

The image shows a large, modern exhibition booth for FERIA CEVISAMA. The booth's walls are covered in a detailed mural of a tropical jungle scene, featuring various plants like monstera and palm trees, and a large tree trunk. A large, arched mirror is mounted on the back wall, reflecting the interior of the booth. In the foreground, there are two modern, minimalist kitchen or bathroom vanity units. The unit on the left is dark-colored, and the one on the right is light-colored. The booth is set up in a large, open hall with a high ceiling and industrial lighting. Two people are visible in the background, looking at the booth. The overall atmosphere is sophisticated and artistic.

# FERIA CEVISAMA

cerámica, arquitectura  
, baños y cocinas

Contacta con nosotros si estás dispuesto  
a exponer tus productos

@FERIAVALENCIA

**SUPUESTO 2:** Cevisama ha hecho una gran inversión para atraer a más compradores internacionales a la feria. Ahora, necesitan comunicarle esto a los expositores (las empresas que tienen stands) para que entiendan el beneficio y el potencial que esto les trae, como más oportunidades de negocio y mayor visibilidad para sus productos.

## **INDICE**

1. Objetivos del proyecto
2. Análisis y público objetivo
3. Diseño y justificación
4. Solución técnica
5. Test de usuarios (formularios, gráficos, feedback)
6. Integración con redes
7. Control de versiones
8. Datos
9. Conclusiones y posibles mejoras

## **12. Entrega y Defensa del Proyecto (9–10 de junio)**

- Entrega en Github y web de Feria Valencia
- Presentación del proyecto
- Demostración funcional del correo
- Resultados del test con usuarios
- Explicación del proceso de desarrollo y herramientas
- Conclusión y propuestas de mejora

## **1. Retos de la Campaña**

En este caso, he decidido coger el supuesto 2, que trata de que Cevisama invertirá mucho dinero para atraer a más compradores internacionales. Necesitan que los expositores (las empresas que tienen stands) sepan sobre esta inversión para que entiendan el esfuerzo y el beneficio que esto les traerá.

## **2. Análisis y Planificación**

- Análisis de requerimientos

El público objetivo principal de esta campaña son los expositores actuales de Cevisama 2026, en este caso dentro de ese grupo podemos encontrar otros:

- Expositores que van todos los años a participar en Cevisama cada año.
  - Nuevos expositores que están considerando participar y para los que es una oportunidad.
  - Personal de marketing y ventas de las empresas expositoras, son los que piensan y implementan las estrategias en la feria.
- En este caso, el objetivo es convencer a todas estas partes de que la inversión de Cevisama significa un retorno mayor para ellos.

- Método DAFO



- Análisis de la competencia

En este caso, hay varias ferias que hacen algo parecido a lo que hace Cevisama como IDEO BAIN o MosBuild, pero nosotros lo que estamos haciendo con la inversión es poder darle más oportunidades de negocio a los expositores que participan. Además de que la clave es demostrarles que la feria pese a las competencias es una inversión rentable para su negocio.

- Objetivos de la campaña

Los objetivos de esta campaña son:

- Comunicar la inversión histórica para atraer compradores internacionales
- Fomentar la participación de expositores
- Generar visitas a la web oficial
- Captar leads mediante newsletter

### **3. Diseño**

- Creatividad e innovación

El diseño de la campaña se destaca por:

- Enfoque profesional sobre una ampliación de presupuesto para los expositores.
- Colores en contraste al perfil mandado (azul, verde agua, amarillo y gris).
- Imágenes en concordancia al correo (Logo de Cevisama, un expositor, una pieza que se puede vender).
- Diseño responsivo, gracias a él se puede ver tanto en pantalla grande (ordenador) o pequeña (móvil).
- Estructura clara: los espacios entre los diferentes apartados se ven de manera clara.

- Mockup (Figma)

El diseño que he creado es el siguiente:



Este diseño se compone de tres partes:

1. Cabecera: Encontramos el titular principal y el logo de Cevisama.
2. Cuerpo: En el cuerpo encontramos dos apartados, el primer apartado, introduce el concepto de la inversión. Con texto detallando las acciones (campañas reforzadas, programas VIP) y una imagen de ambiente de feria, mientras que el segundo encontramos: "Queremos que tu participación en Cevisama 2026 sea la más rentable hasta la fecha.", esto quiere decir que estamos haciendo todo lo posible para que quieran participar y por último, cerramos el cuerpo con una imagen de interacción profesional y un mensaje de disponibilidad.
3. Pie: En este apartado podemos encontrar las redes sociales que tiene en este caso Feria Valencia, además de un enlace para más información de Cevisama y de feria Valencia.

- Responsive avanzado con MJML

En este caso, el propio MJML te lo hace Responsive.

#### 4. Test con Usuarios

- Selección de muestra

En este caso, he hecho una selección de 10 (familia y amigos).

- Formulario de valoración

En el formulario las preguntas que se encuentran son:

1. ¿Qué tan atractivo te parece el diseño del correo?
2. ¿Qué tan claro encuentras el mensaje del correo?
3. ¿Consideras que el correo refleja la imagen de la Feria Cevisama 2026?
4. ¿Qué mejorarías en el diseño? (Respuesta abierta)
5. ¿Considerarías participar en el evento tras leer este correo?
6. ¿El correo es fácil de leer y entender?

- Feedback

El feedback que he tenido ha sido muy bueno, de manera que las respuestas han sido positivas.

- Rediseño

No habría que rediseñar nada.

## 5. Desarrollo y Envío de Correos

- Tecnologías (AWS EC2 + mjml + node.js)

En este caso los pasos son los mismos que hemos hecho en Feria de Dos Ruedas:

```
dam@aulainf04-pc16:~$ scp -i "/home/dam/Escritorio/trabajo.pem" /home/dam/Escritorio/MGP.mjml ubuntu@50.17.118.215:~/home/ubuntu/MGP.mjml
MGP.mjml
100% 4293 35.4KB/s 00:00
dam@aulainf04-pc16:~$
```

```

buntueip-172-31-19-23:~$ nano node.js
buntueip-172-31-19-23:~$ node node.js
servicio de envío de correos iniciado. Esperando la hora programada (17:00).
Ejecutando la tarea programada: envío de correos a las 17:00
Intentando enviar correos...
correo enviado a salvaferiavalencia@gmail.com: <2f978735-85fa-512f-58f7-7a3c1fb0e1d@gmail.com>
correo enviado a irenemazarodriguez@gmail.com: <631a0a67-9c0d-29da-9f45-662737240bf@gmail.com>
correo enviado a ncubillos2006@gmail.com: <73ba5af5-3866-b27d-c880-9b48ac210a51@gmail.com>
C
buntueip-172-31-19-23:~$ nano node.js

```

```

buntueip-172-31-19-23:~$ nano node.js
buntueip-172-31-19-23:~$ pwd
/home/ubuntu
buntueip-172-31-19-23:~$ ls
node.js node_modules package-lock.json package.json
buntueip-172-31-19-23:~$ nano node.js
buntueip-172-31-19-23:~$ ls
node.js node_modules package-lock.json package.json
buntueip-172-31-19-23:~$ PWD
PWD: command not found
buntueip-172-31-19-23:~$ pwd
/home/ubuntu
buntueip-172-31-19-23:~$ ls
node.js node_modules package-lock.json package.json
buntueip-172-31-19-23:~$ ls
MGP.mjml node.js node_modules package-lock.json package.json
buntueip-172-31-19-23:~$ nano node.js
buntueip-172-31-19-23:~$ nano MGP.mjml
buntueip-172-31-19-23:~$ nano MGP.mjml
buntueip-172-31-19-23:~$ nano node.js
buntueip-172-31-19-23:~$ node node.js
node:internal/modules/cjs/loader:1137
    throw err;
    ^

```

```

Running kernel version:
6.8.0-1024-aws
Diagnostics:
The currently running kernel version is not the expected kernel version
6.8.0-1029-aws.

Restarting the system to load the new kernel will not be handled automatically,
so you should consider rebooting.

Restarting services...

Service restarts being deferred:
/etc/needrestart/restart.d/dbus.service
systemctl restart networkd-dispatcher.service
systemctl restart systemd-logind.service
systemctl restart unattended-upgrades.service

No containers need to be restarted.

User sessions running outdated binaries:
ubuntu @ session #2: sshd[1055]
ubuntu @ user manager service: systemd[1060]

In VM guests are running outdated hypervisor (qemu) binaries on this host.

```

(En este caso son las mismas fotos, pero lo hago así porque es el mismo proceso.)

- En este caso, lo que he gastado para poder hacer que el correo se mande automáticamente es:

```

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: 'ncubillos2006@gmail.com', // Tu dirección de correo
    pass: 'crndxfpxswqntfed' // Contraseña de aplicación si usas 2FA, o tu
    contraseña normal
  }
});

```

// Lista de correos a los que se enviará

```

const recipients = [
  'salvaferiavalencia@gmail.com',

```



```

    'cesarsueca@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.');
```

return;

```

      }
    });
    const mailOptions = {
      from: 'ncubillos2006@gmail.com', // Remitente
      subject: 'Cevisama' + 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('53 14 * * *', () => {
    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).');

```

(Esta parte del trabajo, me ayudaron mis compañeros a poder realizarla)

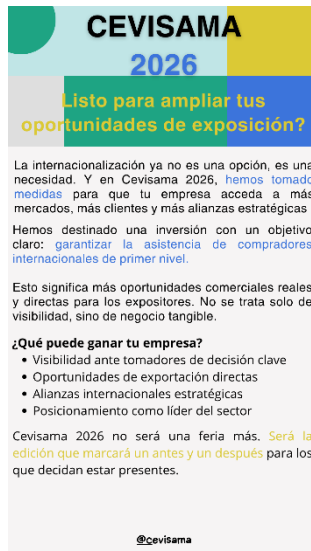
## 6. Integración con Redes Sociales

- Compartir newsletter (Telegram o LinkedIn)

En este caso, lo que pretendemos con la integración con redes Sociales es poder captar la atención de los interesados en la propuesta que expone Feria Valencia sobre Cevisama.

Por eso la propuesta es poder transmitir un anuncio a partir de LinkedIn y Instagram, sin molestar a la gente que no le llegue a gustar la propuesta.

En primer lugar, encontramos un artículo en LinkedIn donde explica un poco lo que va a ser Cevisama en la feria de 2026, para que tengáis una pequeña referencia:



Encontramos una pequeña cabecera con los colores que representa el logo de la campaña, con unas noticias que pueden ayudar a entender mejor la propuesta hecha.

- Captura para stories de Instagram

En segundo lugar, he creado una publicación de Instagram donde se pretende crear un poco de intriga sobre cómo será la próxima feria.

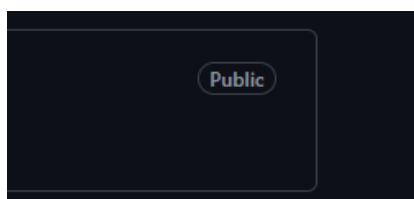


Encontramos un título y una imagen donde se puede ver reflejada de que se trata la feria de Cevisama, además se le aplica un pequeño texto para enriquecer un poco la curiosidad de los expositores.

## 7. Control de Versiones con GitHub

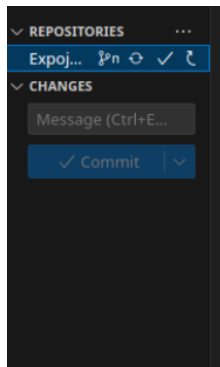
- Repositorio público

En este caso, para que todo el mundo pueda ver lo que hay en mi GitHub, hay que hacer el repositorio público:



- Extensión en Visual Studio Code

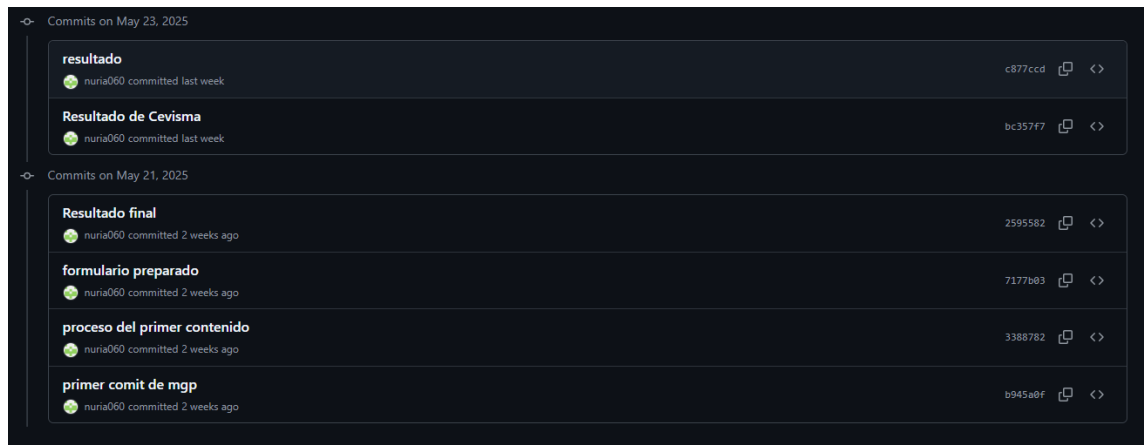
Por otra parte, como estamos haciendo el modelo desde Visual Studio Code, hay que poner la extensión para poder subir los comits de manera repetida para no perder nada.



En este caso, al poner las extensiones podremos poner los commits en este apartado de VSC.

- Commits regulares y documentación

En este caso se encuentra todo de manera ordenada en GitHub, el resultado sería el siguiente.



- Uso de ramas

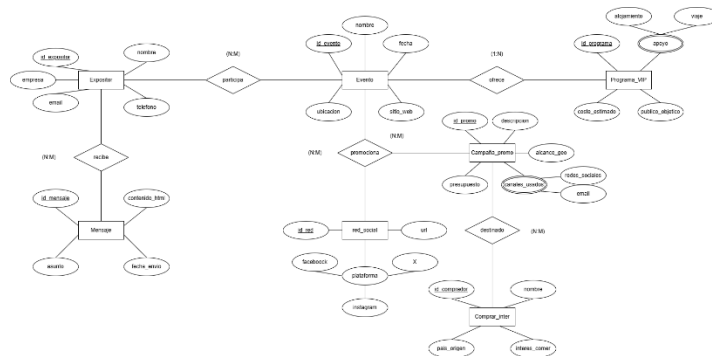
Yo en este caso, no he implementado el método de ramas, ya que no veía necesario usarlos en este tipo de trabajo.

## 8. Datos

- Diseño del esquema
  - Descripción de la estructura
    - Se compone de 5 tablas

- Cada uno tiene sus atributos correspondientes

- Modelado ER



En este caso, no me ha dado tiempo a hacer el paso a tablas, ni hacer las consultas

## 9. Conclusiones y mejoras

Para concluir, he desarrollado una solución para el email automático con una estrategia de marketing que es eficiente y atractiva

Combinando la simplicidad de MJML para diseñar nuestro correo sobre Cevisama, poder gastar AWS para la automatización, logrando una campaña para captar la atención de los expositores, los cuales se pueden ver impulsados para querer participar.

Algunas de las mejoras que puedo aportar yo a mi propio trabajo es la gestión del tiempo, ya que hay muchos conceptos que no sabía y me los dejaba para el final.