



Práctica 3 UF3. StuEvento APP Back Barcelona.

Enunciado

eventbrite

Buscar eventos

En esta práctica, deberás crear un sistema de registro y autenticación para una aplicación que gestiona eventos. Habrá que emplear por tanto la librería **Auth y el ORM Eloquent**.

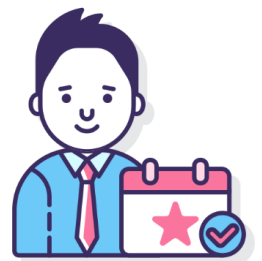


Laravel deberá permitir a los usuarios crear eventos, registrar asistentes. Para ello, deberás crear varias entidades y definir varias relaciones entre ellas.

Deberás cargar las **migraciones que están en el fichero migrations_events_P3UF3.zip**.

Creación de Modelos-Entidades

1. Crea una entidad "**User**" que extienda la clase "Authenticatable" de Laravel.
 - Este modelo se usará para almacenar los datos de los usuarios que se registren en la aplicación.
2. Crea una entidad "**Event**" que represente un evento que los usuarios pueden crear.
 - Este modelo deberá tener una relación "**belongsTo**" con el modelo "**User**", ya que cada evento será creado por un usuario.
3. Crea una entidad "**UserEventsAttendee**" que represente a un asistente registrado en un evento. Esta tabla intermedia relaciona User y Event.
 - Debe tener una relación "**belongsTo**" con el modelo "User" (representando al usuario que registró al asistente)
 - Y una relación "**belongsTo**" con el modelo "Event" (representando al evento al que el asistente está registrado).
4. Define los métodos necesarios en cada modelo para permitir la creación, lectura, actualización y eliminación de registros. Utiliza Eloquent para definir las relaciones entre los modelos.





Métodos de "EventController"

Se deberá crear un controlador "EventController" que permita a los usuarios crear eventos y gestionar asistentes a los mismos. Deberá incluir los siguientes métodos y endpoints:

- **GET /events (1 punto)**
 - **BODY**
 - Sin parámetros
 - **Método asociado**
 - index()
 - **Descripción**
 - Este método debe mostrar una lista de todos los eventos creados por el usuario autenticado.
- **GET /events/create (1 punto)**
 - **BODY**
 - Sin parámetros.
 - **Método**
 - create()
 - **Descripción**
 - Este método debe mostrar un formulario para crear un nuevo evento.
- **POST /events (2 puntos)**
 - **BODY**
 - Evento a crear
 - **Método**
 - store()
 - **Descripción**
 - Este método debe procesar el formulario de creación de eventos y crear un nuevo registro en la base de datos.
- **GET /events/{id} (1 punto)**
 - **BODY**
 - Sin parámetros
 - **Método asociado**
 - show(\$id)
 - **Descripción**
 - Este método debe mostrar los detalles de un evento específico, incluyendo la lista de asistentes.



- **GET /events/{id}/register (1 punto)**
 - **BODY**
 - Sin parámetros
 - **Método asociado**
 - register(\$id)
 - **Descripción**
 - Este método debe mostrar un formulario para registrar un nuevo asistente en un evento específico.
- **POST /events/{id}/attendees (2 puntos)**
 - **BODY**
 - Asistente que se registra en el evento (sin ID ya que lo recibe la URL)
 - La información vendrá dada por el formulario que permite añadir asistentes a un evento.
 - Método **register(\$id)**.
 - **Método asociado**
 - storeAttendee(\$idEvent).
 - **Descripción**
 - Este método debe procesar el formulario de registro de asistentes y crear un nuevo registro en la base de datos.
- **GET /events/{id}/edit (1 punto)**
 - **BODY**
 - No se pasa nada
 - **Método asociado**
 - edit(\$id)
 - **Descripción**
 - Este método debe mostrar un formulario para editar los detalles de un evento.



- **PUT /events/{id} (1 punto)**
 - **BODY:** Evento con datos actualizados (sin ID ya que lo recibe la URL).
 - La información vendrá dada tras completar el formulario derivado de la edición de un evento.
 - Método **edit(\$id)**.
 - **Método asociado**
 - update(\$id)
 - **Descripción**
 - Este método debe procesar el formulario de edición de eventos y actualizar el registro correspondiente en la base de datos.
 - Recibiendo datos y cualquier otro archivo necesario para ejecutar la aplicación.

Entrega. Buenas prácticas y GitLab-Github



Además, se espera que el código siga las buenas prácticas de programación, la escritura de código legible y el uso de comentarios para documentar el código.

En resumen, deberás crear una aplicación que...

1. Permita a los usuarios registrar y autenticarse, crear eventos y registrar asistentes.
 - Para ello, deberás **crear** varias **entidades** y **definir relaciones** entre ellas, así como crear un controlador para gestionar todas las acciones necesarias.
2. Finalmente, asegúrate de que tu código esté bien organizado, limpio y documentado, y sube el proyecto a un repositorio de **GitLab o Github** para su entrega.



Extra – Aplicar GitFlow y Etiquetas (1 punto extra)

Crea una rama feature por cada desarrollo de endpoint y una vez finalizado, fúndela cada rama con la rama **develop**, cada 2 endpoints desarrollados tendrán una versión etiquetada **v0.x**.

Por ejemplo: v0.1 se corresponderá con GET /events y GET /events/create. V0.2 se corresponderá con POST /events y GET /events/{id}...

Cuando se desarrollen todos los endpoints y esté la práctica realizada tendremos la V1.0.

