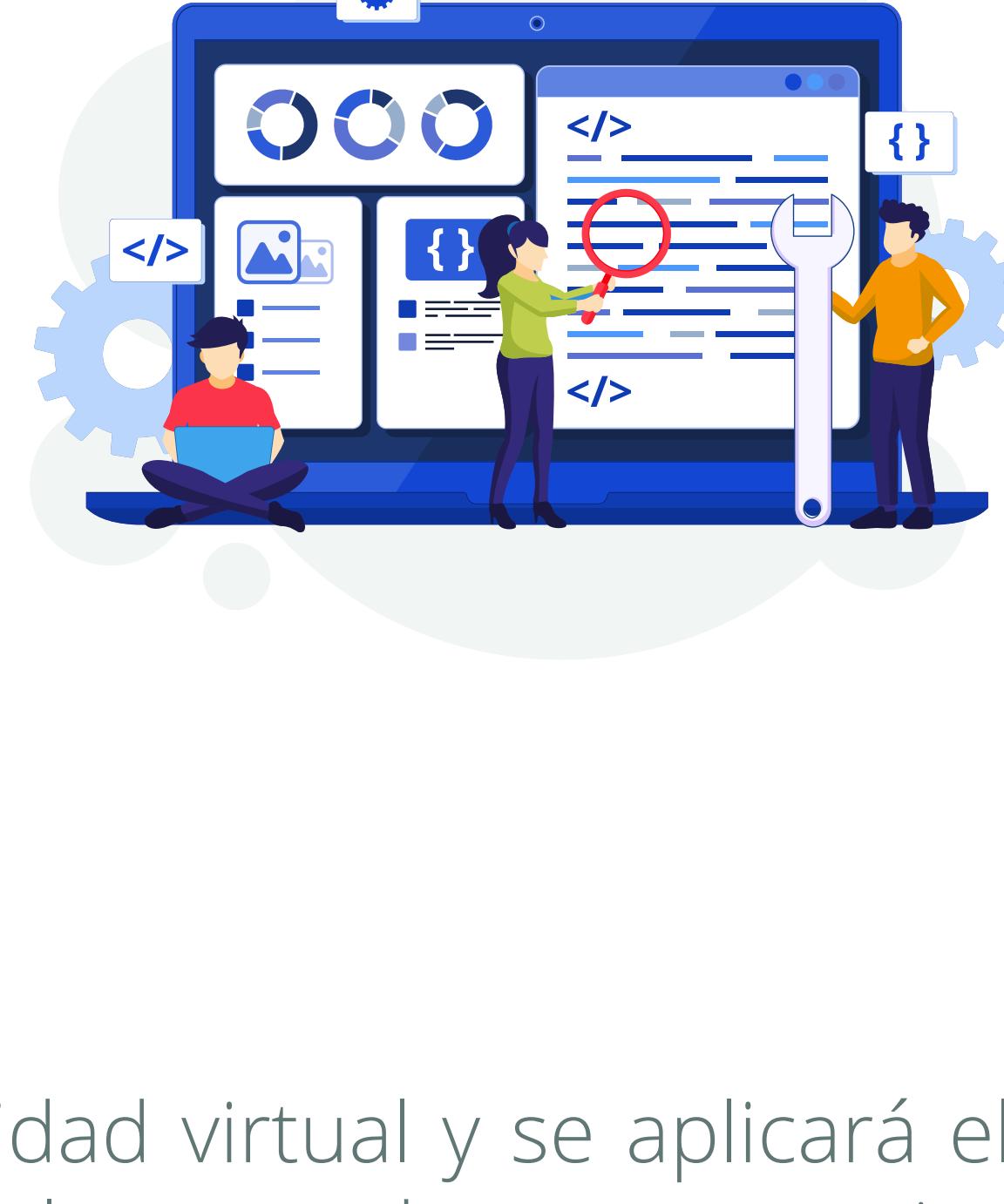


[FORMACIÓN POR CICLOS]

# Desarrollo de APLICACIONES WEB



## Metodología:

Te reiteramos que el curso se desarrolla en la modalidad virtual y se aplicará el “aprendizaje basado en proyecto” (ABP), que busca involucrarse de manera activa al proceso de aprendizaje partiendo de una situación con una problemática real que tenga relación con tu entorno y con tu objeto de estudio; dicha problemática se te presentará más adelante. Esta modalidad incrementa la necesidad de investigar, imaginar e impulsar tu creatividad y tu autoaprendizaje, como también tu capacidad para resolver problemas que permitan conectar el aula con los requerimientos del mundo laboral; además, requiere de un trabajo colaborativo y sistemático entre tú y tus compañeros del equipo.

El ciclo está compuesto por siete semanas de estudio, durante las cuales se irán desarrollando diversas actividades, como los *sprints*, que te permitirán ejecutar el desarrollo de aplicaciones web. Al finalizar el ciclo en la semana 8, deberás presentar el proyecto que desarrollaste por medio de un **Business Pitch**.

Para el desarrollo de la formación contaremos con lo siguiente:

**1. Componente teórico y práctico.** En esta etapa podrás acceder al aula virtual y estudiar los contenidos disponibles de cada tema, los cuales te ayudarán a desarrollar el proyecto. Además, cada encuentro con el tutor se realizará en sesiones sincrónicas con el fin de aclarar las dudas que te surjan. El trabajo práctico estará enfocado en el desarrollo de la aplicación web, en la cual deberás emplear diversas herramientas de programación.

**2. ¿Cómo se desarrollará el proyecto?** El desarrollo de aplicaciones web tendrá como base un proyecto dividido en cinco (5) sprints y deberá ser construido en un equipo de cinco (5) integrantes, que llevarán a cabo el desarrollo de un proyecto que se plantea más adelante. Entre los integrantes del equipo en el que te encuentras deben establecer un cronograma de actividades para sincronizar de manera ágil el trabajo a llevar a cabo.

Cada grupo tendrá asignado un tutor que le dará soporte dependiendo de las necesidades que surjan en la ejecución del proyecto. Además, el equipo debe diligenciar diariamente un formulario que encontrarán en plataforma, en el cual deberán registrar los avances, las dificultades y el cumplimiento de compromisos de cada uno de los integrantes.

El tutor hará un registro de seguimiento de los avances que haga el equipo, como también de las dificultades y el cumplimiento de los compromisos de cada uno de sus miembros según lo registrado en los **Dailies** y en el administrador de proyectos (**Trello**). Si dentro de ese seguimiento se tiene alguna consulta o se requiere aclarar dudas e inquietudes, podrás solicitar al tutor un encuentro sincrónico por medio de la plataforma; dicho encuentro quedará grabado en plataforma para que accedas a su contenido cada vez que lo necesites.

### 3. ¿Cuál es el proyecto a desarrollar?

#### Planteamiento de la situación problema a desarrollar

En los grupos de investigación de las diferentes universidades se desarrollan diversos tipos de proyectos de investigación y se detectan dificultades en el cumplimiento del plan de trabajo de dicha investigación. Mediante los reportes de avance del proyecto se controlará el progreso parcial del proyecto de investigación. Nos proponemos plantear un **modelo de sistema de información** que soporte la **gestión de proyectos de investigación** y mejore los procesos.

La siguiente figura corresponde a la arquitectura de la aplicación web que debes tener presente para desarrollar el proyecto.

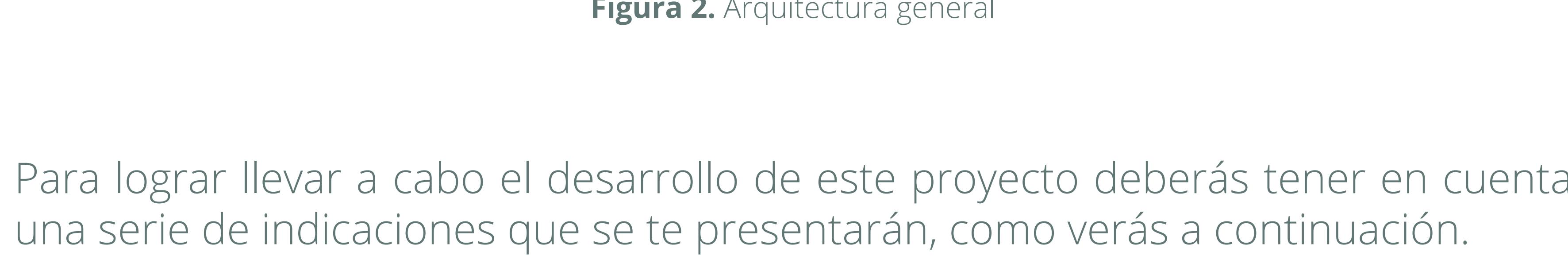


Figura 1. Alternativa 1: estructura con Nodejs y GraphQL

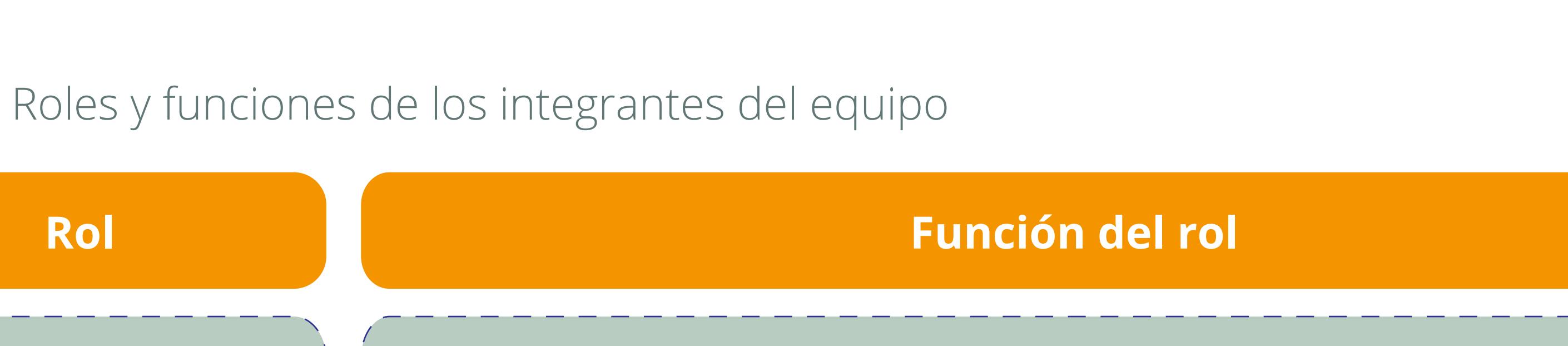


Figura 2. Arquitectura general

Para lograr llevar a cabo el desarrollo de este proyecto deberás tener en cuenta una serie de indicaciones que se te presentarán, como verás a continuación.

**Primera indicación.** Debes reunirte con cinco compañeros del curso y conformar un equipo de cinco integrantes, los cuales deben asignar un nombre al proyecto. Posteriormente se deben distribuir los roles para cada uno de los integrantes (ver la tabla 1 de los roles con sus funciones), teniendo presente siempre que todos los del equipo deben ser desarrolladores.

Tabla 1. Roles y funciones de los integrantes del equipo

Rol	Función del rol
-----	-----------------

1. Desarrollador Generar las fuentes para el frontend y el backend

2. Analista Levantar las necesidades del negocio para convertirlas en requerimientos del sistema

3. Product Owner Conocer el producto software que se va a elaborar en el proyecto

4. Administrador de BD Gestionar y disponer el motor de base de datos para el proyecto de desarrollo de software

5. Scrum master Gestionar todas las ceremonias y actividades de la metodología Scrum, y asignar los recursos a cada rol

## 4. ¿Cuál es la metodología que se utilizará para llevar a cabo el proyecto?



Figura 3. Metodología Scrum aplicada en el proyecto

El desarrollo del proyecto se abordará mediante metodologías ágiles, usando el marco de trabajo **Scrum** (figura 3), consiste en un proceso que se aplica de manera regular y frecuente a las buenas prácticas para trabajar colaborativamente en equipo y obtener el mejor resultado posible de un proyecto.

En Scrum se realizan entregas parciales y regulares de un producto final, y las actividades se priorizan de acuerdo a las necesidades para lograr la entrega de un producto completo. Las entregas son iterativas e incrementales, mostrando en cada entrega una versión funcional y completa del producto en general. Para desarrollar una aplicación web, Scrum ejecuta ciclos cortos de tiempos llamados **sprints**, que tienen como fin obtener un *feedback* en cada una de las entregas para continuar priorizando las actividades en un backlog, las cuales se van evacuando de acuerdo al método FIFO (*first in - first out*) y se priorizan según la importancia en el proyecto. Scrum prioriza los objetivos según las necesidades en cada una de las entregas.

Para lograr que se cumpla con el objetivo de dichas entregas, como se expresó anteriormente con relación al **Daily** (diligenciamiento de formulario), se debe establecer la ruta de ejecución de cada uno de los sprints. Un sprint normalmente tiene las siguientes ceremonias:

- **Sprint Planning**, que se realiza al comenzar el sprint.
- **Daily Scrum**, conocido comúnmente sólo como **la Daily**, que es una reunión diaria de 15 minutos en la que deben participar todos los integrantes del equipo. En esta reunión todas y cada una de las personas del equipo responden a las siguientes preguntas:
  - ¿Qué hice ayer para contribuir al Sprint Goal?
  - ¿Qué voy a hacer hoy para contribuir al Sprint Goal?
  - ¿Tengo algún impedimento que me impida entregar?

Dicha reunión tiene como objetivo diligenciar el formulario con las preguntas que se presentaron anteriormente, y además se debe hacer el seguimiento y la administración al proyecto a través del software **Trello**, en el cual se llevará el registro de las metas determinadas en los sprints según las historias de usuario con cada uno de los responsables.

- **Sprint Review**, que es la reunión que ocurre al final del sprint. Es organizada por el **Product Owner** y se revisa cuál es la situación de todas las historias (**Backlog**) y sus nuevas condiciones para priorizar el trabajo posterior.

En cada uno de los cinco sprints debes desarrollar las actividades planteadas. De igual manera, el equipo debe cumplir con los compromisos estipulados correspondientes al sprint en la semana asignada. Esta entrega recibirá una nota que será el soporte para la certificación, pero, aunque el trabajo sea grupal, la nota será de manera individual. Debes tener presente que cada sprints tienen un valor de **20%** para obtener un **100%** correspondiente al curso.

## 5. ¿Cuáles son las herramientas que se emplearán para desarrollar el proyecto?

Para lograr llevar a cabo el proyecto es necesario contar con un grupo de herramientas que te permitirán organizar, planear, estructurar y desarrollar la aplicación web.

- **Git**. Es un software de control de versiones de código distribuido, que se instala en la máquina local del desarrollador.
- **Github (Repositorio)**. Es una carpeta para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador
- **Trello (Administrador de Proyecto)**. Es un software de administración de proyectos que permite llevar el registro de todas las metas determinadas para cada uno de los sprints, así como su responsable.
- **Azure Devops (Suite)**. Es una nube mundial para hospedar máquinas virtuales Windows o Linux, independientemente de que use ASP.NET, Java, Node.js o PHP para desarrollar aplicaciones. Es un producto de Microsoft, que provee herramientas de control de versiones, gestión de requisitos, gestión de proyectos, compilación automatizada (*builds*), pruebas, y de manejo de despliegue cubriendo la totalidad del ciclo de vida del desarrollo.
- **Docker (Contenedor)**. Es un proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos.
- **Stack tecnológico**. La plataforma tecnológica expuesta en el curso “Desarrollo de aplicaciones web” es Azure DevOps, pero se puede utilizar otras plataformas en la nube, como Heroku, o la que se crea más pertinente para los despliegues.

Además, el lenguaje que se puede usar es JavaScript, y el stack tecnológico a emplear será MERN:

- MongoDB Atlas
- MongoDB
- Express
- React JS
- Node.js
- GraphQL

## 6. ¿Qué debo alcanzar para desarrollar la aplicación web?

Para que puedas desarrollar el proyecto que se te planteó, la aplicación web debe contar con los siguientes módulos:

- **Módulo de gestión de usuarios.**
  - **Gestión de ingreso al sistema de información.** El sistema tendrá las interfaces gráficas para el registro e ingreso a la aplicación. Para el registro se deberá ingresar el correo, identificación, nombre completo y contraseña, además el usuario podrá seleccionar qué tipo de usuario desempeñará en la plataforma (Estudiante, Líder o administrador), el usuario quedará registrado en el sistema por defecto con el estado de pendiente (Independientemente si se registra como administrador, líder o estudiante). Para la autenticación el usuario debe ingresar el correo y la contraseña.
  - **Gestión de estado de usuarios.** El sistema tendrá una interfaz gráfica para que el administrador pueda ver y cambiar el estado de los registrados como administrador, líder y estudiante (Pendiente/Autorizado/No autorizado), un líder puede cambiar el estado de los usuarios registrados como estudiantes (Pendiente/Autorizado).
  - **Gestión de perfil.** El sistema tendrá una interfaz gráfica para que el investigador o estudiante pueda actualizar los datos personales que ingresó cuando se registró (Incluyendo la contraseña).

- **Módulo de gestión de proyectos.** El sistema tendrá una interfaz gráfica para que los líderes registren los proyectos y otra para que el administrador, líder y estudiante puedan listar y ver los detalles de los proyectos, los líderes tendrán la opción que los direccione a la interfaz de actualizar los proyectos, mientras que los estudiantes podrán realizar la inscripción a los proyectos en los que desean trabajar, y seleccionar el proyecto en el que están trabajando y quieren agregar un nuevo avance.

Cada proyecto debe contar con los siguientes atributos: Identificador único del proyecto (Inmutable), nombre del proyecto, objetivos generales, objetivos específicos, presupuesto, fecha de inicio y terminación del proyecto, el documento de identificación y nombre del líder, el estado del proyecto (activo/inactivo) que será por defecto inactivo, y la fase del proyecto (iniciado, en desarrollo, terminado) que será nula por defecto.

- **Módulo de gestión de inscripciones.** Cómo se mencionó en el módulo de gestión de proyectos los estudiantes podrán inscribirse a un proyecto, si aún no lo están, mediante la interfaz que lista los proyectos presionando un botón que generará la inscripción. El sistema tendrá una interfaz para que los líderes de cada proyecto puedan listar las inscripciones y definir sus estados. Cada inscripción tendrá los siguientes atributos: Identificador único (inmutable), los identificadores del proyecto y estudiante (para relacionarlos), el estado de la inscripción (aceptada/rechazada), la fecha de ingreso, que se pondrá automáticamente cuando el líder actualice el estado de la inscripción a aceptada; y la fecha de egreso, que se pondrá automáticamente cuando la fase del proyecto se actualice a terminada o el estado del proyecto sea actualizado a inactivo.

- **Módulo de gestión de avances.** El sistema tendrá una interfaz para que los estudiantes registren los avances y otra para que los estudiantes y líderes puedan listar los avances, solo los estudiantes podrán actualizar los avances del proyecto en el que están inscritos, y el líder sólo podrá añadir sus observaciones a dicho avance. Cada avance debe contar con los siguientes atributos: Identificador único del avance (inmutable), el identificador del proyecto (para relacionarlo), la fecha del avance, la descripción del avance donde se especifique cuáles fueron los aportes a los objetivos del proyecto, y las observaciones del líder.

## Precisiones

1. Se requiere que el proyecto cumpla con las historias de usuario presentadas en este documento, pero se da libertad para que el equipo de desarrollo pueda acordar con el tutor cualquier funcionalidad adicional.
2. No se considera cambio de rol, es decir, cuando un usuario se registra como líder o como estudiante, ese será su rol y no podrá ser modificado.
3. Un líder y un estudiante pueden tener más de un proyecto asociado.
4. Despues de que un proyecto haya alcanzado la fase "Terminado", su estado cambiará automáticamente a inactivo y ni el líder o el estudiante podrán hacerle modificaciones (Aregar avances en el caso de los estudiantes o modificar algún atributo del proyecto para el caso del líder) y tampoco podrá ser reabierto (Cuando el proyecto tenga la fase terminado).
5. En caso tal de que ocurra un problema durante el desarrollo del proyecto, durante las fases "Iniciado" o "En desarrollo", el administrador podrá cambiar el estado de "Activo" a "Inactivo", y la fase permanece igual (En este caso el proyecto podrá ser reabierto).
6. Cuando un proyecto pasa de estar en estado "Activo" a "Inactivo", la fecha de egreso en las inscripciones que están en estado "Aceptado" y que cuya fecha de egreso está vacía, se debe guardar la fecha en la que se hizo la inactivación del proyecto.
7. En caso tal de que un proyecto sea reabierto, y un estudiante que estaba inscrito anteriormente en este proyecto desea continuar trabajando en él, deberá realizar una nueva inscripción (se contempla que un estudiante puede tener varias inscripciones en un proyecto en diferentes periodos de tiempo), en caso tal de ser rechazado en la nueva inscripción, este estudiante no podrá agregar avances al proyecto.
8. En caso tal de que un proyecto sea reabierto, nuevos estudiantes pueden enviar su solicitud de inscripción.
9. Cuando se sube el primer avance de un proyecto, automáticamente cambia la fase de "Iniciado" a "En desarrollo".
10. Un administrador sólo puede: Listar los usuarios registrados en la plataforma, actualizar el estado de los usuarios (Pendiente/Autorizado/No autorizado), listar los proyectos registrados en la plataforma, aprobar la creación de un proyecto (Activar un proyecto recién creado), cambiar el estado de un proyecto (Cuando el administrador cambia el estado de "Inactivo" a "Activo" y el campo de la fase está vacío, la fase se actualiza a "Iniciado" y se captura la fecha inicial automáticamente), cambiar la fase de un proyecto (De "En desarrollo" a "Terminado" y se captura la fecha final automáticamente), abrir nuevamente un proyecto que no tiene fase "Terminado" y tiene estado "Inactivo".
11. Un estudiante sólo puede: Registrarse en la plataforma, editar su información personal, listar todos los proyectos en plataforma, inscribirse en un proyecto en caso de que no lo esté, listar los avances de un proyecto al que pertenece, agregar avances a un proyecto en específico (Siempre y cuando esté inscrito en él), actualizar los datos de un avance en específico.
12. Un líder sólo puede: Registrarse en la plataforma, editar su información personal, ver la lista de todos los estudiantes registrados en la plataforma, autorizar el registro de un estudiante (Solo puede colocarlo en estado Autorizado o dejarlo en estado Pendiente, pero no puede colocarlo en estado No autorizado), crear un nuevo proyecto (Por defecto su estado será inactivo y el campo de la fase debe estar vacío), listar sólo los proyectos registrados en la plataforma que él lidera, actualizar los datos de un proyecto que él lidera (Y tienen estado "Activo"), aceptar o rechazar las inscripciones de los estudiantes a uno de los proyectos que lidera, listar los avances de un proyecto que lidera, agregar observaciones a un avance en específico de un proyecto que lidera.

**Segunda indicación.** Luego de conformado el equipo, se deben abordar los sprints, los cuales contienen las indicaciones necesarias para el desarrollo de la aplicación web. Cada sprint se detalla en un documento que encontrarás en la plataforma y que deberás ejecutar para poder alcanzar los objetivos propuestos. En la tabla 2 se presenta un resumen de lo que contiene cada sprint y el porcentaje de evaluación para cada uno de ellos.

Tabla 2

Semana de entrega	Sprint	¿Qué se desarrollará?	Porcentaje de evaluación
Semana (2)	<b>Sprint 1:</b> Creación de un repositorio en Github <a href="#">Clic aquí</a>	<ul style="list-style-type: none"><li>Tener creado el repositorio de las fuentes en GitHub</li><li>Realizar pull/request con cada cambio al proyecto</li><li>Tener creadas las ramas principales Development, Release, Master</li><li>Permisos a todos los integrantes del equipo</li><li>Realizar comentarios sobre cada cambio que se sube al repositorio</li><li>Todos los integrantes deben añadir su presentación al repositorio usando pull request</li></ul>	20%
Semana (3)	<b>Sprint 2:</b> Creación y modelamiento de bases de datos en MongoDB <a href="#">Clic aquí</a>	<ul style="list-style-type: none"><li>El modelado de base de datos debe tener todas las entidades (documento) como usuarios, roles, proyectos, estado y tipo plenamente identificados</li><li>Las entidades deben tener definidos los valores de las propiedades</li><li>Identificar los comandos sobre MongoDB para poblar datos en las estructuras del proyecto</li><li>Identificar los comandos para traer la información almacenada en las estructuras</li></ul>	20%
Semana (5)	<b>Sprint 3:</b> Desarrollo de Backend con NodeJS and GraphQL <a href="#">Clic aquí</a>	Para este sprint se deberán realizar las tareas derivadas de las historias de usuario relacionadas con el backend y la respectiva integración con la base de datos que fue creada en el sprint anterior.	20%
Semana (6)	<b>Sprint 4:</b> Desarrollo del Frontend con ReactJS <a href="#">Clic aquí</a>	Para este sprint se deberán realizar las tareas derivadas de las historias de usuario relacionadas con el frontend y la respectiva integración con el backend desarrollado en el sprint anterior.	20%
Semana (7)	<b>Sprint 5:</b> Adopción de prácticas DevOps <a href="#">Clic aquí</a>	<ul style="list-style-type: none"><li>Todo código implementado debe tener pruebas unitarias</li><li>Implementar integración continua</li><li>Usar contenedores de Docker</li><li>Desplegar la aplicación</li></ul>	20%

## Product backlog (Historias de usuario)

Uno de los recursos que se emplean dentro de la metodología Scrum para organizar la información es la historia de usuarios, la cual busca brindar la información de manera corta y simple. A continuación, te presentamos un esquema de la historia de usuario que debes manejar para el proyecto. También puedes profundizar un poco sobre la manera como se emplea esta herramienta a través del siguiente enlace:

<https://scrum.mx/informe/historias-de-usuario>

**Feature:** Como administrador, líder o estudiante necesito administrar la información de los proyectos.

### Historia de usuario: HU\_001

**Como** usuario

**Dado** que voy a ingresar al sistema de gestión de proyectos

**Cuando** necesite registrarme en el sistema

**Entonces** podré ingresar los datos de registro (Incluyendo elegir el rol al que aspiro)

### Historia de usuario: HU\_002

**Como** usuario autorizado

**Dado** que voy a ingresar al sistema de gestión de proyectos

**Cuando** necesite autenticarme en el sistema

**Entonces** podré ingresar mi correo y contraseña para ser validados

### Historia de usuario: HU\_003

**Como** usuario

**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** necesite actualizar la información personal

**Entonces** podré ingresar los datos que deseo actualizar

### Historia de usuario: HU\_004

**Como** administrador

**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** requiera ver la lista de los usuarios registrados en la plataforma (Tanto autorizados como no autorizados)

**Entonces** podré ver la información de los usuarios registrados en la plataforma

### Historia de usuario: HU\_005

**Como** administrador

**Dado** que estoy viendo la lista de los usuarios registrados en la plataforma

**Cuando** requiera aceptar un usuario en la plataforma

**Entonces** podré cambiar el estado del usuario

### Historia de usuario: HU\_006

**Como** administrador

**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** requiera ver la lista de proyectos registrados en la plataforma (Tanto autorizados como no autorizados)

**Entonces** podré ver la información de los proyectos registrados en la plataforma

### Historia de usuario: HU\_007

**Como** administrador

**Dado** que estoy viendo la lista de los proyectos registrados en la plataforma

**Cuando** requiera aprobar la creación de un proyecto

**Entonces** podré actualizar el estado del proyecto

### Historia de usuario: HU\_008

**Como** administrador

**Dado** que estoy viendo la lista de los proyectos registrados en la plataforma

**Cuando** requiera activar o inactivar un proyecto

**Entonces** podré actualizar el estado del proyecto

### Historia de usuario: HU\_009

**Como** administrador

**Dado** que estoy viendo la lista de los proyectos registrados en la plataforma

**Cuando** requiera cambiar la fase de un proyecto de "En desarrollo" a "Terminado"

**Entonces** podré actualizar la fase del proyecto.

### Historia de usuario: HU\_010

**Como** líder

I

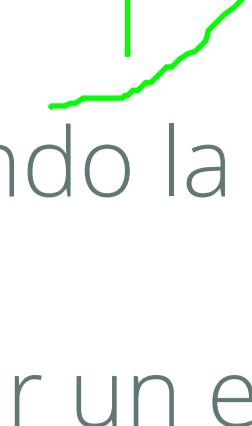
**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** requiera ver la lista de los estudiantes registrados en la plataforma (Tanto autorizados como no autorizados)

**Entonces** podré ver la información de los estudiantes registrados en la plataforma

## **Historia de usuario: HU\_011**

**Como** líder



**Dado** que que estoy viendo la lista de los estudiantes registrados en la plataforma

**Cuando** requiera aceptar un estudiante en la plataforma

**Entonces** podré cambiar el estado del estudiante de "Pendiente" a "Autorizado"

## **Historia de usuario: HU\_012**

**Como** líder



**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** requiera liderar un nuevo proyecto

**Entonces** podré crear un nuevo proyecto

## **Historia de usuario: HU\_013**

**Como** líder



**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** requiera ver los proyectos que lidero (Tanto los que están en estado "Activo" como "Inactivo").

**Entonces** sólo podré listar los proyectos que tengo a cargo

## **Historia de usuario: HU\_014**

**Como** líder



**Dado** que estoy viendo la lista de los proyectos que he registrado en la plataforma

**Cuando** requiera actualizar los datos de uno de los proyectos que lidero y que está en estado "Activo"

**Entonces** podré editar la información relacionada al proyecto cuya información necesito actualizar (Solamente puede actualizar: Nombre del proyecto, los objetivos generales, específicos y el presupuesto)

## **Historia de usuario: HU\_015**

**Como** líder

**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** requiera revisar las solicitudes pendientes por aceptar o rechazar de estudiantes de inscripción a mis proyectos

**Entonces** podré listar las solicitudes realizadas por los estudiantes.

## **Historia de usuario: HU\_016**

**Como** líder

**Dado** que estoy viendo la lista de solicitudes de inscripción a los proyectos que lidero

**Cuando** requiera cambiar el estado a la solicitudes

**Entonces** podré aceptar o rechazar sus inscripciones

## **Historia de usuario: HU\_017**

**Como** líder



**Dado** que estoy viendo la lista de los proyectos que he registrado en la plataforma

**Cuando** necesite realizar una revisión a uno de mis proyectos

**Entonces** podré listar la información relacionada al proyecto que deseo revisar (Incluyendo los avances).

## **Historia de usuario: HU\_018**

**Como** líder

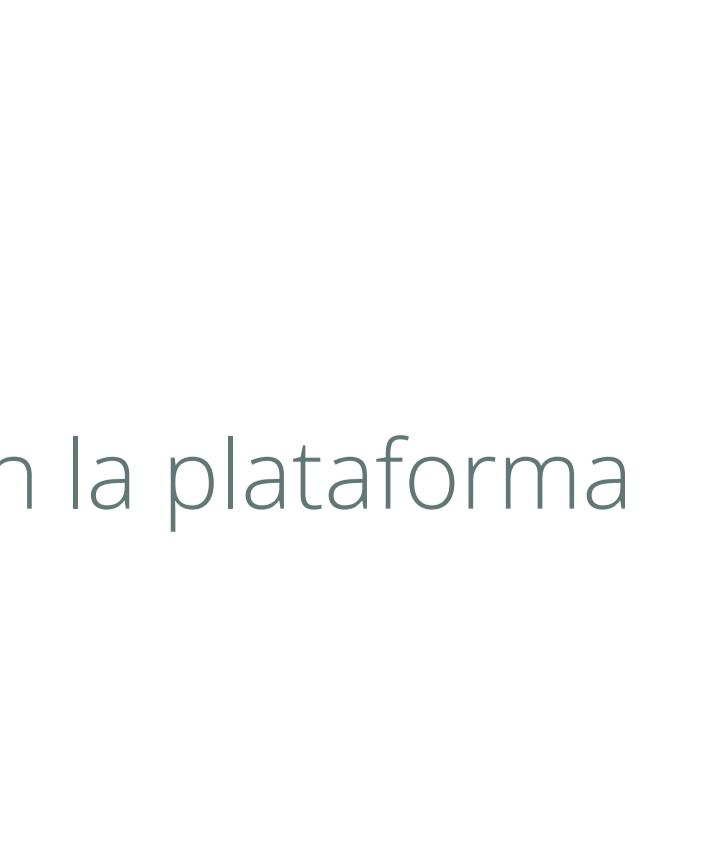
**Dado** que estoy viendo la lista de los avances registrados en uno de los proyectos que lidero

**Cuando** necesite agregar observaciones a un avance en uno de mis proyectos

**Entonces** podré actualizar el campo de observaciones del avance seleccionado.

## **Historia de usuario: HU\_019**

**Como** estudiante



**Dado** que ingresé al sistema de gestión de proyectos

**Cuando** requiera ver la lista de proyectos registrados en la plataforma

**Entonces** podré ver la lista de los proyectos registrados en la plataforma

## **Historia de usuario: HU\_020**

**Como** estudiante

**Dado** que estoy viendo la lista de los proyectos registrados en la plataforma

**Cuando** requiera inscribirme en un proyecto

**Entonces** podré generar una solicitud de inscripción al proyecto

## **Historia de usuario: HU\_021**

**Como** estudiante

**Dado** que mi inscripción a un proyecto fue aceptada

**Cuando** requiera listar los avances de un proyecto en el que estoy inscrito

**Entonces** podré ver la lista de los avances del proyecto registrados

## **Historia de usuario: HU\_022**

**Como** estudiante

**Dado** que estoy viendo la lista de proyectos

**Cuando** requiera registrar avances a un proyecto en el que estoy inscrito (Y este se encuentra en estado activo)

**Entonces** podré ingresar la descripción de mi avance en el proyecto

## **Historia de usuario: HU\_023**

**Como** estudiante

**Dado** estoy viendo la lista de avances de un proyecto

**Cuando** requiera actualizar la información de un avance

**Entonces** podré modificar la descripción del avance

## **Historia de usuario: HU\_024**

**Como** usuario

**Dado** que quiero ingresar a la aplicación mediante un navegador

**Cuando** intente ingresar a la aplicación mediante una URL

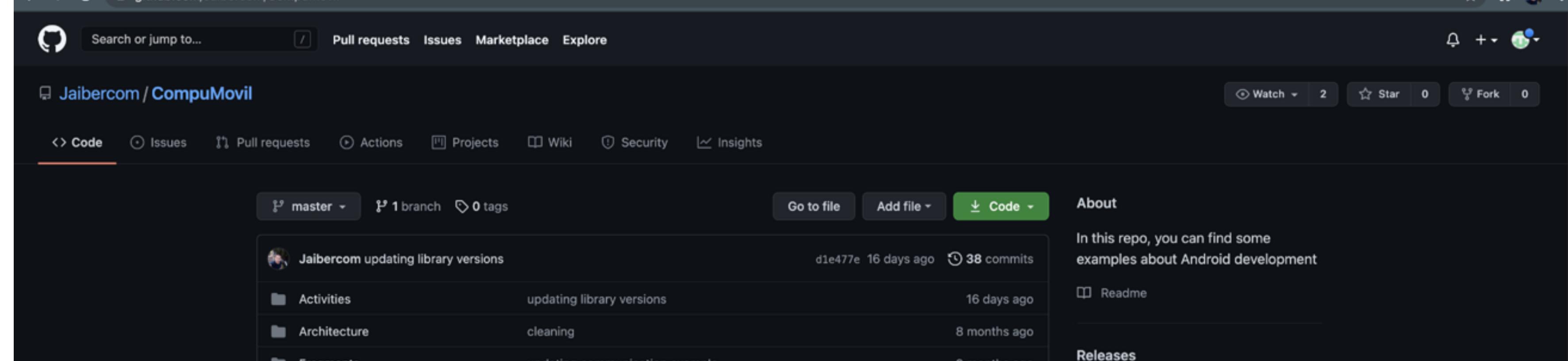
**Entonces** podré interactuar con la aplicación web

## **¿Cómo se llevará el seguimiento del desarrollo de la aplicación web?**

Para llevar a cabo el seguimiento del proceso de desarrollo vamos a utilizar dos herramientas: [Trello](#) y [Github](#). En Trello vamos a llevar un control de las tareas asignadas a cada miembro del grupo. Se recomienda crear las diferentes tareas para completar las historias de usuario, de tal manera que, toda la historia realizada de manera individual permita obtener la completitud de la historia de usuario.

Se deben crear cuatro columnas Backlog, ToDo, Progress y Done (figura 4).

- **Backlog.** En esta columna se encuentran las tareas a realizar, en general, de todo el proyecto; se obtienen al dividir cada historia de usuario en tareas.
- **ToDo.** Aquí van las tareas que se van a desarrollar en el sprint correspondiente.
- **Progress.** Aquí van las tareas que se están desarrollando en el momento; deben tener un responsable y un label que indique el sprint.
- **Done.** En esta columna se mueven las tareas que ya se terminaron.



**Figura 4.** Tareas en Trello

**Github** es un sitio web en la nube que permite hospedar el código de las aplicaciones de cualquier desarrollador, y usa el protocolo Git. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y puedan colaborar con el desarrollo. Estas herramientas nos permiten tener procesos bien definidos para gestionar todos los cambios suscitados a lo largo del proyecto, y mediante el repositorio GitHub controlar e identificar los artefactos para llevar la trazabilidad de los mismos, mediante la adecuada información de las características del cambio que se presenta y poder devolverse en caso de error hacia el estado del componente anterior, permitiendo así saber quiénes están involucrados. Ver (figura 5)



**Figura 5.** Ventana de GitHub