



PROYECTO LARAVEL – CREACIÓN DE EVENTOS

Vamos a crear una web de eventos. Tendremos dos niveles web diferentes, la parte pública donde la gente podrá ver los eventos que hay e inscribirse previo login, y otra parte, privada, que será una intranet donde los organizadores de eventos crearán los eventos.

CLASES

- **Asistente:** dni, nombre, apellidos, edad, email, dirección, ciudad, teléfono
- **Categoría:** nombre (cine, concierto, teatro, magia, festival, gaming, ...)
- **Evento:** nombre, fecha, descripción, ciudad, dirección, aforomax, tipo (online – presencial), numMaxEntradas, *categoría*.
- **Inscripción:** *asistente*, *evento*, numEntradas, estado (recibida, confirmada, cancelada).
- **CreadorEventos:** dni, nombre, apellidos, teléfono, email, dirección, ciudad, empresa.

ROLES

- **Asistente:** tendrás que registrarte en la web para asistir a eventos.
- **Creador eventos:** entrará en la “intranet” para dar de alta su evento. Puede cancelar un evento o una inscripción determinada.
- **Administrador:** entrará en la “intranet” para gestionar todo lo que se hace.

WEB PRINCIPAL

- **Login** de usuarios
- **Página principal:** saldrán los eventos próximos
- **Búsqueda de eventos por fecha, por ciudad y categoría.** Se puede ver en un mapa la localización del restaurante.
- **Inscripción:** se selecciona un evento y se inscribe uno pidiendo un número máximo de entradas. Cuando se le da a comprar se creará la inscripción como recibida. Si se puede hacer (aforo, entradas disponibles, ...) se le mandará un email de confirmación al email del asistente (si es en pdf mejor).
- **Acerca De:** información sobre la empresa de eventos. Un teléfono de contacto, etc. Es una página estática.
- **Pagos:** se deja como “opcional” implementar los pagos con Laravel Cashier o Laravel Stripe.

INTRANET

- **Asistente:** ver todos, eliminar (solo acceso de admin)
- **Eventos:** ver todos, añadir, modificar, eliminar o cancelar. Ver inscripciones, y cancelar una inscripción si se quiere.
- **Categoría:** añadir, eliminar



- **Inscripciones:** ver todos, eliminar, cancelar

AUTENTICACIÓN

- Tendréis que hacer un login para la web principal (sólo pueden entrar asistentes) y otro login para el portal de la Intranet. Sólo los roles con permiso podrán entrar en la Intranet.
- Debéis proteger las rutas de la web principal y de la intranet, agrupándolas, de modo que sólo los asistentes entren en la web principal, y el resto en la intranet.
- Cuando un usuario de un rol se logué le saldrán sólo las opciones de menú de la intranet que pueda realizar. Se deja como opcional proteger las rutas que no pueda acceder.
- El usuario con rol administrador podrá verlo todo.

API

- Diseñar una API de consulta de información y para hacer inscripciones.
- La API deberá devolver siempre información en Json, tanto cuando se devuelvan datos, como cuando se devuelvan mensajes (“realizado correctamente”).
- Deberá ir protegida con Token.
- Verbos a implementar:
 - o GET /eventia/evento: todos los eventos
 - o GET /eventia/evento/{id}: info de un evento
 - o GET /eventia/categorías: todas las categorías de eventos
 - o GET /eventia/asistente/{dni}: información de un asistente
 - o GET /eventia/asistente/{dni}/inscripciones: todas sus inscripciones
 - o GET /eventia/asistente/{dni}/inscripcion/{id}: información de una inscripción de un asistente
 - o PUT /eventia/evento: crea un evento
 - o DELETE /eventia/evento/{id}: borra un evento e inscripciones asociadas.

MEJORAS

- Mostrar los eventos geolocalizados en un mapa.
- Diseñar una batería de test para cada una de las clases y mostrar capturas de las pruebas.

PRODUCTO A ENTREGAR

1. Repositorio independiente Github, desde el principio, que se vean los cambios realizados desde el primer día (commits, ...).
2. Entrega del diseño relacional (diagrama relacional con las clases)



3. Plantillas html con diseño visual previo. Únicamente el diseño de la web principal y de la intranet, sus layouts (estructura, colores, fonts, ...). Deben ser diseños muy diferentes.
4. Despliegue en Heroku o similar de la aplicación.
5. Elementos evaluables: paginación de resultados, filtros de búsqueda, no errores, logging, tests, api correcto, etc.