



## Prueba técnica Full Stack

El objetivo de la presente prueba es evaluar el nivel de conocimiento de desarrollo backend y frontend para posibles aspirantes al cargo de desarrollador Full Stack. Para todo problema de programación existen múltiples soluciones, pero usualmente hay mejores prácticas a la hora de abordar un determinado desafío. Con esta prueba se busca no solo recibir por parte del aspirante un código funcional, sino también un código implementado adecuadamente.

### Descripción

Este proyecto está diseñado para poner a prueba su conocimiento de tecnologías de backend y frontend, evaluando su capacidad para crear APIs RESTful e interfaces de usuario prestando atención a los detalles. También es importante la aplicación de buenas prácticas de desarrollo.

### Funcionalidades de backend

Debe implementar un API RestFul en cualquier framework que cuente con los siguientes endpoints:

- Crear un evento. El evento debe tener los siguientes campos:
  - Nombre
  - Descripción
  - Lugar (Lista de algunas ciudades de Colombia, obtenidas de una tabla).
  - Color (Lista con algunos colores para poder clasificar los eventos, obtenidos de una tabla).
  - Fecha
  - Hora de inicio
  - Hora de finalización
- Editar un evento en uno o más de sus campos.
- Eliminar un evento.
- Obtener todos los eventos del mes.

Consideraciones:

- No se permite que dos eventos ocurran simultáneamente o se traslapen, por lo cual el API deberá enviar un error en caso de que se intente crear o editar un evento que entre en conflicto con otro.
- La hora de finalización no puede ser anterior a la hora de inicio.
- No se permiten eventos que abarquen dos días diferentes.

## Funcionalidades de frontend

Debe crear una aplicación de un calendario usando React, Vue o Angular. En primer lugar debe renderizar una vista de un mes completo, similar a la que se muestra a continuación:

| Sunday | Monday | Tuesday | Wednesday | Thursday | Friday | Saturday |
|--------|--------|---------|-----------|----------|--------|----------|
| 30     | 31     | 1       | 2         | 3        | 4      | 5        |
| 6      | 7      | 8       | 9         | 10       | 11     | 12       |
| 13     | 14     | 15      | 16        | 17       | 18     | 19       |
| 20     | 21     | 22      | 23        | 24       | 25     | 26       |
| 27     | 28     | 29      | 30        | 31       | 1      | 2        |

La aplicación debe tener las siguientes funcionalidades:

- Debe mostrar los eventos existentes en cada uno de los días del mes (muestre solo el campo **nombre**).
- Al dar click en un día debe mostrarse en un modal o una nueva vista el listado completo de eventos programados para la fecha correspondiente.
- Al dar click en un evento deben aparecer en un modal o una nueva vista todos los campos del evento, un botón para editar y un botón para eliminar.
- Se debe permitir crear nuevos eventos por medio de un formulario con validaciones, bien sea en un modal o una nueva vista. El evento tiene los siguientes campos obligatorios:
  - Nombre
  - Lugar
  - Fecha
  - Hora de inicio
  - Hora de finalización

### Consideraciones:

- Solo debe mostrar un mes de 31 días. No hay navegación entre meses.
- Maneje adecuadamente el número de eventos en un mismo día para no desbordar el recuadro.
- Se espera ver un adecuado manejo del estado de la aplicación (Redux, Vuex, etc).
- Se espera ver una buena experiencia de usuario (indicación de errores, manejo de errores del backend, etc).
- Puede usar cualquier librería de elementos UI (modales, input fields, validadores) y manejo de fechas y horas. Sin embargo no use librerías de calendarios como Bootstrap Calendar.

### Entregable

Una vez haya completado la prueba haga llegar el link (Github, Gitlab, etc) o el comprimido de los repositorios al correo [mttovarg@slabcode.com](mailto:mttovarg@slabcode.com).

Si no logró terminar toda la prueba, por favor indíquenos las funcionalidades implementadas.