|  |
| --- |
| **SPRINT 1: Definiendo el Proyecto - Planificación Scrum – Repositorio Código** |

|  |  |
| --- | --- |
| **Identificación Proyecto** | |
| Nombre Proyecto: | Implementar un sistema que permita realizar citas médicas de un usuario previamente registrado. |
| Número Equipo: | 5 |
| **Integrantes del equipo** | |
| Rol  (Líder-Desarrollador – Cliente) | Nombre |
| Lider, desarr, cl | William Camilo Lopez Oviedo |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **Descripción Proyecto (Mundo del Proyecto)** | |
| Se requiere realizar una pagina web que permita a un administrador o a un usuario registrarse en la misma, para sacar una cita medica en el caso de un usuario; todo esto con el fin de evitar largas filas y/o colapsar el sistema de solicitud físico; y para el caso de un administrador, pueda tener control de los usuarios que estén registrados en el sistema y también del personal medico, así como realizar los respectivos CRUD en caso de necesitarlo. La pagina principal compone de un formulario de los campos necesarios para sacar una cita medica, tales como cedula paciente, tipo de cita, y el id del medico que lo puede atender; confirmando asi en una vista la visualización exitosa de la solicitud de la cita. Cada sesión de usuario debe disponer de un token que debe ser validado en cada cargue de pagina.  Tecnologias: para la realización de la base de datos, se usara Mongo DB y MySQL. Para el backend se utilizara Springboot, frontend se utilizara Angular y para estilo de maquetación se utilizara Boostrap . Se debe hacer uso de 1 repositorio (Github) con 3 ramas, donde una rama sea para guardar backend, la otra para frontend, y por ultimo los archivos del proyecto (MER, diseño de plantillas, pantallazos, reuniones, etc) | |
| **Objetivo General** | |
| * Desarrollar un sistema de información que permita sacar citas medicas a un usuario de una EPS. | |
| **Objetivos Específicos** | |
| * Documentar la información de los requerimientos funcionales y no funcionales del sistema de informacion. * Diseñar el modelo entidad-relacion así como los diagramas de casos de uso. * Implementar el sistema de información que permita sacar citas medicas. * Aplicar las pruebas unitarias a cada componente del sistema de información. * Desplegar el sistema de información en la nube | |

|  |
| --- |
| **Requerimientos Funcionales** |

**Gestionar Usuarios:**

* El sistema debe tener un formulario que contenga 8 campos (tipo de documento, numero, nombre completo, fecha de nacimiento, dirección, teléfono, correo electrónico, eps)
* El sistema debe ser responsive (Adaptara dispositivos móviles, tablets, PC)
* El sistema debe conectarse a la base de datos para que pueda guardar la información que se llena y se envía por el formulario.
* El sistema debe tener un botón para el envio de información del formulario
* El sistema debe contener un diseño manejando UI cómoda y amigable al usuario.

**Admins:**

* El sistema debe tener un formulario que contenga 5 campos (tipo de documento, numero, nombre completo, teléfono, correo electrónico)
* El sistema debe ser responsive
* El sistema debe conectarse a la base de datos para que pueda guardar la información que se llena y se envía por el formulario.
* El sistema debe tener un botón para el envio de información del formulario
* El sistema debe permitir realizar los CRUD de usuarios y de los médicos

**Medicos:**

* El sistema debe tener un formulario que contenga 8 campos (tipo de documento, numero, nombre completo, teléfono, correo electrónico, especialidad)
* El sistema debe ser responsive
* El sistema debe conectarse a la base de datos para que pueda guardar la información que se llena y se envía por el formulario.
* El sistema debe tener un botón para el envio de información del formulario

|  |
| --- |
| **Requerimientos NO Funcionales** |

**Gestionar Usuarios:**

* El sistema debe validar que los campos del formulario se escriban de forma correcta, validando la cedula, email, nombre etc.
* El encargado de la funcionalidad debe validar que el sistema de gestión de usuarios se cargue de forma correcta en cada uno de los dispositivos
* El sistema debe generar mensajes de error por las diferentes validaciones que no esten correctas
* El sistema debe validar que los botones funcionen de manera correcta y si no mostrar ventanas con mensajes de error informativos.

**Admins:**

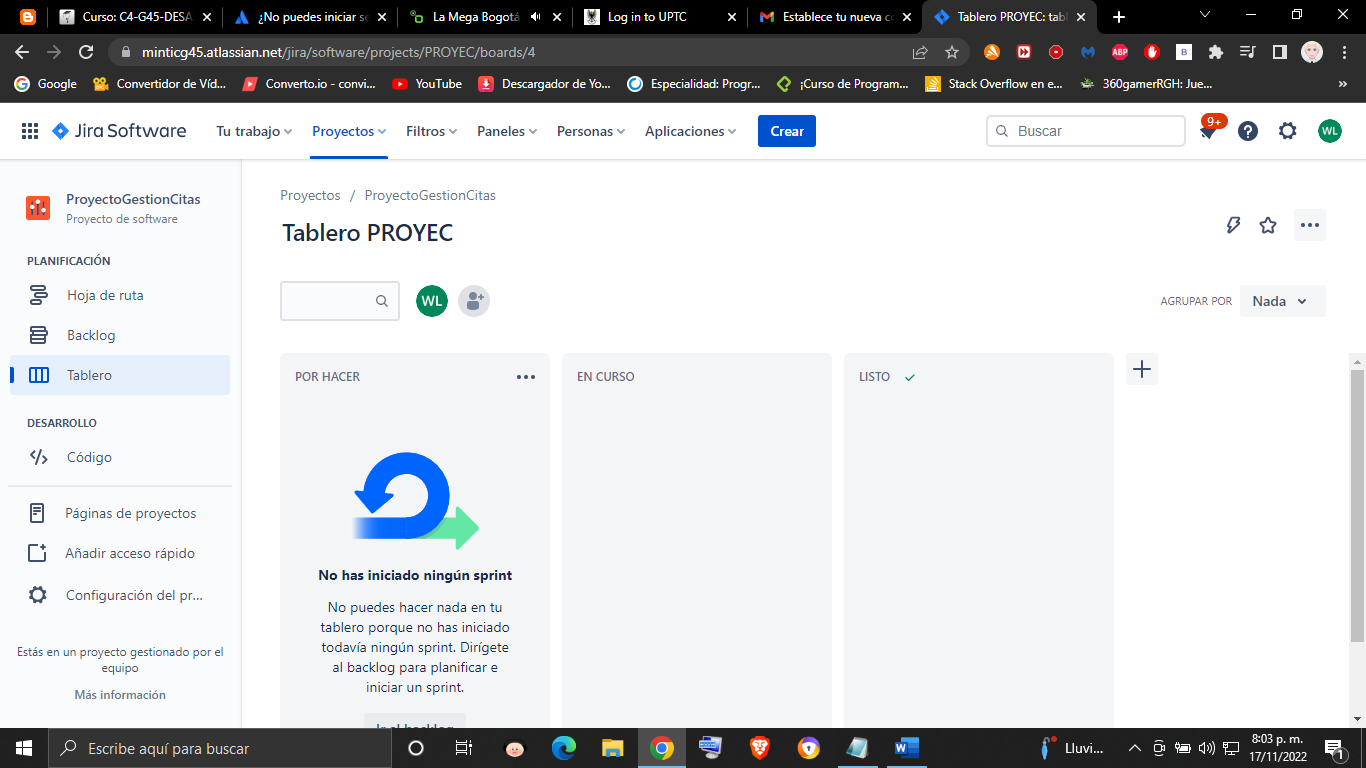
* El sistema debe validar que los campos del formulario se escriban de forma correcta, validando la cedula, email, nombre etc.
* El encargado de la funcionalidad debe validar que el sistema de gestión de usuarios se cargue de forma correcta en cada uno de los dispositivos
* El sistema debe generar mensajes de error por las diferentes validaciones que no esten correctas.
* El sistema debe validar que los botones funcionen de manera correcta y si no mostrar mensajes de error informativos.

**Medicos:**

* El sistema debe validar que los campos del formulario se escriban de forma correcta, validando la cedula, email, nombre etc.
* El encargado de la funcionalidad debe validar que el sistema de gestión de médicos se cargue de forma correcta en cada uno de los dispositivos
* El sistema debe generar mensajes de error por las diferentes validaciones que no esten correctas.
* El sistema debe validar que los botones funcionen de manera correcta y si no mostrar mensajes de error informativos.

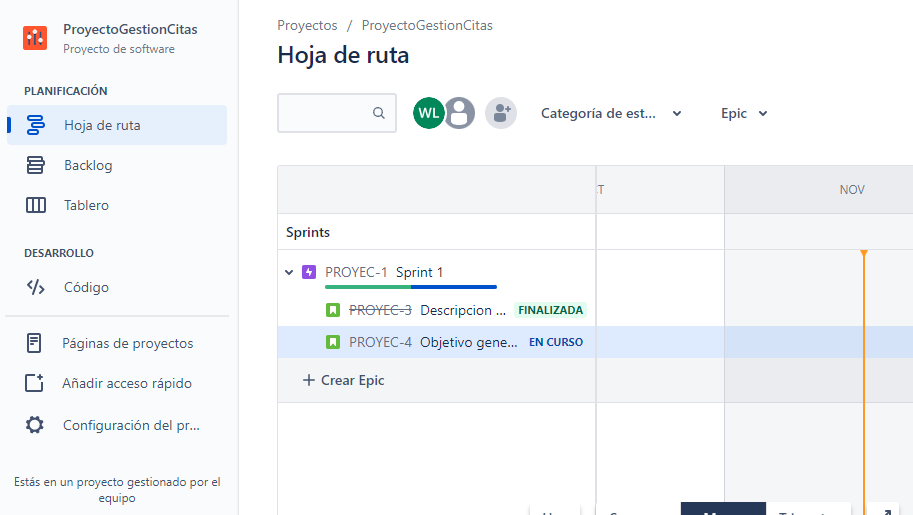
|  |
| --- |
| **Planificación SCRUM - JIRA** |

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:

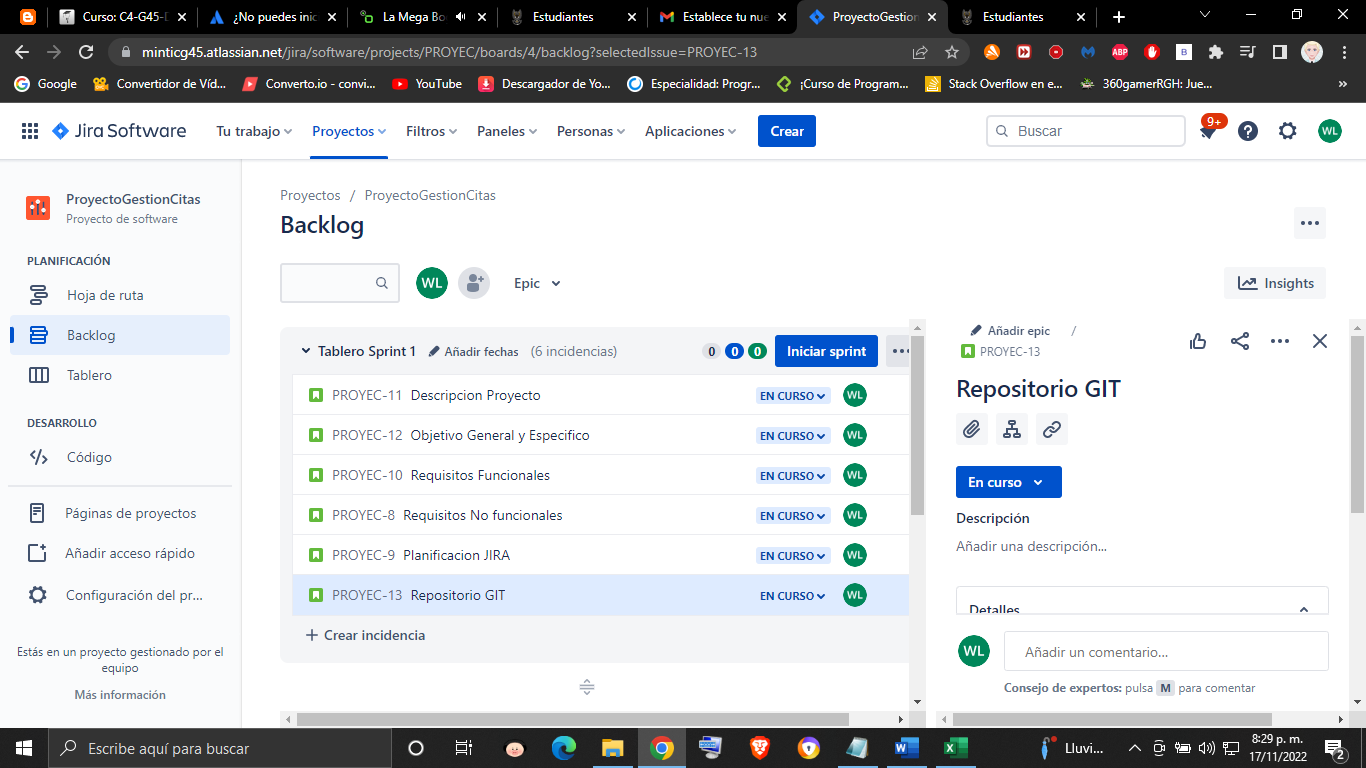
* Creación del proyecto.
* 
* Integrantes del equipo invitados en JIRA

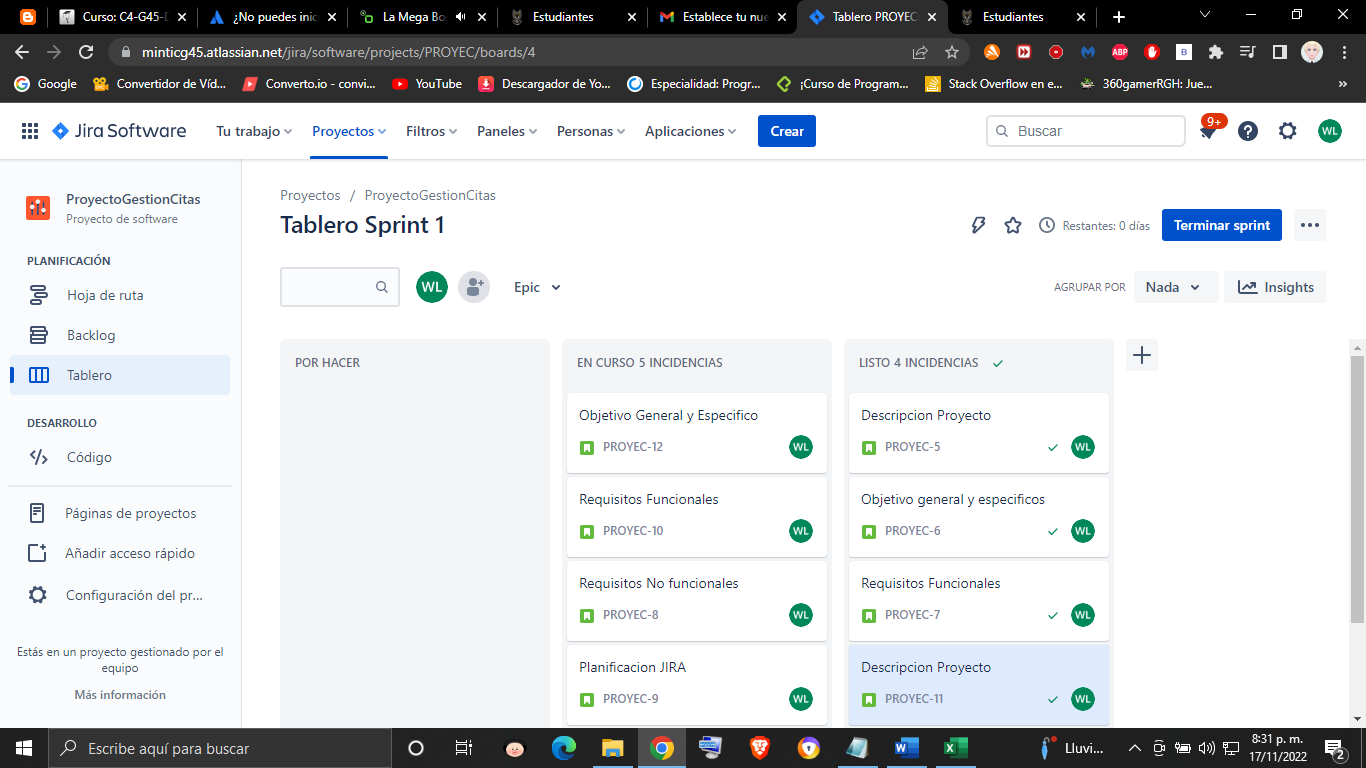


* Épicas e historias de usuario (Por lo menos una épica) (Hoja de Ruta)



* Creación y lanzamiento de un Sprint (Backlog y Tablero)





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

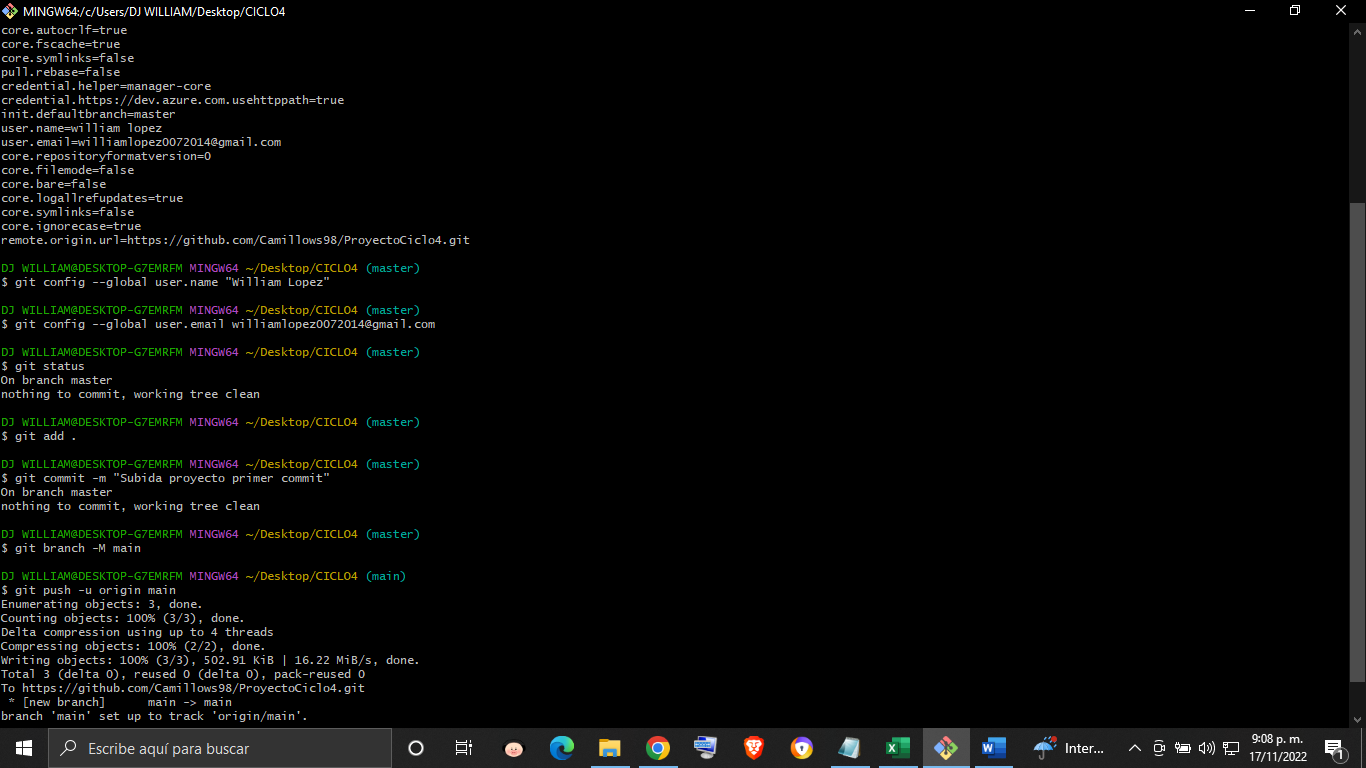
Como evidencia del repositorio de código, creado con GitLab o GitHub, además de la URL del repositorio, se debe presentar capturas de pantalla donde se visualicen aspectos:

* Creación del proyecto del repositorio.



* Integrantes del equipo invitados.

**ESTOY SOLO**

* Evidencia de la realización de alguna actualización (commit), donde se visualice la actualización y el historial de actualizaciones (Versiones)
* 

|  |
| --- |
| **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.

**NO HAY EVIDENCIA DE REUNIONES, YA QUE ESTOY DESARROLLANDO EL PROYECTO YO SOLO.**