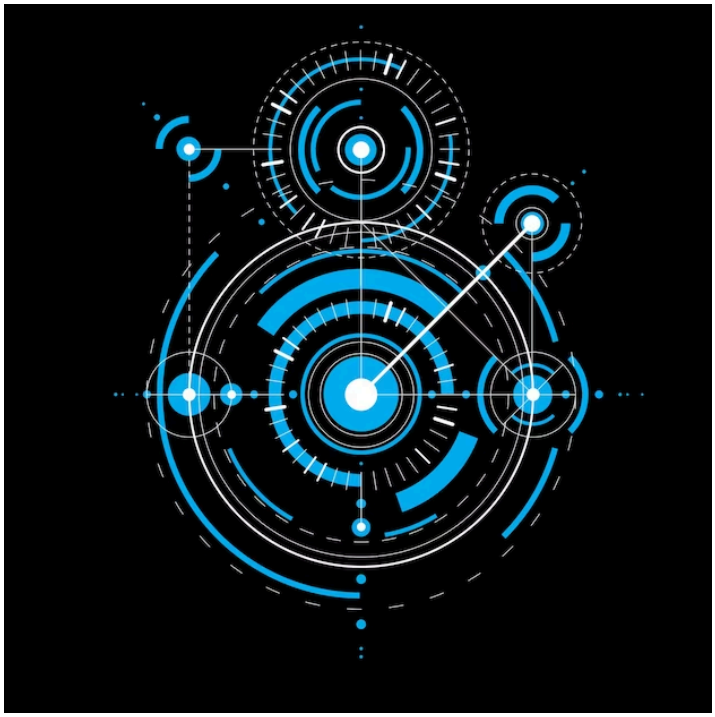
Diseño técnico del sistema Clase Velada

Joel Acoran Cruz Morales



DTS Clase Velada Joel A Cruz Morales

Documento para el diseño técnico del sistema de la clase velada.

Fecha de versión: 15/04/2024

Historial de revisiones

Fecha	Descripción	Autor
05/04/2024	Creación del documento y documentación de la clase	Joel A. Cruz Morales
10/04/2024	Implementación de Arquitectura de la solución e infraestructura utilizada	Joel A. Cruz Morales
15/04/2024	Última versión actualizada y terminada del documento	Joel A. Cruz Morales

1. Arquitectura de la solución

La clase Velada, se trata de una clase desarrollada en el Framework de Laravel usando PHP, JavaScript y conexiones a una base de datos mediante una api. La cual maneja los datos y es con la que tiene interacción directa.

El objetivo de esta clase es organizar de una manera óptima las peleas, con su ubicación y participantes.

Con una estructura de CRUD implementada en el proyecto, con control de errores y validaciones de datos.

1.1. Dependencias

Tiene dependencias de las clases Localización y Pelea.

1.1.1. Librerías

No aplica

1.2. Entornos

Principalmente se desarrolla en un entorno local, está alojado en Xampp, pero además tiene una versión actualizada en GitHub con el siguiente enlace: https://github.com/Joelcrmo/ProyectoUFC/blob/main/app/Http/Controllers/VeladaDBController.php , utiliza además:

- **Servidor Web:** Apache
- **Base de datos:** MySQL

2. Infraestructura utilizada

2.1. Plataforma Hardware

La infraestructura de hardware utilizada para el desarrollo e implementación del proyecto ha sido realizada mediante un portátil, con recursos suficientes para ejecutar el servidor web y la base de datos de manera local.

2.2. Plataforma Software

La infraestructura de software incluye los siguientes componentes principales:

- **Laravel Framework**: Utilizado como el marco de trabajo principal para el desarrollo de este proyecto.
- **Servidor Web:** Apache para servir la aplicación web.
- **Base de Datos:** Mysql y PostgreSQL para almacenar y administrar los datos de la BBDD y la utilización de la API.

2.3. Configuración / Parametrización

La configuración del entorno de desarrollo y producción para este proyecto, se realiza en archivos diferentes de configuración, en Laravel en el archivo '.env', deberemos configurar nuestra ip , usuario y contraseña de nuestro servidor donde esté alojado la base de datos.

En la configuración de nuestra BBDD, deberemos configurar nuestro puerto y la configuración óptima para nuestro servidor en 'my.ini'

3. NameSpaces Clase Velada

3.1. Class Velada

Mediante el NameSpaces VeladaDBController, podremos acceder a nuestra clase Velada.

3.1.1. Constructores

La clase Velada es parte del modelo de datos del proyecto y se utiliza para representar una velada específica. Su constructor puede incluir propiedades y métodos relevantes para la manipulación de datos de veladas.

Así como devolver todas las peleas que existen asociadas a esa velada, así como la creación de las mismas.

3.1.2. Métodos

public function obtenerVeladas()
- **Descripción:** Método que devuelve todas las veladas con información de su localización. - **Parámetros:** Ninguno - **Retorno:** Retorna un JSON con todas las veladas y sus respectivas localizaciones.

public function obtenerVeladaPorID($ID_Velada)

- **Descripción:** Método que devuelve una velada específica según su ID.
- **Parámetros:** $ID_Velada: (int) El ID de la velada que se desea obtener.
- **Retorno:** Retorna un JSON con los detalles de la velada especificada por su ID.

public function crearVelada(Request $request)

- **Descripción:** Método que crea una nueva velada con los datos proporcionados.
- **Parámetros:** $request: (Request) La solicitud HTTP que contiene los datos de la nueva velada.
- **Retorno:** Redirige a la página de listado de veladas después de crear una nueva velada.

public function eliminarVelada($ID_Velada)

- **Descripción:** Método que elimina una velada según su ID.
- **Parámetros:** $ID_Velada: (int) El ID de la velada que se desea eliminar.
- **Retorno:** Retorna un JSON indicando el éxito de la operación de eliminación.

public function actualizarVeladaEnBD($ID_Velada, $nombreVel, $idLocalizacion, $fechaVel)

- **Descripción:** Método que actualiza una velada existente en la base de datos.
- **Parámetros:**
  $ID_Velada: (int) El ID de la velada que se desea actualizar.
  $nombreVel: (string) El nuevo nombre para la velada.
  $idLocalizacion: (int) El nuevo ID de la localización para la velada.
  $fechaVel: (string) La nueva fecha para la velada.
- **Retorno:** Retorna true si la actualización fue exitosa.

public function edit($ID_Velada)

- **Descripción:** Método que obtiene los detalles de una velada para su edición.
- **Parámetros:** $ID_Velada: (int) El ID de la velada que se desea editar.
- **Retorno:** Retorna un JSON con los detalles de la velada especificada por su ID, o un mensaje de error si la velada no se encuentra o si hay un error al obtenerla.