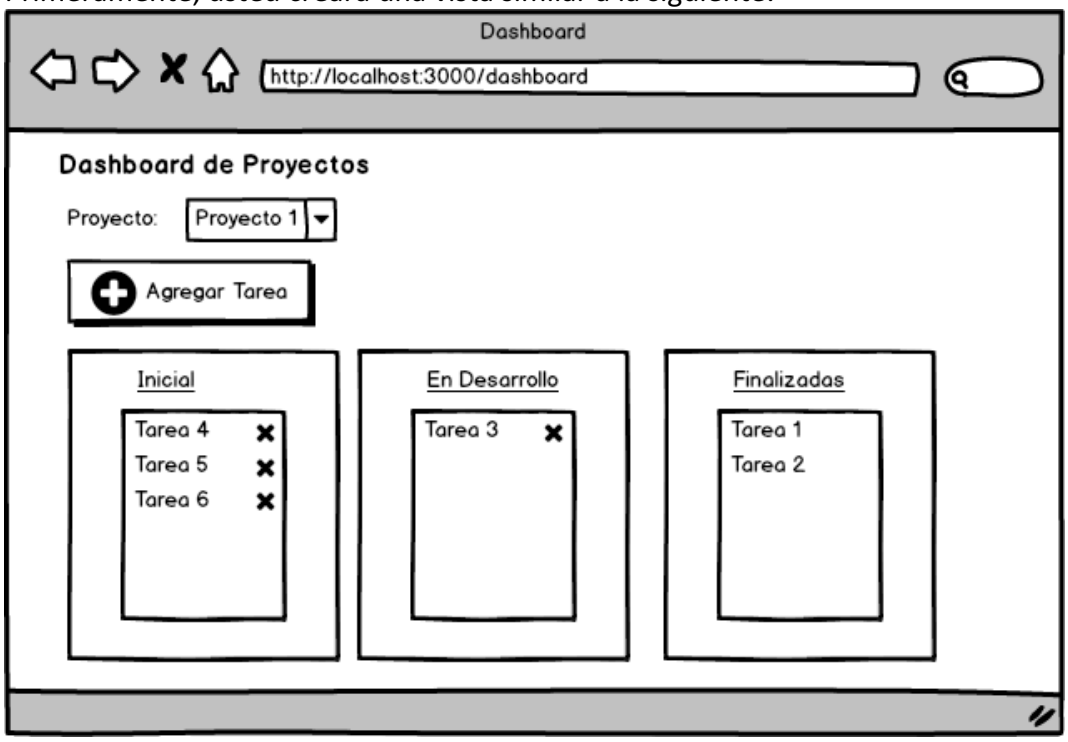
Dashboard de Tareas

Para esta practica se le solicita implementar un administrador de tares.

Vista dashboard

Primeramente, usted creara una vista similar a la siguiente:

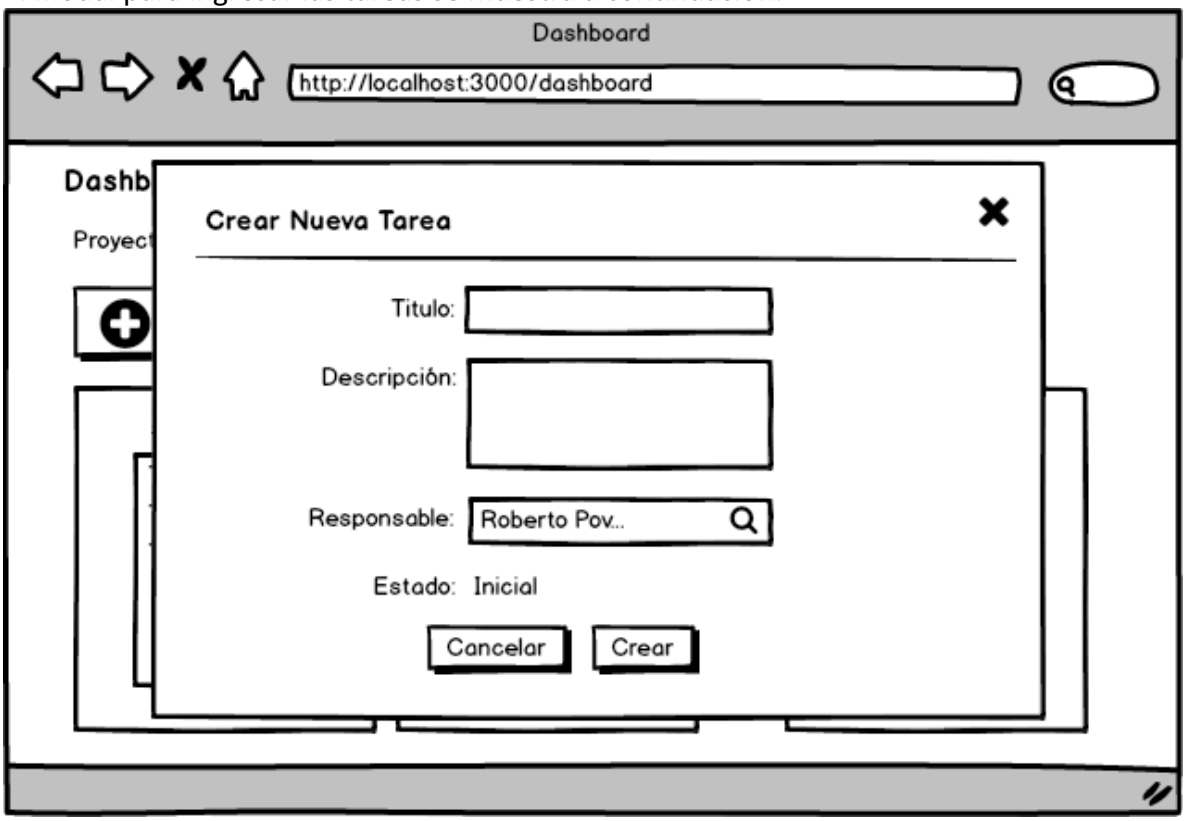


La siguiente vista esta compuesta de las siguientes partes:

1) En la parte superior se encuentra un combobox que mostrara todos los proyectos existentes. Este combo será cargado a través de un API que retornara todos los proyectos disponibles en el sistema.
2) Existe un botón llamado “Agregar Tarea” que abrirá un modal para el ingreso de una nueva tarea.
3) Existen tres “cajas” que muestran el progreso de las tareas donde existen tres estados (inicial, en desarrollo, finalizadas).
4) La información de estas cajas serán cargadas a través de un API REST que devolverán un JSON con todas las tareas de un proyecto determinado. Al seleccionar otro ítem del combo, se tendrá que actualizar la información de acuerdo al nuevo proyecto seleccionado.
5) Todas las tareas de los estados inicial y en desarrollo deberán poseer un botón para poder eliminarlas. Si un usuario presiona el botón, usted deberá preguntar si esta seguro de realizar esa acción. En el endpoint que usted implemente deberá validar que la tarea este en esos estados.

Modal para ingreso de tareas

El modal para ingresar las tareas se muestra a continuación:

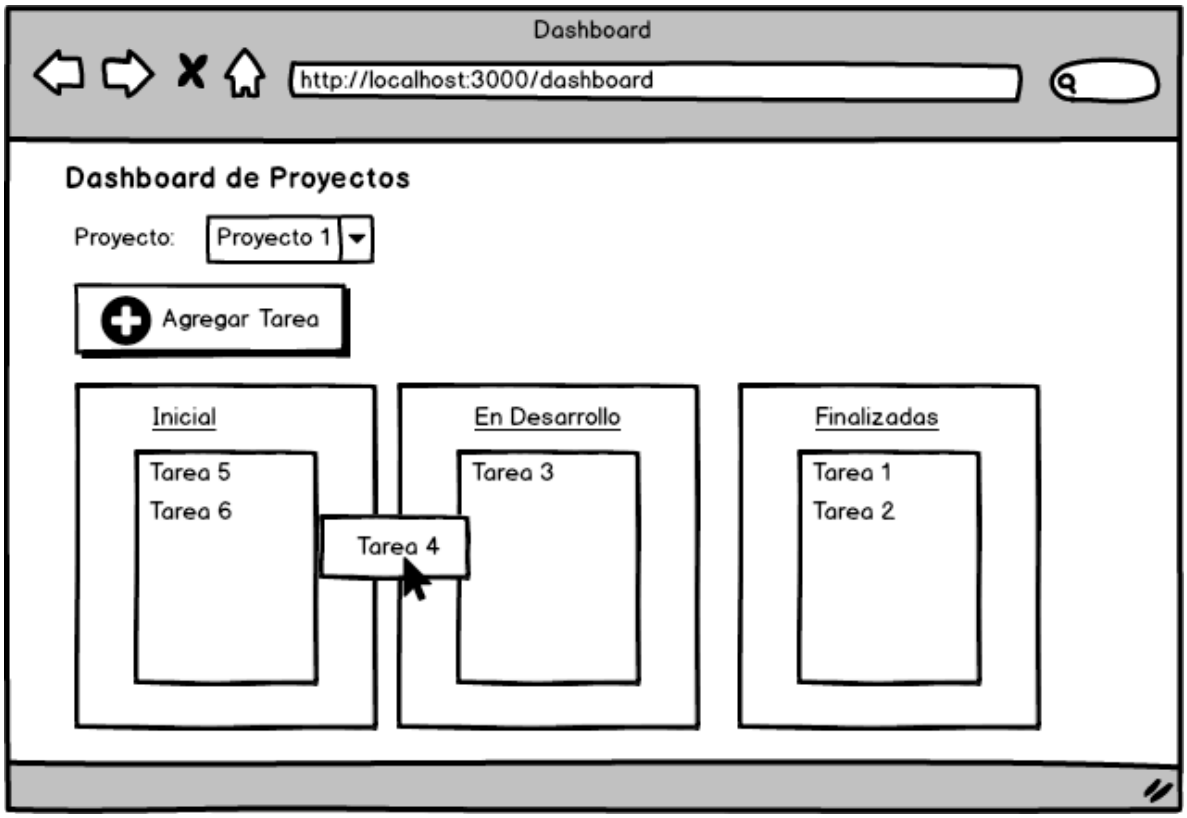


Este modal esta compuesto por:

1) Titulo
2) Descripción
3) Responsable, el cual tendrá que ser implementado utilizando un “autocomplete” y los datos los tendrá que obtener enviando un requerimiento http con las letras tipeadas hasta ese momento y el servidor deberá responder con un JSON de nombres de usuarios del sistema.
4) Las tareas cuando se crean siempre tendrá el estado de Inicial.

Cambiar el estado de una tarea

Para cambiar el estado de una tarea deberá ser implementado utilizando la técnica del drag and drop, donde en este caso el usuario seleccionara una tarea y deberá mover la tarea hacia una de las cajas, y automáticamente se guardara el nuevo estado para esa tarea. No existe ningún orden para las tareas, si usted desea las puede ordenar de acuerdo a fecha de creación.



API REST

Usted implementara un API para:

- Obtener la lista de proyectos disponibles
- Obtener la lista de nombres de usuarios a partir de un prefijo.
- CRUD para las tareas.

Arquitectura

Usted deberá utilizar:

- Node.js en conjunto con el framework express
- Todas las librerías del lado del servidor que utilice deberán estar descritas en el archivo package.json
- Un template engine a su elección.
- Bower para cargar las librerías del lado del cliente.
- Tres módulos para las rutas:
    - Proyectos
    - Usuarios
    - Tareas
- Deberá utilizar la base de datos no relacional mongodb (y deberá aprender a como modelar las relaciones ver: ref1, ref2, ref3). Defina los atributos que usted crea necesarios para todas las entidades.
- Para su facilidad, usted puede utilizar el servicio de mlab que ofrece mongodb en la nube para proyectos pequeños (plan sandbox). Revisar: https://mlab.com/

- Deberá utilizar la librería mongoose.
- Deberá crear un directorio llamado **models** donde estarán todos los modelos descritos.
- Finalmente, existe una excelente guía para la construcción de un API Rest con express y mongodb: https://scotch.io/tutorials/build-a-restful-api-using-node-and-express-4