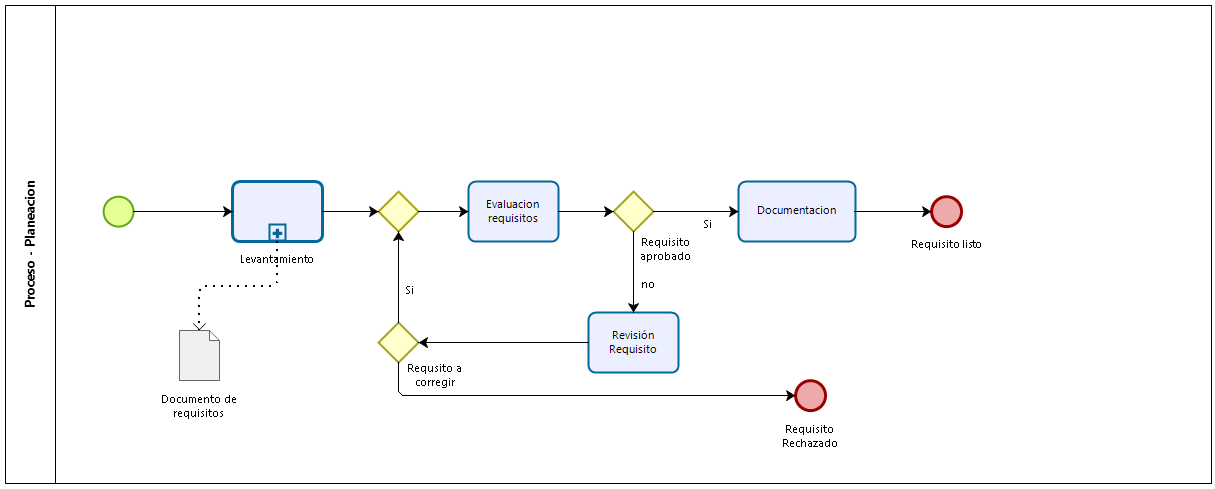


Ilustración 1. Desarrollo del proceso.

Para llevar un manejo claro de los requisitos se decidió realizar el manejo de versiones sobre el documento que incluye los requisitos del sistema, donde:

· El primer número, asciende cuando se realiza la modificación dentro del proceso de planificación. Principalmente porque el requisito tendrá un cambio muy grande.

· El segundo número, que está después del punto, asciende cuando un requisito es modificado en la revisión de este.

## Levantamiento

El levantamiento es el punto de inicio para cualquier modelo o proyecto de desarrollo de software. En este se pueden encontrar todos los detalles y pasos a seguir para llevar a cabo cada funcionalidad del sistema, como también las múltiples características con las que contará el software, el proceso de levantamiento empieza con un hito de observación, en el cual se evidencian los puntos críticos a adoptar, luego de ello por medio de reuniones grupales se consolidan las ideas y posteriormente se refinan para documentarse, cada refinamiento permite entrar nuevamente en un proceso de análisis, en el que al final el requisito será aceptado.

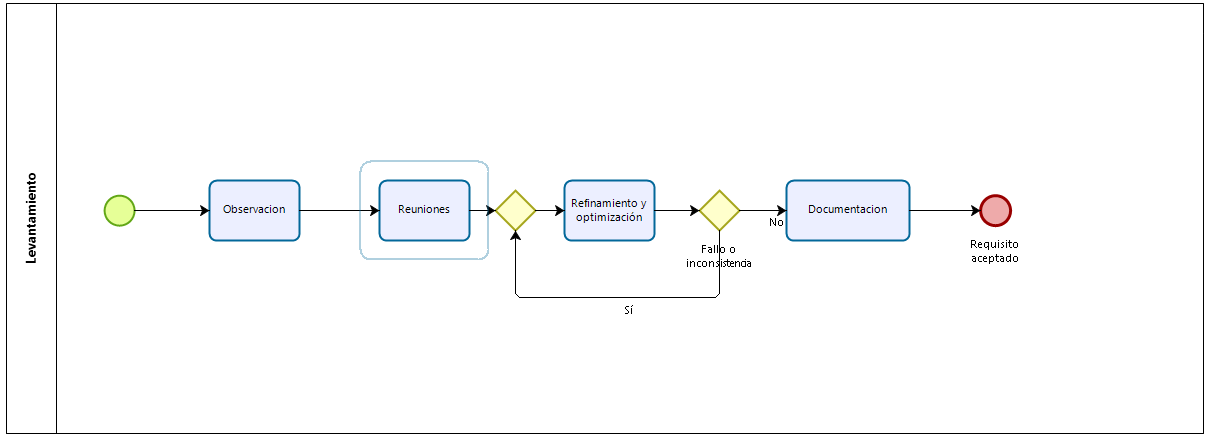


Ilustración 2. Levantamiento.

* Observación: básicamente se planteó la idea desde un miembro del grupo, el cual siempre ha tenido problemas a la hora de buscar una vivienda en Bogotá para hospedarse. Desde este punto se empezaron a tomar perspectivas grupales para poder encontrar una solución viable y eficiente.
* Reuniones: Para el grupo siempre ha sido vital el reunirse consistentemente para dialogar los futuros cambios y puntos críticos del proyecto para llevar una correcta gestión, por tanto, esto fue desarrollado en la primera entrega, sin embargo, para el SRS se realizaron múltiples reuniones con el fin de evaluar el proceso y estado del proyecto.
* Refinamiento: El énfasis de esta etapa es garantizar que cada requisito esté lo más pulido posible, esto con el fin de entregar un software tanto funcional a nivel de software como bien documentado para su posterior archivado o puesta en producción.

## Especificación

En esta sección del documento se encuentra la especificación de los requisitos con el fin que estos puedan ser utilizados y comprendidos por el equipo de desarrollo. La plantilla que se utilizó para esta especificación de requisitos fue la matriz de trazabilidad que se puede encontrar en el siguiente [ver anexo (Matriz de trazabilidad)](https://drive.google.com/file/d/1YuhbTb5xP-aTauvKWM0OfYcAi83YmQv8/view?usp=sharing). Dentro de la matriz de trazabilidad se encuentra la descripción y estado del requisito junto con el aspecto de aceptación y su prioridad. Esto es importante a la hora del desarrollo para identificar cual requisito requiere mayor atención y cuidado a la hora de la implementación. Para la siguiente entrega la matriz contará con las estrategias de prueba y comentarios sobre los defectos encontrados al probar los requisitos.

# Proceso de verificación y validación

En esta sección pueden encontrarse los procesos a desarrollar para validar y consolidar los diferentes requisitos del software Web Buenvecino, para ello a continuación se evidenciará cada uno de los diferentes métodos y formas a seguir para la validación de los requisitos.

## Proceso de validación

El proceso de validación consiste en una revisión general de la estructura y documentación del proyecto, enfocada a cada requisito. Esta documentación de procesos consiste en 2 actividades principales las cuales pueden ser definidas como la verificación de la estructura del requisito y la verificación de la estructura de la documentación.

## Validación de la estructura del requisito

En esta fase se hace la verificación sintáctica del requisito, es decir, cómo está escrito. El requisito debe cumplir con los criterios establecidos en [ver anexo (Verificación requisitos).](https://drive.google.com/file/d/1PP-55kMFfvFl-nkU_T4nje5UkN4a7Q89/view?usp=sharing) Para que el requisito sea aprobado debe tener la aprobación para cada uno de los criterios definidos, una vez el requisito cumpla con estos criterios, el requisito será incluido en los requisitos del sistema como un requisito valido.

## Validación de la estructura de la documentación

En esta actividad se realiza la validación del contexto de la documentación del requisito, es decir, si la documentación propuesta para el requisito es acorde a las especificaciones del sistema, a las reglas de negocio y se encuentra escrita de manera correcta y adecuada. Esta actividad se desarrolla en paralelo a la actividad anterior, a la validación de la estructura del requisito.

## Proceso de verificación

Con el fin de verificar que los requisitos establecidos para la aplicación cumplan con lo acordado en la documentación del requisito, se realizará el diseño asociado a cada uno de estos requisitos, junto con la implementación de estos en la plataforma, con esto se comprobará que el requisito realiza justamente la funcionalidad por la que fue incluido.

# Anexos

1. Diagrama de clases (Acceso en: [Enlace](Anexos/7.1%20Diagrama%20de%20Clases.png))
2. Diagrama modelo de dominio (Acceso en: [Enlace](Anexos/ModeloDeDominio.PNG))
3. Documentación modelo de dominio (Acceso en: [Enlace](Anexos/7.1%20DocumentacionModeloDominio.xlsx))
4. Diagrama Casos de uso (Acceso en: [Enlace](Anexos/7.3%20Diagrama%20Casos%20de%20Uso.png))
5. Documentación Casos de uso (Acceso en: [Enlace](Anexos/7.2.1-7.3%20CasosdeUsoECO-Documentados.docx))
6. Requisitos Funcionales (Acceso en: [Enlace](Anexos/8.1%20RequisitosIngeSoft.xlsx))
7. Documentación Requisitos Funcionales (Acceso en: [Enlace](Anexos/8.1%20RequisitoDocumentacion.docx))
8. Documentación Requisitos Desempeño (Acceso en: [Enlace](Anexos/8.2%20Requisitos%20de%20%20Desempeño.xlsx))
9. Schema Json (Acceso en: [Enlace](Anexos/8.4.3%20schema.json))
10. Matriz de Trazabilidad (Acceso en: [Enlace](Anexos/9.3%20Matriz%20de%20Trazabilidad%20ECO.xlsx))
11. Reporte Gerencial (Acceso en: [Enlace](Anexos/Reporte%20Gerencial.docx))
12. Diagrama de Gantt (Acceso en: [Enlace](Anexos/Diagrama%20de%20Gantt%20SRS.xlsx))

# Referencias

[1] “All you need to know about Firebase.” https://www.the-reference.com/en/blog/pieter-de-busschere/firebase (accessed Sep. 09, 2020).