**MANUAL TECNICO**

**SOFTWARE PARA EL MANEJO DE USUARIOS Y LA REALIZACION DE RESERVAS PERSONALIZADAS PARA EL CONJUNTO RESIDENCIAL RECODO DE CEDRO SUBA**

**JOSEPH NICOLAS VARON VARGAS**

**JUAN SEBASTIAN CASTRO MARTINEZ**

**FICHA: 2771440V2**

**SERVICIO NACIONAL DE APRENDIZAJE SENA**

**CENTRO DE ELECTRICIDAD, ELECTRONICA Y TELECOMUNICACIONES CEET**

**TECNICO EN PROGRAMACION DE SOFTWARE**

**BOGOTA D.C., SEPTIEMBRE 2012**

**INTRODUCCION**

Este manual describe una solución tecnológica para mejorar el proceso de agendamiento en el Conjunto Residencial Recodo de Cedro Suba. Actualmente, se utilizan métodos anticuados como hojas sueltas y libros de registro, lo que causa problemas de organización y eficiencia.

El nuevo sistema propuesto es una aplicación digital que centralizará toda la información de agendamientos. Sus principales características incluyen:

* Una base de datos unificada
* Una interfaz fácil de usar
* Acceso para múltiples usuarios desde diferentes dispositivos

Con esta herramienta, los administradores podrán crear y modificar citas, registrar asistencias y generar informes automáticamente. Esto ayudará a reducir errores, ahorrar tiempo y facilitar el análisis de datos.

La implementación se realizará gradualmente, empezando por instalar el software y trasladar la información existente. Luego, se capacitará al personal y se hará una prueba antes de usarlo definitivamente.

### **CONTENIDO**

### **INTRODUCCION ................................................................... 2**

### **1. REQUERIMIENTOS TÉCNICOS.......................................................... 4**

### **2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO .......................... 5**

### **2.1.1 PHP.......................................................................................................... 5**

### **2.2.2 DISEÑO WEB.......................................................................................................... 5**

### **2.2.3 HTML.......................................................................................................... 5**

### **2.2.4 CSS.......................................................................................................... 5**

### **2.2.5 JAVASCRIPT.......................................................................................................... 6**

### **2.3 MYSQL .................................................................................................... 6**

### **2.4 XAMPP.................................................................................................. 6**

### **2.4 AZURE JUNTO A MYSQL WORKBENCH............................................ 7**

### **3 CASOS DE USO............................................ 7**

**LISTA DE TABLAS CU**

1. **REQUERIMIENTOS TÉCNICOS.**

**REQUERIMIENTOS TÉCNICOS**

**REQUERIMIENTOS MÍNIMOS DE HARDWARE**

* Procesador: Intel Pentium 4 o equivalente en Amd
* Memoria RAM: Mínimo: 1 Gigabyte (GB)
* Disco Duro: 500 GB de espacio libre

**REQUERIMIENTOS MÍNIMOS DE SOFTWARE**

* Sistema Operativo: Windows 7 o posterior
* Navegador web: Google Chrome (versión 50 o posterior), Mozilla Firefox (versión 45 o posterior), o Microsoft Edge

**Notas adicionales:**

* Se recomienda una conexión a Internet de al menos 1 Mbps para un rendimiento óptimo.
* Aunque Gmail puede funcionar con especificaciones inferiores, estos requisitos aseguran una experiencia de usuario fluida.
* Para sistemas operativos más antiguos como los mencionados (Windows NT/98/Me/2000/2003/XP/Vista), es posible que algunas funciones de Gmail no estén disponibles o el rendimiento sea inferior.

**2. HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO**

**2.1 PHP**

PHP es un lenguaje de programación diseñado para crear páginas web dinámicas. Se integra fácilmente con HTML, permitiendo a los desarrolladores combinar código PHP dentro del marcado HTML.

Las ejecuciones de PHP se realizan en el servidor, y el cliente (navegador web) recibe solo los resultados de estas ejecuciones. Cuando un cliente hace una petición, el servidor procesa el código PHP y genera contenido dinámico que se envía de vuelta al cliente.

Una de las principales ventajas de PHP es su capacidad para conectarse con diversos sistemas de gestión de bases de datos, incluyendo MySQL, Oracle, PostgreSQL y SQL Server. Esto permite crear aplicaciones web robustas y escalables.

PHP puede ejecutarse en la mayoría de los sistemas operativos modernos y es compatible con los servidores web más populares. Su versatilidad lo hace ideal para desarrollar desde sitios web simples hasta complejas aplicaciones empresariales, sistemas de gestión de contenidos (CMS) y plataformas de comercio electrónico.

Además, PHP cuenta con un amplio ecosistema de frameworks y bibliotecas que facilitan el desarrollo rápido y eficiente de aplicaciones web modernas.

**2.2 DISEÑO WEB**

**2.2.1 HTML (HyperText Markup Language):**

* Propósito: Es el lenguaje de marcado principal para estructurar el contenido en la web.
* Características: Utiliza etiquetas para definir elementos como encabezados, párrafos, enlaces, listas y otros componentes de la página. HTML establece la estructura básica de una página web.

**2.2.2 CSS (Cascading Style Sheets):**

* **Propósito:** Se usa para estilizar y diseñar la apariencia de los elementos HTML.
* **Características:** Permite aplicar estilos como colores, fuentes, márgenes y layout a los elementos HTML. CSS ayuda a hacer que las páginas web sean visualmente atractivas y coherentes con el diseño deseado.

**2.2.3 JavaScript:**

* **Propósito:** Es un lenguaje de programación que se usa para agregar interactividad y funcionalidad dinámica a las páginas web.
* **Características:** Permite manipular el contenido HTML y CSS, responder a eventos del usuario (como clics y desplazamientos), y realizar operaciones más avanzadas como validación de formularios y manipulación de datos.

**2.3 MYSQL WORKBENCH**

MySQL Workbench es una herramienta gráfica desarrollada por Oracle que se utiliza para el diseño, desarrollo y administración de bases de datos MySQL. Proporciona una interfaz visual para gestionar y modelar bases de datos, crear y ejecutar consultas SQL, y administrar usuarios y permisos. También permite la administración de instancias de MySQL, la realización de copias de seguridad y la restauración de datos, así como la generación de informes. Su objetivo es simplificar y agilizar las tareas relacionadas con la gestión de bases de datos MySQL.

**2.4 XAMPP**

XAMPP es una solución integral de software libre diseñada para simplificar el desarrollo web local. Incluye los siguientes componentes clave:

Apache: Un servidor web de código abierto que permite gestionar y servir contenido web. Es ampliamente utilizado debido a su flexibilidad y robustez.

MySQL/MariaDB: Sistemas de gestión de bases de datos relacionales. MySQL es conocido por su rendimiento y fiabilidad, mientras que MariaDB es una bifurcación de MySQL que ofrece mejoras y características adicionales.

XAMPP agrupa estos componentes en un solo paquete, facilitando la instalación y configuración de un entorno de desarrollo web completo en la computadora del usuario. Esto permite a los desarrolladores crear, probar y depurar aplicaciones web en un entorno local antes de su implementación en servidores de producción. La interfaz gráfica de XAMPP también simplifica la administración de los servicios, como iniciar y detener Apache o MySQL.

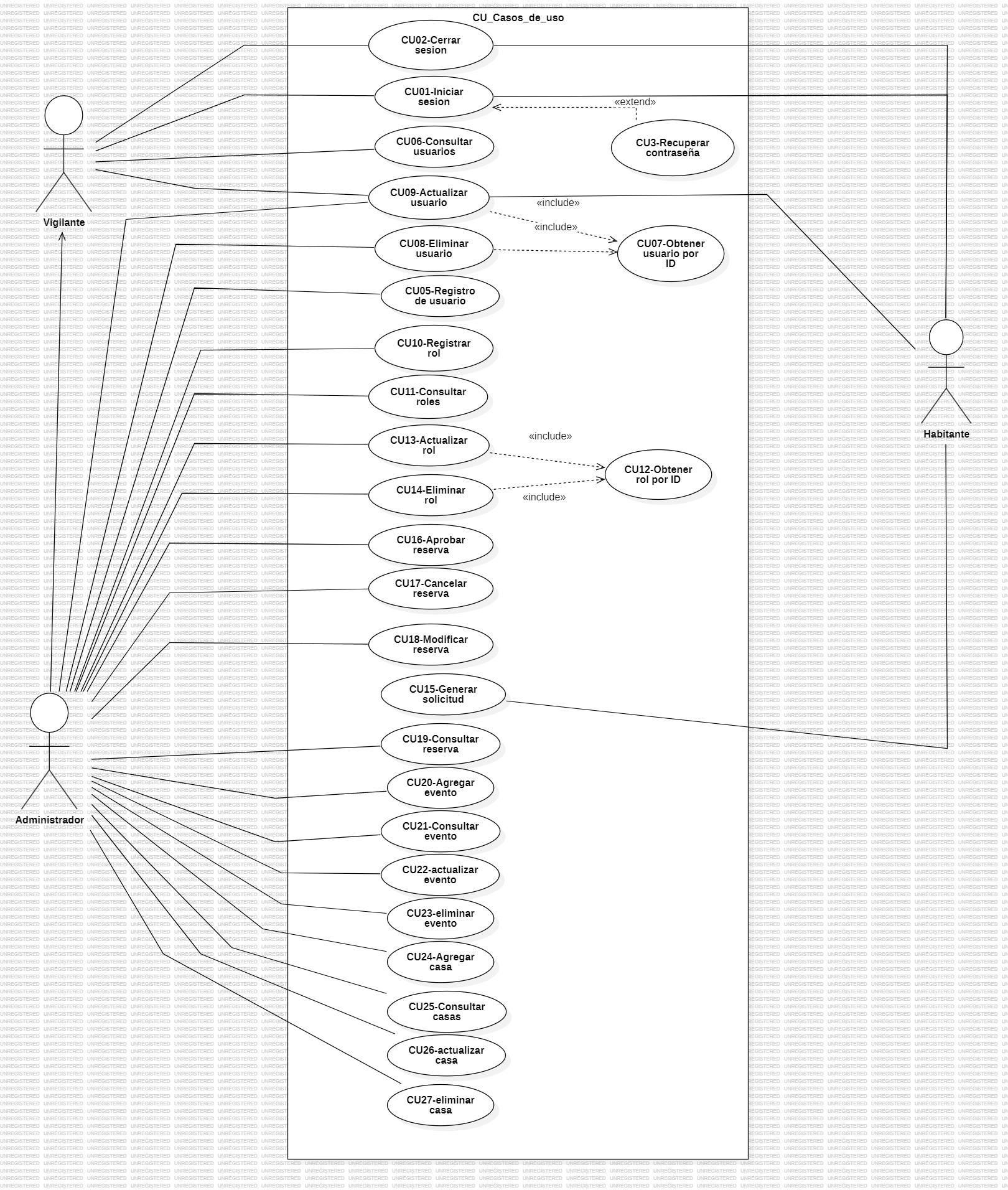
**2.5 AZURE JUNTO A MYSQL WORKBENCH**

Azure es la plataforma de servicios en la nube de Microsoft, que ofrece una amplia gama de servicios en la nube, incluyendo almacenamiento, cómputo, análisis y bases de datos. Uno de los servicios disponibles en Azure es Azure Database for MySQL, que es una base de datos relacional basada en MySQL completamente administrada y escalable.

MySQL Workbench es una herramienta gráfica para el diseño, desarrollo y administración de bases de datos MySQL. Puedes usar MySQL Workbench para conectarte y gestionar bases de datos MySQL que se ejecutan en Azure. Aquí te explico cómo se conectan ambos:

1. **CASOS DE USO**

En el desarrollo se realizaron 4 roles, uno de estos mas complementario que funcional puesto que cumple las mismas funciones que otro. Este rol es el de arrendatario ya que va a poder realizar las mismas funciones que un habitante del conjunto.



**TABLA 1 – Inicio de sesión**

|  |
| --- |
| **NOMBRE: CU1** Inicio de sesión |
| **Rol(es):** Administrador, vigilante, habitante y arrendatario |
| **Descripción:** el usuario debe dirigirse al login para poder ingresar sus credenciales y poder así efectuar sus tareas posteriores |
| **Flujo normal:**   1. El usuario abre el login 2. Ingresa su usuario 3. Ingresa su contraseña 4. Inicia sesión |
| **Pos condición:**   1. Sesión iniciada |

**TABLA 2 – Cerrar sesión**

|  |
| --- |
| **NOMBRE: CU2** cerrar sesión |
| **Rol(es):** Administrador, vigilante, habitante y arrendatario |
| **Descripción:** el usuario debe dirigirse al logout para poder salir de su sesión |
| **Flujo normal:**   1. El usuario se dirige al botón de cerrar sesión 2. Presiona cerrar sesión |
| **Pos condición:**   1. Sesión cerrada |

**TABLA 3 – Recuperar/cambiar contraseña**

|  |
| --- |
| **NOMBRE: CU3** recuperar o cambiar contraseña |
| **Rol(es):** Administrador, vigilante, habitante y arrendatario |
| **Descripción:** el usuario debe dirigirse al login para poder ubicar el botón de recuperar contra seña. |
| **Flujo normal:**   1. El usuario abre el login 2. Ubica el botón de recuperar contraseña 3. Ingresa su código de usuario junto a su nueva contraseña 4. Inicia sesión de nuevo |
| **Pos condición:**   1. Contraseña recuperada |

**TABLA 4 – Registrar rol**

|  |
| --- |
| **NOMBRE: CU4** registro de roles |
| **Rol(es):** Administrador |
| **Descripción:** el administrador debe registrar los roles esto dirigiéndose a la barra lateral con el nombre de roles |
| **Flujo normal:**   1. El administrador entra al módulo de roles 2. Llena los datos del rol a registrar (nombre del rol) 3. Guarda el rol |
| **Pos condición:**   1. rol registrado |

**TABLA 5 – consultar roles**

|  |
| --- |
| **NOMBRE: CU5** Consultar roles |
| **Rol(es):** Administrador, vigilante |
| **Descripción:** el administrador y vigilante podrán revisar todos los roles registrados |
| **Flujo normal:**   1. Se abre el módulo de roles y se ubica el campo de consultar rol 2. Se visualizan los roles 3. Se encuentra el botón de actualizar o eliminar rol |
| **Flujo alternativo:** Debe haber roles registrados |
| **Pos condición:**   1. Se visualizan los roles |

**TABLA 6 – consultar roles por id**

|  |
| --- |
| **NOMBRE: CU6** Consultar roles por id |
| **Rol(es):** Administrador, vigilante |
| **Descripción:** el administrador y vigilante podrán revisar todos los roles registrados por id para hacer una búsqueda especifica de los mismos |
| **Flujo normal:**   1. Se abre el módulo de roles y se ubica el campo de consultar roles 2. Se visualizan los roles 3. Se encuentra el botón de actualizar o eliminar roles |
| **Flujo alternativo:** Debe haber roles registrados |
| **Pos condición:**   1. Se visualizan los roles |

**TABLA 7 – eliminar roles**

|  |
| --- |
| **NOMBRE: CU7** Eliminar roles |
| **Rol(es):** Administrador |
| **Descripción:** el administrador podrá eliminar los roles a registrados |
| **Flujo normal:**   1. Se abre el módulo de roles y se ubica el campo de consultar rol 2. Se visualizan los roles 3. Se encuentra el botón de eliminar rol 4. Se oprime y se elimina el rol |
| **Flujo alternativo:** Debe haber roles registrados |
| **Pos condición:**   1. Se visualizan los roles 2. Se elimina el rol |

**TABLA 8 – actualizar roles**

|  |
| --- |
| **NOMBRE: CU8** Actualizar roles |
| **Rol(es):** Administrador |
| **Descripción:** el administrador podrá actualizar los roles a registrados |
| **Flujo normal:**   1. Se abre el módulo de roles y se ubica el campo de consultar rol 2. Se visualizan los roles 3. Se encuentra el botón de actualizar rol 4. Se oprime y se envía a la vista de actualizar el rol |
| **Flujo alternativo:** Debe haber roles registrados |
| **Pos condición:**   1. Se visualiza la información del rol 2. Se actualiza el rol |

**TABLA 9– Registrar casa**

|  |
| --- |
| **NOMBRE: CU9** registro de casas |
| **Rol(es):** Administrador |
| **Descripción:** el administrador debe registrar las casas esto dirigiéndose a la barra lateral con el nombre de casas |
| **Flujo normal:**   1. El administrador entra al módulo de casas 2. Llena los datos de la casa y registra la registra (nombre de la casa) 3. Guarda la casa |
| **Pos condición:**   1. casa registrada |

**TABLA 10– consultar casas**

|  |
| --- |
| **NOMBRE: CU10** Consultar casas |
| **Rol(es):** Administrador, vigilante |
| **Descripción:** el administrador y vigilante podrán revisar todas las casas registradas |
| **Flujo normal:**   1. Se abre el módulo de casas y se ubica el campo de consultar casa 2. Se visualizan las casas 3. Se encuentra el botón de actualizar o eliminar casa |
| **Flujo alternativo:** Debe haber casas registradas |
| **Pos condición:**   1. Se visualizan las casas |

**TABLA 11 – consultar casas por id**

|  |
| --- |
| **NOMBRE: CU10** Consultar casas por id |
| **Rol(es):** Administrador, vigilante |
| **Descripción:** el administrador y vigilante podrán revisar todas las casas registradas por id y poder consultarlas mejor |
| **Flujo normal:**   1. Se abre el módulo de casas y se ubica el campo de consultar casa 2. Se visualizan las casas 3. Se encuentra el botón de actualizar o eliminar casa |
| **Flujo alternativo:** Debe haber casas registradas |
| **Pos condición:**   1. Se visualizan las casas |

**TABLA 12 – eliminar casas**

|  |
| --- |
| **NOMBRE: CU12** Eliminar casas |
| **Rol(es):** Administrador |
| **Descripción:** el administrador podrá eliminar las casas registradas |
| **Flujo normal:**   1. Se abre el módulo de casas y se ubica el campo de consultar casa 2. Se visualizan las casas 3. Se encuentra el botón de eliminar casa 4. Se oprime y se elimina la casa |
| **Flujo alternativo:** Debe haber casas registradas |
| **Pos condición:**   1. Se visualizan las casas 2. Se elimina las casas |

**TABLA 13 – actualizar casas**

|  |
| --- |
| **NOMBRE: CU13** Actualizar casas |
| **Rol(es):** Administrador |
| **Descripción:** el administrador podrá actualizar las casas registradas |
| **Flujo normal:**   1. Se abre el módulo de casas y se ubica el campo de consultar casa 2. Se visualizan las casas 3. Se encuentra el botón de actualizar casa 4. Se oprime y se envía a la vista de actualizar la casa |
| **Flujo alternativo:** Debe haber casas registradas |
| **Pos condición:**   1. Se visualiza la información de la casa 2. Se actualiza la casa |

**TABLA 14 – registrar usuarios**

|  |
| --- |
| **NOMBRE: CU14** registro de usuarios |
| **Rol(es):** Administrador |
| **Descripción:** el administrador debe registrar los usuarios esto dirigiéndose a la barra lateral con el nombre de usuarios |
| **Flujo normal:**   1. El administrador entra al módulo de usuarios 2. Llena los datos del usuario a registrar (rol, casa, nombres, apellidos, id…) 3. Se le asigna un correo y contraseña |
| **Pos condición:**   1. Usuario registrado |

**TABLA 15 – consultar usuarios**

|  |
| --- |
| **NOMBRE: CU15** Consultar usuarios |
| **Rol(es):** Administrador, vigilante |
| **Descripción:** el administrador y vigilante podrán revisar todos los usuarios registrados |
| **Flujo normal:**   1. Se abre el modulo de usuarios y se ubica el campo de consultar usuarios 2. Se visualizan los usuarios 3. Se encuentra el botón de actualizar o eliminar usuario |
| **Flujo alternativo:** Debe haber usuarios registrados |
| **Pos condición:**   1. Se visualizan los usuarios |

**TABLA 16 – consultar usuarios por id**

|  |
| --- |
| **NOMBRE: CU16** Consultar usuarios por id |
| **Rol(es):** Administrador, vigilante |
| **Descripción:** el administrador y vigilante podrán revisar todos los usuarios registrados por id para hacer una búsqueda especifica de los mismos |
| **Flujo normal:**   1. Se abre el módulo de usuarios y se ubica el campo de consultar usuarios 2. Se visualizan los usuarios 3. Se encuentra el botón de actualizar o eliminar usuario |
| **Flujo alternativo:** Debe haber usuarios registrados |
| **Pos condición:**   1. Se visualizan los usuarios |

**TABLA 17 – eliminar usuarios**

|  |
| --- |
| **NOMBRE: CU17** Eliminar usuarios |
| **Rol(es):** Administrador |
| **Descripción:** el administrador podrá eliminar los usuarios a registrados |
| **Flujo normal:**   1. Se abre el módulo de usuarios y se ubica el campo de consultar usuarios 2. Se visualizan los usuarios 3. Se encuentra el botón de eliminar usuario 4. Se oprime y se elimina el usuario |
| **Flujo alternativo:** Debe haber usuarios registrados |
| **Pos condición:**   1. Se visualizan los usuarios 2. Se elimina el usuario |

**TABLA 18 – consultar usuarios**

|  |
| --- |
| **NOMBRE: CU18** Actualizar usuario |
| **Rol(es):** Administrador |
| **Descripción:** el administrador podrá actualizar los usuarios a registrados |
| **Flujo normal:**   1. Se abre el módulo de usuarios y se ubica el campo de consultar usuarios 2. Se visualizan los usuarios 3. Se encuentra el botón de actualizar usuario 4. Se oprime y se envía a la vista de actualizar el usuario |
| **Flujo alternativo:** Debe haber usuarios registrados |
| **Pos condición:**   1. Se visualiza la información del usuario 2. Se actualiza el usuario |