

CASO PRÁCTICO 2024: INSCRIPCIÓN A EVENTOS

Nuestro cliente nos envía lo siguiente:

- La solución de software que necesitamos nos debería permitir al organizador crear nuevos eventos (principalmente charlas abiertas) con una descripción, fecha, lugar (mostrar en Google maps), límite de personas.
- Luego los usuarios, utilizando la aplicación móvil (a desarrollar a futuro por un equipo externo) o desde una web, podrán listar los eventos y anotarse a un evento en estado “pendiente”. Deben ingresar nombre, apellido, teléfono, DNI y dirección de correo.
- El usuario debe recibir un correo electrónico con un link de confirmación para validación de dirección de correo existente.
- El organizador debe poder aprobar o rechazar a los inscriptos. Se informará al inscripto por correo electrónico del cambio de estado.
- Los usuarios deben poder consultar la lista de eventos a las que se anotaron y su estado de inscripción.

Del análisis de requerimientos inicial surgieron las siguientes preguntas, con sus correspondientes respuestas:

- ¿El usuario puede darse de baja de un evento?
Sí, debería poder darse de baja.
- ¿Es necesario que un usuario cree una cuenta? ¿cómo hace para consultar su estado?
No es necesario. Podríamos consultar por el mail y el DNI, y si coinciden, mostramos el estado de inscripción o le permitimos darse de baja.
- ¿Cuántos eventos suelen existir disponibles para inscribirse?
En promedio, hay entre 5 y 10 eventos por mes y se cargan los de los próximos 6 meses, totalizando entre 30 y 60 eventos. Los eventos promedian entre 50 y 250 inscriptos.
- ¿Existe un dominio ya comprado para utilizar?
No, habría que comprarlo.
- ¿Existe un proveedor de servicios cloud para publicar la aplicación web? ¿Hay preferencia por alguno en particular?
No.

- ¿Existe un servidor de correo para utilizar para el envío de validaciones y notificaciones?

No.

Actividad 1) Elaborar el documento de proyecto tomando como referencia al del ejemplo de [guru99](#), incluyendo inicialmente las siguientes secciones:

- Revisión y aprobación del documento
- 1. Introducción
 - 1.1 – Propósito
 - 1.2 – Scope de las pruebas, indicando que probaríamos y que no entre lo siguiente:
 - Funcionalidad, Usabilidad, Seguridad, Compatibilidad, Performance, UI, Hardware.
 - 1.3 – Definiciones, definiendo nombre de los roles para evitar confundir.
- 2. Requerimientos.
 - 2.1 – Descripción de Roles
 - 2.2 – Features por rol

Actividad 2) Escribir las historias de usuario para las features listadas en el apartado 2.2 del documento de proyecto, incluyendo las AC.

Incluir las user stories en el documento de proyecto, creando una nueva sección "2.3 – User Stories" para tal fin.

Actividad 3) Escribir los test cases asociados a las user stories en el documento de proyecto, creando una nueva sección "2.4 – Test Cases" para tal fin.

Actividad 4) Incluir en el documento la sección "3. Especificaciones", incluyendo el apartado "3.1 - Arquitectura" con un diagrama y una breve descripción de la arquitectura a implementar.

Actividad 5) Diseñar en borrador el contrato de API a construir.

- Listar los endpoints disponibles, así como los query parameters y los formatos de request y response
- Especificar los HTTP status codes para cada endpoint
- Tener en cuenta cuestiones de seguridad

Generar un contrato de API utilizando open API e incluir un link al archivo de contrato de API en el documento en la sección "3.2 – Definición de API"

Actividad 6) Construir la API tomando como base el contrato de la actividad 6. Colocar link al repositorio de código fuente en el documento en la sección “4 - Repositorio de código fuente”

Actividad 7) Desarrollar los test unitarios para las capas de controlador y de servicio de la API. Tener presente que:

- Para los test unitarios, se debe utilizar una base de datos en memoria.
- No podemos enviar e-mails cada vez que se ejecute el test unitario de inscripción a un evento. Deberemos hacer un mock.

Actividad 8) Utilizando el ejemplo del caso de, como administrador, añadir un evento, disponible en <https://eventify.tiiny.site/>, construir test automatizado utilizando Selenium para los siguientes casos:

1. Creación de evento fallida debido a descripción con longitud menor al mínimo de 40 caracteres.
2. Creación de evento fallida debido a capacidad de evento inferior a 1 persona
3. Creación de evento exitosa.

Elaborar un documento con capturas de pantalla de cada test mostrando los pasos y que se ejecutó correctamente y adjuntarlo. NO VAN EN EL MISMO DOCUMENTO DE LAS ACTIVIDADES ANTERIORES.