



Diseño de una aplicación web que centraliza las notificaciones de diferentes ayuntamientos en un solo portal de manera automatizada

Sergio Fabregat Ramos
Grado en Ingeniería Informática
Desarrollo Web

Consultor: Gregorio Robles Martínez
Profesor responsable de la asignatura: Santi Caballe Llobet

17 de enero de 2023



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	Diseño de una aplicación web que centraliza las notificaciones de diferentes ayuntamientos en un solo portal de manera automatizada
Nombre del autor:	Sergio Fabregat Ramos
Nombre del consultor:	Gregorio Robles Martínez
Fecha de entrega (mm/aaaa):	01/2021
Área del Trabajo Final:	Desarrollo Web
Titulación:	Grado en Ingeniería Informática
Resumen del Trabajo (máximo 250 palabras):	
<p>Desarrollar y documentar la creación de una aplicación web que registre de manera centralizada y automatizada las notificaciones (información, avisos, noticias...) aportadas por los diferentes ayuntamientos de varios municipios de tal manera que el usuario al ingresar al aplicativo web pueda escoger a su preferencia de que municipios en concreto quiere leer las notificaciones, todo en un mismo portal web.</p> <p>El proyecto se divide en dos partes, por un lado, crear un portal web desarrollado con los lenguajes HTML, CSS, PHP y Javascript.</p> <p>Por otro lado, crear el sistema automatizado de registro de notificaciones utilizando Python. Los datos se obtienen mediante "Web scraping", una técnica utilizada para extraer información de sitios web que consiste en consultar los datos obtenidos por el protocolo HTTP.</p> <p>Los datos de usuarios, como los de los ayuntamientos y notificaciones se registran mediante el sistema de gestión de base de datos MySQL.</p>	

Abstract (in English, 250 words or less):

Develop and document the making of a web app that registers in a centralized and automated way the notifications (information, notices, news...) provided by the different city councils, where the users, login in the website, can choose which notifications the app have to show.

The project is split into two parts, on the one hand, developing the web portal with HTML, CSS, PHP and Javascript languages.

On the other hand, develop the automated notification logging software using Python. The data is obtained through "Web scraping", used for extracting data from websites through the HTTP protocol.

User data, such as city councils and notifications, are stored using the MySQL database management system.

Palabras clave (entre 4 y 8):

Web Scraping

Database Management

Web App

Software

HTTP protocol

Índice

1. Introducción.....	1
1.1 Contexto y justificación del Trabajo.....	1
1.2 Objetivos del Trabajo.....	1
1.3 Enfoque y método seguido.....	2
1.4 Planificación del Trabajo.....	3
1.5 Breve resumen de productos obtenidos.....	4
1.6 Breve descripción de los otros capítulos de la memoria.....	4
2. Análisis.....	5
2.1 Tabla de riesgos:.....	5
2.2 Tabla de acciones para mitigar los riesgos:.....	5
2.3 Requisitos del proyecto e implementación:.....	6
2.4 Grupos de interés (Stakeholders).....	6
2.5 Requisitos funcionales.....	7
2.6 Roles.....	7
2.7 Flujo de navegación.....	8
3. Diseño.....	9
3.1 Wireframes.....	9
3.2 Wireframe alta definición (prototipo).....	15
3.3 Bases de datos.....	16
4. Implementación y funcionamiento.....	18
4.1 Funcionamiento del Script.....	22
4.2 Ejecución del código.....	23
4.3 Funcionamiento del aplicativo a nivel usuario.....	27
4.4 Proceso de instalación.....	29
5. Conclusiones.....	43
6. Glosario.....	44
7. Bibliografía.....	45
8. Anexos.....	46

Lista de figuras

Figura 1 – Diagrama de Gantt, temporalización de del proyecto.....	3
Figura 2 - Diagrama de Flujos	8
Figura 3 – Wireframe Landing page	9
Figura 4 – Wireframe User Homepage	10
Figura 5 – Wireframe Admin Homepage	11
Figura 6 – Wireframe página de Login	12
Figura 7 – Wireframe página de Ajustes	13
Figura 8 – Wireframe página de Registro	14
Figura 9 – Wireframe alta definición Homepage	15
Figura 10 – Modelo entidad-relación de la base de datos	16
Figura 11 – Modelo relacional de la base de datos	17
Figura 12 – Página de aterrizaje de un usuario sin sesión	18
Figura 13 – Código fuente de la Página de aterrizaje encargado de verificar el usuario	18
Figura 14 – Versión final de la página de registro de usuario	19
Figura 15 - Versión final de la página de inicio de sesión de usuario.....	19
Figura 16 – Página principal de usuario donde se muestran las notificaciones	20
Figura 17 – Página de la sección de ajustes de usuario	20
Figura 18 – Panel de administración (Página principal de usuario con rol administrador)	21
Figura 19 – Ejemplo diseño “responsive”	21
Figura 20 – Representación gráfica de la interacción de Python con la base de datos	22
Figura 21 – Importación de las librerías necesarias para la ejecución del código	23
Figura 22 – Ejemplo de obtención de la última notificación	23
Figura 23 – Revisión del documento que contiene la última notificación .	23
Figura 24 – Comprobación de nueva notificación	24
Figura 25 – Solicitud HTTP de la nueva dirección obtenida	24
Figura 26 – Ejemplo de obtención de título y posterior traducción	24

Figura 27 – Ejemplo de inserción de datos en la Base de Datos mediante la librería PyMysl	25
Figura 28 – Ejemplo de escritura en el documento de ultima notificación con la dirección obtenida	25
Figura 29 – Python trabajando en conjunto con Bs4 y Requests	25
Figura 30 – Listado de aplicaciones Python en el servidor donde esta subida la aplicación en la nube	26
Figura 31 – Listado de trabajos cron	26
Figura 32 – Notificación de registro de nueva cuenta exitoso	27
Figura 33 – Panel de notificaciones después de la creación de cuenta ..	27
Figura 34 – Panel de selección de ayuntamientos	27
Figura 35 – Aviso de modificación exitosa de datos	28
Figura 36 – Verificación del correcto funcionamiento del aplicativo en el panel inicial	28
Figura 37 – Proceso de instalación XAMPP	30
Figura 38 – Panel de control de XAMPP	31
Figura 39 – Estructura de archivos y ficheros	32
Figura 40 – Panel de phpMyAdmin	33
Figura 41 – Inserción del nombre de la base de datos	33
Figura 42 – Botón Importar en la barra superior	33
Figura 43 – Importación de la base de datos	34
Figura 44 – Correcta importación de la base de datos	34
Figura 45 – Panel inicial sin sesión ejecutado en versión local	35
Figura 46 – Panel inicial con sesión ejecutado en versión local	36
Figura 47 – Panel de administración de usuario ejecutado en versión local	36
Figura 48 – Botón de cierre de sesión	37
Figura 49 – Panel de administrador ejecutado en versión local	37
Figura 50 – Página de descarga de Python	38
Figura 51 – Instalador de Python	38
Figura 52 – Ayuntamientos utilizados para las pruebas de funcionamiento	39
Figura 53 – Proceso de instalación de pip	40
Figura 54 – Proceso de instalación de las librerías necesarias	40
Figura 55 – Notificaciones mostradas antes de la ejecución del código .	41

Figura 56 – Ejecución del script	41
Figura 57 – Respuesta en consola de la ejecución del script	42
Figura 58 – Notificaciones mostradas después de la ejecución del código	42
Figura 57 – Líneas de trabajo futuras.....	43
Figura 59 – Proyecto completo en GitHub	46

Lista de tablas

Tabla 1 – Evaluación de riesgos	5
Tabla 2 – Acciones para la mitigar los riesgos	5
Tabla 3 – Requisitos del proyecto: Servidor	6
Tabla 4 – Requisitos del proyecto: Cliente	6
Tabla 5 – Requisitos funcionales de la aplicación	7

1. Introducción

1.1 Contexto y justificación del Trabajo

En los últimos años se ha podido apreciar como cada vez más existe una intención por parte de las organizaciones de digitalizar productos, servicios u operaciones no digitales derivando en una búsqueda constante de soluciones relacionadas con la tecnología digital. [1]

Uno de los aspectos a destacar es la mejora respecto a las comunicaciones, tanto entre personas, como en las comunicaciones emitidas desde diferentes entidades. Este proyecto se focaliza en las comunicaciones emitidas desde organismos estatales como pueden ser los ayuntamientos de los diferentes municipios, que desde hace años emiten las notificaciones telemáticamente.

Los usuarios, para consultar estas comunicaciones tienen que entrar en los diferentes portales web dependiendo de la entidad emisora puesto que son independientes. La solución que se aporta en este proyecto es la de centralizar dichas comunicaciones en un solo portal web de tal manera que el usuario al ingresar al aplicativo web pueda escoger a su preferencia de que municipios quiere leer las notificaciones todo en un mismo portal web.

Se trata de una solución muy útil para aquellas personas o entidades que deseen recibir las notificaciones de diferentes ayuntamientos. También para agencias de noticias que dependen de recibir las notificaciones por parte del responsable de comunicación de cada ayuntamiento o revisar periódicamente los diferentes portales.

1.2 Objetivos del Trabajo

Desarrollar y documentar la creación de una aplicación web que registre de manera centralizada y automatizada las notificaciones (información, avisos, noticias...) aportadas por los diferentes ayuntamientos de varios municipios de tal manera que el usuario al ingresar al aplicativo web pueda escoger a su preferencia de que municipios en concreto quiere recibir notificaciones, todo en un mismo portal web.

1.3 Enfoque y método seguido

Las herramientas y tecnologías que se utilizaran para el proyecto son:

Para la extracción, recopilación e inyección de datos den la bbdd de manera automatizada se utilizará el lenguaje de programación Python con las siguientes librerías [2], [3].

- Beautiful Soup, para el análisis de documentos HTML [4]
- Requests, para realizar solicitudes HTTP [5]

Este método de obtención de datos se conoce como “*Web scraping*” y se trata de una técnica utilizada mediante programas de software para extraer información de sitios web simulando la navegación humana normalmente mediante el protocolo HTTP. [6]

El programa de Python se encargará de rellenar la base de datos con la información aportada por los diferentes ayuntamientos en sus respectivos portales de:

- Título
- Foto (Link)
- Artículo

Este proceso se realizará de forma automatizada cada cierto periodo de tiempo mediante el uso de un “cronjob” y será mostrado a los usuarios que deseen recibir las notificaciones de dicho municipio.

Para la creación de el portal web se utilizará:

- PHP + HTML, para el diseño y la ejecución de las páginas.
- CSS + Bootstrap, para el diseño de las hojas de estilo [7]
- Javascript, para el funcionamiento de scripts
- MySQL, como gestor de base de datos

Para el desarrollo del portal se creará una plataforma con todas las funcionalidades que requiere una web que utiliza sistema de usuarios (Login, registro, sesiones, panel de usuario...).

También se programará mediante el uso de PHP y MySQL la conexión a la base de datos para permitir que el usuario pueda realizar las consultas y que el programa sea funcional.

Puesto que cada ayuntamiento tiene una estructura diferente, hay que programar una versión del código diferente para cada uno, en este caso se implementará en 4 ayuntamientos diferentes.

Casos de uso

Se trata de un aplicativo muy útil tanto para todos los usuarios interesados en obtener información de los diferentes municipios de manera rápida y centralizada. También puede utilizarse para fines profesionales como una agencia de noticias.

1.4 Planificación del Trabajo

En la figura 1 se representa la planificación detallada mediante un diagrama de Gantt.

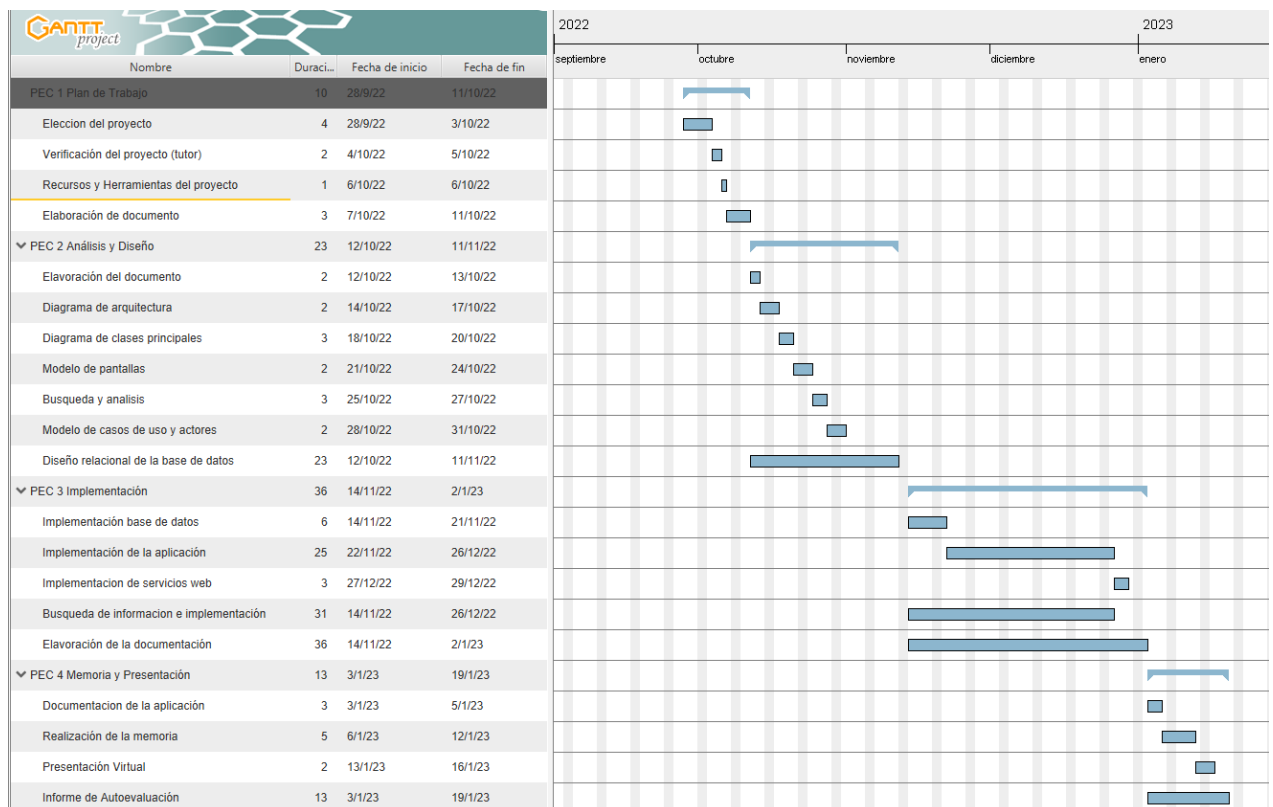


Figura 1 – Diagrama de Gantt, temporalización del proyecto

1.5 Breve resumen de productos obtenidos

Los productos obtenidos del desarrollo de este proyecto son:

- Documentación del proceso y elaboración del producto.
- Código fuente de la aplicación.
- Versión online del producto final.
- Memoria del trabajo final.
- Video explicativo de la aplicación.
- Presentación con diapositivas.

1.6 Breve descripción de los otros capítulos de la memoria

Evaluación de riesgos y acciones: Se identifican los posibles riesgos y las acciones para mitigarlos en caso de que se produzcan.

Requisitos del proyecto e implementación: Se detallan los recursos que se utilizan para la implementación del proyecto.

Grupos de interés (Stakeholders): Se muestran los grupos de interés que se relacionan con las actividades y decisiones.

Roles: Se explican los diferentes roles de usuario.

Diseño: Diseño de la aplicación mediante el uso de wireframes del resultado esperado.

Modelo entidad relación: Diagrama del modelo entidad-relación de la base de datos.

Diseño relacional de la base de datos: Diseño del modelo relacional de la base de datos con los nombres y atributos de las tablas.

Implementación y funcionamiento: Pasos seguidos para la realización del proyecto y muestra de resultados.

Proceso de instalación: Pasos a seguir para la instalación de la aplicación.

Conclusiones: Conclusiones acerca del resultado del proyecto y futuro del mismo.

Glosario: Catalogo de expresiones con la finalidad de facilitar la comprensión del proyecto.

2. Análisis

Evaluación de riesgos y acciones:

En la siguiente tabla se muestran los potenciales riesgos para el desarrollo y buen funcionamiento del proyecto.

2.1 Tabla de riesgos:

En la tabla 1 están listados los principales riesgos junto con la probabilidad e impacto que tendrían sobre el proyecto.

	Riesgo	Probabilidad	Impacto
1	Cambios en las estructuras web de los diferentes ayuntamientos	BAJA	ALTO
2	Problemas de comunicación entre las diferentes tecnologías para la realización del proyecto.	BAJA	MEDIO
3	Limitación Temporal para la realización del proyecto	MEDIA	ALTO
4	Diversidad de estructuras web de los diferentes ayuntamientos	ALTA	ALTA
5	Fallos de programación que pueden afectar al aplicativo en su totalidad	MEDIA	ALTA

Tabla 1 – Evaluación de riesgos

2.2 Tabla de acciones para mitigar los riesgos:

En la Tabla 2 se representan las acciones a llevar a cabo para mitigar cada uno de los riesgos mostrados en la tabla anterior respectivamente.

	Acciones
1	Revisar periódicamente que el aplicativo sigue funcionando correctamente y realizar los cambios oportunos en caso de alguna modificación en la estructura web.
2	Destinar el tiempo adecuado a cada tecnología de manera separada y tras comprobar su correcto funcionamiento iniciar el proceso de combinación de estas.
3	Seguir rigurosamente las fechas previstas para cada acción y dar prioridad a las funciones principales del aplicativo.
4	Estudiar la estructura concreta de cada uno de los diferentes portales web para asegurar que la obtención de los datos se puede realizar correctamente.
5	Realizar copias de respaldo periódicamente y separar por segmentos cada parte del proceso creando de esta manera un "cortafuegos"

Tabla 2 – Acciones para la mitigar los riesgos

2.3 Requisitos del proyecto e implementación:

En la tabla 3 y en la tabla 4 se detallan los recursos necesarios tanto del servidor como del cliente respectivamente para el correcto funcionamiento de la aplicación web junto con las versiones que serán utilizadas para la realización del proyecto.

Servidor	
Apache	Versión 2.4.54
PHP	Versión 8.1.10
MySQL	Versión 8.0
Dominio	https://www.sergiofabregat.online
Web Hosting	GoDaddy
Servidor Local	Xampp
phpMyAdmin	Versión 5.2.0
Python	Versión 3.11.0

Tabla 3 – Requisitos del proyecto: Servidor

Cliente	
Software	Navegador Web (Google Chrome, Firefox, Safari, Microsoft Edge...)
Hardware	Ordenador, Móvil o Tablet.

Tabla 4 – Requisitos del proyecto: Cliente

2.4 Grupos de interés (Stakeholders)

En este apartado podemos distinguir las partes interesadas en el uso de la aplicación:

Responsables de comunicación de los ayuntamientos

Son los encargados de hacer llegar la información desde el ayuntamiento hacia al ciudadano. Muchas veces se trata de una tarea difícil ya que no cuentan con la infraestructura necesaria y tienen que acudir a servicios de terceros como periódicos y publicidad institucional.

Usuario

Se trata del usuario principal de la aplicación, este busca estar informado de las noticias y eventos que publican los diferentes ayuntamientos.

2.5 Requisitos funcionales

En la tabla 5 se listan los requisitos funcionales que tendrá la aplicación y la importancia que tienen los mismos dentro del proyecto destacando cuales de ellos son obligatorios (*must have*) y cuales opcionales (*nice to have*).

Requisitos funcionales	Importancia
Un usuario podrá registrarse en la plataforma y seguir al número de ayuntamientos que desee	OBLIGATORIO
Las noticias publicadas por los ayuntamientos deben registrarse correctamente de manera automatizada en la base de datos	OBLIGATORIO
Las notificaciones serán mostradas al usuario ordenadas por fecha, de manera descendente	OBLIGATORIO
Al usuario solo le aparecerán las notificaciones de los ayuntamientos que haya seleccionado	OBLIGATORIO
El administrador tendrá acceso a un panel de gestión donde podrá administrar los usuarios con opción de eliminarlos	OPCIONAL

Tabla 5 – Requisitos funcionales de la aplicación

2.6 Roles

En el aplicativo web se pueden diferenciar tres roles diferentes:

- **Administrador:** Es un rol destinado a solventar cualquier incidencia el cual tiene una serie de privilegios de administración y gestión de usuarios. Este rol tiene acceso al panel de gestión.
- **Cliente:** Es el rol principal que adquiere cualquier usuario en el momento del registro. Este usuario tendrá acceso a su panel de control de usuario y al portal principal donde aparecerán las últimas notificaciones de los ayuntamientos que este siguiendo.
- **Invitado:** Se trata de un usuario que todavía no tiene cuenta creada. Este tiene el acceso limitado ya que es necesaria una cuenta para poder tener acceso al portal por lo que tendrá acceso solo al panel de registro, login y a la página principal de invitado.

2.7 Flujo de navegación

En la Figura 2 está representado el diagrama de flujos que muestra la estructura del entorno.

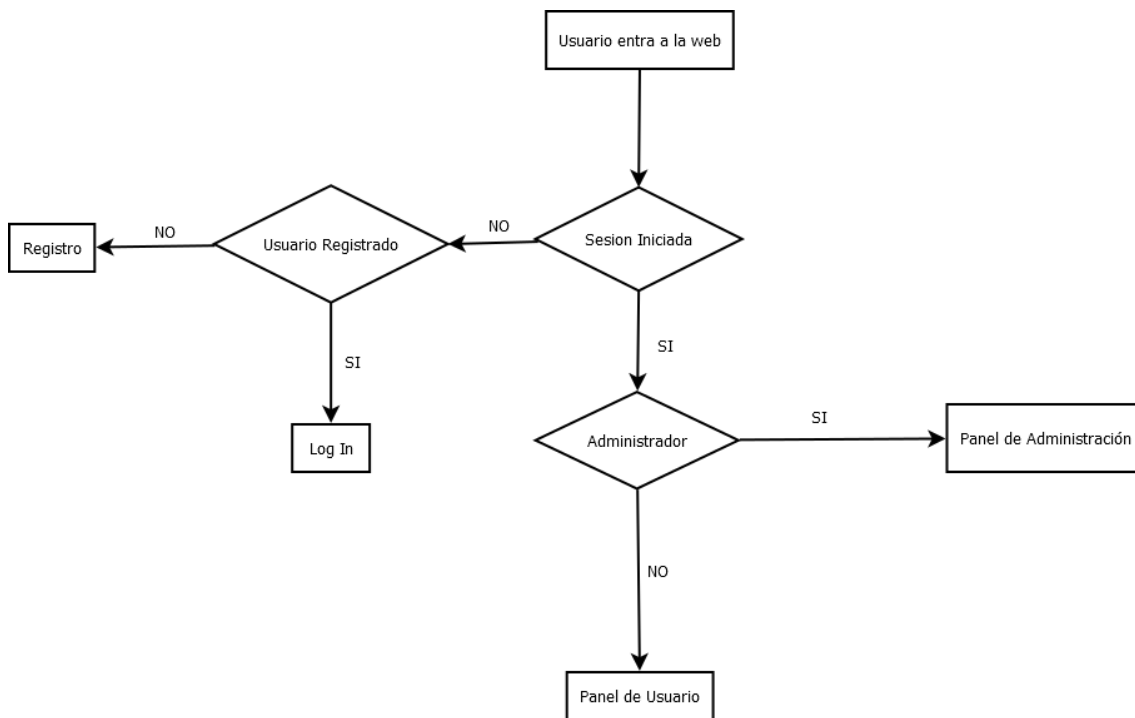


Figura 2 - Diagrama de Flujos

Como se puede apreciar una vez el usuario entra en el aplicativo web, este analizará si el usuario tiene una sesión activa o no. Tras realizar esta comprobación en el caso de que no la tenga, redirige al usuario al panel de login o registro dependiendo si este necesita crear una cuenta o es poseedor de una.

Una vez realizado el log in y con una sesión iniciada, el aplicativo verifica si se trata de un usuario común o de un usuario con privilegios de administrador. En el caso de que se trate de un usuario común, este lo redirigirá al panel de usuario donde podrá consultar las últimas novedades de los ayuntamientos que tenga marcados para el seguimiento.

En el caso de que se trate de un administrador, este accederá al panel de administración donde tendrá el acceso a la administración y gestión de los usuarios.

3. Diseño

Modelo de plantillas (prototipo)

Para enfocar el esquema de las páginas se presentan una serie de *wireframes* como guía visual representativa de cada página incluyendo interfaces y sistemas de navegación.

Separaremos los *wireframes* en dos grupos, los de baja definición donde se muestra el esqueleto básico de cada página y de alta definición que serán una réplica semejante al producto final.

3.1 Wireframes

Página de aterrizaje (Landing page)

En la Figura 3 se muestra el *wireframe* de página que se muestra al usuario nada más entrar en la web.

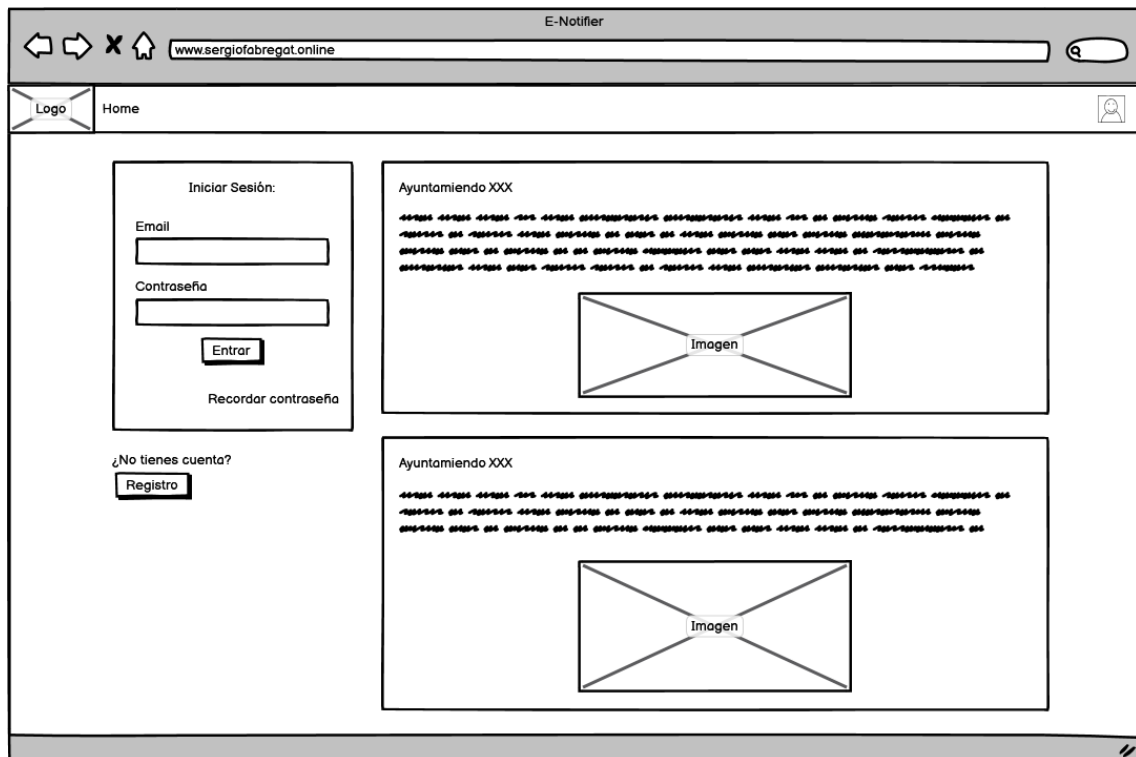


Figura 3 – Wireframe Landing page

Página Principal de Usuario (Home Page)

En la figura 4 se muestra la página de Inicio o Home de un usuario registrado donde puede apreciar tanto sus datos de usuario como las ultimas notificaciones de los ayuntamientos que sigue.

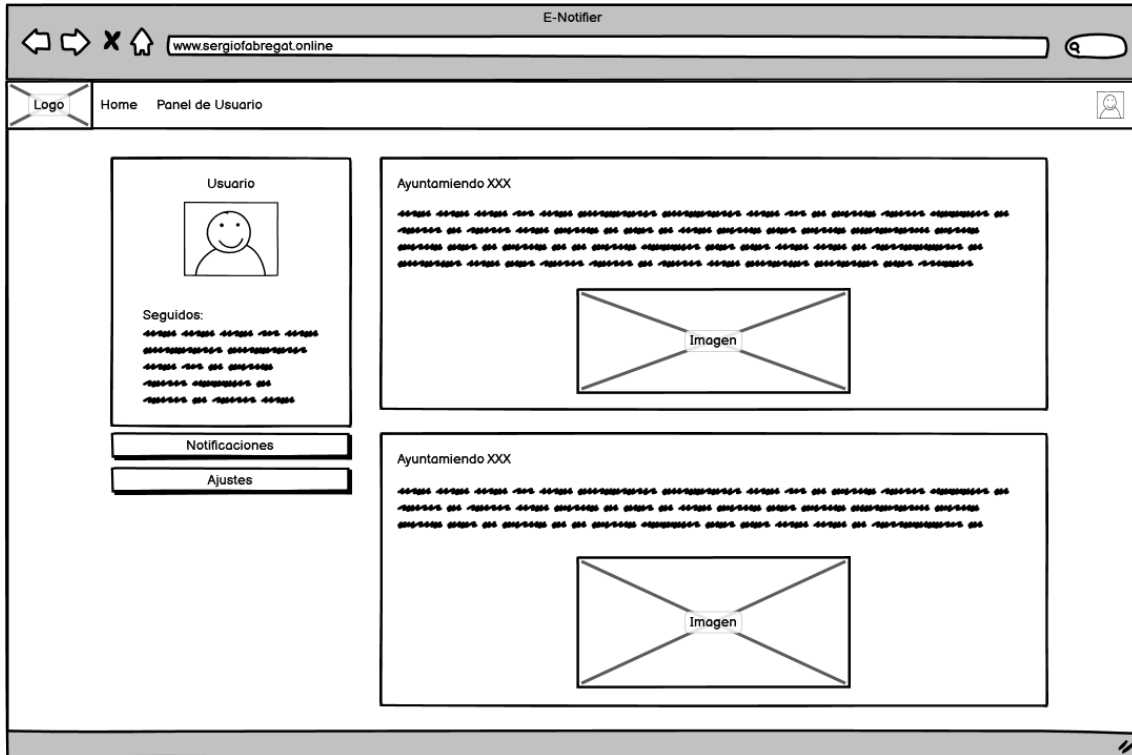


Figura 4 – Wireframe User Homepage

Página Principal de Administrador

En la Figura 5 está representada la página Inicial o Home que ve un usuario con el rol de administrador.

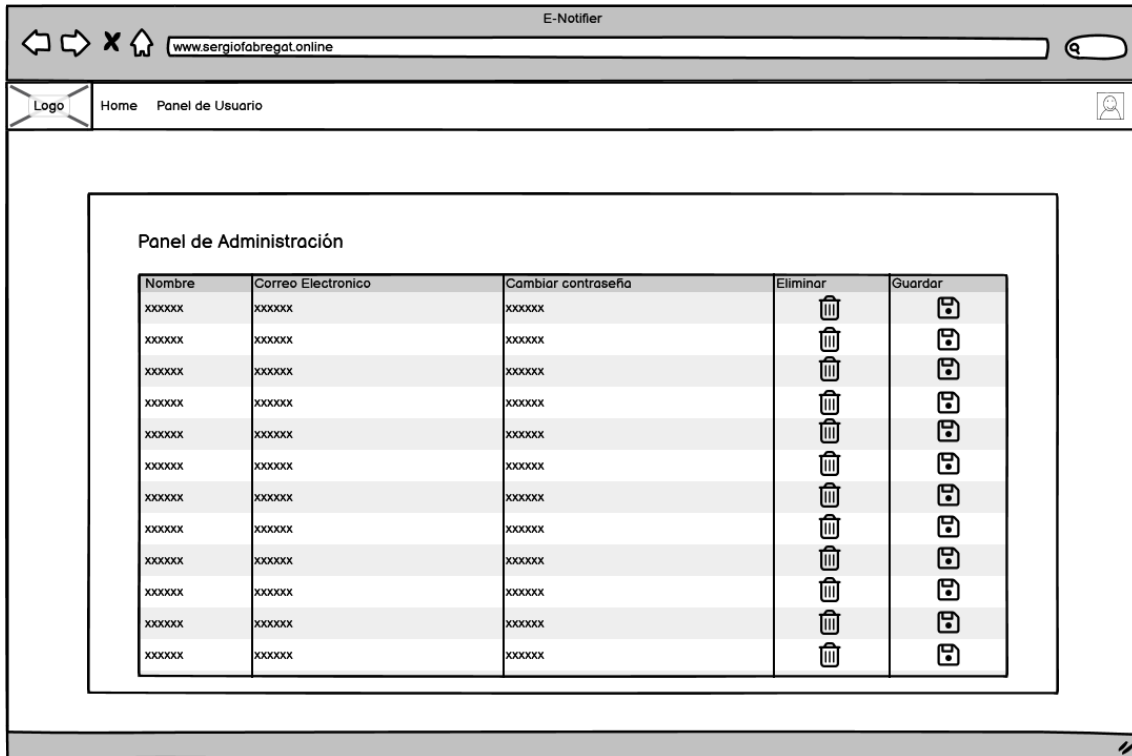


Figura 5 – Wireframe Admin Homepage

Página de Login

En la Figura 6 se muestra un *wireframe* de la página de login necesaria para que el usuario inicie la sesión y pueda acceder a su cuenta.

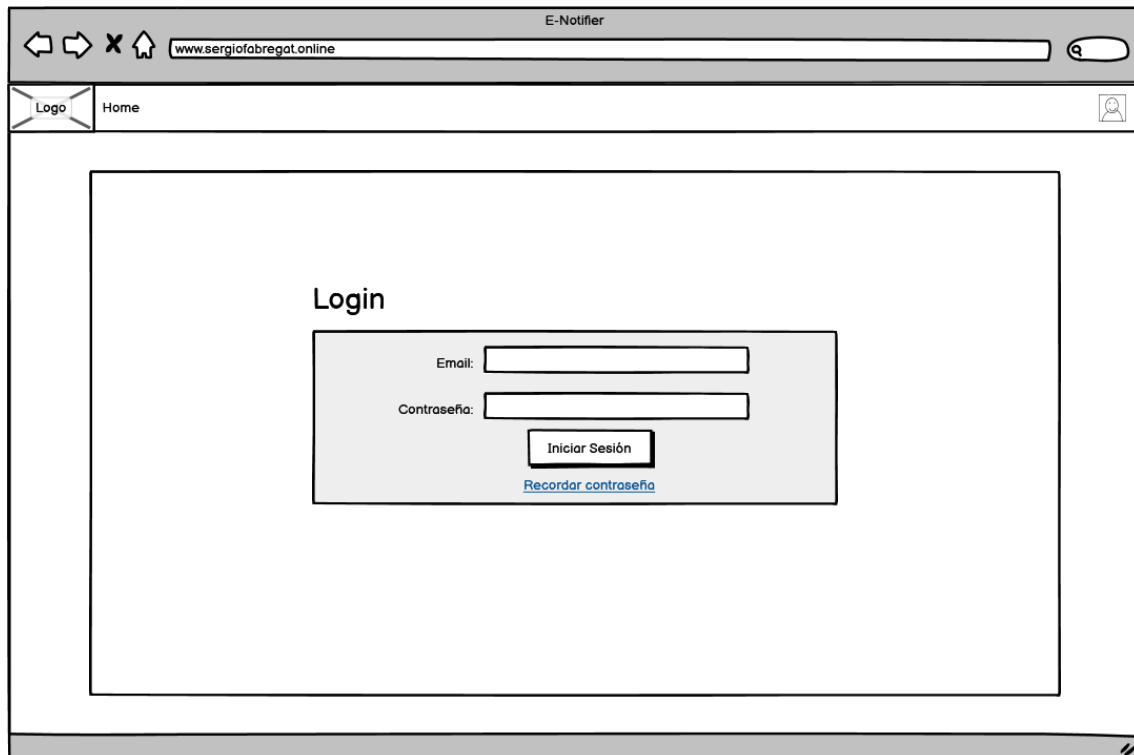


Figura 6 – Wireframe página de Login

Página de Ajustes

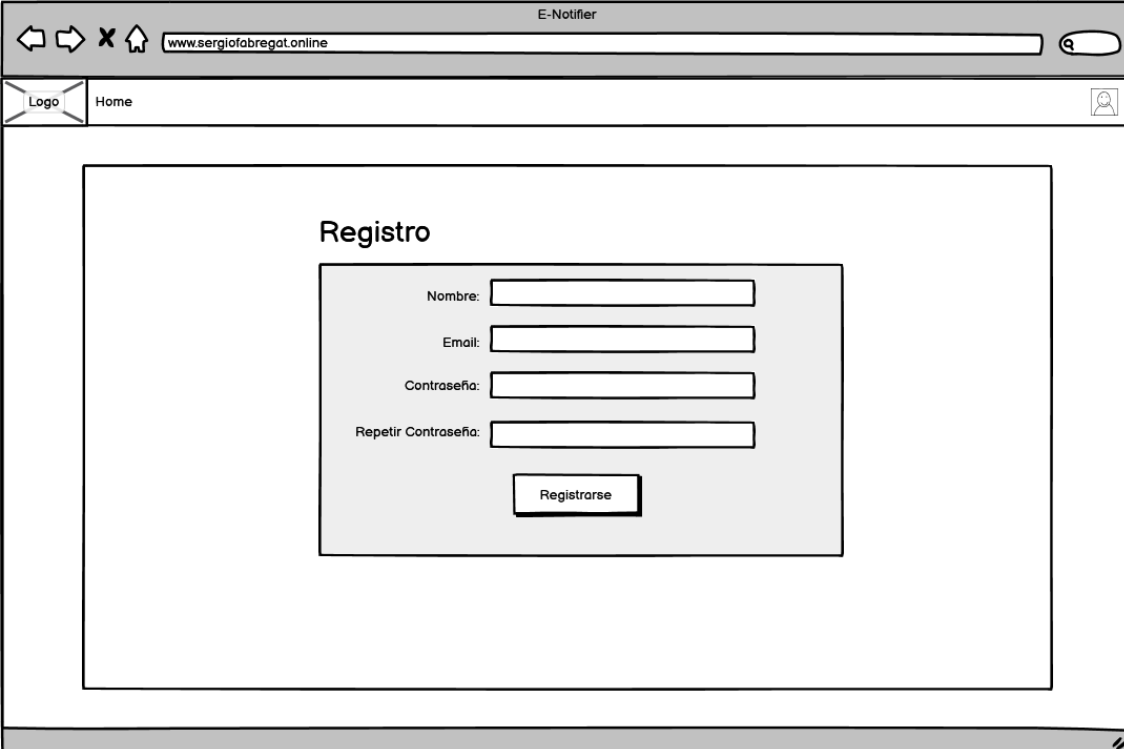
En la Figura 7 se muestra la página de ajustes donde el usuario puede cambiar sus datos o bien escoger los ayuntamientos que desea seguir.

The image shows a wireframe of a web browser window. The browser's address bar contains the URL "www.sergiofabregat.online" and the text "E-Notifier". The browser's navigation buttons (back, forward, stop, refresh, home) are visible on the left. The page header includes a "Logo" button, "Home", and "Panel de Usuario" links, along with a user profile icon on the right. The main content area is enclosed in a large rectangular frame. On the left side of this frame, there is a placeholder for a user profile picture with a "Cambiar imagen" button below it. To the right of the profile picture is the "Datos de Usuario" section, which contains three input fields labeled "Nombre:", "Email:", and "Password:". Below this section is the "Listado de Ayuntamientos" section, featuring a list of five items, each with a checkbox and the text "Ayuntamiento XXXX". The first two checkboxes are checked, while the last three are unchecked. A "Guardar" button is positioned at the bottom right of the main content area.

Figura 7 – Wireframe página de Ajustes

Página de Registro

En la Figura 8 está representada la página para el registro de un nuevo usuario necesaria para poderse dar de alta en la plataforma y poder obtener una cuenta de usuario.



The image shows a wireframe of a web browser window. The browser's address bar contains the URL "www.sergiofabregat.online" and the page title is "E-Notifier". The browser's navigation bar includes a "Logo" placeholder and a "Home" link. The main content area features a registration form titled "Registro". The form is contained within a light gray box and includes the following fields and elements:

- Nombre:
- Email:
- Contraseña:
- Repetir Contraseña:
- Registrarse

Figura 8 – Wireframe página de Registro

3.2 Wireframe alta definición (prototipo)

En la Figura 9 podemos observar un prototipo de alta definición de la página principal de usuario.

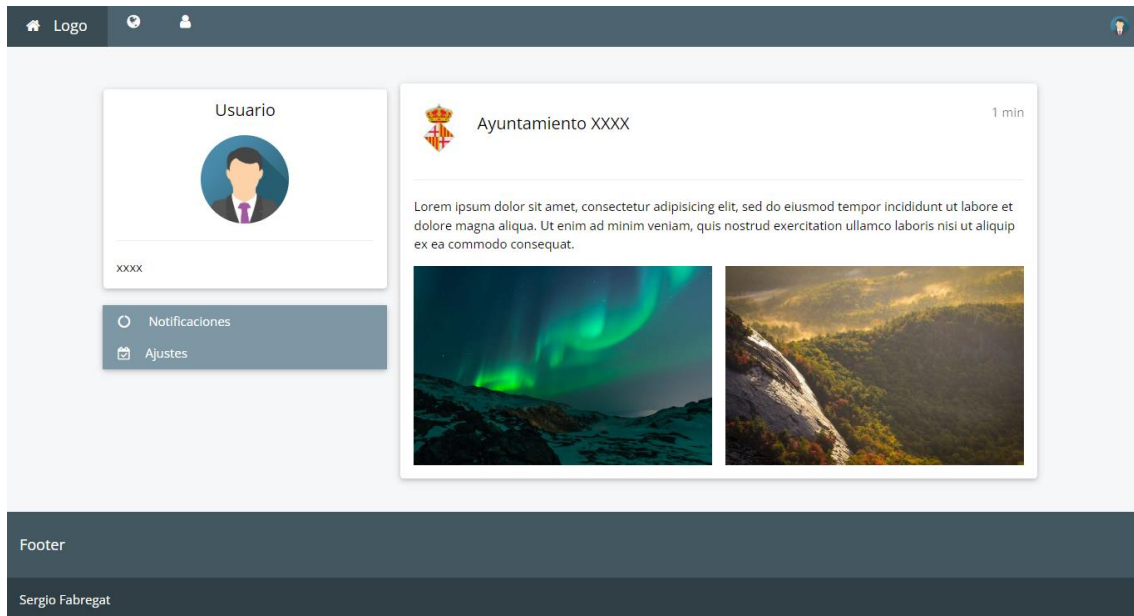


Figura 9 – Wireframe alta definición Homepage

3.3 Bases de datos

En el siguiente apartado se mostrará el diseño de la base de datos mediante un modelo entidad relación y el diseño relacional de la base de datos.

3.3.1 Modelo entidad relación

En la figura 10 se muestra el diseño entidad-relación de la base de datos.

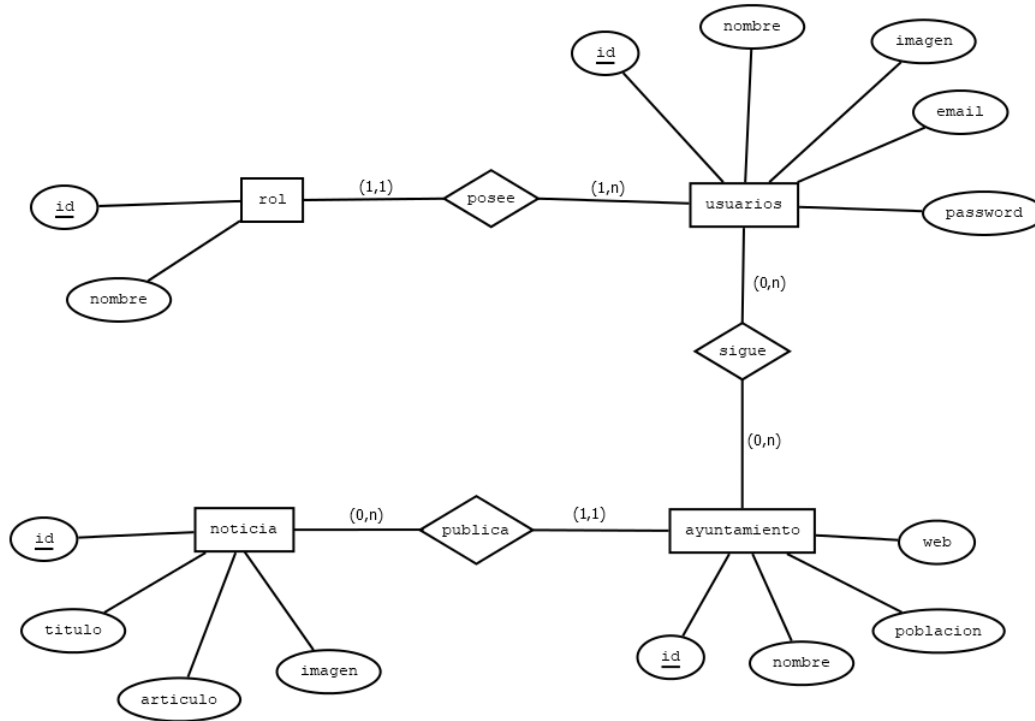


Figura 10 – Modelo entidad-relación de la base de datos

3.3.2 Diseño relacional de la base de datos

En la siguiente figura se muestra un diagrama del sistema de tablas de la base de datos con sus respectivos títulos y atributos además de las relaciones entre las mismas.

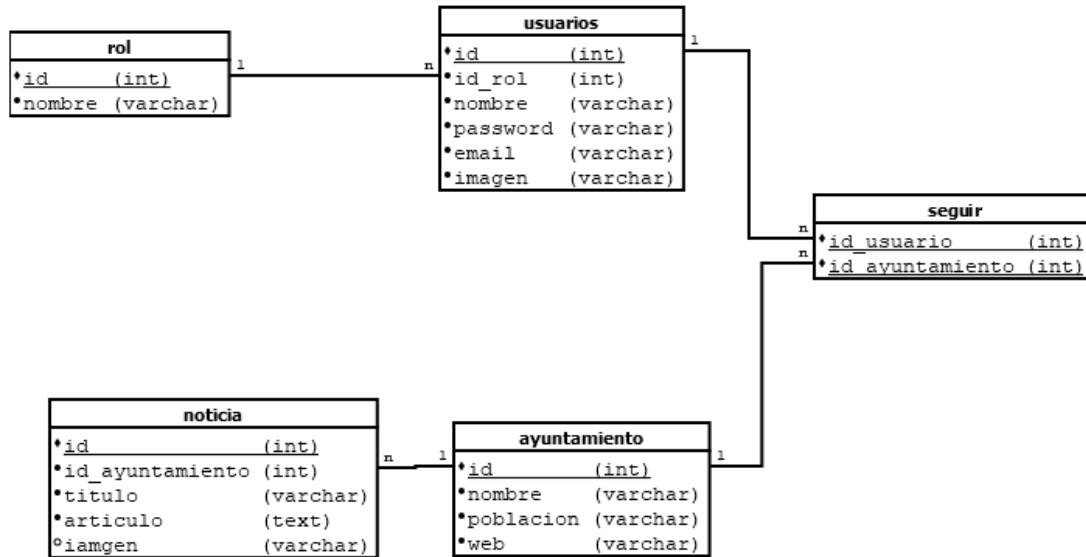


Figura 11 – Modelo relacional de la base de datos

4. Implementación y funcionamiento

En este apartado se mostrarán los diferentes resultados finales tras la implementación del proyecto y los usos de este. [8], [9], [10]

Página de aterrizaje (Landing page)

Esta pagina es la que se encuentra el usuario al entrar en la web cuando no tiene una sesión iniciada. Hasta que no inicie la sesión, este no podrá ver ninguna notificación.

El sistema sabe si el usuario tiene una sesión iniciada dependiendo de la variable `$_SESSION`. En el caso que la variable contenga datos verifica si se trata de un usuario común, donde se le mostrará la página principal de usuario o de un administrador que tendrá acceso al menú de administración.

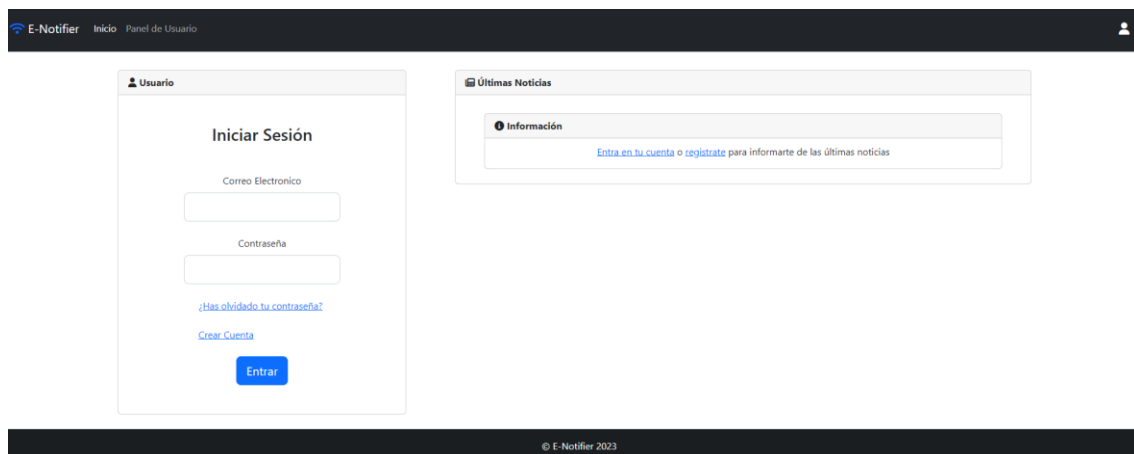


Figura 12 – Página de aterrizaje de un usuario sin sesión.

```
<?php
include 'php/bd_con.php';
include 'templates/header.php';

if(!isset($_SESSION['email'])){
    include 'templates/no-user.php';
}else{

    if($_SESSION['id_rol'] == 1){
        include 'templates/user-page.php';
    }

    if($_SESSION['id_rol'] == 2){
        include 'templates/admin-page.php';
    }
}

include 'templates/footer.php';
?>
```

Figura 13 – Código fuente de la página de aterrizaje encargado de verificar el usuario.

Página de Registro

Diseño final del apartado de registro. Se trata del primer paso para que un usuario pueda obtener el acceso al uso de la aplicación y a la vez necesario para saber las preferencias del usuario en cuanto a las notificaciones que mostrar. Las contraseñas como medida de seguridad estas guardadas en la base de datos bajo el algoritmo de reducción criptográfico MD5.[5]

Usuario

Registro

Nombre

Correo Electronico

Contraseña

Repetir Contraseña

Crear cuenta

© E-Notifier 2023

Figura 14 – Versión final de la página de registro de usuario.

Página de Log In

Diseño final del apartado de Log In donde el usuario realiza el inicio de sesión y el servidor otorga la variable de sesión dependiendo si coinciden los datos aportados con los presentes en la base de datos.

Usuario

Iniciar Sesión

Correo Electronico

Contraseña

[¿Has olvidado tu contraseña?](#)

[Crear Cuenta](#)

Entrar

© E-Notifier 2023

Figura 15 - Versión final de la página de inicio de sesión de usuario.

Página Principal de Usuario (Home Page)

Se trata del núcleo de la aplicación, esta es la pantalla que ve el usuario al iniciar sesión.

En ella se muestran las notificaciones que el usuario ha marcado en la sección de ajustes. Estas notificaciones están ordenadas según han sido publicadas independientemente del origen.

Las notificaciones están almacenadas en la base de datos mediante un script programado en Python que lee y almacena las mismas.

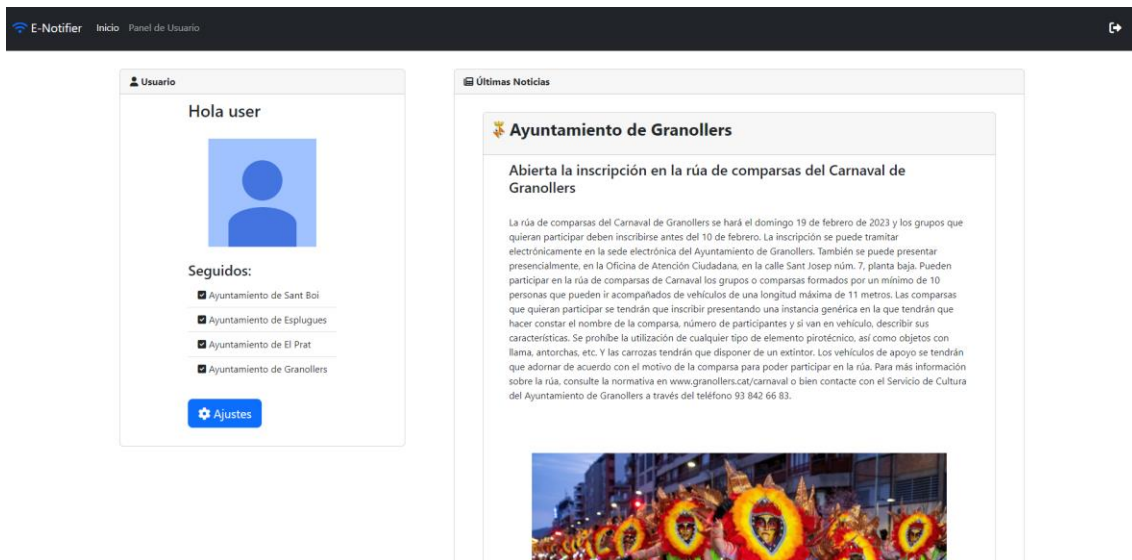


Figura 16 – Página principal de usuario donde se muestran las notificaciones.

Página de Ajustes

Se trata de la página de ajustes del usuario donde se pueden escoger las entidades de las que se quiere recibir notificación. También el usuario tiene acceso a un apartado para modificar sus datos.

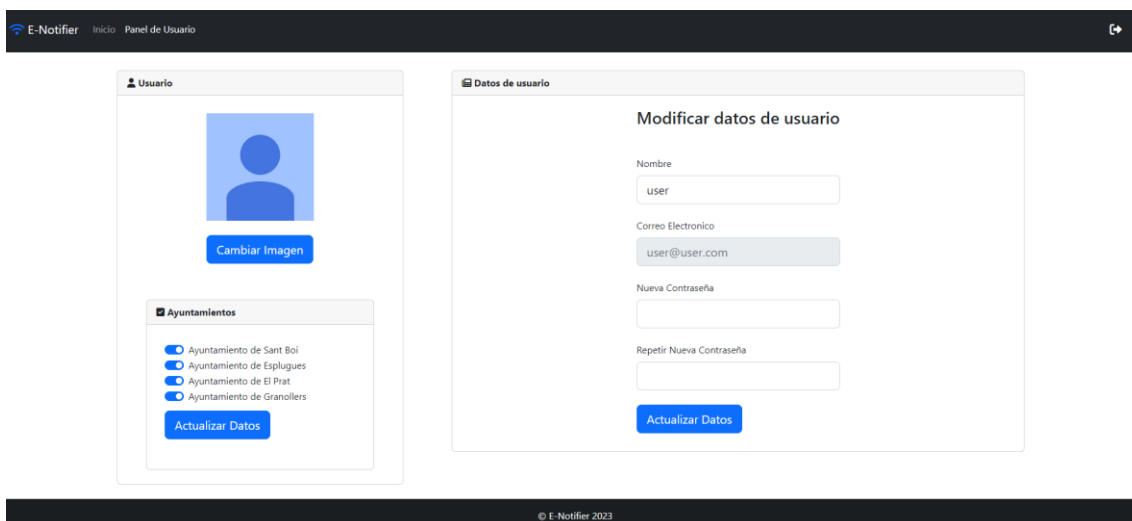
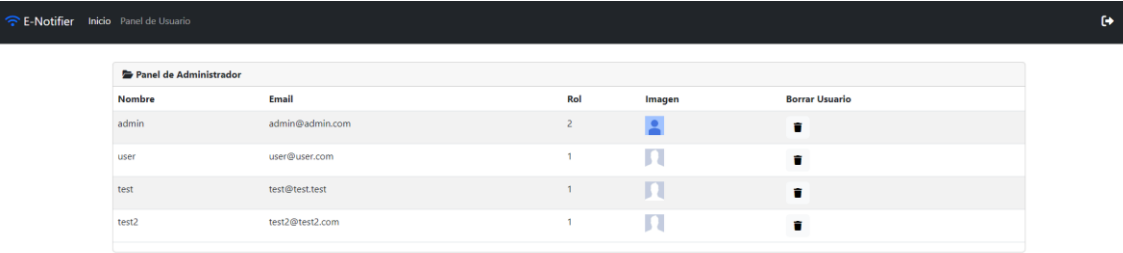










Figura 17 – Página de la sección de ajustes de usuario.

Página Principal de Administrador

Este es el panel de administración al que tiene acceso el usuario con rol administrador donde puede ver datos de los usuarios como el nombre, email, rol e imagen. También tiene acceso a la función de eliminar usuario.



The screenshot shows the 'Panel de Administrador' interface. At the top, there is a navigation bar with 'E-Notifier Inicio Panel de Usuario'. Below it is a table with the following data:

Nombre	Email	Rol	Imagen	Borrar Usuario
admin	admin@admin.com	2		
user	user@user.com	1		
test	test@test.test	1		
test2	test2@test2.com	1		

At the bottom of the screenshot, there is a footer with the text '© E-Notifier 2023'.

Figura 18 – Panel de administración (Página principal de usuario con rol administrador)

Diseño Adaptable

La filosofía de diseño para el proyecto es simple y adaptable. De tal manera que el ancho de la pantalla condicione como se muestran los elementos ajustándose a la resolución necesaria en cada caso.

De esta manera se consigue que el aplicativo se pueda utilizar correctamente en múltiples dispositivos. Esto se conoce como un diseño “responsive”. [12]

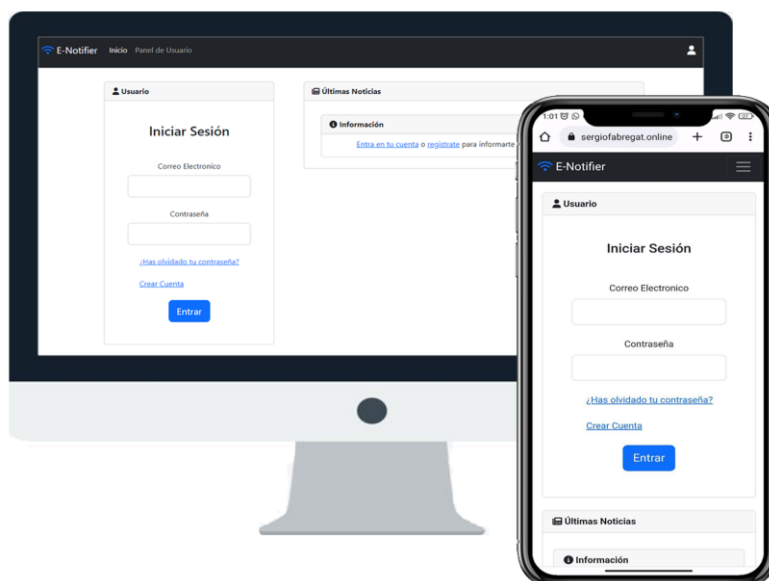


Figura 19 – Ejemplo diseño “responsive”

4.1 Funcionamiento del Script

En este apartado se expone el funcionamiento interno del script utilizado para rellenar automáticamente la base de datos con las notificaciones aportadas por las distintas entidades.

Extracción de datos

Para la extracción automática de datos y posterior inserción en la base de datos se utiliza un script programado en el lenguaje Python.

Para ello utilizan principalmente estas librerías:

- **Requests:** Se trata de una biblioteca HTTP cuyo objetivo es hacer que las solicitudes HTTP sean más simples y amigables.
- **Beautiful Soap 4:** Biblioteca de Python para analizar documentos HTML. Su funcionamiento consiste en crear un árbol con todos los elementos del documento.
- **Googletrans:** Es una biblioteca que implementó la API de Google Translate. Utiliza la API Ajax de Google Translate para realizar llamadas a métodos.
- **Pymysql:** Esta librería se utiliza para ejecutar una interacción con las bases de datos MySQL.

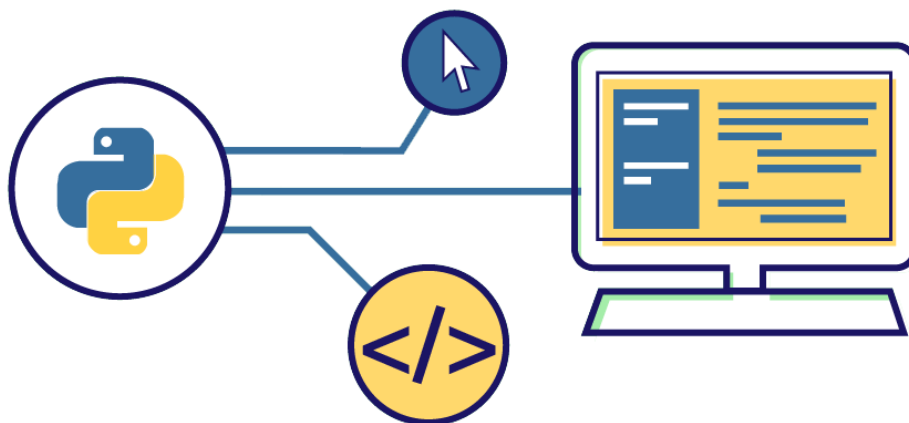


Figura 20 – Representación gráfica de la interacción de Python con la Base de Datos.

4.2 Ejecución del código

Para el funcionamiento del script es necesario inicialmente importar las librerías.

```
from bs4 import BeautifulSoup

import requests

from googletrans import Translator

import pymysql
```

Figura 21 – Importación de las librerías necesarias para la ejecución del código.

El script hace una llamada mediante la librería requests a la dirección que aporta el ayuntamiento con un listado ordenado por fecha de las últimas notificaciones y mediante Beautiful Soap 4 se consulta cual es la última notificación aportada.

```
# get last

website_last = 'https://www.elprat.cat/actualitat/noticies'

result_last = requests.get(website_last)

content_last = result_last.text

soup_last = BeautifulSoup(content_last, 'lxml')

link_list_last = soup_last.find('h3', class_="noticia_titol")

archivo_last = link_list_last.find('a')['href']
```

Figura 22 – Ejemplo de obtención de la última notificación.

Tras esto, se revisa un documento que contiene la dirección exacta de la última notificación que se publicó y posteriormente se compara con la nueva dirección obtenida previamente.

```
with open("last.txt") as f:

    verult = f.read()

f.close()
```

Figura 23 – Revisión del documento que contiene la última notificación.

En el caso de que no sean iguales se da luz verde para seguir con el proceso de inserción en la base de datos, en caso contrario, no se ha publicado una notificación nueva por lo que el proceso termina.

```
> if verult != archivo_last: ...  
  
    else:  
        print('No hay noticias nuevas')
```

Figura 24 – Comprobación de nueva notificación.

Una vez realizada la comprobación, se realiza otra petición al servidor mediante requests referente a la dirección de la nueva notificación.

```
# How To Get The HTML  
  
result = requests.get(archivo_last)  
  
content = result.text  
  
soup = BeautifulSoup(content, 'lxml')
```

Figura 25 – Solicitud HTTP de la nueva dirección obtenida.

Una vez obtenidos los datos y mediante un análisis de clases e identificadores se separa el título, el artículo y la url de la imagen mediante el uso de BeautifulSoup 4.

Cada entidad está estructurada de una manera diferente por lo que el código a ejecutar no es genérico, hay que analizar cada una independientemente.

En el caso de las entidades de las que se obtiene la información analizada en este proyecto, están redactadas en catalán.

Para dar una uniformidad al proyecto y puesto que está elaborado en castellano se han traducido los textos mediante la librería que utiliza la API de Google Translate (googletrans).

```
# Titulo  
  
box = soup.find('h1', class_='title-slim')  
  
box = ''.join(box)  
  
traducido = translator.translate(box, dest='es')  
  
box = traducido.text
```

Figura 26 – Ejemplo de obtención de título y posterior traducción.

Una vez obtenida toda la información se procede a ejecutar el volcado de los datos a la BBDD. Este proceso se ejecuta mediante la librería pymysql.

```
#database connection
connection = pymysql.connect(host="****",user="****",passwd="****",database="****")
cursor = connection.cursor()

id="3"

# queries for inserting values
insert = 'INSERT INTO noticia(id_ayuntamiento, titulo, articulo, imagen) VALUES('+str(id)+' , ''+title+', ''+str1+', ''+imagen+'');'

#executing the quires
cursor.execute(insert)

#commiting the connection then closing it.
connection.commit()
connection.close()
```

Figura 27 – Ejemplo de inserción de datos en la Base de Datos mediante la librería PyMysql.

Para finalizar el proceso, se actualiza el documento que recoge la dirección de la última noticia con la nueva dirección obtenida.

```
file = open("last.txt", 'w')

file.write(archivo_last)

file.close()

print('Noticia publicada')
```

Figura 28 – Ejemplo de escritura en el documento de ultima notificación con la dirección obtenida.

Si todos los pasos se han realizado correctamente la consola mostrará un mensaje conforme la noticia se ha publicado.

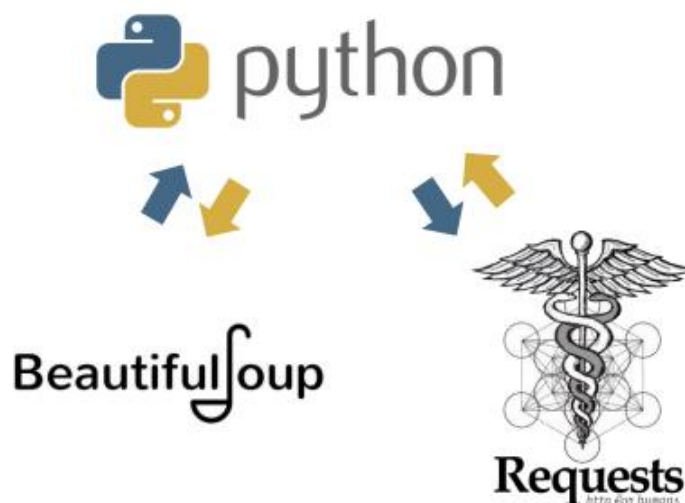


Figura 29 – Python trabajando en conjunto con Bs4 y Requests.

En el caso del aplicativo subido en la nube funciona mediante trabajos cron:

En este caso el programa se ejecuta 2 veces al día a las 0h y 12h ya que estamos en un entorno de pruebas.



WEB APPLICATIONS CREATE APPLICATION

App URI ↑	App Root Directory	Status	Actions
sergiofabregat.online/santbol_data	/home/nxd060c1m06e/santbol_data	● started (v3.7.12)	■ ↺ ✎ 🗑
sergiofabregat.online/granollers_data	/home/nxd060c1m06e/granollers_data	● started (v3.7.12)	■ ↺ ✎ 🗑
sergiofabregat.online/esplugues_data	/home/nxd060c1m06e/esplugues_data	● started (v3.7.12)	■ ↺ ✎ 🗑
sergiofabregat.online/elprat_data	/home/nxd060c1m06e/elprat_data	● started (v3.7.12)	■ ↺ ✎ 🗑

Figura 30 – Listado de aplicaciones Python en el servidor donde esta subida la aplicación en la nube.

Trabajos de cron actuales

Minuto	Hora	Día	Mes	Día de la semana	Comando	Acciones
0	0,12	*	*	*	<code>wget -q "https://www.sergiofabregat.online/esplugues_data"</code>	✎ Eliminar
0	0,12	*	*	*	<code>wget -q "https://www.sergiofabregat.online/elprat_data"</code>	✎ Eliminar
0	0,12	*	*	*	<code>wget -q "https://www.sergiofabregat.online/granollers_data"</code>	✎ Eliminar
0	0,12	*	*	*	<code>wget -q "https://www.sergiofabregat.online/santbol_data"</code>	✎ Eliminar

Figura 31 – Listado de trabajos cron.

Para poder obtener todas las noticias sería conveniente ejecutar el script con un menor intervalo de tiempo, por ejemplo cada 15 minutos durante todo el día, ya que el script solo obtiene la última noticia, que utiliza para compararla con la última registrada en la base de datos y en el caso de ser nueva añadirla.

Este es un dato relevante ya que si el intervalo de ejecución es cada mucho tiempo un ayuntamiento podría añadir dos noticias/notificaciones en un intervalo corto de tiempo, siendo posible que una de ellas (la penúltima) no quedara registrada en la base de datos.

4.3 Funcionamiento del aplicativo a nivel usuario

El primer paso que debe hacer un nuevo usuario es registrarse en el aplicativo. Para ello tendrá que cumplimentar los datos solicitados en el registro y posteriormente acceder a su cuenta de usuario.

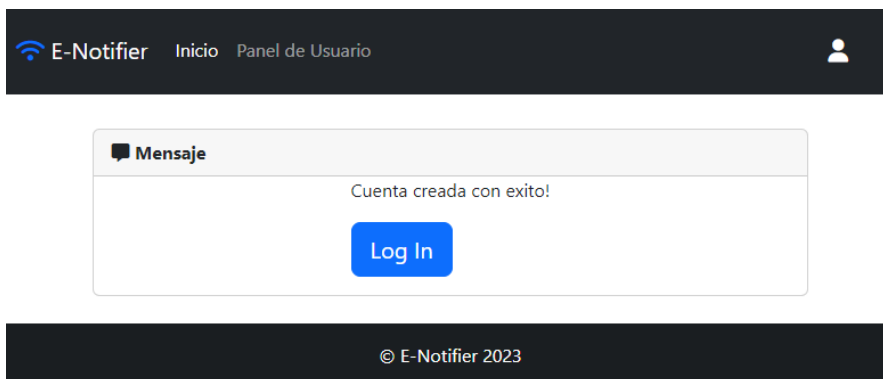


Figura 32 – Notificación de registro de nueva cuenta exitoso.

Una vez hecho esto el usuario accederá a su panel de notificaciones donde al ser la primera vez, no le aparecerá ninguna.

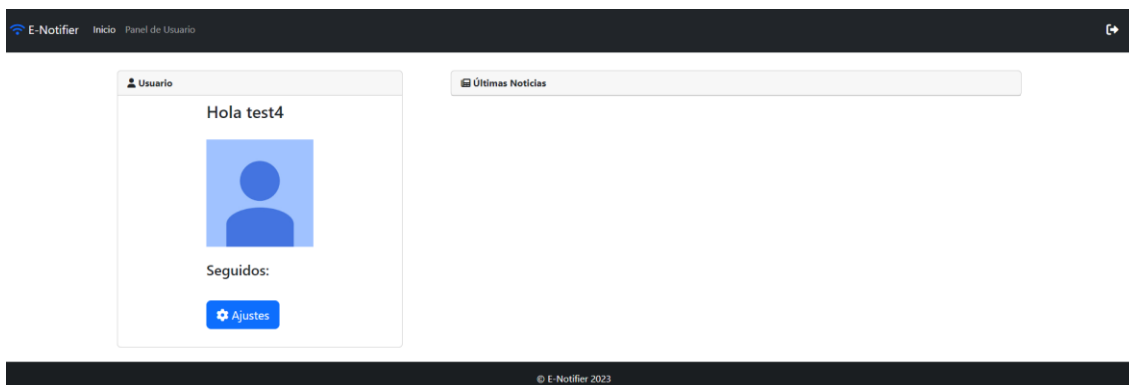


Figura 33 – Panel de notificaciones después de la creación de cuenta.

Para comenzar a seguir ayuntamientos hay que acceder al menú de ajustes y posteriormente marcar las entidades a preferencia del usuario.

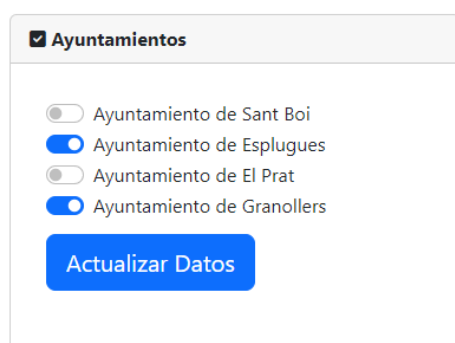


Figura 34 – Panel de selección de ayuntamientos.

Una vez realizado este ultimo paso el usuario recibe un aviso conforme los datos han sido modificados con éxito.

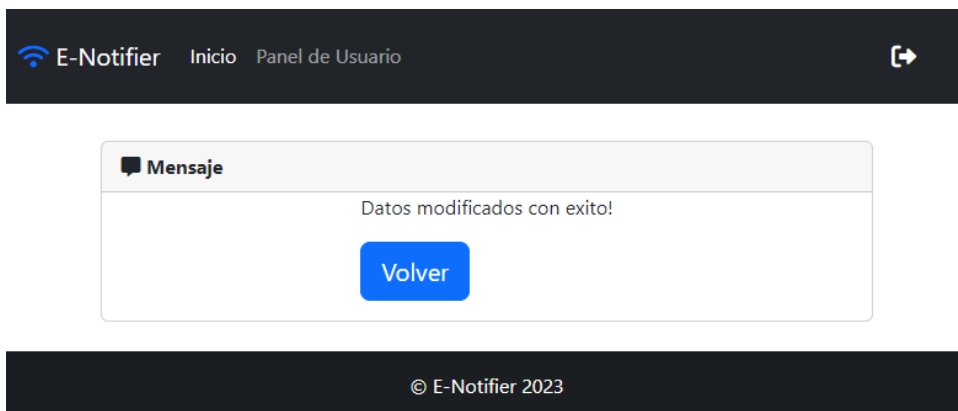


Figura 35 – Aviso de modificación exitosa de datos.

Finalmente, el usuario ya puede entrar en su panel inicial y aparecerán las notificaciones correspondientes a los ayuntamientos marcados.

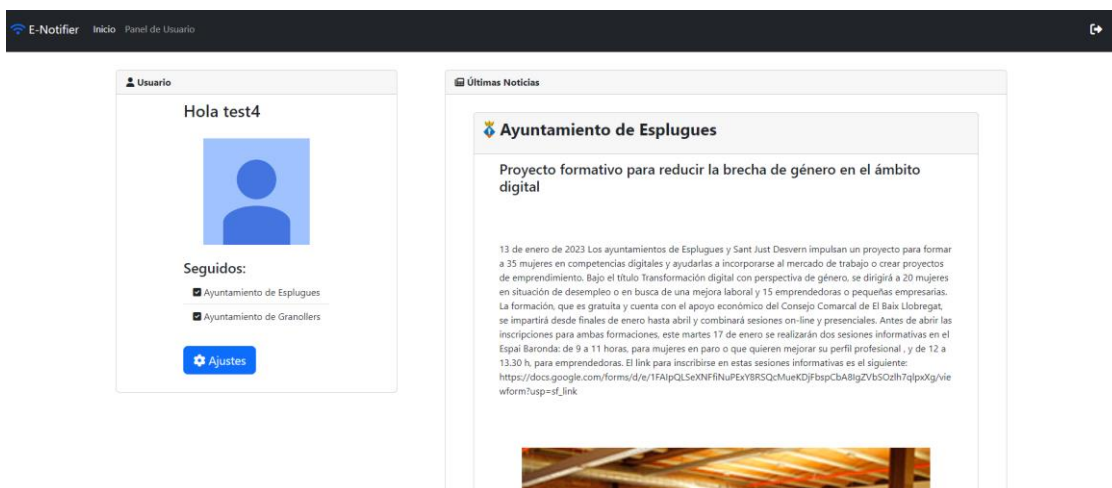


Figura 36 – Verificación del correcto funcionamiento del aplicativo en el panel inicial.

4.4 Proceso de instalación

El proceso de instalación está dividido en dos partes:

Por un lado, la guía de instalación del aplicativo web y por otro lado la instalación de Python junto con las librerías correspondientes para la obtención de los datos de manera automatizada.

Puede obtenerse el código fuente en GitHub:

<https://github.com/sfabregatr/enotifier>

Software

XAMPP(Contiene Apache + MariaDB + PHP + Perl):

<https://www.apachefriends.org/es/index.html>

Python:

<https://www.python.org/>

4.4.1 Instalación Aplicativo Web

Para realizar la instalación del aplicativo web primero realizaremos la instalación de un servidor local, en este caso se utilizará:

XAMPP (<https://www.apachefriends.org/es/index.html>).

Realizaremos la instalación por defecto del programa, en este caso se está utilizando la versión xampp-windows-x64-8.2.0-0-VS16.

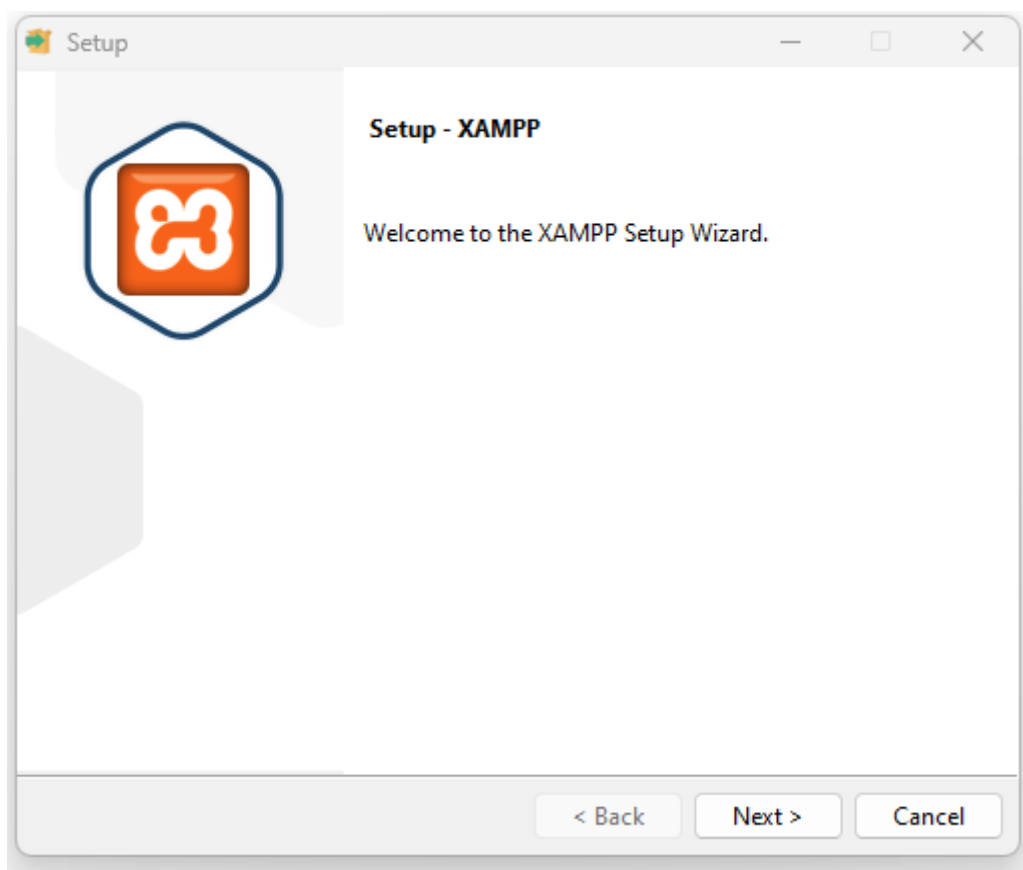


Figura 37 – Proceso de instalación XAMPP.

Una vez realizada la instalación y ejecutado el programa activaremos los módulos de Apache y MySQL.

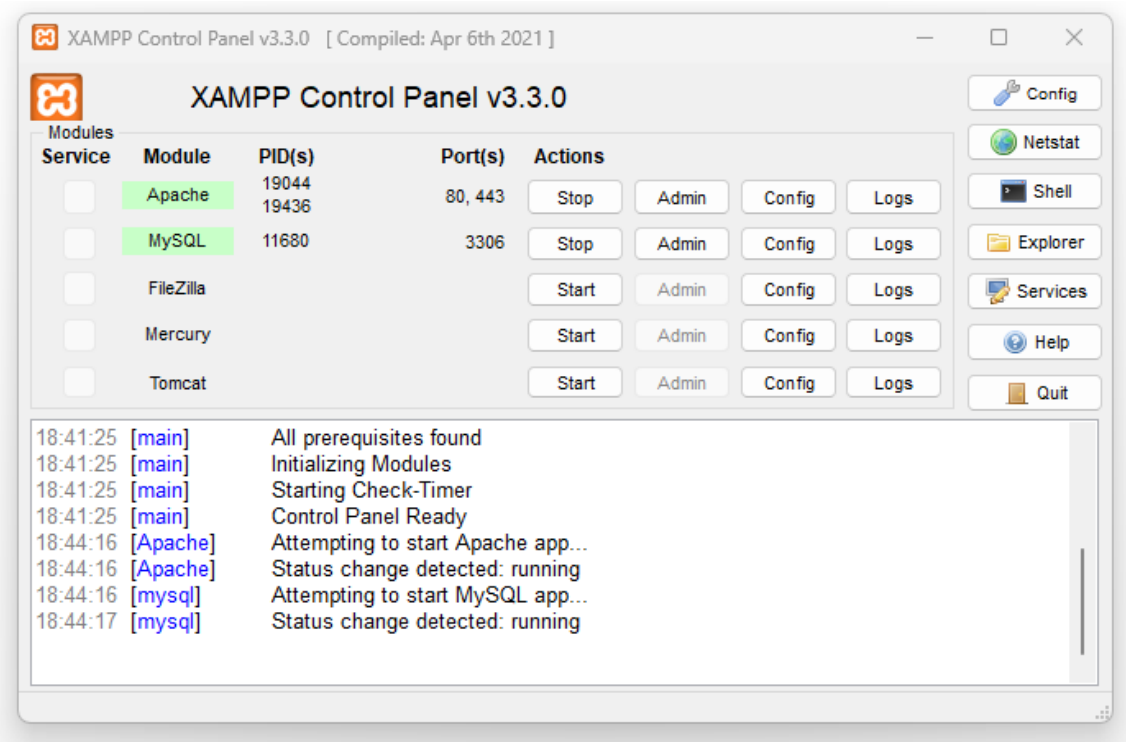


Figura 38 – Panel de control de XAMPP.

A continuación, accederemos en el explorador de archivos a la siguiente ruta:
C:\xampp\htdocs

Una vez dentro del directorio volcaremos todas las carpetas y archivos exceptuando el archivo SQL.

Tras completar este paso la carpeta **htdocs** debería tener una estructura similar a la siguiente:

Este equipo > Disco local (C:) > xampp > htdocs

Nombre	Fecha de modificación	Tipo	Tamaño
css	29/12/2022 21:31	Carpeta de archivos	
cuenta-creada	29/12/2022 21:31	Carpeta de archivos	
cuenta-update	29/12/2022 21:31	Carpeta de archivos	
elprat_data	29/12/2022 21:31	Carpeta de archivos	
error-cuenta	29/12/2022 21:31	Carpeta de archivos	
esplugues_data	29/12/2022 21:33	Carpeta de archivos	
fontawesome	29/12/2022 21:33	Carpeta de archivos	
granollers_data	29/12/2022 21:33	Carpeta de archivos	
images	29/12/2022 21:31	Carpeta de archivos	
js	29/12/2022 21:33	Carpeta de archivos	
login	29/12/2022 21:31	Carpeta de archivos	
panel-de-usuario	29/12/2022 21:31	Carpeta de archivos	
php	29/12/2022 21:31	Carpeta de archivos	
register	29/12/2022 21:31	Carpeta de archivos	
santboi_data	29/12/2022 21:31	Carpeta de archivos	
templates	29/12/2022 21:31	Carpeta de archivos	
.htaccess	29/12/2022 21:31	Archivo HTACCESS	1 KB
error_log	29/12/2022 21:31	Archivo	149 KB
favicon.ico	29/12/2022 21:31	Archivo ICO	15 KB
index.php	29/12/2022 21:31	Archivo de origen ...	1 KB

Figura 39 – Estructura de archivos y ficheros.

Llegado a este punto procederemos a la importación de la base de datos. Para ello entraremos en un explorador web (en este caso Google Chrome) y accederemos a la siguiente dirección: <http://localhost/phpmyadmin/>

A continuación, crearemos una base de datos haciendo click en “nueva” dentro del menú lateral e introduciremos en el nombre de la nueva base de datos: sfabregat_tfg



Figura 40 – Panel de phpMyAdmin.

Tras esto haremos click en crear.

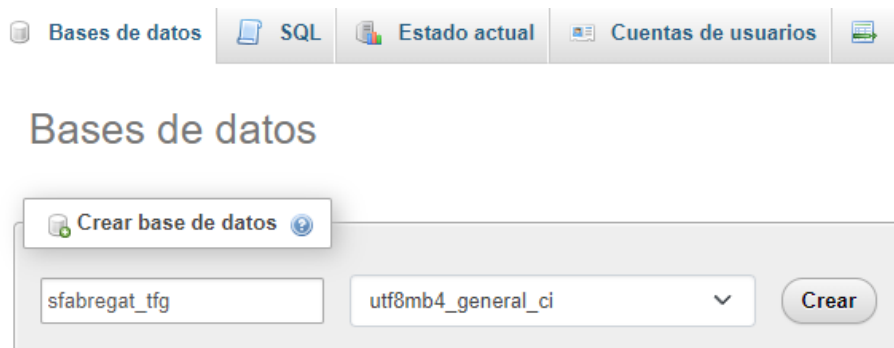


Figura 41 – Inserción del nombre de la base de datos.

Una vez creada accederemos al apartado de “Importar” que aparece en la barra superior e introduciremos el documento .sql

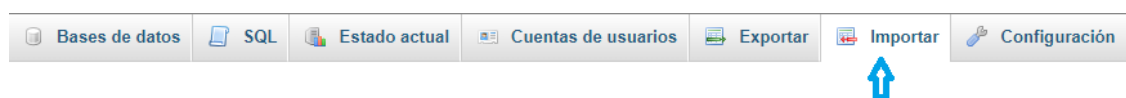


Figura 42 – Botón Importar en la barra superior.

Para finalizar la importación haremos click en el botón “Importar” que aparece en la parte inferior de la página.

Importando al servidor actual

Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2) o descomprimido.
Un archivo comprimido tiene que terminar en `.[formato].[compresión]`. Por ejemplo: `.sql.zip`

Buscar en su ordenador: (Máximo: 40MB)

sfabregat_tfg.sql

Figura 43 – Importación de la base de datos.

Si todo ha funcionado correctamente nos mostrará un mensaje que indicará:
Importación ejecutada exitosamente

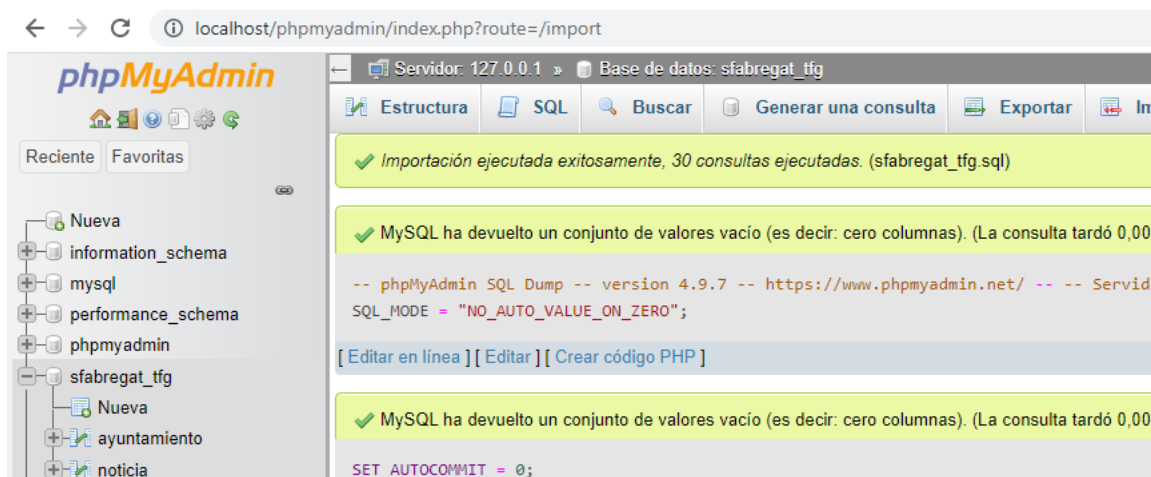


Figura 44 – Correcta importación de la base de datos.

Nota: El archivo php de configuración para la conexión con la base de datos para la prueba en local contiene los valores de acceso como root por defecto (con el nombre de usuario “root” y contraseña vacía).

Una vez hecho esto, la instalación de la aplicación web ya estaría en funcionamiento y podemos acceder mediante el siguiente directorio: <http://localhost/>

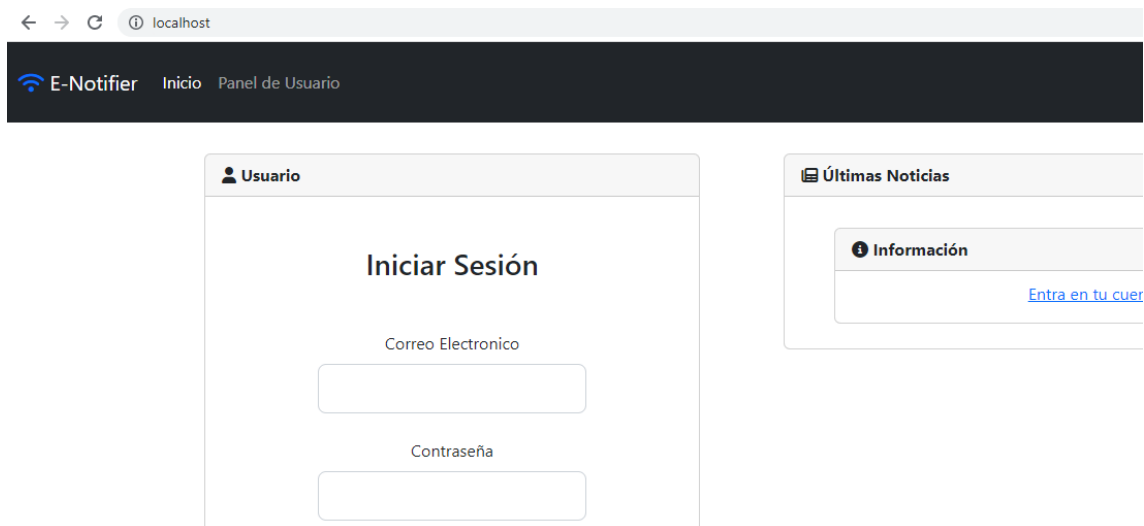


Figura 45 – Panel inicial sin sesión ejecutado en versión local.

Una vez realizado este último paso, habríamos finalizado la parte de la instalación del servidor junto con la base de datos, donde ya podríamos observar entrando con la cuenta de usuario de testeo el funcionamiento de la aplicación con las siguientes credenciales:

- **Correo electrónico:** user@user.com
- **Contraseña:** user111

Como se puede comprobar ya podríamos acceder al panel principal donde ya se mostrarían las noticias de los diferentes ayuntamientos a los que estemos “siguiendo”.

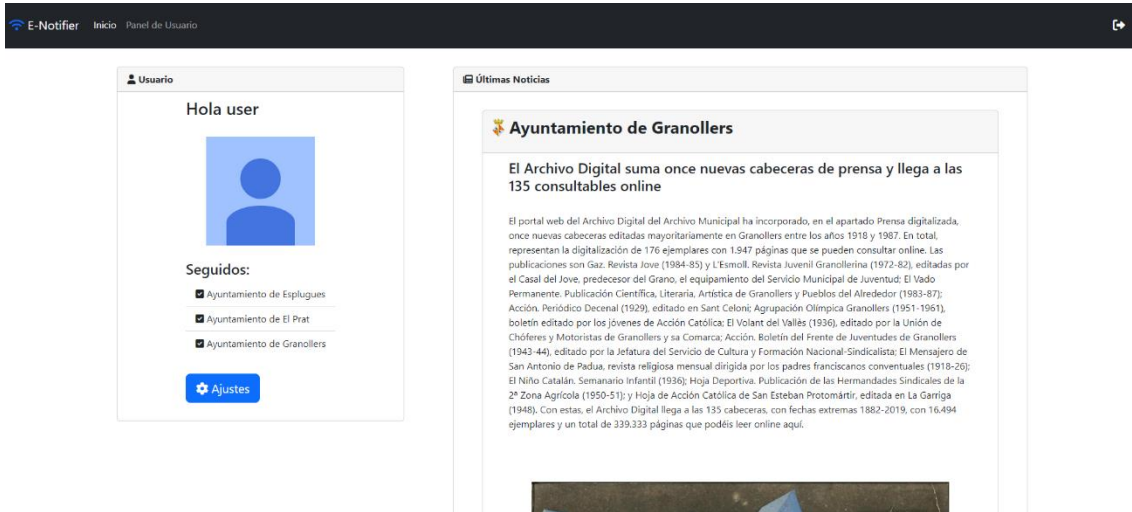


Figura 46 – Panel inicial con sesión ejecutado en versión local.

Para modificar esta configuración podemos hacerlo en el apartado ajustes:

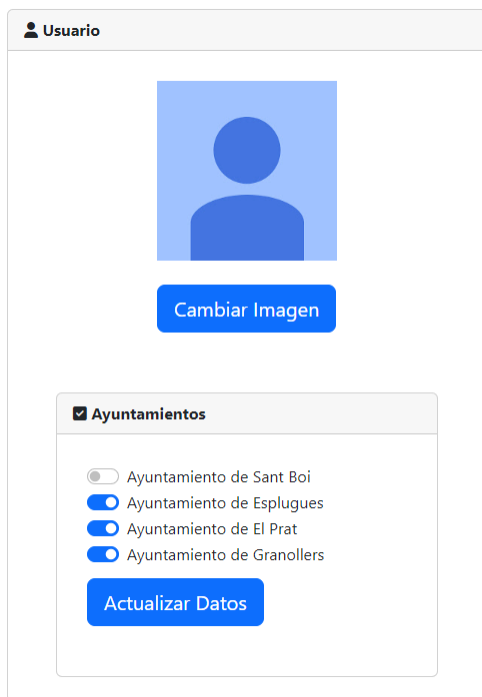


Figura 47 – Panel de administración de usuario ejecutado en versión local.

Para cerrar sesión solamente hay que hacer click en el icono de la parte superior derecha:

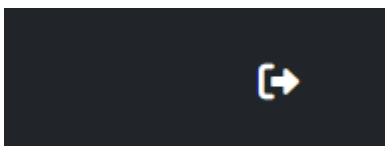


Figura 48 – Botón de cierre de sesión

Por otro lado, podemos acceder al apartado de administración mediante el usuario:

- **Correo electrónico:** admin@admin.com
- **Contraseña:** admin000

Accediendo con este usuario podemos ver un listado de los usuarios actuales junto a sus datos (exceptuando la contraseña que igualmente estaría codificada criptográficamente en md5) y tenemos opción de eliminar cada uno de ellos.

Panel de Administrador				
Nombre	Email	Rol	Imagen	Borrar Usuario
admin	admin@admin.com	2		
user	user@user.com	1		
test	test@test.test	1		

Figura 49 – Panel de administrador ejecutado en versión local.

* El rol 2 equivale al de administrador y el 1 al de usuario común.

El usuario con rol administrador tiene únicamente la finalidad de gestión de usuarios por lo que no tiene acceso a las últimas noticias como el resto de los usuarios.

Para tener la aplicación funcionando al 100% es necesario que la base de datos vaya acumulando las últimas noticias de los ayuntamientos y para ello pasaremos a la segunda parte de la guía donde se mostrara la instalación y ejecución del código Python junto con el de las librerías que se utilizarán.

4.4.2 Instalación Python

Para la parte de obtención, manipulación e inserción de los datos en la base de datos utilizaremos el lenguaje de programación Python.

Para ello descargaremos Python desde su web en este caso usaremos la versión 3.11.1:

<https://www.python.org/downloads/>



Figura 50 – Página de descarga de Python.

Una vez descargado Python haremos una instalación común con los valores por defecto.

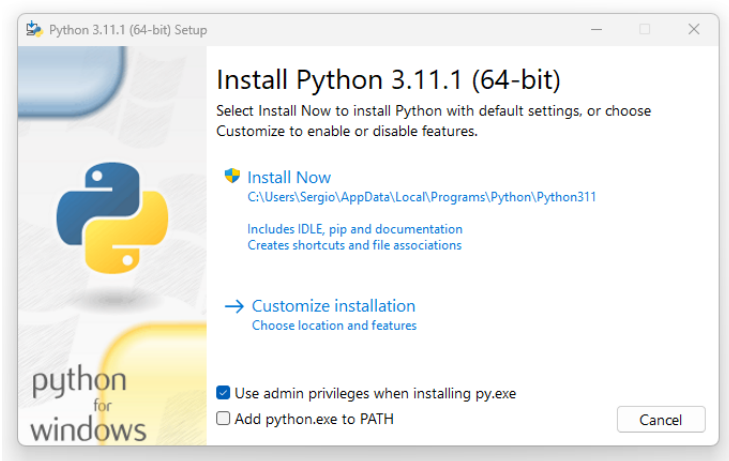


Figura 51 – Instalador de Python.

En los documentos descargados encontraremos una carpeta por cada ayuntamiento. En este caso se ha realizado la prueba del aplicativo con los siguientes 4 ayuntamientos:





 elprat	29/12/2022 2:14	Carpeta de archivos
 esplugues	27/12/2022 1:18	Carpeta de archivos
 granollers	29/12/2022 2:42	Carpeta de archivos
 sant boi	01/01/2023 20:07	Carpeta de archivos

Figura 52 – Ayuntamientos utilizados para las pruebas de funcionamiento.

Cada ayuntamiento tiene un script diferente puesto que no comparten la misma estructura, por lo que para actualizar la base de datos con las nuevas noticias de cada ayuntamiento habrá que ejecutar cada uno de los scripts de las diferentes carpetas.

Este proceso también podemos automatizarlo mediante un administrador regular de procesos en segundo plano para ejecutar procesos con intervalos regulares como puede ser cron.

En cada carpeta hay un archivo `upnoticiaxxxx.py` donde las “xxxx” representan el nombre del ayuntamiento en cuestión. Este es el script que ejecutaremos, pero antes de esto deberemos instalar las librerías correspondientes.

Para ello primero instalaremos pip (un sistema de gestión de paquetes utilizado para instalar y administrar paquetes de software escritos en Python).

Abriremos un terminal cmd. Y ejecutaremos estos dos comandos:

- `curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py`
- `python get-pip.py`


```
cmd. Símbolo del sistema
Microsoft Windows [Versión 10.0.22621.963]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Sergio>curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2509k  100 2509k    0     0  7043k      0  --:--:-- --:--:-- --:--:--  7048k

C:\Users\Sergio>python get-pip.py
Collecting pip
  Downloading pip-22.3.1-py3-none-any.whl (2.1 MB)
----- 2.1/2.1 MB 8.7 MB/s eta 0:00:00
Collecting wheel
  Downloading wheel-0.38.4-py3-none-any.whl (36 kB)
Installing collected packages: wheel, pip
  Attempting uninstall: pip
    Found existing installation: pip 22.0.4
    Uninstalling pip-22.0.4:
      Successfully uninstalled pip-22.0.4
Successfully installed pip-22.3.1 wheel-0.38.4
```

Figura 53 – Proceso de instalación de pip

A continuación, procedemos a instalar las librerías necesarias, para ello hay que localizar el archivo “**requirements.txt**”.

Este documento se encuentra dentro de cualquier carpeta de cualquier ayuntamiento y solo se tendrá que ejecutar una vez (vale para todos los demás ayuntamientos ya que se usan las mismas librerías).

Para poder instalarlo tenemos que ejecutar el siguiente comando dentro de la ubicación de cualquier carpeta en la terminal. En este caso lo haremos dentro del ayuntamiento de Esplugues.

- pip install -r requirements.txt

```
C:\Users\Sergio>cd C:\Users\Sergio\Desktop\TFG\TFG FINAL\TFG\esplugues
C:\Users\Sergio\Desktop\TFG\TFG FINAL\TFG\esplugues>pip install -r requirements.txt
```

Figura 54 – Proceso de instalación de las librerías necesarias

Una vez hecho esto ya tenemos el entorno preparado para ejecutar los scripts.

Haremos la prueba con un ayuntamiento que tiene una notificación nueva en este caso el Ayuntamiento de Esplugues.

Menú de noticias antes de ejecutar el script:



Figura 55 – Notificaciones mostradas antes de la ejecución del código.

Para ejecutar los scripts podemos hacerlo accediendo al IDLE de Python mediante Windows > Python > IDLE (Python 3.10 64-Bit).

Ejecutamos el script mediante el atajo F5 o bien Run > Run Module.

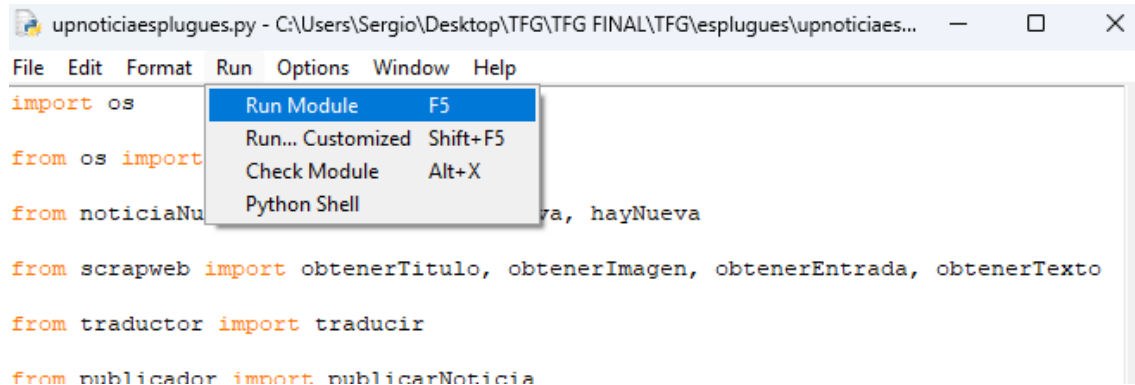


Figura 56 – Ejecución del script.

```
Python 3.10.4 (tags/v3.10.4:9d38120, Mar 23 2022, 23:13:41) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Sergio\Desktop\TFG\TFG FINAL\TFG\esplugues\upnoticiaesplugues.py
Obtener Noticia: OK
Traduccion: OK
Publicando noticia...
Noticia publicada...
>>>
```

Figura 57 – Respuesta en consola de la ejecución del script.

Como podemos comprobar la última noticia se ha publicado correctamente:

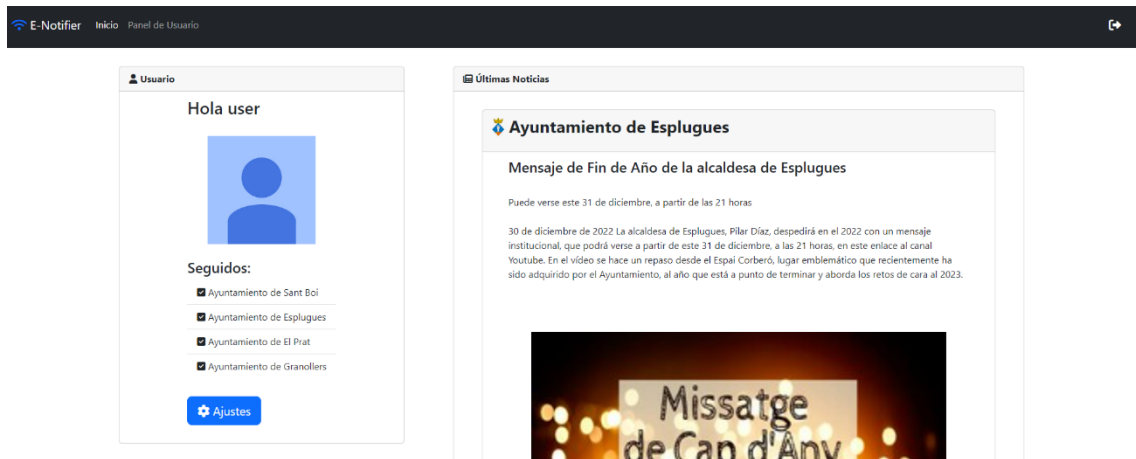


Figura 58 – Notificaciones mostradas después de la ejecución del código.

Demo: <https://sergiofabregat.online/>

5. Conclusiones

Este proyecto es el resultado de los conocimientos adquiridos en los grados superiores de Desarrollo de Aplicaciones Web (DAW), Desarrollo de Aplicaciones Multiplataforma (DAM), el actual Grado de Ingeniería Informática y 7 años de experiencia como técnico informático.

Mi pasión por la programación web junto con mi punto de vista futurístico sobre la automatización es lo que me ha empujado a llevar a cabo el presente proyecto. Una nueva tecnología que poco a poco junto a la IA se va abriendo camino en la dirección que está tomando la informática los últimos años.

Se han cumplido los objetivos principales del proyecto dando como resultado un producto mínimo viable totalmente funcional.

En cuanto a la planificación, se han realizado algunos cambios puesto que tal y como era de esperar es una tarea muy complicada la de calcular una temporalización ajustada al tratar con nueva tecnología y los posibles errores que esta pueda tener. No obstante, las líneas temporales globales marcadas para la presentación del curso del proyecto se han cumplido satisfactoriamente.

Líneas de trabajo futuro

El proyecto cuenta con una muy buena base siendo 100% funcional. Uno de los principales problemas que presenta es la limitación de ayuntamientos que ofrece, puesto que cada uno de ellos posee una estructura web diferente y por tanto no se puede crear un código genérico para el análisis de estos.

Por tanto, uno de los pasos a seguir es sin ninguna duda ampliar el número de ayuntamientos con tal de alcanzar un mayor público objetivo.

Por otro lado, uno de los aspectos que no eran necesarios para la ejecución del proyecto, pero sí para la personalización del usuario es la modificación de su imagen, ya que por el momento presenta una por defecto.

Finalmente, para culminar esta segunda fase se deberán añadir mejoras en el diseño ya que este es simple.

También sería una muy buena opción la creación de un aplicativo móvil ya que pese a que se dispone de un diseño responsive una gran parte de los usuarios consultan las notificaciones desde sus dispositivos móviles.



Figura 57 – Líneas de trabajo futuras.

6. Glosario

- **API:** Se trata de un conjunto de subrutinas, funciones y procedimientos ofrecidos por cierta biblioteca para ser utilizada por otro software. (Wikipedia)
- **HTTP:** Es el protocolo de comunicación que permite las transferencias de información a través de archivos a través de la red. (Wikipedia)
- **JavaScript:** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos. (Wikipedia)
- **HTML:** Hace referencia al lenguaje de marcado para la elaboración estándar de páginas web. (Wikipedia)
- **PHP:** Es un lenguaje de programación de uso general que se adapta especialmente al desarrollo web que puede ser incrustado en HTML. (Wikipedia)
- **MySQL:** Es un sistema de gestión de bases de datos relacional. (Wikipedia)
- **SQL:** Es un lenguaje de dominio específico, diseñado para administrar, y recuperar información de sistemas de gestión de bases de datos relacionales. (Wikipedia)
- **Scraping:** Técnica utilizada mediante programas de software para extraer información de sitios web. (Wikipedia)
- **Cron:** Es un administrador regular de procesos en segundo plano que ejecuta procesos o guiones a intervalos regulares. (Wikipedia)
- **Script:** Secuencia de comandos que se usa para designar un lenguaje de programación que se utiliza para manipular, personalizar y automatizar las instalaciones de un sistema existente. (Wikipedia)
- **Python:** Es un lenguaje de programación de propósito general y alto nivel. (Wikipedia)
- **Wireframe:** Es una guía visual que representa el esqueleto o estructura visual de un sitio web. (Wikipedia)

7. Bibliografía

- [1] Digital transformation (Última consulta: 15-01-2023)
https://en.wikipedia.org/wiki/Digital_transformation
- [2] Bo Zhao - University of Washington Seattle - DOI:10.1007/978-3-319-32001-4_483-1 In book: Encyclopedia of Big Data (pp.1-3)
Publisher: Springer International Publishing (2017) Web Scraping «*Web Scraping*» (Última consulta: 26-10-2022)
https://www.researchgate.net/publication/317177787_Web_Scraping
- [3] Priya Matta , Nikita Sharma, Devyani Sharma, Bhasker Pant, Sachin Sharma DOI: <https://doi.org/10.30534/ijtcse/2020/185952020> «Web Scraping: Applications and Scraping Tools» (Última consulta: 26-10-2022)
<https://www.warse.org/IJATCSE/static/pdf/file/ijtcse185952020.pdf>
- [4] Leonard Richardson «*Beautiful Soup*» (Última consulta: 15-01-2023)
<https://pypi.org/project/beautifulsoup4/>
- [5] Kenneth Reitz «*Requests*» (Última consulta: 15-01-2023)
<https://pypi.org/project/requests/>
- [6] «*Web Scraping*» (Última consulta: 15-01-2023)
https://es.wikipedia.org/wiki/Web_scraping
- [7] Bootstrap team «*Bootstrap*» (Última consulta: 25-12-2022)
<https://getbootstrap.com/>
- [8] The PHP Group «*Manual de PHP*» (Última consulta: 14-01-2023)
<https://www.php.net/manual/es/index.php>
- [9] Refsnes Data «*W3Schools*» (Última consulta: 02-01-2023)
<https://www.w3schools.com/>
- [10] Stack Exchange Inc «*Stack Overflow*» (Última consulta: 02-01-2023)
<https://stackoverflow.com/>
- [11] «*MD5*» (Última consulta: 15-12-2022)
<https://es.wikipedia.org/wiki/MD5>
- [12] «*Diseño web adaptable*» (Última consulta: 15-01-2023)
https://es.wikipedia.org/wiki/Dise%C3%B1o_web_adaptable

8. Anexos

El proyecto puede descargarse en GitHub:

<https://github.com/sfabregatr/enotifier>

Incluye: Código Fuente, Archivo SQL y Scripts de Python.

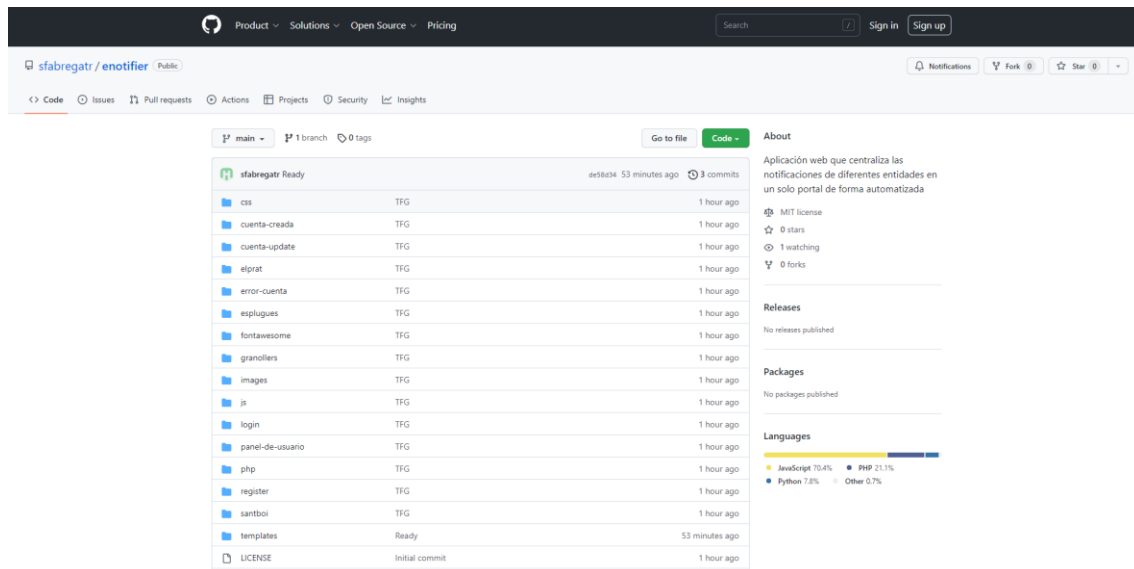


Figura 59 – Proyecto completo en GitHub.

Se puede acceder a la versión demo en: <https://www.sergiofabregat.online/>