|  |
| --- |
| **SPRINT 1: Definiendo el Proyecto - Planificación Scrum** |

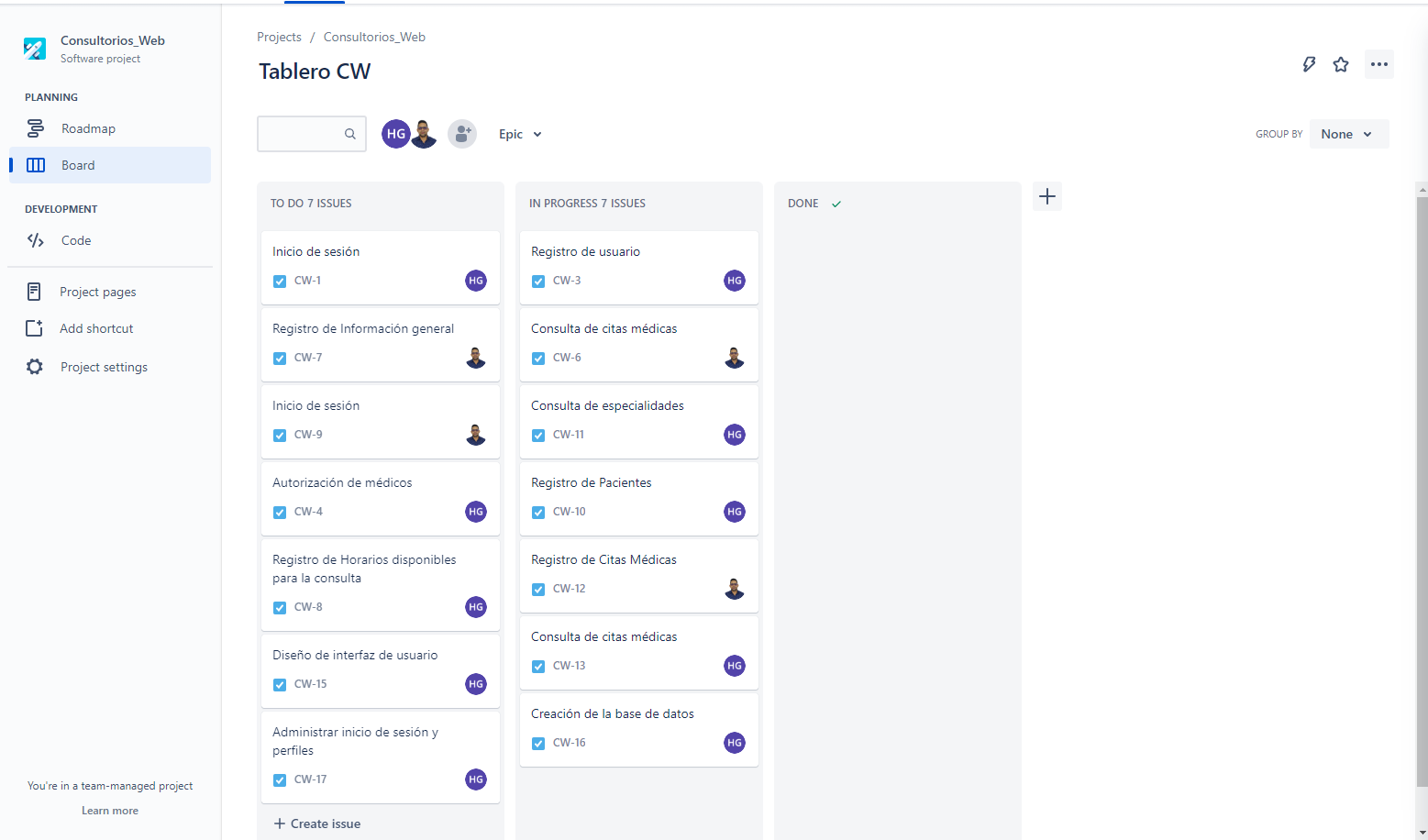
|  |  |
| --- | --- |
| **Identificación Proyecto** | |
| Nombre Proyecto: | **Medical Office** |
| Número Equipo: | 1 |
| **Integrantes del equipo** | |
| **Rol**  (Líder-Desarrollador – Cliente) | **Nombre** |
| Líder – Analista - Desarrollador | Hernan Dario Gomez |
| **Descripción Proyecto (Mundo del Proyecto)** | |
| Los sistemas informáticos en la actualidad brindan un amplio soporte en el ámbito empresarial, debido a que contribuyen en la optimización de procesos y recursos en los distintivos niveles jerárquicos de las organizaciones.  El proyecto a desarrollar un aplicativo web, que tiene como finalidad coordinar y mejorar los métodos de ingreso de información; ya que el software automatiza los diferentes procesos enfocados en los consultorios médicos de medicina general como el acceso a pacientes, asignación de citas, formulas médicas.  La aplicación de este proyecto busca tener un impacto social y económico, brindándole a los pacientes calidad e integridad en la prestación de los servicios de salud, optimizando los procesos administrativos, asistenciales y la implementación de las políticas del manejo del aseguramiento de la información, la gestión documental, así como la custodia y guarda de la historia clínica de los pacientes. | |
| **Objetivo General** | |
| Diseñar e implementar un sistema de información web, para optimizar los procesos administrativos y asistenciales, control de agendamiento de citas médicas, acceso a pacientes, y medicamentos formulados en consultorios médicos. | |
| **Objetivos Específicos** | |
| * Realizar el análisis de requerimientos para establecer los alcances y limitaciones del sistema de información. * Seleccionar un modelo de ingeniería de software que este acorde a los estándares de la arquitectura de un sistema de información. * Analizar las reglas del negocio requeridas para el desarrollo de servicios en el área de la salud, particularmente lo relacionado con citas, registro de pacientes y medicamentos formulados. * Diseñar e implementar la base de datos. * Generar el diseño del modelo del sistema de información integrado a la arquitectura del software. * Realizar una fase de pruebas e implementación con su respectiva documentación, para validar y verificar el correcto funcionamiento del sistema de información. * Adoptar el cumplimiento de los lineamientos y restricciones de la política de la entidad en cuanto a los componentes tecnológicos del sistema (lenguaje de programación, estilo arquitectural y almacenamiento de la información) asegurando la implementación de buenas prácticas al proceso de desarrollo de la atención de consultorios de medicina general. | |

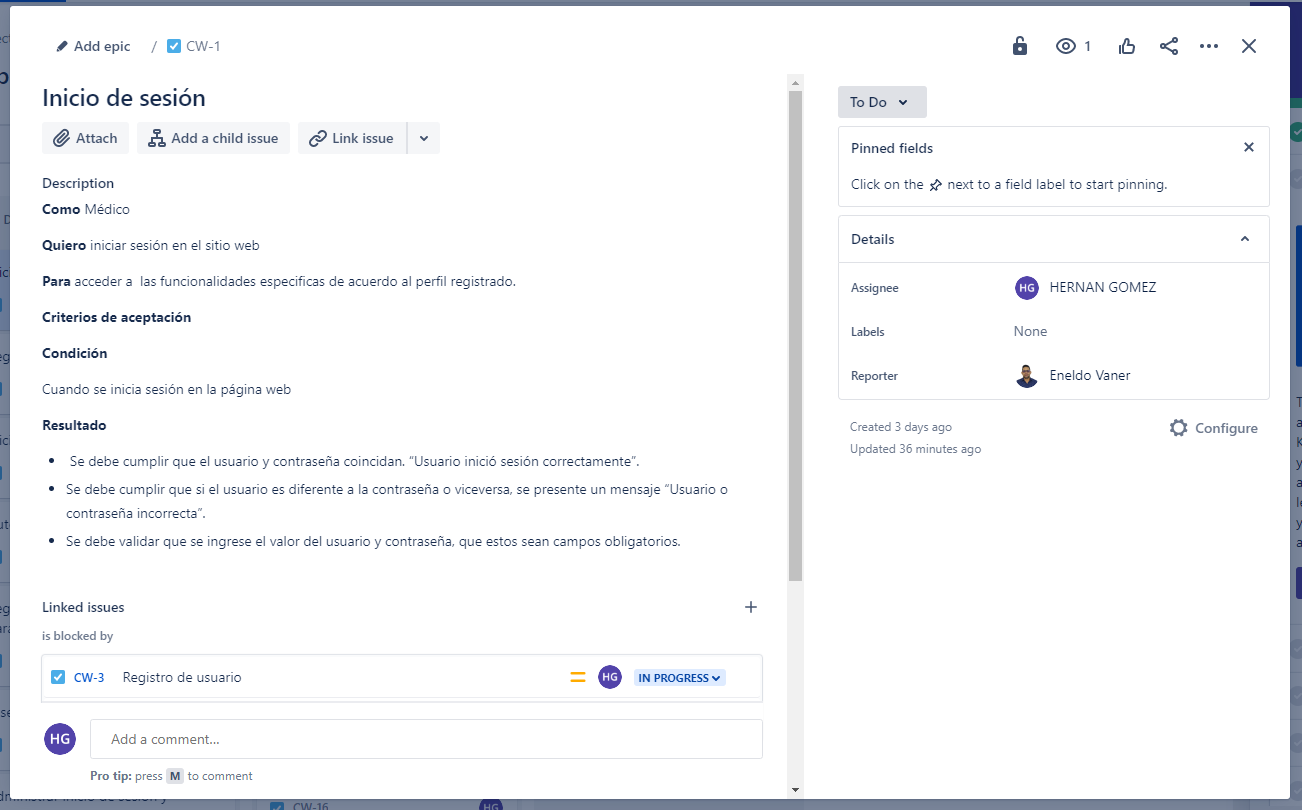
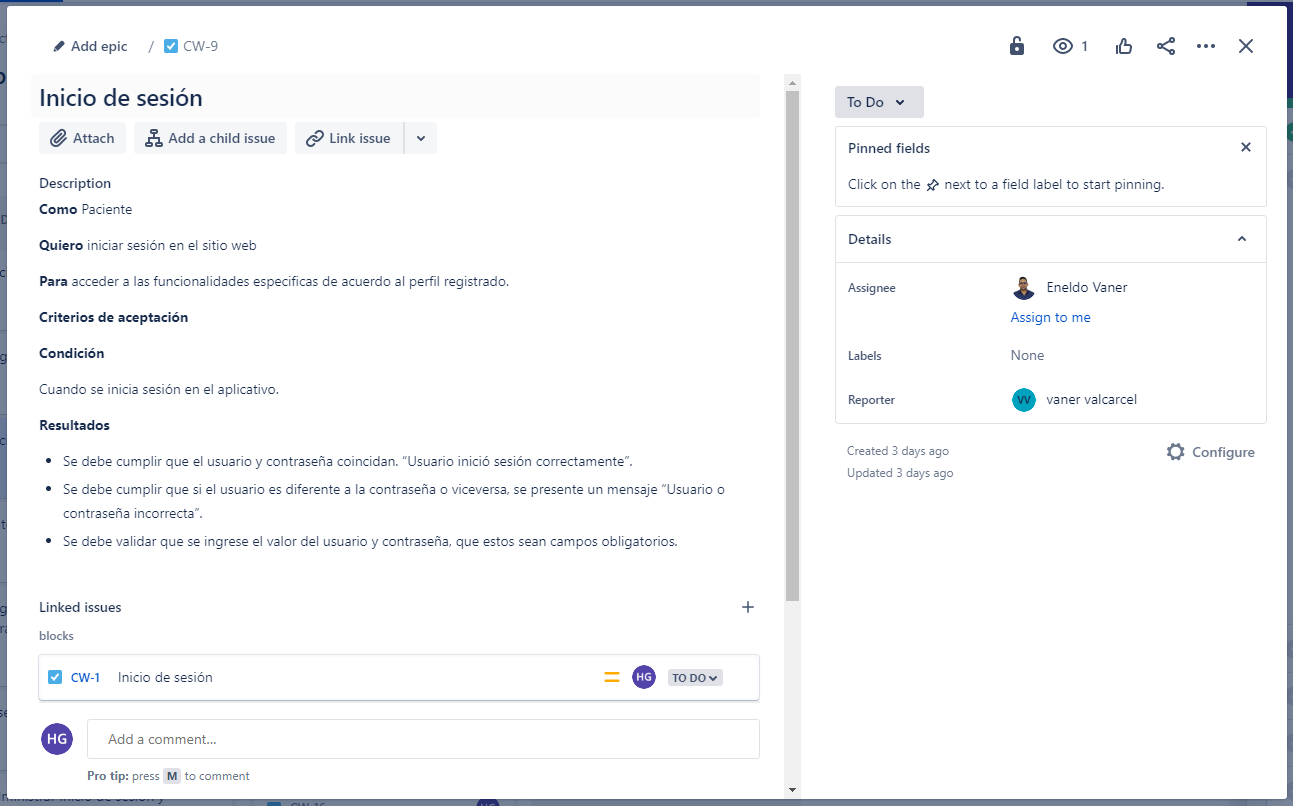
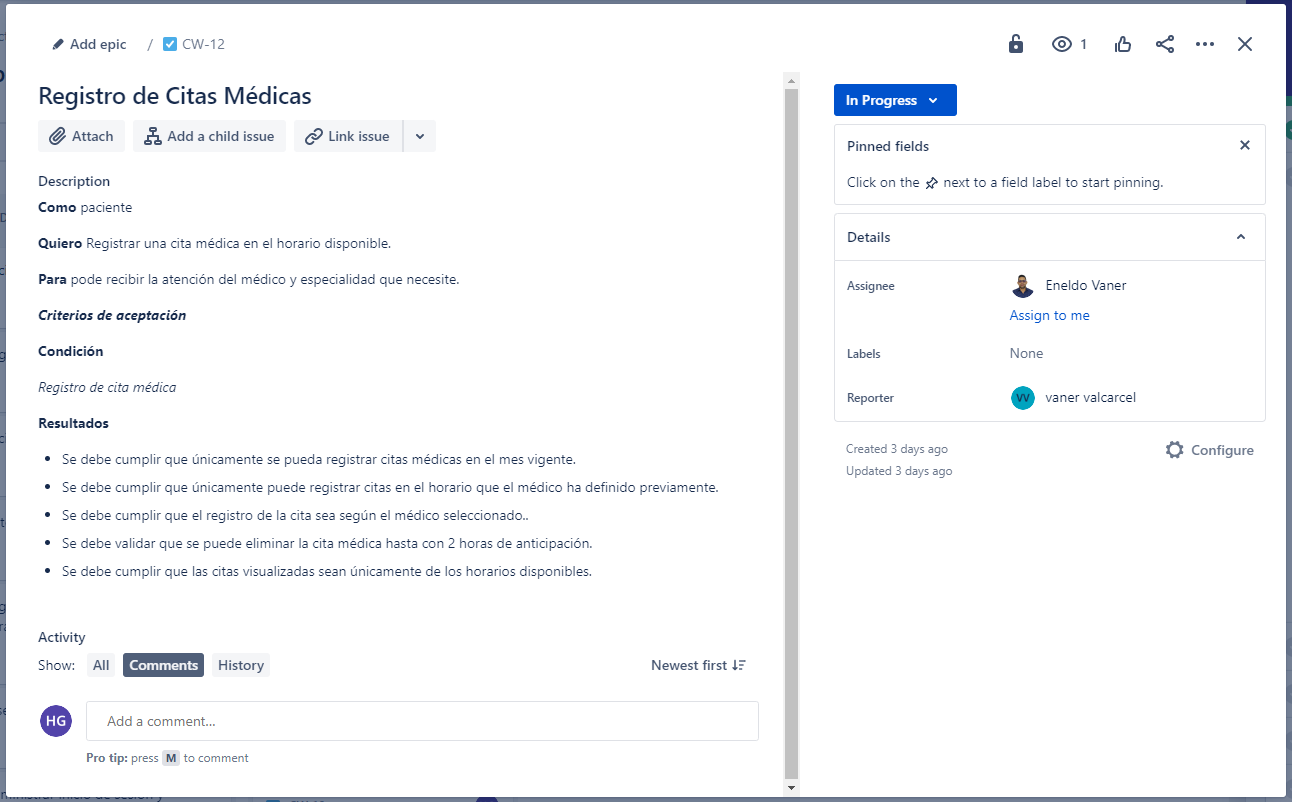
|  |
| --- |
| **Requerimientos Funcionales** |
| |  |  | | --- | --- | | CODIGO DEL REQUERIMIENTO | RF01 | | Nombre: | Crear usuario | | Entrada:   1. Usuario 2. Contraseña 3. Nombre 4. Dirección | Salida:  El usuario recibe una notificación por pantalla  informando de que su usuario ha sido creado  satisfactoriamente o en su defecto de rechazo,  indicando la razón por la que la operación no fue  completada | | Precondición | | | * El campo usuario debe contener solo caracteres alfanuméricos entre 8 y 11 números * El campo contraseña debe ser de carácter String y debe estar compuesta por números, letras, mayúsculas, minúsculas, caracteres especiales y tener una longitud de 8 hasta 15 caracteres * El campo nombre debe ser de carácter String y no puede contener caracteres alfanuméricos * El campo dirección es de carácter String. | | | Manejo de situaciones anormales | | | 1. Al registrar un usuario ya encontrado en la base de datos del sistema, se arroja una alerta de notificación “El usuario ya existe en la base de datos”. 2. La contraseña debe estar compuesta por números, letras, mayúsculas, minúsculas, caracteres especiales y tener una longitud de 8 hasta 15 caracteres, si es incorrecto, se le notificará al usuario “Su contraseña es demasiado corta” no cumple con lo solicitado. 3. El usuario al registrar un carácter numérico en el campo de nombre, se le mostrará una notificación diciendo “Solo se aceptan letras en el nombre” 4. Si un usuario no rellena todos los campos se le mostrará una alerta notificando “Todos los campos deben ser diligenciados”. | | | Criterios de aceptación:   1. Si el usuario completo en su totalidad los campos de registro 2. El usuario coloco un usuario no existente en la base de datos 3. El usuario registro los campos con los datos correctos 4. El usuario al terminar selecciona el botón “Crear usuario” donde se validará la información ingresada, y posteriormente la creación exitosa de la cuenta | | | Prioridad: Alta | | | CODIGO DEL REQUERIMIENTO | **RF02** | | Nombre: | Consultar usuario | | Entrada:  Seleccionar el botón “MI cuenta” | El usuario podrá consultar los datos registrados en  la última actualización de su cuenta. | | Precondición | | | * El usuario debe encontrarse autenticado en la aplicación   Descripción:  El usuario podrá consultar los datos de registro asociados a su cuenta en la última actualización que haya realizado | | | Manejo de situaciones anormales | | | 1. Al registrar un usuario ya encontrado en la base de datos del sistema, se arroja una alerta de notificación “El usuario ya existe en la base de datos”. 2. La contraseña debe estar compuesta por números, letras, mayúsculas, minúsculas, caracteres especiales y tener una longitud de 8 hasta 15 caracteres, si es incorrecto, se le notificará al usuario “Su contraseña es demasiado corta” no cumple con lo solicitado. 3. El usuario al registrar un carácter numérico en el campo de nombre, se le mostrará una notificación diciendo “Solo se aceptan letras en el nombre” 4. Si un usuario no rellena todos los campos se le mostrará una alerta notificando “Todos los campos deben ser diligenciados”. | | | Criterios de aceptación:   1. El usuario debe encontrarse autenticado en la aplicación para poder consultar los datos relacionados en su cuenta. | | | Prioridad: Alta | | | CODIGO DEL REQUERIMIENTO | **RF03** | | Nombre: | Modificar usuario | | Entrada:   1. Usuario 2. Contraseña 3. Nombre 4. Dirección | **Salida:**  Al cumplir con las condiciones requeridas en los diferentes campos, se mostrar una notificación informando al usuario que le proceso de actualización de datos ha sido exitoso. | | Precondición | | | * El campo usuario debe contener solo caracteres alfanuméricos entre 8 y 11 números * El campo contraseña debe ser de carácter String y debe estar compuesta por números, letras, mayúsculas, minúsculas, caracteres especiales y tener una longitud de 8 hasta 15 caracteres * El campo nombre debe ser de carácter String y no puede contener caracteres alfanuméricos * El campo dirección es de carácter String.   Descripción:  El usuario puede realizar la modificación de los datos ingresados al momento de la creación de la cuenta | | | Manejo de situaciones anormales | | | 1. Al registrar un usuario ya encontrado en la base de datos del sistema, se arroja una alerta de notificación “El usuario ya existe en la base de datos”. 2. La contraseña debe estar compuesta por números, letras, mayúsculas, minúsculas, caracteres especiales y tener una longitud de 8 hasta 15 caracteres, si es incorrecto, se le notificará al usuario “Su contraseña es demasiado corta” no cumple con lo solicitado. 3. El usuario al registrar un carácter numérico en el campo de nombre, se le mostrará una notificación diciendo “Solo se aceptan letras en el nombre” 4. Si un usuario no rellena todos los campos se le mostrará una alerta notificando “Todos los campos deben ser diligenciados”. | | | Criterios de aceptación:   1. Si el usuario completo en su totalidad los campos de registro 2. El usuario coloco un usuario no existente en la base de datos 3. El usuario registro los campos con los datos correctos 4. El usuario al terminar selecciona el botón “Crear usuario” donde se validará la información ingresada, y posteriormente la creación exitosa de la cuenta | | | Prioridad: Alta | | | CODIGO DEL REQUERIMIENTO | **RF04** | | Nombre: | Eliminar usuario | | Entrada:   1. Usuario 2. Contraseña 3. Nombre 4. Dirección | Salida:  Se almacena la acción en un log | | Precondición | | | * La cuenta por eliminar debe existir en la base de datos. Si el usuario que realiza esta acción es un administrador, podrá eliminar cualquier otro tipo de usuario, si es Gestor únicamente podrá eliminar médicos, esta opción estará deshabilitada para los médicos.   Descripción:   * El usuario busca la(s) cuenta usando su usuario selecciona el botón de eliminar.   Postcondición: Se eliminan los datos de la base de datos y se guarda en un log. | | | Manejo de situaciones anormales | | | 1. Si un rol de nivel inferior intenta eliminar uno de nivel superior debe aparecer un mensaje de error; con un mensaje de texto “No se puede eliminar un usuario con un rol superior”. 2. Si un gestor intenta eliminar a otro del mismo rol, se debe informar al administrador vía notificaciones y debe aparecer un mensaje de error al usuario gestor “No se puede eliminar un gestor”. | | | Criterios de aceptación:   1. El rol con el que se intenta eliminar siempre tiene que ser superior al eliminado, una vez se cumpla con esta condición se debe aceptar la acción. | | | Prioridad: Alta | |  |  |  | | --- | --- | | CODIGO DEL REQUERIMIENTO | RF05 | | Nombre: | Iniciar sesión | | Entrada:   1. Usuario 2. Contraseña | Salida:  La validación exitosa de las credenciales de usuario, lo redirige a la ventana principal de la  Aplicación. | | Precondición:   * El usuario ingresado debe encontrarse registrado en la base de datos. * La contraseña ingresada debe estar asociada al usuario registrado en la base de datos.   Descripción:  El usuario realizará el inicio de sesión correspondiente, donde permitirá ingresar a la función correspondiente.  Postcondición   * Permite el acceso al aplicativo web, dependiendo del tipo de Rol a que esté asociado. | | | Manejo de situaciones Anormales |  | | 1. Si el usuario ingresado no se encuentra en la base de datos se mostrará un mensaje de notificación: “Los datos ingresados son incorrectos” 2. Si el usuario ingresa una contraseña errónea, se le notifica por medio de un mensaje “Los datos ingresados no son correctos”. | | | Criterios de Aceptación | | | 1. El usuario ingresa un usuario registrado anteriormente en la base de datos 2. El usuario ingresa una contraseña relacionada con el usuario ingresado anteriormente | |  |  |  | | --- | --- | | CODIGO DEL REQUERIMIENTO | RF06 | | Nombre: | Cerrar sesión | | Entrada:   1. Solicitud cierre de sesión | Salida:  El usuario será direccionado a la página principal. | | Precondición:   * El usuario debe encontrarse logueado en el aplicativo.     Descripción:  El usuario podrá realizar el cierre de su sesión cuando así lo requiera, permitiéndole consultar la información de la aplicación sin haber validado sus credenciales de usuario en la aplicación.  Postcondición   * Permite el acceso al aplicativo web, dependiendo del tipo de Rol a que esté asociado. | | | Criterios de Aceptación | | | 1. El usuario debe encontrarse autenticado en la aplicación con credenciales registradas en la base de datos. | |  |  |  | | --- | --- | | CODIGO DEL REQUERIMIENTO | RF07 | | Nombre: | **Crear agenda** | | Entrada:   1. Nombre médico 2. Especialidad 3. Fecha de inicio 4. Fecha fin 5. Hora ingreso 6. Hora salida 7. Días disponibles | Salida:   1. Una vez registrados los datos de entrada el sistema debe dar un mensaje de éxito. “Agenda creada con éxito”. 2. Agenda registrada en la base de datos. | | Precondición:   * El usuario debe encontrarse logueado en el aplicativo * El profesional médico debe estar ya creado en la base de datos * La especialidad debe estar creada en la base de datos * Los roles autorizados para la creación de agendas médicas es el administrador   Descripción:  El usuario podrá realizar la creación de las agendas médicas de acuerdo con las horas contratadas de los profesionales y a la disponibilidad de consultorios, una vez se haya registrado los datos de entrada el sistema debe validar los horarios disponibles de los especialistas, y sugerir la creación de la agenda  Postcondición   * La agenda se creará en la base de datos del sistema. | | | Manejo de situaciones Anormales |  | | 1. Si el usuario no registra la información completa el sistema debe sacar un mensaje de advertencia indicando los campos que están sin registrar “campos sin registrar”. | | | Criterios de Aceptación | | | 1. El usuario debe encontrarse autenticado en la aplicación con credenciales registradas en la base de datos 2. La especialidad debe estar creada en la base de datos. 3. El profesional médico debe estar creado en la base de datos 4. El consultorio debe estar disponible | |  |  |  | | --- | --- | | CODIGO DEL REQUERIMIENTO | RF08 | | Nombre: | Consultar agenda | | Entrada:   1. Nombre 2. Médico | Salida:  Una vez registrados los datos de entrada el sistema debe mostrar la agenda consultada | | Precondición:   * El usuario debe encontrarse logueado en el aplicativo. * La agenda debe estar creada en la base de datos. * Los roles autorizados para la consulta de agendas médicas son el administrador, y el médico.   Descripción:  El usuario realizará el inicio de sesión correspondiente, donde permitirá ingresar a la función correspondiente.  Postcondición   * El usuario podrá realizar la consulta de las agendas médicas ya creadas en la base de datos. | | | Manejo de situaciones Anormales |  | | 1. Si la agenda no ha sido creada en la base de datos, debe aparecer un mensaje indicando “agenda no creada”. | | | Criterios de Aceptación | | | 1. El usuario debe encontrarse autenticado en la aplicación con credenciales registradas en la base de datos. 2. La agenda debe estar creada en la base de datos. | |  |  |  | | --- | --- | | CODIGO DEL REQUERIMIENTO | RF09 | | Nombre: | **Cerrar agenda** | | Entrada:   1. Nombre médico 2. Fecha inicio 3. Fecha final 4. Hora inicio 5. Hora final 6. Tipo novedad | Salida:   1. Una vez registrados los datos de entrada el sistema debe mostrar la agenda a cerrar. 2. Seleccionado la fecha y hora para cerrar el sistema debe informar el éxito de la transacción. | | Precondición:   * El usuario debe encontrarse logueado en el aplicativo * La agenda debe estar creada en la base de datos * Los roles autorizados para el cierre de agendas médicas son el administrador y el médico.   Descripción:   * El usuario podrá realizar el cierre de agendas médicas ya creadas en la base de datos. En el caso que el profesional registre un cierre por novedades como permisos el sistema validará los horarios disponibles de los consultorios para reponer el tiempo. * Si el cierre reportado es por novedades como calamidad o incapacidad, el profesional médico debe registrar los documentos que soporten la novedad.   Postcondición   * En la base de datos se debe actualizar la liberación de consultorio de la agenda cerrada, y las horas pendientes a reponer del profesional en el caso que sea un permiso, para otras novedades como incapacidades o calamidades el profesional médico debe soportar la novedad cargando la imagen en el sistema de información. | | | Manejo de situaciones Anormales |  | | 1. Si la agenda no ha sido creada en la base de datos, debe aparecer un mensaje indicando “agenda no creada” | | | Criterios de Aceptación | | | 1. El usuario debe encontrarse autenticado en la aplicación con credenciales registradas en la base de datos. 2. La agenda debe estar creada en la base de datos. | |   **Autenticación:**   * Permite el ingreso de los usuarios, con su respectivo usuario y contraseña, además de elegir su respectivo perfil de acceso.   **Módulos de usuarios:**   * Permite la creación de nuevos usuarios en el sistema, y asignarle sus permisos de acceso, permite la modificación del perfil y del usuario.   **Ingreso de pacientes:**   * El usuario tiene acceso de registrar los datos de los pacientes y a la vez agendar cita médica en la aplicación web.   **Agendamiento de citas:**   * El usuario paciente podrá agendar cita médica y escoger su médico de preferencia, directamente a la aplicación web.   **Reportes:**   * Permite generar reporte de citas médicas entre fechas estipuladas * Generar el listado de pacientes con su respectivo médico. * Generar el listado de pacientes según el médico.     **Notificación de agendamiento de cita:**   * Permite al usuario reservar una cita médica la cual será confirmada mediante correo electronico. |
| **Requerimientos NO Funcionales** |
| Los requerimientos no funcionales, en los cuales se evalúan las operaciones que dan un enfoque importante en el despliegue del sistema de información.  **Interfaces**   * Fácil de interactuar para el usuario * Fácil manejo en el control de procesos * Satisfacer la necesidad de los consultorios médicos.   **Eficiencia**   * Los datos modificados en la base de datos deben ser actualizados para todos los usuarios que acceden en menos de 5 segundos.   **Usabilidad**   * El sistema debe proporcionar mensajes de error que sean informativos y orientados a usuario final. * El sistema debe poseer interfaces gráficas bien formadas. * El sistema debe contar con manuales de usuario estructurados adecuadamente.   **Autorización**   * El sistema de información debe permitir el ingreso y tener los permisos de acuerdo con el rol definido.   **Disponibilidad**   * El sistema deberá estar disponible las 24 horas de los 365 días del año, sin embargo, puede admitirse un máximo de 10 horas al año que el sistema esté fuera de línea ya sea por caída de servidores o por reparaciones.   **Capacidad de peticiones**  El sistema en producción, en un momento de estrés, deberá soportar hasta 1000 peticiones concurrentes, sin llegar a afectar los tiempos de respuesta a los usuarios de la aplicación. |

|  |
| --- |
| **Planificación SCRUM - JIRA** |

Como evidencia de la planificación del proyecto con la metodología ágil SCRUM, utilizando el software JIRA, se debe presentar capturas de pantalla donde se visualicen aspectos:

* Creación del proyecto.
* Integrantes del equipo invitados en JIRA
* Épicas e historias de usuario (Por lo menos una épica) (Hoja de Ruta)
* Creación y lanzamiento de un Sprint (Backlog y Tablero)



Graphical user interface, application, Teams

Description automatically generated

Tenemos el proyecto en JIRA con historias de Usuarios, y cada miembro del grupo tiene una tarea por hacer.

Ejemplo de una historia de usuario

Principio del formulario

* **Como** Usuario Administrador
* **Quiero** que el sistema me muestre el listado de los médicos.
* **Para** Aprobar los Médicos que cuenten con un número de registro válido

**Criterios de Aceptación**

Cuando consulte los médicos registrados:

1. Se debe cumplir que en el listado de médicos únicamente se muestren los que se encuentren en estado pendiente de autorización (activos).

Cuando autorice el registro:

1. Se debe cumplir que se autoricen únicamente los médicos que el administrador seleccione y guarde
2. Se debe cumplir que se guarde un registro de la fecha, hora y el usuario que lo autorizo.

Final del formulario

**Actividad**

Mostrar:

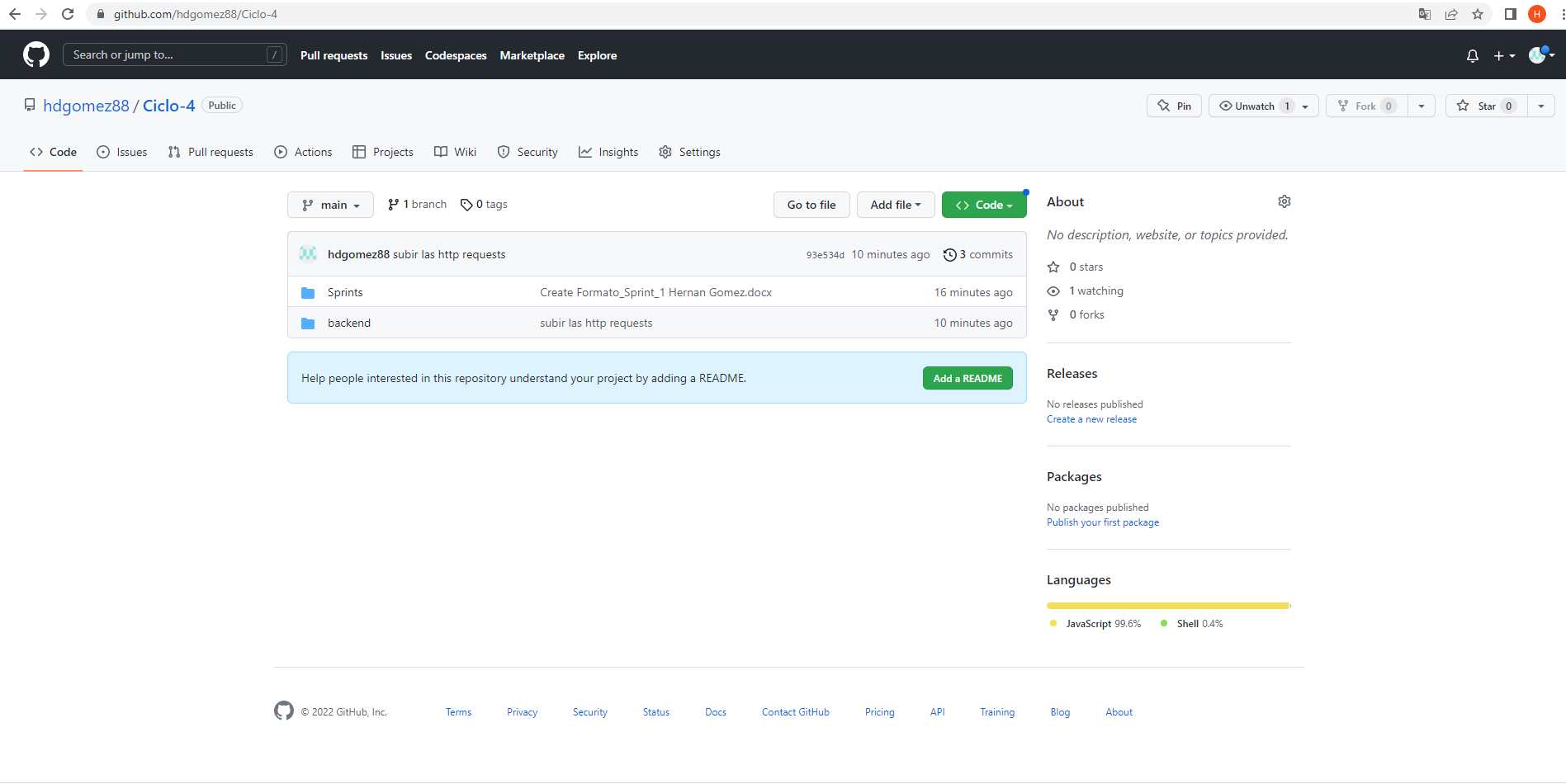
Comentarios

Más recientes primero

|  |
| --- |
| **Repositorio de Código GitLab o GitHub** |

Como evidencia del repositorio de código, creado con GitLab o GitHub, además de la URL del repositorio, se debe presentar capturas de pantalla donde se visualicen aspectos:

* Creación del proyecto del repositorio.
* Integrantes del equipo invitados.
* Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versiones)

Graphical user interface, text, application, email

Description automatically generated

|  |
| --- |
| **Evidencias de las Reuniones de Equipo** |

Como evidencia de las reuniones que efectúa el equipo del proyecto, presentar capturas de pantalla de las reuniones efectuadas y si lo consideran pertinente algunas actas de las reuniones.

He venido trabajando el proyecto en colaboración con un compañero que no esta en el grupo G27, debido a que al principio no había suficientes asistentes a la clase y nadie se inscribió en primera instancia en el listado proporcionado por el docente.