

ShowTickt

Memoria

Brian Tobias Arrua, Joan Galindo, Camilo Jumelle

Table of Contents

1. Introducció	1
2. Disseny	2
2.1. Diagrama entitat relació	2
2.2. Diagrama per classes	2
2.3. Diagrama per classes	2
2.4. Mockup de interfaces	4
2.5. MetaDades y compartició a las redes sociales	4
2.6. Instalació del servidor	4
2.7. Configuració de la aplicació	5
3. Confecció del manual de usuario integrado en la aplicació	10
3.1. Aplicaciones necesarias	10
3.2. Guía de la instalación	10
4. Etiquetas utilizadas	12
5. Linias futuras	13
6. Conclusiones	14
7. Glosario	15

Chapter 1. Introducción

Los miembros del proyecto somos Brian Tobias Arrua, Galindo Palacio Joan y Camilo Jumelle. Este proyecto tratará de realizar una aplicación que venderá entradas para eventos de varios tipos, este proyecto incluye una sección de promotores donde el promotor deberá ir añadiendo y administrando los eventos a vender.

Chapter 2. Diseño

2.1. Diagrama entitat relacio

La siguiente imagen es el primer diseño de la base de datos

[primerDiagrama] | *images/primerDiagrama.png*

Esta imagen es un rediseño del primer diagrama

Hicimos este cambio principalmente por que en el primero no especificamos cual era la tabla fuerte y la tabla debil en "Esdeveniment, categoria y recinte"

[segonDiagrama] | *images/segonDiagrama.png*

Este es el Diagrama Actualizado

Añadimos las tablas de usuario, y colocamos bien sus atributos.

[quartDiagrama] | *images/quartDiagrama.png*

2.2. Diagrama per classes

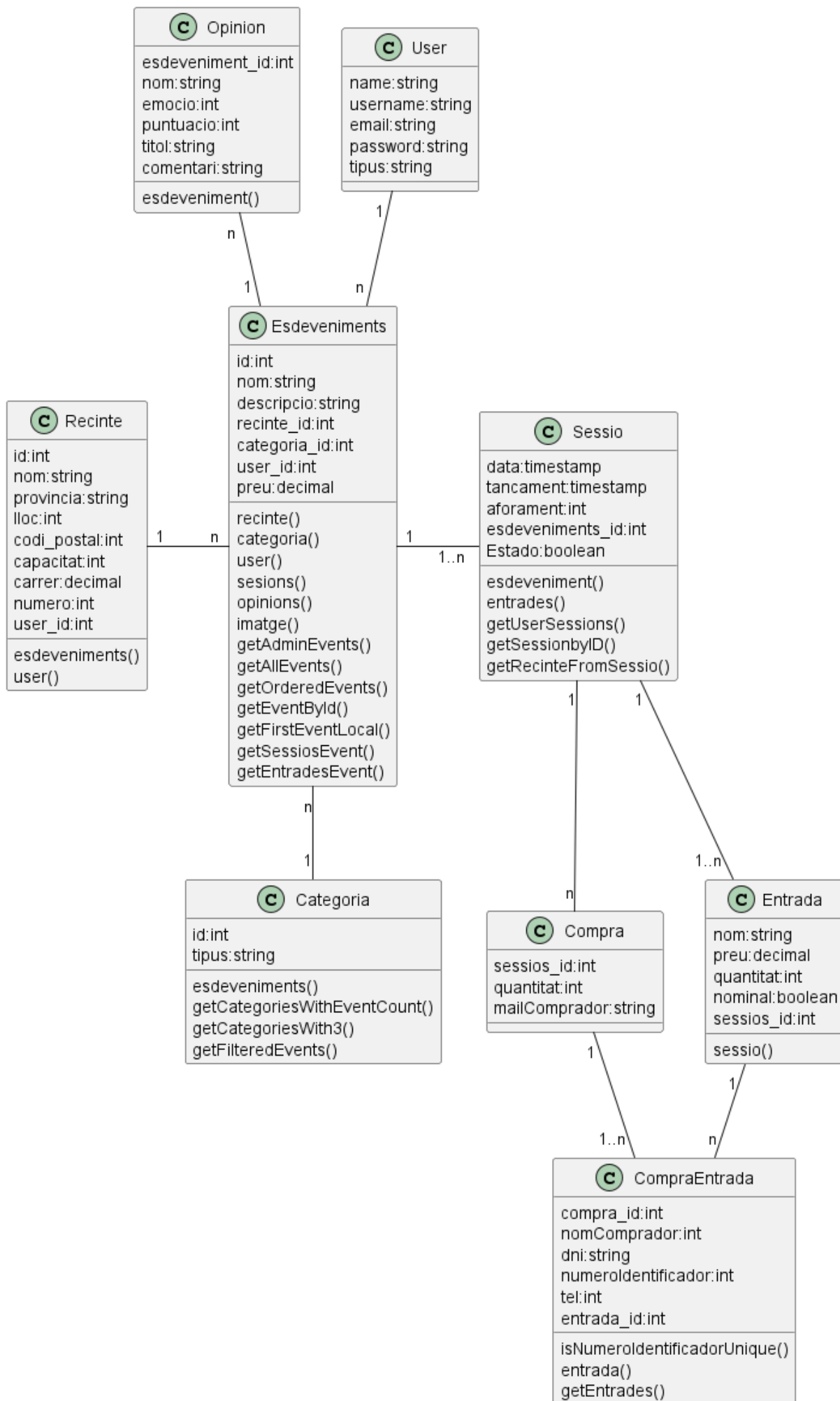
Este es el Diagrama Actualizado

Añadimos las tablas de opinion para guarda las opiniones de los clientes y la de compra, Compra_Entradas

[UltimoDiagrama] | *images/UltimoDiagrama.PNG*

2.3. Diagrama per classes

Este es el diagrama definitivo de la base de datos modelo por clases



2.4. Mockup de interfaces

[Link al figma](#)

2.5. MetaDades y compartición a las redes sociales

Para poder controlar el compartir de las páginas de nuestra aplicación hemos añadido un meta de descripción para que se coloque una breve descripción de lo que la página realiza o muestra, y si la página es muestra algún evento a vender o demás mostrara también la imagen principal de este evento.

```
<meta name="description" content="descripción de la página" >
<meta name="image" content="imagen de evento si es que la página tiene">
```

2.6. Instalación del servidor

Para poder utilizar la aplicación deberemos de tener instalado lo siguiente

- GIT
- Composer
- Apache
- PHP
- Docker

2.6.1. Instalación apache, git y php

```
sudo apt-get install git php apache2 -y
```

2.6.2. Instalación composer

```
apt install git curl php-cli unzip

curl -sS https://getcomposer.org/installer -o /tmp/composer-setup.php

HASH=`curl -sS https://composer.github.io/installer.sig`

php -r "if (hash_file('SHA384', '/tmp/composer-setup.php') === '$HASH') { echo
'Installer verified'; } else { echo 'Installer corrupt'; unlink('composer-setup.php');
} echo PHP_EOL;"

# Deberá de salir 'Installer verified' si se instalo correctamente

sudo php /tmp/composer-setup.php --install-dir=/usr/local/bin --filename=composer
```

```
composer #Para saber que se instalo correctamente
```

2.6.3. Instalación Docker

```
sudo apt-get update

sudo apt-get install ca-certificates curl gnupg

sudo install -m 0755 -d /etc/apt/keyrings

curl -fsSL https://download.docker.com/linux/debian/gpg | sudo gpg --dearmor -o
/etc/apt/keyrings/docker.gpg

sudo chmod a+r /etc/apt/keyrings/docker.gpg

echo \
"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]
https://download.docker.com/linux/debian \
$(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin
docker-compose-plugin
```

2.7. Configuración de la aplicación

Primero clonamos el repositorio con la app

```
cd /var/www/html/
sudo git clone https://git.copernic.cat/arrua.dominguez.brian.tobias/gr6-arrua-
galindo-jumelle.git
```

Luego deberás de configurar el apache con el directorio de la carpeta de la app

```
sudo nano /etc/apache2/sites-available/laravel.conf
```

```
<VirtualHost *:80>
    ServerAdmin isard@localhost
    ServerName 10.2.115.24 #ip o dirección web
    DocumentRoot /var/www/html/gr6-arrua-galindo-jumelle/site/public/ #Esta es la
carpeta (Tenés que poner la public)

    <Directory /var/www/html/gr6-arrua-galindo-jumelle/site/public>
```

```

        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
    </Directory>

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

```

Activamos la configuración creada y desactivamos la que viene por defecto

```

sudo a2ensite laravel
sudo a2dissite 000-default.conf
sudo service apache2 restart

```

2.7.1. HTTPS Configuration

Para realizar la redirección de http a https se deben realizar los siguientes comandos

```

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout
/etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

```

Al ejecutarlo verán que pide que completen una serie de preguntas, la única importante es la de 'Common Name' donde deberán de colocar su ip del servidor.

```

sudo nano /etc/apache2/conf-available/ssl-params.conf
sudo cp /etc/apache2/sites-available/default-ssl.conf /etc/apache2/sites-
available/default-ssl.conf.bak
sudo nano /etc/apache2/sites-available/default-ssl.conf

```

En el archivo que se abre deberán de colocar lo siguiente

```

<IfModule mod_ssl.c>
    <VirtualHost _default_:443>
        ServerAdmin webmaster@localhost #Acá colocaran lo que colocaron en
        email adress con la primera comanda

        DocumentRoot /var/www/html

        ErrorLog ${APACHE_LOG_DIR}/error.log
        CustomLog ${APACHE_LOG_DIR}/access.log combined

        SSLEngine on

        SSLCertificateFile      /etc/ssl/certs/ssl-cert-snakeoil.pem
        SSLCertificateKeyFile   /etc/ssl/private/ssl-cert-snakeoil.key
    </VirtualHost>
</IfModule>

```



```

        <FilesMatch "\.(cgi|shtml|phtml|php)$">
            SSLOptions +StdEnvVars
        </FilesMatch>
        <Directory /usr/lib/cgi-bin>
            SSLOptions +StdEnvVars
        </Directory>

    </VirtualHost>
</IfModule>

```

Tamibién deben de cambiar/crear el siguiente fichero

```
sudo nano /etc/apache2/conf-available/ssl-params.conf
```

Donde colocarán lo siguiente:

```

SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
SSLProtocol All -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
SSLHonorCipherOrder On
# Disable preloading HSTS for now.  You can use the commented out header line that
# includes
# the "preload" directive if you understand the implications.
# Header always set Strict-Transport-Security "max-age=63072000; includeSubDomains;
# preload"
Header always set X-Frame-Options DENY
Header always set X-Content-Type-Options nosniff
# Requires Apache >= 2.4
SSLCompression off
SSLUseStapling on
SSLStaplingCache "shmcb:logs/stapling-cache(150000)"
# Requires Apache >= 2.4.11
SSLSessionTickets Off

```

Después configuramos el mismo archivo donde configuramos el apache con anterioridad

```
sudo nano /etc/apache2/sites-available/laravel.conf
```

En el archivo pondran la siguiente linea

```

<VirtualHost *:80>
    . . .

    Redirect "/" "https://your_domain_or_IP/"

    . . .

```

```
</VirtualHost>
```

A posteriori realizaremos las siguientes comandas

```
sudo ufw app list #Si sale que no conoce la comanda realiza antes 'sudo apt install ufw'
sudo ufw status #Si sale inactive realiza 'sudo ufw enable'
sudo ufw allow 'WWW Full'
sudo ufw delete allow 'WWW'
sudo ufw status #Revise si funciona con la siguiente comanda
```

Con la última comanda deberá de salir:

```
Output
Status: active
```

To	Action	From
--	-----	----
OpenSSH	ALLOW	Anywhere
WWW Full	ALLOW	Anywhere
OpenSSH (v6)	ALLOW	Anywhere (v6)
WWW Full (v6)	ALLOW	Anywhere (v6)

Y por último debemos activar lo siguiente:

```
sudo a2enmod ssl
sudo a2enmod headers
sudo a2ensite default-ssl
sudo a2enconf ssl-params
sudo apache2ctl configtest #Si salio bien saldrá 'Syntax OK'
sudo systemctl restart apache2
```

Para aplicar los cambios:

```
sudo systemctl restart apache2
```

Ahora seguimos configurando la app, en la carpeta de site deberás de ejecutar los siguientes comandos de **Composer**

```
cd gr6-arrua-galindo-jumelle/site
sudo composer install
sudo cp .env.example .env
sudo php artisan key:generate
sudo npm install fullcalendar fullcalendar-scheduler #Si no encuentra el comando
```

```
realiza 'sudo apt install npm'
```

```
cd docker/  
sudo docker compose up -d
```

Al ya tener el docker/bd, se debe de colocar los puertos, nombre de la bd y demás en el archivo **.env**, si no tocase nada del docker el archivo estará correcto al copiarlo.

También deberás de añadir el grupo www-data a los permisos de la carpeta, como grupo propietario además de otorgar permisos de escritura para grupos

```
sudo chown -R www-data:www-data site/  
sudo chmod -R 775 site/  
cd site/  
sudo php artisan migrate  
sudo php artisan db:seed --class=CategoriesTableSeeder
```

Con todo esto, si no se ha encontrado con errores, uno ya debería de poder usar la aplicación con libertad.

Chapter 3. Confección del manual de usuario integrado en la aplicación

Sigue estos pasos para instalar y configurar correctamente nuestra aplicación.

3.1. Aplicaciones necesarias

Asegúrate de contar con las siguientes herramientas instaladas en tu sistema antes de proceder con la instalación de nuestra aplicación:

- GIT
- PHP (recomendamos la v8.2)
- Composer
- Docker
- NodeJS

En el archivo php.ini descomentar las extensiones:

- extension=gd
- extension=pgsql

3.2. Guía de la instalación

3.2.1. Clonado del repositorio y configuración inicial

Realice el git clone con el siguiente comando

```
git clone https://git.copernic.cat/arrua.dominguez.brian.tobias/gr6-arrua-galindo-jumelle.git
```

Luego de que se haya clonado sin problemas colóquese en la carpeta site y realice lo siguiente

```
composer install  
copy .env.example .env  
php artisan key:generate  
npm install fullcalendar fullcalendar-scheduler
```

3.2.2. Configuración de Docker

A posteriori colóquese en la carpeta docker para poder realizar:

```
cd ../docker
```

```
docker composer up
```

3.2.3. Configuración de la base de datos

Si se creó el contenedor sin problemas ya puede empezar a crear la base de datos necesaria para la app

```
cd ../site  
php artisan migrate  
php artisan db:seed --class=CategoriesTableSeeder
```

Con esto la base de datos se habrá creado correctamente y con ello creado las tablas necesarias, y con el último comando habrás creado unas columnas de categorías para tus eventos.

3.2.4. Creación de usuarios (opcional)

Si deseas tener unos usuarios, como por el momento no hay manera de registrarse, se pueden crear con

```
php artisan db:seed --class=UsersTableSeeder
```

3.2.5. Listo para usar

Ahora la aplicación está lista para ser utilizada. Accede a ella desde tu navegador web con

```
php artisan serve
```

y comienza a disfrutar de sus funcionalidades.

Chapter 4. Etiquetas utilizadas

- Descripcion: Hemos utilizado esta etiqueta para hacer una breve descripcion en la pagina que te encuentras.
- Imagen: Esta etiqueta se utiliza en los diferentes eventos, ya que tienen imagen y lo utilizamos como imagen promocional

Chapter 5. Linias futuras

Para futuro queremos finalizar tanto la api de imagenes, la cual le falta verse las imagenes, se guardan en la bd pero no se muestran correctamente, como la api de validación de entradas que nos faltaria su totalidad.

Chapter 6. Conclusiones

Con este proyecto se aprendió a planificar tareas, diseñar y desarrollar funcionalidades de una aplicación web, trabajar en equipo, resolver problemas técnicos, y mejorar el diseño y la usabilidad del proyecto a lo largo de múltiples iteraciones.

Chapter 7. Glosario

<https://laravel.com/docs>

<https://www.nigmacode.com>

<https://stackoverflow.com>

<https://laracasts.com/discuss>

<https://www.php.net>

<https://developer.mozilla.org/en-US/>

<https://w3schools.com>

<https://www.digitalocean.com>

<https://robindirksen.com/blog/>