

PROBLEMÁTICA

Una compañía que se dedica a organizar eventos, desea validar su negocio en Internet. Por ello, decide contratarte para realizar un MVP que resuelva las siguientes necesidades:

- La aplicación debe ser web y contar con API para la interacción con el frontend.

Nota: tanto el backend (modelo y reglas de negocio) como la API deberá desarrollarse preferentemente en Golang. No se especifica ningún framework en particular.

- Los eventos podrán gestionarse (alta, baja, modificación) sólo por un rol administrador.
- Los eventos cuentan con un título, descripción corta, descripción larga, fecha y hora, organizador, lugar, y un estado (borrador o publicada).
- Cuando una publicación tiene estado borrador, sólo los administradores pueden visualizarla.
- Los usuarios pueden visualizar e inscribirse en eventos publicados, siempre y cuando la fecha del evento sea futura.
- Los eventos cuya fecha y hora ya hayan transcurrido, pueden visualizarse pero no inscribirse.
- Los usuarios (administradores o no) pueden visualizar los eventos activos (publicados y con fecha-hora en el futuro) como así los eventos completados (publicados pero con fecha y hora en el pasado). Este listado de eventos se debe poder filtrar por fecha, estado, y título.
- A su vez, se contará con un endpoint en el cual se mostrarán todos los eventos inscriptos, filtrándose por activos o completados.

CONSIDERACIONES GENERALES

1. Podes utilizar algún framework o combinación de ellos para resolver el problema, como así paquetes o librerías Golang que solucionen problemas puntuales.
2. Se espera poder reproducir el entorno de desarrollo fácilmente (gestión de dependencias mediante un docker, etc)
3. Estilo de código sujeto a los estándares del lenguaje.
4. Uso de clean architecture y patrones de diseño.
5. Como entregable se espera:
 - a. Un repositorio git con el desarrollo, que evidencia su utilización durante el desarrollo.
 - b. Un alto porcentaje de los requisitos cubiertos, especificando requisitos no alcanzados por el MVP.
 - c. Deseable: test automáticos (unitarios y/o integración).

OPCIONAL

De manera opcional, puede desarrollarse el frontend que interactúe con la API desarrollada. Esta aplicación web deberá ser desarrollada preferentemente en React.

En esta web, podrá:

- Ver el listado de eventos general, con los filtros disponibles.
- Visualizar un evento de forma detallada.
- Inscribirse al evento.
- Listado de eventos suscriptos.

Siéntase libre de añadir otras funcionalidades si lo ve necesario.

ENTREGA

La fecha de entrega está establecida para 7 días después de enviado el ejercicio práctico.

Para consultas puedes comunicarte a: contacto@wiselink.app