

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 |
| 06/11/2020 | 0.1 | Resumen, Introducción | Juan Afanador, Santiago Roa |
| 07/11/2020 | 0.2 | Vista lógica del sistema | Gabriel Niño |
| 07/11/2020 | 0.3 | Vista física del sistema | John Gonzalez |
| 07/11/2020 | 0.4 | Vista de procesos del sistema | Gabriel Niño, Stiven Gonzalez |
| 10/11/2020 | 0.5 | Estructura del sistema | Santiago Roa, Juan Afanador |
| 10/11/2020 | 0.6 | Comportamiento del sistema | Santiago Roa, Stiven Gonzalez |
| 14/11/2020 | 0.7 | Vista lógica del sistema | Gabriel Niño |
| 15/11/2020 | 0.8 | Vista de procesos del sistema | Juan Afanador, John Gonzalez |
| 16/11/2020 | 0.9 | Persistencia | Santiago Roa |
| 18/11/2020 | 1.0 | Interfaz de usuario | Gabriel Niño |
| 18/11/2020 | 1.1 | Tabla de contenido, lista de figuras, lista de tablas. | Santiago Roa, Juan Afanador |

# Resumen

Este documento tiene como propósito dar a conocer la descripción del diseño de software para el aplicativo web Buen Vecino, cuyo objetivo principal es servir de intermediario virtual entre arrendador y arrendatario para aquellos estudiantes universitarios en Bogotá que necesiten de un lugar para hospedarse mientras cumplen con sus actividades académicas.

Para cumplir con el propósito de este documento, se inicia presentando la arquitectura del software definiendo la estructura y comportamiento general del sistema con un alto nivel de abstracción. Se incluye, además, la vista física y lógica de los procesos del aplicativo web. Después, se presenta el proceso de diseño detallado. Esto es, la estructura del sistema, representada por el modelo de diagrama de clases junto con el comportamiento del sistema donde se enuncian los procesos que conforman la columna vertebral del aplicativo web Buen Vecino. En esta sección también se encuentra la información necesaria para el correcto funcionamiento del aplicativo y la interfaz de usuario donde se describe la interacción usuario-sistema. Finalmente, se encuentran los anexos necesarios para complementar el entendimiento de este documento.

# Tabla de Contenidos

[1 Historial de Cambios 3](#_Toc57579033)

[2 Resumen 4](#_Toc57579034)

[3 Tabla de Contenidos 5](#_Toc57579035)

[4 Lista de Figuras 6](#_Toc57579036)

[5 Lista de Tablas 7](#_Toc57579037)

[6 Introducción 8](#_Toc57579038)

[7 Arquitectura 9](#_Toc57579039)

[7.1 Vista lógica del sistema 9](#_Toc57579040)

[7.2 Vista física del sistema 11](#_Toc57579041)

[7.3 Vista de procesos del sistema 13](#_Toc57579042)

[8 Diseño Detallado 18](#_Toc57579043)

[8.1 Estructura del sistema 18](#_Toc57579044)

[8.2 Comportamiento del sistema 18](#_Toc57579045)

[8.3 Persistencia 19](#_Toc57579046)

[8.4 Interfaz de usuario 23](#_Toc57579047)

[9 Anexos 26](#_Toc57579048)

[10 Referencias 27](#_Toc57579049)

# Lista de Figuras

[Figura 1 Diagrama de navegabilidad de usuario no registrado. 18](#_Toc57581100)

[Figura 2 Diagrama de navegabilidad de usuario arrendador. 19](#_Toc57581101)

[Figura 3 Diagrama de navegabilidad de usuario arrendatario. 20](#_Toc57581102)

# Lista de Tablas

[Tabla 1 Requisito de Hardware - Software 9](#_Toc57580879)

[Tabla 2 Componente Base de datos 10](#_Toc57580880)

[Tabla 3 Atributos Firebase de Inmueble 15](#_Toc57580881)

[Tabla 4 Atributos Firebase de Arrendador 16](#_Toc57580882)

[Tabla 5 Atributos Firebase de Arrendatario 16](#_Toc57580883)

[Tabla 6 Atributos Firebase de Casa 17](#_Toc57580884)

[Tabla 7 Atributos Firebase de Apartamento 17](#_Toc57580885)

[Tabla 8 Atributos Firebase de Chat 17](#_Toc57580886)

[Tabla 9 Atributos Firebase de Favorito 18](#_Toc57580887)

# Introducción

Este documento tiene como objetivo describir el diseño del software utilizado para el desarrollo de la página Buen Vecino. A medida que se vaya transcurriendo por las secciones, se encontrarán puntos relevantes para definir las funcionalidades entre el sistema y los usuarios de este.

Primeramente, en la introducción se dará a conocer brevemente la estructura del diseño planeado para Buen Vecino para hacerse una idea de cómo interactúan las partes y es de la siguiente manera: inicialmente se planteará la arquitectura del sistema, sección donde se describe la información más relevante para definir la estructura y comportamiento del sistema, manejando altos niveles de abstracción. Se debe mencionar cuáles son los componentes más gruesos del sistema, esto se puede visualizar por medio de diagramas de paquetes junto con su documentación. Además, es necesario incluir un diagrama de despliegue que permita visualizar la vista física del sistema pues es necesario describir las características mínimas de software y hardware solicitadas por el sistema. Después se procede a la sección de vista de procesos donde se analizan los principales procesos en los que los usuarios interactúan con el sistema. Finalmente, en la sección de diseño detallado se pueden consultar todos los detalles en cuanto a la estructura, comportamiento y diseño de bajo nivel del sistema.

Como producto tenemos por tanto un aplicativo web denominado 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 hospedaje, ya que el aplicativo notificará las comodidades de cada vivienda como aire acondicionado o *Wifi*, así como puntos de interés cercanos.

El aplicativo web además permite la edición de viviendas o solicitudes de vivienda. Esto se logra 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.

# Arquitectura

## Vista lógica del sistema

En esta sección se describen los principales componentes de Buen Vecino, para una mejor visualización de esto, se realizó el siguiente diagrama de paquetes [ver anexo.](Anexos/Diagrama%20de%20Paquetes.png)

El patrón arquitectónico de software que se escogió es el Modelo Vista Controlador (MVC) debido a una serie de razones especificadas a continuación:

* Flexibilidad para hacer modificaciones. Esto hace que las modificaciones que se realicen no afecten el proyecto entero ya que el modelo no depende directamente de las vistas, por ende, al realizar una modificación no se afectará la arquitectura completa.
* MVC soporta el desarrollo ágil y en paralelo por lo que se le atribuye que este proceso sea más beneficioso para el proyecto. Esto se da por que un miembro del equipo puede trabajar en las vistas del proyecto mientras que otro puede ir trabajando en el controlador o en los modelos.
* MVC tiene una mayor probabilidad de detección y corrección de errores pues aísla y contiene los bugs durante las pruebas.

Paquete Modelo: Corresponde a los principales paquetes de negocio que fueron identificados por el grupo de trabajo.

* **Arrendador:** Módulo de gestión del arrendador. Es la clase encargada del registro del usuario de tipo arrendador. Este usuario es el encargado de subir inmuebles a la plataforma para ser arrendados y recibir un pago por ello.
* **Arrendatario:** Módulo de gestión del arrendatario, esta es la clase encargada del registro del usuario de tipo arrendatario. Este usuario ingresa a la plataforma con el fin de buscar un inmueble para arrendar por un periodo de estadía determinado a cambio de un pago.
* **Inmueble:** Módulo de gestión de los inmuebles, se encarga de registrar el inmueble con información esencial para el conocimiento de los interesados en hospedar un inmueble.
* **Habitación:** Módulo de gestión de una habitación, es un tipo de inmueble.
* **Casa:** Módulo de gestión de una casa, es un tipo de inmueble.
* **Apartamento:** Módulo de gestión de un apartamento, es un tipo de inmueble.
* **Servicio:** Módulo que se encarga de gestionar los servicios extras de un inmueble en particular.
* **Chat:** Módulo encargado de comunicar mensajes entre dos usuarios.
* **Favorito:** Módulo encargado de gestionar los inmuebles favoritos de un usuario de tipo arrendatario.
* **Usuario:** Módulo encargado de gestionar las solicitudes y chats de los usuarios.
* **Mensaje:** Módulo que se encarga de llevar la información de un mensaje.
* **Pago:** Módulo que se encarga de llevar la información necesaria para realizar un pago.
* **Reservación:** Módulo que se encarga de la gestión que conlleva hacer una reserva sobre un inmueble.
* **Solicitud Reserva:** Módulo que contiene la información con respecto a una solicitud de reserva realizada por un arrendatario.
* **Valoración:** Módulo que se encarga de calificar a un arrendatario o inmueble.
* **SistemaBV:** Módulo que se encarga de conectar todos los demás componentes del sistema.

Paquete Controlador: describe como el controlador “controla” el modelo por medio de la arquitectura MVC.

* **Controlador**: Se encarga de comunicar la vista con el sistema.

Paquete Vista: Representan las interfaces de usuario de Buen Vecino.

* **Información Inmueble:** Pantalla en donde se pude ver toda la información del inmueble seleccionada por el usuario, de igual forma se observa un mapa con la ubicación de este
* **Landing:** Pantalla que presenta el inicio del sistema BuenVecino, se muestra la opción de buscar Inmueble, y las características del sistema BuenVecino
* **Búsqueda:** Pantalla que muestra la ubicación de los Inmuebles en un mapa ofrecido por Google Maps, y una descripción general del inmueble con su respectivo precio. Se encuentra la barra de búsqueda con la opción de buscar por barrio o localidad.
* **Chat:** Pantalla en donde se un arrendador se puede comunicar con un arrendatario y viceversa, en donde se muestran los mensajes anteriores y los chats que el usuario tenga
* **Gestionar Inmueble:** Pantalla en donde el arrendador puede observar todos los inmuebles que ha publicado con la información de cada uno y en donde puede modificar, eliminar o agregar otro inmueble
* **Gestionar Perfil:** Pantalla en la cual usuario, puede modificar la información registrada por el
* **Historial de Pagos:** Pantalla en la cual el arrendador puede observar los pagos que le han realizado, en caso tal que ya le hayan realizado algún pago, y donde un arrendatario puede ver los pagos que haya realizado
* **Mis Favoritos:** Pantalla en la cual, un arrendatario puede observar los inmuebles que más le gusten, para poder observarlos este tuvo que agregar inmuebles previamente
* **Modificar Inmueble:** Pantalla la cual le permite arrendador modificar la información ingresada del inmueble, en caso de que ingresara mal un dato o una imagen del inmueble
* **Notificaciones:** Pantalla en la cual el usuario puede verificar si paso algo mientras no estaba
* **Registrar Inmueble:** Pantalla en la cual se le muestra al arrendador la información que debe ingresar del inmueble, junto con la ubicación y las imágenes de este.
* **Registrar Usuario:** Pantalla en la cual se le muestra al usuario, recién creado, los datos que debe ingresar para poder validar su perfil, como lo son el nombre, su correo y si desea una foto de perfil

## Vista física del sistema

En esta sección se encuentran los componentes de hardware más relevantes para hacer uso del sistema de Buen Vecino. Además, se describen las características mínimas de hardware y software que debe cumplir el dispositivo del usuario. A continuación, se presenta el diagrama de despliegue [ver anexo (Diagrama de despliegue).](Anexos/Anexo_Diagrama_Despliegue.PNG) Es importante mencionar dado que la arquitectura del sistema se basa en el patrón Modelo-Vista-Controlador y modelo cliente-servidor; los componentes físicos más importantes son los siguientes: dispositivo electrónico con conexión a internet y el servidor de Firebase, utilizado para almacenamiento y autenticación y la pagina 000webhost para el hosting del programa.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | 1 | **Nombre** | | Dispositivo electrónico | **Tipo** | Dispositivo |
| **Descripción** | Dispositivo usado por el usuario con el fin de acceder a las funcionalidades de Buen Vecino*.* El dispositivo debe contar con acceso a internet. | | | | | |
| **Requisitos de Software** | | | **Descripción** | | | |
| Chrome / Safari | | | Navegador web en el cual el usuario puede acceder a las funcionalidades ofrecidas por Buen Vecino. | | | |
| Windows 7 (o versiones posteriores) | | | Sistema operativo del dispositivo electrónico que usa el usuario para acceder a la página de Buen Vecino. | | | |
| **Requisitos de Hardware** | | | **Descripción** | | | |
| 128 MB de RAM | | | Memoria interna de procesamiento del equipo del usuario para acceder a la aplicación. | | | |
| 100 MB de disco duro | | | Dispositivo de almacenamiento de datos. | | | |
| Procesador Intel Pentium 4 (o una versión posterior) | | | Procesador del equipo del usuario para acceder a la página de Buen Vecino. | | | |

Tabla 1 Requisito de Hardware - Software

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID** | 2 | **Nombre** | | Google Firebase | **Tipo** | Servidor |
| **Descripción** | Plataforma de desarrollo de Google que permite desarrollar aplicaciones tanto para web como para Android y *iOS*. | | | | | |
| **Componentes** | | | **Descripción** | | | |
| *Autenticación* | | | Proporciona los servicios de *backend* para la autenticación de usuarios y administradores de la aplicación | | | |
| *Firestore* | | | Componente que almacena y procesa el contenido generado por usuarios los cuales no son de tipo multimedia. | | | |
| *Storage* | | | Componente que almacena y procesa el contenido generado por usuarios como las fotos e información de dietas y ejercicios | | | |

Tabla 2 Componente Base de datos

## Vista de procesos del sistema

Los procesos en los que interactúa el usuario **no registrado** son los siguientes:

* **Registro**: este proceso inicia cuando un usuario no registrado en la plataforma desea registrar sus datos para crear una cuenta. El usuario no registrado debe presionar el botón “Regístrate” en la pantalla principal de Buen Vecino. Después, el usuario debe ingresar sus datos y presiona en el botón de “Aceptar”. Si no existe ningún inconveniente, el sistema registra al usuario y este puede iniciar sesión.
* **Geolocalizar inmueble**: el proceso inicia cuando un usuario ingresa a la plataforma y oprime el botón “buscar”. Esto lo direcciona a otra pantalla donde se visualiza un mapa y el usuario puede ingresar un nombre de una localidad o barrio para mirar los inmuebles disponibles para arrendar. Cabe mencionar que, al ser un usuario no registrado, este no puede realizar una solicitud de arrendamiento.

Los procesos en los que interactúa el usuario **arrendador** son los siguientes:

* **Buscar inmueble:** el proceso comienza cuando un usuario registrado inicia sesión en la plataforma y presiona el botón de “buscar”. Esto lo redirecciona a otra pantalla donde el usuario puede modificar filtros para acomodar la búsqueda a sus preferencias. Una vez el usuario tenga lista su búsqueda deberá presionar el botón “ok” para enviar la consulta al sistema y que este retorne los resultados en pantalla.
* **Registrar inmueble:** el proceso comienza cuando un usuario de tipo arrendador oprime el botón de “registra tu vivienda”. Se muestra una nueva pantalla donde el usuario de tipo arrendador debe ingresar datos acerca del inmueble a registrar. Cuando el usuario haya completado el registro de la vivienda deberá confirmar presionando el botón “ok”.
* **Geolocalizar inmueble**: el proceso inicia cuando un usuario ingresa a la plataforma y oprime el botón “buscar”. Esto lo direcciona a otra pantalla donde se visualiza un mapa y el usuario puede ingresar un nombre de una localidad o barrio para mirar los inmuebles disponibles para arrendar. Cabe mencionar que, al ser un usuario de tipo arrendador, este no puede realizar una solicitud de arrendamiento.
* **Gestionar inmueble:** el proceso inicia cuando un usuario de tipo arrendador selecciona la opción de “gestionar inmueble”. En una nueva pantalla se visualizan todos los inmuebles propiedad del usuario donde este puede escoger entre las opciones de editar, eliminar, o agregar en caso de que desee cargar un nuevo inmueble a la plataforma.
  + Agregar: una vez el usuario de tipo arrendador presione el botón de “agregar”, el sistema despliega una nueva pantalla donde el usuario debe llenar todos los campos solicitados.
  + Editar: una vez que el usuario oprima el botó de “editar”, el sistema debe mostrar en pantalla la información solicitada con respecto a un inmueble. Esto permite que el arrendador pueda modificar la información de sus propiedades y para efectuar estos cambios debe oprimir el botón de “guardar”.
  + Eliminar: si el arrendador desea eliminar un inmueble de su lista, este debe presionar el botón de “eliminar”. A continuación, el sistema debe generar un mensaje pidiendo la confirmación del usuario para ejecutar la acción.
* **Gestionar perfil:** el proceso comienza cuando un usuario de tipo arrendador selecciona la opción de “gestionar perfil”. Esto lo direcciona a una nueva pantalla donde se los datos del usuario donde este puede escoger entre las opciones de editar o eliminar para modificar su perfil cumpliendo con los requisitos de cada campo.
  + Editar: una vez que el usuario oprima el botó de “editar”, el sistema debe mostrar en pantalla la información solicitada por el usuario para que este pueda modificar dicho campo de su perfil. Esto permite que el arrendador pueda modificar la información de su perfil y para efectuar estos cambios debe oprimir el botón de “guardar”.
  + Eliminar: si el arrendador desea eliminar algún dato de su perfil, este debe presionar el botón de “eliminar”. A continuación, el sistema debe generar un mensaje pidiendo la confirmación del usuario para ejecutar la acción.
* **Calificar usuario arrendatario:** el proceso inicia cuando se termina el periodo de estadía de algún arrendatario en el inmueble que estaba hospedando. Una vez que se confirme la salida del arrendatario de la vivienda, el arrendador, propietario de dicha vivienda, se dirige al perfil y en el inmueble que estaba en arriendo le sale la opción de calificar y selecciona la calificación que considere pertinente, también puede escribir un comentario si lo desea.
* **Chatear con otro usuario:** el proceso inicia cuando un usuario desea enviar algún mensaje a otro usuario. El usuario que desea comunicarse con otro usuario debe oprimir el botón de “contactar” y a continuación el sistema debe mostrar en pantalla un recuadro en el cual el usuario puede enviar su mensaje al destinatario deseado.

Los procesos en los que interactúa el usuario **arrendatario** son los siguientes:

* **Buscar inmueble:** el proceso comienza cuando un usuario registrado inicia sesión en la plataforma y presiona el botón de “buscar”. Esto lo redirecciona a otra pantalla donde el usuario puede modificar filtros para acomodar la búsqueda a sus preferencias. Una vez el usuario tenga lista su búsqueda deberá presionar el botón “ok” para enviar la consulta al sistema y que este retorne los resultados en pantalla.
* **Geolocalizar inmueble**: el proceso inicia cuando un usuario ingresa a la plataforma y oprime el botón “buscar”. Esto lo direcciona a otra pantalla donde se visualiza un mapa y el usuario puede ingresar un nombre de una localidad o barrio para mirar los inmuebles disponibles para arrendar. Al ser un usuario de tipo arrendatario, este sí puede realizar una solicitud de arrendamiento.
* **Reservar inmueble:** el proceso inicia cuando un usuario registrado de tipo arrendatario hace una búsqueda de inmuebles, después de mirar las diferentes opciones de vivienda, el usuario oprime el botón “reservar”. A continuación, se muestra en pantalla las fechas en las que dicho inmueble está disponible. El arrendatario selecciona las fechas que se va a hospedar en el inmueble y el sistema se comunica con el arrendador, para que este confirme la estadía. Esto se realiza para que el arrendador pueda ver la calificación que este tiene y así pueda decidir si se puede quedar o no en su inmueble. Si el arrendador confirma la estadía, el sistema le notifica al arrendatario que la reserva se ha realizado de forma exitosa. En caso contrario, se le notifica al arrendador que no se puedo realizar la reserva.
* **Gestionar lista favoritos:** el proceso inicia cuando un usuario de tipo arrendatario selecciona la opción de “gestionar lista favoritos”. En una nueva pantalla se visualizan todos los inmuebles favoritos del usuario donde este puede escoger entre las opciones de editar, eliminar.
  + Agregar: Una vez que el usuario realiza una búsqueda de un inmueble, este puede oprimir el botón de “agregar a favoritos” que aparece sobre el inmueble.
  + Eliminar: si el arrendatario desea eliminar un inmueble de su lista de favoritos, este debe presionar el botón de “eliminar”. A continuación, el sistema debe generar un mensaje pidiendo la confirmación del usuario para ejecutar la acción.
* **Gestionar perfil:** el proceso comienza cuando un usuario de tipo arrendador selecciona la opción de “gestionar perfil”. Esto lo direcciona a una nueva pantalla donde se los datos del usuario donde este puede escoger entre las opciones de editar o eliminar para modificar su perfil cumpliendo con los requisitos de cada campo.
  + Editar: una vez que el usuario oprima el botó de “editar”, el sistema debe mostrar en pantalla la información solicitada por el usuario para que este pueda modificar dicho campo de su perfil. Esto permite que el arrendador pueda modificar la información de su perfil y para efectuar estos cambios debe oprimir el botón de “guardar”.
  + Eliminar: si el arrendador desea eliminar algún dato de su perfil, este debe presionar el botón de “eliminar”. A continuación, el sistema debe generar un mensaje pidiendo la confirmación del usuario para ejecutar la acción.
* **Calificar inmueble:** el proceso de calificar inmueble comienza en el momento que termina el periodo de estadía de un arrendatario en un inmueble. A continuación, el arrendatario tiene la opción de calificar el inmueble recientemente hospedado u omitir esta acción. Si el usuario decide omitir este paso, la página lo redireccionará a su perfil. En caso contrario, el arrendatario podrá asignar una a cinco estrellas donde 1 corresponde a “No lo recomiendo” y 5 corresponde a “Inmueble espectacular”.
* **Chatear con otro usuario:** el proceso inicia cuando un usuario desea enviar algún mensaje a otro usuario. El usuario que desea comunicarse con otro usuario debe oprimir el botón de “contactar” y a continuación el sistema debe mostrar en pantalla un recuadro en el cual el usuario puede enviar su mensaje al destinatario deseado.
* **Realizar pago:** este proceso comienza cuando al usuario le aceptan la solicitud de arrendar el inmueble, el sistema se comunica con pse el usuario ingresa los datos correspondientes y se realiza el pago
* **Consultar historial de pago:** este proceso comienza cuando un usuario autenticado por el sistema, de tipo arrendatario, desea consultar su historial de pago. Para esto se debe dirigir a la parte superior derecha de la pantalla y oprimir en su foto de perfil, a continuación, se desplegará un menú de opciones donde podrá seleccionar “Historial de pago”. Una vez estando en esa pantalla, el usuario puede visualizar los inmuebles que ha hospedado y cuanto ha de arriendo.

# Diseño Detallado

## Estructura del sistema

A continuación, se describe la estructura de todos los componentes del aplicativo web Buen Vecino, esto se puede visualizar de una mejor manera con el modelo de diagrama de clases donde se presenta la descripción de las abstracciones realizadas en el código sobre los objetos y actores que son participes del sistema. [Ver anexo (Diagrama de clases).](Anexos/Diagrama%20de%20Clases.png) Así mismo, se puede consultar la documentación de este diagrama en el siguiente anexo: [ver anexo (documentación diagrama de clases)](Anexos/DocumentacionClases.docx)

## Comportamiento del sistema

A continuación, se documentan los procesos que conforman la columna vertebral de la página web Buen Vecino. Cabe mencionar que dichos procesos representan la interacción del usuario con las funcionalidades principales de Buen Vecino. La documentación de estos procesos se puede visualizar en los diagramas de secuencia que fueron creados para ellos. Estos diagramas pueden ser consultados en el siguiente anexo: [ver anexo (Diagramas de secuencia)](Anexos/Diagrama%20de%20secuencia).

* **Registro:** Este proceso permite a nuevos usuarios de Buen Vecino crear una cuenta en la plataforma para poder hacer uso de las funcionalidades dependiendo del tipo de usuario seleccionado. Para visualizar los pasos y pantallas presentadas al usuario en este proceso, [ver anexo (Documentación CU)](Anexos/Documentación%20Casos%20de%20Uso.docx)
* **Buscar inmueble:** Este proceso le permite al usuario, autenticado por el sistema, hacer una búsqueda de inmuebles de acuerdo con sus preferencias utilizando un filtro para mejorar el resultado de la búsqueda. Una vez el usuario oprima el botón de buscar, se desplegará una lista con los diferentes inmuebles que concuerden con las preferencias del usuario. Para visualizar los pasos y pantallas para este proceso, [ver anexo (Documentación CU)](Anexos/Documentación%20Casos%20de%20Uso.docx)
* **Reservar inmueble:** El proceso de reservar inmueble le permite a un usuario de tipo arrendatario, autenticado por el sistema, solicitar una reserva de un inmueble en particular que esté disponible. Después de que el usuario oprima el botón de reserva, se le enviará una solicitud de arrendamiento al arrendador dueño de dicho inmueble. Este la puede aceptar o rechazar, si la acepta, se envía esta confirmación al usuario arrendatario para que efectúe el pago usando la pasarela de pagos de PSE.
* **Gestionar datos de usuario:** Este proceso permite a los usuarios de tipo arrendador y arrendatario gestionar su propio perfil en el aplicativo web. Es decir, pueden agregar, modificar o eliminar datos como dirección, edad, correo, teléfono, etc. Para visualizar los pasos y pantallas para este proceso, [ver anexo (Documentación CU)](Anexos/Documentación%20Casos%20de%20Uso.docx)
* **Gestionar datos de inmueble:** Este proceso permite a los usuarios de tipo arrendador gestionar sus inmuebles cargados al aplicativo web Buen Vecino. Es decir, pueden cargar o eliminar inmuebles para poner en arrendamiento, así mismo como modificar cualquier dato que requiera el inmueble. Para visualizar los pasos y pantallas para este proceso, [ver anexo (Documentación CU)](Anexos/Documentación%20Casos%20de%20Uso.docx)

## Persistencia

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | BD-01 | **Nombre:** | Inmueble |
| **Atributo** | **Tipo de dato** | **Restricciones** | **Detalles** |
| idInmueble | String |  | El id es proporcionado por el sistema |
| Tipo | String | Solo recibe los caracteres “c”, “h”, “a” | Para saber si es una casa, una habitación o un apartamento |
| Nombre | String | No debe ser mayor a 30 caracteres | Nombre del inmueble proporcionado por el dueño |
| Precio | Integer | - | El precio que el arrendador considera para el inmueble |
| Descripción | String | No debe ser mayor a 150 caracteres | La descripción del inmueble proporcionada por el dueño de este |
| nBaños | Integer | Se tiene como un mínimo de 1 | La cantidad de baños que tiene el inmueble |
| area | Integer | Debe ser un número mayor a 1 | El área en m2 que tiene el inmueble |
| esAmoblado | Boolean | - | Si el inmueble tiene o no muebles para la comodidad del cliente |
| idPropietario | String | - | El id del arrendador que publico el inmueble |

Tabla 3 Atributos Firebase de Inmueble

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | BD-02 | **Nombre:** | Arrendador |
| **Atributo** | **Tipo de dato** | **Restricciones** | **Detalles** |
| idArrendador | String | - | El id es proporcionado por el sistema |
| Dni | Integer | Expresiones regulares que sirven para validad la información | El número de identificación del usuario |
| TipoDni | String | Solo recibe los caracteres “cc”, “ce”, “ti”, “pa” | El tipo de identificación que tiene el usuario |
| Nombre | String | No debe ser mayor a 50 caracteres | Nombre y apellidos del usuario |
| genero | String | Solo recibe los caracteres “M”, “F”, “O”, “N” | El género del usuario |
| FechaNacimiento | Integer | - | La fecha de nacimiento del usuario |
| email | String | Expresiones regulares que sirven para validar la información del correo | El correo que usara para ingresar al sistema |
| telefono | Integer | Debe ser un número mayor a 1 | El teléfono del usuario que se usara par |

Tabla 4 Atributos Firebase de Arrendador

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | BD-03 | **Nombre:** | Arrendatario |
| **Atributo** | **Tipo de dato** | **Restricciones** | **Detalles** |
| idArrendador | String |  | El id es proporcionado por el sistema |
| Dni | Integer | Expresiones regulares que sirven para validad la información | El número de identificación |
| TipoDni | String | Solo recibe los caracteres “cc”, “ce”, “ti”, “pa” | El tipo de identificación que tiene el usuario |
| Nombre | String | No debe ser mayor a 50 caracteres | Nombre y apellidos del usuario |
| genero | String | Solo recibe los caracteres “M”, “F”, “O”, “N” | El género del usuario |
| FechaNacimiento | Integer | - | La fecha de nacimiento del usuario |
| email | String | Expresiones regulares que sirven para validar la información del correo | El correo que usara para ingresar al sistema |
| telefono | Integer | Debe ser un número mayor a 1 | El teléfono del usuario que se usara par |

Tabla 5 Atributos Firebase de Arrendatario

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | BD-04 | **Nombre:** | Casa |
| **Atributo** | **Tipo de dato** | **Restricciones** | **Detalles** |
| NHabitaciones | Integer | Tiene que ser un número mayor o igual a 1 | El número de habitaciones que tiene la casa |
| esCompartido | Boolean | - | Si la casa va a ser arrendada por alguien más |
| nPisos | Integer | Tiene que ser un número mayor o igual a 1 | La cantidad de pisos que tiene la casa |
| nCocinas | Integer | Tiene que ser un número mayor o igual a 1 | La cantidad de cocinas que tiene la casa |

Tabla 6 Atributos Firebase de Casa

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | BD-05 | **Nombre:** | Apartamento |
| **Atributo** | **Tipo de dato** | **Restricciones** | **Detalles** |
| NHabitaciones | Integer | Tiene que ser un número mayor o igual a 1 | El número de habitaciones que tiene el apartamento |
| esCompartido | Boolean | - | Si el apartamento va a ser arrendada por alguien más |

Tabla 7 Atributos Firebase de Apartamento

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | BD-06 | **Nombre:** | Chat |
| **Atributo** | **Tipo de dato** | **Restricciones** | **Detalles** |
| Usuario1 | String | No debe ser mayor a 350 caracteres | El usuario que le envía el mensaje al otro, para empezar la conversación |
| Usuario2 | String | No debe ser mayor a 350 caracteres | El receptor del mensaje enviado |
| Mensajes | String | Es un arreglo que tiene los mensajes que se han mandado | Guarda la conversación que han tenido los usuarios |

Tabla 8 Atributos Firebase de Chat

|  |  |  |  |
| --- | --- | --- | --- |
| **Identificador:** | BD-07 | **Nombre:** | Favorito |
| **Atributo** | **Tipo de dato** | **Restricciones** | **Detalles** |
| idInmueble | String |  | El id es proporcionado por el sistema |
| fechaAgreagado | Number | Expresiones regulares que permitan validar la información de la fecha | La fecha en la que el arrendatario agrego ese inmueble a la lista de favoritos |
| Comentario | String | No debe ser mayor a 150 caracteres | Si el arrendatario le quiere agregar un comentario privado al inmueble |

Tabla 9 Atributos Firebase de Favorito

## Interfaz de usuario

En esta sección se describe el orden de navegación entre las distintas pantallas que posee el aplicativo web y cómo el usuario se puede desplazar dentro del sistema. Para esto se realizaron dos diagramas WBS los cuales contienen las diferentes funcionalidades con las que cuenta el sistema. Pese a que los dos tipos de usuario cumplen muchas funcionalidades similares, existen algunas que son únicas propias del tipo de usuario, por esta razón se realizaron dos diagramas WBS. Se puede destacar el diseño de la interfaz de geolocalización de usuario pues su desarrollo e implementación fue bastante tediosa para el grupo de trabajo. Esto provocó unos retrasos con el desarrollo de la interfaz ya que se debía tener buenos conocimientos acerca de las herramientas utilizadas para hacer la geolocalización de un inmueble utilizando su dirección y coordenadas de latitud y longitud.

Diagrama

Descripción generada automáticamente

Figura 1 Diagrama de navegabilidad de usuario no registrado.

Gráfico de cajas y bigotes

Descripción generada automáticamente

Figura 2 Diagrama de navegabilidad de usuario arrendador.

Interfaz de usuario gráfica

Descripción generada automáticamente

Figura 3 Diagrama de navegabilidad de usuario arrendatario.

# Anexos

1. Diagrama de paquetes (Acceso en: [enlace](Anexos/Diagrama%20de%20Paquetes.png))
2. Diagrama de despliegue (Acceso en: [enlace](Anexos/Anexo_Diagrama_Despliegue.PNG))
3. Diagramas de actividad (Acceso en: [enlace](Anexos/Diagrama%20de%20actividad))
4. Diagrama de clase (Acceso en: [enlace](Anexos/Diagrama%20de%20Clases.png))
5. Documentación diagrama de clases (Acceso en: [enlace](Anexos/DocumentacionClases.docx))
6. Documentación casos de uso (Acceso en: [enlace](Anexos/Documentación%20Casos%20de%20Uso.docx))
7. Diagramas de secuencia (Acceso en: [enlace](Anexos/Diagrama%20de%20secuencia))
8. Schema Json (Acceso en: [enlace](Anexos/schema.json))
9. Reporte gerencial (Acceso en: [enlace](Anexos/Reporte%20Gerencial.docx))
10. Post Mortem (Acceso en: [enlace](Anexos/Post%20Mortem.docx))

# Referencias