**Plataforma Sistema de Gestión de Proyectos**

Naughty Dog es una empresa de desarrollo, la cual no tiene de una forma organizada con la que realizar sus proyectos. Por lo que Naughty Doglo contacta a usted y su equipo para tercerizar el desarrollo de una plataforma de gestión de proyectos, puesto que en el momento no disponen de los recursos humanos para realizar dicha plataforma. Se debe tener en cuenta que un empleado puede estar asignado a varios proyectos.

Los tipos de usuarios que debe manejar el sistema son administrador, usuario líder y usuario subalterno.

Un usuario líder debe estar en la capacidad de:

* Crear un proyecto. Se debe definir un título, una descripción y una fecha de inicio. La fecha final del proyecto es igual a la mayor fecha de finalización entre las actividades asignadas al proyecto.
* Crear las actividades del proyecto, indicando el tiempo estimado de las mismas, la fecha de inicio y la fecha de finalización. La fecha de inicio de una actividad debe ser igual o superior a la fecha de inicio del proyecto.
* Asignar usuarios subalternos a un proyecto.
* Finalizar un proyecto.
* Generar los siguientes informes:
  + Proyectos terminados.
  + Proyectos iniciados.
  + Horas por proyecto.
    - Sumatoria de horas definidas dentro de las actividades.
    - Sumatoria de horas reportadas por los empleados.
  + Horas por empleado y proyecto.

El usuario subalterno deberá estar en la capacidad de:

* Ver proyectos asignados.
* Ver actividades de un proyecto asignado.
* Cargar horas al desarrollo de una actividad de un proyecto.
* Marcar actividad como iniciada/completada.

Consideraciones:

* El tiempo de una fase se calcula con base en los tiempos individuales de las actividades que la conforman.
* El tiempo del proyecto es la suma del tiempo de desarrollo de las actividades.

Un administrador se encarga de gestionar los usuarios internos y además ejerce control total de la plataforma.

***Notas***:

* ***Los datos suministrados por los usuarios deben cumplir la política de privacidad de datos vigente, es decir las contraseñas de los usuarios deben almacenarse de forma cifrada y la conexión al servidor debe realizarse de forma segura.***
* ***La base de datos debe ser no relacional.***
* ***Se debe utilizar una arquitectura desacoplada, es decir, una API back end y un front end que consuma dicha API.***
* ***La API back end debe estar desarrollada en Express.js***
* ***El front end debe estar desarrollado en React.js***
* ***Se debe utilizar Bootstrap como librería CSS para manejar los estilos de su aplicación.***