



DISSENYA EL TEU FUTUR

# EXPO JOVE 2025

PALAU DE FIRES / GIRONA  
2, 3, 4 i 5 D'ABRIL

@FiradeGirona  
#Expojove



ORGANITZA



PATROCINA



COL·LABORA



# Diseño y envío de una campaña de marketing digital usando mjml

## Análisis de requerimientos

Expojove se divide en varios grupos de personas que lo componen, los cuales son:

1. **Visitantes:** Como el propio nombre indica, son las personas que visitan el lugar y prueban los productos y actividades que proporcionan las marcas.
2. **Expositores:** Los expositores son los que compran un espacio reservado en la feria de muestras de Valencia, lo que también es llamado stand. Los expositores son los más importantes en este proyecto ya que son ellos a los que les tendré que mandar ese email para informales los servicios que les ofrece ExpoJove.
3. **Organizadores:** Son las personas que se encargan de organizar el evento y planificarlo, los cuales son los trabajadores de la feria de muestras de Valencia.

# D.A.F.O

D

## DEBILIDADES

Yo creo que las debilidades más importantes de este proyecto es que no todos los motores de correo electronicos soportan todas las funciones que puede proporcionar mjml, por lo que hay que hacerlo con cautela.

F

## FORTALEZAS

Las fortalezas de mjml es que proporciona más estabilidad en cuanto a formato responsive y de una manera mas sencilla ya que con el mjml se puede generar automaticamente un documento html.

A

## AMENAZAS

Las amenazas a las que me voy a enfrentar con el email hacia los expositores es que tengo que intentar ofrecerles servicios los cuales no se puedan proporcionar por si mismos, lo cual es el algo complicado.

O

## OPORTUNIDADES

Tengo que aprovechar cualquier oportunidad que me permita vender la máxima cantidad de servicios posibles, para ello tendré que redactar todo de una manera incitante para los expositores.

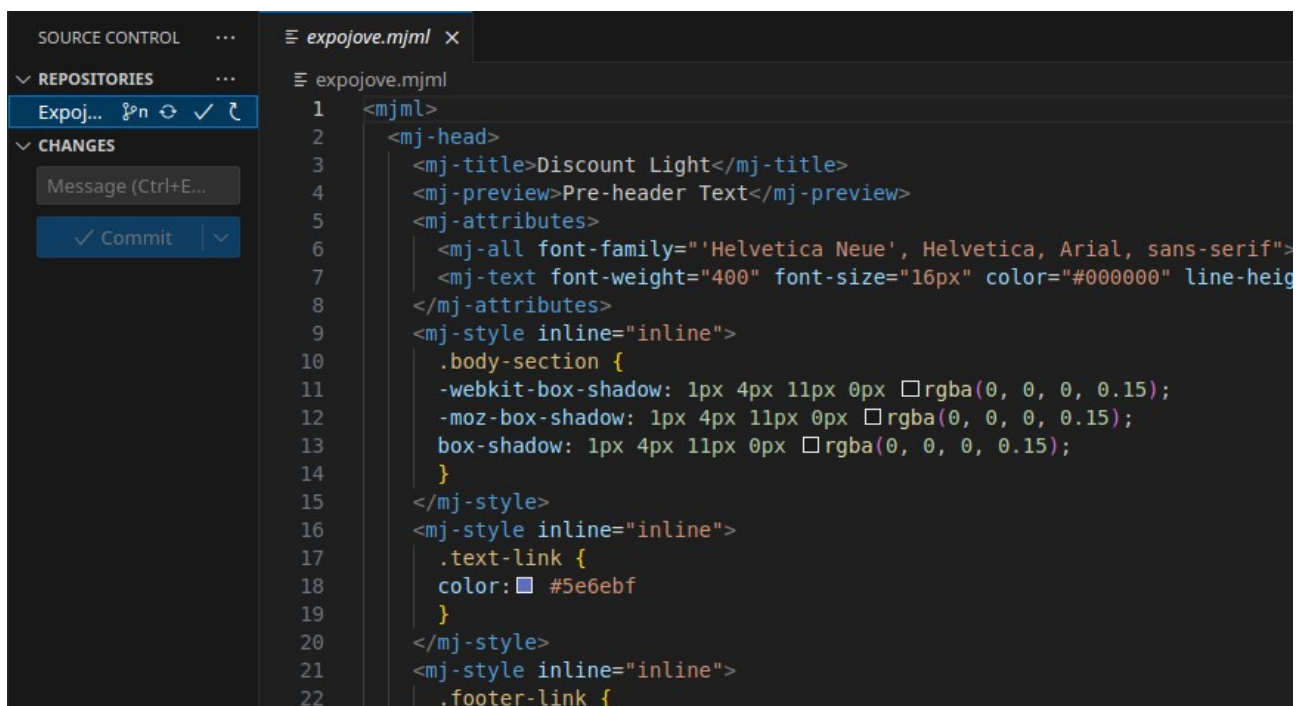
## Objetivos de la campaña

El objetivo de esta campaña es conseguir poder presentar a los expositores los servicios que la feria de muestras le pueda ofrecer, para poder conseguir mas compras y gastos por parte de los expositores.

Para ello se tendrá que hacer de una manera que se les pueda convencer, que tenga que ser creativa, dinámica, visualmente atractiva para los ojos, etc. En este caso es através de un correo electrónico que se les mandaría a todos los expositores con un documento creado con mjml para que pueda ser responsive y se adapte a distintos tipos de dispositivos.

# Git en GitHub

Lo he realizado todo con la organización de ramas y commits para manejarlo todo muy bien y puede volver atrás en el caso de que sea necesario.

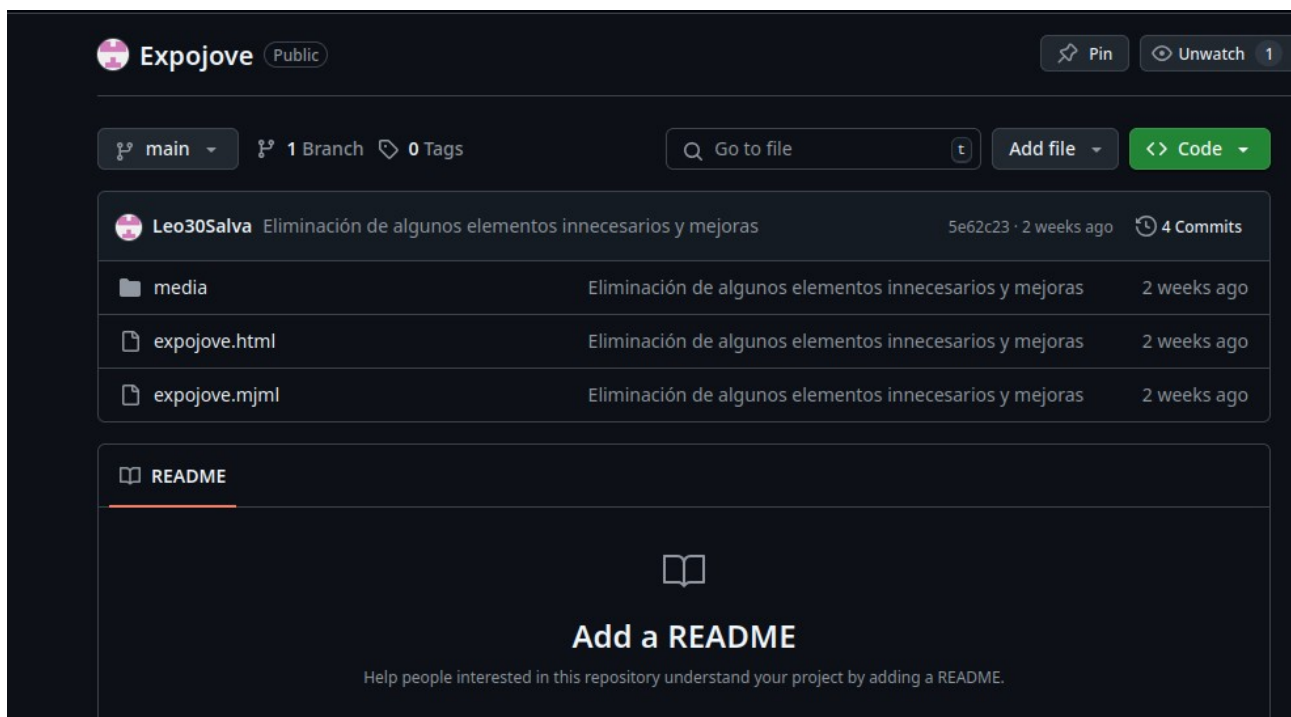


The screenshot shows a code editor interface with a sidebar on the left and a main editor area on the right. The sidebar contains a 'SOURCE CONTROL' section with a 'REPOSITORIES' list showing 'Expoj...' and a 'CHANGES' section with a 'Message (Ctrl+E...)' input field and a 'Commit' button. The main editor area displays a commit message in a dark theme, with line numbers 1 through 22 on the left. The commit message content is as follows:

```
1 <mjml>
2 <mj-head>
3   <mj-title>Discount Light</mj-title>
4   <mj-preview>Pre-header Text</mj-preview>
5   <mj-attributes>
6     <mj-all font-family="Helvetica Neue", Helvetica, Arial, sans-serif">
7     <mj-text font-weight="400" font-size="16px" color="#000000" line-heig
8   </mj-attributes>
9   <mj-style inline="inline">
10     .body-section {
11       -webkit-box-shadow: 1px 4px 11px 0px rgba(0, 0, 0, 0.15);
12       -moz-box-shadow: 1px 4px 11px 0px rgba(0, 0, 0, 0.15);
13       box-shadow: 1px 4px 11px 0px rgba(0, 0, 0, 0.15);
14     }
15   </mj-style>
16   <mj-style inline="inline">
17     .text-link {
18       color: #5e6ebf
19     }
20   </mj-style>
21   <mj-style inline="inline">
22     .footer-link {
```

Actualmente ya tengo todos los commits finalizados, en este caso para el apartado de Expojove he realizado 4 commits en los que he ido escribiendo mensajes del proceso, a continuación dejaré un link de mi página de GitHub para poder consultarlo:

<https://github.com/Leo30Salva>



# AWS

Con este comando paso la carpeta donde se encuentran los documentos del mjml, fotos, etc.

```
dam@aulainf04-pc15:~$ scp -r -i /home/dam/Escritorio/clave_AWS.pem /home/dam/Escritorio/Comic/ ubuntu@ec2-3-83-149-167.compute-1.amazonaws.com:/home/ubuntu/scripts_correo
```

En la siguiente imagen se puede ver como se van pasando poco a poco

```
COMMIT_EDITMSG      100%  57      0.5KB/s   00:00
description          100%  73      0.7KB/s   00:00
expojoye.html        100% 25KB    111.4KB/s 00:00
ninios.jpg           100% 125KB   161.8KB/s 00:00
banner_expojoye.jpg  100%  44KB   132.0KB/s 00:00
logo-fv.png          100%  10KB    89.0KB/s  00:00
cuadraditos.jpg      100% 117KB   151.4KB/s 00:00
banner_expojoye-removebg-preview.png 100%  70KB   158.8KB/s 00:00
fondo.jpg            100% 8574    75.5KB/s  00:00
expojoye_nadal.jpg   100%  89KB   161.9KB/s 00:00
cochecitos.jpg       100% 164KB   165.8KB/s 00:00
dam@aulainf04-pc15:~$ scp -r -i /home/dam/Escritorio/clave_AWS.pem /home/dam/Escritorio/Comic/ ubuntu@ec2-3-83-149-167.compute-1.amazonaws.com:/home/ubuntu/scripts_correo
comic.mjml           100% 4937    41.9KB/s   00:00
comic.html           100%  21KB    92.5KB/s   00:00
config               100%  442     3.8KB/s    00:00
```

Una vez ya están todos los archivos pasados al servidor, empezaré a utilizar una herramienta que me permite instalar en ubuntu repositorios para javascript, en este caso instalé un repositorio de mjml, otro que me servirá para administrar todo el sistema de correos que es “nodemailer” y por último “node-cron” que me servirá para programar la hora a la que quiero que se ejecute el script que he creado.

```
ubuntu@ip-172-31-94-19:~/correo-automatico$ sudo npm install -g mjml
added 167 packages in 9s
39 packages are looking for funding
  run `npm fund` for details
ubuntu@ip-172-31-94-19:~/correo-automatico$ sudo npm install nodemailer node-cron
added 1 package, and audited 3 packages in 640ms
found 0 vulnerabilities
```

Una vez descargados los repositorios hay que iniciar un repositorio en el directorio donde se vaya a realizar todo, para poder utilizar correctamente las herramientas de npm.

```
package.json package.json sendEmail.js
~/correo-automatico$ npm init -y
```



A continuación utilizo el comando “pm2 start para que el script siga en marcha aún que la terminal este cerrada, es decir, en segundo plano.

Esto es algo que serviría aplicarlo en una práctica normal pero en este caso no tendría mucha funcionalidad ya que cada 4 horas el servidor se apaga y esto que hago no serviría de nada, pero para un caso real si.

```
ubuntu@ip-172-31-94-19:~/correo-automatiko$ pm2 start sendEmail.js --name "email-scheduler"
```

-----



#### Runtime Edition

PM2 is a Production Process Manager for Node.js applications  
with a built-in Load Balancer.

Start and Daemonize any application:  
\$ pm2 start app.js

Load Balance 4 instances of api.js:  
\$ pm2 start api.js -i 4

Monitor in production:  
\$ pm2 monitor

Make pm2 auto-boot at server restart:  
\$ pm2 startup

To go further checkout:  
<http://pm2.io/>

-----

```
[PM2] Spawning PM2 daemon with pm2_home=/home/ubuntu/.pm2
[PM2] PM2 Successfully daemonized
[PM2] Starting /home/ubuntu/correo-automatiko/sendEmail.js in fork_mode (1 instance)
[PM2] Done.
```

id	name	mode	♿	status	cpu	memory
0	email-scheduler	fork	0	online	0%	14.0mb

Este sería el script que ejecuta todas las tareas necesarias. (Aumentar pantalla para poder leerlo bien)

```
GNU nano 7.2                                                                    sendEmail.js
// sendEmail.js

const nodemailer = require('nodemailer');
const cron = require('node-cron');
const fs = require('fs');
const { exec } = require('child_process'); // Para ejecutar el comando mjml

// Configuración del transporte de Nodemailer
// Aquí debes configurar tu servicio de correo electrónico (Gmail, Outlook, SendGrid, etc.)
// Ejemplo con Gmail:
const transporter = nodemailer.createTransport({
  service: 'gmail',
  auth: {
    user: 'salvaferiavalencia@gmail.com', // Tu dirección de correo
    pass: 'relrsyumvrikgrae' // Contraseña de aplicación si usas 2FA, o tu contraseña normal
  }
});

// Lista de correos a los que se enviará
const recipients = [
  'salvaferiavalencia@gmail.com',
  'irenemazarodriguez@gmail.com',
  'ncubillos2006@gmail.com'
];

// Función para compilar MJML a HTML
function compileMjmlToHtml(mjmlFilePath, callback) {
  exec(`mjml ${mjmlFilePath}`, (error, stdout, stderr) => {
    if (error) {
      console.error(`Error al compilar MJML: ${error.message}`);
      return callback(error);
    }
    if (stderr) {
      console.error(`Error de stderr al compilar MJML: ${stderr}`);
      // A veces mjml imprime advertencias aquí que no son errores fatales
    }
    callback(null, stdout); // stdout contiene el HTML compilado
  });
}

// Función para enviar el correo
async function sendDailyEmail() {
  console.log('Intentando enviar correos...');

  const mjmlFilePath = '/home/ubuntu/scripts_correo/Expojove/expojove.mjml';

  compileMjmlToHtml(mjmlFilePath, async (err, htmlContent) => {
    if (err) {
      console.error('No se pudo obtener el contenido HTML del MJML.');
```

Read 81 lines

```

        console.error('No se pudo obtener el contenido HTML del MJML.');
```

```

        return;
    }

    const mailOptions = {
        from: 'tu_correo@gmail.com', // Remitente
        subject: 'Expojove - ' + new Date().toLocaleDateString(), // Asunto
        html: htmlContent // Contenido HTML del correo
    };

    for (const recipient of recipients) {
        mailOptions.to = recipient; // Establece el destinatario para cada envío
        try {
            let info = await transporter.sendMail(mailOptions);
            console.log(`Correo enviado a ${recipient}: %s`, info.messageId);
        } catch (error) {
            console.error(`Error al enviar correo a ${recipient}:`, error);
        }
    }
});

// Programar el envío de correos a las 17:00
// La sintaxis de cron es: minuto hora día_del_mes mes día_de_la_semana
// '0 17 * * *' significa: A los 0 minutos de las 17 horas, cada día del mes, cada mes, cada día de la semana.
cron.schedule('37 19 * * *', () => {
    console.log('Ejecutando la tarea programada: envío de correos a las 17:00');
    sendDailyEmail();
}, {
    timezone: "Europe/Madrid" // Asegúrate de que la zona horaria sea correcta para tu ubicación
});

console.log('Servicio de envío de correos iniciado. Esperando la hora programada (17:00).');
```

El script se divide en varias secciones, la primera sería de “transporter” en la que defino cual será el correo de origen que en este caso es uno que me he creado, y por otra parte una “contraseña de aplicación de google” que he generado para esto en concreto, ya que como en este caso estoy utilizando gmail es necesario crear esta contraseña que es específicamente para esto.

La siguiente función es “recipients” que básicamente contiene las direcciones de correo a las que voy a enviar.

Luego otra función que compila el archivo “mjml” a “html”

Después otra función que es “sendDailyEmail” que tendrá que esperar para ser ejecutada.

Por último una función de cron que es el sistema que se encarga de enviar automáticamente los correos, en el se debe indicar la hora(en la imagen se ve una hora de prueba, no es la definitiva) los 2 primeros dígitos son para los minutos, los siguientes para la hora, los siguientes los segundos y día de la semana.

En mi caso como quiero que se ejecute todos los días a las 17 sería: (00 17 \* \* \*).

Una vez llegada esa hora se ejecutaría la función que envía los correos a todos los destinatarios indicados en “recipients”.

```
ubuntu@ip-172-31-94-19:/opt$ sudo ufw allow 'Apache Full'
Rule added
Rule added (v6)
ubuntu@ip-172-31-94-19:/opt$ sudo ufw reload
Firewall reloaded
ubuntu@ip-172-31-94-19:/opt$
```

Añado una regla en el cortafuegos para que permita acceso al apache tanto al puerto 80 como el 443.

Habilito ssl para que funcione el 443.

Bash

```
sudo a2enmod ssl
sudo systemctl restart apache2
```

Necesito crear certificados para que funcione el 443, por lo que ejecutaré este comando para crearlos.

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

Le cambio los permisos a la clave privada para que nadie tenga acceso aparte de mi.

```
sudo chmod 600 /etc/ssl/localcerts/apache-selfsigned.key
```

Creo otro site para el 443

```
/etc/apache2/sites-available/imagenes-ssl.conf  
  
<IfModule mod_ssl.c>  
  <VirtualHost _default_:443>  
    # Puedes usar tu IP local aquí si quieres ser específico, pero _default_:443 funciona bien  
    # ServerName tu_ip_local_del_servidor:443  
    # Por ejemplo: ServerName 192.168.1.100:443  
  
    ServerAdmin webmaster@localhost  
    DocumentRoot /var/www/html # Esto es el DocumentRoot por defecto, puedes dejarlo o apuntarlo a un directorio vacío.  
  
    ErrorLog ${APACHE_LOG_DIR}/error.log  
    CustomLog ${APACHE_LOG_DIR}/access.log combined  
  
    # Configuración SSL/TLS  
    SSLEngine on  
    SSLCertificateFile /etc/ssl/localcerts/apache-selfsigned.crt  
    SSLCertificateKeyFile /etc/ssl/localcerts/apache-selfsigned.key  
  
    # Tu configuración de Alias para las imágenes  
    Alias /imagenes/ "/opt/mis-imagenes/"  
  
    <Directory "/opt/mis-imagenes/">  
      Options Indexes FollowSymLinks  
      AllowOverride None  
      Require all granted  
    </Directory>  
  
    # Configuración para evitar algunos errores comunes en navegadores con certificados autofirmados (opcional, pero recomendado)  
    SSLProtocol all -SSLv2 -SSLv3  
    SSLCipherSuite HIGH:!aNULL:!MD5  
    SSLHonorCipherOrder on  
  
    BrowserMatch "MSIE [2-6]" \  
      nokeepalive ssl-unclean-shutdown \  
      downgrade-1.0 force-response-1.0  
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown  
  
  </VirtualHost>  
</IfModule>
```

Al final opté por otra opción ya que los certificados autofirmados no sirven en google y dan muchos problemas, y la mayoría de páginas que ofrecen un dominio no se puede acceder.

Por lo que la opción que opté es utilizar las imagenes que tengo subidas a github, ya que mediante los commit realizados haciendo el diseño se subieron.

Expojove / media /

Leo30Salva Eliminación de algunos elementos innecesarios y mejoras 5e62c23 · 2 weeks ago History

Name	Last commit message	Last commit date
..		
banner_expojove-removebg-preview.png	Avance de la página, puesto el banner y algunos textos	2 weeks ago
banner_expojove.jpg	Avance de la página, puesto el banner y algunos textos	2 weeks ago
cochecitos.jpg	Añadidas redes sociales, más imagenes y eliminación de algunos elementos	2 weeks ago
cuadraditos.jpg	Avance de la página, puesto el banner y algunos textos	2 weeks ago
expojove_nadal.jpg	Eliminación de algunos elementos innecesarios y mejoras	2 weeks ago
fondo.jpg	Avance de la página, puesto el banner y algunos textos	2 weeks ago
logo-fv.png	Avance de la página, puesto el banner y algunos textos	2 weeks ago
ninios.jpg	Añadidas redes sociales, más imagenes y eliminación de algunos elementos	2 weeks ago



## Redes Sociales



En la anterior imagen se puede ver como sería el diseño de la publicidad en redes sociales, tiene una votación que es puro marketing para incitar a la gente a asistir al Expojove, pero no tiene ninguna funcionalidad específica en si, sería una historia de instagram la cual deslizando hacia arriba se podría visualizar el newsletter.



# Diseño con Figma

[Pinchando aquí verás el documento Figma](#)



El lugar dispone de varias ventajas, como por ejemplo el hecho de que cada visitante tenga acceso durante una jornada entera, algo que beneficia en poder entender y prestar atención a la marca a la perfección.

También ofrecemos la participación de los medios más importantes de la comunidad Valenciano.

Posibilidad de exponer la marca de manera estimulante y dinámica para la generación alfa.



---

## Contáctanos

Cualquier duda puedes consultarnosla y te la resolveremos.

### Dirección

Av. de les Fires, s/n,  
Pobles de l'Oest  
46035 València, Valencia

### Información de contacto

Numero: 963 86 12 00  
[Página de contacto](#)

[View this email in your browser](#)

Recibiste este anuncio por correo electrónico porque eres un actual expositor de Expojove. (Av. de les Fires, s/n, Pobles de l'Oest 46035 València, Valencia) y aceptó recibir correos electrónicos nuestros sobre nuevas funciones, eventos y ofertas especiales.

