**PARCIAL 2**

**Objetivo**

Producir un API Rest que permita comunicarse vía protocolo HTTP con un front web basado en HTML + Javascript.

**Consigna**

El grupo de desarrollo **Global Solutions SAS** está buscando una aplicación web que permita organizar tareas en un tablero de seguimiento estilo **Trello**.

Para esto, la empresa ha hecho una convocatoria entre los estudiantes de Programación en Red de Ingeniería Telemática de la Universidad Icesi.

**Vista previa**

La empresa espera algo como la siguiente ilustración:

**Global Solution SAS**

*Descripción de la tarea*

Nueva Tarea

Done

Doing

To do

2020-10-15

2020-10-30

2020-10-25

Hacer document de análisis del proyecto final de Ecosistemas de Aplicaciones

Desarollar el Quiz 2 de ecosistemas de aplicaciones

Hacer propuesta gráfica del proyecto final de Ecosistemas de Aplicaciones

2020-10-30

Implementar proyecto final de Ecistemas de aplicaciones

NOTA: No necesariamente debe ajustarse a este diseño, pero sí debe tener las secciones *To do*, *Doing* y *Done*. Y cada tarjeta debe dar la posibilidad de enviar la tarea a la siguiente etapa, anterior etapa o poder eliminar la tarjeta.

**Requerimientos**

Para el tablero de tareas cree tres secciones: *To do*, *Doing* y *Done*.

Cuando el usuario crea una nueva tarea, esta queda en la sección *To do*. La tarjeta contiene la descripción de la tarea, la fecha en la que se escribió la tarea y dos botones:

1. *Botón de eliminación (Rojo)*

Este botón permite eliminar la tarea esté en la fase en la que esté.

1. *Botón de avance (Azul)*

Este botón permite ascender la tarea: de *To do* pasa a *Doing*, de *Doing* pasa a *Done*.

1. *Botón de retroceso (Amarillo)*

Este botón permite descender la tarea: de *Done* pasa a *Doing*, de *Doing* pasa a *To Do*.

**Calificación**

1. **Modelo de datos propuesto para el sistema. 1 punto:**

Las tablas y entidades del sistema tienen las variables necesarias y suficientes para el desarrollo de la aplicación.

NOTA: NO OLVIDE ADJUNTAR SU BASE DE DATOS (.SQL)

1. **Lógica del negocio*.* 2 punto:**

El backend ofrece los servicios necesarios y suficientes para el desarrollo de la aplicación.

-0.4 puntos si la estructura no está dividida en: entidades, dtos, providers, servicios.

1. **Comunicación del front web. 1 punto:**

Cree una página web similar a la ilustración que se comunique con los servicios generados en el punto 2.

1. **Generación de componentes. 1 punto:**

Implementa un modelo de página web orientada a componentes. Genera los componentes necesarios y suficientes para crear la página

Exponga su solución usando algún diagrama. Recomendados: **Diagrama de secuencia** de las acciones de la aplicación o un **diagrama de componentes** que resuma todo el sistema