

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. Sketching	4
2.4. Mockup de interfaces	4
2.5. Guia de estils	4
2.6. MetaDades y compartició a las redes sociales	4
3. Confecció del manual de instal·lació/distribució de la aplicació	6
4. Isard	7
4.1. Instal·lació del servidor	7
4.2. Configuració de la aplicació	8
5. Confecció del manual de usuari integrat en la aplicació	10
6. Etiquetas utilizadas	11
7. Conclusiones	12
8. Glosario	13
9. Presentación del proyecto	14

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\BD\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\BD\segonDiagrama.png*

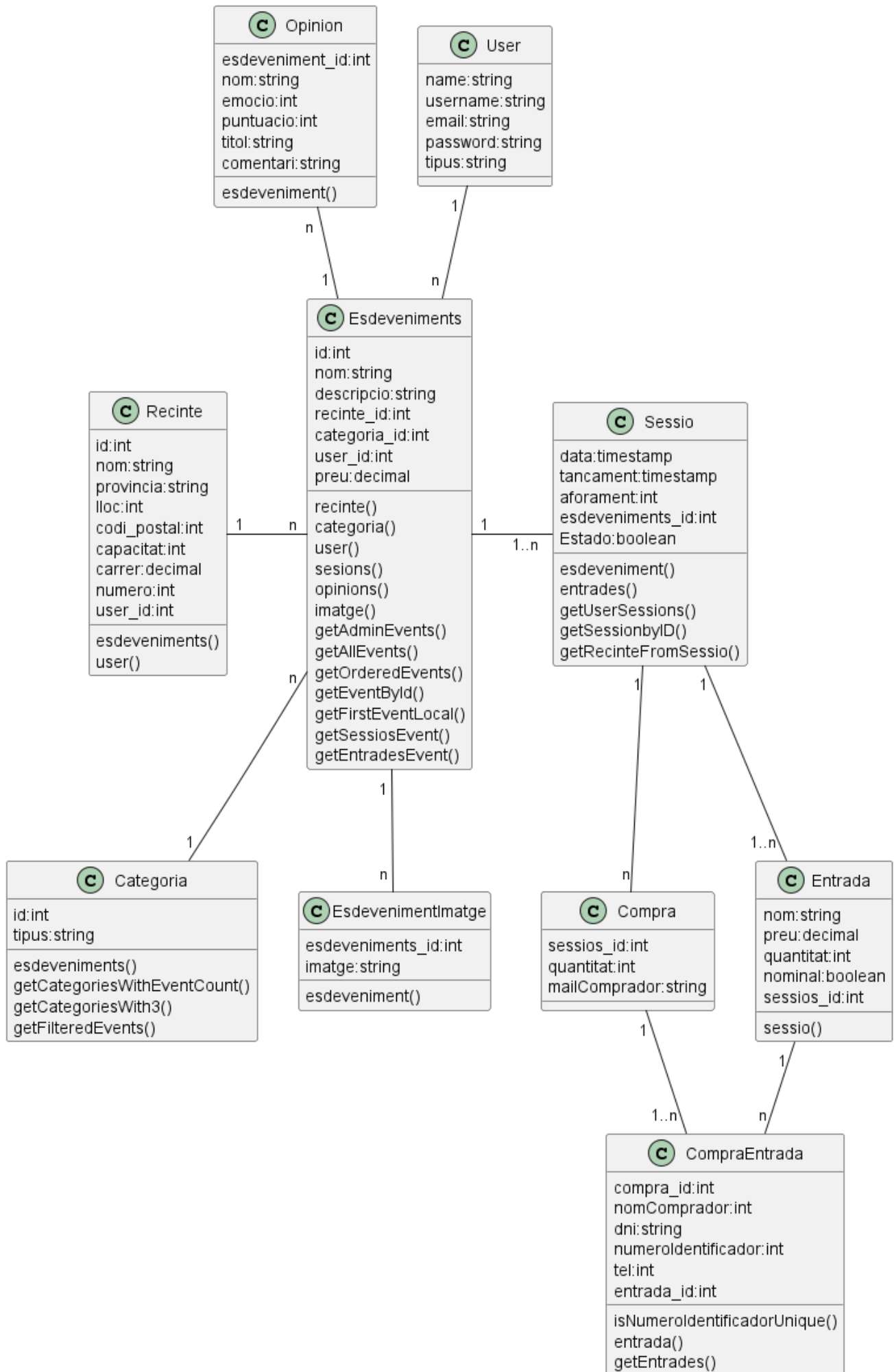
Este es el Diagrama Actualizado

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

[quartDiagrama] | *images\BD\quartDiagrama.png*

2.2. Diagrama per classes

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



Explicación de las clases

- Esdeveniment: La tabla de esdeveniments es la tabla que nos guarda los diferentes eventos que tenemos en la pagina, donde guardamos el nombre, el día, la imagen y el precio
- Categoria: En la tabla de categoria hemos decidido guardar las diferentes categoria de los events
- Recinte: En la tabla de recinte hemos decidido guardar los recintos de los eventos
- Sessio:
- Tipus Entrada:
- Compra:
- Entrada:

2.3. Sketching

2.4. Mockup de interfaces

2.5. Guía de estilos

Para la página seguimos elegimos la siguiente **paleta de colores** , hemos decidido estos ya que llaman la atención y el contraste con el negro (que sera el color del texto) no obliga al usuario a forzar la vista.

[paletaDeColores] | *images\GuiaEstils\paletaDeColores.png*

La **tipografía** elegida fue 'Nunito' se selecciono por su facilidad lectora

[tipografia] | *images\GuiaEstils\tipografia.png*

Los **iconos** elegidos son los siguientes:

El icono para busquedas

[iconoBusqueda] | *images\GuiaEstils\iconoBusqueda.png*

Los **botones** serán de este estilo:

[botones] | *images\GuiaEstils\botones.png*

Los **enlaces** serán de la siguiente forma:

[enlaces] | *images\GuiaEstils\enlaces.png*

2.6. MetaDades y compartición a las redes sociales

Para poder controlar el compartir de las páginas de nuestra aplicación hemos añadido un meta de descripcion 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">
```

Chapter 3. Confección del manual de instalación/distribución de la aplicación

Chapter 4. Isard

4.1. Instalación del servidor

Para poder utilizar la aplicación deberemos de tener instalado lo siguiente

- GIT
- Composer
- Apache
- PHP
- Docker

4.1.1. Para instalar composer deberás de realizar los siguientes comandos

```
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
```

4.1.2. Para instalar GIT, PHP y Apache

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

4.1.3. Para instalar 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.gpg]
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
```

4.2. Configuración de la aplicación

Con git se deberá de traer el repositorio con

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

Al tener ya el repositorio descargado, en la carpeta de site deberás de ejecutar los siguientes comandos de **Composer**

```
composer install
cp .env.example .env
php artisan key:generate
npm install fullcalendar fullcalendar-scheduler #Para instalar las API del calendario
que se ve al visualizar los eventos
```

Con esto ya funcionaria el laravel, y para tener la base de datos tendremos que tener docker

```
cd docker/
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 tocaste nada del docker el archivo estará correcto al copiarlo.

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>
```

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

```
chown :www-data site/
chmod -R 775 site/
```

Al tener todo esto ya deberías de poder visualizar la página al reiniciar el apache con el siguiente comando

```
sudo systemctl restart apache2
```

Chapter 5. Confección del manual de usuario integrado en la aplicación

Chapter 6. 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 7. Conclusiones

Chapter 8. Glosario

Chapter 9. Presentación del proyecto