

PARCIAL 2

Objetivo

Desarrollar una aplicación fullstack conectada a base de datos que permita solucionar el problema planteado.

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:

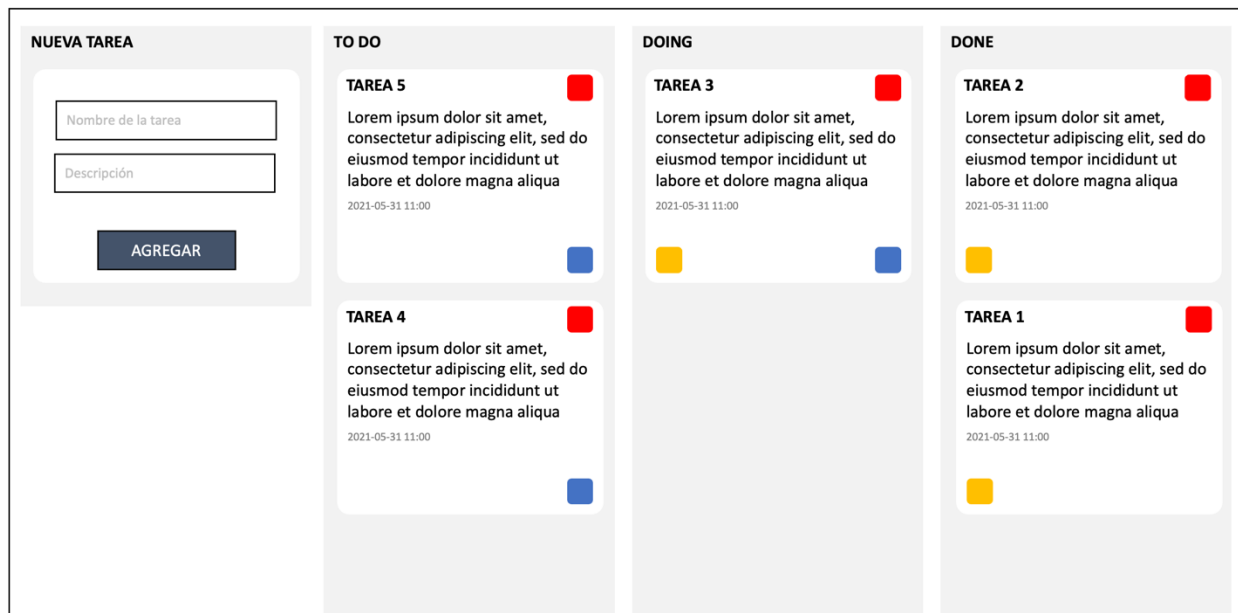


Figura 1. Ilustración del parcial 2

NOTA: El parcial no debe tener esta misma apariencia. Puede tener una apariencia más simple. La interfaz gráfica no está siendo evaluada



Programación en red
Ingeniería Telemática

Requerimientos

1. Modelo

Cree un modelo de la aplicación que permita soportar la funcionalidad de tablero de tareas.

2. API Rest

Cree un API con los endpoints necesarios para ofrecer la funcionalidad completa.

3. Containers

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 algunos botones cuya función se describe a continuación.

4. Componentes

Cree un componente para la tarea. Este debe representar los datos de la tarea (descripción, título, fecha de creación) y además debe tener algunos botones:

A. Botón de eliminación (Rojo)

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

B. Botón de avance (Azul)

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

C. Botón de retroceso (Amarillo)

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

El botón azul (de ascenso) no debe aparecer en la sección de done. Asimismo el botón amarillo (de descenso) no debe aparecer en la sección de 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. Monte su(s) tabla(s) en la base de datos MySQL de Icesi

NOTA: ADJUNTE UN ARCHIVO SEÑALANDO CUÁLES SON SU(S) TABLA(S)

2. API Rest. 1 punto:

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

3. 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, para ofrecer al usuario el funcionamiento de la aplicación.

4. Generación de componentes. 1 punto:

El componente de tarea está implementado correctamente. Permite representar gráficamente los datos y dotar a cada tarea de las funciones planteadas.

5. Despliegue. 1 punto:

La aplicación se encuentra desplegada. Ya sea como un monolito o como despliegue backend separado del despliegue front.

NOTA: ADJUNTE LA URL DE SU PÁGINA DESPLEGADA

BONUS. 0.5 puntos

Modifique el componente de modo que pueda implementar la misma funcionalidad, pero arrastrando y soltando las tareas (drag and drop)