

AMPLIACIÓN DEL RETO FERIA VALENCIA.



DISEÑO Y ENVÍO DE UNA CAMPAÑA DE MARKETING DIGITAL USANDO MJML RETOS.



ANÁLISIS Y PLANIFICACIÓN

En este caso practico lo que se pide es crear un correo de **Expojove** el cual tiene que enviar una **comunicación** a sus **expositores** destacando los **servicios** a los que tienen acceso y **como solicitarlos**.

Para la **paleta** de **colores** como debíamos de usar una imagen que nos ha proporcionado la feria de muestras he usado los mismos colores tanto como de la **imagen** para darle el **mismo estilo**.

En este caso he puedo los **titulos** en **naranja** y **resaltado** en **azul oscuro** **palabras clave** para que **destaquen** sobre las otras además les he subido el **tamaño** de la **fuentes** para que se vean mas aun, **resaltando** los **visitantes** que suelen venir cada año y los **servicios** que **ofrece la feria**.

Esto lo hice asi ya que los expositores de las ferias reciben muchos mensajes y estos podrían llegar a perderse en su correo

Debo de **analizar** los **objetivos de este correo**, el cual ira para los **expositores** de la de la **feria** los cuales son **adultos** los cuales irán desde gente de unos **30 años hasta unos 50**.

DAFO

Debilidad: Un **exceso** de elementos **resaltados** o de colores muy **contrastados** podría **abrumar** a algunos lectores. Es importante encontrar un **equilibrio**.

Amenaza: Los **expositores** reciben muchos correos; el de Expojove debe destacarse para no pasar desapercibido.

Fortaleza: Mencionar la **afluencia** de **visitantes anuales** puede ser un **gran incentivo** para los expositores, mostrando el **potencial** de la **feria**.

Oportunidad: Un correo claro y útil puede animar a los expositores a utilizar más servicios, mejorando su experiencia en la feria y, potencialmente, aumentando su fidelidad.

Objetivos de la campaña

El objetivo de esta campaña es crear un correo de Expojove el cual tiene que enviar una comunicación a sus **expositores** destacando los **servicios** a los que tienen acceso y **como solicitarlos**.

DISEÑO

La **planificación** del **diseño** de mi correo ha sido **principalmente** haciendo una **plantilla** en **figma** en la cual puse la estructura que iba a tener, añadiendo a el principio para captar la atención una imagen del Expojove con la fecha cuando se realizara y debajo los principales invitados y mas abajo las principales noticias y promociones.

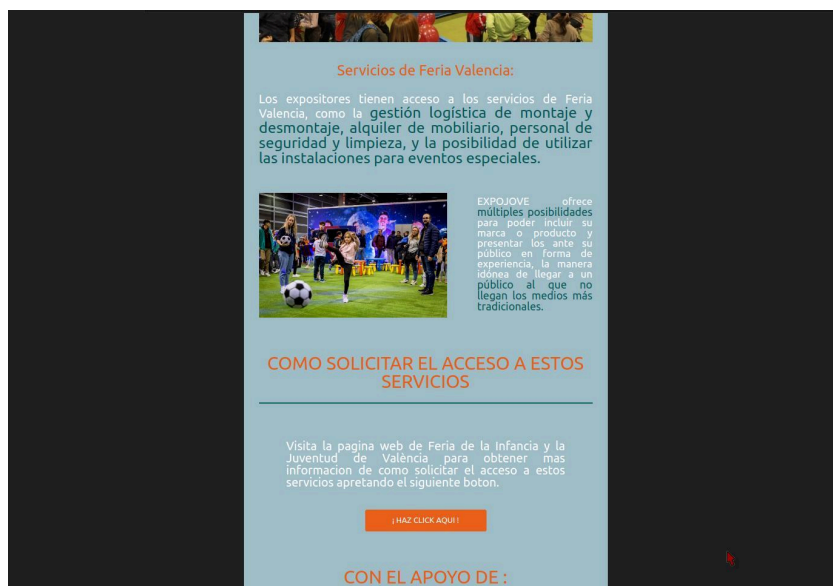
He usado **colores llamativos** para llamar la **atención** de la gente de los **expositores** para que no dejen de lado el correo ya que contiene **información importante** para ellos, la selección de **colores** ha sido **correspondiente** con la imagen de **expojove**.

En el correo tenemos los servicios que se ofrecen en la feria, además de como acceder a la pagina web con los servicios correspondientes.

Aqui podrás ver el modelo en figma que hize para poder ver como cree la plantilla para poder crear el correo.

<https://www.figma.com/design/d5CmRQupjQTKbLmYVYXoLj/TRABAJO-MJML-EXPOJOVE?node-id=0-1&p=f&t=5pjDhDHepaTlqWfO-0>

Para que el MJML sea responsive se piden 3 media queries aqui pondre unas imágenes usando 3 distintas.



Aqui se veria con las dimensiones mas grandes posibles.



Servicios de Feria Valencia:

Los expositores tienen acceso a los servicios de Feria Valencia, como la **gestión logística de montaje y desmontaje**, **alquiler de mobiliario**, **personal de seguridad y limpieza**, y la posibilidad de utilizar las instalaciones para eventos especiales.



EXPOJOVE ofrece múltiples posibilidades para poder incluir su marca o producto y presentar los ante su público en forma de experiencia, la manera idónea de llegar a un público al que no llegan los medios más tradicionales.

COMO SOLICITAR EL ACCESO A ESTOS SERVICIOS

Visita la pagina web de Feria de la Infancia y la Juventud de Valencia para obtener mas informacion de como solicitar el acceso a estos servicios apretando el siguiente boton.


[¡ HAZ CLICK AQUI !](#)

CON EL APOYO DE :

Este seria un tamaño mediano y se veria de esta manera.

Servicios de Feria Valencia:

Los expositores tienen acceso a los servicios de Feria Valencia, como la **gestión logística de montaje y desmontaje**, **alquiler de mobiliario**, **personal de seguridad y limpieza**, y la posibilidad de utilizar las instalaciones para eventos especiales.



EXPOJOVE ofrece múltiples posibilidades para poder incluir su marca o producto y presentar los ante su público en forma de experiencia, la manera idónea de llegar a un público al que no llegan los medios más tradicionales.

COMO SOLICITAR EL ACCESO A ESTOS SERVICIOS

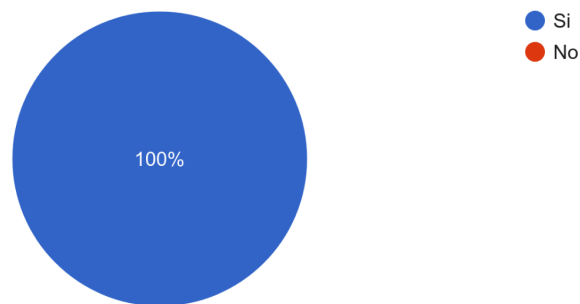
Por ultimo como se veria en un formato mas pequeño como en un teléfono.

TEST CON USUARIOS

<https://docs.google.com/forms/d/e/1FAIpQLSfkzTT9hiGcZw1lb8EEruRu7pntdrcZ3RX5Y1jvh4pm7FvKjA/viewform?usp=dialog>

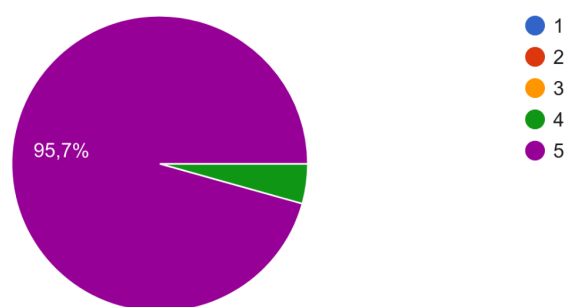
1 - ¿Considera que el trabajo cumplió con los objetivos planteados?

23 respuestas



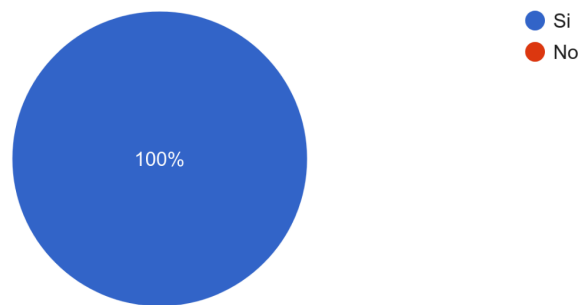
2 - En una escala de 1 a 5, ¿cómo calificaría el trabajo?

23 respuestas



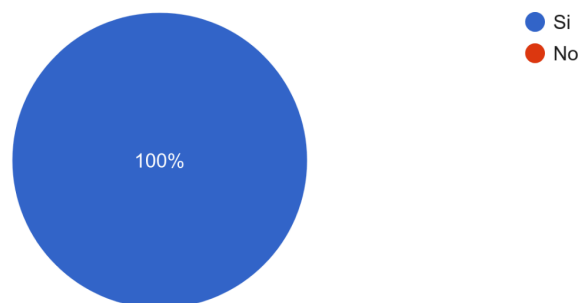
3 - ¿Crees que la información es relevante y útil para los usuarios o interesados?

23 respuestas



4 - ¿Crees que el proyecto ha sido diseñado de forma intuitiva y accesible?

23 respuestas



5 - ¿Qué sugerencias o recomendaciones tendrías para mejorar el proyecto?

9 respuestas

Esta muy bien no se puede mejorar

Lo encuentro muy vistoso y muy util

Me gusta tal y como esta !!

No tengo ninguna

No tendria ninguna sugerencia, ya que para mi es un trabajo muy completi

Esta así bien

Nada, está muy bien hecho

Ningunano

No tengo ninguna

6 - ¿Crees que falta alguna información relevante en el proyecto?

12 respuestas

No

No, está toda la información

Esta muy completo

No, está completo

No.



DESARROLLO Y ENVÍO DE CORREOS

Para empezar lo que hare es crear una maquina virtual con AWS, la crearemos buscando EC2 y creando una instancia. Seleccionamos ubuntu y le damos a los ajustes gratuitos para estudiantes, en el momento de seleccionar par de claves le damos a crear claves y le ponemos un nombre y en formato .pem una vez lo hagamos se descargaran y deberemos de guardarla ya que es muy importante, una vez echo crearemos la maquina virtual.

Para conectar mi ordenador a mi ec2 con ssh, para lo cual necesito el archivo (**jaade.pem**) y la ip publica de mi EC2 (**100.24.61.249**) y el nombre de usuario (**ubuntu**).

Un archivo **.pem** es un formato de archivo muy común que se usa para almacenar claves criptográficas y certificados digitales.

Voy donde he creado mi .pem este caso en una carpeta llamada pem :

```
cd /home/dam/Escritorio/pem
```

Y ejecutas el siguiente comando para darle los permisos necesarios :

```
chmod 400 jaade.pem
```

Ahora lo que debes hacer es conectarte a la instancia EC2 :

```
ssh -i /home/dam/Escritorio/pem/jaade.pem ubuntu@100.24.61.249
```

De esta manera ya esta conectado nuestro ordenador con la instancia EC2.

A continuación voy a crear un **Node.js** como un entorno donde puedes ejecutar código JavaScript fuera de un navegador web.

Creo una carpeta yo la llame **enviador-emails-mjml** y dentro de ella dos archivos llamados:

package.json : Guarda información sobre tu aplicación y las librerías de Node.js que necesito.

index.js : Este será el archivo principal donde estará todo el código JavaScript para leer datos, procesar MJML y enviar correos.

En el archivo **package.json** estara este codigo:

```
{
  "name": "enviador-emails-mjml",
  "version": "1.0.0",
  "description": "Script para enviar correos masivos con MJML en Node.js",
  "main": "index.js",
```



```

    }

    // Datos para personalizar la plantilla (adapte esto a los campos de tu JSON)
    // Por ejemplo, si tu JSON tiene 'nombre' y 'apellido', puedes usarlos así:
    const datosPersonalizar = {
      nombreCliente: cliente.nombre || 'Estimado Cliente', // Usa el nombre si existe, si no, un valor por defecto
      // Añade más placeholders que uses en tu MJML, por ejemplo:
      // apellidoCliente: cliente.apellido || 'Señor/a',
      // emailCliente: cliente.email || 'correo@cliente.com'
    };

    // Asunto del correo (puedes personalizarlo también)
    const asuntoCorreo = `Hola, ${datosPersonalizar.nombreCliente}! Tenemos noticias para ti.`;

    // Renderizar la plantilla MJML a HTML, con los datos del cliente
    const htmlFinal = renderEmail({
      template: plantillaMJML,
      data: datosPersonalizar
    });

    // Enviar el correo
    await enviarCorreoConGmail({
      email: cliente.email,
      asunto: asuntoCorreo,
      html: htmlFinal
    });

    // Opcional: Pequeño delay entre envíos para evitar saturar el servidor de correo
    // Usa el timeo (límite de tiempo) o quito (simular un envío más lento)
    // await new Promise(resolve => setTimeout(resolve, 1000)); // Espera 1 segundo
  }

  console.log('--- Proceso de envío de correos finalizado ---');
}

// Ejecutar la función principal para iniciar el proceso
main();

```

lo que hace este código es:

- **Configuración Segura:** Primero, el script se asegura de tener tu contraseña de aplicación de Gmail y mi correo de remitente configurados como variables de entorno. Si no están, detiene el proceso.
- **Plantilla MJML:** Lee una plantilla de correo electrónico en formato MJML (un lenguaje para crear emails responsivos) desde un archivo.
- **Datos de Clientes:** Carga una lista de clientes y sus datos (como nombres y correos) desde un archivo JSON.
- **Personalización y Conversión:** Por cada cliente en la lista, toma la plantilla MJML, reemplaza los datos genéricos (como `{{nombreCliente}}`) con la información específica de ese cliente, y luego convierte el MJML a HTML (el formato que entienden los clientes de correo).
- **Envío por Gmail:** Finalmente, utiliza Nodemailer (una librería para enviar correos) para enviar el correo personalizado a cada cliente usando

En la carpeta **enviador-emails-mjml**, creare estos dos archivos:

plantilla-email.mjml (con el mjml)

clientes.json

```
[
  {
    "nombre": "irene",
    "email": "irenemaza@gmail.com"
  },
  {
    "nombre": "salva",
    "email": "leo30salva@gmail.com"
  },
  {
    "nombre": "javi",
    "email": "javierdescalsfernandez@gmail.com"
  }
]
```

A continuación lo que hay que hacer es poner el correo electrónico y el código de aplicación de la misma para que el programa pueda enviar correos desde este.

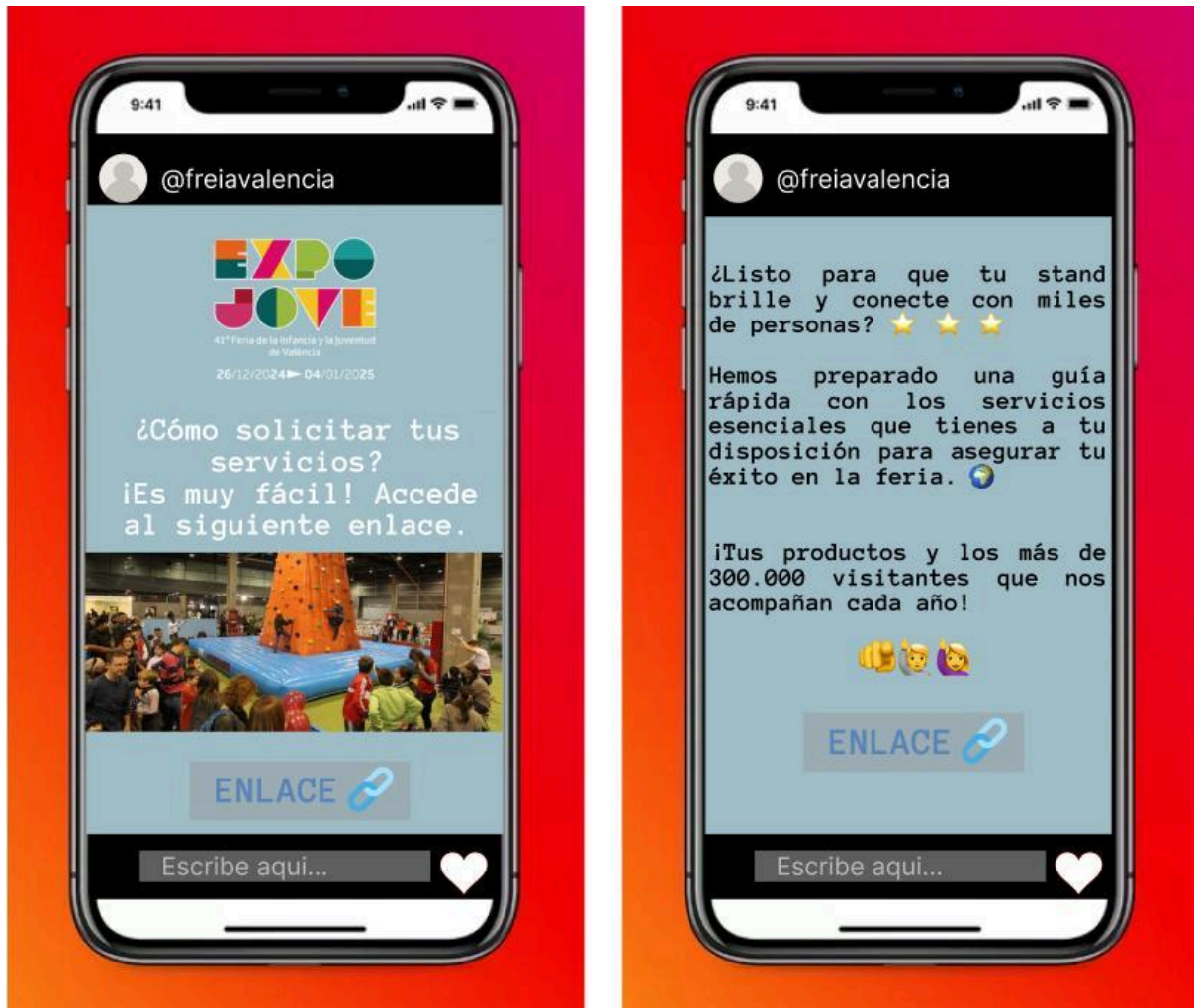
```
export GMAIL_APP_PASSWORD='CONTRASEÑA_DE_APLICACION'
```

```
export SENDER_EMAIL_GMAIL='CORREO_QUE_ENVIARA_MJ@gmail.com'
```

Por último usamos este comando y se enviarán los correos:

```
node index.js
```

INTEGRACIÓN CON REDES SOCIALES



CONTROL DE VERSIONES CON GITHUB

El control de versiones de este MJML se encuentra en este link:

<https://github.com/LiLjaade/mjmlExpojove>

DATOS

Como se pedia la creacion de una base de datos instale mySQL

```
sudo apt update
```

```
sudo apt install mysql-server -y
```

Ejecuta el script de seguridad de MySQL

sudo mysql_secure_installation

Me conecto a mysql y creo la base de datos

sudo mysql -u root -p**CREATE DATABASE emails_db;**

Creamos un usuario y le damos permisos

Creamos una tabla clientes

```
USE emails_db;  
CREATE TABLE clientes (  
  id INT AUTO_INCREMENT PRIMARY KEY,  
  nombre VARCHAR(255) NOT NULL,  
  email VARCHAR(255) UNIQUE NOT NULL  
);
```

He inserto datos de prueba:

```
INSERT INTO clientes (nombre, email) VALUES ('Ana', 'ana.ejemplo@gmail.com');  
INSERT INTO clientes (nombre, email) VALUES ('Carlos', 'carlos.ejemplo@gmail.com');  
INSERT INTO clientes (nombre, email) VALUES ('Maria', 'maria.ejemplo@gmail.com');  
INSERT INTO clientes (nombre, email) VALUES ('Javier',  
'javierdescalsfernandez@gmail.com');
```

Salimos de la consola exit;

Modificamos el [index.js](#) para que funcione con la base de datos

Y configuramos las variables de entorno

```
export GMAIL_APP_PASSWORD='CONTRASEÑA_DE_16_DIGITOS_SIN_ESPACIOS'  
export SENDER_EMAIL_GMAIL='tu_correo_de_envio@gmail.com'  
export DB_HOST='localhost'  
export DB_USER='app_user'  
export DB_PASSWORD='tu_contraseña_segura'  
export DB_NAME='emails_db'
```

Y por ultimo ejecutamos el script

node index.js

PERSONALIZACIÓN DEL CORREO MJML



- Por cada cliente en la lista, toma la plantilla MJML, reemplaza los datos genéricos (como {{nombreCliente}}) con la información específica de ese cliente, y luego convierte el MJML a HTML (el formato que entienden los clientes de correo).

DOCUMENTACIÓN FINAL

Gracias a este trabajo he aprendido a como realizar un **proyecto desde 0** empezando desde **análisis y bocetos** hasta hacer el **envío de correos** y ver las **opiniones** de varios **usuarios** y así poder **mejorar** en varios aspectos

De primeras tuve que ver **cual era el objetivo** de este correo para así poder ver como plantearlo.

He aprendido a analizar el público objetivo y adaptar la estrategia de diseño del email.

Planificar el diseño visual del email desde el boceto inicial hasta la creación de un modelo en Figma, prestando atención a la paleta de colores y la jerarquía visual.

Adaptabilidad del diseño (responsive) para diferentes dispositivos (grandes, medianos y móviles).

Realizando **test de usuarios** para ver en que puedo mejorar y realizar cambios para tener un mayor acercamiento al publico.

Desarrollar una infraestructura de envío de correos desde cero:

- La configuración de una **máquina virtual en AWS (EC2)** y la conexión segura a través de SSH.
- La implementación de un **entorno Node.js** con librerías clave como MJML (para convertir el diseño a HTML).
- La **personalización de correos** con datos dinámicos de los destinatarios.
- Uso de **variables de entorno** para manejar credenciales de forma segura.

Integrar una base de datos MySQL para gestionar la lista de clientes, demostrando habilidades en la creación de bases de datos, tablas, usuarios y la inserción de datos.

Implementar un **control de versiones** eficaz usando GitHub para gestionar el código del proyecto.