



universidad
de león



Escuela de Ingenierías I. I.

Industrial, Informática y Aeroespacial

GRADO EN INGENIERÍA INFORMÁTICA

Trabajo de Fin de Grado

**Aplicación web de gestión de información de la
ciudad de León y alrededores**

Autor: Fernando Fernández Iglesias

Tutor: Lidia Sánchez González

UNIVERSIDAD DE LEÓN
Escuela de Ingenierías I.I.
GRADO EN INGENIERÍA INFORMÁTICA
Trabajo de Fin de Grado

ALUMNO: Fernando Fernández Iglesias

TUTOR: Lidia Sánchez González

TÍTULO: Aplicación web de gestión de información de la ciudad de León y alrededores

CONVOCATORIA: Septiembre, 2022

RESUMEN:

En este Trabajo de Fin de Grado se describe el diseño, planificación y elaboración de una aplicación web en la cual se mostrarán los datos, los cuales son recompilados mediante la extracción de estos de distintas páginas web como LeoNoticias, LeonOcio, yumping, minube, Terranostum utilizando otra aplicación que utiliza web Scraping. La aplicación que contiene el web Scraper obtiene de las páginas elegidas su HTML procesando el mismo y seleccionando los datos de importancia almacenándolos en la base de datos. La cual estará conectada a la aplicación web la cual es desplegada en el servidor de aplicaciones payara. Esta aplicación web está desarrollada utilizando la técnica móvil First desarrollando primero la interfaz para dispositivos móviles y después adaptándola para ordenador de sobremesas.

La aplicación web permite a los usuarios registrarse pudiendo acceder a información sobre León como puede ser actividades que realizar, eventos próximos, rutas cercanas, parques, Instalaciones deportivas, hoteles, restaurantes, noticias y edificios culturales. Los usuarios podrán seleccionar qué contenidos son sus favoritos además de poder realizar los comentarios oportunos. Esta aplicación también consta de un administrador encargado de comprobar si el contenido dentro de esta correctamente generado pudiendo eliminarlo en el caso de que esto no sea así, además de poder eliminar comentarios y usuarios.

Palabras clave: aplicación, web Scraper, Móvil First, Payara, Primefaces

Firma del alumno:

VºBº Tutor:

Índice de contenidos

Índice de contenidos	I
Índice de figuras	II
Índice de tablas	IV
Glosario de términos.....	VI
Introducción	1
Planteamiento del problema	1
Objetivos	1
Metodología.....	2
Estructura del trabajo	3
1 Estudio del problema.....	5
1.1 El contexto del problema	5
1.2 El estado de la cuestión	5
1.3 La definición del problema.....	7
2 Gestión de proyecto software	8
2.1 Alcance del proyecto.....	8
2.2 Plan de trabajo	10
2.3 Gestión de recursos	19
2.4 Gestión de riesgos.....	19
2.5 Legislación y normativa.....	22
3 Solución.....	23
3.1 Descripción de la solución.....	23
3.2 El proceso de desarrollo.....	23
3.3 El producto del desarrollo.....	66
4 Evaluación.....	76
4.1 Proceso de evaluación	76
4.2 Análisis de resultados.....	79
5 Conclusión.....	79
5.1 Aportaciones realizadas	79
5.2 Trabajos futuros	79
5.3 Problemas encontrados	79
5.4 Opinión personal.....	80
Lista de referencias	81
ANEXO A: Control de Versiones	83
ANEXO B: Seguimiento de proyecto fin de carrera.....	84
ANEXO C: Manual de usuario.....	86
C.1 Manual para Usuarios	86
C.2 Manual para Usuario Administrador	93
Anexo D: Manual del programador	97
D.1. Manual de instalación de la aplicación web.....	98

Índice de figuras

Figura 1.1 Modelo en cascada	2
Figura 1.1 SmartAppCity	5
Figura 1.2 Barcelona.cat.....	6
Figura 1.3 #MiCiudad	7
Figura 3.1 Diagrama de casos de uso (Usuario)	31
Figura 3.2 Diagrama de casos de uso (Administrador)	31
Figura 3.3 Diagrama de secuencia 1: Iniciar Sesión	33
Figura 3.4 Diagrama de secuencia 2: Ver contenido.....	34
Figura 3.5 Diagrama de secuencia 3: Eliminar contenido	36
Figura 3.6 Diagrama de secuencia 4: Añadir a Favoritos	38
Figura 3.7 Diagrama de secuencia 5: Ver comentarios.....	40
Figura 3.8 Diagrama de secuencia 6: Realizar comentarios.....	42
Figura 3.9 Diagrama de secuencia 7: Eliminar comentarios	44
Figura 3.10 Diagrama de secuencia 8: Ver Favoritos	45
Figura 3.11 Diagrama de secuencia 9: Eliminar Favoritos	47
Figura 3.12 Diagrama de secuencia 10: Ver avisos	48
Figura 3.13 Diagrama de secuencia 11: Crear avisos	50
Figura 3.14 Diagrama de secuencia 12: Visualizar Perfil.....	51
Figura 3.15 Diagrama de secuencia 13: Modificar Perfil.....	53
Figura 3.16 Diagrama de secuencia 14: Ver Usuarios.....	54
Figura 3.17 Diagrama de secuencia 15: Eliminar Usuario.....	56
Figura 3.18 Diagrama de secuencia 16: Web Scraper.....	58
Figura 3.19 Diagrama de dominio	59
Figura 3.20 Diagrama de arquitectura	60
Figura 3.21 Diagrama Entidad-Relación	61
Figura 3.22 Modelo Vista Controlador	62
Figura 3.23 Test unitarios.....	65
Figura 3.24 Aplicación Web Scraper	66
Figura 3.25 Temporizador contenido.....	67
Figura 3.26 Métodos comprobar conexión.....	67
Figura 3.27 Método obtener HTML	68
Figura 3.28 Web Scraper Noticia.....	68
Figura 3.29 Pagina inicial.....	69
Figura 3.30 Menú Aplicación.....	69
Figura 3.31 Interfaz parques ordenador	70
Figura 3.32 Interfaz parques móvil	70
Figura 3.33 Pagina ver Información noticia	71
Figura 3.34 Tarjeta hotel	71
Figura 3.35 Interfaz Comentarios.....	71
Figura 3.36 Interfaz avisos	72
Figura 3.37 Interfaz favoritos	72
Figura 3.38 Contenido administrador	73
Figura 3.39 Tabla usuarios administrador	73
Figura 3.40 HTML Noticia	73
Figura 3.41 Llamada a EJB	74
Figura 3.42 Clase modelo Avisos.....	74

Figura 3.43 Consulta JPQL	75
Figura 3.44 Método Verificar Usuario.....	75
Figura A.1 Diagrama de Gantt.....	85
Figura C.1 Interfaz Iniciar sesión	86
Figura C.2 Interfaz recuperar contraseña	86
Figura C.3 Interfaz registrarse	87
Figura C.4 Aviso Contraseña invalida	87
Figura C.5 Menú de la aplicación	88
Figura C.6 Pagina interfaz actividades	88
Figura C.7 Pregunta interfaz añadir a favoritos	89
Figura C.8 Contenido añadido a favoritos.....	89
Figura C.9 Interfaz comentarios	89
Figura C.10 Filtro de contenido	90
Figura C.11 Visualizar noticias.....	90
Figura C.12 Creación de un aviso	91
Figura C.13 interfaz favoritos.....	91
Figura C.14 Interfaz mi perfil.....	92
Figura C.15 Interfaz primer inicio administrador.....	93
Figura C.16 interfaz rutas administrador	94
Figura C.17 Eliminar Contenido.....	94
Figura C.18 Tabla usuarios	95
Figura C.19 Interfaz Web Scraper	96
Figura C.20 Filtro web Scraper	96
Figura D.1 Como crear Clases EJB	97
Figura D.2 Carpeta dominio/lib.....	98
Figura D.3 Iniciar Payara	98
Figura D.4 Pool de conexión.....	99
Figura D.5 Propiedades	99
Figura D.6 JDBC recurso	99
Figura D.7 Persistence.xml.....	99

Índice de tablas

Tabla 2.1. Presupuesto del Personal.....	9
Tabla 2.2 Gastos Totales	9
Tabla 2.3 Historia de usuario HU01	12
Tabla 2.4 Historia de usuario HU02	12
Tabla 2.5 Historia de usuario HU03	12
Tabla 2.6 Historia de usuario HU04	13
Tabla 2.7 Historia de usuario HU05	13
Tabla 2.8 Historia de usuario HU06	14
Tabla 2.9 Historia de usuario HU07	14
Tabla 2.10 Historia de usuario HU08	15
Tabla 2.11 Historia de usuario HU09	15
Tabla 2.12 Historia de usuario HU10	16
Tabla 2.13 Historia de usuario HU11	16
Tabla 2.14 Análisis de riesgos.....	21
Tabla 3.1 Requisito Funcional RF01	24
Tabla 3.2 Requisito Funcional RF02	24
Tabla 3.3 Requisito Funcional RF03	24
Tabla 3.4 Requisito Funcional RF04	24
Tabla 3.5 Requisito Funcional RF05	25
Tabla 3.6 Requisito Funcional RF06	25
Tabla 3.7 Requisito Funcional RF07	25
Tabla 3.8 Requisito Funcional RF08	26
Tabla 3.9 Requisito Funcional RF09	26
Tabla 3.10 Requisito Funcional RF10	26
Tabla 3.11 Requisito Funcional RF11	26
Tabla 3.12 Requisito Funcional RF12	27
Tabla 3.13 Requisito Funcional RF13	27
Tabla 3.14 Requisito Funcional RF14	27
Tabla 3.15 Requisito Funcional RF15	27
Tabla 3.16 Requisito Funcional RF16	28
Tabla 3.17 Requisito Funcional RF17	28
Tabla 3.18 Requisito Funcional RF18	28
Tabla 3.19 Requisito Funcional RF19	28
Tabla 3.20 Requisito Funcional RF20	29
Tabla 3.21 Requisito Funcional RF21	29
Tabla 3.22 Requisito Funcional RF22	29
Tabla 3.23 Requisito Funcional RF23	29
Tabla 3.24 Requisito Funcional RF24	30
Tabla 3.25 Caso de uso CU01	32
Tabla 3.26 Caso de uso CU03	35
Tabla 3.27 Caso de uso CU04	37
Tabla 3.28 Caso de uso CU05	39
Tabla 3.29 Caso de uso CU06	41
Tabla 3.30 Caso de uso CU07	43
Tabla 3.31 Caso de uso CU08	45
Tabla 3.32 Caso de uso CU09	46
Tabla 3.33 Caso de uso CU10	48

Tabla 3.34 Caso de uso CU11	49
Tabla 3.35 Caso de uso CU12	51
Tabla 3.36 Caso de uso CU13	52
Tabla 3.37 Caso de uso CU14	54
Tabla 3.38 Caso de uso CU15	55
Tabla 3.39 Caso de uso CU16	58

Glosario de términos

Contenido: La palabra contenido se utiliza para referirse instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales y actividades

CSS: Cascading Style Sheets.

EJB: Enterprise JavaBeans.

HTML: HyperText Markup Language.

JPA: Java Persistence API.

JSF: JavaServer Faces.

MVC: Modelo vista controlador estilo de arquitectura del software que separa los datos de la aplicación en tres partes

Primefaces: biblioteca de componentes para JFS.

Scrum: marco de trabajo para desarrollo ágil.

Sprint: nombre que recibe cada uno de los ciclos o iteraciones que se tienen dentro de dentro de un proyecto Scrum.

SQL: lenguaje de dominio específico, diseñado para administrar base de datos

Url: Uniform Resource Locator.

Web Scraper: es una técnica utilizada para extraer información de sitios web.

Introducción

Planteamiento del problema

Este trabajo de fin de grado intenta resolver el problema al que se enfrentan las persona al buscar información sobre eventos, actividades, restaurantes, hoteles, noticias, edificios culturales, instalaciones deportivas, rutas en León donde el usuario tiene que realizar muchas búsquedas para conseguir la información que este quiere.

Para eso a través de este trabajo se pretende desarrollar una herramienta que permita recompilar esta información repartida en distintas páginas web y mostrándola en una sola ahorrándole tiempo a las personas a la hora de realizar sus búsquedas.

Objetivos

El objetivo principal de este proyecto consiste en el diseño de una aplicación web Scraper el cual recompila la información y una aplicación que se encarga de mostrarla.

Por ello los objetivos son:

- Creación de una aplicación Web Scraper que recompila la información sobre noticias, todo tipo de actividades, eventos, restaurantes, hoteles, edificios culturales, parques, rutas e instalaciones deportivas la página web muestre.
- Creación de una aplicación web que muestre el contenido recompilado por el web Scraper, activándose esta cada hora para las noticias y cada día para el resto de contenido debido a que las noticias es un contenido que se actualiza cada poco mientras que los eventos, actividades y el resto de las cosas se actualizan menos frecuentemente.
- Utilización de la técnica Móvil First desarrollando primero las interfaces en móvil después adaptándola a ordenador.
- Creación de la aplicación utilizando el patrón MVC.
- Creación de una base de datos que contenga todos los contenidos que genere el web Scraper y el generado por la aplicación web.
- Aplicar la tecnología ágil Scrum para el desarrollo del software
- Crear la aplicación utilizando JFS, JPA, Primefaces y EJB.
- Utilizar Git como herramienta de control de versiones guardando el código en GitHub
- Utilizar y perfeccionar el máximo de conocimientos obtenidos durante el grado, como la programación orientada a objetos, gestión de la base de datos y programación web.

Metodología

En este apartado se describen las metodologías utilizadas durante el ciclo de vida del desarrollo del proyecto, el cual sigue el modelo en cascada y que se puede apreciar en la figura 1.1.

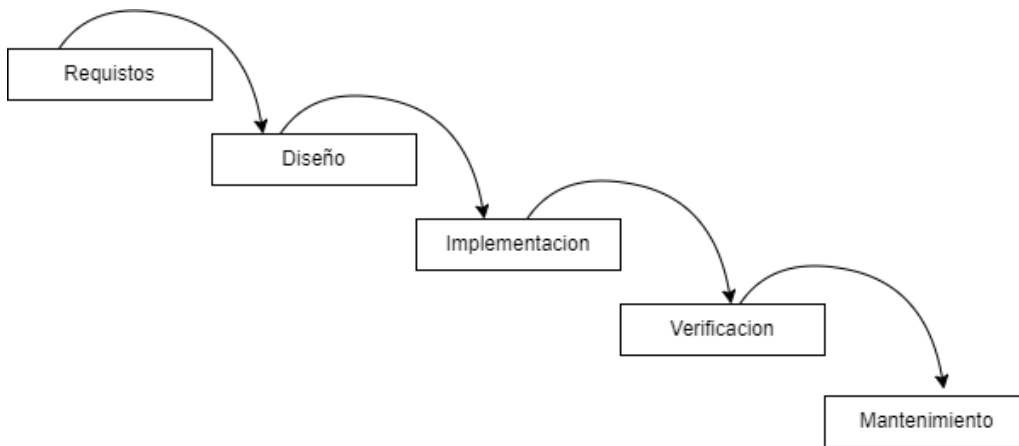


Figura 1.1 Modelo en cascada

Además, durante la fase de implementación se ha utilizado la metodología scrum, la cual entra dentro de las metodologías ágiles las cuales están pensadas para aceptar los cambios y proporcionar tiempo de desarrollo reducidos y de alta calidad. Se ha escogido esta metodología por la resistencia que tiene ante los cambios que puedan surgir durante el desarrollo del proyecto.

Dentro de esta metodología podemos encontrar 3 roles principales:

- **Product Owner:** El cual representa a los usuarios/clientes de este proyecto. Este es el que define los objetivos principales para obtener el producto final deseado.
- **Equipo de desarrollo (Scrum Team):** Representa a el equipo encargado del desarrollo de la aplicación y que entrega el producto al cliente.
- **Scrum Máster:** Representa a la persona encargada de enseña y guiar al equipo para utilizar el scrum correctamente, resolviendo los problemas que el equipo pueda encontrar.

Esta metodología empieza por la creación de la Pila del producto (Product Backlog), donde se elaboran las historias de usuario, las cuales marcaran los requisitos principales del proyecto.

Una vez realizadas las historias de usuario la siguiente etapa consiste en crear la pila de sprints con las tareas que hay que realizar en cada historia de usuario para completarla. Un sprint representa un periodo de tiempo en el cual se deben realizar las tareas que se encuentran en la pila del sprint.

Cada sprint cuenta con las siguientes etapas:

- Reunión de planificación del sprint: reunión dedicada a revisar que tiene mayor prioridad a la hora de establecer los objetivos del sprint, definiéndose las tareas de la pila del sprint dependiendo de la duración y la cantidad de miembros del equipo.
- Reunión diaria: Reunión de actualización y coordinación de tareas que se van a realizar durante el día.
- Reunión de revisión del sprint: Reunión realizada para inspeccionar el trabajo realizado, Si el resultado del sprint no fuera el esperado se alteraría la pila del producto añadiendo tareas al siguiente sprint.
- Reunión de retrospectiva: reunión por parte del equipo de desarrollo buscando mejorar el producto entre cada sprint.

Después de definir la metodología Scrum, se tomaron como tareas de la pila del producto las tareas que se detallan en el apartado 2.2.1. En cuanto a los Sprints se detallan en el apartado 2.2.3, los cuales tienen una duración de tres semanas cada uno, cada uno de ellos con una reunión de planificación y una reunión de revisión. Una vez realizada la reunión de revisión se realizaba una reunión de retrospectiva con aspectos a mejorar de cara al producto final.

Estructura del trabajo

- **Introducción:**
Es la sección actual, se presenta brevemente la estructura principal del proyecto y se detallan los objetivos y motivos para la realización de este trabajo, describiéndose también la estructura del documento.
- **Capítulo 1. Estudio del problema:**
En este capítulo, se describe el contexto de la realización del proyecto de forma que se revisan de forma detallada los trabajos, tecnologías y herramientas existentes acerca del problema a tratar.
- **Capítulo 2. Gestión del proyecto:** En este capítulo se encuentran los elementos que se han considerado para realizar la planificación del producto. Se encuentran detalladas las tareas ejecutadas a lo largo del ciclo de vida del proyecto, el alcance de este, la gestión de recursos necesarios y el marco legal.
- **Capítulo 3. Solución:**
En este capítulo se describe la solución desarrollada.
- **Capítulo 4. Evaluación:**
Este capítulo demuestra la validez de la solución descrita en el capítulo anterior, La forma de evaluación y el análisis de los resultados.
- **Capítulo 5. Conclusión:**
En esta sección, se detallan las ideas y conceptos adquiridos tras el desarrollo del proyecto, así como los problemas encontrados y la previsión del futuro.
- **Anexo A. Control de versiones:** En este anexo se detalla los controles de versiones usados.
- **Anexo B. Planificación del proyecto de fin de carrera:** Incluye diagrama de Gantt perteneciente a la planificación del proyecto.
- **Anexo C. Manual de usuario:** En este anexo se muestra al usuario los pasos que tiene que realizar al utilizar la aplicación.

- **Anexo D. Manual del programador:** En este anexo se describen los pasos que tiene que seguir el programador a la hora de programar las aplicaciones y un manual para la instalación de la página web en NetBeans.

1 Estudio del problema

1.1 El contexto del problema

Hoy en día hay una gran cantidad de información como puede ser noticias de actualidad, actividades, eventos, parques, rutas, turismo, entre otras muchas cosas que uno puede consultar a la hora de visitar una ciudad y no solo cuando se visita sino también siendo ciudadano de esta.

Pero esta información normalmente está descentralizada, es decir, se encuentra dispersa en diferentes páginas web y teniendo que gastar nuestro tiempo en visitar varias páginas hasta encontrar la información que estábamos buscando.

Por eso en algunas ciudades podemos encontrarnos con páginas web que se encarguen de agrupar la información pudiendo ser accesible desde una página web.

1.2 El estado de la cuestión

A continuación, mostraré algunos ejemplos de páginas web que ese encargan de centralizar la información de ciudades ayudando así al ciudadano o turista a recorrer u obtener información sobre la misma.

SmartAppCity

Es una aplicación de móvil actualmente en desarrollo la cual tiene una demo.

Esta página aplicación se encarga de centralizar los servicios que puedes encontrar en una ciudad, más concretamente de la ciudad permite segmentación entre turista y ciudadano debido a que pueden llegar a necesitar diferente información mostrando la información que necesitara el usuario dependiendo de la elección.

La información que te puedes encontrar en la aplicación es la misma tanto para el ciudadano como para el turista, lo que cambia cuando realizas esta elección es la información que se muestra en la portada.

La aplicación detecta la ubicación donde te encuentras para sacar la información de la ciudad más cercana, aunque actualmente solo se puede encontrar información de Logroño.



Figura 1.1 SmartAppCity

Aplicación web Barcelona.cat

Aplicación web centrada en la ciudad de Barcelona donde podemos encontrar información de todo tipo.

Podemos encontrar información referente a lugares de interés para turistas llevándote esta página a otras más especializadas en esto como puede ser checkbarcelona mostrándote un mapa para saber cómo llegar al destino correspondiente.

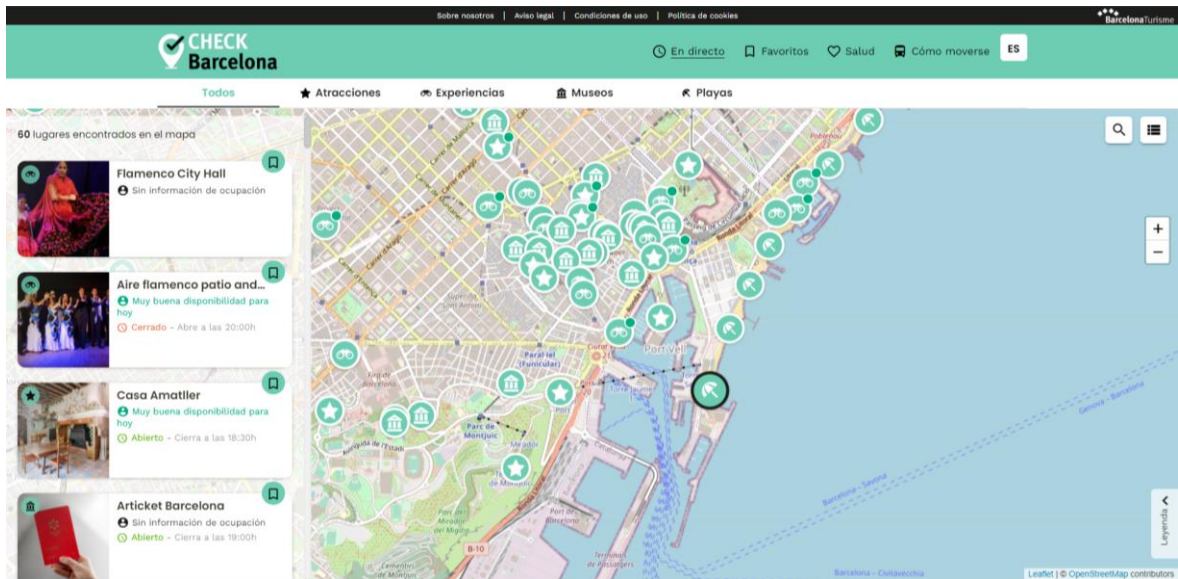


Figura 1.2 Barcelona.cat

También se puede encontrar un apartado para el ciudadano con información sobre parques, bibliotecas, institutos, supermercados, etc.

Además de tiene un apartado llamado trabajo y empresa donde podemos encontrar información sobre el mundo laboral, empresarial y del emprendimiento.

#MiCiudad

Aplicación móvil diseñada para venderse a los ayuntamientos la cual sirve para gestionar información de la ciudad como puedes ser la gestión de publicaciones de noticias y eventos.

Dentro de la aplicación podemos encontrar un apartado de dedicado a incidencias donde los ciudadanos pueden colaborar generando avisos sobre las incidencias ocurridas en la ciudad.

Por último, como en algunas aplicaciones ya tratadas anteriormente podemos encontrar apartados dedicados al turismo pudiendo ver los puntos de interés que podemos encontraremos en dicha ciudad.

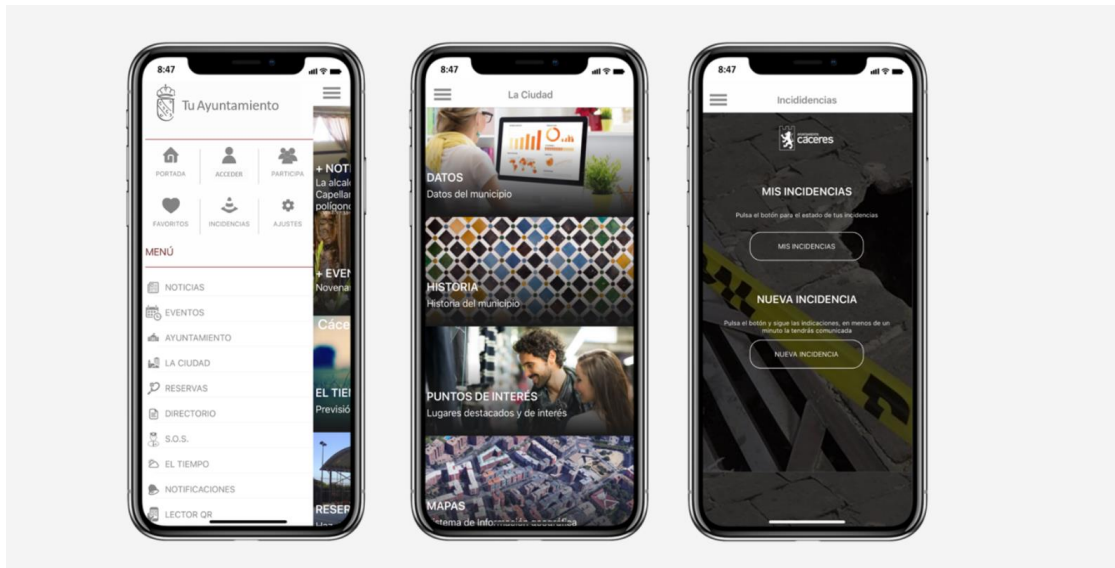


Figura 1.3 #MiCiudad

1.3 La definición del problema

Como podemos ver dentro de una ciudad hay una gran cantidad de información repartida en distintas páginas web y actualmente en León no existe alguna aplicación que las agrupe. Por lo tanto, la intención del presente proyecto es crear una página web que agrupe esta información.

La página web sería desarrollada mediante la técnica móvil First, es decir primero se daría prioridad a la visualización en dispositivos móviles y después la estructura se adaptaría para ordenadores. El diseño móvil First ha tomado fuerza en los últimos años debido a que la cantidad de móviles ya supera la cantidad de ordenadores a la hora de usar internet.

Dentro de esta aplicación podemos encontrar distinta información de interés actual como noticias de actualidad, eventos, actividades, también podremos encontrar lugares de interés como parques, monumentos, museos, instalaciones deportivas, rutas.

Toda esta información que podemos encontrar en distintas páginas web será extraída mediante Web Scraping la cual es una técnica para extraer información de sitios web y posteriormente ser guardada en una base de datos para su posterior uso, pudiendo así crear una aplicación que centralice la información.

2 Gestión de proyecto software

2.1 Alcance del proyecto

2.1.1 Definición del proyecto

Se pretende desarrollar una página web que permita a los usuarios tener una herramienta que recompile toda la información de la ciudad de León como puede ser noticias, rutas, eventos, etc. Esta información está desperdigada en distintas páginas web lo que obliga a el usuario a visitar distintas paginas para encontrar la información. La página web consultará estas páginas web recopilando su información actual para su posterior uso y almacenándose en la base de datos.

La aplicación permitirá al usuario iniciar sesión para guardar el contenido: eventos, actividades, edificios culturales, instalaciones deportivas, parques, hoteles, rutas, restaurantes que más le interese en favoritos, crear avisos y realizar comentarios.

Existirá un usuario administrador que se encargará de iniciar el web Scraper que se encarga de recopilar información. Este usuario podrá eliminar, modificar contenido indeseado que se encuentre en la aplicación.

2.1.2 Estimación de tareas y recursos

2.1.2.1 Tareas:

Este proyecto se desarrolla utilizando la metodología ágil Scrum. Empezando por una fase que abarca la investigación, diseño del sistema y planificación, seguida de 6 Sprints que incluirán las tareas que se detallan en el apartado del plan de trabajo

2.1.2.2 Recursos:

Los recursos necesarios para desarrollar la aplicación web se dividen en varios tipos:

Recursos materiales:

Donde se incluirán la oficina para la planificación y el desarrollo de la aplicación, el gasto de la luz e internet, En relación con el hardware se necesita un ordenador por cada trabajador del equipo. No abra costes en tecnología software utilizado dado que todas son open source.

Recursos Humanos:

Se contrata un equipo para el desarrollo del proyecto. El equipo estará formado por un Scrum Master que guiará a el equipo en el uso de la metodología Scrum y el equipo de desarrollo formado por dos Desarrolladores junior, un diseñador gráfico y un Tester de software.

2.1.3 Presupuesto

El presupuesto tiene como finalidad calcular los gastos que se van a tener desde el inicio del proyecto hasta la finalización de este. El presupuesto estará formado por el presupuesto derivado de la contratación del personal y el presupuesto derivado del coste en material.

El presupuesto del personal se calcula en función de los gastos de cada miembro del equipo del proyecto. La estimación del gasto de los salarios se ha realizado a través de una búsqueda en Internet.

- Scrum Máster: Según Glassdor el promedio de este cargo en España está en 23 €/hora.(1)
- Desarrollador Software: El sueldo promedio de este cargo sería de 15€/hora según Glassdor.(2)
- Diseñador Gráfico: según la siguiente fuente el salario medio en España es de 10,35 €/hora.(3)
- Tester de software: según Glassdor la media de sueldo en España es de 13,23 €/hora.(4)

Rol	Cantidad(horas)	Salario(€/hora)	Total (€)
Scrum Máster	30	23,00 €	690,00 €
Desarrollador Software	200	15,00 €	3.000,00 €
Diseñador Gráfico	100	10,35 €	1.035,00 €
Tester de software	50	13,23 €	661,50 €
Total			5.386,00 €

Tabla 2.1. Presupuesto del Personal

Presupuesto material incluye los gastos en oficina, luz, compra de ordenadores para cada integrante del equipo, limpieza de la oficina.

Supone un 15% del presupuesto total y se indica en el epígrafe Costes indirectos.

Presupuesto total:

Presupuesto	Coste (€)
Personal	5.386,00 €
Costes indirectos (15%)	1150,42 €
Beneficio Industrial (11%)	841,42 €
Subtotal	7.377,84 €
Iva aplicable (21%)	1.549,35 €
Total	8.927,19 €

Tabla 2.2 Gastos Totales

2.2 Plan de trabajo

Dentro de este apartado se indicarán y definirán las tareas e hitos del desarrollo del proyecto.

2.2.1 Identificación de tareas

- Investigación
- Diseño de la interfaz de la página web.
- Planificación
- Implementación
 - Desarrollo del web Scraper (Sprint 1)
 - Identificación de Páginas con información
 - Creación y diseño del web Scraper
 - Creación de la Base de Datos (Sprint 2)
 - Diseño del diagrama relacional de la base de datos
 - Creación de la base de datos
 - Desarrollo de la interfaz (Sprint 3)
 - Implementación de la interfaz del usuario
 - Implementación de la interfaz del administrador
 - Desarrollo del software del sistema (Sprint 4)
 - Creación de las clases modelo
 - Conexión del sistema con la base de datos.
 - Creación de las clases de tipo controlador.
 - Implementación del web Scraper (Sprint 5)
 - Conexión del web Scraper la aplicación web.
 - Testeo del sistema (Sprint 6)
 - Pruebas del sistema
- Documentación del proyecto
- Revisión de la documentación

2.2.2 Estimación de tareas

En este apartado se detallan las tareas que se han mencionado en el apartado anterior.

Dentro de las cuales podemos encontrar la **introducción**, el **diseño**, la **planificación** y **documentación** las cuales son tareas que no se encuentran dentro de la metodología Scrum debido a que no pertenecen al entorno de desarrollo del software, ya que son fases previas a su implementación pero que componen una parte fundamental.

Investigación

Esta es la primera fase del proyecto por la cual se obtuvo a través de artículos internet información acerca de tecnologías para poder desarrollar la página web pudiendo estas cumplir con los requisitos.

Lo primero que se buscó debido a que este proyecto se basa en la recopilación de datos sobre León fue una tecnología encargada de esto, se optó por una librería de java llama Jsoup para la obtención del HTML de las páginas con datos

de interés y procesamiento de este, además de la búsqueda de páginas web necesarias para la obtención de los datos.

Después debido a la decisión de utilizar Jsoup se decidió que el lenguaje de programación en el que sería desarrollada la web sería Java, luego se llevó a cabo la búsqueda de tecnologías java (EJB, JPA) y el servidor donde se alojaría esta aplicación, se optó por payara servidor.

Finalmente, para crear las interfaces se investigó distintas opciones como puede ser React o Vue, pero se optó por JFS al ser un proyecto realizado en java junto con Primefaces debido a que es un framework que quería aprender.

Diseño de la interfaz de la página web

En esta fase se realizó una serie de diseños de prototipo de la interfaz que consiste en realizar un prototipo a papel de la interfaz que utilizara el usuario.

Dentro del diseño inicial incluía para cada apartado de la página web dos diseños un para móvil y otro para ordenador.

Planificación

La fase de planificación establece una propuesta de los materiales necesarios para el desarrollo de la aplicación y detalla tanto la gestión de requisitos como la estimación de los recursos para el proyecto.

Implementación

En esta fase comienza a aplicarse la metodología Scrum.

Participantes:

- Scrum Master
- Equipo de desarrollo:
 - 1 desarrollador de software
 - 1 diseñador grafico
 - 1 Tester del software

Dentro de la pila del producto podemos encontrar las historias de usuario que se especifican a continuación, dentro de ellas se establecen los siguientes campos:

- ID: Identificador único asignado a este elemento del proyecto, el formato será HUXX.
- Nombre: Nombre que describe brevemente la historia de usuario.
- Valor: Preferencia de cara al desarrollo de la historia de usuario respecto a las demás en un rango de 0 a 100.
- Estimación: Esfuerzo que requiere el desarrollo de las tareas en un rango de 0 a 10.
- Descripción: Explicación de la historia de usuario.
- Condiciones de satisfacción: Condiciones que se deben cumplir para que la historia de usuario se dé por concluida.
- Dependencias: las dependencias sobre otras historias de usuario para poder realizarlas.

Identificación de Páginas con información	
Id	HU01
Valor	80
Estimación	4
Descripción	Búsqueda de páginas web con información de León que se pueda recopilar utilizando web Scraping
Dependencias	
Condiciones de satisfacción	La página se conecta satisfactoriamente con el web Scraper

Tabla 2.3 Historia de usuario HU01

Creación y diseño del web Scraper	
Id	HU02
Valor	100
Estimación	8
Descripción	Diseño de las clases pertenecientes al web Scraper captando información de páginas web.
Dependencias	<ul style="list-style-type: none"> • HU01
Condiciones de satisfacción	<ul style="list-style-type: none"> • El web Scraper capta la información que se le solicita

Tabla 2.4 Historia de usuario HU02

Diseño del diagrama relacional de la base de datos	
Id	HU03
Valor	80
Estimación	6
Descripción	Diseño de las tablas, atributos y relaciones de la base de datos
Dependencias	
Condiciones de satisfacción	<ul style="list-style-type: none"> • Se ha completado el diseño

Tabla 2.5 Historia de usuario HU03

Creación de la base de datos	
Id	HU04
Valor	90
Estimación	7
Descripción	Creación de la base de datos, sus tablas, sus columnas y relaciones usando MySQL
Dependencias	<ul style="list-style-type: none"> • HU03
Condiciones de satisfacción	<ul style="list-style-type: none"> • Comprobar que se pueden crear tablas y columnas • Comprobar que se pueden realizar consultas

Tabla 2.6 Historia de usuario HU04

Implementación de la interfaz del usuario	
Id	HU05
Valor	80
Estimación	7
Descripción	Creación de las distintas interfaces pertenecientes al usuario normal de la aplicación web utilizando HTML, CSS y Java server faces con el framework Primefaces.
Dependencias	
Condiciones de satisfacción	<ul style="list-style-type: none"> • Cada interfaz funciona correctamente • Las interfaces tienen un diseño que escalable, es decir, un diseño para móvil y otro para ordenador

Tabla 2.7 Historia de usuario HU05

Implementación de la interfaz del administrador	
Id	HU06
Valor	80
Estimación	6
Descripción	Creación de las distintas interfaces pertenecientes al usuario administrador de la aplicación web utilizando HTML, CSS y Java server faces con el framework Primefaces.
Dependencias	
Condiciones de satisfacción	<ul style="list-style-type: none"> • Cada interfaz funciona correctamente • Las interfaces tienen un diseño que escalable, es decir, un diseño para móvil y otro para ordenador

Tabla 2.8 Historia de usuario HU06

Creación de las clases modelo	
Id	HU07
Valor	50
Estimación	3
Descripción	Creación de las clases modelo que sirven para la conexión con la base de datos
Dependencias	<ul style="list-style-type: none"> • HU04
Condiciones de satisfacción	<ul style="list-style-type: none"> • Se ha creado una clase modelo por cada tabla de la base de datos

Tabla 2.9 Historia de usuario HU07

Conexión del servidor con la base de datos	
Id	HU08
Valor	80
Estimación	7
Descripción	Creación de las clases entidad y sus beans de sesión para conectarse a la base de datos mediante EJB
Dependencias	<ul style="list-style-type: none"> • HU04
Condiciones de satisfacción	<ul style="list-style-type: none"> • La conexión entre el servidor y la base de datos permite crear datos, borrarlos y modificarlos.

Tabla 2.10 Historia de usuario HU08

Creación de las clases de tipo controlador	
Id	HU09
Valor	90
Estimación	9
Descripción	Creación de clases controladoras de cada interfaz
Dependencias	<ul style="list-style-type: none"> • HU05, HU06
Condiciones de satisfacción	<ul style="list-style-type: none"> • Cada interfaz tiene su clase controladora que le permite funcionar correctamente • Las clases controlador pueden conectarse con la base de datos

Tabla 2.11 Historia de usuario HU09

Conexión del web Scraper con la Base de datos	
Id	HU10
Valor	90
Estimación	9
Descripción	Unión del web Scraper desarrollado con la base de datos pudiendo introducir los datos conseguidos por este para su posterior uso.
Dependencias	<ul style="list-style-type: none"> • HU07, HU08, HU09
Condiciones de satisfacción	<ul style="list-style-type: none"> • El web Scraper se ejecuta correctamente y los datos son almacenados en la base de datos • Los datos que se han obtenido a través del web Scraper se muestran en la interfaz • El web Scraper se ejecuta cada día comprobando si hay nuevos datos que deban ser añadidos

Tabla 2.12 Historia de usuario HU10

Pruebas del sistema	
Id	HU11
Valor	70
Estimación	5
Descripción	Realización de pruebas para comprobar si el sistema cumple con los requisitos establecidos.
Dependencias	<ul style="list-style-type: none"> • HU09, HU10
Condiciones de satisfacción	<ul style="list-style-type: none"> • La aplicación funciona correctamente sin ningún fallo • El sistema cumple con los requisitos establecidos

Tabla 2.13 Historia de usuario HU11

Documentación y Revisión del Proyecto

La fase de documentación y revisión ponen fin al proyecto. Dentro de la documentación podemos ver descrito todo el sistema para su muestra y exposición. La fase de revisión de la documentación es una fase dedicada a la búsqueda de errores y la corrección de estos.

2.2.3 Planificación de tareas

Las tareas en las que se descompone el proyecto se han organizado de forma independiente excepto las de desarrollo que formarían parte de los sprints como historias de usuario, así lo indica la metodología Scrum.

Cada fase de sprint está formada por una reunión de planificación, el propio sprint y una reunión de revisión.

A continuación, se presenta una especificación de la organización del proyecto:

- **Investigación:** 08/03/22 - 17/03/22
 - Participantes: *Scrum Master* y *Scrum Team*.
- **Diseño de la interfaz de la página web:** 18/03/22 - 29/03/22
 - Participantes: *Scrum Master* y *Scrum Team*.
- **Planificación:** 30/03/22 - 08/04/22
 - Participantes: *Scrum Master* y *Scrum Team*.
- **Sprint 1:** 11/04/22 - 28/04/22
 - Participantes: *Scrum Master* y *Scrum Team*.
 - Reunión de planificación: Durante esta reunión se decidirán las historias de usuario que se van a realizar en el sprint, las cuales son las referentes a desarrollar el código perteneciente al web Scraper que saca la información.
 - Sprint: Las historias de usuario que forman la pila del sprint son: HU01 y HU02.
 - Reunión de revisión: En esta reunión se comprobó que todos las historias de usuario se completaron de forma satisfactoria.
- **Sprint 2:** 29/04/22 - 18/05/22
 - Participantes: *Scrum Master* y *Scrum Team*.
 - Reunión de planificación: Durante esta reunión se decidirán las historias de usuario que se van a realizar en el sprint, las cuales han sido las que pertenecen al diseño y creación de la base de datos.
 - Sprint: Las historias de usuario que forman la pila del sprint son: HU03 y HU04.
 - Reunión de revisión: En esta reunión se comprobó que todos las historias de usuario se completaron de forma satisfactoria.
- **Sprint 3:** 19/05/22 - 07/06/22
 - Participantes: *Scrum Máster* y *Scrum Team*.
 - Reunión de planificación: Durante esta reunión se decidirán las historias de usuario que se van a realizar en el sprint. En este sprint se decide se desarrollarán las interfaces pertenecientes a la aplicación.
 - Sprint: Las historias de usuario que forman la pila del sprint son: HU05 y HU06.
 - Reunión de revisión: Las tareas se realizaron correctamente, pero la interfaz a la hora de mostrar las imágenes que se encuentran en la base de datos no funciona. Esto se debía a un fallo a la hora de representar la imagen en JSF.

- **Sprint 4: 08/06/22 - 27/06/22**
Participantes: *Scrum Master* y *Scrum Team*.
 - Reunión de planificación: Durante esta reunión se decidirán las historias de usuario que se van a realizar en el sprint. Se decide realizar la parte backend del sistema, es decir, conexión con la base de datos, clases modelo y controladoras.
 - Sprint: Las historias de usuario que forman la pila del sprint son: HU07, HU08 y HU09.
 - Reunión de revisión: En esta reunión se comprobó que todas las historias de usuario se completaron de forma satisfactoria.
- **Sprint 5: 28/06/22 - 15/07/22**
Participantes: *Scrum Master* y *Scrum Team*.
 - Reunión de planificación: Durante esta reunión se decidirán las historias de usuario que se van a realizar en el sprint. Se decide implementar el web Scraper en el sistema guardando sus datos en la base de datos para su posterior uso.
 - Sprint: Las historias de usuario que forman la pila del sprint son: HU10.
 - Reunión de revisión: En un principio se intentó introducir el web Scraper dentro de la aplicación web desplegada en el servidor payara, pero provocaba que la respuesta de la aplicación web fuera más lenta de lo normal además de que debido a esto el servidor se reiniciaba por lo que se tomó la decisión de realizar el web Scraper en otra aplicación aparte. Por lo que se logró completar la historia del usuario.
- **Sprint 6: 18/07/22 - 04/08/22**
Participantes: *Scrum Master* y *Scrum Team*.
 - Reunión de planificación: Durante esta reunión se decidirán las historias de usuario que se van a realizar en el sprint. En este sprint se decide que se efectuarán las pruebas del sistema para verificar su correcto funcionamiento.
 - Sprint: Las historias de usuario que forman la pila del sprint son: HU11
 - Reunión de revisión: Las historias de usuario se completaron satisfactoriamente.
- **Documentación del proyecto: 05/08/22 - 30/08/22**
Participantes: *Scrum Master* y *Scrum Team*.
- **Revisión de la documentación: 30/08/22 - 06/09/22**
Participantes: *Scrum Master* y *Scrum Team*.

Una vez organizadas todas las tareas se presentan en el diagrama de Gantt el cual podemos encontrar en el Anexo B.

2.3 Gestión de recursos

2.3.1 Especificación de recursos

Humanos:

- Scrum Master
- Equipo de desarrollo:
 - 1 desarrollador de software
 - 1 diseñador grafico
 - 1 Tester del software

Materiales:

- 4 ordenadores

Costes Indirectos:

- Luz
- Internet
- Alquiler de oficina
- Servicio de limpieza
- Material de oficina

Económicos:

- Un total de 8.927,19 €, los cuales se han especificado en el apartado del presupuesto

Asignación de recursos

Cada miembro del equipo recibirá un ordenador personal para su trabajo asignado, además el uso de la luz, internet y el material de la oficina.

2.4 Gestión de riesgos

2.4.1 Identificación de riesgos

- **Riesgos externos:**
 - Problemas con el hardware de los ordenadores
 - Tecnologías open source deja de serlo.
- **Riesgos de la implementación del sistema:**
 - Fallo al conectar u obtener la información de las páginas a través del web Scraper
 - Fallo al conectar con la base de datos
 - Fallo al conectar con el servidor.
- **Riesgos de planificación**
 - Incumplimiento de los plazos
 - Incumplimiento de los objetivos.
 - Pérdida de datos

2.4.2 Análisis de riesgos

En la siguiente tabla podemos ver la probabilidad, impacto, causa y solución de los distintos riesgos identificados en el punto anterior.

ID	Riesgo	Probabilidad	Impacto	Causa	Solución
R01	Problemas con el hardware de los ordenadores	Baja	Alto	Un ordenador deja de funcionar debido a un fallo en sus componentes debido a un defecto de fabricación.	Consultar si el ordenador está en garantía para poder arreglarlo y aplazar las tareas correspondientes a la persona encargada de ese ordenador
R02	Tecnologías open source deja de serlo:	Baja	Bajo	Una tecnología utilizada en el proyecto que era open source y por lo tanto gratis deja de serlo	Comprobar si hay una tecnología open source parecida a la que se estaba usando para reemplazarla.
R03	Fallo al obtener la información de las páginas a través del web Scraper:	Medio	Medio	La página web de donde se obtenía la información deja de estar activa.	En este caso habría que esperar para ver si la página web vuelve a estar activa y seguir recolectando la información o en caso de que esta haya cerrado buscar una nueva página para recolectar la información.

R04	Fallo al conectar con la base de datos:	Baja	Medio	Mala programación de las clases entidad pertenecientes a las tablas de la base de datos. El servidor ha dejado de funcionar.	Revisión del programa correspondiente a la conexión con la base de datos. Volver a iniciar el servidor.
R05	Fallo al conectar con el servidor:	Baja	Media	Fallo en la red por motivos ajenos al proyecto	Realizar otras tareas mientras se soluciona el problema
R06	Incumplimiento de los plazos	Media	Medio	Una tarea ha llevado más tiempo del que se tenía previsto.	Aumentar el plazo para la realización de esa tarea.
R07	Incumplimiento de los objetivos.	Media	Alto	Mala organización a la hora de llevar a cabo la tarea	Buscar una manera de organización para llevar a cabo los objetivos del proyecto.
R08	Pérdida de datos	Bajo	Alto	Perdida de datos de la documentación o de datos referentes al proyecto	Intentar recuperar los documentos perdidos, si no se consigue volver a reescribirlos

Tabla 2.14 Análisis de riesgos

2.5 Legislación y normativa

Dentro de este apartado podemos encontrar las leyes y normas que se aplican en este proyecto.

El sistema guarda datos personales como puede ser el nombre, apellidos y correo electrónico por lo que se tiene que cumplir la Ley Orgánica 14/1996, de 13 de diciembre, de protección de datos de carácter personal que dice(5):

Artículo 1. Objeto. La presente Ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor e intimidad personal y familiar.

La web está desarrollada tanto para el uso de aplicaciones web como para ordenadores cumpliendo el Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público el cual dice (6):

Artículo 1. Objeto.1. Este real decreto tiene por objeto garantizar los requisitos de accesibilidad de los sitios web y aplicaciones para dispositivos móviles de los organismos del sector público y otros obligados incluidos en el ámbito de aplicación del artículo 2.2. A los efectos de este real decreto se entiende por accesibilidad el conjunto de principios y técnicas que se deben respetar al diseñar, construir, mantener y actualizar los sitios web y las aplicaciones para dispositivos móviles para garantizar la igualdad y la no discriminación en el acceso de las personas usuarias, en particular de las personas con discapacidad y de las personas mayores.

También en este proyecto se aplica la Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico.(7)

Artículo 1. Es objeto de la presente Ley la regulación del régimen jurídico de los servicios de la sociedad de la información y de la contratación por vía electrónica, en lo referente a las obligaciones de los prestadores de servicios incluidos los que actúan como intermediarios en la transmisión de contenidos por las redes de telecomunicaciones, las comunicaciones comerciales por vía electrónica, la información previa y posterior a la celebración de contratos electrónicos, las condiciones relativas a su validez y eficacia y el régimen sancionador aplicable a los prestadores de servicios de la sociedad de la información.

3 Solución

3.1 Descripción de la solución

La aplicación desarrollada permite la recolección de datos sobre León y alrededores los cuales son recolectados mediante un web Scraper. Dentro de los datos recolectados se encontrarían actividades, noticias, eventos, hoteles, parques naturales, rutas, cultura, eventos y restaurantes. Todos estos datos serían guardados dentro de la base de datos junto con el URL de la página web a la que pertenecen.

Dentro de la aplicación podemos encontrar dos tipos de usuarios: los usuarios y el administrador.

Los usuarios mediante un menú podrán acceder a las diferentes interfaces las cuales muestran los datos recolectados, estos datos serán mostrados con una imagen título e información además de estos se puede encontrar un botón “ver más” el cual te redireccionará a la página web de donde se ha obtenido esa información pudiendo así informarte más en profundidad.

Estos usuarios podrán registrarse dentro de la aplicación iniciando sesión, una vez iniciada la sesión podrán guardar los datos que más interesantes les parezcan en favoritos, además de que podrán realizar avisos sobre cualquier incidencia que haya en la ciudad como por ejemplo un corte de una calle o un accidente.

El administrador será la persona encargada de activar el web Scraper para que este recolecte la información, además de poder eliminar datos que se encuentren obsoletos y usuarios.

El web Scraper una vez activado y siempre que se encuentre activo cada cierto tiempo se ejecutara automáticamente en el caso de las noticias cada hora y para el resto del contenido cada día debido a que las noticias se actualizan más frecuentemente que el resto del contenido, además este contara con un filtro por el que poder hacer búsquedas de contenido concreto rápidamente y sin tener que esperar.

3.2 El proceso de desarrollo

3.2.1 Análisis

En este apartado se detallan los requisitos que debe cumplir la aplicación final describiéndose tanto los servicios que tiene como sus limitaciones. Estos requisitos tendrán que cumplirse para que la aplicación final sea aceptada como correcta. La especificación de requisitos de este proyecto está basada en el estándar IEE 830 - 1998 (8).

3.2.1.1 Definición de requisitos

Requisitos funcionales:

Referencia	RF01
Título	Crear Usuario
Descripción	Un usuario podrá darse de alta en la aplicación mediante una interfaz la cual te pedirá los siguientes datos: Nombre de usuario, email, nombre, apellidos y contraseña
Dependencia	-

Tabla 3.1 Requisito Funcional RF01

Referencia	RF02
Título	Eliminar Usuario
Descripción	El Administrador podrá eliminar a los usuarios que él quiera pudiendo ver su información y su última conexión a la aplicación para tomar decisiones.
Dependencia	RF12

Tabla 3.2 Requisito Funcional RF02

Referencia	RF03
Título	Modificar Usuario
Descripción	La aplicación web permitirá modificar los datos personales del cliente, como su puede ser nombre, apellido, email, contraseña, pero no podrá cambiar el nombre de usuario.
Dependencia	RF01

Tabla 3.3 Requisito Funcional RF03

Referencia	RF04
Título	Visualizar avisos
Descripción	El usuario podrá visualizar los avisos creados por otros usuarios sobre incidencias que estén ocurriendo en la ciudad como puede ser un corte de una calle o un accidente.
Dependencia	-

Tabla 3.4 Requisito Funcional RF04

Referencia	RF05
Título	Crear avisos
Descripción	El usuario podrá crear un aviso sobre alguna incidencia que haya podido ver en la ciudad, cada aviso está formado por el nombre de usuario del creador, asunto, descripción y fecha. Para poder realizar un aviso será necesario estar registrado en el sistema.
Dependencia	RF01

Tabla 3.5 Requisito Funcional RF05

Referencia	RF06
Título	Visualizar contenido
Descripción	El usuario podrá ver todo el contenido que se encuentra disponible en la página web, el contenido que se podrá encontrar será recopilado por el web Scraper y este contendrá instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales, actividades y eventos de la ciudad de León y alrededores. Cada contenido tendrá al menos una foto, un título y un botón ver más, el cual nos redirigirá a la página web desde donde el web Scraper ha sacado la información para que el usuario pueda ver más en detalle esta información.
Dependencia	-

Tabla 3.6 Requisito Funcional RF06

Referencia	RF07
Título	Añadir contenido a favoritos
Descripción	El usuario añadir cualquier contenido que le resulte de interés a favoritos pudiendo guardarlo para su posterior consulta.
Dependencia	RF12, RF06

Tabla 3.7 Requisito Funcional RF07

Referencia	RF08
Título	Visualizar favoritos
Descripción	El usuario visualizar el contenido que este allí añadido a favoritos anteriormente.
Dependencia	R07, RF12

Tabla 3.8 Requisito Funcional RF08

Referencia	RF09
Título	Eliminar contenido de favoritos
Descripción	los usuarios podrán eliminar el contenido que tengan en favoritos siempre que ellos así lo quieran.
Dependencia	R08, RF12

Tabla 3.9 Requisito Funcional RF09

Referencia	RF10
Título	Visualizar comentarios
Descripción	los usuarios ver los comentarios que hayan realizado otros usuarios sobre algún contenido que contenga de la página web como instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales, actividades y eventos
Dependencia	R06

Tabla 3.10 Requisito Funcional RF10

Referencia	RF11
Título	Realizar comentarios
Descripción	Un usuario podrá realizar comentarios sobre cualquier contenido que le haya sido de ayuda pudiendo realizar una reseña de él.
Dependencia	RF10, RF12

Tabla 3.11 Requisito Funcional RF11

Referencia	RF12
Título	Iniciar Sesión
Descripción	Cualquier usuario registrado previamente podrá Iniciar sesión en la aplicación mediante la interfaz de inicio de sesión.
Dependencia	RF01

Tabla 3.12 Requisito Funcional RF12

Referencia	RF13
Título	Creación del administrador
Descripción	La creación del administrador se realizará automáticamente la primera vez que se inicie la aplicación si este no se encuentra en la base de datos.
Dependencia	-

Tabla 3.13 Requisito Funcional RF13

Referencia	RF14
Título	Primer inicio del administrador
Descripción	El sistema detectará la primera vez que el administrador inicie sesión y se le pedirá un cambio de contraseña, así como los datos básicos como lo es nombre, apellidos, email.
Dependencia	-

Tabla 3.14 Requisito Funcional RF14

Referencia	RF15
Título	Web Scraper
Descripción	El administrador podrá activar y desactivar el web Scraper en cualquier momento. Una vez activado el web Scraper se activará automáticamente cada día buscando nuevo contenido para añadir a la web, a excepción de las noticias las cuales se actualizarán cada hora debido a que las noticias actualizan más frecuentemente que el resto del contenido.
Dependencia	-

Tabla 3.15 Requisito Funcional RF15

Referencia	RF16
Título	Eliminación del contenido
Descripción	El administrador podría ver la información que contiene la aplicación pudiendo eliminar los datos obsoletos o sin interés para la aplicación.
Dependencia	-

Tabla 3.16 Requisito Funcional RF16

Referencia	RF17
Título	Eliminación de comentarios
Descripción	El administrador podría ver los comentarios realizados por los usuarios y eliminarlos si este lo considera un comentario inapropiado.
Dependencia	RF10

Tabla 3.17 Requisito Funcional RF17

Referencia	RF18
Título	Visualización y eliminación de usuarios
Descripción	El administrador podría ver los usuarios y sus datos en una tabla además de su ultimo inicio de sesión pudiendo así enviar un correo informando de que su cuenta será eliminada si estos llevan más de un año sin contarse.
Dependencia	-

Tabla 3.18 Requisito Funcional RF18

Referencia	RF19
Título	Visualización de nuevos datos
Descripción	El administrador podrá visualizar una tabla los últimos datos creados por el web Scraper pudiendo corroborar que estos se han creado correctamente.
Dependencia	-

Tabla 3.19 Requisito Funcional RF19

Referencia	RF20
Título	Almacenamiento de datos
Descripción	Los datos obtenidos serán almacenados en la base de datos para su posterior uso.
Dependencia	-

Tabla 3.20 Requisito Funcional RF20

Referencia	RF21
Título	Menú
Descripción	La aplicación web constará de un menú desde el cual se podrá acceder a Noticias, eventos, deportes, avisos, cultura, hoteles, parques, restaurantes, rutas y actividades.
Dependencia	-

Tabla 3.21 Requisito Funcional RF21

Referencia	RF22
Título	Filtro de contenido
Descripción	las páginas donde se muestra el contenido constaran de un filtro para mejora de la búsqueda siempre que este sea necesario.
Dependencia	RF06

Tabla 3.22 Requisito Funcional RF22

Referencia	RF23
Título	Eliminar aviso
Descripción	El sistema una vez que pasen tres días de la creación del aviso eliminara el aviso automáticamente.
Dependencia	-

Tabla 3.23 Requisito Funcional RF23

Referencia	RF24
Título	Filtro Web Scraper
Descripción	El web Scraper contara con un filtro desde el cual el administrador puede realizar búsquedas de contenido concreto como: las actividades, los edificios culturales, los eventos, los hoteles, los parques, las instalaciones deportivas, los restaurantes y las rutas.
Dependencia	-

Tabla 3.24 Requisito Funcional RF24

Requisitos no funcionales:**Rendimiento:**

Cualquier operación realizada en la aplicación no debe de tardar más de un segundo en completarse, a excepción de la obtención de datos por el web Scraper.

Usabilidad:

La aplicación debe tener una interfaz intuitiva, para que cualquier usuario pueda utilizarla sin ninguna complicación.

Cada página de la aplicación tendrá una interfaz responsive tanto para dispositivos móviles como para ordenador.

Seguridad:

La aplicación dispondrá de un sistema de seguridad basado en registro con un nombre de usuario y contraseña. El nombre de usuario es único y no puede repetirse, y la contraseña debería estar formada por al menos 8 caracteres formados por números y letras.

Disponibilidad:

La aplicación debe estar disponible siempre que un usuario quiera acceder a ella.

Mantenibilidad:

Los desarrolladores de la aplicación, una vez finalizada realizaran un mantenimiento de este cada cierto tiempo pudiendo añadir más cantidad de información o modificando las páginas web donde se saca la información. Contribuyendo así a la mejora de la aplicación de cara al futuro.

3.2.1.2 Especificación de requisitos

En este apartado se describen los casos de uso mostrados en las figuras 3.1 y figura 3.2, los cuales complementan los requisitos expuestos en la sección anterior. Además de incluir sus respectivos diagramas de secuencia y el diagrama de dominio.

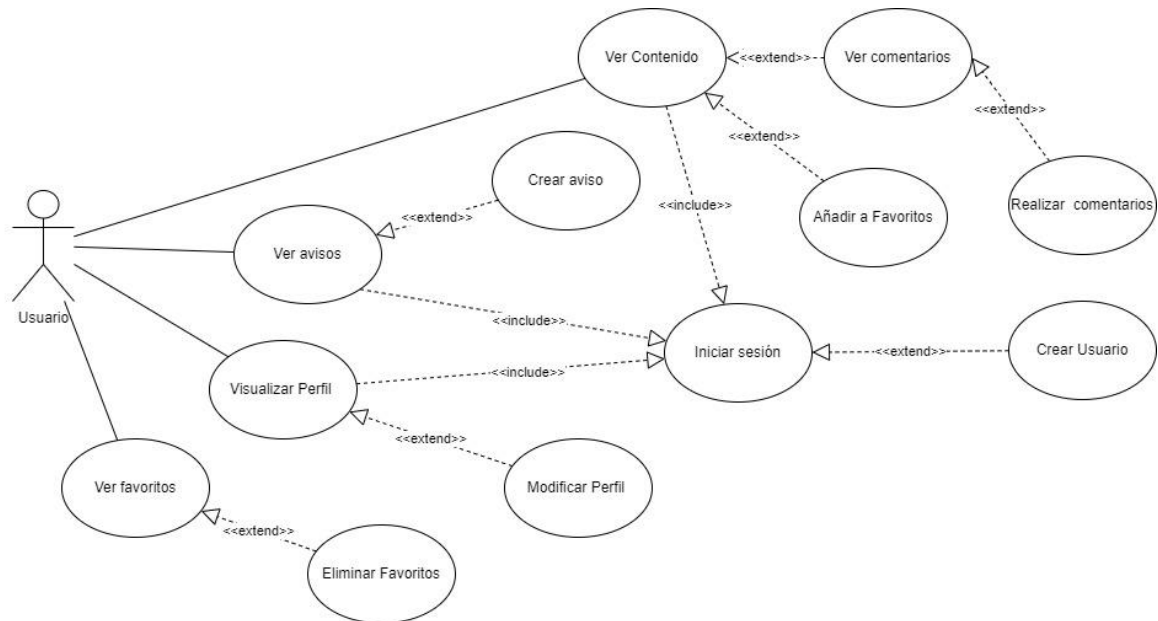


Figura 3.1 Diagrama de casos de uso (Usuario)

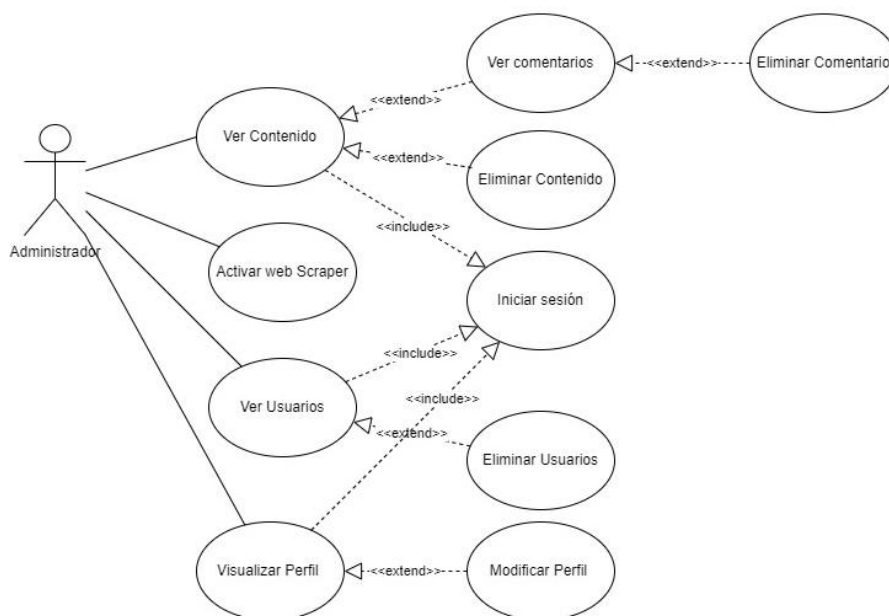


Figura 3.2 Diagrama de casos de uso (Administrador)

Número de caso de uso	CU01
Caso de Uso	Iniciar sesión
Actores	Usuario, Administrador
Resumen	Un usuario o administrador quiere iniciar sesión en la aplicación.
Precondiciones	La base de datos y el servidor se encuentra activo.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario abre la aplicación 2. El usuario o Administrador presiona el botón iniciar sesión 3. El sistema carga la interfaz iniciar sesión 4. EL usuario introduce su usuario, contraseña y presiona el botón iniciar sesión. 5. El sistema consulta la base de datos comprobando si estos datos son correctos. 6. El sistema inicia sesión y redirecciona al usuario a la página principal del usuario o administrador con la sesión iniciada
Postcondiciones	El usuario o administrador puede realizar las actividades en las que es necesario estar registrado.
Extensiones (o Flujo alternativo)	<ol style="list-style-type: none"> 3a. El usuario presiona el botón registrarse <ol style="list-style-type: none"> 1. El sistema carga la interfaz Registrar usuario. 2. El usuario introduce sus datos personales, junto a su nombre de usuario y contraseña. 3. El sistema comprueba si el nombre de usuario existe. 4. El sistema comprueba si la contraseña es valida 5. El sistema comprueba si el nombre de usuario existe 6. El sistema notifica al usuario que los datos introducidos son correctos y por lo tanto la cuenta se ha creado con éxito. 4a. El sistema avisa al usuario de que los datos introducidos son incorrectos.

Tabla 3.25 Caso de uso CU01

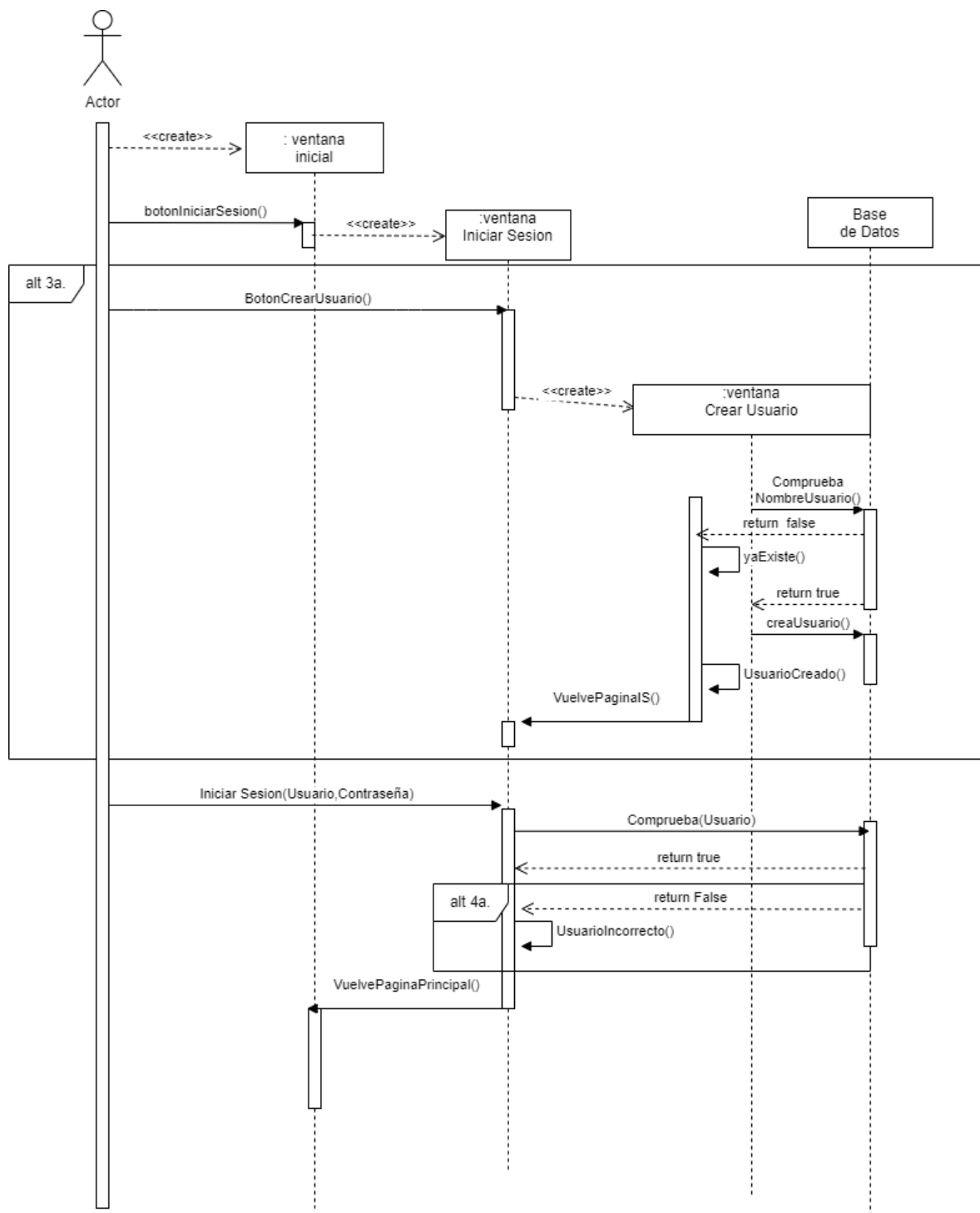


Figura 3.3 Diagrama de secuencia 1: Iniciar Sesión

Número de caso de uso	CU02
Caso de Uso	Ver contenido
Actores	Usuario, Administrador
Resumen	El sistema muestra el contenido (instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales) dependiendo de lo que el usuario haya elegido en el usuario en el menú.
Precondiciones	La base de datos y el servidor se encuentra activo. El web Scraper se ha activado por lo menos una vez. El usuario ha iniciado sesión.
Flujo de eventos	1. El usuario abre la aplicación 2. El usuario selecciona una categoría en el menú. 3. El sistema consulta la base de datos obteniendo los datos necesarios. 4. El sistema muestra en la interfaz los datos obtenidos.
Postcondiciones	El usuario puede ver los datos correctamente.
Extensiones (o Flujo alternativo)	

Tabla 3.25 Caso de uso CU02

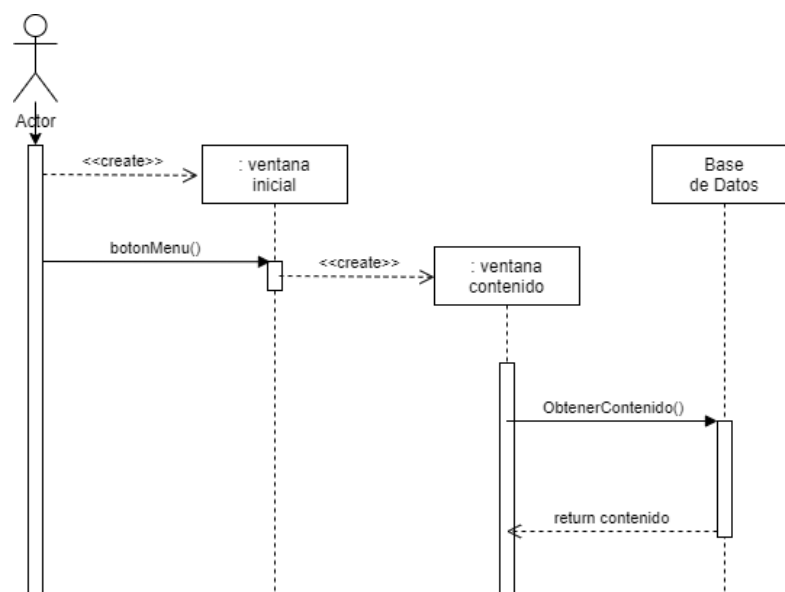


Figura 3.4 Diagrama de secuencia 2: Ver contenido

Número de caso de uso	CU03
Caso de Uso	Eliminar contenido
Actores	Administrador
Resumen	El administrador quiere eliminar un contenido (instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales) debido a que no se ha generado correctamente.
Precondiciones	La base de datos y el servidor se encuentra activo. El web Scraper se ha activado por lo menos una vez. El administrador ha iniciado sesión.
Flujo de eventos	<ol style="list-style-type: none"> 1. El administrador abre la aplicación 2. El administrador selecciona una categoría en el menú. 3. El sistema consulta la base de datos obteniendo los datos necesarios. 4. El sistema muestra en la interfaz los datos obtenidos. 5. El administrador pulsa el botón eliminar 6. El sistema envía contenido a eliminar de la base de datos. 7. La página se recarga.
Postcondiciones	El contenido es eliminado
Extensiones (o Flujo alternativo)	

Tabla 3.26 Caso de uso CU03

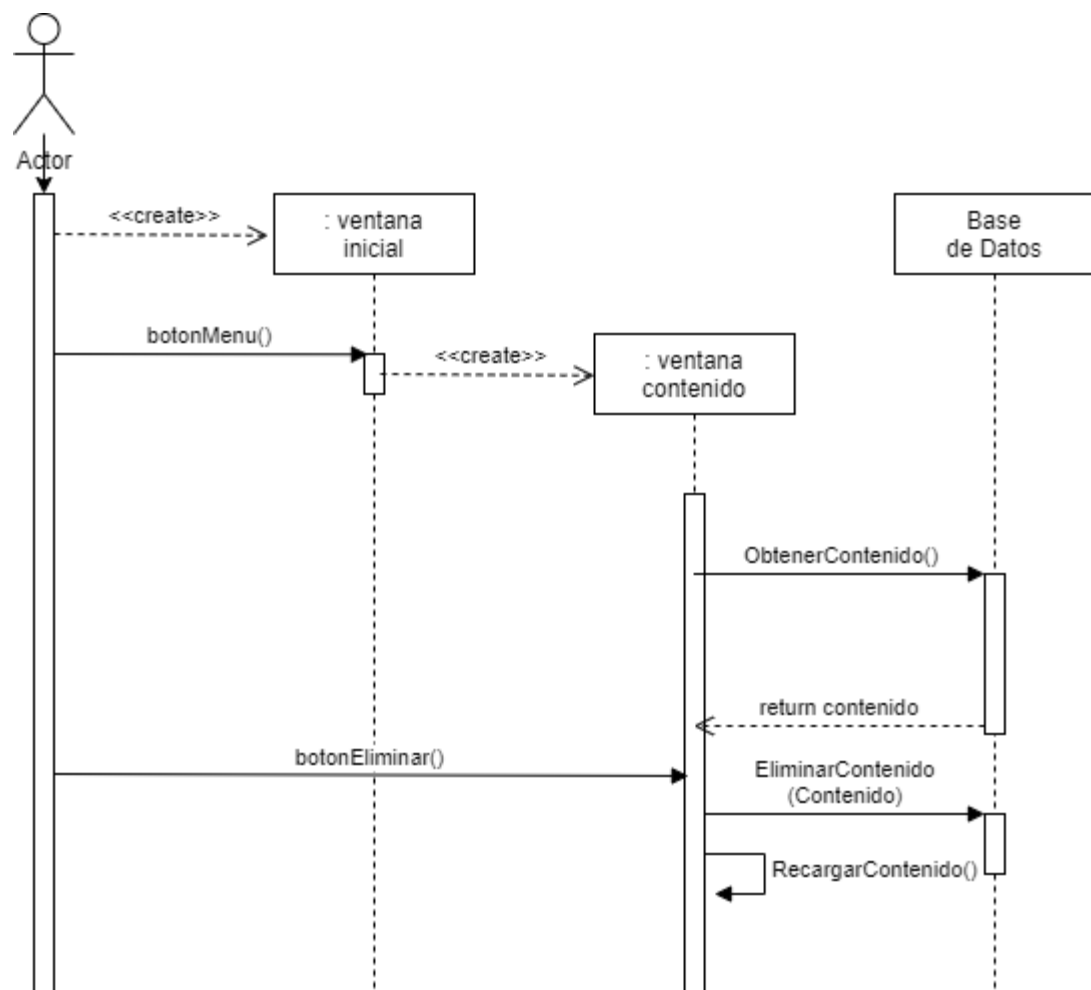


Figura 3.5 Diagrama de secuencia 3: Eliminar contenido

Número de caso de uso	CU04
Caso de Uso	Añadir a Favoritos
Actores	Usuario
Resumen	El usuario añade un contenido (instalaciones deportivas, hoteles, restaurantes, edificios culturales, rutas, parques naturales) a favoritos
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión.
Flujo de eventos	<ol style="list-style-type: none">1. El usuario abre la aplicación2. El usuario selecciona una categoría en el menú.3. El sistema consulta la base de datos obteniendo los datos necesarios.4. El sistema muestra en la interfaz los datos obtenidos.5. El usuario presiona el botón añadir a favoritos6. El sistema crea un favorito y lo añade a la base de datos7. El sistema notifica al usuario que se ha añadido correctamente
Postcondiciones	El contenido se ha añadido a favoritos.
Extensiones (o Flujo alternativo)	

Tabla 3.27 Caso de uso CU04

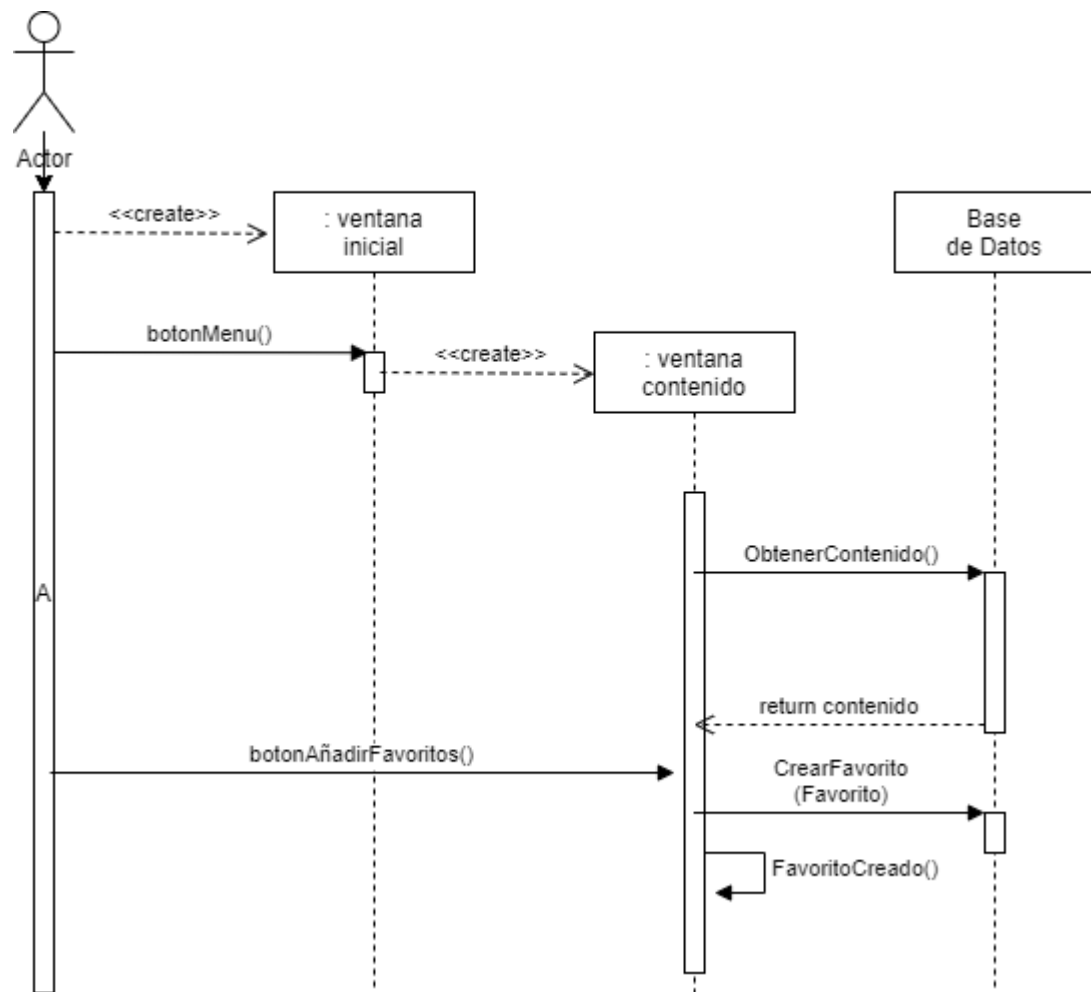


Figura 3.6 Diagrama de secuencia 4: Añadir a Favoritos

Número de caso de uso	CU05
Caso de Uso	Ver comentarios
Actores	Usuario, Administrador
Resumen	El Actor visualiza los comentarios
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión.
Flujo de eventos	<ol style="list-style-type: none">1. El Actor abre la aplicación2. El Actor selecciona una categoría en el menú.3. El sistema consulta la base de datos obteniendo los datos necesarios.4. El sistema muestra en la interfaz los datos obtenidos.5. El Actor presiona el botón ver comentarios de un contenido6. El sistema consulta la base de datos para obtener los comentarios7. El sistema muestra los comentarios
Postcondiciones	Se muestran los comentarios correctamente.
Extensiones (o Flujo alternativo)	

Tabla 3.28 Caso de uso CU05

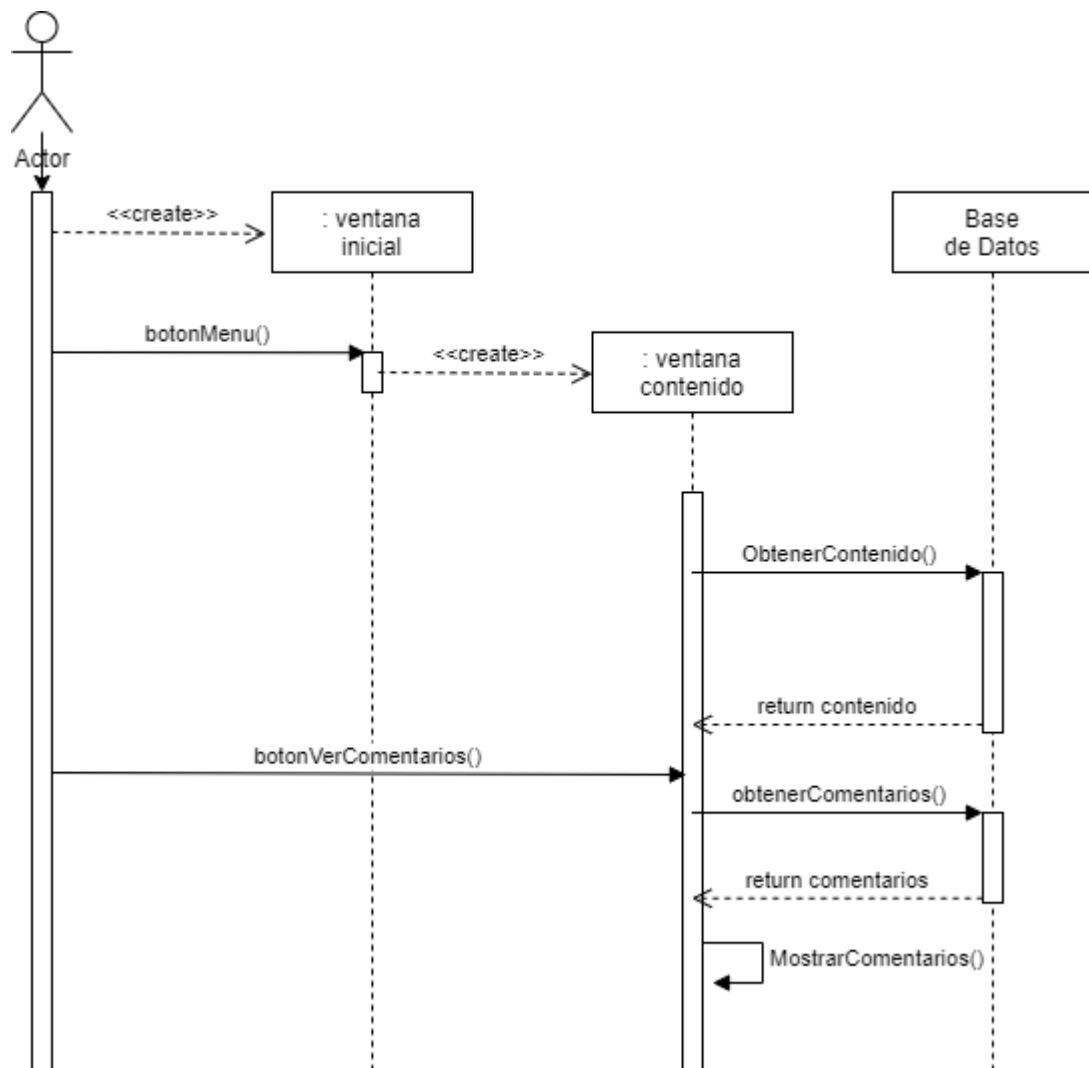


Figura 3.7 Diagrama de secuencia 5: Ver comentarios

Número de caso de uso	CU06
Caso de Uso	Realizar comentarios
Actores	Usuario
Resumen	El Actor visualiza los comentarios
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario abre la aplicación 2. El usuario selecciona una categoría en el menú. 3. El sistema consulta la base de datos obteniendo los datos necesarios. 4. El sistema muestra en la interfaz los datos obtenidos. 5. El usuario presiona el botón ver comentarios de un contenido 6. El sistema consulta la base de datos para obtener los comentarios 7. El sistema muestra los comentarios 8. El usuario presiona el botón añadir un comentario 9. El usuario crea el comentario. 10. El comentario se guarda en la base de datos.
Postcondiciones	El comentario se ha creado
Extensiones (o Flujo alternativo)	

Tabla 3.29 Caso de uso CU06

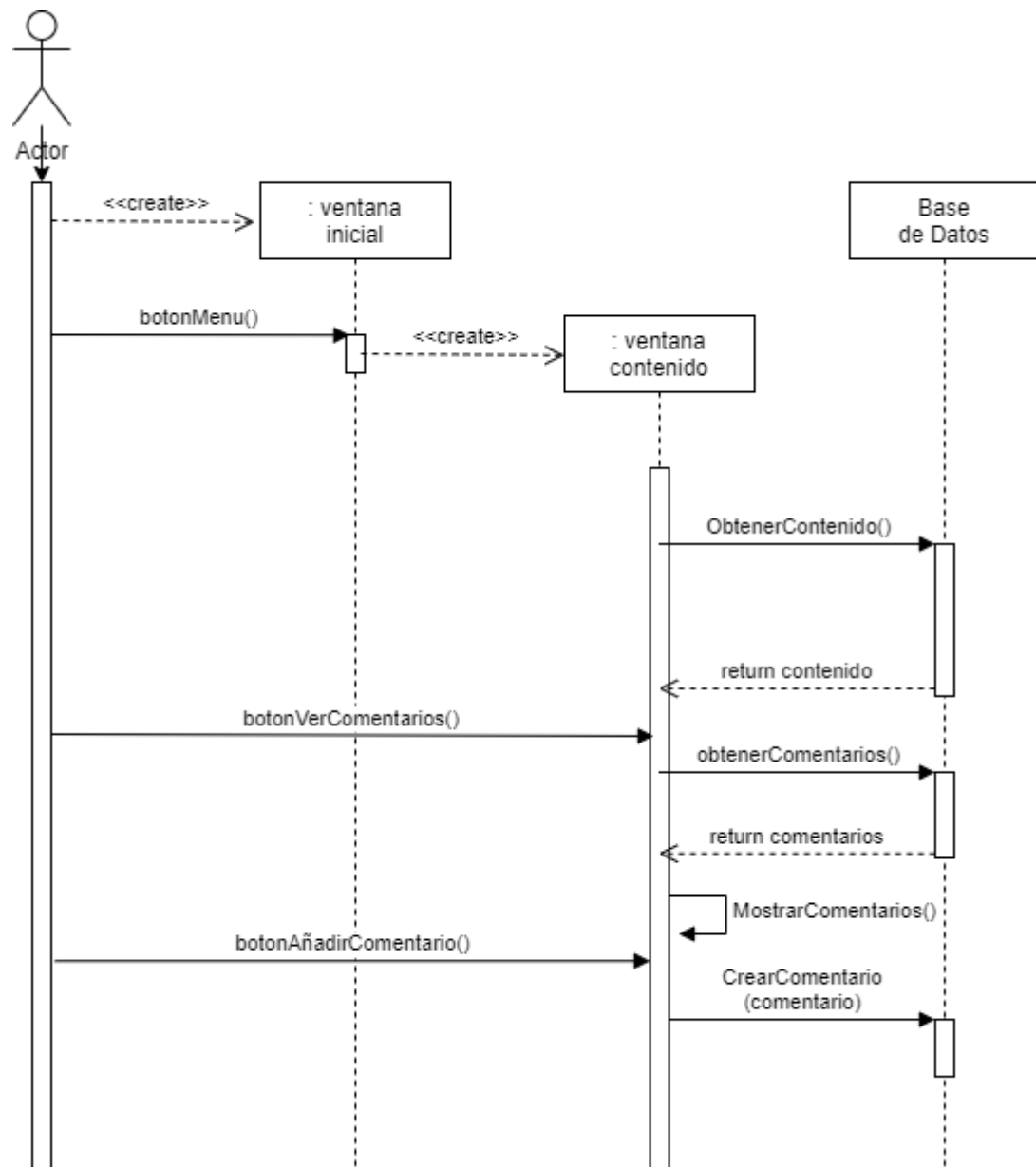


Figura 3.8 Diagrama de secuencia 6: Realizar comentarios

Número de caso de uso	CU07
Caso de Uso	Eliminar comentarios
Actores	Administrador
Resumen	El administrador elimina un comentario
Precondiciones	La base de datos y el servidor se encuentra activo. El administrador ha iniciado sesión.
Flujo de eventos	<ol style="list-style-type: none"> 1. El administrador abre la aplicación 2. El administrador selecciona una categoría en el menú. 3. El sistema consulta la base de datos obteniendo los datos necesarios. 4. El sistema muestra en la interfaz los datos obtenidos. 5. El administrador presiona el botón ver comentarios de un contenido 6. El sistema consulta la base de datos para obtener los comentarios 7. El sistema muestra los comentarios 8. El administrador presiona el botón eliminar un comentario 9. El comentario se elimina de la base de datos. 10. Los comentarios se recargan
Postcondiciones	El comentario se ha eliminado
Extensiones (o Flujo alternativo)	

Tabla 3.30 Caso de uso CU07

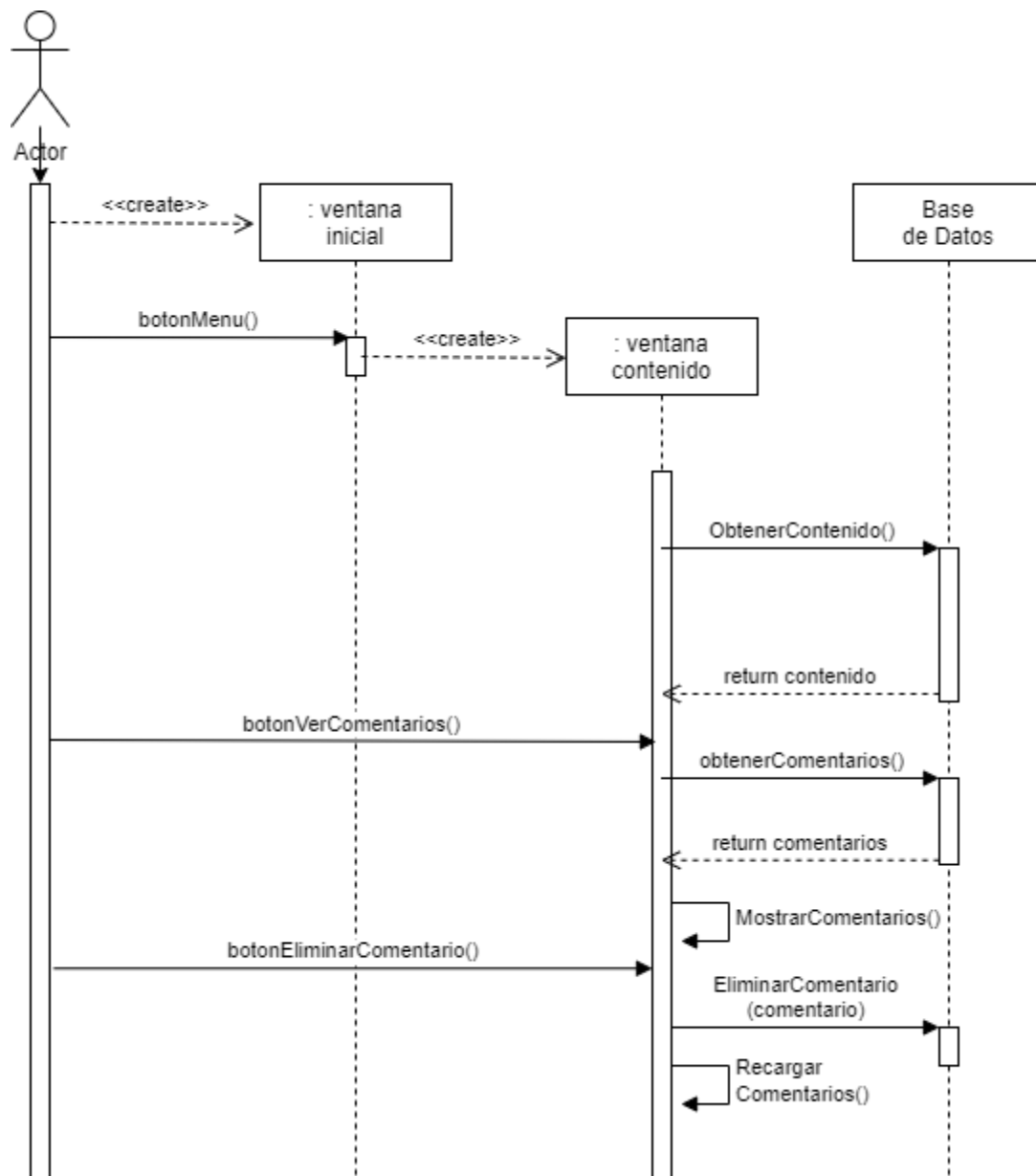


Figura 3.9 Diagrama de secuencia 7: Eliminar comentarios

Número de caso de uso	CU08
Caso de Uso	Ver favoritos
Actores	Usuario
Resumen	El usuario visualiza el contenido (instalaciones deportivas, hoteles, restaurantes, edificios culturales, rutas, parques naturales) que este tiene en favoritos.
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión en la aplicación.
Flujo de eventos	1. El usuario selecciona en el menú ver favoritos. 3. El sistema consulta la base de datos obteniendo los favoritos de ese usuario. 3. El sistema muestra el contenido que el usuario tiene en favoritos.
Postcondiciones	El sistema muestra el contenido que el usuario a añadido a favoritos.
Extensiones (o Flujo alternativo)	

Tabla 3.31 Caso de uso CU08

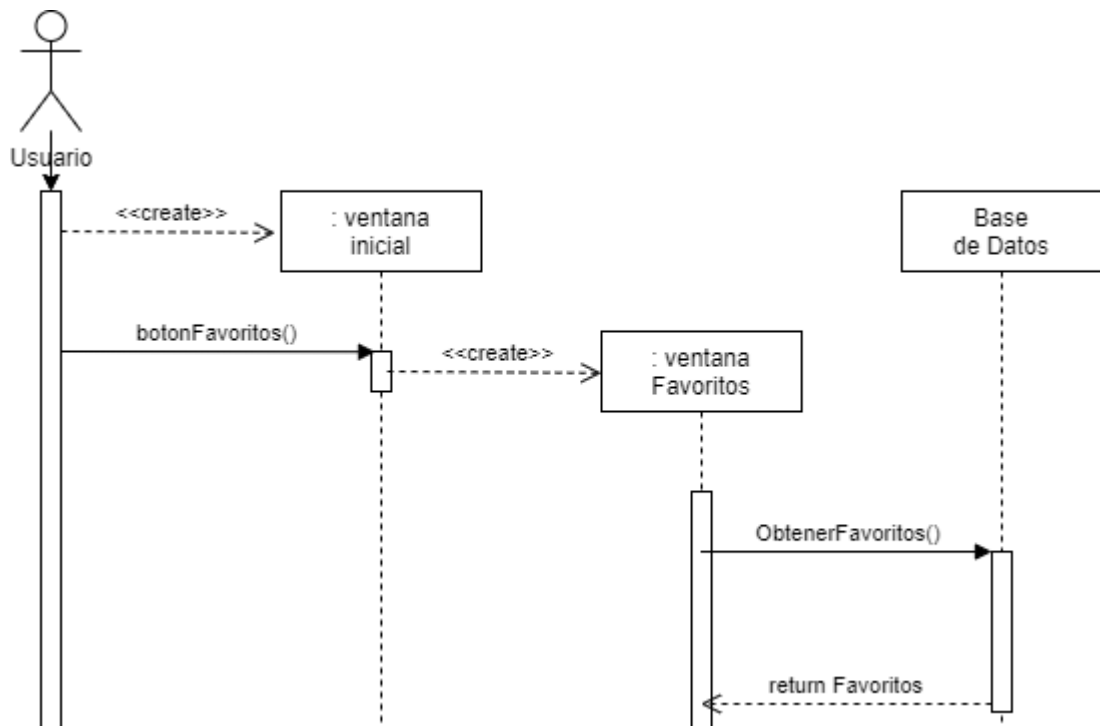


Figura 3.10 Diagrama de secuencia 8: Ver Favoritos

Número de caso de uso	CU09
Caso de Uso	Eliminar favoritos
Actores	Usuario
Resumen	El usuario elimina contenido (instalaciones deportivas, hoteles, restaurantes, edificios culturales, rutas, parques naturales) que se encuentra en favoritos.
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión en la aplicación.
Flujo de eventos	<ol style="list-style-type: none"> 1. El usuario selecciona en el menú ver favoritos. 3. El sistema consulta la base de datos obteniendo los favoritos de ese usuario. 3. El sistema muestra el contenido que el usuario tiene en favoritos. 4. El usuario presiona el botón eliminar favoritos. 5. El sistema elimina de la base de datos el favorito 6. El sistema recarga la pagina
Postcondiciones	El sistema muestra el contenido que el usuario a añadido a favoritos a excepción del eliminado.
Extensiones (o Flujo alternativo)	

Tabla 3.32 Caso de uso CU09

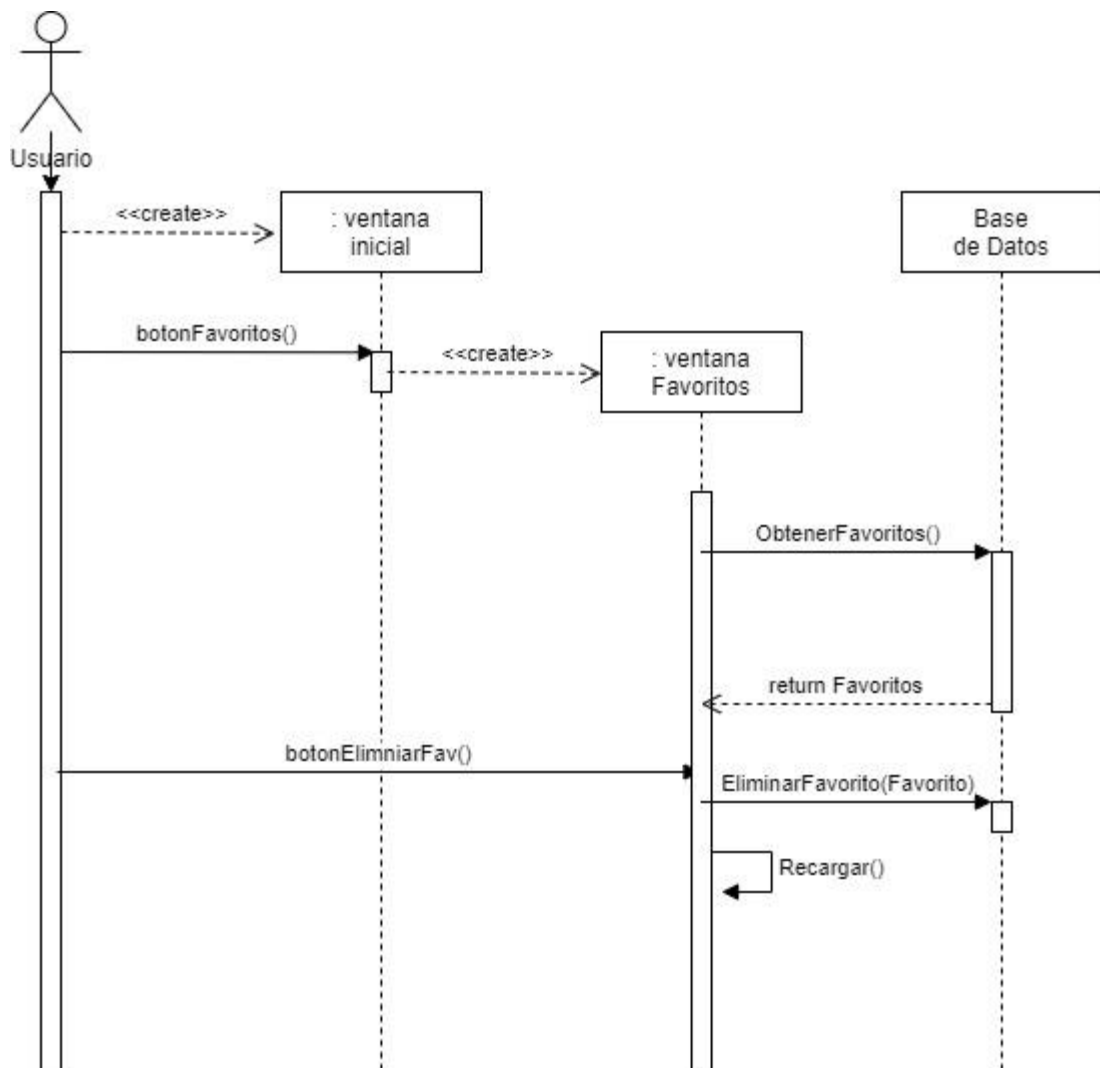


Figura 3.11 Diagrama de secuencia 9: Eliminar Favoritos

Número de caso de uso	CU10
Caso de Uso	Ver avisos
Actores	Usuario
Resumen	El usuario inicia la página avisos que contiene avisos sobre incidencias realizados por otros usuarios
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión en la aplicación.
Flujo de eventos	1. El usuario selecciona en el menú ver avisos. 3. El sistema consulta la base de datos obteniendo los avisos. 3. El sistema muestra los avisos encontrados.
Postcondiciones	El usuario visualiza los avisos correctamente.
Extensiones (o Flujo alternativo)	

Tabla 3.33 Caso de uso CU10

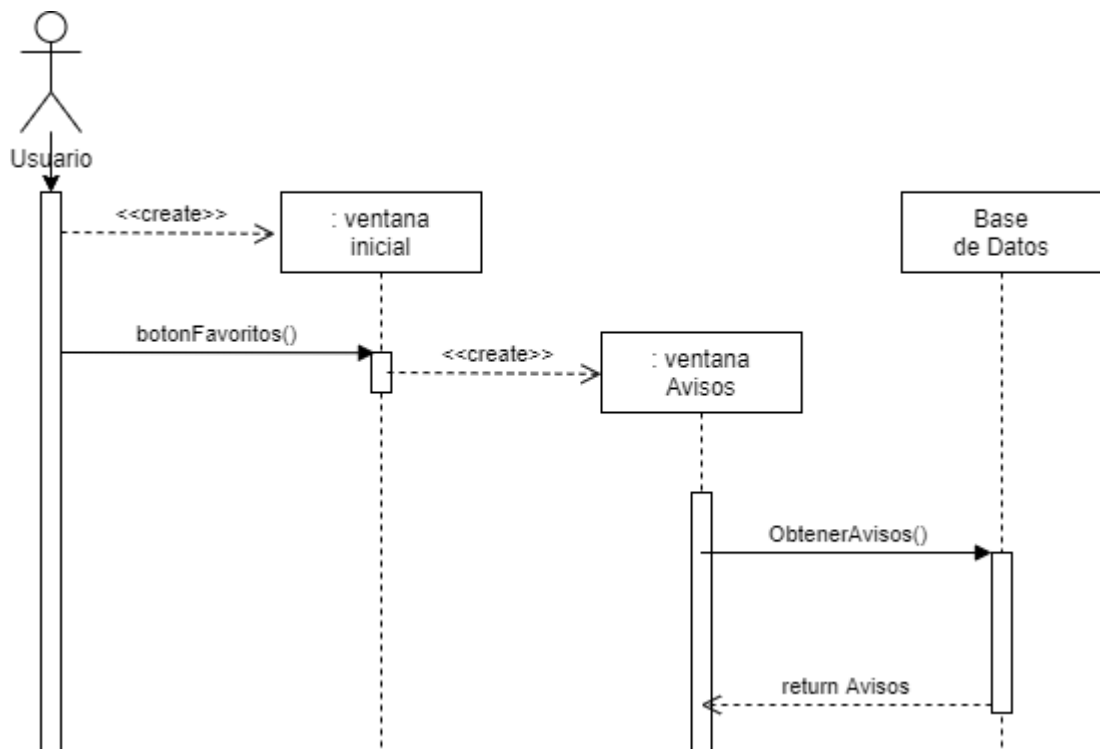


Figura 3.12 Diagrama de secuencia 10: Ver avisos

Número de caso de uso	CU11
Caso de Uso	Crear avisos
Actores	Usuario
Resumen	El usuario crea un aviso sobre alguna incidencia.
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión en la aplicación.
Flujo de eventos	<ol style="list-style-type: none">1. El usuario selecciona en el menú ver avisos.2. El usuario presiona el botón Crear aviso.3. El sistema le muestra un formulario para rellenar.4. El usuario rellena los datos y presiona el botón crear aviso5. El sistema crea un aviso y los añade a la base de datos
Postcondiciones	El usuario crea el aviso correctamente.
Extensiones (o Flujo alternativo)	

Tabla 3.34 Caso de uso CU11

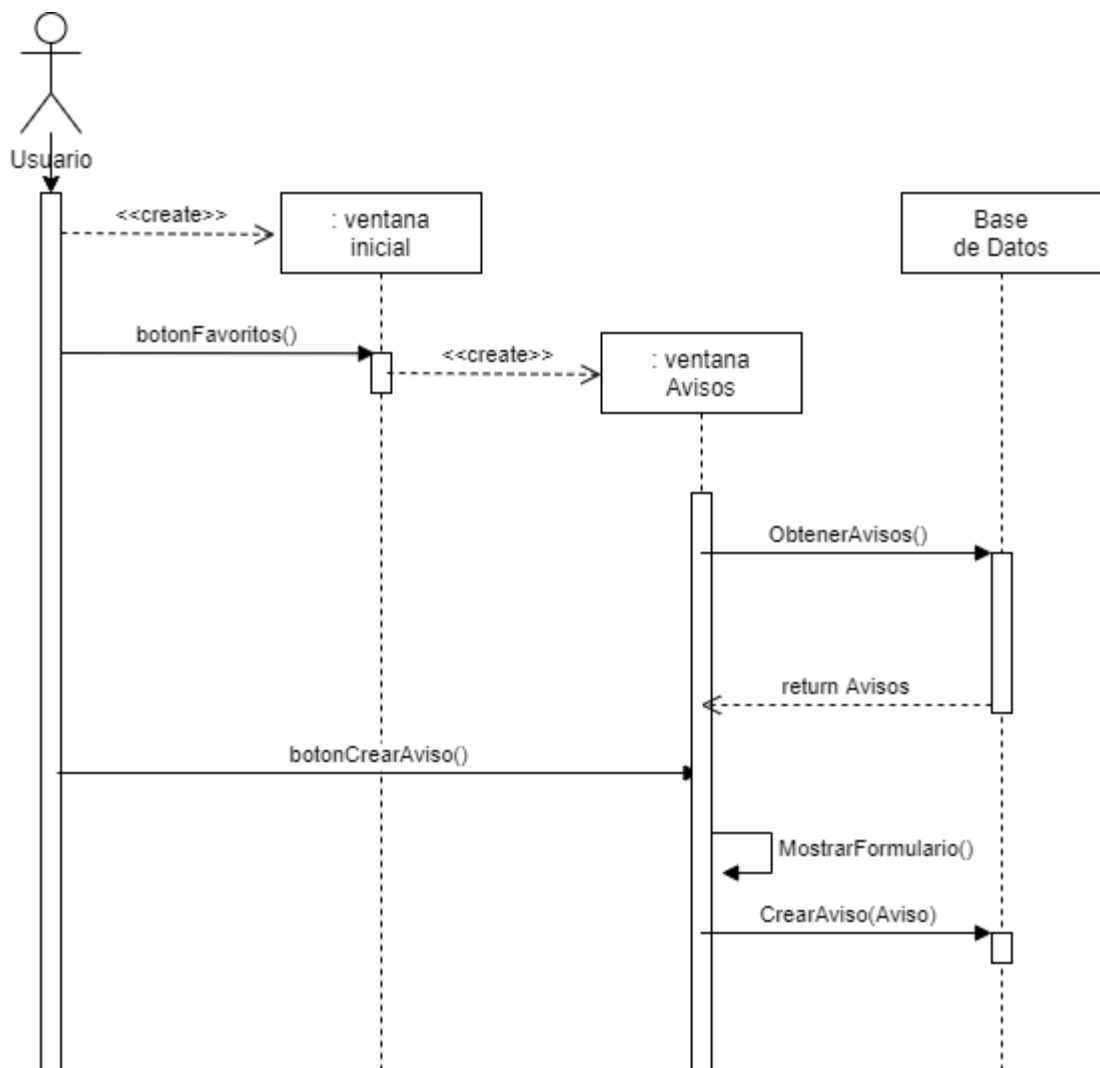


Figura 3.13 Diagrama de secuencia 11: Crear avisos

Número de caso de uso	CU12
Caso de Uso	Visualizar Perfil
Actores	Usuario, Administrador
Resumen	El Actor visualiza sus datos personales en la ventana visualizarDatos.
Precondiciones	La base de datos y el servidor se encuentra activo. El Actor ha iniciado sesión en la aplicación.
Flujo de eventos	1. El Actor selecciona el botón Mi perfil. 3. El sistema consulta la base de datos obteniendo los datos del usuario. 3. El sistema muestra los datos del usuario.
Postcondiciones	Los datos personales del Actor se muestran correctamente.
Extensiones (o Flujo alternativo)	

Tabla 3.35 Caso de uso CU12

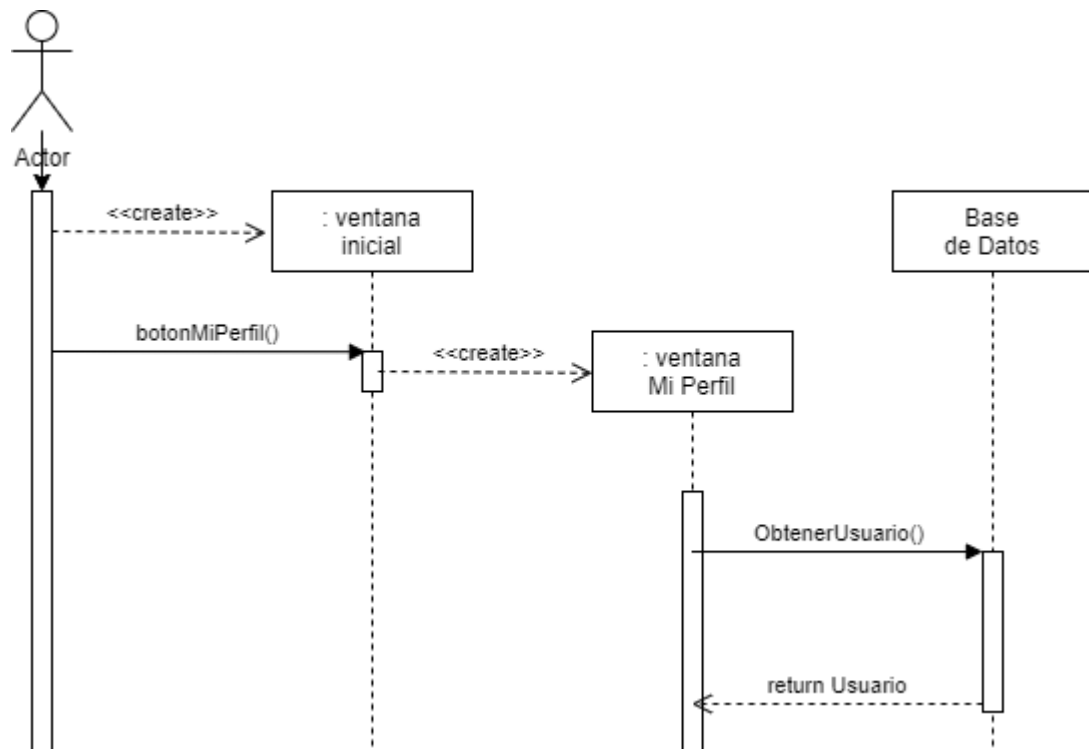


Figura 3.14 Diagrama de secuencia 12: Visualizar Perfil

Número de caso de uso	CU13
Caso de Uso	Modificar Perfil
Actores	Usuario, Administrador
Resumen	El Actor modifica sus datos personales o contraseña
Precondiciones	La base de datos y el servidor se encuentra activo. El usuario ha iniciado sesión en la aplicación.
Flujo de eventos	<ol style="list-style-type: none">1. El Actor selecciona el botón Mi perfil.2. El sistema consulta la base de datos obteniendo los datos del usuario.3. El sistema muestra los datos del usuario.4. El Actor modifica sus los datos5. El sistema modifica sus datos en la base de datos6. El sistema crea un aviso para el usuario.
Postcondiciones	Los datos personales del usuario se modifican correctamente.
Extensiones (o Flujo alternativo)	

Tabla 3.36 Caso de uso CU13

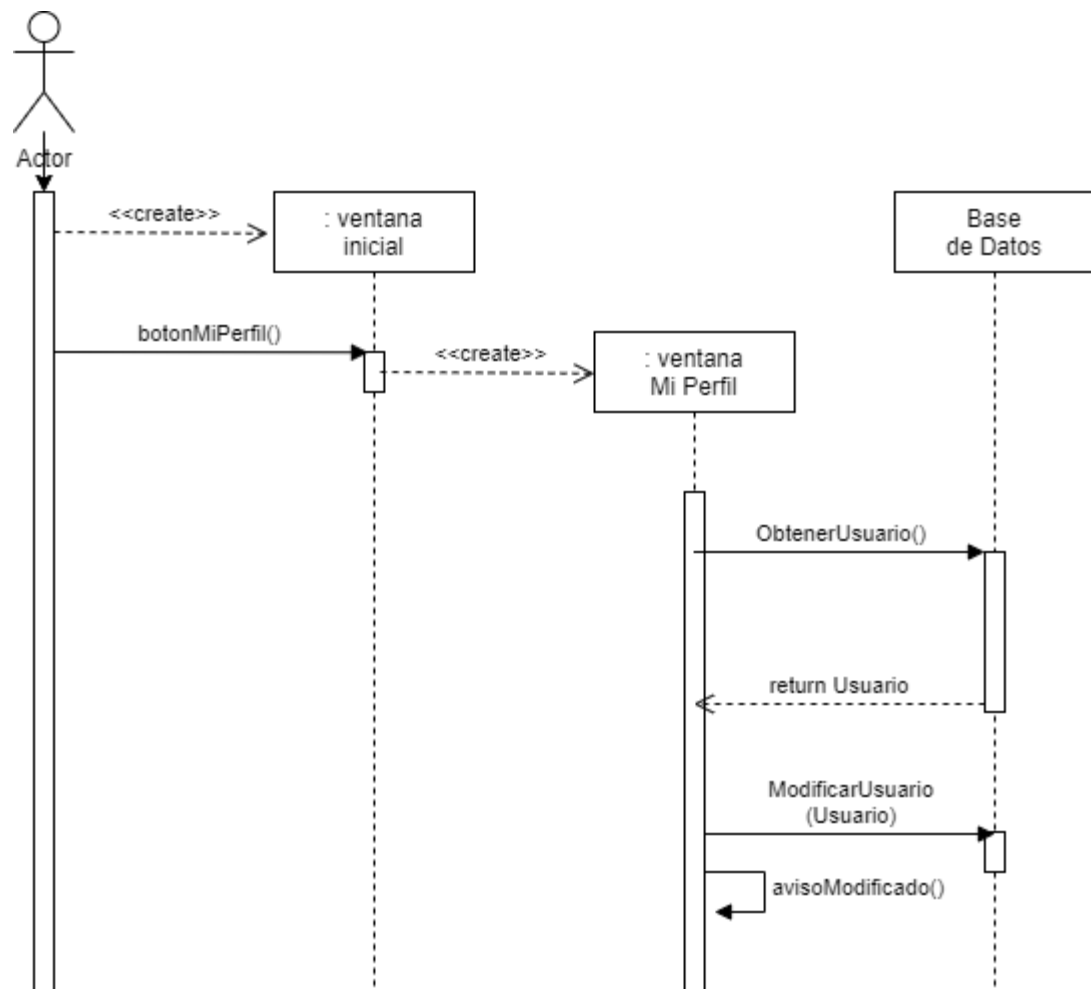


Figura 3.15 Diagrama de secuencia 13: Modificar Perfil

Número de caso de uso	CU14
Caso de Uso	Ver usuarios
Actores	Administrador
Resumen	El administrador quiere ver los usuarios junto con sus datos y su última conexión.
Precondiciones	La base de datos y el servidor se encuentra activo. El administrador ha iniciado sesión en la aplicación.
Flujo de eventos	1. El administrador selecciona ver usuarios. 2. El sistema consulta la base de datos obteniendo los usuarios registrados. 3. El sistema muestra en los usuarios en una tabla con sus datos y ultima conexión.
Postcondiciones	Los datos de los usuarios se muestran correctamente.
Extensiones (o Flujo alternativo)	

Tabla 3.37 Caso de uso CU14

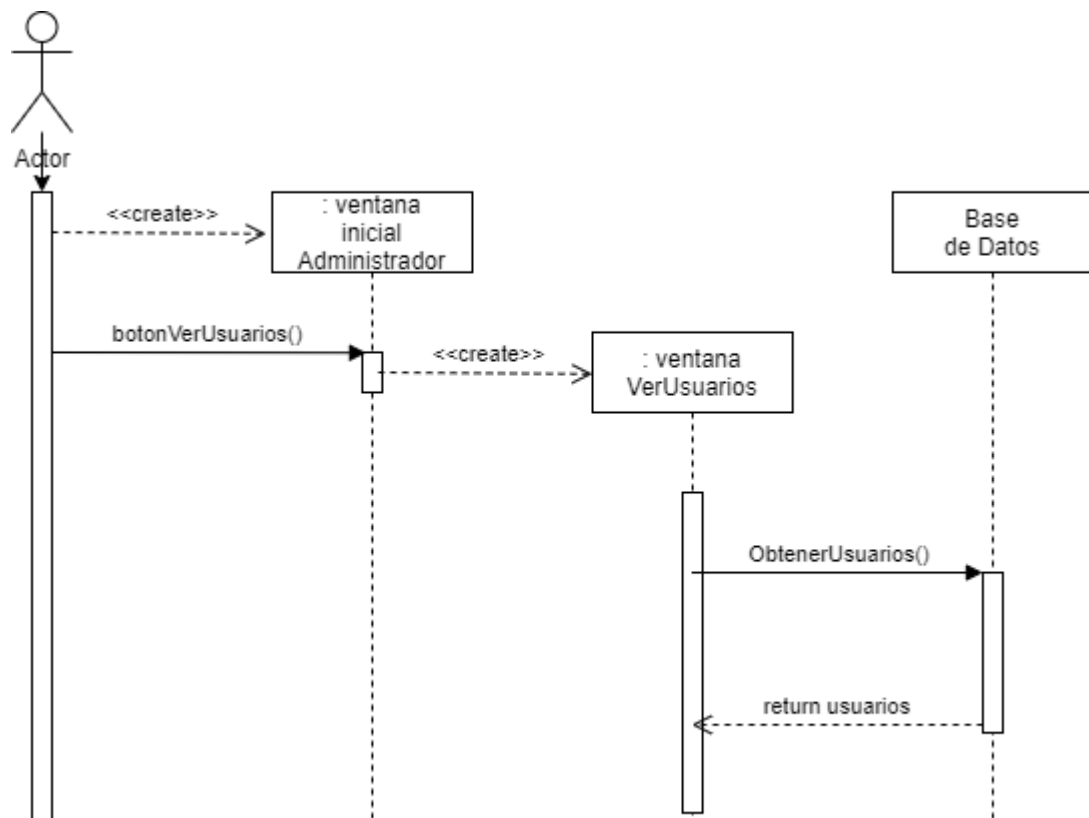


Figura 3.16 Diagrama de secuencia 14: Ver Usuarios

Número de caso de uso	CU15
Caso de Uso	Eliminar usuario
Actores	Administrador
Resumen	El administrador quiere ver los usuarios junto con sus datos y su última conexión.
Precondiciones	La base de datos y el servidor se encuentra activo. El administrador ha iniciado sesión en la aplicación.
Flujo de eventos	<ol style="list-style-type: none">1. El administrador selecciona ver usuarios.2. El sistema consulta la base de datos obteniendo los usuarios registrados.3. El sistema muestra en los usuarios en una tabla con sus datos y ultima conexión.4. El administrador presiona el botón eliminar usuario.5. El sistema envía un mensaje a la base de datos para que se elimine el usuario6. El sistema recarga los usuarios mostrados en la página.
Postcondiciones	El usuario ha sido eliminado de la base de datos.
Extensiones (o Flujo alternativo)	

Tabla 3.38 Caso de uso CU15

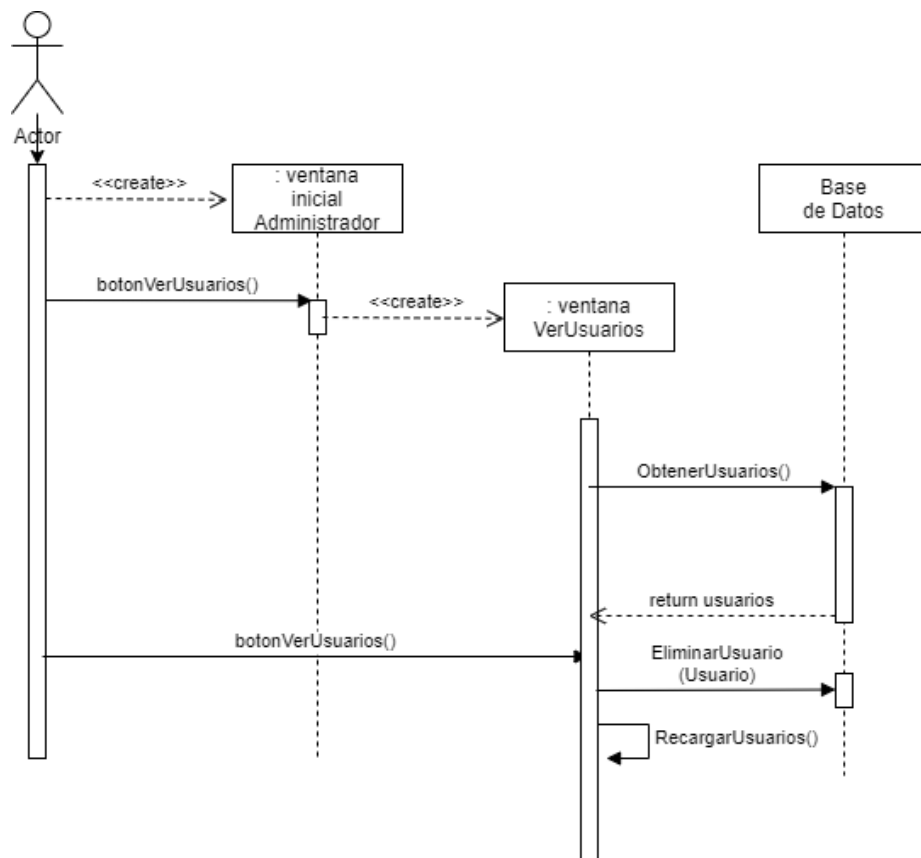


Figura 3.17 Diagrama de secuencia 15: Eliminar Usuario

Número de caso de uso	CU16
Caso de Uso	Web Scraper
Actores	Administrador
Resumen	El administrador activa el web Scraper para recolectar datos y estos transfórmalos en instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales para la página web.
Precondiciones	La base de datos y el servidor se encuentra activo. El administrador ha iniciado sesión en la aplicación.
Flujo de eventos	<ol style="list-style-type: none"> 1. El administrador abre la aplicación web Scraper. 2. El sistema carga el web Scraper. 3. El administrador presiona el botón Activar 4. El web Scraper comprueba si hay conexión con las páginas web donde este sacara el contenido. 5. El web Scraper obtiene el HTML de la página en cuestión. 6. El web Scraper procesa el HTML guardando el contenido necesario. 7. El web Scraper crea un contenido añadiéndolo a la base de datos.
Postcondiciones	El contenido recolectado se ha introducido en la base de datos.
Extensiones (o Flujo alternativo)	3a. No hay conexión con la página web <ol style="list-style-type: none"> 1. El sistema notifica al administrador de que no hay conexión.

Tabla 3.39 Caso de uso CU16

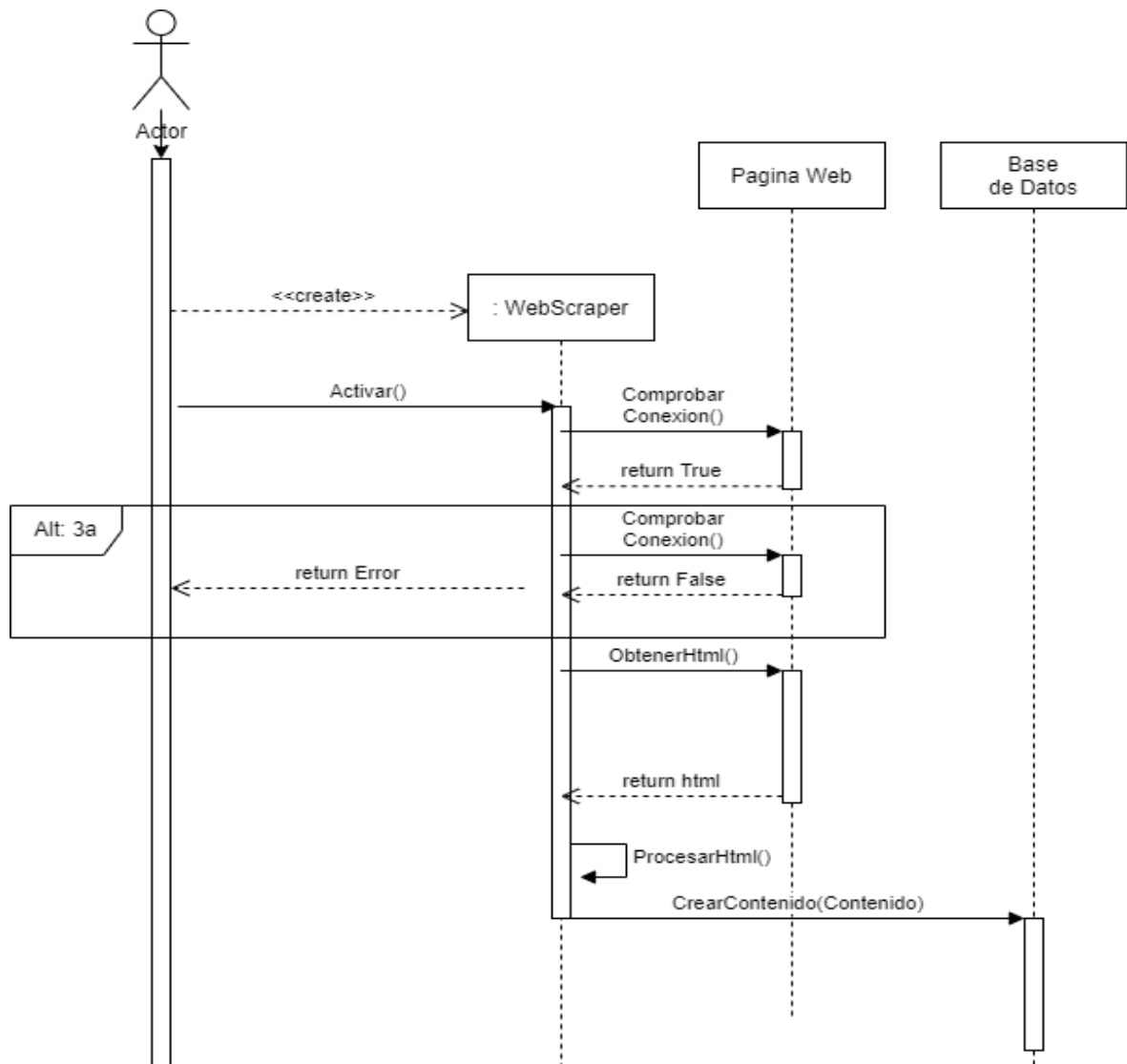


Figura 3.18 Diagrama de secuencia 16: Web Scraper

Modelo de dominio:

El modelo de dominio nos permite realizar un análisis de la aplicación el cual podemos visualizar en el diagrama representado en la figura 3.19, dentro de este diagrama podemos observar los elementos más representativos de la aplicación, sus atributos y relaciones.

Usuario: El usuario representa a la persona física la cual quiere interactuar con la aplicación ya sea visualizando el contenido (instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales, actividades, eventos), creando avisos para otros usuarios, realizando comentarios o añadiendo el contenido a sus favoritos.

Administrador: El administrador representa a la persona encargada de gestionar el sistema pudiendo eliminar usuarios, eliminar el contenido y activar el web Scraper para que este busque y cree nuevo contenido para la aplicación.

Contenido: El contenido representa a cualquier Noticia, Evento, Actividad, Parque, Instalación deportiva, elemento cultural, hotel, ruta o Parque natural que la aplicación muestre en la interfaz.

Comentarios: representa a los comentarios que los usuarios han realizado sobre algún contenido que la aplicación tenga.

Avisos: Elemento que representa los avisos que los usuarios han creado sobre alguna incidencia que haya ocurrido en la ciudad.

Favoritos: Representa el contenido que los usuarios han marcado como favoritos.

Web Scraper: Representa al programa se encarga de obtener los datos necesarios para crear nuevo contenido para la aplicación web.

Base de Datos: Elemento que permite el almacenamiento de los datos pertenecientes a la aplicación de forma persistente.

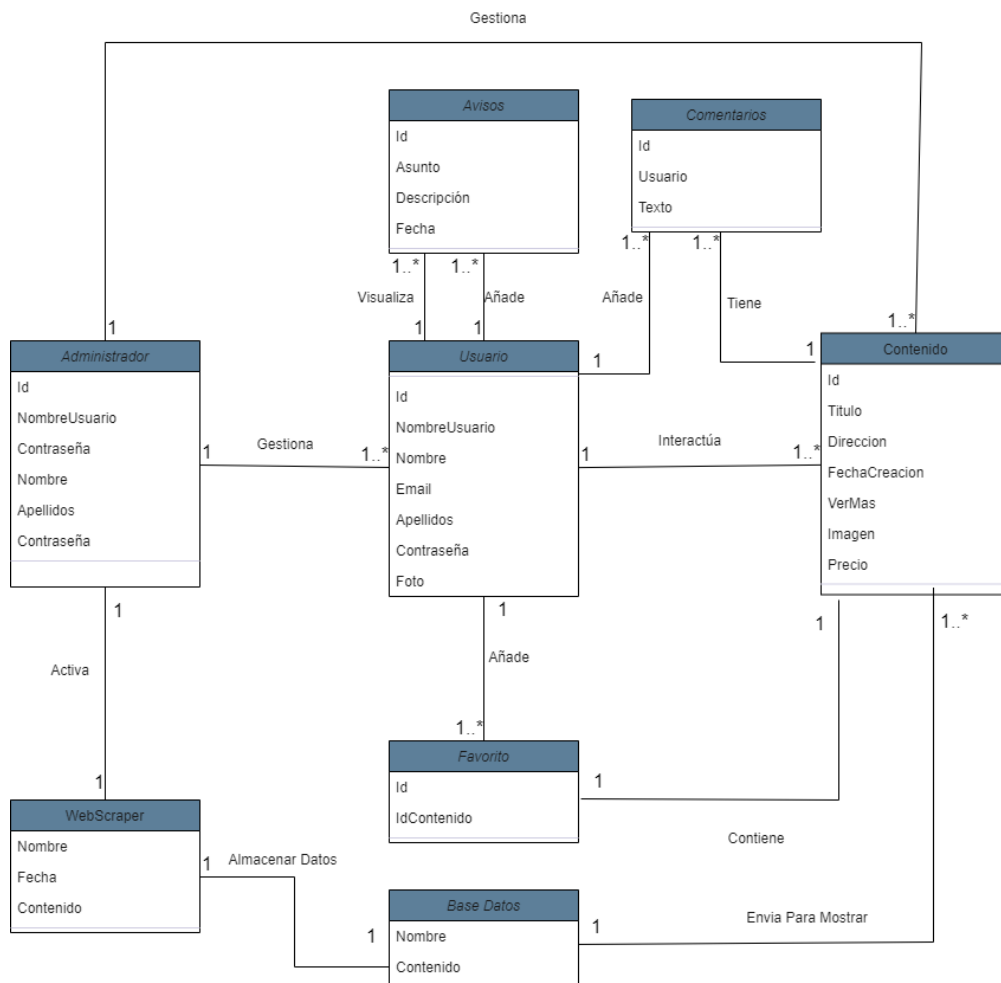


Figura 3.19 Diagrama de dominio

3.2.2 Diseño

Fase de diseño

3.2.2.1 Diseño de sistema

La arquitectura del sistema se puede ver en la figura 3.20 en la que se puede observar el funcionamiento del sistema creado mediante este proyecto.

Se puede observar como el navegador web y el servidor payara interactúan mediante mensajes de petición respuesta, en este servidor es en donde la aplicación web desarrollada mediante java se encuentra alojada.

También se puede observar que el servidor y la base de datos se envían mensajes de petición respuesta al igual que el servidor con el navegador.

Por Ultimo, se pueden observar las aplicaciones por un lado la aplicación web Scraper la cual envía el contenido generado a la base de datos y por otro el entorno de ejecución de la aplicación web el cual le envía los datos al servidor payara para que este la despliegue.

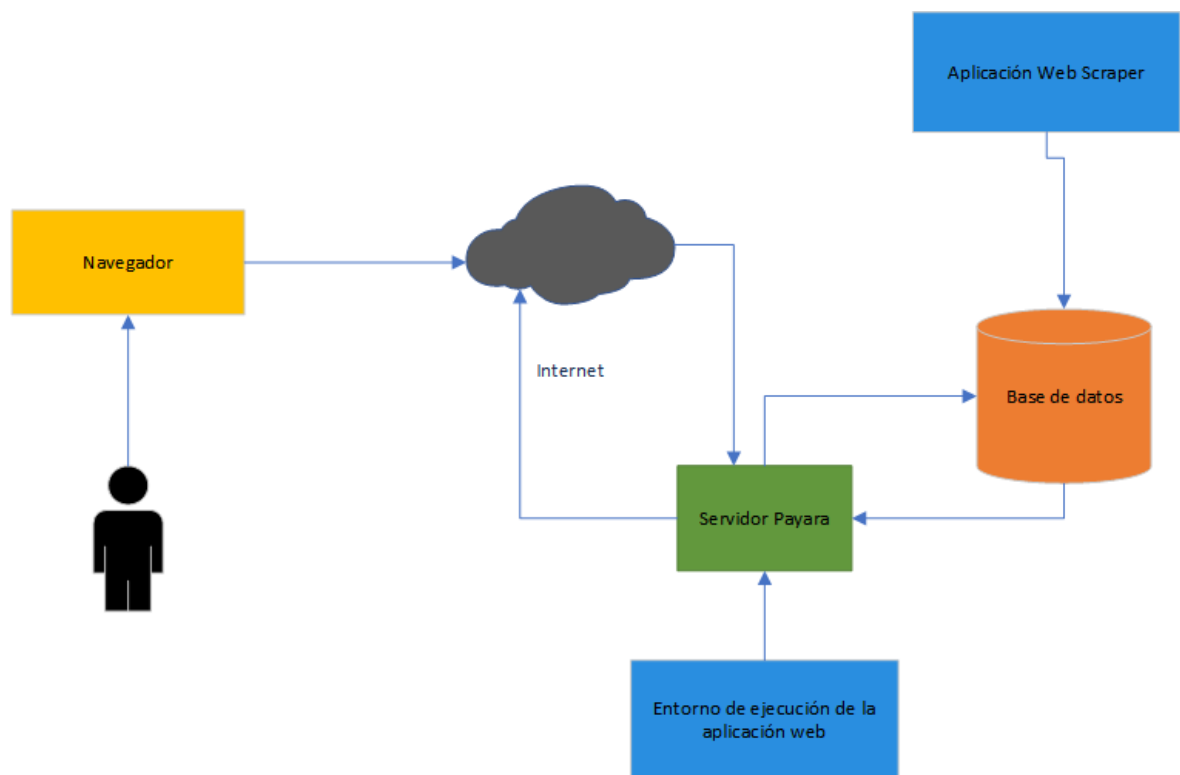


Figura 3.20 Diagrama de arquitectura

3.2.2.2 Diseño detallado.

La arquitectura del sistema está formada por tres elementos principales: La aplicación perteneciente al web Scraper mostrada con la imagen Jsoup, La base de datos y el servidor de aplicaciones payara donde se carga la aplicación desarrollada mediante java.

Web Scraper: Aplicación java desarrollada a parte de la aplicación web la cual permite al administrador activar el web Scraper creado con la tecnología Jsoup Introduciendo las instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales en la base de datos mediante sentencias SQL, además de poder visualizar lo que se ha introducido en la base de datos gracias a una interfaz gráfica creada con la herramienta Java Swing.

Base de datos MySQL: Base de datos donde se almacenan los datos que el web Scraper ha extraído de las páginas web, estos datos almacenados serán posteriormente usados por la aplicación web para visualizar el contenido que este representa, además de contener también información correspondiente a la aplicación web como puede ser Usuarios, Avisos, Comentarios.

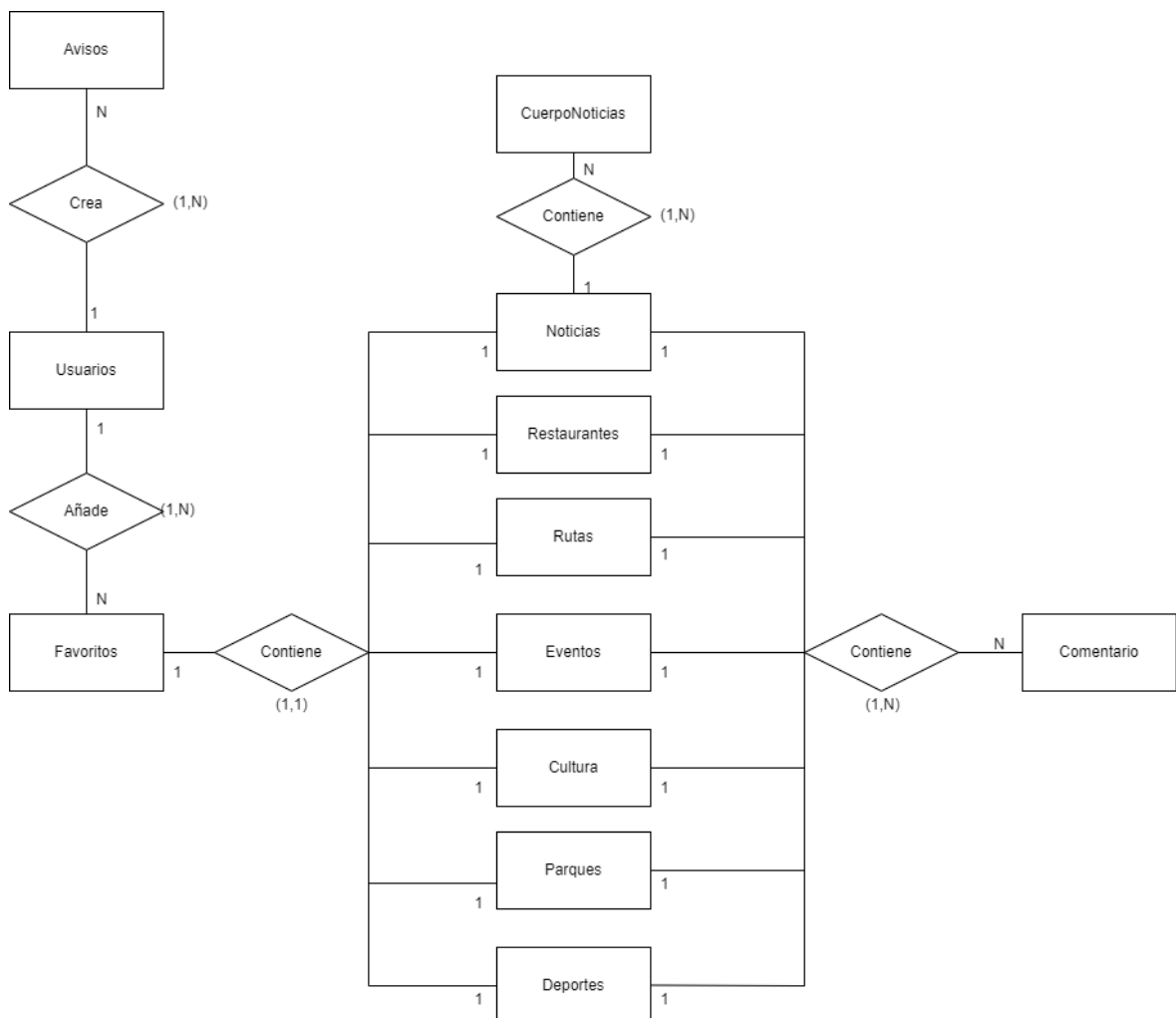


Figura 3.21 Diagrama Entidad-Relación

Servidor de aplicaciones Payara: Payara es el servidor donde se carga la aplicación web desarrollada.

La aplicación web ha sido desarrollada utilizando el patrón de diseño MVC (Modelo-Vista-Controlador). A través de este diseño se ha podido separar los distintos aspectos de la aplicación en partes diferenciadas las cuales serían el modelo, la vista, controlador.

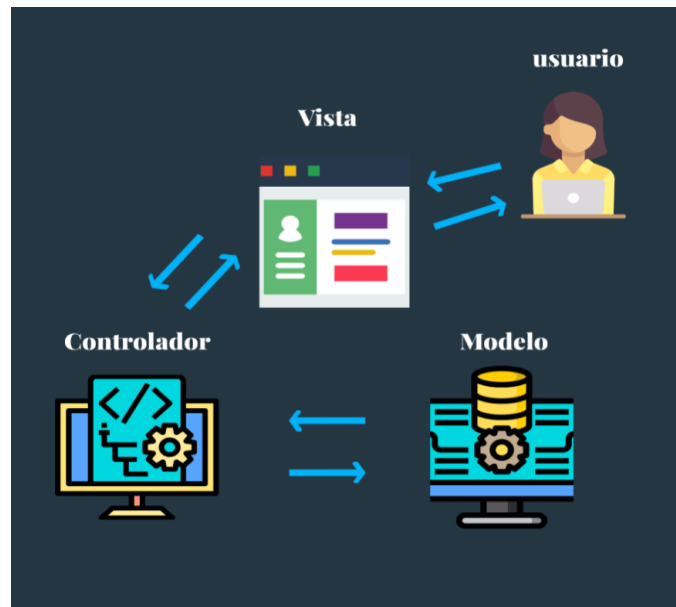


Figura 3.22 Modelo Vista Controlador

Modelo: Representa los datos que la aplicación va a manejar, gestionando los accesos a los datos de la aplicación, las consultas, las actualizaciones y la persistencia en la base de datos. En esta aplicación se ha realizado una clase entidad para cada tabla de la base de datos, creando después los beans de sesión para las clases entidad. Estos beans de sesión son las clases encargadas de la conexión con la base de datos y por lo tanto de la persistencia. Dentro de estos beans se pueden realizar consultas JPQL para realizar consultas en la base de datos.

Vista: Representa la capa que el usuario ve y con la cual interactúa, es decir la interfaz de la aplicación. La interfaz ha sido desarrollada en HTML con el framework JavaServer Faces y primefaces. Cada uno de los HTML creados pertenece a una interfaz que muestra el sistema. Esta capa se encarga de mostrar los datos recibidos desde la capa controladora y presentárselos al usuario. Para modelar los elementos HTML se ha utilizado CSS pudiendo así modificar el tamaño, posición y color.

Controlador: La parte del sistema que se encarga de gestionar el flujo de información entre la capa modelo y la capa vista.

3.2.3 Implementación

Hablar de las HERRAMIENTAS utilizadas durante el desarrollo, la ORGANIZACIÓN del proyecto y aquellas peculiaridades de la forma de implementación.

Lenguajes:

Java: Es un lenguaje de programación ampliamente utilizado, es un lenguaje multiplataforma y orientado a objetos (9). En el desarrollo de esta aplicación web se ha utilizado java por sus funciones, bibliotecas incorporadas, por su facilidad de uso y por su gran cantidad de tecnologías open source que podemos encontrar.

CSS: Es un lenguaje que se utiliza para realizar la presentación de los documentos HTML, es decir, es el lenguaje que se encarga de ordenar, dar forma y color a los elementos del HTML (10). En esta aplicación ha sido utilizado para editar el estilo de las diferentes páginas web, pudiendo así aplicar móvil First mediante este lenguaje.

HTML: Lenguaje de marcado de hipertexto utilizado para la elaboración de las páginas web (11). En este proyecto se ha utilizado para la representación de datos y diseño de la interfaz. Se ha escogido este lenguaje por su familiaridad.

SQL: Es un lenguaje diseñado para ser utilizado para la creación y administración de bases de datos para el procesamiento de los datos (12). En este proyecto se ha utilizado para crear la base de datos y para realizar consultas desde la aplicación a esta.

Herramientas:

Interfaz runnable: es una interfaz la cual nos permite crear instancias de clases ejecutadas por un hilo. En este proyecto se ha utilizado a la hora de realizar el temporizador del web Scraper creando un hilo que duerme una hora para después recompilar noticias y un hilo que duerme un día para después recompilar el contenido.

JSF: La tecnología Java Server Faces es un framework utilizado para la interfaz de los componentes de usuario del lado del servidor para las aplicaciones web en las que se utiliza la tecnología java (13). En este proyecto se ha utilizado para realizar las distintas interfaces.

Primefaces: Es una librería que no necesita configuraciones la cual nos proporciona una serie de utilidades y componentes para usar con java Server Faces (14).

Jsoup: Es una librería de java que sirve para extraer y analizar HTML de distintas páginas web pudiendo extraer datos de esta (15). En este proyecto se ha utilizado para la recolección de información para la página web.

Payara server: Servidor de aplicaciones de código abierto para el desarrollo de aplicaciones java (16). Es el servidor de aplicaciones utilizado en este proyecto para desplegar la aplicación web.

JPA (Java Persistence API): Especificación la cual indica como se debe realizar la persistencia de los objetos en los programas desarrollados con java(17).

EJB (Enterprise JavaBeans): Especificación creada por Sun Microsystems para desarrollar aplicaciones seguras, robustas y escalables. EJB es un elemento de software del lado del servidor que nos permite resumir la lógica de la aplicación(18). En este proyecto se ha utilizado para poder conectar la aplicación con la base de datos.

Junit: Framework que permite pruebas en el software.

JavaMail: proporciona una plataforma y marco independientes del protocolo para crear correo y mensajería aplicaciones.

Otras herramientas:

Heidi SQL: es un software libre, el cual te permite ver, crear y editar datos que ejecutan uno de los sistemas de base de datos MariaDB, MySQL, Microsoft SQL, PostgreSQL y SQLite(19). En este proyecto se ha utilizado para la creación de la base de datos MySQL.

GitHub: Plataforma desarrollada para el hospedaje de código para el control de versiones donde distintos desarrolladores pueden almacenar, compartir y colaborar permitiéndoles trabajar juntos en proyectos desde donde sea.

NetBeans: Entorno de desarrollo libre principalmente este hecho para java. Este entorno ha permitido el desarrollo de la aplicación.

Drawio: Herramienta que ha permitido crear y diseñar los diagramas de este proyecto.

Microsoft Project: Software comercializado por Microsoft utilizado para la administración de proyectos. Esta herramienta ha sido utilizada para realizar el diagrama de Gantt de la planificación de este proyecto.

Drive: Servicio de alojamiento de archivos desarrollado por Google. Se ha utilizado para el almacenamiento de la documentación del proyecto.

Microsoft Visio: Software comercializado por Microsoft utilizado para la creación de diagramas o gráficos. Esta herramienta ha sido utilizada para realizar algunos de los diagramas de este proyecto.

3.2.4 Pruebas

Las pruebas unitarias es una forma mediante la cual podemos testear los diferentes métodos que se encuentran en las distintas clases pudiendo con ellos garantizar el correcto funcionamiento de estos de manera aislada y al integrarlo en el sistema.

Para realizar este tipo de pruebas se ha utilizado Junit, Herramienta que nos permite probar el software del sistema detectando fallos en el mismo pudiendo corregirlos para que este funcione correctamente.

Debido a que la aplicación web es una aplicación que utiliza EJB conectándose directamente con la base de datos a través del servidor de aplicaciones payara es complicado establecer los casos de prueba para una correcta ejecución. Por ello las pruebas que se han realizado son las pruebas de las funciones que no tienen conexión con la base de datos.

Como se puede ver en la figura 3.23 se han realizado los test que han sido los cuales de los métodos que no necesitaban conexión a la base de datos.

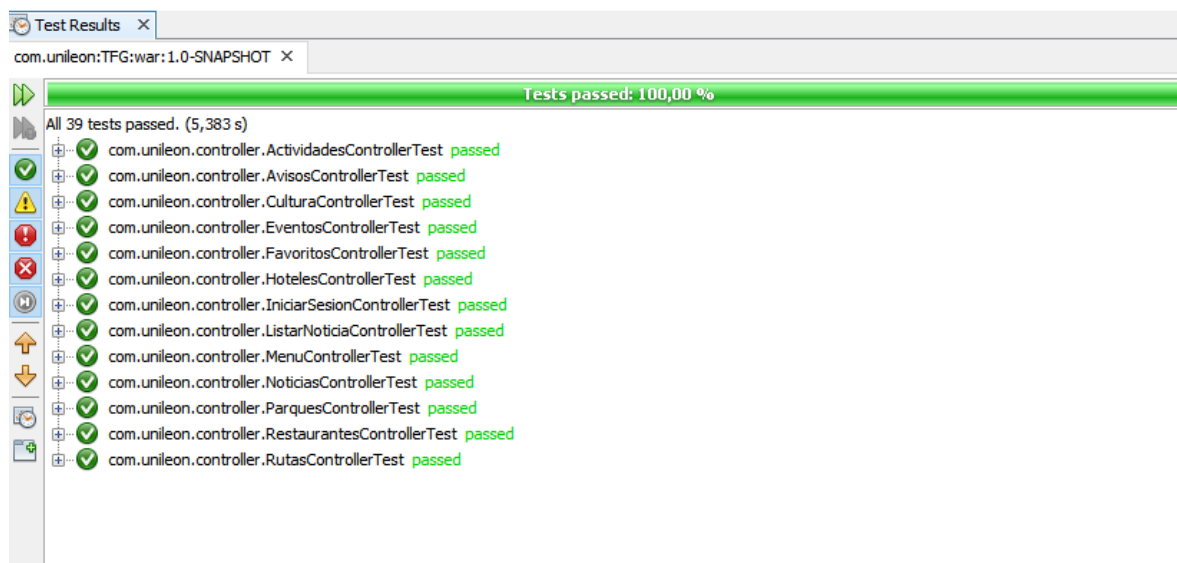


Figura 3.23 Test unitarios

3.3 El producto del desarrollo

3.3.1 Web Scraper

En la Figura 3.24 Podemos ver la interfaz de la aplicación java perteneciente a la parte del web Scraper.

Podemos ver en la interfaz dos botones un botón es el encargado de iniciar el Web Scraper y el otro de detenerlo. En la Figura 3.24. el web Scraper ya se ha activado y podemos ver la información que este ha encontrado e introducido en la base de datos.

Una vez activado el web Scraper tiene un temporizador en el cual cada hora busca noticias nuevas y cada día busca contenido (Actividades, Eventos, Cultura, Instalaciones Deportivas, Eventos, Hoteles, Parques, Rutas, Restaurantes) debido a que las noticias es un contenido que se actualiza más a menudo mientras que el resto no se actualiza con tanta frecuencia.

También podemos observar un filtro para poder seleccionar Actividades, Eventos, Cultura, Instalaciones Deportivas, Eventos, Hoteles, Parques, Rutas, Restaurantes o noticias y realizar una búsqueda rápida de este contenido.

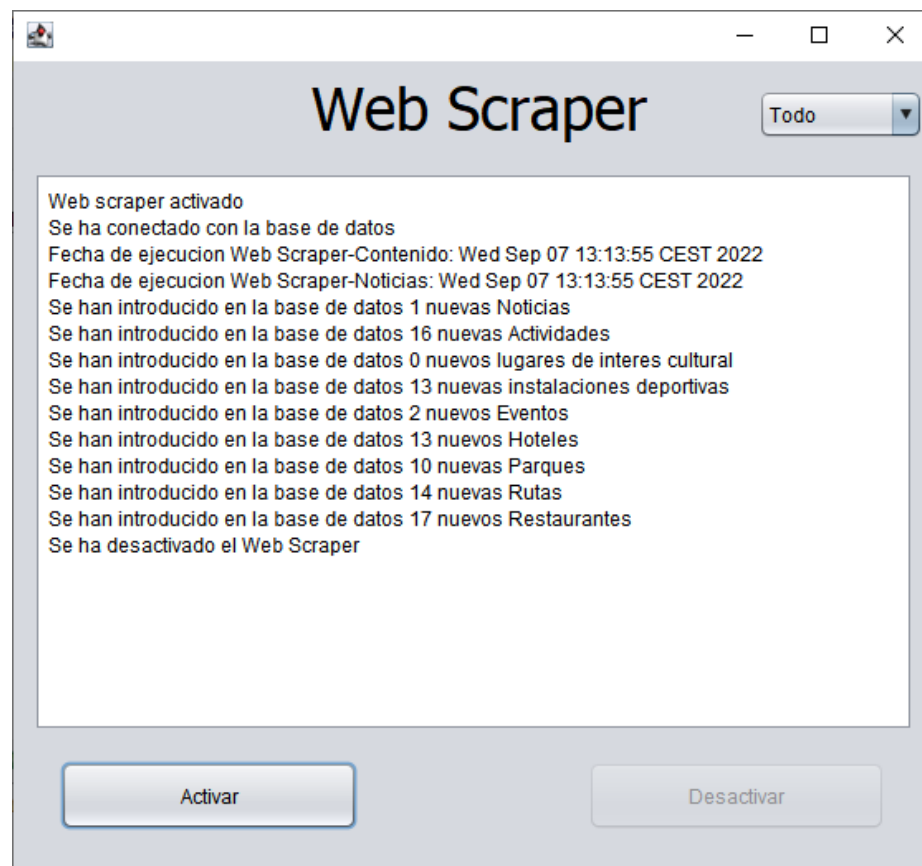


Figura 3.24 Aplicación Web Scraper

El web Scraper recolecta información sobre las noticias en la página LeoNoticias (20) y para el resto del contenido recompilado podemos encontrarlo en Minube(21) , LeonOcio (22), Terranostrum (23) y Yumping (24).

Dejando a un lado la interfaz el web Scraper está formado por dos hilos los cuales se activan al presionar el botón Activar los cuales se encuentran en bucle infinito durmiendo en caso de las noticias una hora y en caso de el contenido un día como se puede observar en la Figura 3.25.

```
Runnable r = new Runnable() {
    @Override
    public void run() {
        while (true) {
            Date d = new Date();
            System.out.println(d.toString());
            JTextArea1.append("Fecha de ejecucion Web Scraper-Contenido: " + d.toString() + "\n");
            Yumping yump = new Yumping(oBD);
            LeonOcio leonO = new LeonOcio(oBD);
            Terranostrum Terra = new Terranostrum(oBD);
            Minube minube = new Minube(oBD);
            int resultado = leonO.getContadorCultura() + Terra.getContadorCultura();
            JTextArea1.append("Se han introducido en la base de datos " + yump.getContadorAct()
                + " nuevas Actividades\n");
            JTextArea1.append("Se han introducido en la base de datos " + resultado
                + " nuevos lugares de interes cultural\n");
            JTextArea1.append("Se han introducido en la base de datos " + leonO.getContadorDepor()
                + " nuevas instalaciones deportivas\n");
            JTextArea1.append("Se han introducido en la base de datos " + leonO.getContadorEvenros()
                + " nuevos Eventos\n");
            JTextArea1.append(
                "Se han introducido en la base de datos " + leonO.getContadorHotel() + " nuevos Hoteles\n");
            resultado = leonO.getContadorParques() + minube.getContadorParq();
            JTextArea1.append("Se han introducido en la base de datos " + resultado + " nuevas Parques\n");
            resultado = leonO.getContadorRutas() + Terra.getContadorRutas();
            JTextArea1.append("Se han introducido en la base de datos " + resultado + " nuevas Rutas\n");
            JTextArea1.append("Se han introducido en la base de datos " + leonO.getContadorrestau()
                + " nuevos Restaurantes\n");
            try {
                Thread.sleep((long) 8.64e+7);
            } catch (InterruptedException ex) {
                Logger.getLogger(NewJFrame.class.getName()).log(Level.SEVERE, null, ex);
            }
        }
    }
};
hilo = new Thread(runnable);
hilo2 = new Thread(r);
```

Figura 3.25 Temporizador contenido

El web Scraper lo primero que hace es comprobar que se tiene conexión con la página de la se va a sacar la información mediante el método que se muestra en la figura 3.26.

```
private int getStatusConnectionCode(String url) {
    Connection.Response response = null;
    try {
        response = Jsoup.connect(url).userAgent("Mozilla/5.0").timeout(100000).ignoreHttpErrors(true).execute();
    } catch (IOException ex) {
        System.out.println("Excepción al obtener el Status Code: " + ex.getMessage());
    }
    return response.statusCode();
}
```

Figura 3.26 Métodos comprobar conexión

Una vez que se ha comprobado que hay conexión se obtiene el HTML de dicha página con el método de la figura 3.27.

```
private Document getHtmlDocument(String url) {
    Document doc = null;
    try {
        doc = Jsoup.connect(url).userAgent("Mozilla/5.0").timeout(100000).get();
    } catch (IOException ex) {
        System.out.println("Excepcii: %n al obtener el HTML de la página" + ex.getMessage());
    }
    return doc;
}
```

Figura 3.27 Método obtener HTML

Ese HTML obtenido se procesa buscando la información que se necesita para después esta introducirla en la base de datos como se puede ver en la figura 3.28.

```
if (getStatusConnectionCode(Urls.get(i)) == 200) {
    Document doc = getHtmlDocument(Urls.get(i));

    if (doc.body().className().equals("leonoticias")) {
        try {
            imagenE = doc.getElementsByClass("voc-horizontal voc-img-icon-link").first().getElementsByTag("img").first();
            imagen = imagenE.attr("abs:src");
            if (imagen.isEmpty()) {
                imagenE = doc.getElementsByClass("voc-detail__cmp ").first().getElementsByTag("img").first();
                imagen = imagenE.attr("abs:data-original");
            }
        } catch (Exception e) {
            System.out.println("No imagen");
        }
        Titulo = doc.getElementsByClass("voc-title").first().text();

        Elements Cuerpos = doc.getElementsByClass("voc-paragraph");
        ParteCuerpo = new ArrayList<String>();
        for (Element Cuerpo : Cuerpos) {
            ParteCuerpo.add(Cuerpo.text());
        }
        Fecha = doc.getElementsByClass("voc-author-info").first().getElementsByTag("time").attr("datetime").substring(0, 10).replace("-", "/");
        creaNoticia(Titulo, ParteCuerpo, Fecha, imagen);
    }
} else {
    System.out.println("El Status Code no es OK es: " + getStatusConnectionCode(Urls.get(i)));
}
```

Figura 3.28 Web Scraper Noticia

3.3.2 Aplicación Web

Sesión Usuario:

En la Figura 3.29 se puede ver la página inicial del usuario cuando este realiza el inicio de sesión, donde podremos observar distintas noticias, Actividades, eventos...

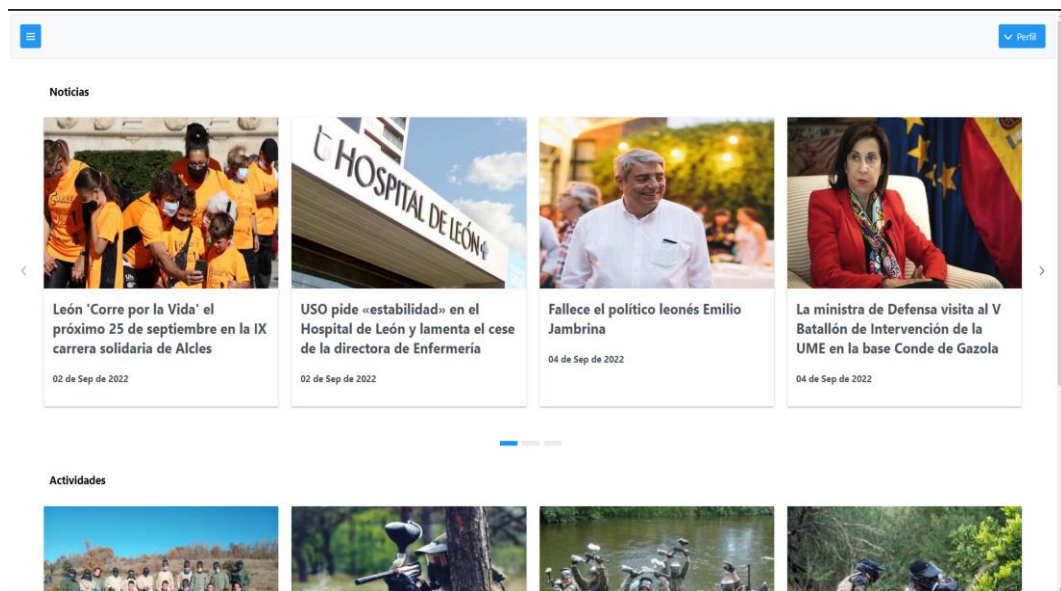


Figura 3.29 Pagina inicial

El usuario puede acceder al menú donde podrá encontrar el contenido que la página web contiene pudiendo presionar sobre el accediendo al mismo como se puede observar en la figura 3.30.

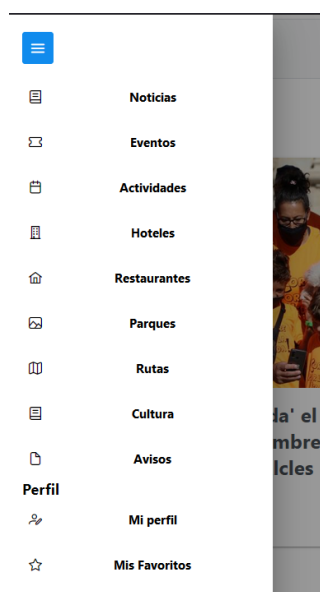


Figura 3.30 Menú Aplicación

La página web fue desarrollada utilizando la técnica de móvil first en la cual primero se realiza la interfaz perteneciente a móvil y después se realiza la de ordenador, en las figuras 3.31 y 3.32 podemos ver las interfaces pertenecientes a la página de la web.

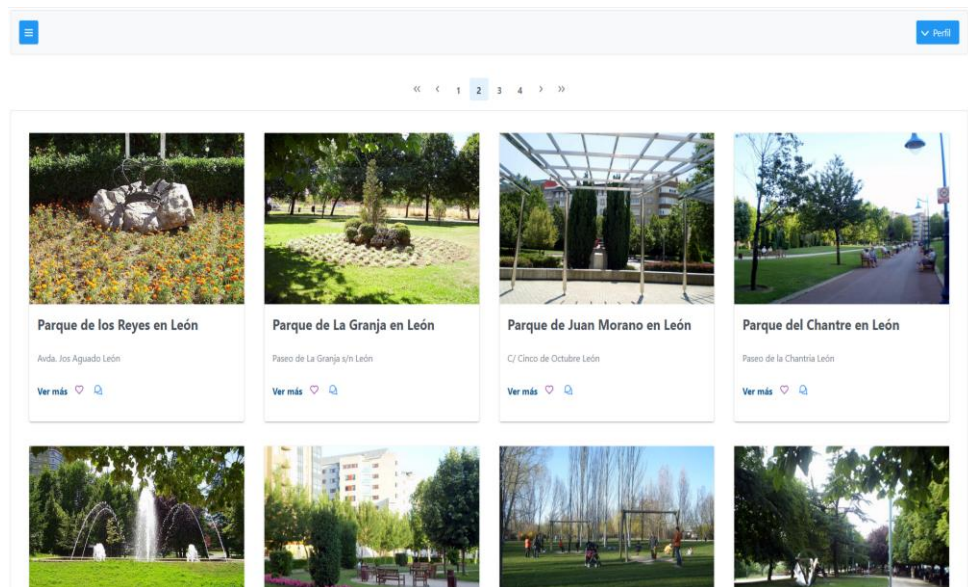


Figura 3.31 Interfaz parques ordenador



Figura 3.32 Interfaz parques móvil

Cada noticia recolectada tiene su propia página en la cual puedes ver la información de esta como se muestra en la Figura 3.33.



Figura 3.33 Pagina ver Información noticia

En cada contenido que nos muestra la web podemos ver en la parte inferior un corazón el cual presionaría el usuario si este lo quiere añadir a favoritos, Un botón perteneciente a los comentarios pudiendo ver los comentarios que los usuarios han realizado en este. Además, podemos ver un texto (“Ver más”) en el cual si presionamos nos llevara hasta la página web de donde este contenido ha sido extraído.



Figura 3.34 Tarjeta hotel

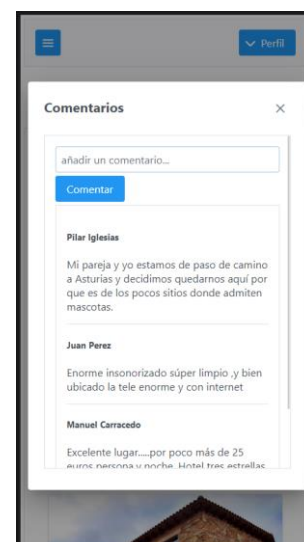


Figura 3.35 Interfaz Comentarios

También dentro de la página web podemos crear avisos con los cuales avisar a los usuarios sobre cualquier incidencia como podemos observar en la figura 3.36

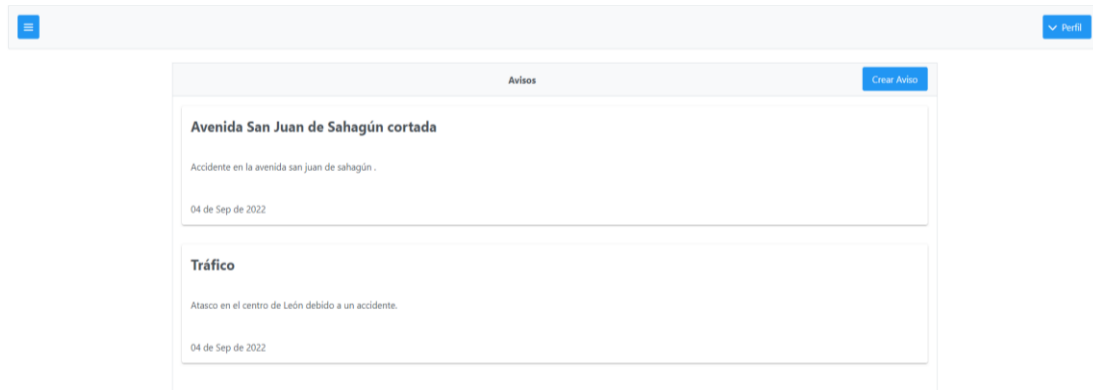


Figura 3.36 Interfaz avisos

En la Figura 3.37 podemos ver el contenido añadido a favoritos por el usuario pudiendo acceder a este cuando él quiere.

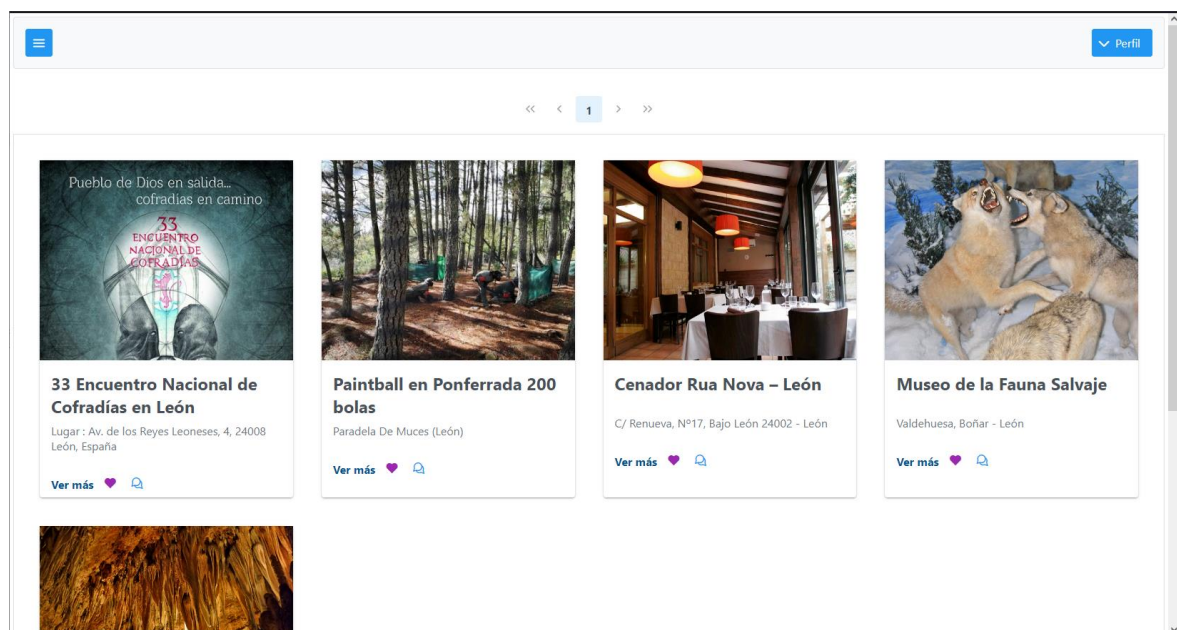


Figura 3.37 Interfaz favoritos

Los HTML tienen una clase java asociada a esa vista en la cual se procesa toda la información que esa vista necesite, Esta clase controladora se conecta con la base de datos a través de las clases EJB las cuales utilizan el patrón fachada a y que son generada gracias a las clases que se encuentran el paquete modelo.

```
@EJB
private UsuarioFacadeLocal usuarioEJB;
```

Figura 3.41 Llamada a EJB

Las clases modelo son la representación en Java de las tablas de la base de datos como se puede observar en la figura 3.41.

```
@Entity
@Table(name = "avisos")
public class Avisos implements Serializable {

    @Id
    @GeneratedValue(strategy = GenerationType.IDENTITY)
    private int idAviso;
    @Column(name = "Asunto")
    private String Asunto;
    @Column(name = "Descripcion")
    private String Descripcion;
    @Column(name = "Fecha")
    @Temporal(TemporalType.DATE)
    private Date Fecha;
    @JoinColumn(name = "idUsuario")
    @ManyToOne
    private Usuario Usuario;

    public int getIdAviso() {
        return idAviso;
    }

    public void setIdAviso(int idAviso) {
        this.idAviso = idAviso;
    }

    public String getAsunto() {
        return Asunto;
    }

    public void setAsunto(String Asunto) {
```

Figura 3.42 Clase modelo Avisos

En estas clases EJB se pueden realizar consultas a la base de datos mediante JPQL obteniéndose los datos que se necesita como se puede ver en la Figura 3.43.

```
public Usuario consultarUsuario(Usuario us){
    Usuario usDevuelto = new Usuario();

    String consultaJPQL = "FROM Usuario u WHERE u.usuario=:param1 and u.password=:param2";
    Query query = em.createQuery(consultaJPQL);
    query.setParameter("param1", us.getUsuario());
    query.setParameter("param2", us.getPassword());
    List<Usuario> resultado = query.getResultList();
    if(!resultado.isEmpty())
        System.out.println("Usuario: "+resultado.get(0).getUsuario()+" Contraseña: "+resultado.get(0).getPassword());
    if(resultado.isEmpty()){
        usDevuelto=null;
    }else{
        usDevuelto=resultado.get(0);
    }

    FacesContext.getCurrentInstance().getExternalContext().getSessionMap().put("usuario", usDevuelto);
    return(usDevuelto);
}
```

Figura 3.43 Consulta JPQL

Dentro de la clase controladora podemos encontrar los métodos a los cuales la vista llama para cumplir con el objetivo que esta tiene, se puede ver en la siguiente figura un ejemplo de método de la clase controlador en este caso es la comprobación de si el usuario es administrador o no y si se encuentra en la base de datos.

```
public String verificarUsuario(){
    Usuario nuevo;
    nuevo = usuarioEJB.consultarUsuario(usuario);
    if(nuevo == null){
        FacesContext.getCurrentInstance().addMessage(null, new FacesMessage(FacesMessage.SEVERITY_ERROR, "Error", "Usuario o contraseña incorrecta"));
    }
    else{
        if(EsAdmin(nuevo)){
            if(nuevo.getFechaAcceso()==null){
                return "/Private/Admin/RegistrarAdmin.xhtml?faces-redirect=true";
            }else{
                return "/Private/Usuario/PrincipalUsuario.xhtml?faces-redirect=true";
            }
        }
        else{
            nuevo.setFechaAcceso(new Date());
            usuarioEJB.edit(nuevo);
            return "/Private/Usuario/PrincipalUsuario.xhtml?faces-redirect=true";
        }
    }
    return null;
}
```

Figura 3.44 Método Verificar Usuario

4 Evaluación

4.1 Proceso de evaluación

4.1.1 Forma de evaluación

Para la evaluación de la página web se han realizado test unitarios para aquellos métodos que no tengan conexión con la base de datos.

Para el resto de los métodos se han realizado pruebas en tiempo de ejecución para comprobar que las instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales que forman el contenido de la página web se generaba, editaba y eliminaba correctamente de la base de datos.

Se ha realizado la comprobación en tiempo de ejecución de las diferentes funcionalidades de la página web para eliminar los posibles errores que esta poseyera.

4.1.2 Casos de prueba

Los casos de prueba llevados a cabo en este proyecto para comprobar su correcto funcionamiento serían:

- Test CU01: Crear usuario
 - Test de acceso a pantalla registrarse
 - Test comprobar datos usuarios.
 - Test rellenar formulario y guardar datos en la base de datos
 - Test cancelar registro usuario
- Test CU02: iniciar sesión
 - Test comprobar usuario y contraseña incorrecta
 - Test iniciar sesión con usuario y contraseña correcta
 - Test iniciar sesión con usuario administrador
- Test CU03: Comprobar menú
 - Test para comprobar el correcto funcionamiento de los botones del menú
 - Test cerrar sesión
- Test CU04: Visualizar Noticias:
 - Test contenido noticias
- Test CU05: Eliminar noticias
 - Test eliminar noticias
- Test CU06: Visualizar cuerpo noticia
 - Test ver cuerpo noticia
- Test CU07: Visualizar eventos:
 - Test contenido eventos
- Test CU08: Eliminar eventos
 - Test eliminar eventos
- Test CU09: Realizar comentario eventos
 - Test realizar comentario eventos
 - Test eliminar comentario
 - Test modificar comentario

- Test CU10: Añadir a favoritos evento
 - Test añadir a favoritos
 - Test eliminar de favoritos
- Test CU11: Visualizar actividades:
 - Test contenido actividades
- Test CU12: Eliminar actividades
 - Test eliminar actividades
- Test CU13: Realizar comentario actividades
 - Test realizar comentario actividades
 - Test eliminar comentario
 - Test modificar comentario
- Test CU14: Añadir a favoritos actividades
 - Test añadir a favoritos
 - Test eliminar de favoritos
- Test CU15: Visualizar hoteles:
 - Test contenido hoteles
- Test CU16: Eliminar hoteles
 - Test eliminar hoteles
- Test CU17: Realizar comentario hoteles
 - Test realizar comentario hoteles
 - Test eliminar comentario
 - Test modificar comentario
- Test CU18: Añadir a favoritos hoteles
 - Test añadir a favoritos
 - Test eliminar de favoritos
- Test CU19: Visualizar Restaurantes:
 - Test contenido Restaurantes
- Test CU20: Eliminar Restaurantes
 - Test eliminar Restaurantes
- Test CU21: Realizar comentario Restaurantes
 - Test realizar comentario Restaurantes
 - Test eliminar comentario
 - Test modificar comentario
- Test CU22: Añadir a favoritos Restaurantes
 - Test añadir a favoritos
 - Test eliminar de favoritos
- Test CU23: Visualizar Parques:
 - Test contenido Parques
- Test CU24: Eliminar Parques
 - Test eliminar Parques
- Test CU25: Realizar comentario Parques
 - Test realizar comentario Parques
 - Test eliminar comentario
 - Test modificar comentario
- Test CU26: Añadir a favoritos Parques
 - Test añadir a favoritos
 - Test eliminar de favoritos
- Test CU27: Visualizar Rutas:

- Test contenido Rutas
- Test CU28: Eliminar Rutas
 - Test eliminar Rutas
- Test CU29: Realizar comentario Rutas
 - Test realizar comentario Rutas
 - Test eliminar comentario
 - Test modificar comentario
- Test CU30: Añadir a favoritos Rutas
 - Test añadir a favoritos
 - Test eliminar de favoritos
- Test CU31: Visualizar Cultura:
 - Test contenido Cultura
- Test CU32: Eliminar Cultura
 - Test eliminar Cultura
- Test CU33: Realizar comentario Cultura
 - Test realizar comentario Cultura
 - Test eliminar comentario
 - Test modificar comentario
- Test CU34: Añadir a favoritos Cultura
 - Test añadir a favoritos
 - Test eliminar de favoritos
- Test CU35: Test Avisos
 - Test crea aviso
 - Test visualizar avisos
 - Test borrar avisos
- Test CU36 Test mi perfil
 - Test comprobar datos mi perfil
- Test CU37: Test Tiempo de respuesta
 - Test comprobación tiempo de respuesta de la aplicación web
- Test CU36: Contenido web Scraper
 - Test comprobación contenido web Scraper
- Test CU37: Test web Scraper automatizado
 - Test comprobación web Scraper noticias cada hora
 - Test comprobación web Scraper contenido cada día
- Test CU38: Test creación de contenido por web Scraper
 - Test comprobación base de datos del contenido del web Scraper
- Test CU39: Test filtro de contenido actividades
- Test CU40: Test filtro de contenido eventos

4.2 Análisis de resultados

Después de realizar los casos de prueba descritos en el apartado anterior comprobando así el funcionamiento de la página web podemos concluir que la página web cumple con los requisitos establecidos.

Pudiendo visualizar el contenido: instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales guardado en la base de datos mediante el web Scraper en la aplicación web y pudiendo interactuar con el mismo.

El contenido mostrado en la página web se almacena correctamente en favoritos, además de que cada contenido tiene asociado unos comentarios los cuales le pueden crear, editar y eliminar.

Pudiendo también visitar la página web de donde el web Scraper ha extraído la información.

Para finalizar se han comprobado la funcionalidad de los avisos comprobando que estos se crean, se borran y se visualizan correctamente.

5 Conclusión

5.1 Aportaciones realizadas

Una vez finalizado el proyecto se ha logrado realizar la creación de un web Scraper que recompila información de León, además de una página web la cual mejorar experiencia de búsqueda sobre información de León y alrededores para los usuarios que la utilicen con un usuario administrador encargado de gestionar la misma.

5.2 Trabajos futuros

- Añadir nuevo contenido a la aplicación.
- Más cantidad de información a la página web.
- Nuevas funcionalidades
- Creación de recomendaciones dependiendo de lo que los usuarios visiten o agreguen a favoritos.

5.3 Problemas encontrados

Problema: Estudio de cómo hacer web Scraping

Solución: Debido a mi poco conocimiento sobre este tema tuve que realizar una búsqueda sobre estas tecnologías y realizar un aprendizaje sobre las mismas, buscando la forma de recompilar la información que me resultara interesante.

Problema: Web Scraper no obtiene el HTML correcto

Solución: Realizando diversas pruebas me di cuenta de que el problema al obtener el HTML era que al realizar una conexión a esa página web siempre te redireccionaba a la página de inicio de esa web. Después buscando la solución a

este problema me di cuenta de que era debido a que la conexión se realizaba sin una cookie provocando que cada conexión fuera como el primer inicio a esa página provocando la redirección a la página principal, por lo tanto, la solución fue guardar una cookie correspondiente a esta página web pudiendo realizar la obtención de datos correctamente.

Problema: Introducción de Fotos en la base de datos

Solución: búsqueda de cómo crear la clase entidad para que esta contenga un archivo blob para mostrar en la aplicación web

Problema: Pasar las fotos almacenadas en la base de datos de tipo byte a SteamedContent para mostrar en la vista.

Solución: Se intento pasar directamente el tipo byte a SteamedContent, pero esto no fue posible, después se intentó descargar las imágenes para mostrarlas antes de cargar la página, pero esto provocaba lentitud al cargar las páginas de la aplicación. Después de buscar en la documentación de Primefaces se dio con un atributo de GraphicImage el cual permitía mostrar imágenes recompiladas mediante vectores tipo byte

Problema: Introducir el Web Scraper dentro de la aplicación Web en el servidor de payara debido a que el web Scraper provocaba el reinicio de esta, además de excesivo tiempo de espera al cargar las distintas paginas dentro de la aplicación.

Solución: Creación de la aplicación perteneciente al web Scraper de manera separada de la aplicación web.

5.4 Opinión personal

El desarrollo de este proyecto me ha permitido utilizar gran parte de los conocimientos aprendidos a lo largo del grado como puede ser creación de la base de datos, programación, los diagramas realizados, además de esto he podido ampliarlos utilizando técnicas desconocidas para mí como es el móvil First y el web Scraping.

Lista de referencias

1. Salario para Scrum Master - Salario Medio. Online. [Consulta: 24 Mayo 2022]. Disponible en: https://www.glassdoor.es/Sueldos/scrum-master-sueldo-SRCH_K00,12.htm
2. Sueldo: Software Developer (Agosto, 2022) | Glassdoor. Online. [Consulta: 12 Agosto 2022]. Disponible en: https://www.glassdoor.es/Salaries/software-developer-salary-SRCH_K00,18.htm?countryRedirect=true
3. Salario para Diseñador Gráfico en España - Salario Medio. Online. [Consulta: 24 Mayo 2022]. Disponible en: <https://es.talent.com/salary?job=dise%C3%B1ador+gr%C3%A1fico#:~:text=El%20salario%20dise%C3%B1ador%20gr%C3%A1fico%20promedio,hasta%20%E2%82%AC%2025.675%20al%20a%C3%B1o.>
4. Salario para Software Tester - Salario Medio. Online. [Consulta: 24 Mayo 2022]. Disponible en: https://www.glassdoor.es/Sueldos/software-tester-sueldo-SRCH_K00,15.htm
5. BOE 298 de 14/12/1999 Sec 1 Pag 43088 a 43099 | Enhanced Reader. .
6. BOE.es - BOE-A-2018-12699 Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público. Online. [Consulta: 5 Septiembre 2022]. Disponible en: <https://www.boe.es/eli/es/rd/2018/09/07/1112/con>
7. BOE.es - BOE-A-2002-13758 Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico. Online. [Consulta: 5 Septiembre 2022]. Disponible en: <https://www.boe.es/buscar/act.php?id=BOE-A-2002-13758&p=20220713&tn=0>
8. LAZCANO, Christian. *Especificación de Requisitos según el estándar de IEEE 830*. Online. [Consulta: 17 Agosto 2022]. Disponible en: https://www.academia.edu/8970934/Especificaci%C3%B3n_de_Requisitos_seg%C3%BA_n_el_est%C3%A1ndar_de_IEEE_830
9. ¿Que es Java? - Guía de Java empresarial para principiantes - AWS. Online. [Consulta: 18 Agosto 2022]. Disponible en: <https://aws.amazon.com/es/what-is/java/>
10. CSS | MDN. Online. [Consulta: 18 Agosto 2022]. Disponible en: <https://developer.mozilla.org/es/docs/Web/CSS>
11. HTML: Lenguaje de etiquetas de hipertexto | MDN. Online. [Consulta: 18 Agosto 2022]. Disponible en: <https://developer.mozilla.org/es/docs/Web/HTML>
12. ¿Qué es SQL y para qué sirve? | Blog UE. Online. [Consulta: 18 Agosto 2022]. Disponible en: <https://universidadeuropea.com/blog/lenguaje-programacion-sql/>
13. JavaServer Faces(JSF) | Marco de Desarrollo de la Junta de Andalucía. Online. [Consulta: 18 Agosto 2022]. Disponible en: <http://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/101>

14. PrimeFaces Showcase. Online. [Consulta: 24 Agosto 2022]. Disponible en: <https://www.primefaces.org/showcase/index.xhtml?jfwid=35dfb>
15. jsoup: analizador HTML de Java, creado para la edición, limpieza, raspado y seguridad de XSS de HTML. Online. [Consulta: 18 Agosto 2022]. Disponible en: <https://jsoup.org/>
16. Payara Platform Community - Payara Services Ltd. Online. [Consulta: 24 Agosto 2022]. Disponible en: <https://www.payara.fish/products/payara-platform-community/>
17. La API de persistencia de Java: ¿Qué es JPA? - JPA vs Hibernate vs EclipseLink vs Spring JPA | campusMVP.es. Online. [Consulta: 24 Agosto 2022]. Disponible en: <https://www.campusmvp.es/recursos/post/la-api-de-persistencia-de-java-que-es-jpa-jpa-vs-hibernate-vs-eclipselink-vs-spring-jpa.aspx>
18. Enterprise Java Beans (EJB) - GeeksforGeeks. Online. [Consulta: 24 Agosto 2022]. Disponible en: <https://www.geeksforgeeks.org/enterprise-java-beans-ejb/>
19. HeidiSQL - MariaDB, MySQL, MSSQL, PostgreSQL and SQLite made easy. Online. [Consulta: 24 Agosto 2022]. Disponible en: <https://www.heidisql.com/>
20. Noticias de León | leonoticias. Online. [Consulta: 6 Septiembre 2022]. Disponible en: <https://www.leonoticias.com/leon/>
21. Viajes a León Ciudad (León) y turismo en León Ciudad | Minube. Online. [Consulta: 6 Septiembre 2022]. Disponible en: <https://www.minube.com/viajes/espana/leon/leon>
22. Agenda de eventos y ocio en León. Online. [Consulta: 6 Septiembre 2022]. Disponible en: <https://leonocio.es/>
23. Terranostrum.es Ocio, turismo y gastronomía en Castilla y León. Online. [Consulta: 6 Septiembre 2022]. Disponible en: <https://www.terranostrum.es/>
24. Deportes de Aventura, Actividades - Yumping.com. Online. [Consulta: 6 Septiembre 2022]. Disponible en: <https://www.yumping.com/>
25. MySQL :: Download MySQL Connector/J (Archived Versions). Online. [Consulta: 6 Septiembre 2022]. Disponible en: <https://downloads.mysql.com/archives/c-j/>

ANEXO A: Control de Versiones

Para el control de versiones de ambas aplicaciones se ha utilizado plataforma GitHub, En el cual se ha creado repositorios donde se ha ido guardando el código generado. Además, se ha vinculado esta plataforma con NetBeans pudiendo generar las distintas actualizaciones.

También se ha utilizado Google drive para el almacenamiento de la copia de seguridad de la documentación del proyecto y los diagramas generados.

Enlace al repositorio de GitHub donde se almacena el código de la aplicación web:

<https://github.com/fferni01/AplicacionWebLeon>

Enlace al repositorio de GitHub donde se almacena el código del web Scraper:

<https://github.com/fferni01/WebScraper>

ANEXO B:

Seguimiento de

proyecto fin de

carrera

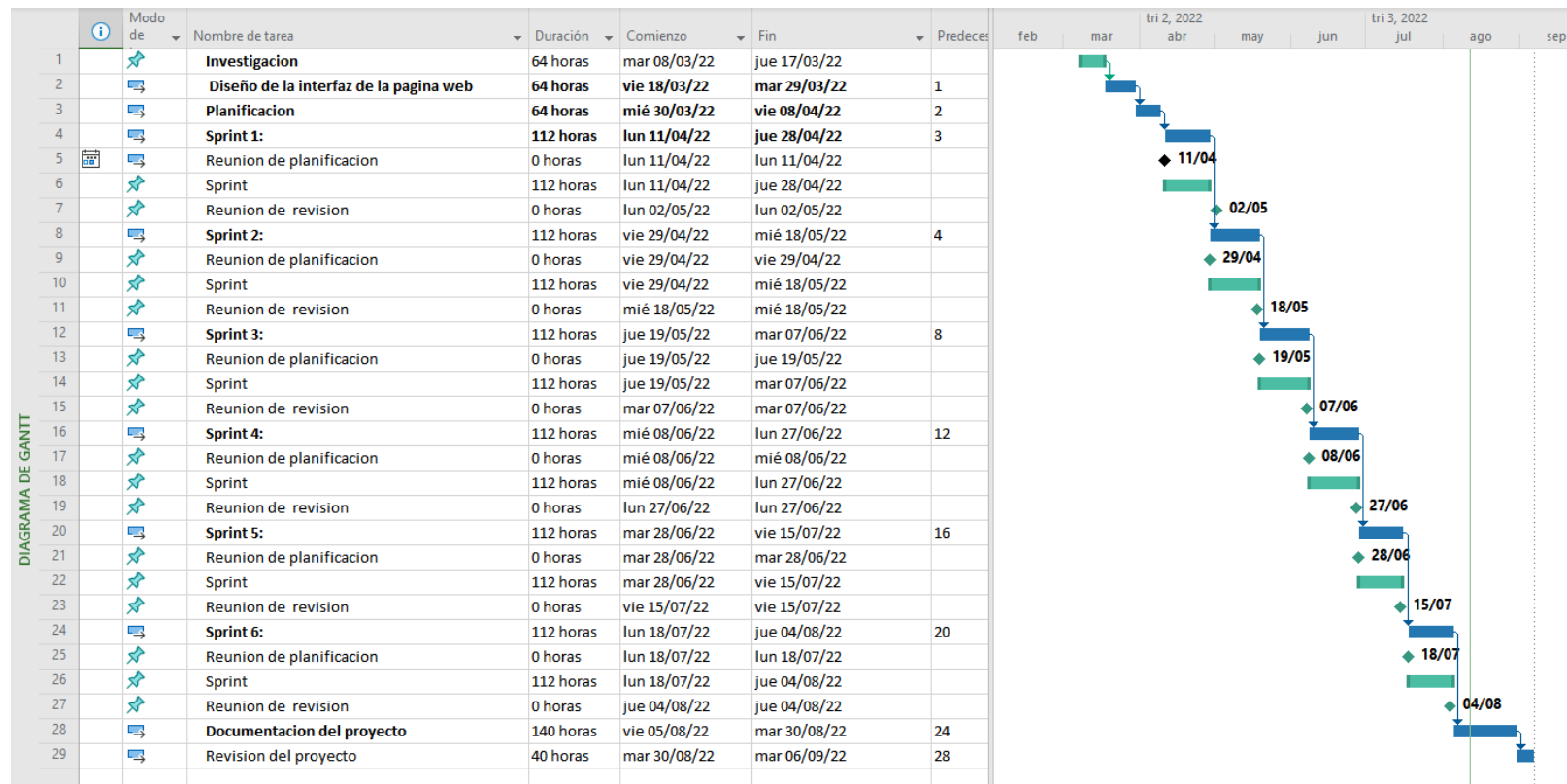


Figura A.1 Diagrama de Gantt

ANEXO C: Manual de usuario

C.1 Manual para Usuarios

Cuando una persona entra en la aplicación web lo primero que esta se encuentra es un inicio de sesión, pudiendo iniciar sesión en la aplicación o registrarse como se puede ver en la figura C.1

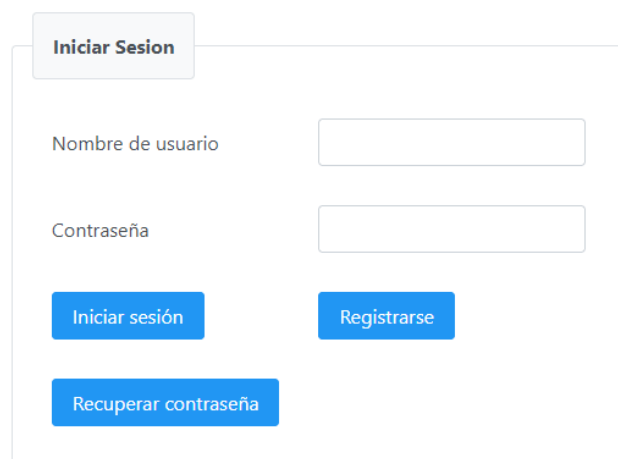
La interfaz de inicio de sesión se presenta en un recuadro con un título "Iniciar Sesión" en un botón superior izquierdo. Dentro del recuadro, hay dos campos de entrada: "Nombre de usuario" y "Contraseña". Debajo de estos campos, hay tres botones azules: "Iniciar sesión", "Registrarse" y "Recuperar contraseña".

Figura C.1 Interfaz Iniciar sesión

Si el usuario no se acuerda de la contraseña puede presionar el botón recuperar contraseña pudiendo introducir el nombre de usuario para que el sistema le mande un correo electrónico con la contraseña.

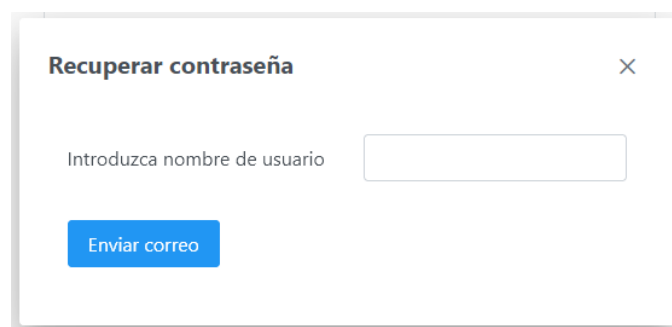
La interfaz de recuperación de contraseña se presenta en un recuadro con un título "Recuperar contraseña" y un botón de cerrar "X" en la esquina superior derecha. Dentro del recuadro, hay un campo de entrada con el texto "Introduzca nombre de usuario". Debajo del campo, hay un botón azul "Enviar correo".

Figura C.2 Interfaz recuperar contraseña

Si el usuario no se ha registrado presionaría sobre el botón registrarse el cual le llevaría a la pantalla de registro como se puede ver en la figura C.2. En esta página el usuario deberá introducir los datos pedidos, en caso de que este no los introduzca los datos necesarios para darse de alta en la aplicación el sistema le mostrara un aviso como podemos ver en la figura C.3. Para la creación de la contraseña el usuario deberá introducir 8 letras y como mínimo una letra mayúscula, una minúscula y un número. También el usuario deberá elegir un nombre de usuario el cual no este dado de alta en la página web.

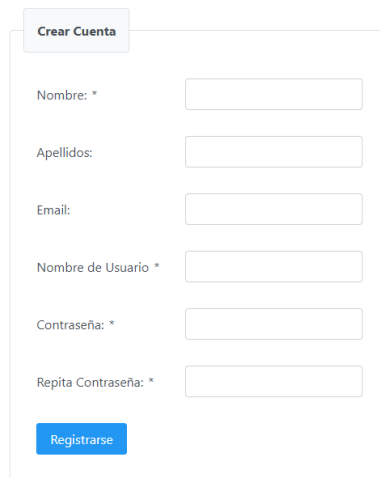
El formulario de creación de cuenta está encerrado en un recuadro con el título "Crear Cuenta" en la parte superior izquierda. Contiene seis campos de entrada de texto con las siguientes etiquetas: "Nombre: *", "Apellidos:", "Email:", "Nombre de Usuario *", "Contraseña: *" y "Repita Contraseña: *". Cada campo tiene un botón de borrar (X) a la izquierda. En la parte inferior del recuadro hay un botón azul con el texto "Registrarse".

Figura C.3 Interfaz registrarse

⊗ Contraseña: Error de validación. Contraseña debería coincidir con Repita Contraseña.

Figura C.4 Aviso Contraseña invalida

Una vez el usuario ha iniciado Sesión se cargará la página principal de la aplicación y arriba a la derecha podemos encontrar un botón el cual al presionarlo mostrará el menú de la aplicación. Dentro del menú podemos observar el distinto contenido que la aplicación web contiene pudiendo

presionar en cualquiera de este para poder visualizarlo como se muestra en la figura C.5.

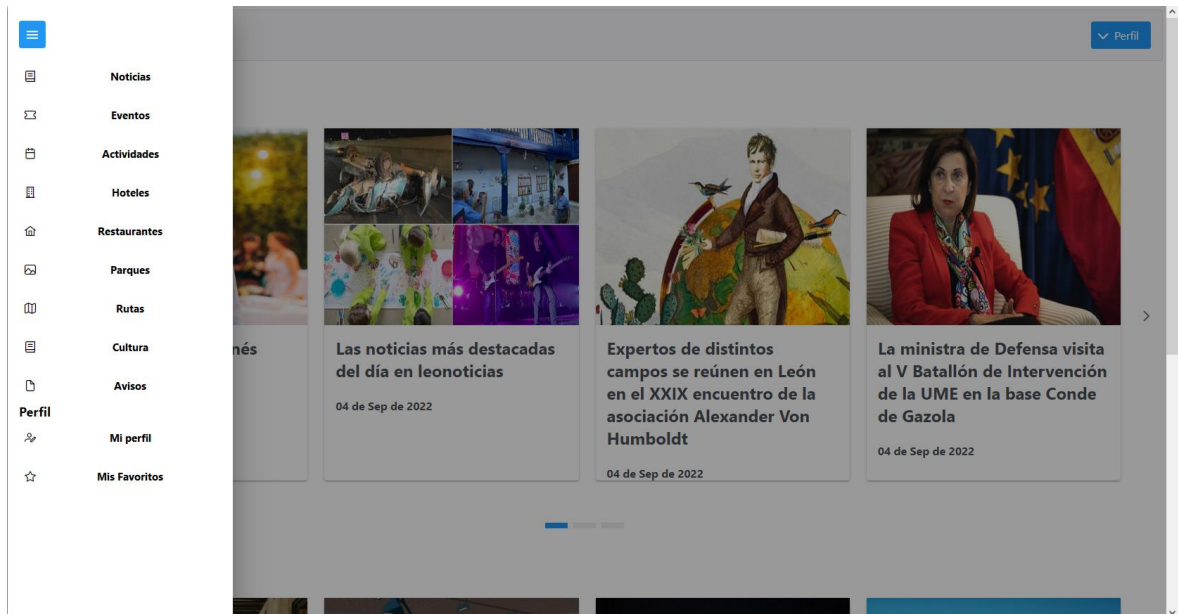


Figura C.5 Menú de la aplicación

Como podemos observar cada contenido tiene un botón que representa un corazón, otro icono que representa los comentarios y un texto ver más.

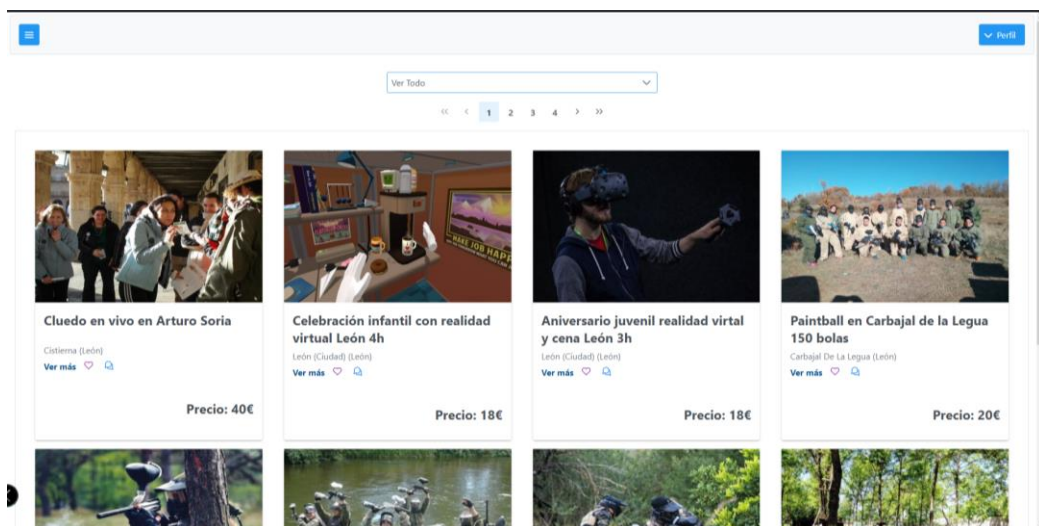


Figura C.6 Pagina interfaz actividades

El texto ver más es un nos permite si presionamos sobre el dirigarnos a la página web de donde esta información ha sido obtenida pudiendo así ver más en profundidad esta.

Si presionamos el icono del corazón el sistema nos preguntara si queremos añadir este contenido a favoritos como se puede ver en la figura C.7. y una vez añadido este el corazón pasa a estar lleno como se ve en la figura C.8.



Figura C.7 Pregunta interfaz añadir a favoritos



Figura C.8 Contenido añadido a favoritos

Si el usuario presiona sobre los comentarios se abran los comentarios pertenecientes a este contenido. Una vez dentro de la pantalla de los comentarios el usuario podrá realizar comentarios, editar y eliminar los comentarios realizados por el como se puede ver en la siguiente figura.

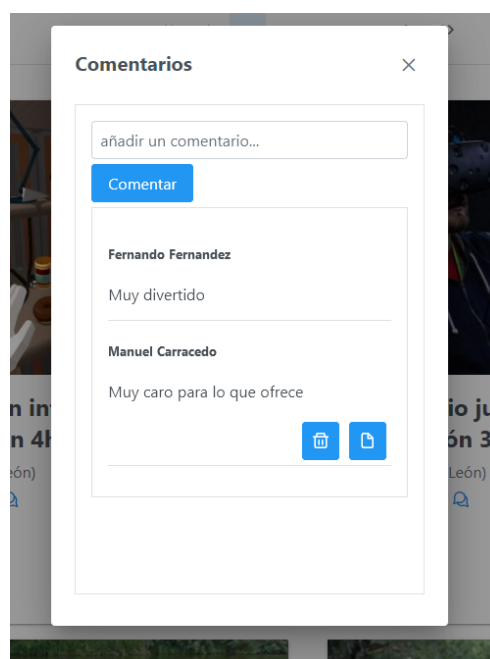


Figura C.9 Interfaz comentarios

También dentro de algunas páginas como pueden ser la página web de actividades o la de eventos puedes encontrar un filtro en con el cual poder filtrar el contenido para buscar algo más específico.

También dentro de algunas páginas como pueden ser la página web de actividades o la de eventos puedes encontrar un filtro en con el cual poder filtrar el contenido para buscar algo más específico.

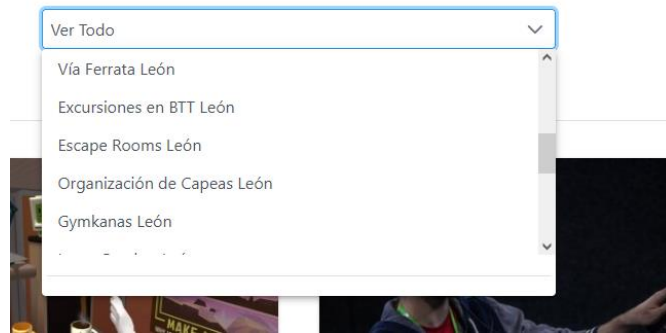


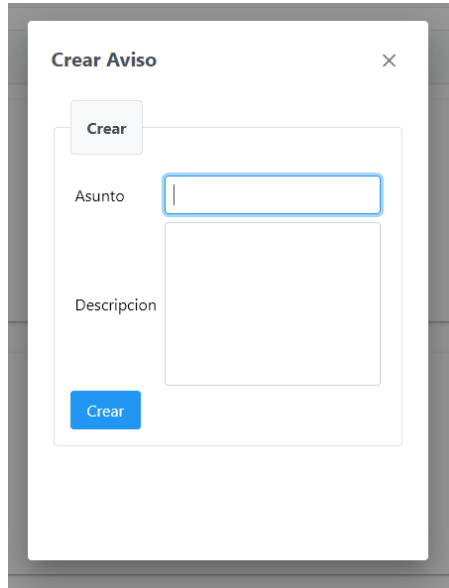
Figura C.10 Filtro de contenido

Si en el menú se accede a las noticias y una vez dentro se presiona sobre alguna de ellas te llevará a la página para poder leer la misma como podemos ver en la figura C.10



Figura C.11 Visualizar noticias

Si el usuario en el menu escoge la opcion avisos podra visualizar los informes que otros usuarios han creado sobre alguna incidencia que haya ocurrido, si el usuario presiona el boton Crear Aviso se mostrar un formulario el cual debera rellenar para crear el aviso.



El formulario 'Crear Aviso' presenta un diseño limpio con un fondo gris. En la parte superior, el título 'Crear Aviso' está acompañado de un icono de cerrar (X). El formulario mismo tiene un fondo blanco y una sombra. Al principio del formulario hay un botón gris con el texto 'Crear'. A continuación, se encuentran dos campos de entrada: 'Asunto' con un campo de texto rectangular y 'Descripción' con un área de texto más grande. En la parte inferior del formulario, hay un botón azul con el texto 'Crear'.

Figura C.12 Creación de un aviso

Si el usuario presiona en favoritos dentro del menu la pagina web cargara los favoritos que este usuario ha añadido como se muestra en la figura C.10

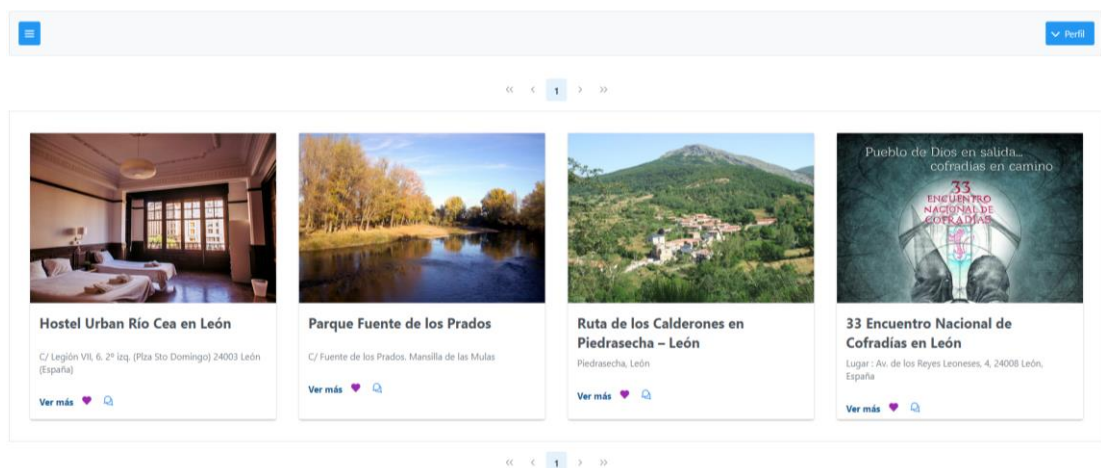


Figura C.13 Interfaz favoritos

Por último, en la parte de arriba a la derecha tiene un menú desplegable el cual contiene dos opciones mi perfil y cerrar Sesión, Si el usuario presionara en mi perfil se le muestran sus datos pudiendo modificarlos como se puede observar en la siguiente figura.

Nueva pestaña Nueva pestaña Facilets Template Payara Server 5.104 Rhadaosfish Facilets Template Facilets Template

localhost:5080/TCG/usuario/Perfil

Perfil

Datos Personales

Nombre: * Fernando

Apellido: * Fernandez

email 052fer5@gmail.com

Modificar

Cambiar Contraseña

Cambiar Contraseña: *

Repita nueva Contraseña: *

Cambiar contraseña

Figura C.14 Interfaz mi perfil

C.2 Manual para Usuario Administrador

El administrador es el usuario encargado de la administración del sistema, además de ser el encargado de activar el web Scraper.

Lo primero que se encontrara el administrador al iniciar la aplicación es página de inicio de sesión, El administrador contara con una cuenta que se le abra administrado con el usuario y contraseña. Una vez el administrador inicie sesión en la aplicación y siendo la primera vez que este inicia se le preguntaran datos sobre este.

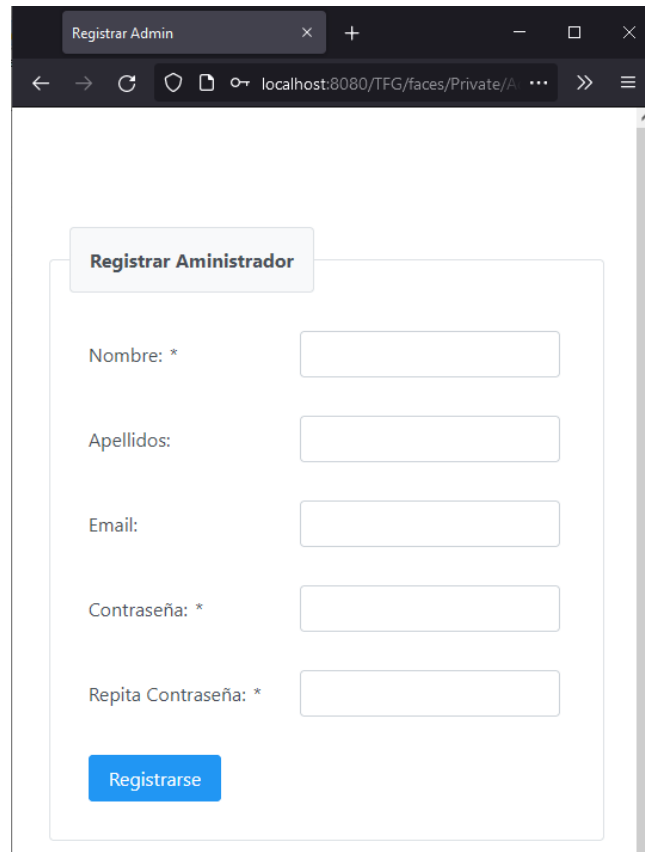
A screenshot of a web browser window showing a registration form for an administrator. The browser's address bar shows 'localhost:8080/TFG/faces/Private/A...'. The form is titled 'Registrar Admin' and contains the following fields: 'Nombre: *', 'Apellidos:', 'Email:', 'Contraseña: *', and 'Repita Contraseña: *'. Each field has a corresponding text input box. At the bottom of the form is a blue button labeled 'Registrarse'.

Figura C.15 Interfaz primer inicio administrador

Una vez iniciada sesión el administrador puede acceder a un menú igual al del usuario donde poder visualizar el contenido, apareciendo dentro de este un botón que con una cruz que representa eliminar, si el administrador presiona en este el sistema le preguntara si está seguro para evitar la eliminación de contenido involuntariamente.

También el administrador dentro podrá ver los comentarios realizados por los usuarios pudiendo eliminarlos si este así lo decide.

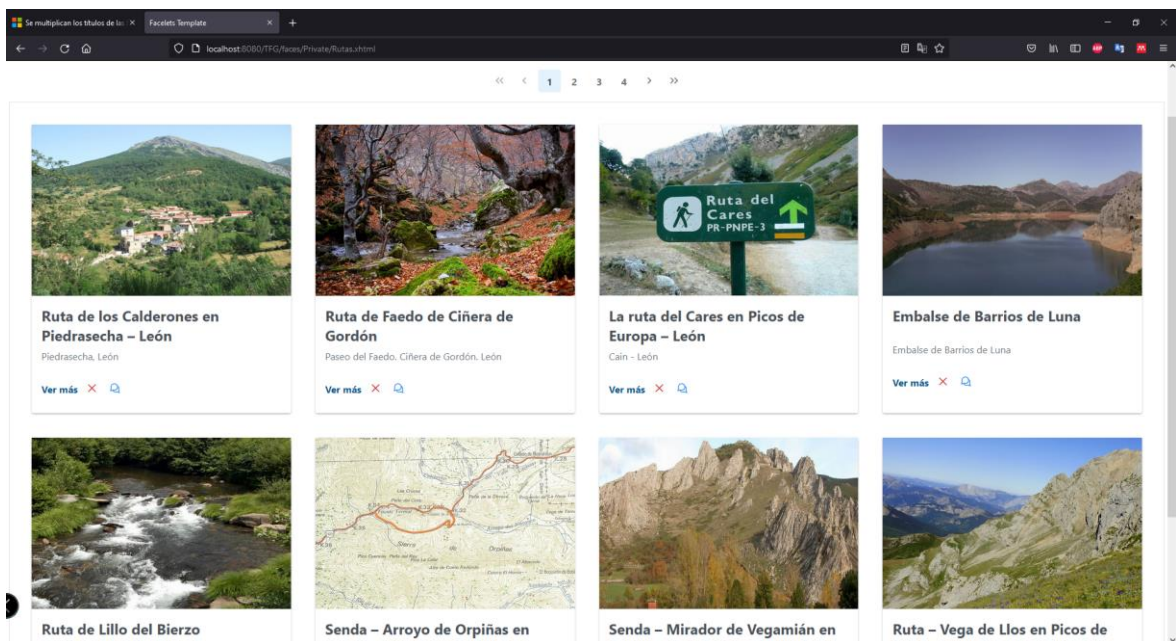


Figura C.16 interfaz rutas administrador

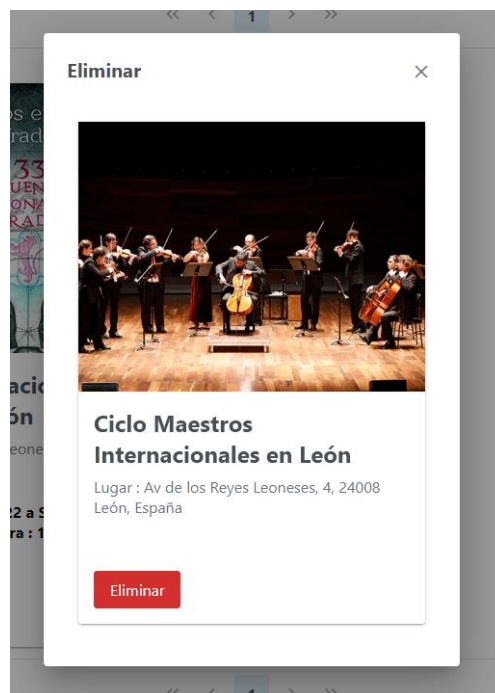
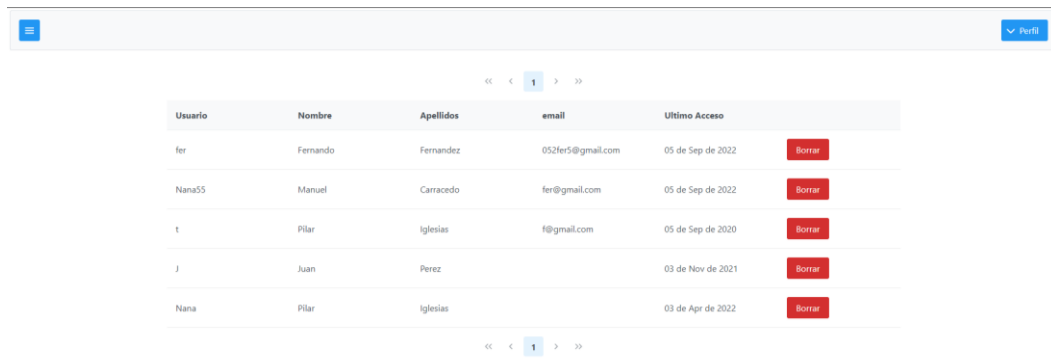


Figura C.17 Eliminar Contenido

También el administrador puede acceder a una tabla donde se le muestran los usuarios pudiendo ver la información de esto con su ultimo acceso pudiendo decidir si darlo de baja de la aplicación como se puede ver en la siguiente figura.



Usuario	Nombre	Apellidos	email	Ultimo Acceso	
fer	Fernando	Fernandez	052fer5@gmail.com	05 de Sep de 2022	Borrar
Nana55	Manuel	Carracedo	fer@gmail.com	05 de Sep de 2022	Borrar
t	Pilar	Iglesias	f@gmail.com	05 de Sep de 2020	Borrar
J	Juan	Perez		03 de Nov de 2021	Borrar
Nana	Pilar	Iglesias		03 de Apr de 2022	Borrar

Figura C.18 Tabla usuarios

Por Ultimo el administrador es el encargado de abrir la aplicación de web Scraper pudiendo activarla y desactivarla cuando este lo considere oportuno una vez creada el sistema creara dos hilos uno encargado de la recopilación de las noticias y otro encargado de la recopilación del resto del contenido. Cada hilo tiene un temporizador el cual se encarga de dormirlo hasta que sea la hora de ejecutar el web Scraper de nuevo y de forma automática. Las noticias se recompilan cada hora mientras que el resto del contenido se comprueba una vez al día debido a que es un contenido que no varía tanto. Una vez activado este mostrara los datos que vaya recompilando al administrador junto a su fecha de ejecución.

También podemos ver un filtro en las figuras C.19 y C.20 de contenido en la parte de arriba a la derecha donde el administrador puede realizar una búsqueda rápida de instalaciones deportivas, hoteles, restaurantes, edificios culturales, noticias, rutas, parques naturales, actividades y eventos

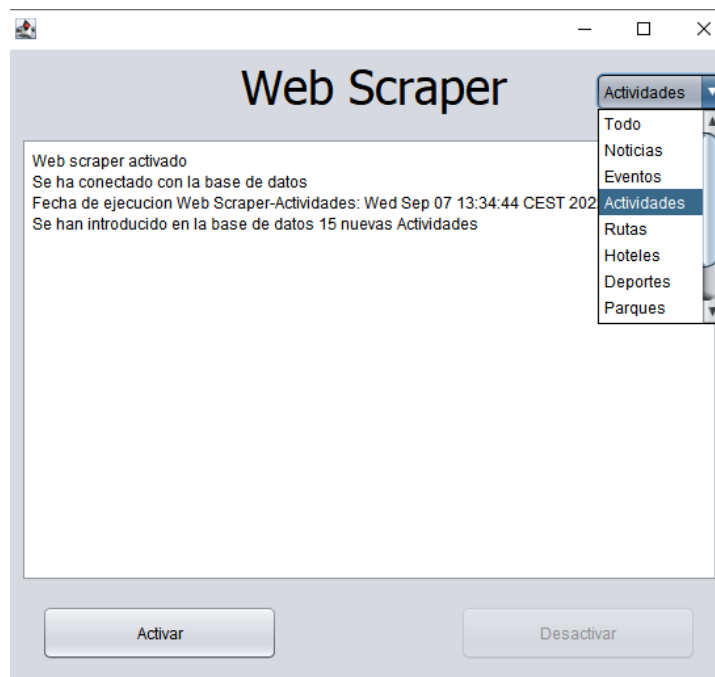


Figura C.19 Filtro web Scraper

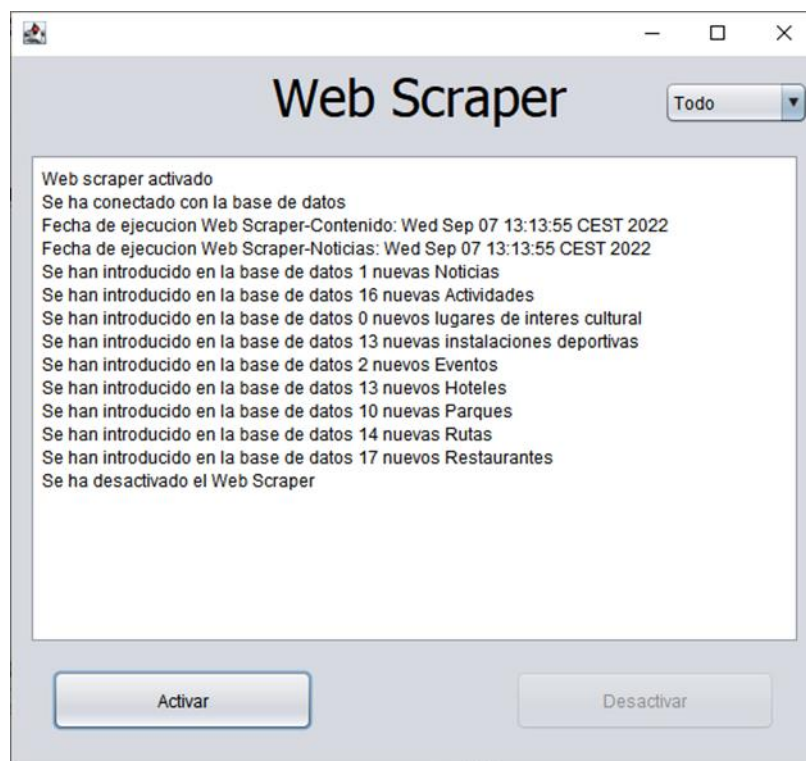


Figura C.20 Interfaz Web Scraper

Anexo D: Manual del programador

EL programador deberá seguir los siguientes puntos al realizar el desarrollo del proyecto.

- Cada vez que se cree una nueva tabla dentro de la base de datos se tiene que crear una clase entidad dentro del paquete modelo.
 - Una vez creada tendrá que crear la clase asociada de la manera en la que se muestra en la Figura D.1 en el paquete EJB.

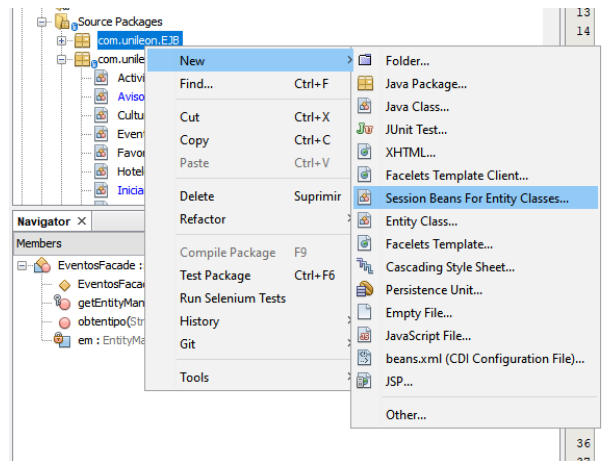


Figura D.1 Como crear Clases EJB

- Si se crea un nuevo HTML el programador deberá crear una nueva clase controladora que será la cual estará asociada al HTML.
- Si se quiere realizar una consulta JPQL se deberá crear la consulta en la clase fachada de la tabla de base de datos asociada.
- Cada página web añadida al web Scraper dispondrá de una clase propia donde se extraigan los datos.

D.1. Manual de instalación de la aplicación web

- Importar la base de datos a MySQL.
- Ejecutar la aplicación java que contiene el Web Scraper
- Importar el proyecto que contiene la aplicación web en NetBeans.
- Colocar el conector JDBC de MySQL versión 5.1.47 el cual puedes encontrar en la página de MySQL(25) en la carpeta lib del dominio que inicie el servidor de aplicaciones que estará dentro de la carpeta asociada a Payara.

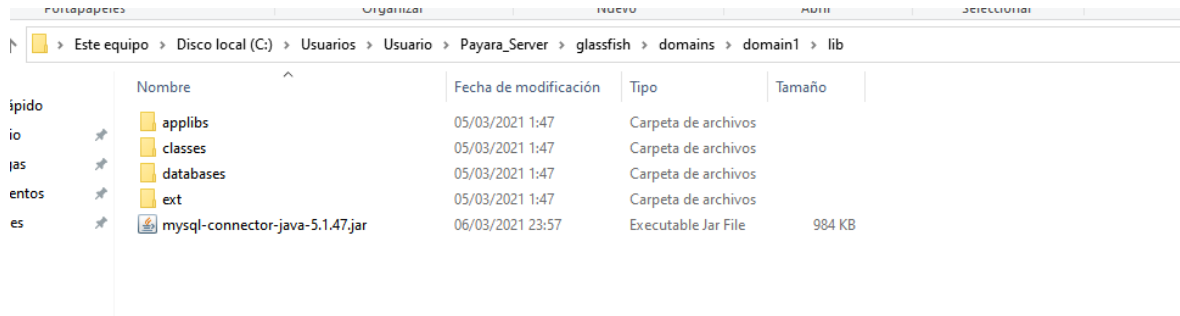


Figura D.2 Carpeta dominio/lib

- Iniciar el servidor Payara

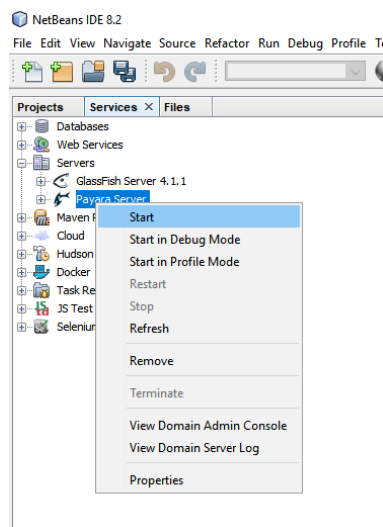


Figura D.3 Iniciar Payara

- Entramos dentro de la página web de payara que se encontrara en el puerto 4848 para crear el pool de conexiones en el servidor de aplicaciones

- Asociar el pool de conexión con el recurso JDBC asociado en el proyecto en JDBC como se muestra en la figura D.4. y añadiendo en las propiedades adicionales como se ven la figura D.5 los siguientes datos:
 - Port: Puerto de la base de datos
 - PortNumber: Puerto de la base de datos
 - Url: Añadiendo al Url el nombre de la base de datos
 - User: Usuario de la base de datos
 - Password: Contraseña de la base de datos
 - DatabaseName: Nombre de la base de datos
 - ServerName: localhost.
 - AllowPublicKeyRetrieval: True
 - Añadimos la una propiedad:
 - driverClass con el valor com.mysql.jdbc.Driver

New JDBC Connection Pool (Step 1 of 2)

Identify the general settings for the connection pool.

* Indicates required field

General Settings

Pool Name: *

Resource Type:
 Must be specified if the datasource class implements more than 1 of the interface.

Database Driver Vendor:
 Select or enter a database driver vendor

Introspect: ☒ Enabled
 If enabled, data source or driver implementation class names will enable introspection.

Figura D.4 Pool de conexión

Additional Properties (231)			
<input type="checkbox"/> <input type="checkbox"/> Add Property Delete Properties			
Select	Name	Value	Description
<input type="checkbox"/>	SelfDestructOnPingSecondsLifetime	0	
<input type="checkbox"/>	UseUsageAdvisor	false	
<input type="checkbox"/>	ServerAffinityOrder		
<input type="checkbox"/>	AllowSlaveOwnConnections	false	
<input type="checkbox"/>	LoadBalanceBlacklistTimeout	0	
<input type="checkbox"/>	QueryTimeoutKillsConnection	false	
<input type="checkbox"/>	CacheServerConfiguration	false	
<input type="checkbox"/>	RoundRobinLoadBalance	false	
<input type="checkbox"/>	ClientCertificateKeyStoreUrl		
<input type="checkbox"/>	UseCursorFetch	false	
<input type="checkbox"/>	JdbcCompliantTruncation	true	

Figura D.5 Propiedades

- Entramos dentro de JDBC Resources y creamos el recurso de ese pool de conexiones.

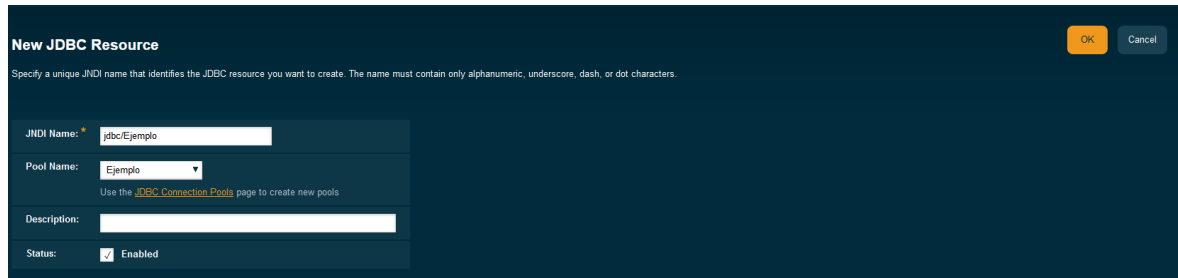


Figura D.6 JDBC recurso

- Dentro de persistence.xml del proyecto cambiamos el recurso que este puesto por el que acabamos de crear.

```
<?xml version="1.0" encoding="UTF-8"?>
<persistence version="2.1" xmlns="http://xmlns.jcp.org/xml/ns/persistence" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="
http://xmlns.jcp.org/xml/ns/persistence/persistence.xml"
<persistence-unit name="leonPU" transaction-type="JTA">
  <jta-data-source>jdbc/tfg</jta-data-source>
  <exclude-unlisted-classes>false</exclude-unlisted-classes>
  <properties>
    <property name="javax.persistence.schema-generation.database.action" value="create"/>
  </properties>
</persistence-unit>
</persistence>
```

Figura D.7 Persistence.xml

- Iniciar la aplicación web.