MEMORIA PROYECTO

Texto

El contenido generado por IA puede ser incorrecto.

**Realizado por:**

**Ivan Jonas Fernandez Correa**

Índice

[**Introducción 3**](#_7em228lxjeja)

[**Programación (Java) 4**](#_lo31x3p356wy)

[Creación Hashmap y ArrayList 4](#_xjrdj9isz70p)

[Menú 6](#_83z4k96rmxiv)

[Caso 1: Añadir Usuarios 7](#_fw4dlvss0pli)

[Caso 2: Eliminar Usuario 8](#_1lf0xt853fzz)

[Caso 3: Añadir evento 8](#_j2nwr5nmrmoa)

[Caso 4: Eliminar Evento 9](#_jdwklwlsi67y)

[Caso 5: Añadir galería. 10](#_ybdaa5mg5ign)

[Caso 6:Eliminar galería. 11](#_vcy5if3au8u6)

[Caso 7:Añadir favorito 12](#_vhryw2f9ucr6)

[Caso 8: Eliminar favorito. 13](#_79tu5iw78kld)

[**Base de Datos 14**](#_dwfwn0mrwron)

[Creación Base de Datos 15](#_glri3xz5fjbf)

[Añadir Datos 16](#_pzxu7ab8ec4c)

[Vistas: 17](#_1kqmy08jgx30)

[● Vista que Devuelvan las galerías anteriores al 28-02-2025: 17](#_rowk1or3svs6)

[● Vista que Devuelvan los eventos favoritos del usuario 1. 17](#_vu00vqrlv6e6)

[● Vista que devuelvan las imágenes de la galería del evento del 12-01-2025(usar su id para crear la vista, no la fecha). 17](#_9ksblsypi490)

[● Vista que Devuelvan los eventos favoritos del usuario 2 posteriores al 28-02-2025. 17](#_wj6jkwfhp6mz)

[**Lenguaje de Marcas 19**](#_xs6g4oqvb665)

[Bocetos. 19](#_csur1bzgg6db)

[● Boceto Página Index: 19](#_nwlknshvc9am)

[● Boceto Páginas de los Eventos 20](#_pq8rwmlncwjp)

[● Boceto Página Sobre Nosotros 21](#_3q521zco7itr)

[● Boceto Página Formulario 21](#_mhf5w2nb1iwz)

[● Boceto Página Fuentes 22](#_hybck7c3my03)

[RSS 24](#_9cynqgeq6vf7)

[**Conclusión 25**](#_a1ffjqvrv3l6)

[**Bibliografía 27**](#_dprvsbmwib97)

# Introducción

HAPPINESS Co, una empresa joven dedicada al sector del ocio, busca desarrollar un portal que reúna todos los eventos de la ciudad en una agenda cultural, facilitando su difusión y conocimiento. Esta empresa encomendó la realización de un proyecto que consiste en el desarrollo de un portal web que recopile y organice eventos culturales (como música, teatro y exposiciones), junto con las bases de datos, la programación en Java y un canal RSS necesarios para su funcionamiento. Para su realización, se estableció una división en cuatro partes principales: Lenguaje de Marcas, RSS, Base de Datos y Programación. Esta memoria detalla los pasos seguidos en cada sección, desde la investigación inicial hasta la implementación, incluyendo capturas de los procesos, así como los retos encontrados y las soluciones aplicadas para cumplir con las especificaciones establecidas.

# Programación (Java)

El desarrollo del proyecto HAPPINESS Co se inició con la sección de Programación, implementada en Java conforme a las especificaciones establecidas. Esta etapa se seleccionó como punto de partida por ser un área consolidada, lo que facilitó su ejecución y permitió comprender mejor los conceptos necesarios para la posterior creación de bases de datos.

Consistió en la creación de las clases necesarias (Usuarios, Eventos, Galerías y Favoritos), en cada una de las clases fueron creados sus atributos, constructores, los métodos getter y setter y el método toString correspondiente a cada uno. También se pidió la creación una clase principal que integra un menú interactivo mediante un bucle Do-While para trabajar con los diferentes métodos que fueron solicitados.

La implementación de la clase principal, que sirve como núcleo del programa, se inició con el diseño de colecciones específicas para la gestión de datos. Se empleó un HashMap para almacenar los Usuarios, utilizando el email como clave única para garantizar su identificación; otro HashMap para los Eventos, con el id como elemento diferenciador; y un ArrayList para los Favoritos, estructurado para registrar de manera dinámica las preferencias de los usuarios.

## 

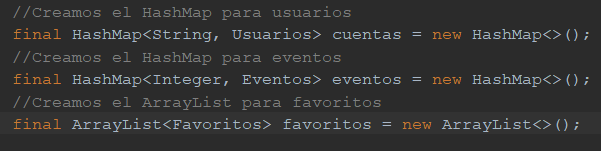
## 

## 

## 

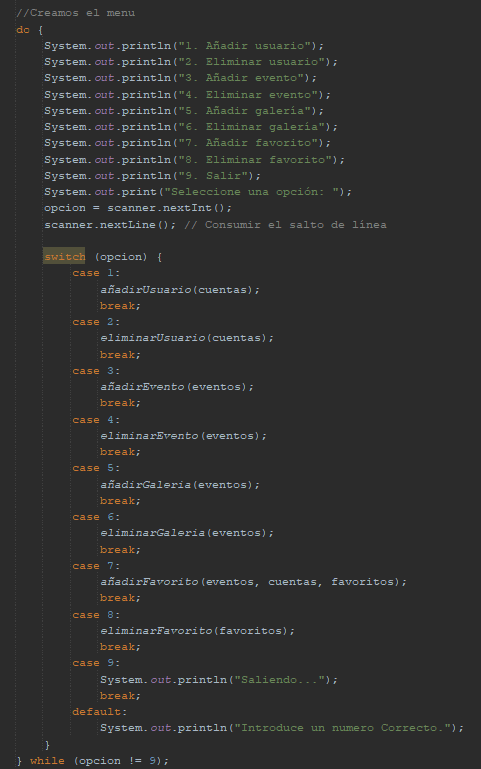
## 

## Creación Hashmap y ArrayList



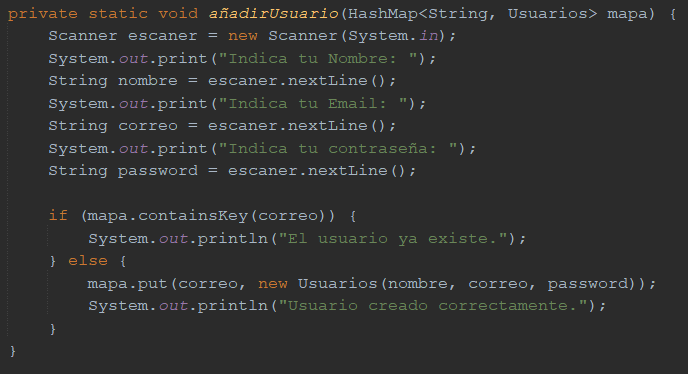
Una vez concluida la gestión de datos, se procedió a desarrollar la interacción con el usuario, organizada mediante un menú interactivo implementado con un bucle Do-While. Este mecanismo, diseñado para mostrar las opciones al inicio de cada iteración, ofrece nueve funcionalidades: añadir y eliminar usuarios, eventos, galerías y favoritos, además de una opción para finalizar el programa las cuales serán mostradas y explicadas a continuación.

## Menú



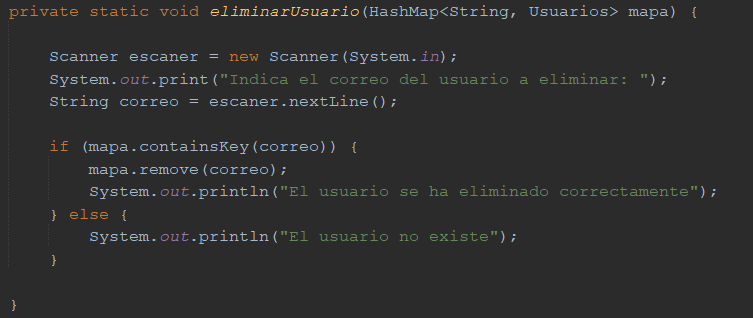
Finalizada la integración del menú en el programa, se procedió al desarrollo de los métodos correspondientes a dicho menú:

## Caso 1: Añadir Usuarios



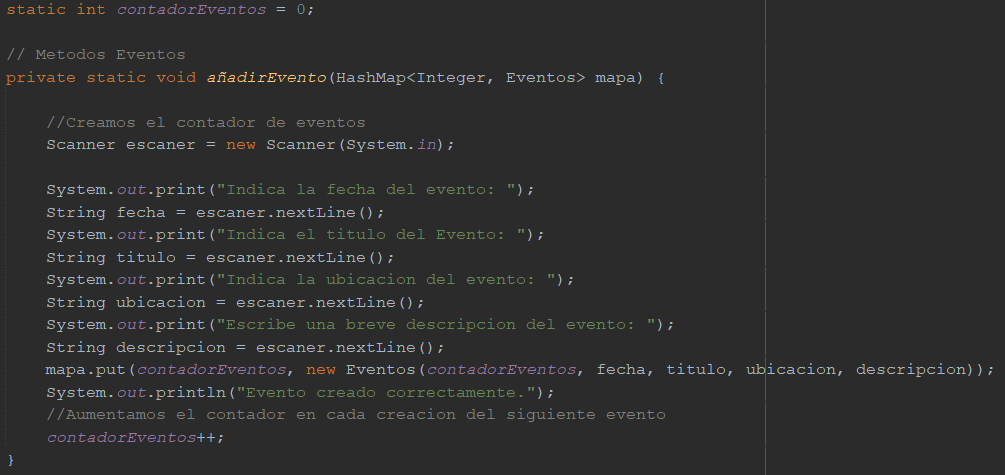
En este método, se gestiona la creación de un nuevo usuario solicitando nombre, email y contraseña por consola. Antes de añadir el objeto al HashMap, se verifica si el email ya existe como clave, mostrando 'El usuario ya existe' en caso afirmativo o 'Usuario creado correctamente' tras su inserción.

## Caso 2: Eliminar Usuario



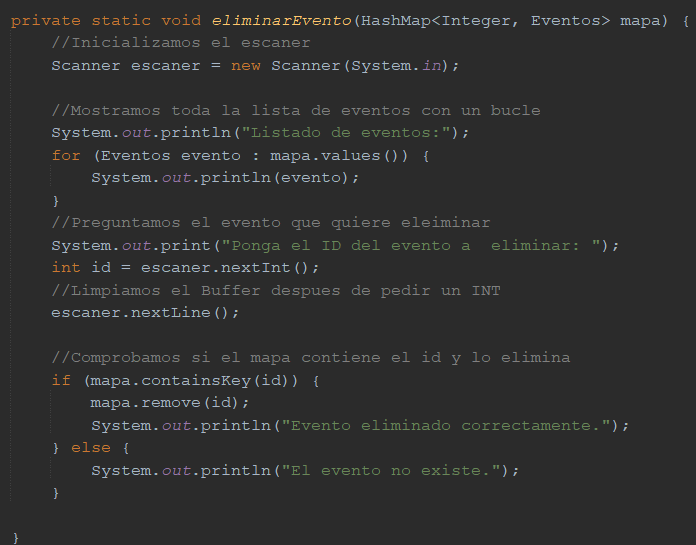
Este método pide al usuario un correo por consola y lo busca en el HashMap de usuarios. Si el correo coincide con una clave existente, se elimina el usuario y se muestra 'Usuario eliminado correctamente'; de lo contrario, se notifica 'El usuario no existe'

## Caso 3: Añadir evento



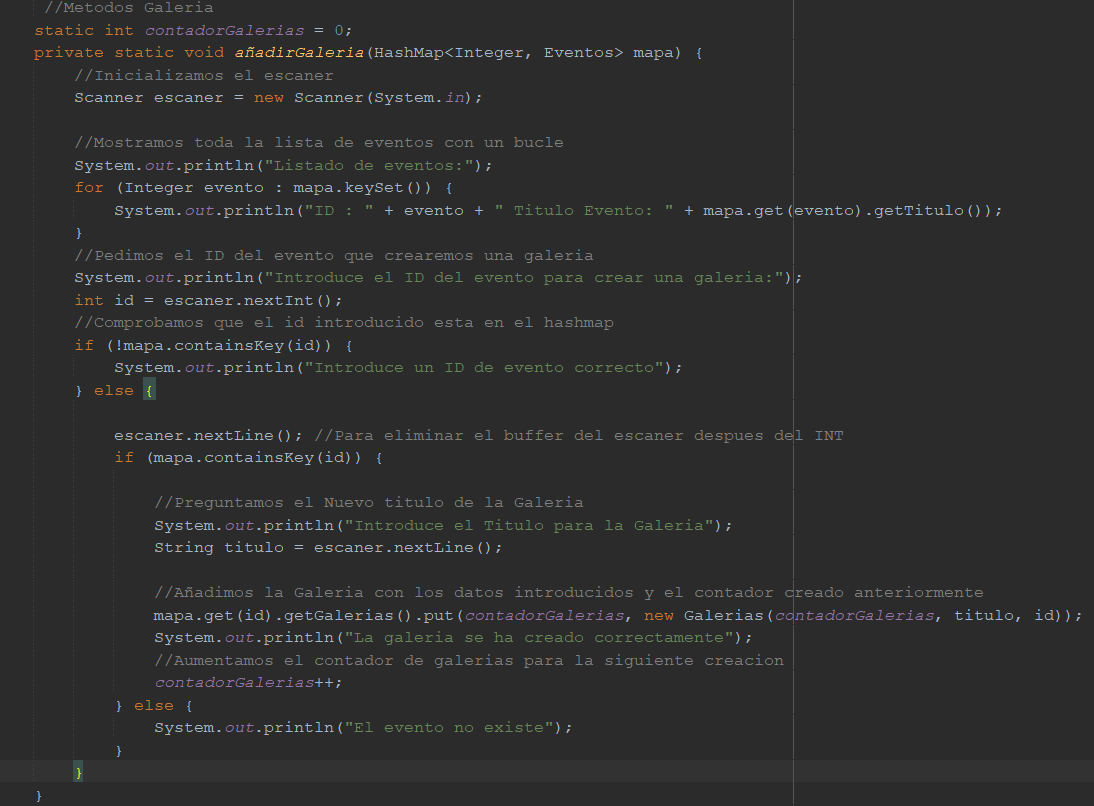
Este método , recoge por consola los datos del evento (fecha, título, ubicación y descripción), generando un id único mediante un contador estático que se incrementa por cada nuevo evento. El objeto se añade al HashMap de eventos con su colección de galerías inicializada vacía, notificando 'Evento creado correctamente' al finalizar. El contador garantiza identificadores únicos.

## Caso 4: Eliminar Evento



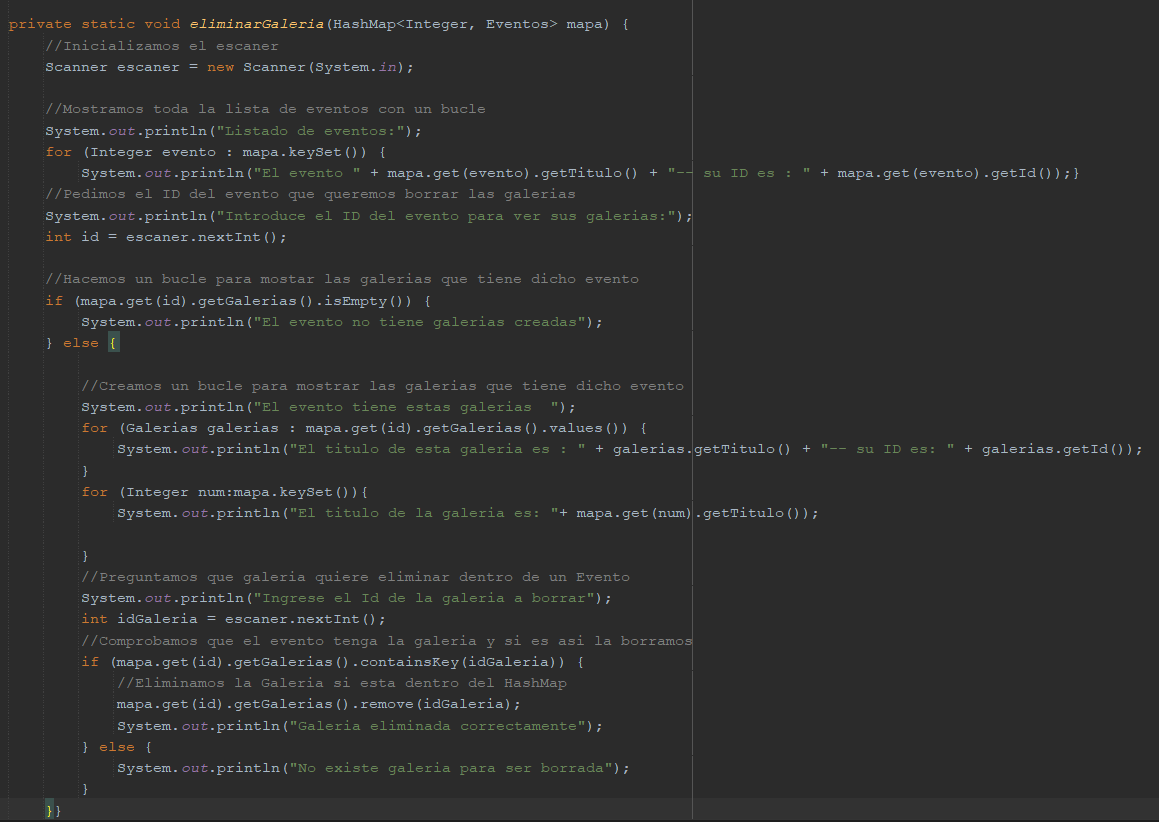
En este método, se gestiona la eliminación de un evento mostrando la lista del HashMap en un bucle. El usuario introduce el id; si existe, se elimina y se notifica 'Evento eliminado correctamente'; si no, se indica 'El evento no existe'. El bucle facilita la visualización previa de las opciones.

## Caso 5: Añadir galería.



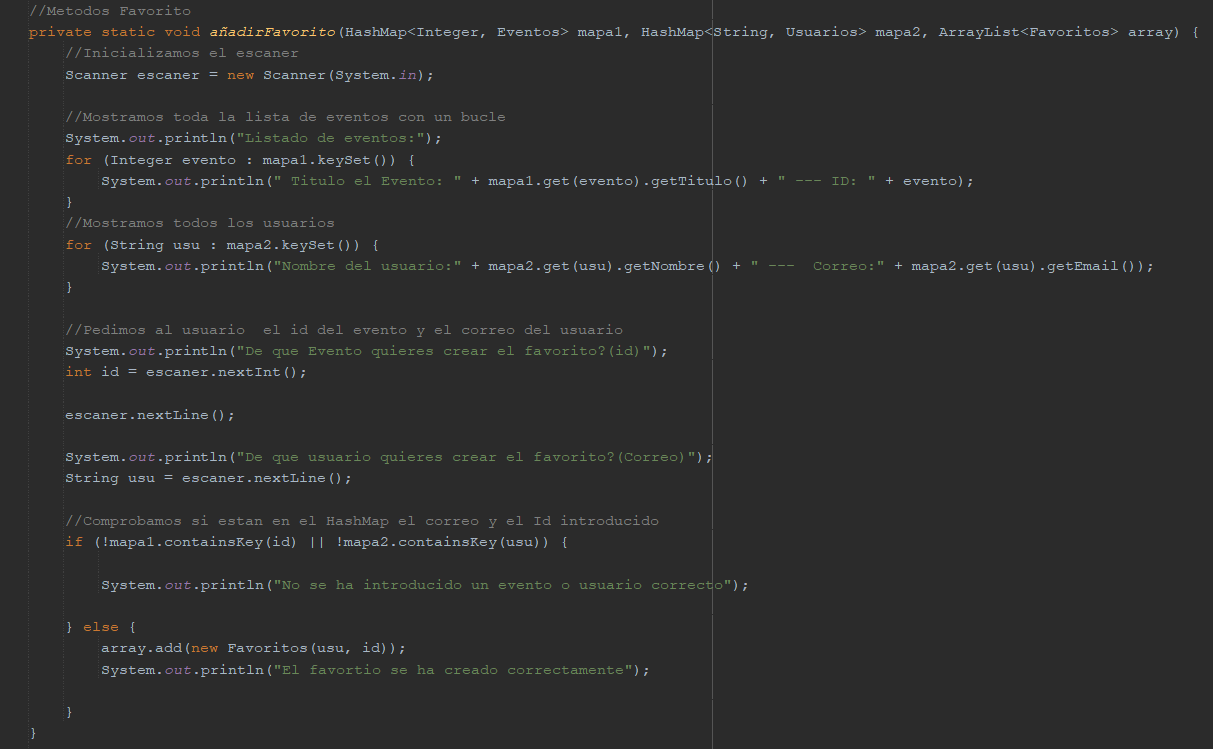
En este método, se gestiona la creación de una galería mostrando los eventos del HashMap en un bucle. Si el id ingresado es válido, se solicitan datos, se genera un id único con un contador y se añade al ArrayList del evento, mostrando 'Galería creada correctamente'.

## Caso 6:Eliminar galería.



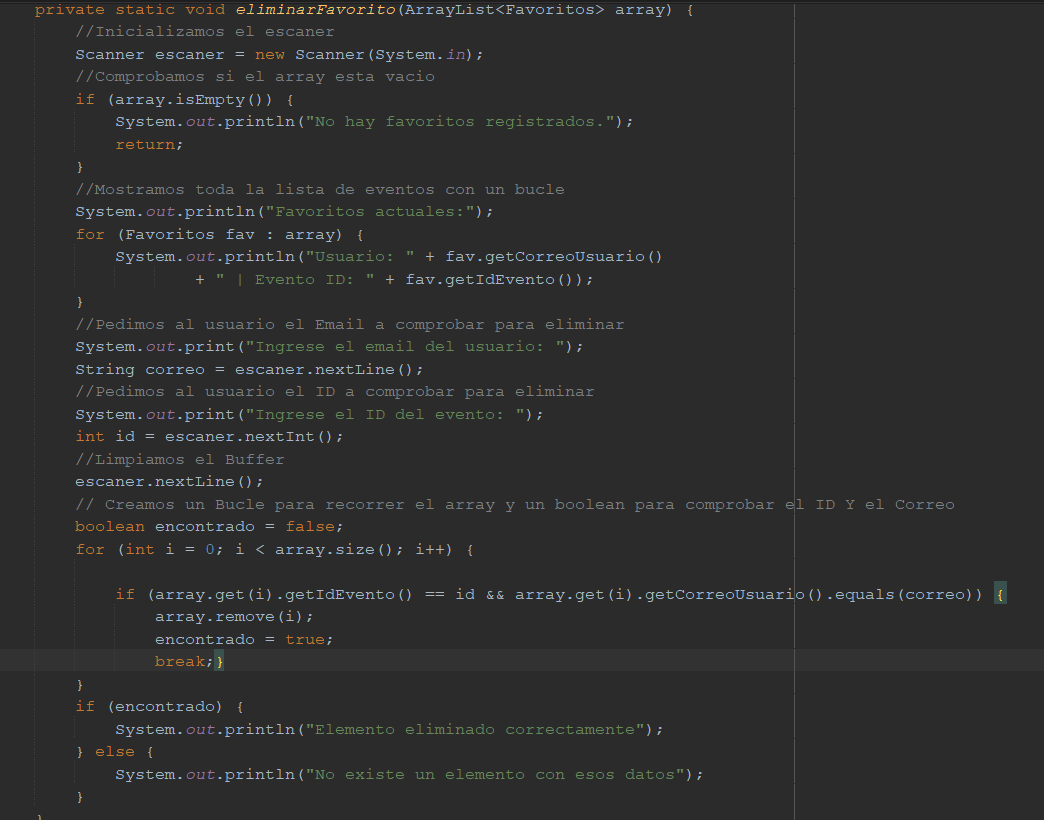
En este método, se gestiona la eliminación de una galería presentando en un bucle todos los eventos con su título e id. Se solicita al usuario el id del evento a revisar; si no tiene galerías, se muestra un mensaje de advertencia. Si las tiene, un segundo bucle lista sus galerías con id y título, pidiendo el id a eliminar. Si existe en el ArrayList, se suprime y se notifica 'Galería eliminada correctamente'; si no, se indica 'La galería no existe'.

## Caso 7:Añadir favorito



En este método, se gestiona la adición de un favorito mostrando en bucles todos los eventos y usuarios disponibles. Se solicita al usuario el id del evento y el correo del usuario; si ambos son válidos, se crea el objeto Favorito y se añade al ArrayList, notificando 'Favorito creado correctamente'. Si algún dato es incorrecto, se muestra un mensaje de error.

## Caso 8: Eliminar favorito.

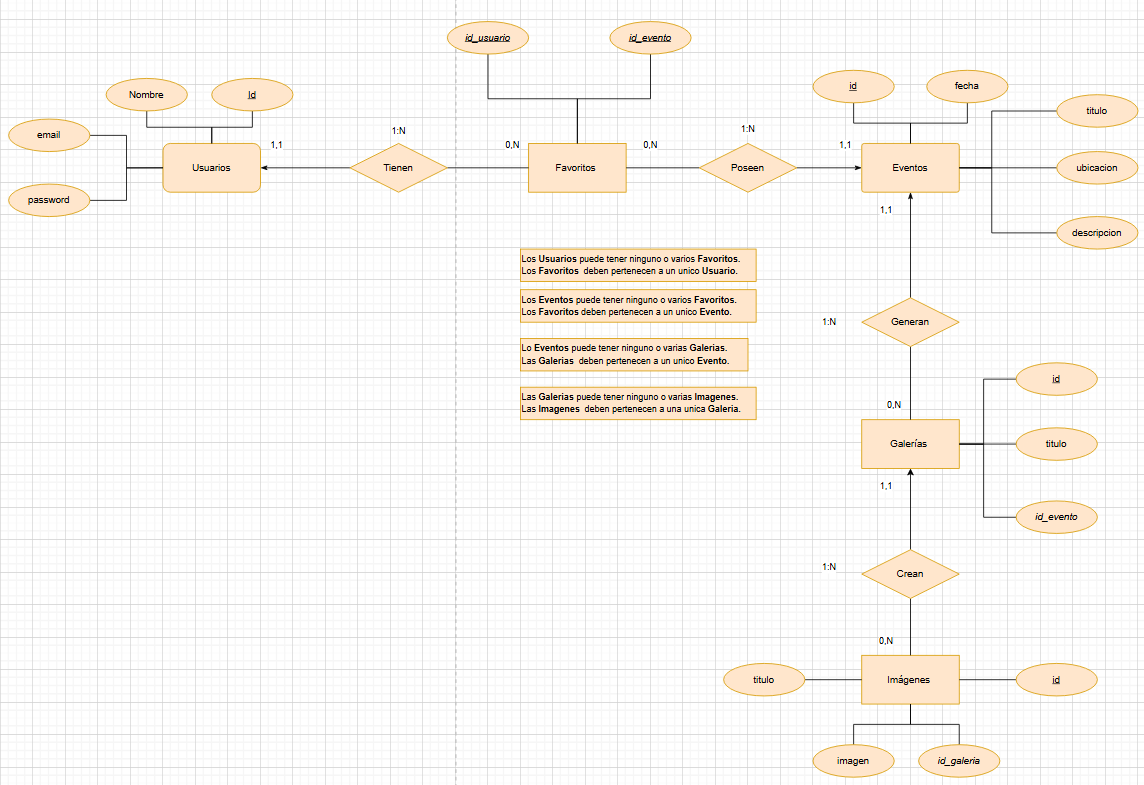


En este método, se gestiona la eliminación de un favorito verificando inicialmente si el ArrayList no está vacío mostrando el mensaje “No hay favoritos registrados ”. Si contiene elementos, se muestra la lista de favoritos con su correo de usuario e id de evento mediante un bucle. Se solicita al usuario un correo y un id; con un boolean como indicador, se recorre la colección y, si coinciden, se elimina el favorito, notificando 'Favorito eliminado correctamente'. Si los datos son erróneos, se indica 'El favorito no existe'.

**Dificultades encontradas:**  
Uno de los retos más importantes fue la gestión correcta del HashMap de usuarios, especialmente la verificación de correos únicos antes de insertar un nuevo usuario , para comprobar que no tenga duplicados. Otro desafío surgió en la eliminación de favoritos, requirió un esfuerzo adicional debido a la necesidad de recorrer el ArrayList de forma eficiente. Para resolverlo, se recurrió a ejemplos en StackOverflow, donde se encontraron modelos de bucles y el uso de un boolean como indicador. La integración de estas soluciones fortaleció la robustez del programa, aunque demandó tiempo para ajustar y probar cada implementación.

# Base de Datos

Concluida la sección de Programación, se procedió a desarrollar la parte de Bases de Datos, una etapa que marcó un paso lógico tras consolidar la lógica de gestión de datos en Java. Inicialmente, se elaboró un modelo entidad-relación en Draw.io. Este diseño sirvió como guía para crear una base de datos coherente y funcional en SQL Server. La experiencia previa en programación resultó clave, ya que los métodos implementados previamente sugerían las asociaciones necesarias, lo que agiliza el proceso y aseguró que las tablas cumplieran con las especificaciones del proyecto HAPPINESS Co.



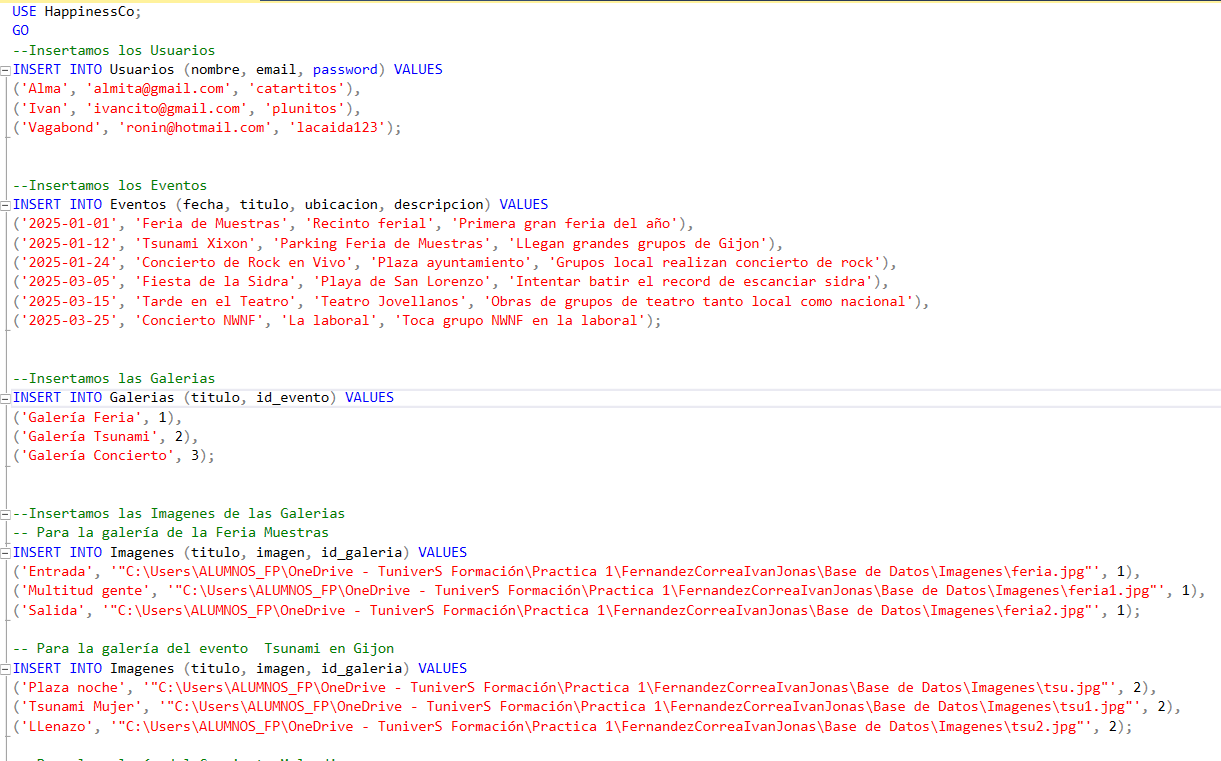
Una vez finalizado el modelo entidad-relación y con una visión más clara de la estructura de la base de datos gracias a su diseño, se dio inicio a la creación de la base de datos y sus tablas, tal como se muestra a continuación.

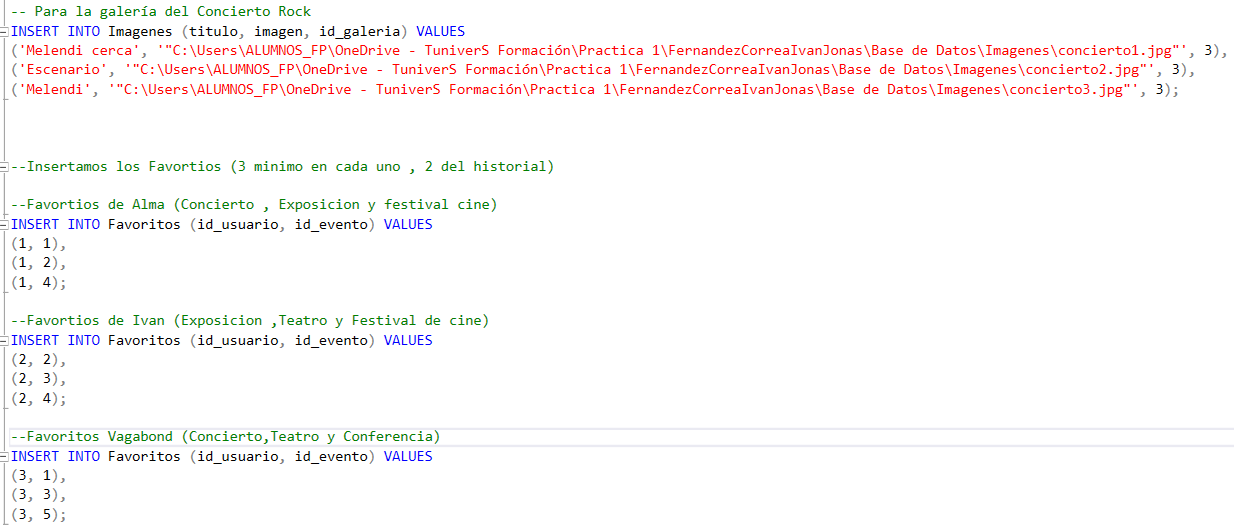
## Creación Base de Datos



Tras haber creado y verificado que todas las tablas se generaron correctamente, se procedió a incorporar los datos a cada una de ellas, asegurando el cumplimiento de las premisas establecidas. Para ello, se seleccionaron eventos que posteriormente serán utilizados en la página web, garantizando coherencia entre las distintas partes del proyecto. Asimismo, se generaron usuarios ficticios para completar los datos requeridos.

## Añadir Datos

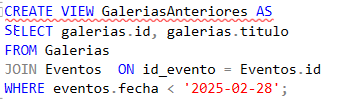




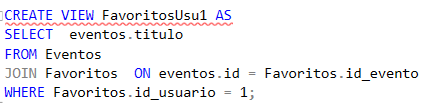
Para finalizar la última etapa de la creación de la base de datos, se procedió a desarrollar las vistas requeridas por el proyecto.

## Vistas:

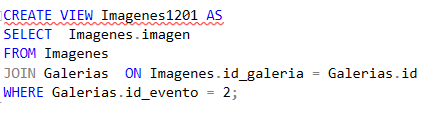
### Vista que Devuelvan las galerías anteriores al 28-02-2025:



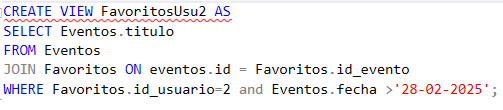
### Vista que Devuelvan los eventos favoritos del usuario 1.



### Vista que devuelvan las imágenes de la galería del evento del 12-01-2025(usar su id para crear la vista, no la fecha).



### Vista que Devuelvan los eventos favoritos del usuario 2 posteriores al 28-02-2025.



**Dificultades encontradas:**  
Uno de los retos al desarrollar la base de datos fue armar el modelo entidad-relación, que se necesitaba para que las vistas cumplieran con lo que se pedía. Como solo se dieron las pautas de las tablas, hacer el modelo se complicó un poco, ya que había que deducir cómo se relacionaban las entidades. Una vez que estuvo listo, el modelo ayudó bastante a las siguientes partes de la base de datos, como crear las tablas y las vistas. Tener las relaciones bien definidas fue clave, porque si algo estaba mal, las vistas no iban a mostrar los datos como se esperaba.

# 

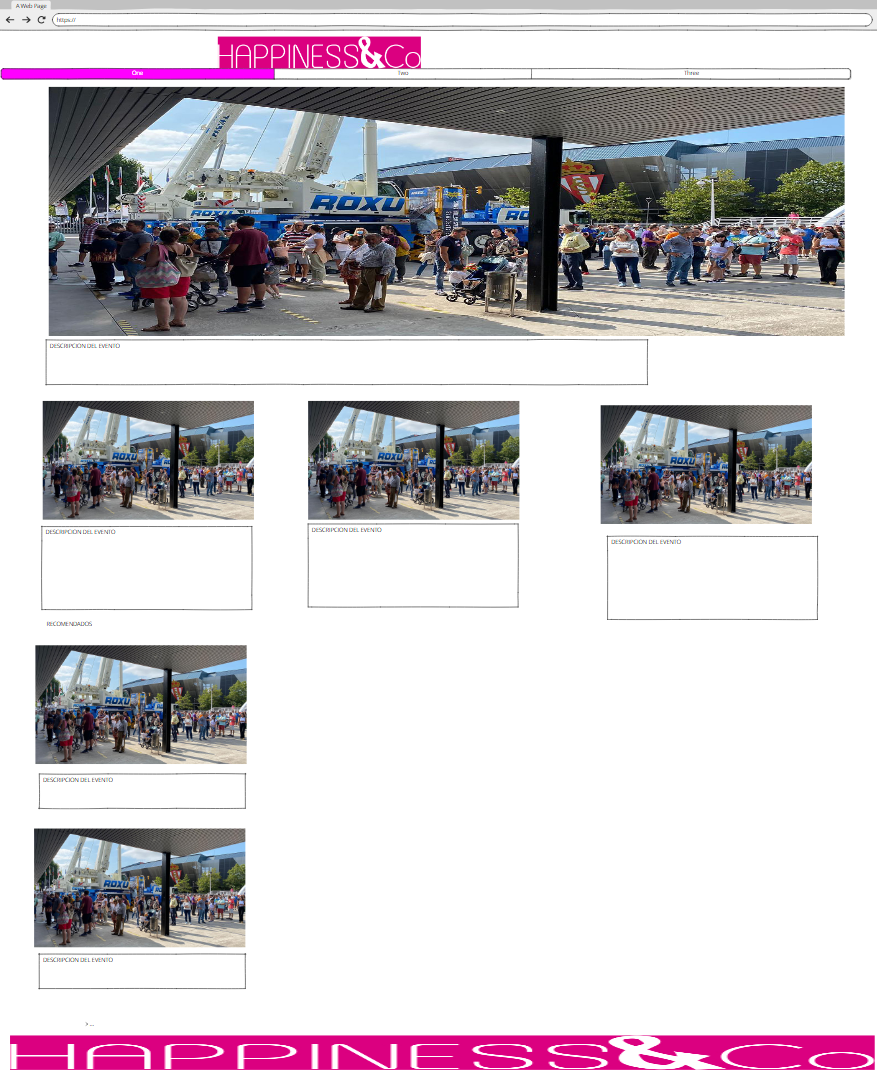
# Lenguaje de Marcas

# 

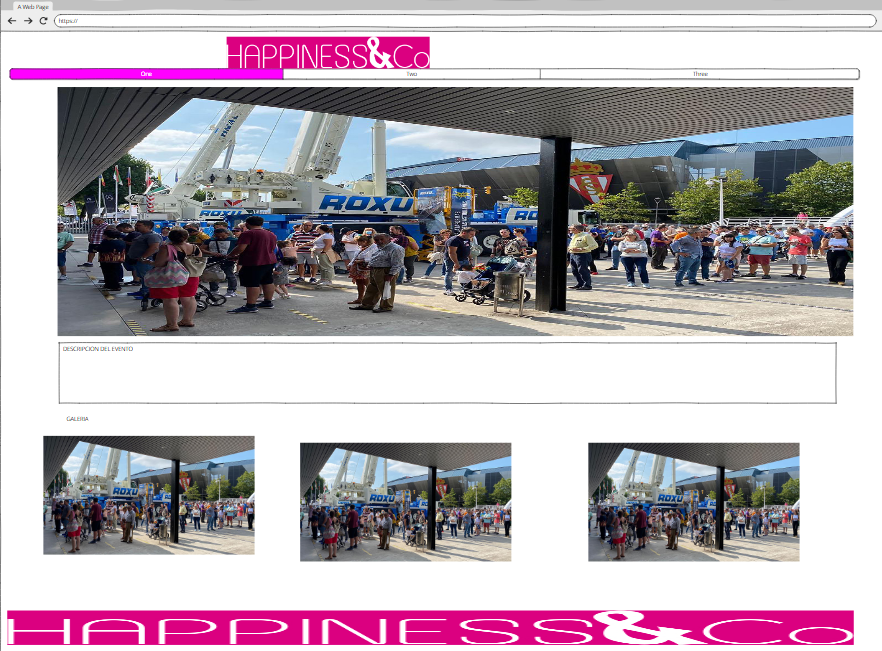
Por último, se llevó a cabo la realización de la parte de Lenguaje de Marcas, que resultó ser la más extensa y complicada del proyecto, debido a la menor experiencia en desarrollo web, especialmente en lo relacionado con estilos y maquetación de páginas. Para abordar esta etapa, se elaboraron unos bocetos iniciales que reflejaban una idea general de cómo debía verse la página, sirviendo como guía para su diseño. Además, se buscaron ideas en otras páginas web que ayudaron a mejorar el resultado final. A continuación, se presentan los resultados obtenidos:

## Bocetos.

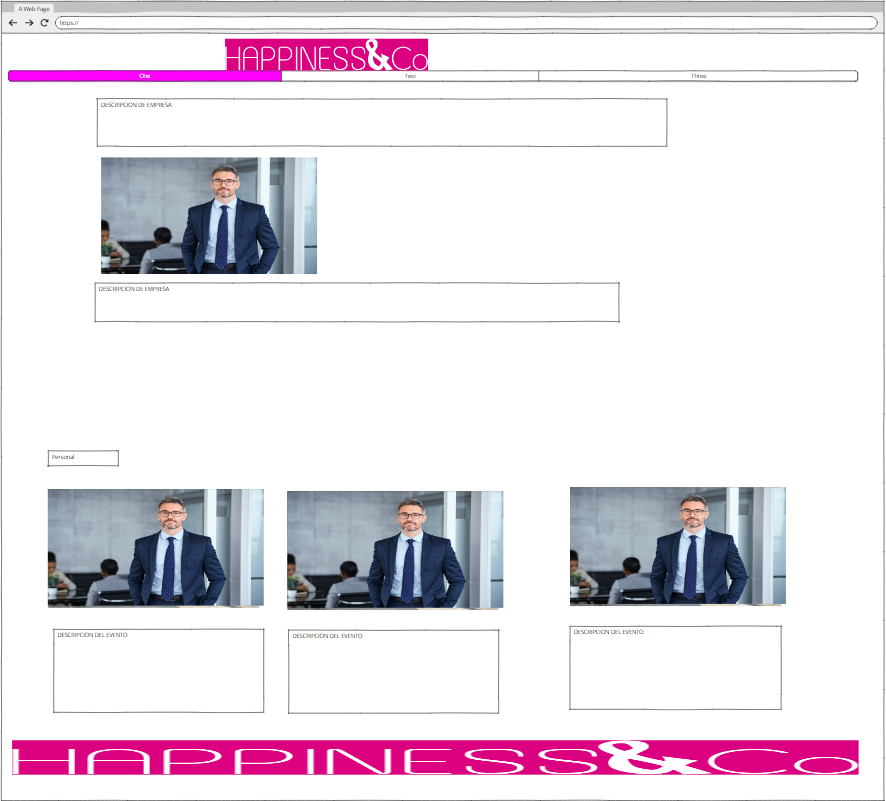
### Boceto Página Index:



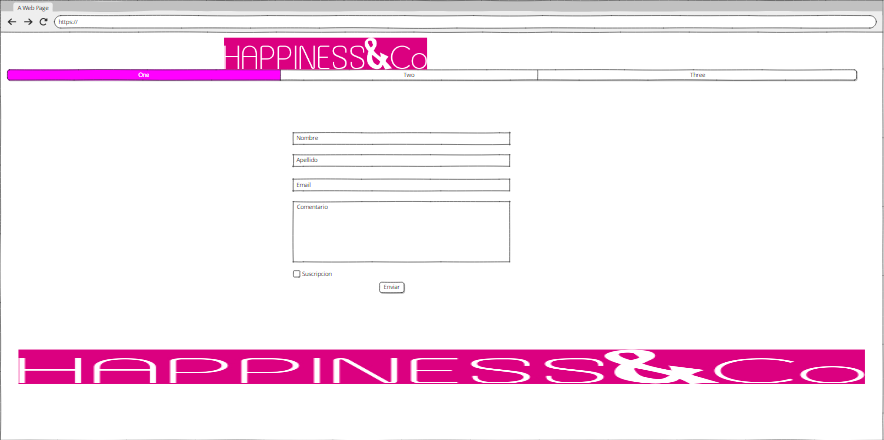
### Boceto Páginas de los Eventos



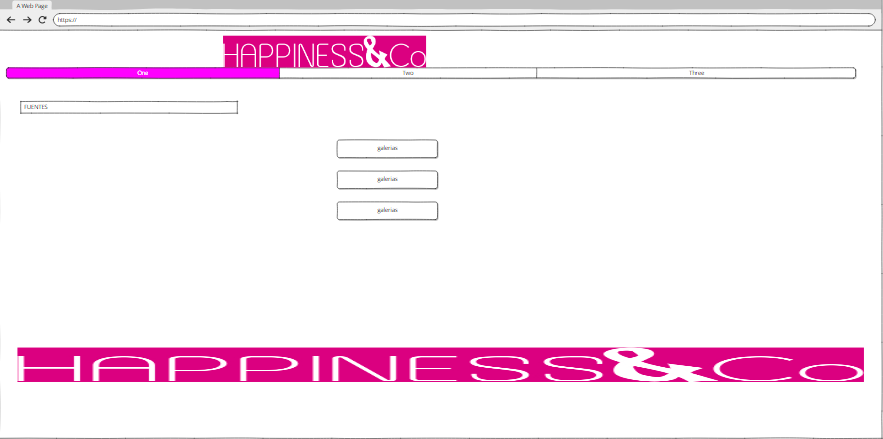
### Boceto Página Sobre Nosotros



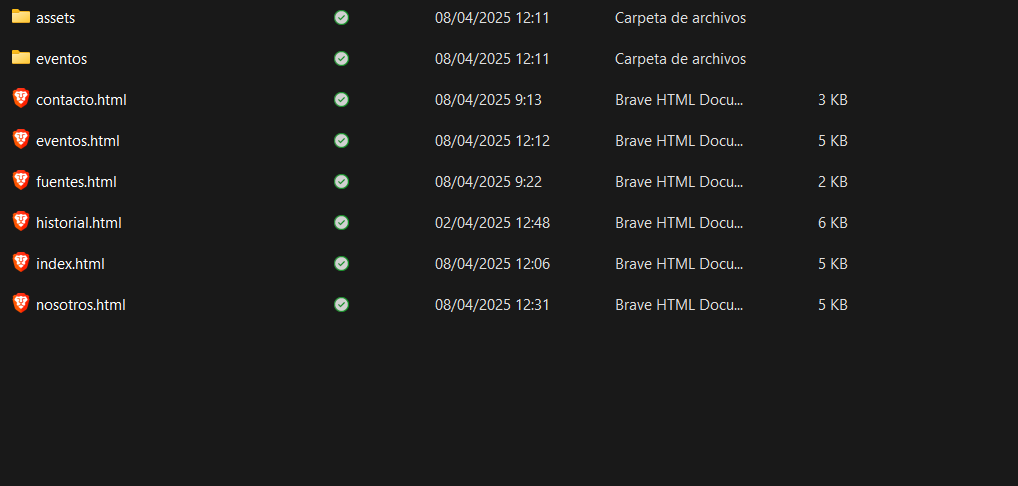
### Boceto Página Formulario



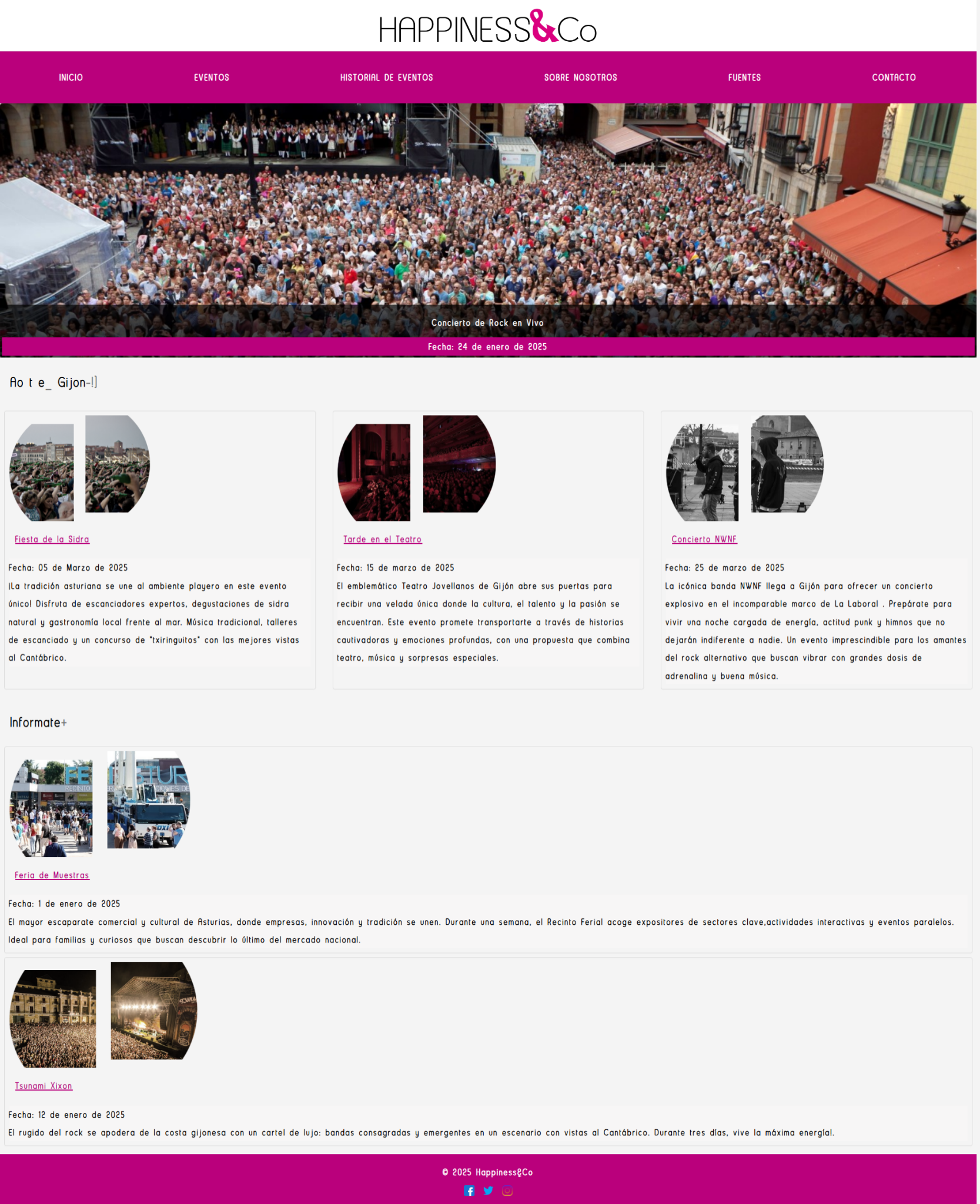
### Boceto Página Fuentes



Una vez definida la idea principal de cómo debía verse la página, considerando que los bocetos solo ofrecían una visión general y gracias a las recomendaciones del tutor de prácticas, Jonhy, sobre cómo organizar las carpetas y estructurar el proyecto, se procedió a la creación y organización en carpetas de todo el contenido del proyecto y de la página, así como de las relaciones entre sus elementos. A continuación, se muestra la distribución del proyecto: la carpeta 'assets' incluye las imágenes, el código CSS y los scripts utilizados por la página, mientras que la carpeta 'eventos' aloja las páginas de cada evento, facilitando la posible incorporación de nuevos eventos en el futuro.

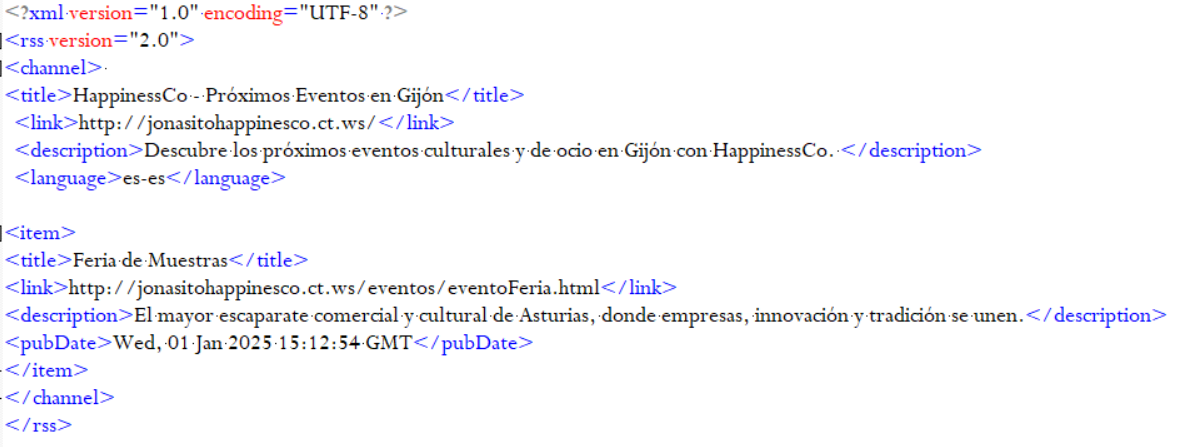


Una vez finalizada la creación de las páginas web y su rellenado de datos y relaciones entre las páginas, se continuó con el diseño de la página, explorando ejemplos e ideas de CSS en diversas páginas para obtener inspiración e implementarlas en el proyecto lo que mejoró enormemente el diseño del mismo. Este proceso resultó ser el más extenso, ya que implicó una investigación exhaustiva del diseño, que definió la apariencia final de la página web. Al analizar ejemplos de CSS, el boceto inicial y la idea final de la página variaron en algunos aspectos, debido a la integración de código CSS proveniente de sitios como 'Codepen' y plantillas de 'Free-CSS', de donde se tomaron algunos elementos para incorporarlos y dar forma al diseño final del proyecto. Una vez aplicadas las ideas investigadas y las plantillas, la página principal quedó configurada como se muestra a continuación, mientras que el resto de las páginas adoptaron este estilo principal, adaptándose cada una a sus objetivos para cumplir con las pautas establecidas.



## RSS

Una vez terminado todo lo relacionado con la página web, incluyendo las relaciones y el contenido de todas las páginas, la organización de los archivos del proyecto, y tras verificar que las implementaciones de CSS y JavaScript funcionaban correctamente, se dio por concluida la sección de Lenguaje de Marcas con la creación de las RSS de la página. Mostrada a continuación:



En la imagen se muestra la estructura utilizada para la creación de las RSS, incluyendo un ejemplo de un evento extraído de la página, con su título, fecha, descripción y enlace correspondiente.

**Dificultades encontradas:**

Durante la creación de la página, se investigaron ideas y plantillas de JavaScript para incorporarlas al proyecto. Este proceso resultó ser uno de los mayores desafíos al diseñar la página, ya que la correcta integración de estos elementos de JavaScript tomó más tiempo del esperado. Entre las integraciones de JavaScript implementadas en el proyecto, se incluyó un script que muestra texto 'dinámico' cambiante sobre eventos próximos y pasados actualizándose cada pocos segundos para captar la atención y hacer la página más interactiva. Además, se añadió un botón que, al hacer clic, muestra eventos recomendados en las páginas de eventos pasados con galería, como se detalla a continuación.



# 

# Conclusión

El desarrollo del proyecto ha representado un pequeño reto enriquecedor que me ha ayudado a consolidar mis conocimientos en los diversos puntos, programación , base de datos pero sobre todo en la creación de las páginas web ya que es mi punto más flojo ya que no tengo muchos conocimientos sobre la creación de las páginas.

# Bibliografía

1. [Codepen](https://codepen.io/trending)
2. [StackOverflow](https://stackoverflow.com/questions)
3. [Balsamiq](https://balsamiq.com/)
4. <https://blog.hubspot.es/website/maquetacion-web>
5. https://developer.mozilla.org/es/docs/Learn\_web\_development/Getting\_started/Your\_first\_website/Adding\_interactivity

# 