



UNIVERSIDAD DE BURGOS
ESCUELA POLITÉCNICA SUPERIOR
Grado en Ingeniería Informática



**TFG del Grado en Ingeniería
Informática**

**ANME - Aplicación web de
noticias, meteorología y
eventos**



Presentado por José Manuel Rodríguez Iglesias
en Universidad de Burgos — 8 de junio
de 2023

Tutores: Dr. Alfredo Bol Arreba, Dr. Bruno
Baruque Zanon y D. Héctor Cogollos Adrian

Índice general

Índice general	i
Índice de figuras	iii
Índice de tablas	v
Apéndice A Plan de Proyecto Software	1
A.1. Introducción	1
A.2. Planificación temporal	1
A.3. Estudio de viabilidad	9
Apéndice B Especificación de Requisitos	15
B.1. Introducción	15
B.2. Objetivos generales	15
B.3. Catálogo de requisitos	16
B.4. Especificación de requisitos	18
Apéndice C Especificación de diseño	35
C.1. Introducción	35
C.2. Diseño de datos	35
C.3. Diseño procedimental	37
C.4. Diseño arquitectónico	39
C.5. Diseño de interfaces	41
Apéndice D Documentación técnica de programación	47
D.1. Introducción	47
D.2. Estructura de directorios	47

D.3. Manual del programador	48
D.4. Compilación, instalación y ejecución del proyecto	49
D.5. Pruebas del sistema	50
Apéndice E Documentación de usuario	59
E.1. Introducción	59
E.2. Requisitos de usuarios	59
E.3. Instalación	59
E.4. Manual del usuario	60
Bibliografía	71

Índice de figuras

A.1. Informe del Sprint 1	2
A.2. Tareas Sprint 1	2
A.3. Informe del Sprint 2	3
A.4. Tareas Sprint 2	3
A.5. Informe del Sprint 3	4
A.6. Tareas Sprint 3	4
A.7. Informe del Sprint 4	5
A.8. Tareas Sprint 4	5
A.9. Informe del Sprint 5	6
A.10.Tareas Sprint 5	6
A.11.Informe del Sprint 6	7
A.12.Tareas Sprint 6	7
A.13.Informe del Sprint 7	8
A.14.Tareas Sprint 7	8
A.15.Informe del Sprint 8	8
A.16.Tareas Sprint 8	9
 B.1. Diagrama de Casos de Uso	18
 C.1. Esquema relacional de la base de datos	36
C.2. Diagrama de Secuencia de Login y Registro de usuarios	38
C.3. Diagrama de secuencia del servicio Noticias	38
C.4. Diagrama de secuencia del servicio Eventos	39
C.5. Diagrama de secuencia del servicio Meteorología	39
C.6. Patrón Modelo-Vista-Presentador. [2]	41
C.7. Prototipo de Registro	42
C.8. Prototipo de Login	42
C.9. Prototipo de Principal	43

C.10. Prototipo de Noticias	43
C.11. Prototipo de Eventos	44
C.12. Prototipo de Información del Evento	44
C.13. Prototipo de Meteorología	45
C.14. Prototipo de Eventos Favoritos	45
D.1. Cloud Build. Podemos observar la lista de compilaciones a lo largo del tiempo y los commits.	57
D.2. Visualización del resultado del Trigger en el repositorio de GitHub de nuestro proyecto.	57
E.1. Página de Inicio	60
E.2. Página de Registro	61
E.3. Barra de Navegación	61
E.4. Página principal	62
E.5. Página de Noticias	63
E.6. Buscar Noticias por ubicación	63
E.7. Buscar Noticias por categoría	64
E.8. Página de Eventos	64
E.9. Buscar eventos por ubicación	65
E.10. Buscar eventos por categoría	65
E.11. Barra de Navegación	65
E.12. Página de Información del Evento	66
E.13. Página de Meteorología	67
E.14. Página de Meteorología	67
E.15. Lista de Eventos Favoritos	68
E.16. Meteorología de Eventos Favoritos	68
E.17. Mapa de Eventos Favoritos	69

Índice de tablas

A.1. Licencias de las librerías y herramientas	13
B.1. CU-1 Inscripción de Usuarios	19
B.2. CU-2 Inicio sesión	20
B.3. CU-3 Mostrar noticias	21
B.4. CU-3 Consultar una noticia	22
B.5. CU-5 Mostrar eventos	23
B.6. CU-6 Consultar evento	24
B.7. CU-7 Añadir eventos a favoritos	25
B.8. CU-8 Buscar noticias por Ubicación	26
B.9. CU-9 Buscar noticias por Categorías	27
B.10. CU-10 Buscar eventos por Ubicación	28
B.11. CU-11 Buscar eventos por Categorías	29
B.12. CU-12 Mostrar Meteorología	30
B.13. CU-13 Buscar Meteorología por Ubicación	31
B.14. CU-14 Mostrar Eventos Favoritos	32
B.15. CU-15 Eliminar eventos de favoritos	33
D.1. Pruebas de Verificación de Requisitos	56

Apéndice A

Plan de Proyecto Software

A.1. Introducción

En esta sección de los anexos se ha abordado la descripción detallada del progreso y la viabilidad del proyecto, divida en dos aspectos principales: la viabilidad legal, que implica considerar la legislación que podría impactar nuestro proyecto, y la viabilidad económica, donde hemos llevado a cabo una estimación de los costos asociados con nuestro proyecto.

A.2. Planificación temporal

Para la planificación temporal de nuestro proyecto hemos utilizado la *metodología SCRUM* [1]. Esta planificación temporal se ha basado en realizar sprints de una duración de dos semanas aproximadamente. Al término de cada sprint, se han llevado a cabo reuniones para evaluar el cumplimiento de los objetivos establecidos para dicho período. Además, en cada reunión se planteaban las tareas a realizar para el siguiente sprint.

La herramienta que hemos utilizado para la gestión del proyecto ha sido *Jira*, la cual provee un tablero que nos ha facilitado la organización del proyecto de manera eficiente.

Sprint 1 - 14/11/2022 - 28/11/2022

Durante la etapa inicial del sprint, hemos destinado la primera semana a realizar la configuración del repositorio del proyecto y la herramienta para la gestión de la planificación del proyecto. Además, se decidió qué entorno de desarrollo utilizaríamos para desarrollar nuestro proyecto.

En la segunda semana del sprint, se ha realizado la investigación de los diferentes servicios (APIs), de los cuales íbamos a extraer la información para nuestra aplicación web.

Finalmente, hemos realizado la implementación del servidor Flask.



Figura A.1: Informe del Sprint 1

Se puede observar que en el informe del Sprint 1 sale una línea horizontal de color naranja, debido a que no se le asignaron *story points* a las tareas planificadas para este sprint.

Informe de estado					
Incidencias terminadas					Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (-)
PROYEC-1	Crear repositorio para el desarrollo del proyecto del TFG	Tarea	Medium	FINALIZADA	-
PROYEC-2	Crear proyecto en Jira para el desarrollo del proyecto del TFG	Tarea	Medium	FINALIZADA	-
PROYEC-3	Investigar sobre Apis que traten información de eventos de ciudades en España	Tarea	Medium	FINALIZADA	-
PROYEC-4	Investigar sobre Apis que traten información de meteorología en España	Tarea	Medium	FINALIZADA	-
PROYEC-5	Investigar sobre Apis que traten información de noticias en España	Tarea	Medium	FINALIZADA	-

Figura A.2: Tareas Sprint 1

Sprint 2 - 29/11/2022 - 13/12/2022

Para este sprint se fijaron como objetivo obtener los casos de uso para nuestra aplicación y la conexión con las APIs.

Durante la primera semana de este sprint se fueron decidiendo los casos de usos que se iban a implementar en nuestra app.

Durante la segunda semana, lo primero que se hizo fue investigar cómo funcionaban las APIs con las que se iban a trabajar, que en nuestro caso han sido OpenWeatherMaps, Tu Tiempo, NewData.io, TicketMaster. Para cada una de las anteriores APIs fue necesario crearse una cuenta de desarrollador para poder tener nuestra API KEY propia y así podemos realizar llamadas a cada una de las APIs para solicitar información. Una vez que ya se nos proporcionó la API KEY de cada una de las APIs, se realizó una demo para cada una de ellas para ver cómo se devolvía la información solicitada.



Figura A.3: Informe del Sprint 2

Informe de estado						
					Ver en el navegador de incidencias	
Incidencias terminadas						
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (14)	
PROYEC-6	Realizar casos de usos	Historia	= Medium	FINALIZADA	4	
PROYEC-7	Crear demo de la api Ticketmaster	Historia	= Medium	FINALIZADA	3	
PROYEC-8	Crear demo de la api OpenWeather	Historia	= Medium	FINALIZADA	3	
PROYEC-9	Crear demo de la api NewsDataIO	Historia	= Medium	FINALIZADA	4	

Figura A.4: Tareas Sprint 2

Sprint 3 - 15/12/2022 - 05/02/2023

Durante las dos primeras semanas de este sprint, nos hemos enfocado en crear un prototipo de la interfaz de usuario de la aplicación a desarrollar, con el objetivo de obtener una visión clara de cómo avanzar en el desarrollo del proyecto. Una vez realizada la anterior tarea, se desarrolló una función en Python que nos devolviera la ubicación actual del usuario para así poder proporcionarle información relevante al usuario.

A partir de la tercera semana, se estuvo desarrollando la template de eventos y generando la construcción del mapa para poder colocar los eventos favoritos en él.

Por último, este sprint se tenía previsto que finalizará el día 18/01/2023, pero se decidió que finalizará el día 05/02/2023 por motivos personales de uno de mis tutores.

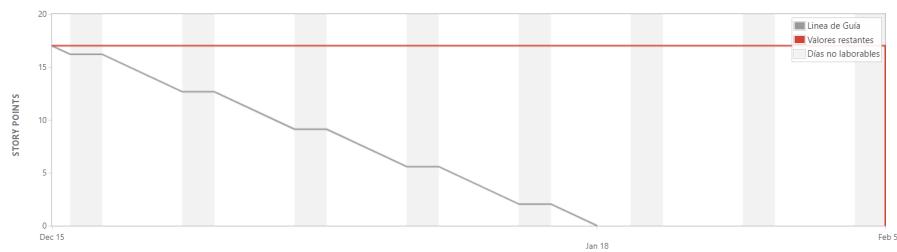


Figura A.5: Informe del Sprint 3

Informe de estado						
Incidencias terminadas					Ver en el navegador de incidencias	
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (17)	
PROYEC-11	Petición de Coordenadas	Historia	Medium	FINALIZADA	5	
PROYEC-12	Prototipo de la interfaz gráfica de la aplicación	Historia	Medium	FINALIZADA	5	
PROYEC-13	Template Mapa	Historia	Medium	FINALIZADA	4	
PROYEC-14	Template Eventos	Historia	Medium	FINALIZADA	3	

Figura A.6: Tareas Sprint 3

Sprint 4 - 05/02/2022 - 06/02/2023

En este sprint, hemos puesto especial atención en el desarrollo del Front-end de la aplicación. Para llevar a cabo este desarrollo se han utilizado tecnologías como HTML, CSS y el framework Bootstrap.

Durante este sprint se han desarrollado las templates para la parte de noticias, meteorología, login y registro de nuestra aplicación.

Se ha de comentar que este sprint ha tenido una duración de dos días debido a una confusión a la hora de crear el sprint en Jira, ya que para este sprint Jira decidió automáticamente la duración para el mismo.



Figura A.7: Informe del Sprint 4

Informe de estado					
Incidencias terminadas					
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (17)
PROYEC-19	Template Noticias	Historia	Medium	FINALIZADA	3
PROYEC-21	Template Login	Historia	Medium	FINALIZADA	4
PROYEC-22	Template Registro	Historia	Medium	FINALIZADA	4
PROYEC-23	Template Meterología	Historia	Medium	FINALIZADA	6

Figura A.8: Tareas Sprint 4

Sprint 5 - 07/02/2023 - 20/02/2023

En este sprint, se estableció como objetivo dedicar la primera semana a realizar un estudio sobre qué tablas íbamos a tener en nuestra base de datos y cómo se relacionarían entre ellas. Una vez que teníamos diseñado el esquema relacional de nuestra base de datos, se creó la base de datos con el fin de tener persistencia sobre los datos. Tras esto, se empezó a trabajar con la API de TicketMaster para poder extraer los eventos disponibles en España.

Después de haber realizado la tarea de extraer los eventos de la API de TicketMaster, se empezó a desarrollar la opción de poder guardar eventos en favoritos. Una vez que teníamos implementado esto, se implementó un mapa en el cual se muestran los eventos favoritos elegidos por el usuario.

Cuando se decidieron las tareas que se iban a llevar a cabo durante este sprint se estimaron con una dificultad un poco alta, pero realmente no fueron tan complejas como parecían, por eso se añadieron nuevas tareas y se modificó la estimación de cada una de las tareas añadidas a este sprint.



Figura A.9: Informe del Sprint 5

Informe de estado						* Incidencia agregada al sprint después de la fecha de comienzo
Incidencias terminadas						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (14 → 20)	
PROYEC-24	Almacenamiento de datos	Historia	= Medium	FINALIZADA	4 → 3	
PROYEC-25 *	Extracción de datos sobre eventos	Historia	= Medium	FINALIZADA	- → 7	
PROYEC-26 *	Guardar eventos favoritos	Historia	= Medium	FINALIZADA	4	
PROYEC-27 *	Generación del Mapa de Eventos	Historia	= Medium	FINALIZADA	6	

Figura A.10: Tareas Sprint 5

Sprint 6 - 27/03/2023 - 11/04/2023

Este sprint se ha centrado principalmente en documentar varios apartados de la memoria de este proyecto. Además, también se han realizado algunas mejoras en el Front-End.

Durante la primera semana y parte de la segunda, nos hemos enfocado en documentar diversos aspectos clave de nuestro proyecto, como los objetivos, las técnicas y herramientas utilizadas, los aspectos relevantes del desarrollo y los trabajos relacionados.

Por otra, se han realizado una serie de modificaciones en las templates de noticias y eventos.

A.2. Planificación temporal

7

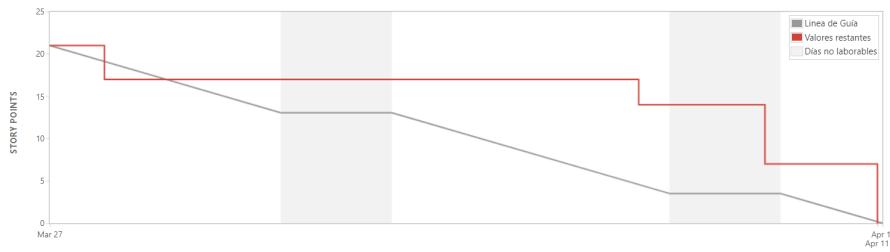


Figura A.11: Informe del Sprint 6

Informe de estado

Incidentes terminados		Ver en el navegador de incidentes			
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (21)
PROYEC-28	Memoria: Actualización de objetivos del proyecto	Historia	Medium	FINALIZADA	3
PROYEC-29	Memoria: Modificación del apartado Técnicas y Herramientas.	Historia	Medium	FINALIZADA	4
PROYEC-31	Memoria: Realizar apartado Aspectos_relevantes_del_desarrollo_del_proyecto	Historia	Medium	FINALIZADA	5
PROYEC-32	Memoria: Realizar apartado Trabajos_relacionados	Historia	Medium	FINALIZADA	5
PROYEC-33	Mejorar la template de la página de eventos de la aplicación	Historia	Medium	FINALIZADA	2
PROYEC-34	Mejorar la template de la pagina de noticias	Historia	Medium	FINALIZADA	2

Figura A.12: Tareas Sprint 6

Sprint 7 - 17/04/2023 - 02/05/2023

En este sprint se ha dedicado la mayor parte del tiempo a realizar el manual de usuario y el manual del programador del presente proyecto. También se ha realizado el cifrado de las contraseñas de nuestros usuarios en la base de datos para tener mayor seguridad en nuestra aplicación.

Durante la primera semana y parte de la segunda, se ha realizado el manual de usuario y el manual del programador. Una vez terminados los manuales, se continuó con el cifrado de contraseñas. Tras esto, se realizaron algunas mejoras sobre el apartado de la memoria Aspectos Relevantes.



Figura A.13: Informe del Sprint 7

Informe de estado						* Incidencia agregada al sprint después de la fecha de comienzo
Incidencias terminadas						Ver en el navegador de incidencias
Clave	Resumen	Tipo de Incidencia	Prioridad	Estado	Story Points (19)	
PROYEC-35	Memoria: Realizar Manual de Usuario	Historia	Medium	FINALIZADA	5	
PROYEC-36	Memoria: Realizar Manual de Programador	Historia	Medium	FINALIZADA	4	
PROYEC-37	Pasar a limpio los diagramas	Historia	Medium	FINALIZADA	3	
PROYEC-38 *	Corregir apartado de Aspectos Relevantes	Historia	Medium	FINALIZADA	3	
PROYEC-39 *	Cifrar las contraseñas de los usuarios	Historia	Medium	FINALIZADA	4	

Figura A.14: Tareas Sprint 7

Sprint 8 - 03/05/2023 - 15/05/2023

Este sprint se ha centrado principalmente en realizar el apartado de requisitos de los anexos e implementar la publicación de la aplicación en un servidor público para su uso. Además, se han realizado varias mejoras sobre el código de la aplicación y de la memoria.

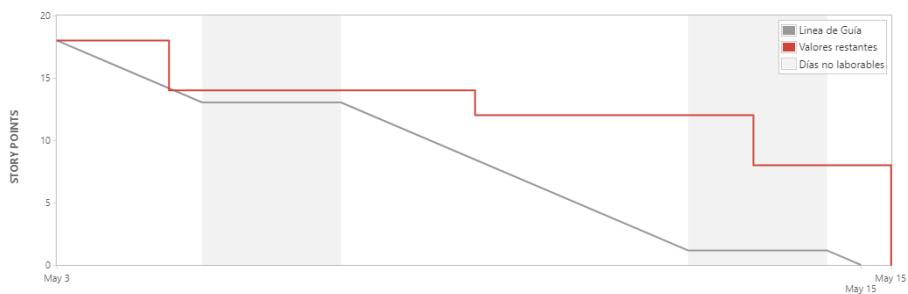


Figura A.15: Informe del Sprint 8

Incidentes terminados					Ver en el navegador de incidentes	
Clave	Resumen	Tipo de Incidente	Prioridad	Estado	Story Points (18)	
PROYEC-40	Memoria: Realizar apartado de Requisitos	Historia	Medium	FINALIZADA	8	
PROYEC-41	Corregir bugs en el Código	Historia	Medium	FINALIZADA	4	
PROYEC-42	Investigar como publicar nuestra aplicación en un servidor público	Historia	Medium	FINALIZADA	4	
PROYEC-43	Memoria: Realizar mejoras en general	Historia	Medium	FINALIZADA	2	

Figura A.16: Tareas Sprint 8

Sprint 9 - 08/06/2023 - 11/06/2023

El propósito de este sprint ha sido llevar a cabo una evaluación exhaustiva tanto de la documentación como del código creado a lo largo de los últimos seis meses durante el proceso de desarrollo de este proyecto. Este sprint ha tenido una duración de cuatro días respecto al resto de sprints, por motivos de exámenes no se ha podido comenzar antes este sprint.

A.3. Estudio de viabilidad

En esta sección, llevaremos a cabo un estudio exhaustivo de la viabilidad económica y legal del presente proyecto. Realizaremos un análisis detallado de los costos asociados con el desarrollo de este proyecto.

Viabilidad económica

Costes de personal

El desarrollo de este proyecto ha sido realizado por un desarrollador que ha trabajado tres horas y media diarias durante un periodo de seis meses. Se ha trabajado un promedio de 23 días al mes, lo que nos da un total de 483 horas de trabajo dedicadas al desarrollo de este proyecto.

Considerando que el salario bruto promedio en España para un desarrollador Junior es de 25698€, el costo total del personal del presente proyecto sería el siguiente:

- Salario neto anual → 20.354€
- Cuota Anual Seguridad Social → 1.631,8€
- Salario bruto anual → 25.698€
- **Salario bruto 6 meses → 12.849€**

Costes de Hardware

Dentro de los costes de Hardware, se encuentra el coste del dispositivo utilizado para el desarrollo de este proyecto. Este dispositivo tiene un precio de mercado de aproximadamente **1400€**.

El dispositivo utilizado para el desarrollo del proyecto, se estima que la amortización del equipo es de aproximadamente de tres años. Dado que hemos utilizado este dispositivo durante un período de seis meses, hemos logrado amortizar alrededor de un 7% de la inversión total realizada en el mismo.

Costes de Software

Estos costes se identifican con los costes de las herramientas utilizadas para el desarrollo de nuestro proyecto. En nuestro caso, todas las herramientas software que hemos utilizado han sido gratuitas, por lo tanto, este coste es cero para nuestro proyecto.

Otros costes

Estos costes se identifican con los costes adicionales que no pertenecen a ninguno de los costes anteriores.

- Dominio web → 4€
- Conexión a Internet → 150€
- Impresión de documentación → 47€
- Electricidad → 190€
- **Coste Total → 391€**

Costes de Totales

A continuación, podemos observar los costes del desarrollo del presente proyecto:

- ***Costes de Personal*** → 12.849,00€
- ***Costes de Hardware*** → 98,00€
- ***Costes de Software*** → 0,00€

- **Otros Costes** → 391,00€

Costes Totales → 14.640€

Para recuperar los costes generados del desarrollo del presente proyecto se ha pensado en incorporar publicidad y buscar patrocinios de empresas interesadas en promocionarse a través de nuestra aplicación web. Esto nos proporcionará ingresos continuos a lo largo del tiempo a través de la visualización de anuncios y acuerdos con patrocinadores. A continuación se realizará una estimación de como recuperaremos los costes del presente proyecto:

- **Ingresos por anuncios = Número de visitantes x CTR x CPC**
 1. **Número de visitantes:** Cantidad de personas que visitan nuestra página web.
 2. **Tasa de clics (CTR):** Porcentaje de visitantes que hacen clic en los anuncios.
 3. **Costo por clic (CPC):** Ingreso generado por cada clic realizado en un anuncio, en términos monetarios.

Supongamos que nuestro sitio web tiene un promedio de 10.000 visitantes mensuales y la tasa de clics (CTR) es del 10 %. El costo por clic (CPC) promedio es de 0,70€. Además, la plataforma de anuncios con la que trabajamos retiene el 15 % de los ingresos generados.

- **Número de visitantes:** 10.000
- **CTR:** 7 %
- **CPC:** 0,70€
- **Porcentaje retenido por la plataforma:** 15 %

A continuación calculamos el número de clics generados:

- **Clics = Número de visitantes x CTR**
 - $\text{Clics} = 10.000 \times 0,10$
 - **Clics = 1.000**

Luego, calculamos los ingresos brutos generados:

- **Ingresos brutos = Clics x CPC**
- Ingresos brutos = $1.000 \times 0.70\text{€}$
- **Ingresos brutos = 700€**

Finalmente, calculamos los ingresos netos teniendo en cuenta el porcentaje retenido por la plataforma de anuncios:

- **Ingresos netos = Ingresos brutos x (1-Porcentaje retenido)**
- Ingresos netos = $700\text{€} \times (1-0,15)$
- **Ingresos netos = 700€ x 0,85**
- **Ingresos netos = 595€**

Una vez realizados los cálculos, los ingresos netos generados a través de la visualización de anuncios en nuestra aplicación web serían 595€ al mes.

Teniendo en cuenta, que los costes generados para el desarrollo del presente proyecto han sido 14.640€, podemos decir que en un periodo 2 años y 1 mes se han recuperado todos los gastos generados para este proyecto. A partir de este momento, se empezaría a obtener beneficios del presente proyecto.

Viabilidad legal

En esta sección, nos dedicaremos a analizar las licencias asociadas a las herramientas que hemos utilizado en el desarrollo del presente proyecto.

En la tabla A.1 se presenta un listado de las herramientas y librerías usadas en el desarrollo del proyecto asociadas a su licencia.

Herramienta/Librería	Licencia	Versión
Flask	BSD	2.3.0
Jinja2	BSD	3.1.2
Geopy	MIT	2.3.0
GitHub	GNU	3.0
SQLite	GNU	3.12.12
StarUml	SHAREWARE	3.2.0
Docker	Licencia Apache 2.0	-

Tabla A.1: Licencias de las librerías y herramientas

Apéndice B

Especificación de Requisitos

B.1. Introducción

En esta sección se listarán y explicarán los objetivos y requisitos previamente establecidos al inicio del proyecto, que son necesarios para la aplicación. Además, también se han incluido algunos objetivos adicionales a medida que se ha avanzado en el desarrollo.

B.2. Objetivos generales

El propósito del presente proyecto se enfoca en los siguientes aspectos:

- Crear una plataforma en línea que proporcione datos relevantes (como noticias, pronósticos climáticos, eventos, etc.) sobre una ciudad o área geográfica de interés para el usuario.
- Proporcionar una experiencia de usuario (UX) fácil de entender, actualizada y accesible, que permita una compresión clara y sencilla de la información presentada en la plataforma.
- La plataforma en línea debe contar con la capacidad de obtener información en tiempo real a través de las APIs correspondientes.
- Organizar de forma sistemática todos los datos requeridos obtenidos mediante las APIs, de manera que se pueda acceder a ellos de manera fácil y eficiente.

- Proporcionar una plataforma en línea que garantice la seguridad de los usuarios y proteja su información personal de posibles amenazas o intrusiones externas.

B.3. Catálogo de requisitos

En este apartado, se detallarán de manera exhaustiva y precisa tanto los requisitos funcionales como los requerimientos no funcionales del proyecto actual.

Requisitos Funcionales

- **RF-1 Registro de Usuarios:** La aplicación debe contar con la opción para que un usuario pueda crear su propia cuenta y que la información que proporcione sea almacenada en la base de datos correspondiente.
- **RF-2 Inicio Sesión:** La aplicación debe tener la capacidad de verificar las credenciales del usuario y permitirle el acceso a todos los servicios disponibles en dicha aplicación.
- **RF-3 Mostrar Noticias:** La app deberá mostrar todas las noticias que se encuentren disponibles en la API *NewsData.io* y que estén redactadas en idioma español.
 - **RF-3.1 Consultar una Noticia:** Como usuario quiero ver la información de una noticia, con el fin de estar informado sobre la misma.
- **RF-4 Mostrar Eventos:** La app deberá mostrar todos los eventos disponibles en España proporcionados por la API de *TicketMaster* y que estén redactadas en idioma español.
 - **RF-4.1 Consultar un Evento:** Como usuario quiero consultar la información de un evento, con el fin de estar informado sobre dicho evento.
 - **RF-4.1.1 Añadir Eventos a Favoritos:** Como usuario deseo tener la opción de añadir eventos a favoritos con el fin de poder acceder a ellos sin tener que realizar búsquedas de nuevo.
- **RF-5 Buscar Noticias:** Como usuario deseo buscar información específica respecto a las noticias.

- **RF-5.1 Buscar Noticias por Ubicación:** Como usuario deseo tener la opción de buscar noticias por ubicación, con el fin de estar informado de las noticias sucedidas en una ubicación determinada.
 - **RF-5.2 Buscar Noticias por Categoría:** Como usuario deseo tener la opción de buscar noticias por categoría, con el fin de conocer todas las noticias en esa categoría específica.
- **RF-6 Buscar Eventos:** Como usuario deseo buscar información precisa y específica respecto a los eventos.
 - **RF-6.1 Buscar Eventos por Ubicación:** Como usuario deseo tener la opción de buscar eventos por ubicación, con el objetivo de estar informado de los eventos disponibles en una zona determinada.
 - **RF-6.2 Buscar Eventos por Categoría:** Como usuario deseo tener la opción de buscar eventos por categoría, con el fin de conocer los eventos disponibles en esa categoría específica.
 - **RF-7 Mostrar datos meteorológicos:** La aplicación debe mostrar la meteorología de la ubicación actual del usuario.
 - **RF-8 Buscar Meteorología por Ubicación:** Como usuario deseo tener la opción de buscar meteorología por ubicación, con el objetivo de estar informado del pronóstico del tiempo de una ubicación determinada.
 - **RF-9 Mostrar Eventos Favoritos:** La aplicación debe mostrar los eventos que han sido añadidos a favoritos por el usuario.
 - **RF-9.1 Eliminar Eventos de Favoritos:** Como usuario deseo tener la opción eliminar eventos de favoritos, con el fin de evitar interfaces de usuario sobrecargadas.

Requisitos No Funcionales

- **RFN-1 Usabilidad:** La aplicación debe ser fácil de usar y de comprender.
- **RFN-2 Integridad de los datos:** La aplicación no debe tener alteraciones.

- **RFN-3 Disponibilidad:** La aplicación debe ser accesible durante la mayor parte del tiempo y en diversos lugares, con el objetivo de brindar a los usuarios un acceso óptimo en todo momento.
- **RFN-4 Mantenimiento:** La aplicación debe ser fácil de actualizar y mantener.
- **RFN-5 Seguridad:** La aplicación debe estar protegida contra el acceso no autorizado.

B.4. Especificación de requisitos

En este apartado, presentaremos el diagrama de Casos de Uso y detallaremos ciertos requerimientos mencionados previamente en el apartado anterior. Cabe destacar que la aplicación solo cuenta con un tipo de usuario, es decir, el usuario cliente, ya que no se contempla la existencia de algún usuario administrador o equivalente.

Diagrama de Casos de Uso

A continuación, se presenta el diagrama que muestra los casos de uso del proyecto actual:

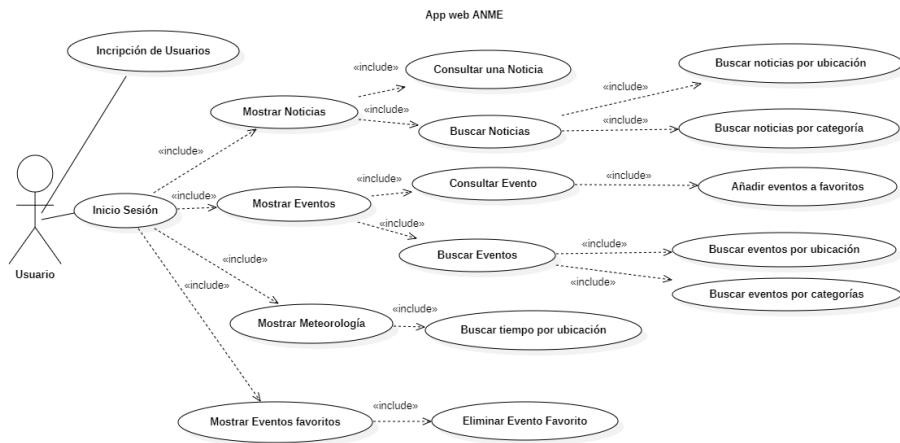


Figura B.1: Diagrama de Casos de Uso

Casos de Uso

CU-1	Inscripción de Usuarios
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-1
Descripción	Permite que el usuario registre una cuenta en el sistema.
Precondición	El sistema debe estar operativo.
Acciones	<ol style="list-style-type: none"> 1. El usuario selecciona la alternativa <i>Crear una cuenta</i> 2. El usuario ingresa su nombre, nombre de usuario y contraseña. 3. El usuario hace clic en el botón de registro.
Postcondición	El username elegido por el usuario no debe estar registrado en el sistema.
Excepciones	El username ya está registrado en el sistema (mensaje).
Importancia	Alta

Tabla B.1: CU-1 Inscripción de Usuarios.

CU-2	Inicio sesión
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-2
Descripción	Permite el acceso al usuario para utilizar la app.
Precondición	El usuario debe estar registrado en la app.
Acciones	<ol style="list-style-type: none"> 1. El usuario accede a la opción de Login. 2. El usuario introduce sus credenciales en el formulario. 3. El usuario pulsa el botón de Acceder.
Postcondición	Las credenciales introducidas por el usuario deben estar registradas en el sistema.
Excepciones	El servicio de noticias no está disponible(Mensaje).
Importancia	Alta

Tabla B.2: CU-2 Inicio sesión.

CU-3	Mostrar Noticias
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-3, RF-3.1
Descripción	Muestra todas las noticias disponibles al usuario.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de noticias debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Noticias</i> situado en la barra de navegación.
Postcondición	Ninguna.
Excepciones	El servicio de noticias no está disponible(Mensaje).
Importancia	Alta

Tabla B.3: CU-3 Mostrar noticias.

CU-4	Consultar una noticia
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-3.1
Descripción	Permite leer al usuario una noticia en concreto.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de noticias debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Noticias</i> situado en la barra de navegación. 2. El usuario hace clic en el botón <i>Ver Noticia</i> en la noticia deseada.
Postcondición	Ninguna
Excepciones	Esta noticia no se encuentra disponible(Mensaje).
Importancia	Alta

Tabla B.4: CU-3 Consultar una noticia.

CU-5	Mostrar Eventos
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	R.F-4, R.F-4.1, R.F-4.1.1
Descripción	Muestra todos los eventos disponibles al usuario.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de eventos debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Eventos</i> situado en la barra de navegación.
Postcondición	Ninguna.
Excepciones	El servicio de eventos no está disponible(Mensaje).
Importancia	Alta

Tabla B.5: CU-5 Mostrar eventos.

CU-6	Consultar un Evento
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-4.1, RF-4.1.1
Descripción	Permite al usuario conocer las características de un evento en específico.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de eventos debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic el botón de <i>Eventos</i> situado en la barra de navegación. 2. El usuario hace clic en el botón <i>Ver Evento</i> en el evento deseado.
Postcondición	Ninguna
Excepciones	Este evento no se encuentra disponible(Mensaje).
Importancia	Alta

Tabla B.6: CU-6 Consultar evento.

CU-7	Añadir Eventos a Favoritos
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-4.1.1
Descripción	Permite al usuario añadir eventos a favoritos.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de eventos debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic el botón de <i>Eventos</i> situado en la barra de navegación. 2. El usuario hace clic en el botón <i>Ver Evento</i> en el evento deseado. 3. Pulsa el botón <i>Añadir a Favoritos</i> en el evento deseado.
Postcondición	El evento que se va a añadir a favoritos no puede existir en los eventos favoritos del usuario.
Excepciones	Este evento ya existe en favoritos(Mensaje).
Importancia	Media

Tabla B.7: CU-7 Añadir eventos a favoritos.

CU-8	Buscar noticias por Ubicación
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-5.1
Descripción	Permite al usuario conocer todas las noticias de una ubicación determinada.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de noticias debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Noticias</i> situado en la barra de navegación. 2. El usuario introduce la ubicación deseada en la barra de búsqueda. 3. Hacer clic en el botón de <i>Buscar</i>.
Postcondición	Ninguna
Excepciones	No existen noticias disponibles para esta ubicación(Mensaje).
Importancia	Alta

Tabla B.8: CU-8 Buscar noticias por Ubicación.

CU-9	Buscar noticias por Categorías
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-5.2
Descripción	Permite al usuario conocer todas las noticias de una categoría específica.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de noticias debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Noticias</i> situado en la barra de navegación. 2. El usuario selecciona la categoría por la que desea buscar. 3. Pulsa el botón de <i>Buscar</i>.
Postcondición	Ninguna
Excepciones	No existen noticias disponibles para esta categoría(Mensaje).
Importancia	Alta

Tabla B.9: CU-9 Buscar noticias por Categorías.

CU-10	Buscar Eventos por Ubicación
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-6.1
Descripción	Permite al usuario conocer todos los eventos de una ubicación determinada.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de eventos debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic el botón de <i>Eventos</i> situado en la barra de navegación. 2. El usuario introduce la ubicación deseada en la barra de búsqueda. 3. Hacer clic en el botón de <i>Buscar</i>.
Postcondición	Ninguna
Excepciones	No existen eventos disponibles para esta ubicación(Mensaje).
Importancia	Alta

Tabla B.10: CU-10 Buscar eventos por Ubicación.

CU-11	Buscar Eventos por Categorías
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-6.2
Descripción	Permite al usuario conocer todos los eventos de una categoría específica.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de eventos debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Eventos</i> situado en la barra de navegación. 2. El usuario selecciona la categoría por la que desea buscar. 3. Pulsa el botón de <i>Buscar</i>.
Postcondición	Ninguna
Excepciones	No existen eventos disponibles para esta categoría(Mensaje).
Importancia	Alta

Tabla B.11: CU-11 Buscar eventos por Categorías.

CU-12	Mostrar Meteorología
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-7
Descripción	Permite conocer el pronóstico del tiempo de una ubicación de interés.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de meteorología debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Meteorología</i> situado en la barra de navegación.
Postcondición	Ninguna.
Excepciones	El pronóstico del tiempo no se encuentra disponible(Mensaje).
Importancia	Alta

Tabla B.12: CU-12 Mostrar Meteorología.

CU-13	Buscar Meteorología por Ubicación
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-8
Descripción	Permite al usuario conocer el pronóstico del tiempo para una ubicación determinada.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. El servicio de meteorología debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Meteorología</i> situado en la barra de navegación. 2. El usuario introduce la ubicación deseada en la barra de búsqueda. 3. Hacer clic en el botón de <i>Buscar</i>.
Postcondición	Ninguna
Excepciones	No se encuentra disponible el pronóstico del tiempo para esta ubicación(Mensaje).
Importancia	Alta

Tabla B.13: CU-13 Buscar Meteorología por Ubicación.

CU-14	Mostrar Eventos Favoritos
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-9
Descripción	Permite al usuario visualizar sus eventos favoritos.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. La base de datos debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Eventos Favoritos</i> situado en la barra de navegación.
Postcondición	Ninguna.
Excepciones	Ninguna.
Importancia	Alta

Tabla B.14: CU-14 Mostrar Eventos Favoritos.

CU-15	Eliminar Eventos de Favoritos
Autor	José Manuel Rodríguez Iglesias
Versión	1.0
Requisitos asociados	RF-9.1
Descripción	Permite al usuario eliminar eventos de favoritos.
Precondición	<ol style="list-style-type: none"> 1. El usuario debe haber iniciado sesión en el sistema. 2. La base de datos debe estar disponible.
Acciones	<ol style="list-style-type: none"> 1. El usuario hace clic en el botón de <i>Eventos Favoritos</i> situado en la barra de navegación. 2. El usuario hace clic en el botón <i>Eliminar Evento de Favoritos</i> en el evento deseado.
Postcondición	Ninguna.
Excepciones	Ninguna.
Importancia	Media

Tabla B.15: CU-15 Eliminar eventos de favoritos.

Apéndice C

Especificación de diseño

C.1. Introducción

En esta sección, se proporciona una explicación sobre cómo se han diseñado y organizado las diversas partes de nuestra aplicación web que hemos desarrollado.

C.2. Diseño de datos

Para el diseño de datos, hemos utilizado una base de datos para estructurar la información almacenada en nuestra aplicación. En nuestro caso, hemos utilizado SQLite, la cual está compuesta por varias tablas:

- **Usuarios:** Esta tabla contiene todos los usuarios que se registran en nuestra aplicación. Para registrarse en nuestra aplicación, el usuario debe introducir un correo electrónico (por ejemplo, una cuenta de Gmail, etc.) y una contraseña (la cual estará cifrada en la base de datos).
- **EventosFavoritos:** En esta tabla se almacenan todos los eventos elegidos como favoritos para el usuario. En dicha tabla se añadirá toda la información relevante para cada uno de los eventos.
- **Ubicaciones:** Esta tabla contiene la información descriptiva de cada ubicación que es relevante para el usuario.

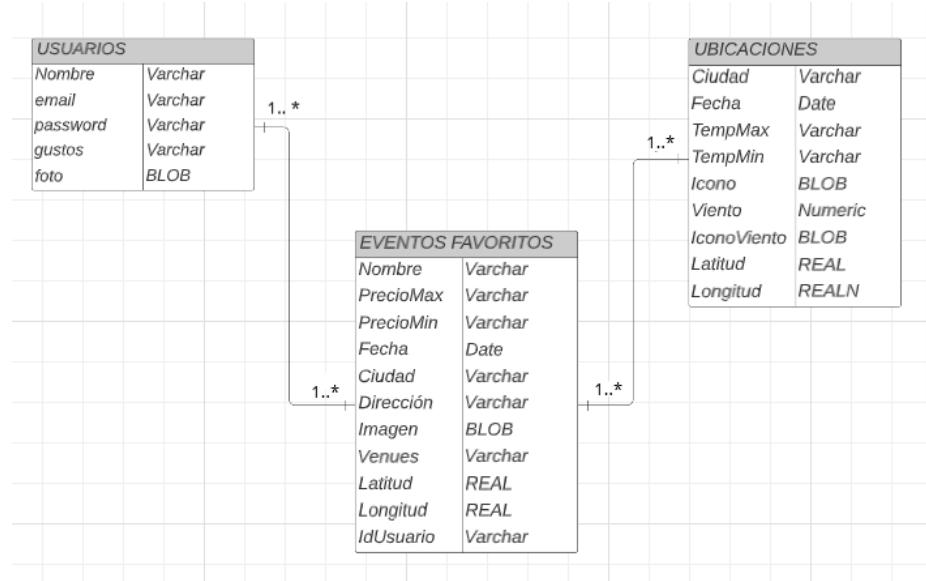


Figura C.1: Esquema relacional de la base de datos

Por otra parte, hemos utilizado dos ficheros con formato JSON, noticias.json y eventos.json, para almacenar la información que nos devuelve la API de noticias y la API de eventos. Se ha tomado esta decisión para almacenar los datos de las dos APIs anteriores porque es más eficiente que almacenar toda la información en la base de datos.

El fichero **Noticias.json** contiene la siguiente información:

- **Title:** Título de la noticia.
- **Link :** Enlace de publicación de la noticia.
- **Country :** Nombre del país donde ha sucedido la noticia.
- **Language:** Idioma en el que está redactada la noticia.
- **Keywords:** Palabras claves de la noticia.
- **Autor:** Nombre de la entidad que ha publicado la noticia.
- **Description:** Descripción del suceso de la noticia.
- **Content:** Resumen de la noticia publicada.
- **PubDate:** Fecha de publicación de la noticia.
- **Category:** Categoría de la noticia.
- **VideoUrl:** Enlace al vídeo de la noticia(Puede no haber enlace)
- **ImageUrl:** Enlaces de las imágenes de la noticia.
- **SourceId:** Identificador de la noticia.

El fichero **Eventos.json** contiene la siguiente información:

- **Name:** Nombre del evento.
- **Type :** Tipo de evento.
- **Id:** Identificador del evento.
- **ImageUrl:** Enlaces de las imágenes del evento.
- **StartTime:** Fecha y hora de comienzo del evento.
- **EndTime:** Fecha y hora de finalización del evento.
- **Timezone:** Zona horaria del lugar donde se celebra el evento.
- **Genero:** Género del evento (música,deportes,etc.)
- **PriceRanges:** Precio mínimo y máximo del evento.
- **Country:** Nombre del país donde se celebra el evento.
- **Address:** Ubicación del evento.
- **Location:** Coordenadas geográficas del lugar donde se celebra el evento.

C.3. Diseño procedimental

El objetivo de esta sección es proporcionar una visión general del proceso completo que deben seguir los usuarios para poder iniciar sesión, registrarse y visualizar las noticias, eventos y meteorología en nuestra aplicación web.

A continuación se presentan los diagramas de secuencia UML [3] obtenidos:

En la figura C.2 se muestra el diagrama de secuencia que representa las interacciones y flujo de acciones necesarias para registrarse y loguearse en la aplicación:

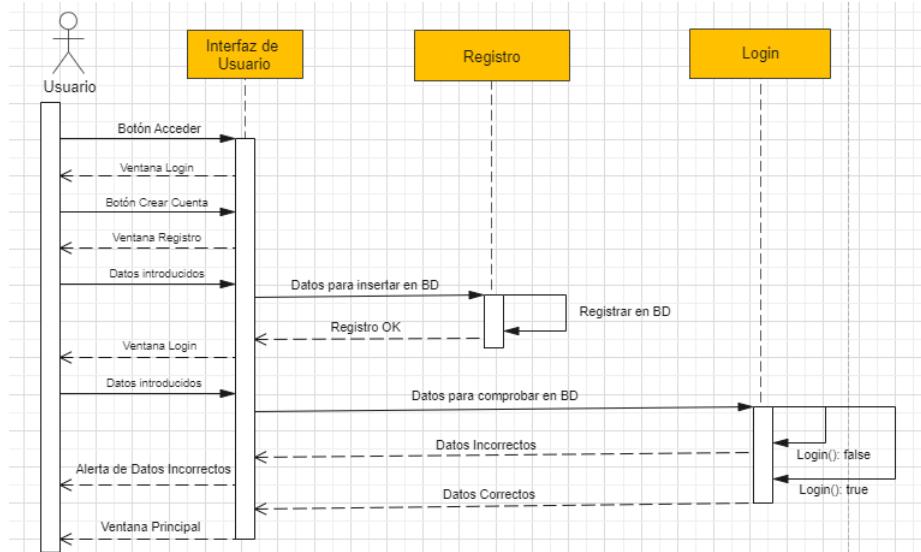


Figura C.2: Diagrama de Secuencia de Login y Registro de usuarios

En la figura C.3 se muestra el diagrama de secuencia que representa las interacciones y flujo de acciones del servicio Noticias de nuestra aplicación:

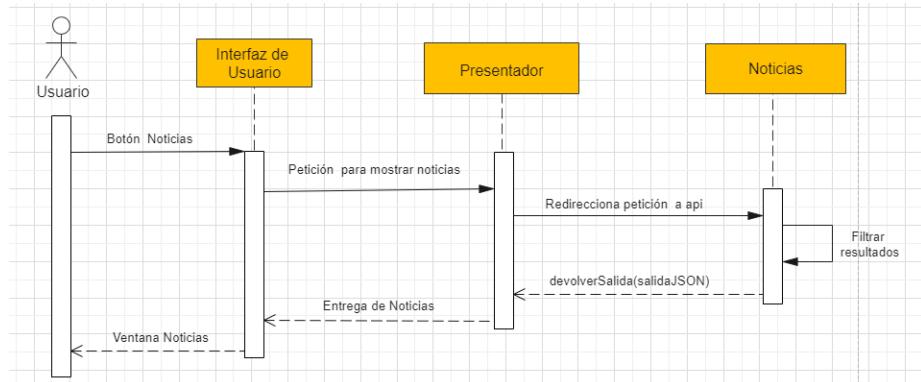


Figura C.3: Diagrama de secuencia del servicio Noticias

En la figura C.4 se muestra el diagrama de secuencia que representa las interacciones y flujo de acciones del servicio Eventos de nuestra aplicación:

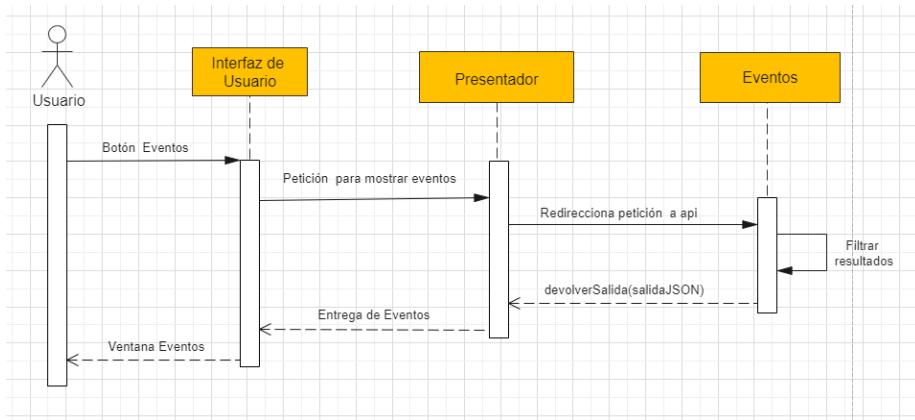


Figura C.4: Diagrama de secuencia del servicio Eventos

En la figura C.5 se muestra el diagrama de secuencia que representa las interacciones y flujo de acciones del servicio Meteorología de nuestra aplicación:

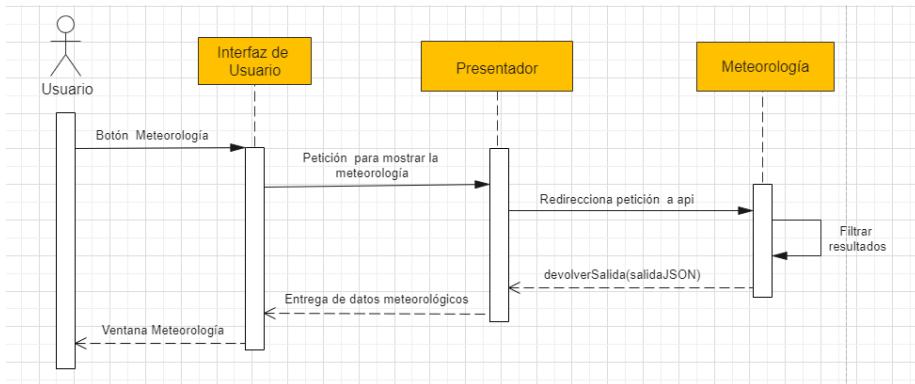


Figura C.5: Diagrama de secuencia del servicio Meteorología

C.4. Diseño arquitectónico

La presente sección está dedicada a detallar la arquitectura del proyecto. Se utiliza una estructura cliente-servidor que se extiende desde la aplicación hasta las APIs. En ambas partes, tanto en el servidor como en el cliente, se llevan a cabo validaciones para verificar la precisión y consistencia de los datos intercambiados.

El reparto de tareas en este sistema se divide en dos partes: el cliente se ocupa de la interfaz y de recibir la solicitud de información que desea el usuario, mientras que el procesamiento de los datos, la inferencia en la red y la devolución de los resultados son tareas a cargo del servidor.

Modelo-Vista-Presentador (MVP)

- **Modelo:** El modelo es el responsable de aplicar la lógica de negocio y de gestionar el acceso a los datos de nuestra aplicación. En nuestro proyecto le hemos dado el nombre *modelo* y está escrito en Python. Dicho modelo está formado por los módulos de noticias, eventos y meteorología, donde se extraen todas las noticias y eventos disponibles, y el pronóstico del tiempo de una ubicación.
- **Vista:** La vista es la responsable de generar la interfaz de usuario de nuestra aplicación para que el usuario interactúe con ella. Está escrita en HTML y CSS, y está compuesta por varios elementos que permiten comunicarse con el presentador para poder navegar entre diferentes páginas de la aplicación.
- **Presentador:** El presentador es el intermediario entre la *Vista* y el *Modelo* para que ambos puedan comunicarse. Al *Presentador* se le ha llamado en nuestro proyecto *app* y está escrito en Python. Este recopila las solicitudes realizadas por el usuario y solicita la ejecución de diversas operaciones a los distintos módulos del modelo.

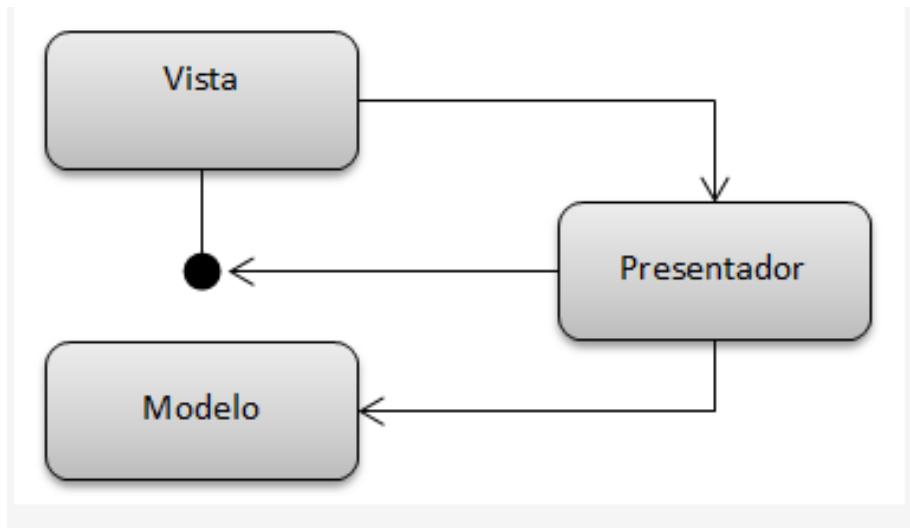


Figura C.6: Patrón Modelo-Vista-Presentador. [2]

C.5. Diseño de interfaces

En este apartado se presentará el prototipo inicial de la aplicación web, creado durante las primeras semanas del proyecto con la ayuda de la herramienta Pencil.

El propósito de este prototipo era tener una visión general de cómo enfocar nuestra aplicación en términos de usabilidad, aunque se han realizado algunas modificaciones respecto de la interfaz inicial.

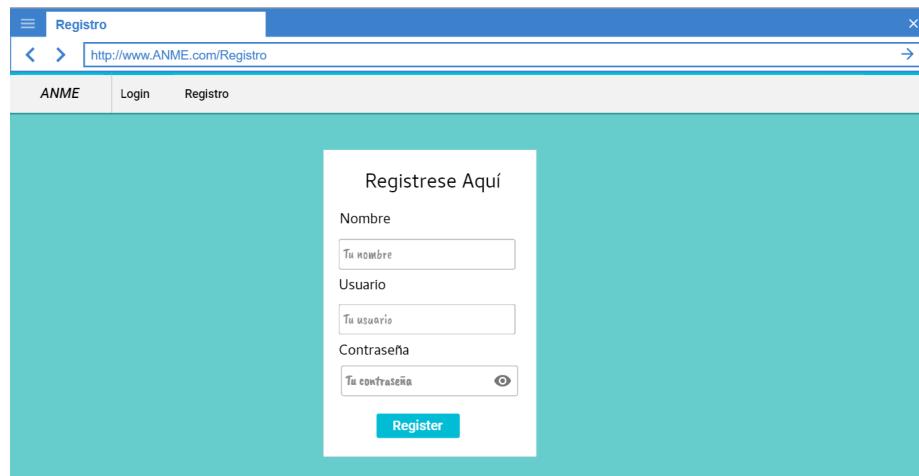


Figura C.7: Prototipo de Registro

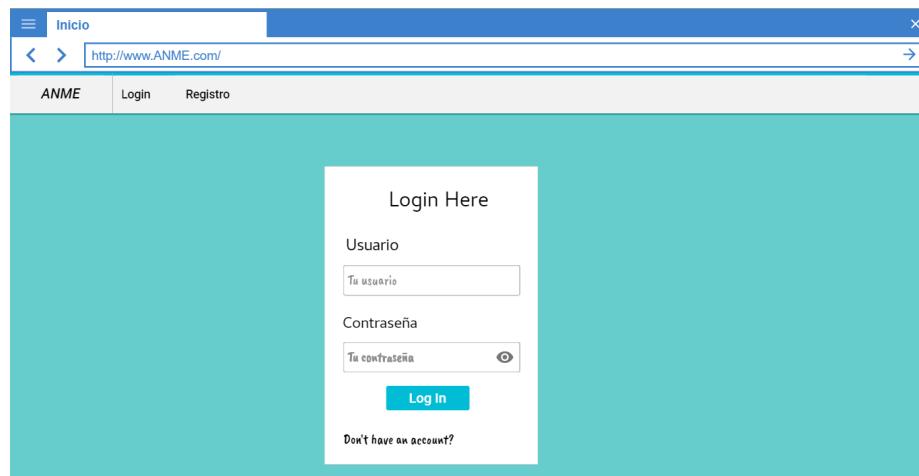


Figura C.8: Prototipo de Login



Figura C.9: Prototipo de Principal

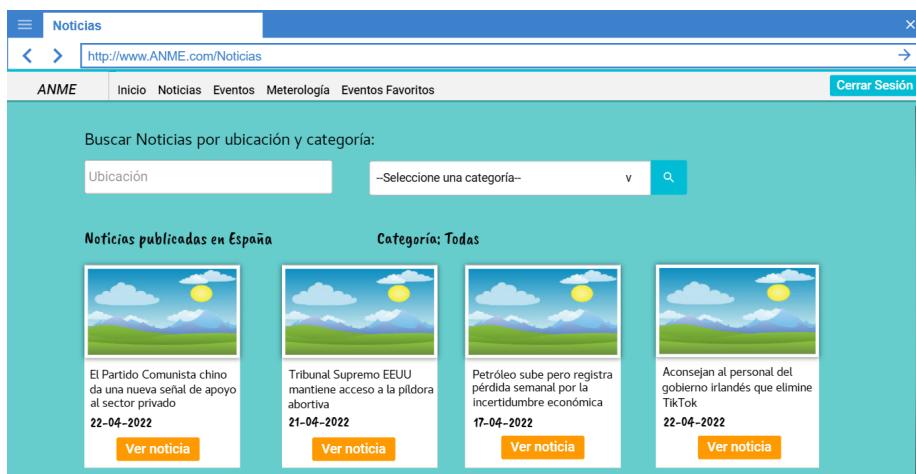


Figura C.10: Prototipo de Noticias

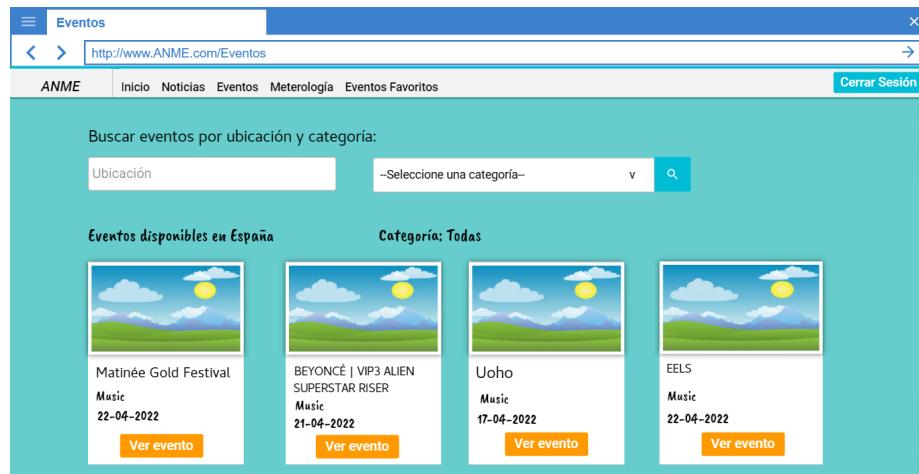


Figura C.11: Prototipo de Eventos

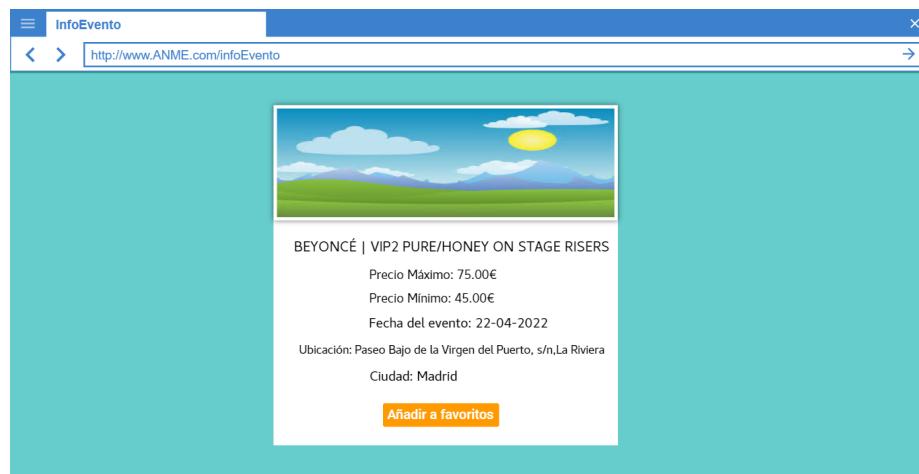


Figura C.12: Prototipo de Información del Evento

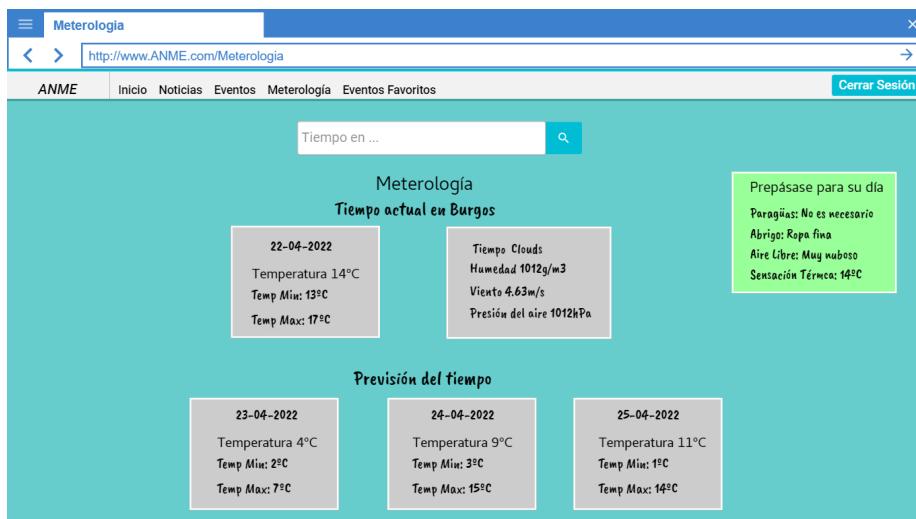


Figura C.13: Prototipo de Meteorología

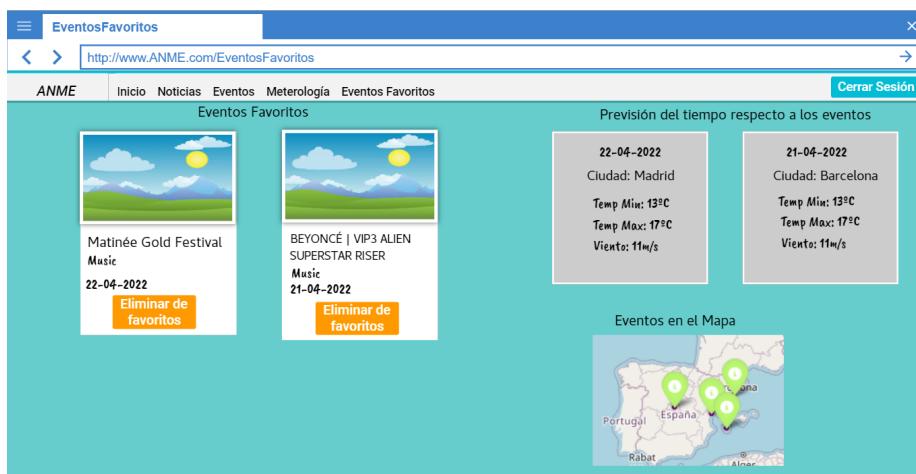


Figura C.14: Prototipo de Eventos Favoritos

Apéndice D

Documentación técnica de programación

D.1. Introducción

En este apéndice, se proporcionará información detallada sobre la estructura del presente proyecto, los requisitos fundamentales para garantizar una correcta ejecución del proyecto y un manual de instrucciones dirigido al programador.

D.2. Estructura de directorios

En este sub apartado detallaremos la estructura de directorios del presente proyecto:

- *docs*: En este directorio se encuentra la memoria y los anexos creados en formato LaTeX del presente proyecto.
- *src*: En este directorio es donde se encuentra tanto el código de la lógica de la aplicación como el código que se utiliza para desarrollar la interfaz de usuario. Además, dentro de este directorio podemos encontrar los siguientes elementos:
 - *static*: Este directorio contiene los archivos de estilos CSS e imágenes utilizadas en nuestra aplicación.
 - *templates*: En este directorio es donde se encuentra la interfaz del usuario, que se utiliza para generar la vista de la aplicación.

- DB.db: Este archivo es la base de datos de nuestra aplicación, que es donde se guardan todos los datos que son necesarios para el correcto funcionamiento de nuestra aplicación.
- app.py: Este archivo contiene el código que conecta la interfaz de usuario con la lógica de la aplicación.
- db.py: En este archivo se encuentra la conexión a la base de datos.
- eventos.json: Aquí se almacenan los eventos disponibles que hay en nuestra aplicación para seguir teniendo el servicio disponible de eventos en el caso de que la API que nos proporciona dichos eventos no esté disponible en el momento que la solicitemos.
- noticias.json: Aquí se almacenan las noticias disponibles que hay en nuestra aplicación para seguir teniendo el servicio disponible de noticias en el caso de que la API que nos proporciona dichas noticias no esté disponible en el momento que la solicitemos.
- servicios: En este directorio se encuentran todos los servicios que se han utilizado en nuestra aplicación. En nuestro caso, se han utilizado tres servicios que son:
 - ServicioNoticias: En este módulo se encuentra el código donde se realizan las peticiones a la API NewsDataIO para extraer de la API las noticias disponibles.
 - ServicioEventos: En este módulo se encuentra el código donde se realizan las peticiones a la API TicketMaster para extraer de la API los eventos disponibles en cada momento.
 - ServicioMeteorología: En este módulo se encuentra el código donde se realizan las peticiones a las APIs OpenWeather y TuTiempo para obtener la meteorología de una ubicación.
 - FuncionesCompartidas: En este módulo se encuentran las funciones comunes que utilizan ambos servicios de nuestra aplicación.
- start.py: Este fichero es el encargado de ejecutar nuestra aplicación web.

D.3. Manual del programador

Instalamos Python

Para poder llevar a cabo la ejecución de este proyecto, es fundamental realizar la instalación de Python en nuestro equipo. Para ello, podemos

obtener la versión necesaria en el siguiente enlace: <https://www.python.org/downloads/>. Una vez que se haya completado la descarga, procederemos a la ejecución e instalación del programa.

D.4. Compilación, instalación y ejecución del proyecto

Para hacer uso del presente proyecto, será necesario descargarlo del repositorio de GitHub en el cual se encuentra alojado. Podemos acceder a este repositorio a través del siguiente enlace: https://github.com/jmri1001/PROYECTO_TFG.git. Una vez descargado el proyecto, procederemos a descomprimirlo y a continuación lo importaremos en nuestra herramienta de trabajo.

El presente proyecto ha sido creado y desarrollado utilizando Visual Studio Code como la herramienta principal de trabajo.

Compilación

Como el código está realizado en Python, no hace falta realizar el paso de compilación, debido a que Python es un lenguaje de programación interpretado.

Instalación de dependencias del proyecto

Una vez que se haya importado el proyecto en nuestra herramienta de desarrollo, instalaremos las dependencias necesarias para nuestro proyecto, ejecutando en la terminal el siguiente comando: -> `pip install -r requirements.txt`

Tras haber instalado las dependencias en nuestro proyecto, lo siguiente que haremos será ejecutar el proyecto.

Ejecución del proyecto

Para poder ejecutar el proyecto en nuestra herramienta de desarrollo, es necesario abrir una terminal en dicha herramienta y, a continuación, accederemos al directorio donde se encuentra ubicado el proyecto utilizando el comando `cd path`.

Una vez que hemos accedido al directorio del proyecto, ejecutaremos el siguiente comando:

-> `python start.py`

Tras haber ejecutado el comando anterior, nuestra aplicación se iniciará en `http://127.0.0.1:8080`

Consejo: Si estamos usando la aplicación, es importante no cerrar la terminal, ya que esto detendrá el proceso de la aplicación.

D.5. Pruebas del sistema

En este apartado se han llevado a cabo una serie de pruebas exhaustivas con el objetivo de verificar que el funcionamiento de la aplicación es el adecuado. También, se ha implementado la integración continua para garantizar la creación de código de alta calidad y así prevenir posibles errores que podrían desencadenar otros más graves.

Especificación de Pruebas del sistema:

- **Prueba 1 :** En esta prueba se verifica los dos casos posibles que se pueden dar al registrarse un nuevo usuario en la aplicación.
 - *Caso 1:* Verificar que se pueda registrar un nuevo usuario correctamente. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de Registro de la aplicación.
 2. Completar todos los campos obligatorios del formulario de registro con datos válidos.
 3. Verificar que se redirija al usuario a la página de Login.
 4. Hacer clic en el botón Registrarse.
 5. Verificar que el nuevo usuario se pueda autenticar correctamente con las credenciales registradas.
 - *Caso 2:* Verificar que se muestre un mensaje de error cuando se intente registrar un usuario con un nombre que ya existe en el sistema. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de Registro de la aplicación.
 2. Ingresar un nombre de un usuario registrado en el formulario de Registro.
 3. Completar los demás campos del formulario con datos válidos.
 4. Hacer clic en el botón Registrarse.

5. Verificar que se muestre un mensaje de error indicando que el nombre de usuario ya existe en el sistema.
- **Prueba 2 :** En esta prueba se verifica los dos casos posibles que se pueden dar a la hora de Iniciar sesión en la aplicación.
 - *Caso 1:* Verificar que se puedan ingresar credenciales válidas y acceder al sistema correctamente. Para comprobar este caso se realizará lo siguiente:
 1. Ingresar un usuario válido y una contraseña válida en la página de Login de la aplicación.
 2. Hacer clic en el botón Iniciar sesión.
 3. Verificar que se redirija correctamente a la página principal del usuario.
 - *Caso 2:* Verificar que se muestre un mensaje de error cuando se ingresen credenciales inválidas. Para comprobar este caso se realizará lo siguiente:
 1. Ingresar un usuario inválido y una contraseña inválida en la página de Login.
 2. Hacer clic en el botón Iniciar sesión.
 3. Verificar que se muestre un mensaje de error indicando que las credenciales son incorrectas.
- **Prueba 3 :** En esta prueba se verifica que el funcionamiento de la ventana de Noticias de la aplicación web funciona correctamente, para ello comprobaremos varios casos:
 - *Caso 1:* Verificar que las noticias se muestren correctamente en la página de noticias de la app. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de noticias de la aplicación.
 2. Verificar que se carguen y muestren las noticias más recientes.
 3. Comprobar que se muestren los detalles de cada noticia, como el título y la fecha de publicación.
 - *Caso 2:* Verificar que se muestre un mensaje de error cuando no haya noticias disponibles. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de noticias de la aplicación.

2. Verificar que se muestre un mensaje indicando que no hay noticias disponibles en este momento.
- *Caso 3:* Verificar que se pueda hacer clic en una noticia y redirigir al usuario a la página de detalle de dicha noticia. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de noticias de la aplicación.
 2. Hacer clic en una noticia en la lista de noticias.
 3. Verificar que se redirija al usuario a la página de detalles de dicha noticia mostrando la información completa de la noticia.
- **Prueba 4 :** En esta prueba se verifica que el funcionamiento de la ventana de Eventos de la aplicación web funciona correctamente, para ello comprobaremos varios casos:
 - *Caso 1:* Verificar que los eventos se muestren correctamente en la página de eventos de la app. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de eventos de la aplicación.
 2. Verificar que se carguen y muestren los eventos disponibles.
 3. Comprobar que se muestren los detalles de cada evento, como el título y la categoría del mismo.
 - *Caso 2:* Verificar que se muestre un mensaje de error cuando no haya eventos disponibles. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de eventos de la aplicación.
 2. Verificar que se muestre un mensaje indicando que no hay eventos disponibles en este momento.
 - *Caso 3:* Verificar que se pueda hacer clic en un evento y redirigir al usuario a la página de detalle de dicho evento. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de eventos de la aplicación.
 2. Hacer clic en un evento.
 3. Verificar que se redirija al usuario a la página de detalles de dicho evento mostrando la información completa del evento.

- **Prueba 5 :** En esta prueba se verifica que se pueda añadir un evento a la lista de favoritos de un usuario. Para comprobar este caso se realizará lo siguiente:

1. Iniciar sesión con un usuario válido.
2. Acceder a la página de eventos.
3. Hacer clic en un evento en la lista de eventos.
4. Comprobar que se muestren los detalles del evento.
5. Hacer clic en el botón Añadir a Favoritos.
6. Verificar que se muestre un mensaje que indique que el evento ha agregado a favoritos exitosamente.
7. Verificar que el evento se ha agregado correctamente a la lista de favoritos del usuario.

- **Prueba 6 :** En esta prueba se verifica que se pueda eliminar un evento de la lista de favoritos de un usuario. Para comprobar este caso se realizará lo siguiente:

1. Iniciar sesión con un usuario válido.
2. Acceder a la página de Eventos Favoritos del usuario dentro la aplicación.
3. Hacer clic en el botón Eliminar de un evento en la lista de eventos favoritos.
4. Verificar que se muestre un mensaje indicando que el evento se ha eliminado de favoritos exitosamente.
5. Comprobar que el evento se ha eliminado correctamente de la lista de eventos favoritos del usuario.

- **Prueba 7 :** En esta prueba se verifica que se muestren todos los eventos marcados como favoritos por un usuario. Para comprobar este caso se realizará lo siguiente:

1. Iniciar sesión con un usuario válido.
2. Acceder a la página de Eventos Favoritos del usuario dentro la aplicación.
3. Verificar que se carguen y muestren solo los eventos marcados como favoritos.

4. Comprobar que se muestren los detalles de cada evento, como el título, la fecha, la ubicación, la meteorología, etc.
- **Prueba 8 :** En esta prueba se verifica que se puedan buscar noticias por ubicación dentro de la aplicación web. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de noticias de la aplicación.
 2. Ingresar una ubicación válida en el campo de búsqueda por ubicación.
 3. Hacer clic en el botón "Buscar"
 4. Verificar que se muestren las noticias relacionadas con la ubicación ingresada.
 - **Prueba 9 :** En esta prueba se verifica que se puedan buscar noticias por categoría dentro de la aplicación web. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de noticias de la aplicación.
 2. Seleccionar una categoría en el campo de búsqueda por categoría.
 3. Hacer clic en el botón "Buscar"
 4. Verificar que se muestren las noticias relacionadas con la categoría seleccionada.
 - **Prueba 10 :** En esta prueba se verifica que se puedan buscar eventos por ubicación dentro de la aplicación web. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de eventos de la aplicación.
 2. Ingresar una ubicación válida en el campo de búsqueda por ubicación.
 3. Hacer clic en el botón "Buscar"
 4. Verificar que se muestren los eventos relacionados con la ubicación ingresada.
 - **Prueba 11 :** En esta prueba se verifica que se puedan buscar eventos por categoría dentro de la aplicación web. Para comprobar este caso se realizará lo siguiente:

1. Acceder a la página de eventos de la aplicación.
 2. Seleccionar una categoría en el campo de búsqueda por categoría.
 3. Hacer clic en el botón "Buscar"
 4. Verificar que se muestren los eventos relacionados con la categoría seleccionada.
-
- **Prueba 12 :** En esta prueba se verifica que se muestre correctamente el pronóstico del tiempo actual. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de meteorología de la aplicación.
 2. Verificar que se muestre la información actual del clima, como la temperatura, el viento, la humedad, etc.

 - **Prueba 13 :** En esta prueba se verifica que se pueda buscar el pronóstico del tiempo para una ubicación en concreto. Para comprobar este caso se realizará lo siguiente:
 1. Acceder a la página de meteorología de la aplicación.
 2. Ingresar una ubicación válida en el campo de búsqueda de ubicación.
 3. Hacer clic en el botón Buscar.
 4. Verificar que se muestre la información actual del clima de la ubicación ingresada.

A continuación podemos observar en la Tabla D.1 las diferentes pruebas que se pueden hacer para ver que se cumplen todos los requisitos definidos en el apéndice de Especificación de Requisitos.

Requisitos	Pruebas de Verificación
RF1: Registro de usuarios	Prueba 1 Pag ⁵⁰
RF2: Inicio Sesión	Prueba 2 Pag ⁵¹
RF3: Mostrar Noticias	Prueba 3 Pag ⁵¹
RF4: Consultar una Noticia	Prueba 3 Pag ⁵¹
RF5: Mostrar Eventos	Prueba 4 Pag ⁵²
RF6: Consultar un evento	Prueba 4 Pag ⁵²
RF7: Añadir eventos a Favoritos	Prueba 5 Pag ⁵³
RF8: Buscar noticias por Ubicación	Prueba 8 Pag ⁵⁴
RF9: Buscar noticias por Categoría	Prueba 9 Pag ⁵⁴
RF10: Buscar Eventos por Ubicación	Prueba 10 Pag ⁵⁴
RF11: Buscar Eventos por Categoría	Prueba 11 Pag ⁵⁴
RF12: Mostrar Meteorología	Prueba 12 Pag ⁵⁵
RF13: Buscar Meteorología por ubicación	Prueba 13 Pag ⁵⁵
RF14: Mostrar Eventos Favoritos	Prueba 5 Pag ⁵³
RF15: Eliminar Eventos de Favoritos	Prueba 6 Pag ⁵³

Tabla D.1: Pruebas de Verificación de Requisitos

Integración Continua

En esta sección se detallará el proceso que se ha seguido para implementar la integración continua en el presente proyecto, es decir, como se ha configurado el esquema de CD/CI, como está organizado, qué acciones se han configurado en GitHub, cómo y donde se despliega ahora mismo la aplicación desarrollada.

Para realizar la integración continua en el presente proyecto se ha utilizado la herramienta Docker, ya que esta es muy efectiva para asegurar que nuestra aplicación se construya, pruebe y despliegue de manera consistente y segura.

Para desplegar el proyecto hemos utilizado uno de los IaaS (Infrastructure as a Service) más famosos, Google Cloud.

A continuación se detallarán los pasos que se han seguido para la implementación de la integración continua:

1. **Creación de la imagen de Docker:** Se ha necesitado crear un archivo de configuración llamado *Dockerfile* en el repositorio del código

del proyecto. Este archivo contiene las instrucciones para construir la imagen de Docker de nuestra aplicación, como por ejemplo las dependencias, el entorno de ejecución, etc.

2. **Configuración del flujo de CI:** Para configurar el flujo de Integración Continua, se ha utilizado una herramienta de Google Cloud llamada Cloud Build. A partir de los denominados Triggers o activadores, es capaz de detectar cuando hacemos commit en el repositorio (u otra opción a elegir como pull request) y generar una nueva compilación.

	Nombre	Descripción	Repository	Commit	Timestamp	Duración	...	
□	5e647c85	jmr1001/PROYECTO_TFG	main	f4f5721	rmpgpb-projecto-tfg-europe-west1-jmr1001-PROYECTO-TFG-mavoz	6/5/23, 21:26	2 min	⋮
□	3906d1ff	jmr1001/PROYECTO_TFG	main	3e8b4ec	rmpgpb-projecto-tfg-europe-west1-jmr1001-PROYECTO-TFG-mavoz	6/5/23, 20:15	2 min 5 s	⋮
□	db74ae12	jmr1001/PROYECTO_TFG	main	e3e94aa	rmpgpb-projecto-tfg-europe-west1-jmr1001-PROYECTO-TFG-mavoz	5/5/23, 00:39	2 min 13 s	⋮
□	5b04b78	jmr1001/PROYECTO_TFG	main	2b568e	rmpgpb-projecto-tfg-europe-west1-jmr1001-PROYECTO-TFG-mavoz	5/5/23, 00:34	1 min 55 s	⋮
□	da2764ef	jmr1001/PROYECTO_TFG	main	e9db557	rmpgpb-projecto-tfg-europe-west1-jmr1001-PROYECTO-TFG-mavoz	26/4/23, 21:38	1 min 54 s	⋮
□	80111187	jmr1001/PROYECTO_TFG	main	e5957a8	rmpgpb-projecto-tfg-europe-west1-jmr1001-PROYECTO-TFG-mavoz	26/4/23, 20:45	2 min 12 s	⋮

Figura D.1: Cloud Build. Podemos observar la lista de compilaciones a lo largo del tiempo y los commits.

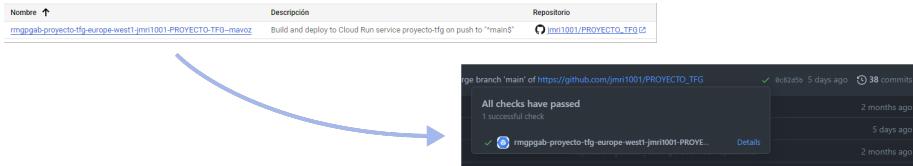


Figura D.2: Visualización del resultado del Trigger en en el repositorio de GitHub de nuestro proyecto.

3. **Almacenamiento de la imagen:** Como almacenamiento de la imagen de docker se ha usado otra herramienta interna de Google Cloud como alternativa a *docker registry* llamada *Artifact Registry*. Esto nos permite almacenar las imágenes de Docker de forma permanente.
4. **Despliegue:** Para generar nuestra aplicación y desplegarla de forma pública, hemos usado *Google Cloud Run*. Es un servicio serveless¹ para el despliegue de contenedores. Este servicio también nos proporciona el acceso mediante un dominio público usando DNS de nuestro proveedor de servicio(en nuestro caso, el proveedor es Porkbun)

¹Un servicio serveless o «sin servidor» es aquel que no mantiene la persistencia de los datos y se apaga cuando no hay tráfico. Cuando el tráfico vuelve se produce el denominado cold start o «inicio en frío».

Se puede acceder a nuestra aplicación web desde:

<https://www.anme.city>

Apéndice E

Documentación de usuario

E.1. Introducción

En el siguiente manual, se describen los pasos que un usuario debe seguir para utilizar nuestra aplicación web de manera sencilla y eficaz.

E.2. Requisitos de usuarios

A continuación, se enumeran los requisitos indispensables para poder usar la aplicación **ANME**:

- Disponer de un dispositivo con un navegador web (por ejemplo Chrome, Firefox, etc.).
- La aplicación **ANME** no funciona offline, por lo que se necesita una conexión a internet para hacer uso de la misma.
- La aplicación **ANME** es accesible desde cualquier dispositivo, pero se recomienda ser usada en un ordenador para tener una experiencia más óptima.

E.3. Instalación

En nuestro caso, no es necesario que el usuario instale ninguna herramienta para poder usar la aplicación.

E.4. Manual del usuario

Podemos acceder a la aplicación **ANME** desde <https://www.anme.city>. El usuario necesitará registrarse para hacer uso de la misma.

Inicio

Una vez que accedemos a la web de la aplicación veremos en la pantalla de nuestro dispositivo una ventana como la que aparece en la siguiente figura:

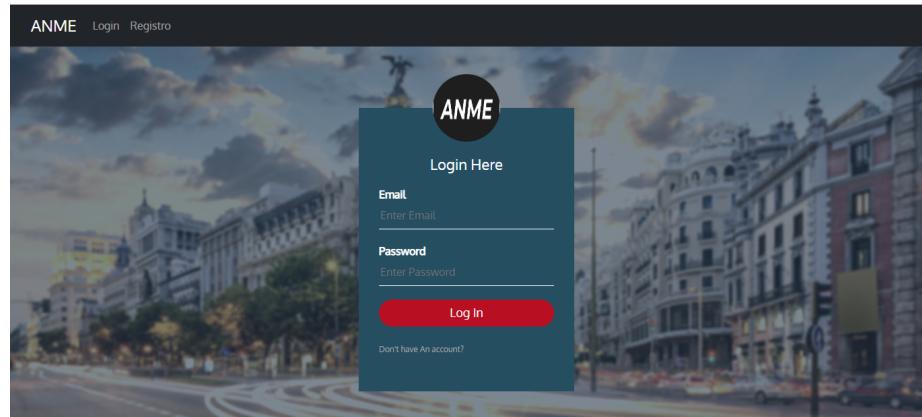


Figura E.1: Página de Inicio

En esta pantalla el usuario debe iniciar sesión con el usuario y contraseña con la que se ha registrado en la aplicación. Si el usuario no se ha registrado, debe crearse una cuenta en el botón superior izquierdo llamado **Registro**, el cual nos redirigirá a la ventana de registro.

En caso de que el campo Usuario o Contraseña sean incorrectos, se mostrará una notificación diciendo que el usuario o la contraseña son incorrectos.

Registro

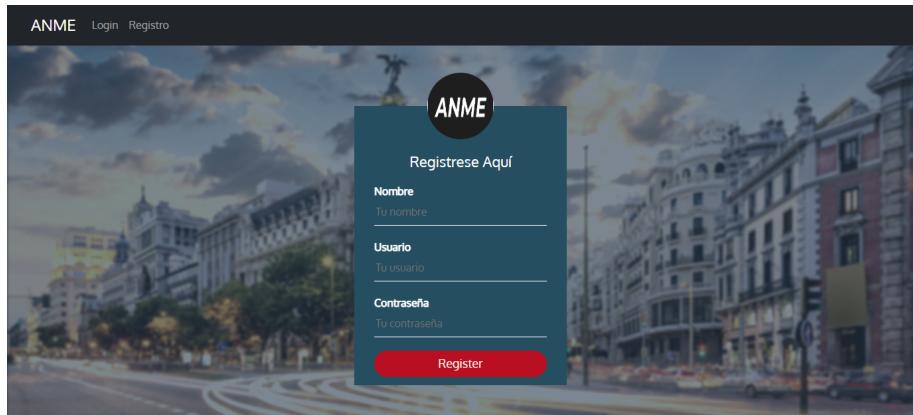


Figura E.2: Página de Registro

Tras haber hecho clic en el botón de **Registro** en la ventana de Inicio, este nos redirige a la ventana de Registro en la cual podremos registrarnos. Para registrarnos en la aplicación es necesario llenar los campos que nos aparecen en la ventana de Registro, que son: Nombre, Usuario y Contraseña.

Una vez que se han llenado todos los campos de la ventana Registro hacemos clic en el botón *Register*, si todos los campos son válidos el usuario se ha registrado exitosamente en la aplicación y se le redirigirá a la ventana de Inicio donde iniciará sesión. En caso contrario, se mostrará una notificación de error.

Barra de Navegación



Figura E.3: Barra de Navegación

La *barra de navegación* es el elemento que va a permitir al usuario poder navegar por la aplicación. Está formada por seis botones.

A continuación se detallará la funcionalidad de cada uno de los botones:

- Botón *Inicio*: Permite al usuario navegar a la ventana principal de la aplicación.

- Botón *Noticias*: Permite al usuario navegar a la ventana de Noticias en la que se muestran noticias de forma general de España.
- Botón *Eventos*: Este botón nos redirigirá a la pantalla de Eventos en la que se muestran todos los eventos disponibles en España.
- Botón *Meteorología*: Este botón nos redirigirá a la pantalla de Meteorología en la que se muestra el tiempo actual y una previsión de 6 días de la ubicación actual del usuario.
- Botón *Eventos Favoritos*: Este botón nos redirigirá a la pantalla de Eventos Favoritos en la que se muestran todos los eventos añadidos a favoritos por el usuario.
- Botón *Cerrar sesión*: Este botón permitirá al usuario cerrar su sesión de la aplicación.

Ventana Principal

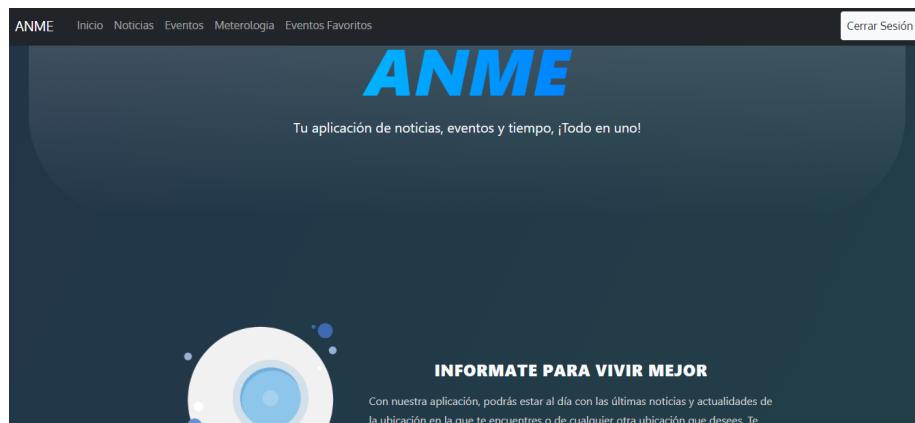


Figura E.4: Página principal

Tras haber iniciado sesión en la aplicación, la web nos redirige a la ventana Principal, donde nos aparece una barra de navegación y una breve descripción de los servicios que nos proporciona la misma.

Desde esta ventana el usuario ya puede acceder a los diferentes servicios que nos proporciona esta aplicación mediante la barra de navegación que aparece en esta ventana.

Noticias

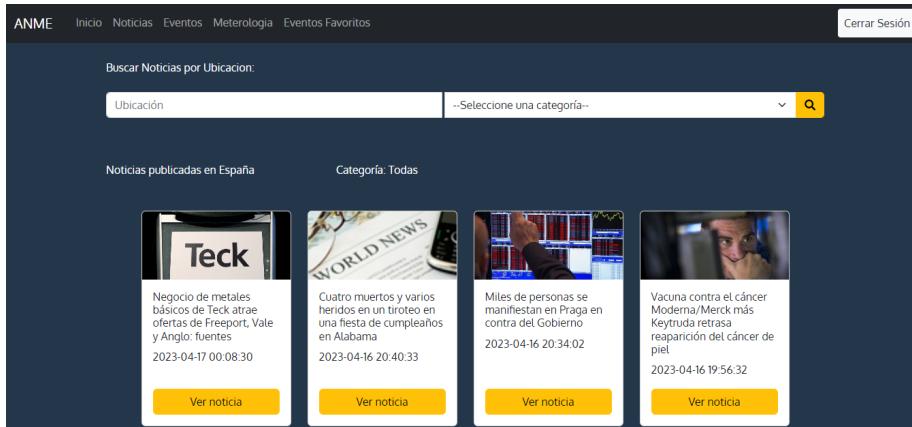


Figura E.5: Página de Noticias

En esta ventana se visualizan un gran número de noticias que han sido publicadas en España. Además, nos permite realizar un filtrado de noticias tanto por ubicación como por categoría en caso de que fuese de interés para el usuario.

Si el usuario deseara leer una noticia de la presente ventana, deberá hacer clic en el botón *Ver Noticia*, y este será redirigido a la noticia que ha solicitado leer.

Si se desea realizar una búsqueda de noticias por ubicación, hay que introducir la ubicación que deseemos en el filtro *Ubicación* y le daremos al botón de *Buscar*.



Figura E.6: Buscar Noticias por ubicación

Si se desea realizar una búsqueda de noticias por categoría, hay que seleccionar la categoría que deseemos en el filtro *Categorías* y le daremos al botón de *Buscar*.



Figura E.7: Buscar Noticias por categoría

También contamos con una barra de navegación en esta ventana desde la cual el usuario se le permite acceder a cualquier servicio de la aplicación.

Eventos

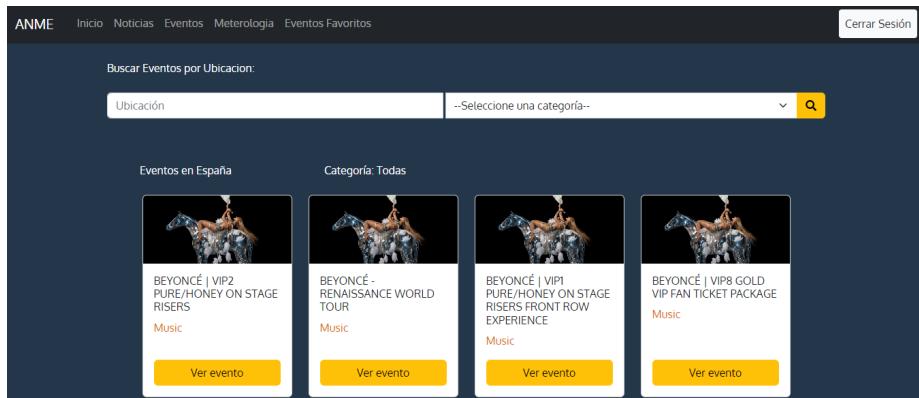


Figura E.8: Página de Eventos

En esta ventana se visualizan eventos de diferentes categorías, como eventos de música, deportes, películas, que están disponibles en toda España. Además, nos permite realizar un filtrado de eventos tanto por ubicación como por categoría en caso de que fuese de interés para el usuario.

Si el usuario deseara ver un evento de la presente ventana, deberá hacer clic en el botón *Ver Evento*, y este será redirigido a la página de información del propio evento.

Si se desea realizar una búsqueda de eventos por ubicación, hay que introducir la ubicación que deseemos en el filtro *Ubicación* y le daremos al botón de *Buscar*.

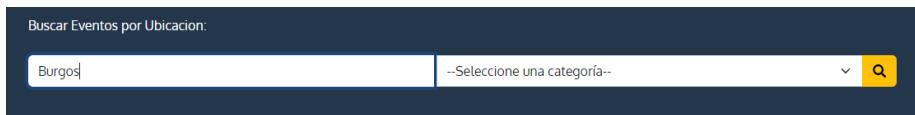


Figura E.9: Buscar eventos por ubicación

Si se desea realizar una búsqueda de eventos por categoría, hay que seleccionar la categoría que deseemos en el filtro *Categorías* y le daremos al botón de *Buscar*.

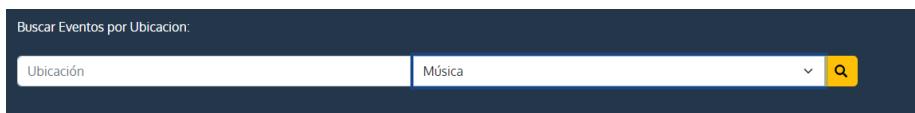


Figura E.10: Buscar eventos por categoría

También contamos con una barra de navegación en esta ventana desde la cual el usuario se le permite acceder a cualquier servicio de la aplicación.



Figura E.11: Barra de Navegación

Ventana Información del Evento

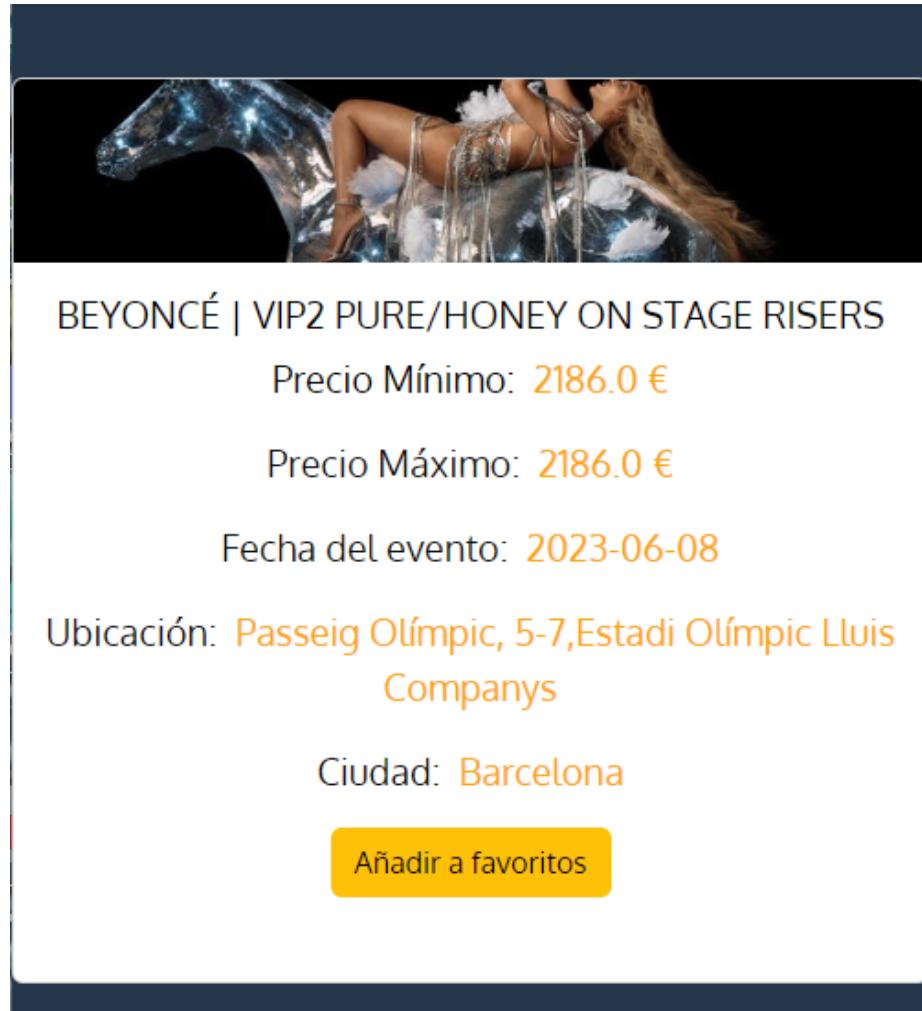


Figura E.12: Página de Información del Evento

Una vez que se ha pulsado el botón *Ver Evento* en la página de eventos, el sitio web redirige al usuario a la ventana de información del evento donde se muestran las características específicas del evento solicitado.

Si el usuario desea guardar un evento debido a su interés en él, simplemente debe hacer clic en el botón 'Añadir a favoritos', y el evento se guardará automáticamente en la lista de eventos favoritos del usuario.

Meteorología

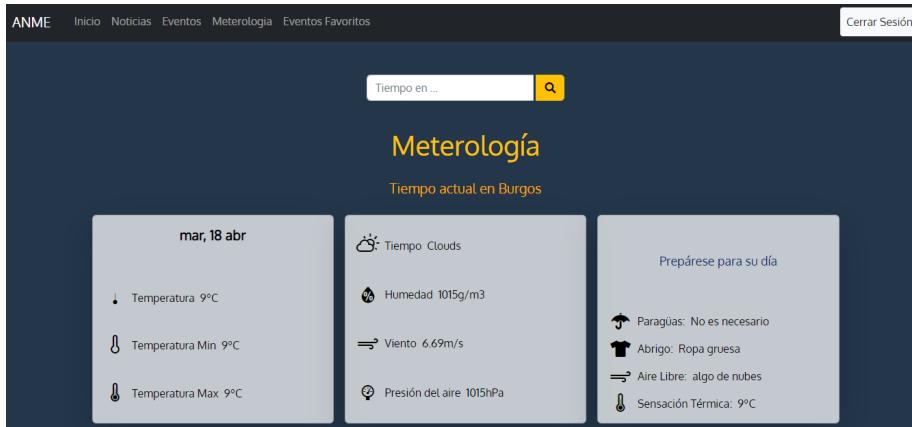


Figura E.13: Página de Meteorología



Figura E.14: Página de Meteorología

En esta ventana, se presenta la información meteorológica de la ubicación actual del usuario, donde el usuario puede conocer las características del tiempo actual. Además, el usuario dispone de la previsión del tiempo para una ubicación específica durante los próximos seis días. Esto le permite al usuario planificar sus actividades al aire libre o hacer planes de viaje con anticipación.

Si el usuario decide consultar la meteorología para una ubicación diferente a la actual, el usuario deberá ingresar la ubicación deseada en la barra de búsqueda de la ventana actual y presionará el botón "Buscar". De esta manera, el usuario podrá visualizar la información del clima para la ubicación especificada.

Eventos Favoritos

En esta ventana el usuario podrá visualizar los eventos que ha añadido a su lista de favoritos. Además, para cada evento se visualiza la información meteorológica de la ciudad donde se llevará a cabo el propio evento. Esto le permite al usuario anticiparse.

Si el usuario lo desea, puede visualizar todos los eventos guardados en favoritos en un mapa para que pueda ubicarlos de manera sencilla.

Si un evento deja de ser de interés para el usuario, puede eliminarlo de la lista de favoritos pulsando el botón *Eliminar de Favoritos*.



Figura E.15: Lista de Eventos Favoritos



Figura E.16: Meteorología de Eventos Favoritos

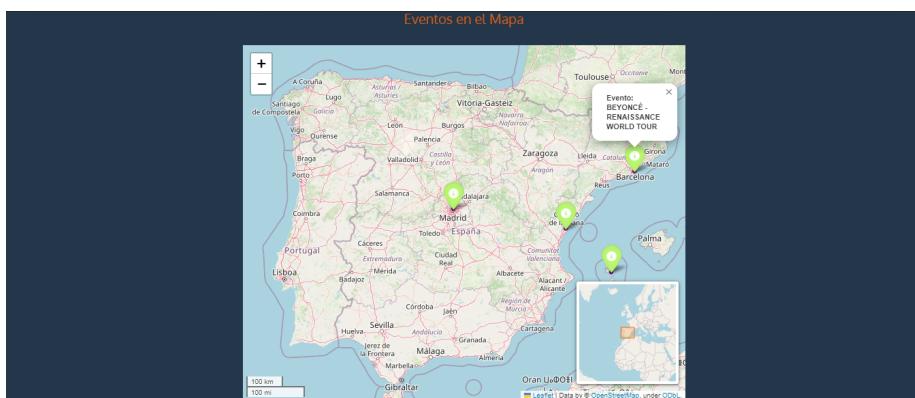


Figura E.17: Mapa de Eventos Favoritos

Bibliografía

- [1] Ana Cristina Domingo Troncho. Metodología scrum. <http://hdl.handle.net/10609/17885>, June 2012.
- [2] Joaquín Medina Serrano. Modelo–vista–presentador. http://joaquin.medina.name/web2008/documentos/informatica/documentacion/logica/patrones/patronMVP/2012_06_09_PatronMVP.html, November 2013.
- [3] Ernest Teniente López, Antoni Olivé Ramon, Enric Mayol Sarroca, and Cristina Gómez Seone. *Diseño de sistemas software en UML*. Univ. Politèc. de Catalunya, 2004.