



PROJECT

Manual de usuario y técnico

MARIGNACIA BELÉN FLORES MUÑOZ

2º Desarrollo de Aplicaciones Web

Índice

Introducción.....	2
Manual de usuario.....	2
Registro e inicio de sesión.....	2
Funcionamiento de agenda y eventos	3
Configurar y personalizar usuario	5
Pie de página	5
Manual técnico.....	6
Software y versiones requeridas.....	6
Esquema E/R	6
Base de datos	7
Funcionamiento general de la página	8
Conclusión.....	10

Introducción

MOA Project es el desarrollo y la implementación de la herramienta MOA, *My Organizational Agenda* además de su manual de usuario y documentación.

MOA es una agenda virtual que te permite la organización de eventos según día y hora con soporte para múltiples usuarios a través de un sistema de registro y login.

El proyecto está basado en Laravel, usando MySQL como soporte para la información. La web se conecta a la base de datos a través de PHP y se organiza la información en tablas a través de migraciones de Laravel.

Manual del usuario

1. Registro e Inicio de sesión.

Al acceder al sitio web por primera vez nos encontraremos con la página de bienvenida, dónde nos solicitará nuestras credenciales de inicio de sesión. En caso de que no tengamos una cuenta, podremos acceder al formulario de registro mediante el enlace “Regístrate” en la parte inferior.



My organizational agenda

A screenshot of a web login form. The form has a title 'Iniciar sesión' at the top. Below it are two input fields: 'Correo electrónico' and 'Contraseña'. The 'Correo electrónico' field is currently active, showing a cursor. Below the fields is a blue button labeled 'Iniciar sesión'. At the bottom of the form, there is a link that says '¿No tienes una cuenta? [Regístrate](#)'.


Una vez accedamos al formulario de registro nos solicitará los siguientes campos, siendo todos obligatorios.

Crear un usuario

Nombre

Apellidos

Correo electrónico

Fecha de nacimiento 

Contraseña

Confirma contraseña

[Registrarse](#)

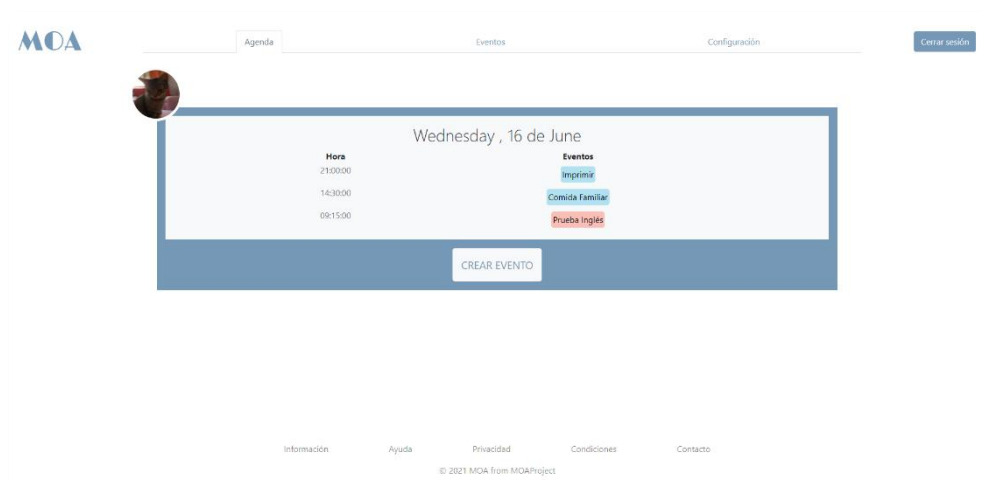
¿Tienes una cuenta? [Entrar](#)

Con todos los campos completados solamente hay que enviar el formulario a través del botón inferior. También disponemos de un enlace para poder volver a la página de inicio de sesión en caso de que sea requerido.

Una vez iniciado sesión, en la barra de navegación se muestra un botón para poder cerrar sesión en cualquier momento.

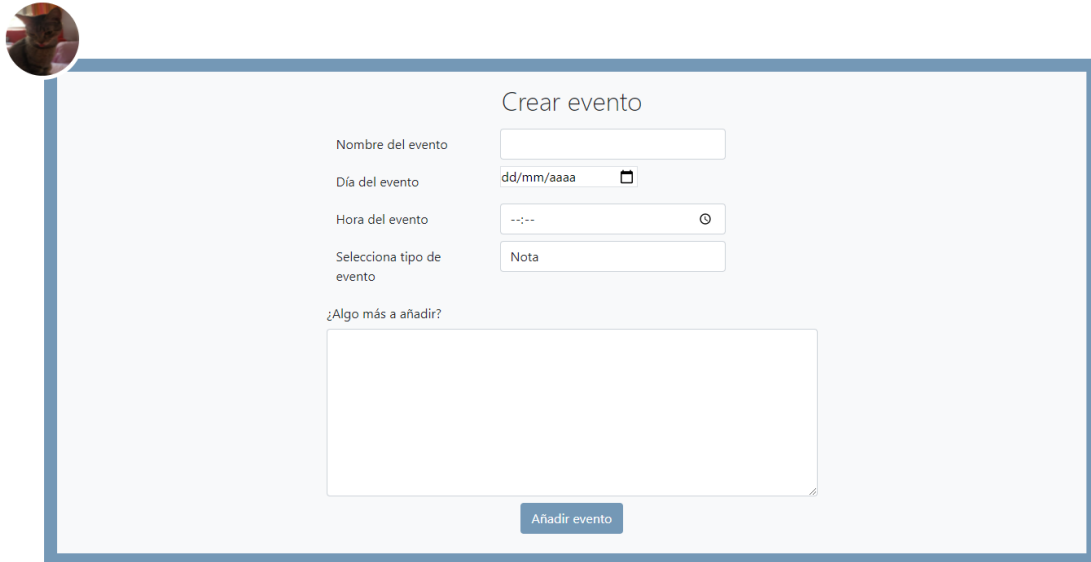
2. Funcionamiento de agenda y eventos.

Al registrarse o al iniciar sesión accedemos a la página home. En esta página se muestra la barra de navegación, nuestra foto de perfil que más adelante podrás modificar y nuestra agenda del día de hoy.



La página home siempre mostrará al entrar la agenda del día actual, en ella se mostrarán todos los eventos que tiene guardado el usuario.

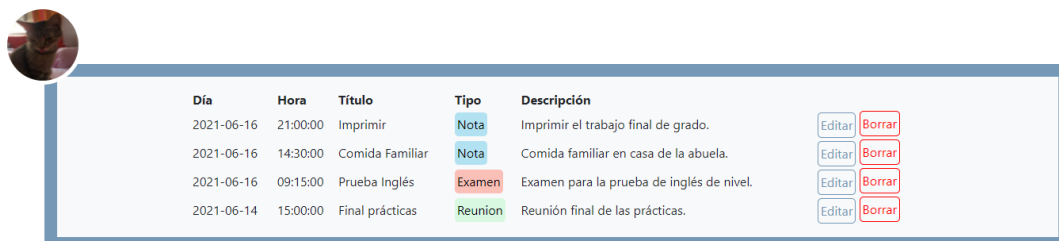
Para crear un evento accedemos al formulario con el botón “CREAR EVENTO”. El formulario que se muestra a continuación nos pide los siguientes campos a rellenar.



Tenemos disponibles cuatro tipos de clasificaciones de eventos a escoger, el predeterminado es “Nota” y dependiendo del que seleccionemos, cada evento de la agenda se mostrará con el color correspondiente.

Al añadir el evento volvemos a la agenda, en caso que no se quiera añadir el evento se puede salir del formulario a través de la barra de navegación.

Dentro de la barra también tenemos el enlace que nos lleva a la página eventos. En esta página se muestran todos los eventos que hemos creado con sus respectivas fechas, descripciones, etc.



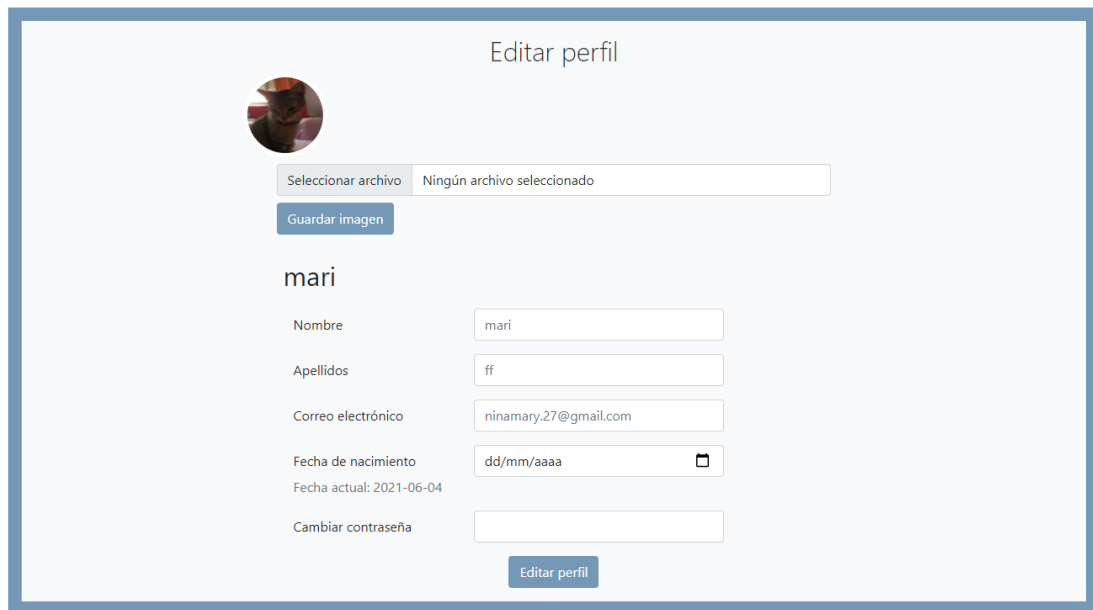
Día	Hora	Título	Tipo	Descripción		
2021-06-16	21:00:00	Imprimir	Nota	Imprimir el trabajo final de grado.	Editar	Borrar
2021-06-16	14:30:00	Comida Familiar	Nota	Comida familiar en casa de la abuela.	Editar	Borrar
2021-06-16	09:15:00	Prueba Inglés	Examen	Examen para la prueba de inglés de nivel.	Editar	Borrar
2021-06-14	15:00:00	Final prácticas	Reunion	Reunión final de las prácticas.	Editar	Borrar

En esta sección de la página se pueden borrar o editar cada uno de los eventos, con los botones correspondientes.

3. Configurar y personalizar usuario.

Nuestro último enlace de la barra de navegación es “Configuración”, este nos lleva a una página donde poder modificar todos los datos de tu usuario, además de poder agregar una imagen de perfil que nos acompañará durante todo el sitio web.

En caso de que alguno de los campos no se quiera modificar bastaría con dejarlo vacío, este se guardará tal y como estaba en la base de datos.



Editar perfil

Seleccionar archivo Ningún archivo seleccionado

Guardar imagen

mari

Nombre mari

Apellidos ff

Correo electrónico ninamary.27@gmail.com

Fecha de nacimiento dd/mm/aaaa

Fecha actual: 2021-06-04

Cambiar contraseña

Editar perfil

Para poder ver que cambios hacer, el mismo formulario te muestra los datos actuales del usuario.

4. Pie de página.

En todas las páginas del proyecto tenemos el “footer” donde se encuentra los enlaces a información de MOA, página de ayuda, condiciones y contacto.

Manual técnico

1. Software y versiones requeridas.

MOA Project ha sido desarrollado principalmente con el framework de PHP Laravel. Para crear la base de datos se ha utilizado MySQL, para crear sus tablas he usado las migraciones de Laravel.

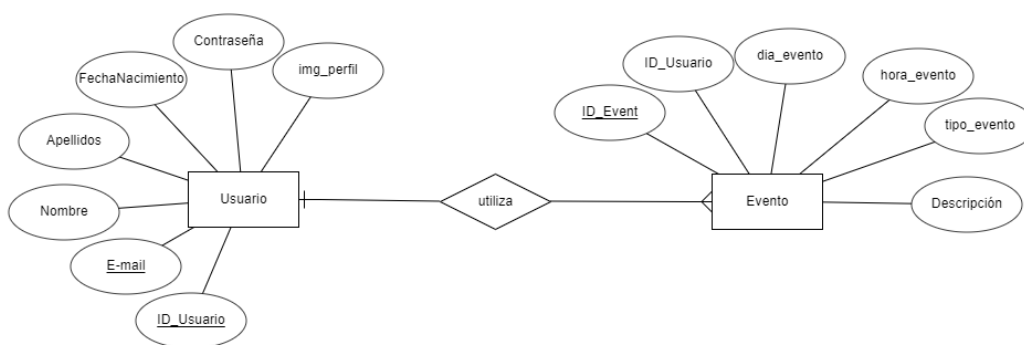
Para el Frontend utilizamos Blade, creando la estructura con etiquetas HTML5, agregando algunas funciones con JavaScript y finalmente dando estilo con CSS3 y con ayuda de la librería Bootstrap v5.0.

Para el Backend, gracias a la facilidad de Laravel, utilizamos PHP v8.0. Para crear la base de datos hemos decidido usar MYSQL.

La página web se ejecuta desde una máquina virtual que contiene un servidor Apache, por lo que lo único necesario es disponer de un navegador que soporte las tecnologías ya mencionadas.

2. Esquema E/R.

Para hacer funcionar la pagina web se ha diseñado el siguiente esquema E/R:



A partir de este esquema se fue creando y adaptando la base de datos a las necesidades del proyecto.

3. Base de datos.

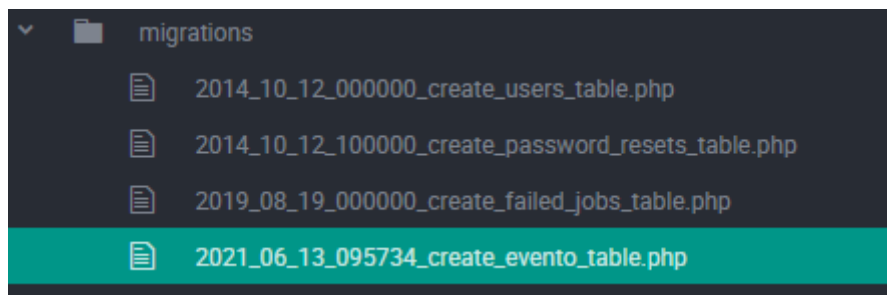
En la máquina virtual con Ubuntu se ha instalado MySQL y se ha creado la base de datos *moaproject*. Para administrarla también se ha creado un usuario con todos los permisos sobre la base de datos.

En el archivo “.env” del proyecto Laravel añadimos las credenciales para poder conectarnos con la base de datos:

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=moaproject
DB_USERNAME=adminserver
DB_PASSWORD=password
```

Se hace la conexión a través de MySQL a la IP de la propia maquina virtual, se utiliza el puerto 3306 para conectar con la base de datos *moaproject* con el usuario *adminserver* y la contraseña *password*.

Una vez creada la base de datos, la creación de las tablas se ejecuta a través de siguientes migraciones:



Instalando el paquete Laravel/ui el framework nos facilita el sistema de autenticación creando las migraciones de user, las vistas que contienen los formularios necesarios, los controladores y modelos y todo con sus respectivas rutas. Por tanto, no ha sido un problema modificarlo para adaptarlo a mi idea principal.

Así es como quedan las tablas de *moaproject*:

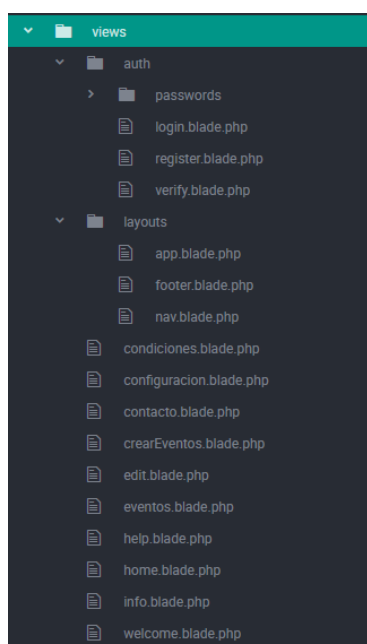
```
public function up()
{
    Schema::create('users', function (Blueprint $table) {
        $table->id();
        $table->string('name');
        $table->string('surname')->nullable();
        $table->string('email')->unique();
        $table->timestamp('email_verified_at')->nullable();
        $table->date('birth')->nullable();
        $table->string('password');
        $table->string('img_perfil')->nullable();
        $table->rememberToken();
        $table->timestamps();
    });
}
```

```
public function up()
{
    Schema::create('evento', function (Blueprint $table) {
        $table->id();
        $table->foreignId('user_ID')->references('id')->on('users');
        $table->date('agenda_dia');
        $table->time('hora_evento');
        $table->string('titulo');
        $table->string('tipo');
        $table->string('descripcion')->nullable(true);
        $table->timestamps();
    });
}
```


4. Funcionamiento general de la página.

El funcionamiento de la página, gracias a Laravel, se puede ver resumido en sus rutas en el archivo web.php:

```
web.php
1  <?php
2
3  use Illuminate\Support\Facades\Route;
4  use App\Http\Controllers\EventsController;
5  use App\Http\Controllers\UserController;
6  use App\Http\Controllers\ImageUploadController;
7
8  /*Welcome página que contiene la página login y link a la página de registro.*/
9  Route::view('/', 'welcome')->name('welcome');
10
11 /*Pagina de creación de evento, el formulario envia los datos a la función save() del controlador*/
12 Route::view('/crearEvento', 'crearEventos')->name('crearEventos');
13 Route::post('/eventos', [EventsController::class, 'save'])->name('create_event');
14 /*Obtenemos desde la ruta el evento para enviarlo a la función edit(),
15 desde la pagina edit se envian Los cambios a la función update()*/
16 Route::get('/eventos/{agenda}/editar', [EventsController::class, 'edit'])->name('edit_event');
17 Route::patch('/eventos/{agenda}', [EventsController::class, 'update'])->name('update_event');
18 /*Llamamos a la función show(), para obtener todos los eventos del usuario, en la pagina
19 eventos tenemos la opción de borrar y editar cada evento*/
20 Route::get('/eventos', [EventsController::class, 'show'])->name('events');
21 Route::delete('/eventos/{agenda}', [EventsController::class, 'drop'])->name('drop_event');
22 /*Pagina con formulario para actualizar datos que se envian a la función update() del controlador de usuario*/
23 Route::view('/perfil', 'configuracion')->name('configuracion');
24 Route::post('/perfil', [UserController::class, 'update'])->name('update_user');
25 /*Rutas para guardar en la carpeta public/images/ Las imagenes de perfil que se suben en la página de configuración*/
26 Route::get('image-upload', [ImageUploadController::class, 'imageUpload'])->name('image.upload');
27 Route::post('image-upload', [ImageUploadController::class, 'imageUploadPost'])->name('image.upload.post');
28
29 /*Rutas de autenticación del paquete Laravel/ui*/
30 Auth::routes();
31 /*Ruta home, se accede cuando la sesión esta iniciada*/
32 Route::get('/home', [App\Http\Controllers\HomeController::class, 'index'])->name('home');
33
34 /*Rutas de paginas del Footer*/
35 Route::view('/informacion', 'info')->name('info');
36 Route::view('/ayuda', 'help')->name('help');
37 Route::view('/condiciones', 'condiciones')->name('condiciones');
38 Route::view('/contacto', 'contacto')->name('contacto');
39
```



Principalmente utilizamos 3 controladores a parte de los que ha creado el paquete de autenticación de Laravel. Cada uno de estos controladores contiene las funciones para crear, editar, borrar y mostrar los datos de *moaproject* con ayuda de un modelo para la tabla users y otro para la tabla evento.

Todas las páginas están creadas con Blade para poder implementar fácilmente Bootstrap y css a través de plantillas: plantilla principal, barra de navegación y pie de página.

Para procesar la información de la base de datos el usuario debe hacerlo a través de formularios HTML, lo que le permite crear, modificar, borrar y actualizar cada uno de los eventos.

Un ejemplo es el formulario de la página edit, que permite modificar el evento seleccionado en la página Eventos:

```
<form class="" action="{{ route('update_event', $agenda) }}" method="POST">
    @csrf @method('PATCH')

    <!--Nombre del evento-->
    <div class="form-group row m-2">
        <label class="col-md-4 col-form-label text-md-right" for="titulo">Nombre del evento </label>
        <div class="col-md-6">
            <input class="form-control" type="text" name="titulo" value="{{ $agenda->titulo }}">
        </div>
    </div>

    <!--Dia del evento-->
    <div class="form-group row m-2">
        <label for="fecha_inicio" class="col-md-4 col-form-label text-md-right">Día del evento</label>
        <div class="col-md-6">
            <p class="text-secondary">{{ $agenda->agenda_dia }}</p>
            <input class="form-control border" type="date" name="fecha_inicio">
        </div>
    </div>

    <!--Hora del evento-->
    <div class="form-group row m-2">
        <label for="hora" class="col-md-4 col-form-label text-md-right">Hora del evento</label>
        <div class="col-md-6">
            <p class="text-secondary">{{ $agenda->hora_evento }}</p>
            <input class="form-control" type="time" name="hora">
        </div>
    </div>

    <!--Tipos de evento-->
    <div class="form-group row m-2">
        <label class="col-md-4 col-form-label text-md-right" for="tipo">Selecciona tipo de evento</label>
        <div class="col-md-6">
            <p class="text-secondary">{{ $agenda->tipo }}</p>
            <select class="form-control" name="tipo">
                <option value="Nota">Nota</option>
                <option value="Aniversario">Aniversario</option>
                <option value="Reunion">Reunion</option>
                <option value="Examen">Examen</option>
            </select>
        </div>
    </div>

    <!--Descripción del evento-->
    <div class="form-group m-2">
        <label class="col-form-label text-md-right" for="tipo">Editar descripción</label>
        <div class="">
            <textarea class="form-control" name="descripcion" rows="8" cols="80">{{ $agenda->descripcion }}</textarea>
        </div>
    </div>

    <!--Botón para enviar el formulario-->
    <div class="d-flex justify-content-center m-2">
        <button type="submit" id="boton_moa" class="btn">Editar evento</button>
    </div>
</form>
```

En este formulario organizamos cada apartado del formulario con las clases de Bootstrap para formularios, a cada elemento le indicamos con php, dentro de {{ }}, su característica del evento. Una vez rellenado enviamos el formulario a través del método PATCH, que indicamos al principio con un @method de Laravel y con el token @csrf.

Conclusión

El proyecto surge de la necesidad real de tener una plataforma en la que poder gestionar notas y recordatorios de forma simple y ordenada en formato digital por contra de las agendas tradicionales. Como programadores nos podemos ofuscar y perder la noción del tiempo a la hora de trabajar o corregir errores, por lo que el hecho de tener una agenda digital de fácil acceso que nos recuerde las responsabilidades y eventos que tenemos en nuestro calendario es algo sumamente útil.

El manual de MOA Project se ha escrito durante el desarrollo de la página, por lo tanto, puede que las imágenes adjuntas de ciertos archivos no coincidan con la versión final.

MOA como concepto está pensado para poder añadir más funciones para poder gestionar el día a día del usuario. Se podrían implementar más herramientas para personalizar la agenda, como el poder escoger el color para cada tipo de evento, un apartado donde tener una lista de “cosas que hacer” tachable, otro apartado con una lista de deseos, etc.

Para escalar el proyecto a más funcionalidades sería necesario modificar y revisar parte del código, aunque debido al framework elegido para el proyecto se podría hacer de forma cómoda y sencilla.

Durante la presentación se mostrarán todas las páginas que el usuario puede acceder, así como ejemplos reales de todas las funcionalidades.

Para concluir se plantea el añadido o cambio de alguno de los lenguajes que se ha utilizado para, en un futuro, implementar herramientas de procesamiento de datos, como gráficos de ahorros/gastos, utilizando Phyton.