**Plataforma Sistema de Nómina**

La empresa Alphabet actualmente maneja su sistema de nómina a empleados de forma manual, es decir la persona encargada del proceso tiene que construir la nómina cada mes y enviarla al banco correspondiente para realizar el proceso de pago y adicionalmente tomar en cuenta días libres o vacaciones. Razón por la cual la empresa Alphabet lo contacta a usted y su equipo con el fin de construir una plataforma donde sea posible gestionar la nómina de sus empleados.

Los tipos de usuarios que debe manejar el sistema son administrador, usuario empleado y usuario nómina.

Un usuario nómina deberá estar en la capacidad de:

* Crear empleados con su información básica y la fecha en la que ingresó.
* Ingresar la información salarial de los empleados (nómina) de los empleados.
* Aprobar/rechazar permisos. Los permisos deben ser remunerados (se pagan los días ausentes) o no remunerados (no se pagan los días ausentes).
* Aprobar/rechazar vacaciones
* Generar los siguientes reportes:
  + Empleados que devengan más de un salario X.
  + Empleados que devengan menos de un salario X.

Un usuario empleado debe estar en la capacidad de:

* Gestionar su información básica (nombre, apellidos, teléfono, número de cédula).
* Solicitar vacaciones. Solo puede solicitar 15 días calendario durante el año y no son acumulables.
* Descargar reporte de pago indicando el mes.
* Solicitar certificados laborales.
* Solicitar vacaciones indicando la fecha.
* Solicitar permisos no remunerados.

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

Notas:

* Solo se van a tener en cuenta los pagos mensuales de salario.
* Las vacaciones se tendrán en cuenta como si la persona siguiese trabajando (remuneradas).

***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.***