

PROYECTO:

Empleos Castilla y León



Realizado por:

NOMBRE EQUIPO: Empleo360

- ***Alejandro Moral Rodríguez.***
- ***Marcos Pouso de la Madrid***



Desarrollo de Aplicaciones Web (DAW)

Valladolid, a 3 de febrero de 2026

Contenido

1. Introducción	3
2. Análisis	3
2.1. <i>Datasets</i> utilizados	3
2.2. Requisitos funcionales	4
2.3. Diagramas de casos de uso.....	12
3. Diseño	13
3.1. Modelo de base de datos	13
3.2. Diseño de la interfaz	14
3.2.1. Interfaz prototipo de la aplicación web en drawio.	14
3.2.2. Interfaz prototipo de la aplicación web en vista móvil.	15
3.3. Paleta de colores:.....	16
4. Desarrollo	17
4.1. <i>Stack</i> tecnológico.....	17
4.2. Estructura de carpetas	17
4.3. Descripción del funcionamiento.....	18
4.4. Acceso al Sistema	19
5. Pruebas	20
5.1. Registro:	20
5.2. Iniciar Sesión:	21
5.3. Publicar empleo:	21
5.4. Comprobación creación de empleo:.....	22
6. Despliegue	24
6.1. Configuración del Servidor Web (IIS)	24
6.2. Configuración de permisos de carpetas	24
6.3. Infraestructura de Red y Base de Datos	24
6.4. Habilitación de Extensiones	25

7. Sostenibilidad y "Green Coding"	26
8. Conclusiones	27
8.1. Autoevaluación	27
8.2. Valoración de la Metodología y Trabajo en Equipo	28
8.3. Comunicación y Cohesión	28
8.4. Líneas futuras	29
9. Bibliografía	29

1. Introducción

El presente Trabajo de Fin de Grado desarrolla una **aplicación web** orientada a facilitar el acceso y la consulta de ofertas de empleo publicadas por la **Junta de Castilla y León**. El proyecto se basa en el uso de un conjunto de **datos abiertos** proporcionado por la administración autonómica a través de su API oficial de empleo, que permite obtener información actualizada sobre vacantes disponibles en diferentes sectores y localidades de la comunidad.

La aplicación ha sido diseñada como un portal intermediario que centraliza y organiza estas ofertas en una interfaz más intuitiva, visual y accesible para la ciudadanía. A diferencia del portal original de la Junta de Castilla y León, que puede resultar complejo o poco ágil para ciertos usuarios, la web desarrollada en este proyecto prioriza la experiencia de usuario mediante filtros de búsqueda, paginación y un sistema de tarjetas (cards) que presentan cada oferta de manera clara y estructurada.

No obstante, para garantizar la validez y oficialidad de la información, cada tarjeta incluye un enlace directo a la publicación original en la web de la Junta de Castilla y León, donde el usuario puede consultar todos los detalles y realizar los trámites correspondientes.

La finalidad principal de la aplicación es mejorar el acceso a la información sobre empleo público y privado dentro de Castilla y León, abordando la necesidad ciudadana de contar con herramientas digitales más eficientes para la búsqueda de trabajo. En un contexto en el que el desempleo y la movilidad laboral siguen siendo desafíos relevantes, este proyecto busca contribuir a una mayor transparencia, accesibilidad y organización de los datos de empleo, facilitando así que cualquier persona, independientemente de su nivel de competencia digital, pueda encontrar oportunidades laborales de manera más sencilla y rápida.

2. Análisis

2.1. *Datasets* utilizados

- **Nombre:** Ofertas de empleo
- **URL:** [Enlace al Dataset](#) (JCYL, Ofertas de Empleo)
- **Campos utilizados:** Título, Provincia, Fecha publicación, ProvinciaAlternativa, FuenteContenido, ID Localidad, Localidad, Latitud, Longitud, Código localidad, Identificador, actualización de metadatos, Enlace al contenido, Posición.

- **Nombre:** Registro de municipios de Castilla y León
- **Url:** [Enlace al Dataset](#) (JCYL, Municipios CYL)
- **Campos Utilizados:** Municipio, Cod_Municipio, Provincia, Cod_Provincia, Cod_INE, Población, Mancomunidades, Entidades_Locales_Menores, Comarca, Longitud, Latitud, CoordenadaX, CoordenadaY.

2.2. Requisitos funcionales

- Iniciar sesión

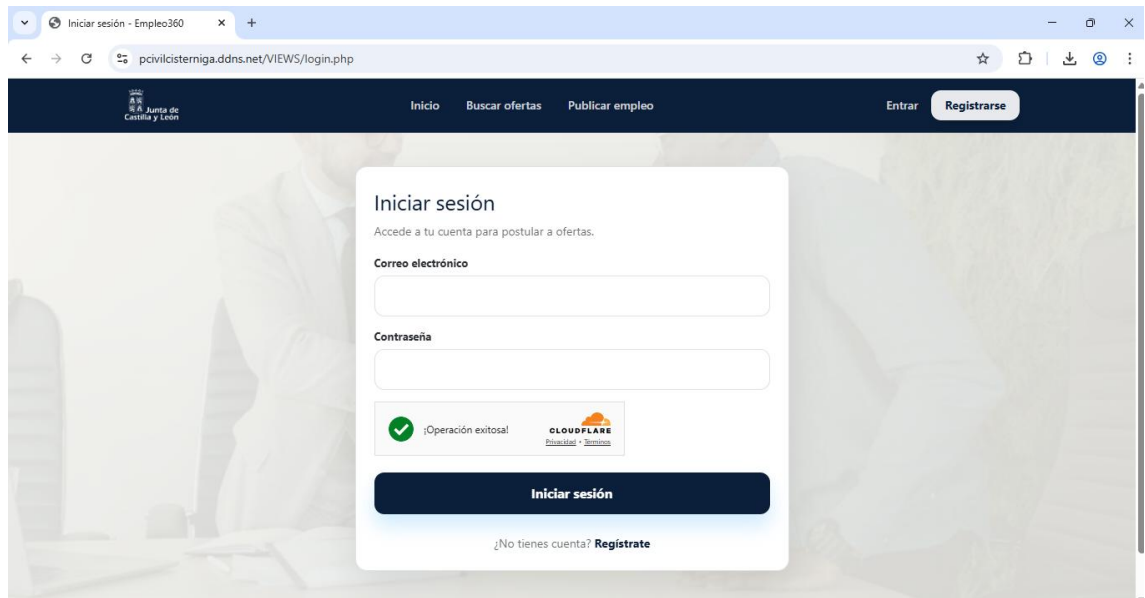
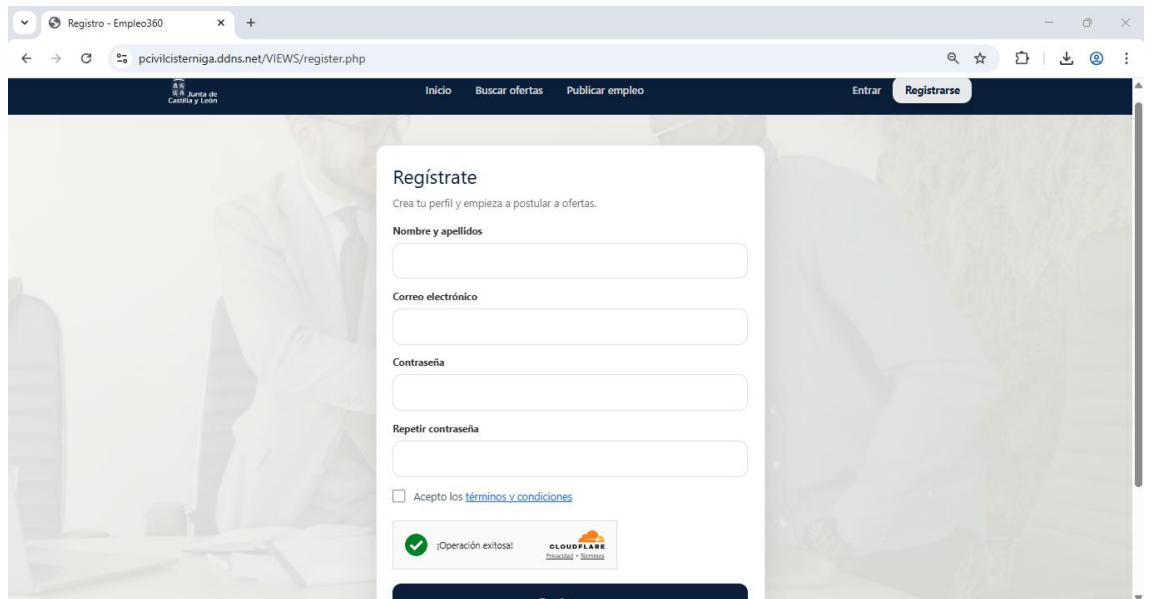


Figura 1. Captura de pantalla de inicio de sesión.

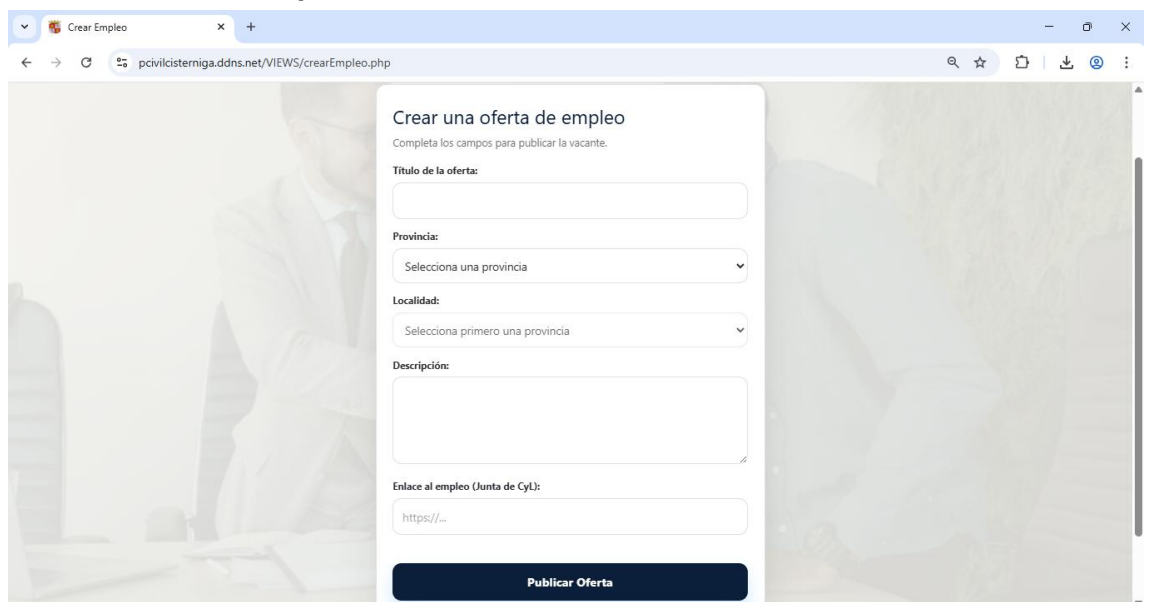
- **Registro**



The screenshot shows a web browser window with the address bar displaying "Registro - Empleo360" and the URL "pcivilcisterniga.ddns.net/VIEWS/register.php". The page features a dark blue header with navigation links: "Inicio", "Buscar ofertas", "Publicar empleo", "Entrar", and "Registrarse". The main content area has a light gray background with a blurred image of a person in a suit. A white registration form is centered on the page. The form is titled "Regístrate" and includes the instruction "Crea tu perfil y empieza a postular a ofertas.". The form fields are: "Nombre y apellidos", "Correo electrónico", "Contraseña", and "Repetir contraseña". Below these fields is a checkbox labeled "Acepto los términos y condiciones". At the bottom of the form, there is a green checkmark icon and the text "¡Operación exitosa!". A "Cloudflare" logo is also visible in the bottom right corner of the form area.

Figura 2. Captura de pantalla de registro de usuario.

- **Crear ofertas de empleo**



The screenshot shows a web browser window with the address bar displaying "Crear Empleo" and the URL "pcivilcisterniga.ddns.net/VIEWS/crearEmpleo.php". The page features a dark blue header with navigation links: "Inicio", "Buscar ofertas", "Publicar empleo", "Entrar", and "Registrarse". The main content area has a light gray background with a blurred image of a person in a suit. A white form titled "Crear una oferta de empleo" is centered on the page. The form includes the instruction "Completa los campos para publicar la vacante.". The form fields are: "Título de la oferta:", "Provincia:" (with a dropdown menu showing "Selecciona una provincia"), "Localidad:" (with a dropdown menu showing "Selecciona primero una provincia"), "Descripción:", and "Enlace al empleo (Junta de CyL):" (with a text input field containing "https://..."). At the bottom of the form is a dark blue button labeled "Publicar Oferta".

Figura 3. Captura de pantalla de crear oferta de empleo.

- **Cambiar contraseña, usuario y email propio**

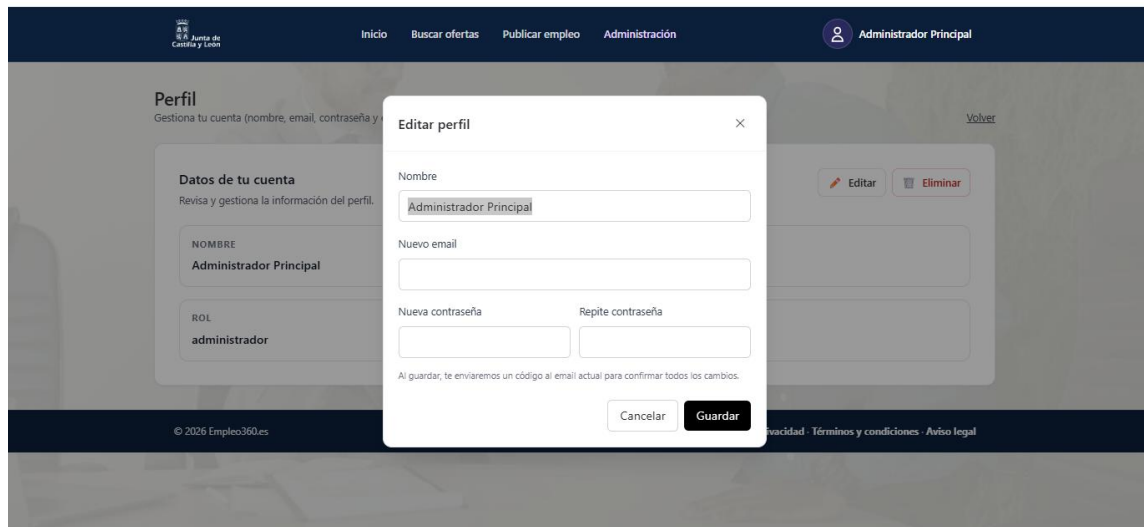


Figura 4. Captura de pantalla de edición de perfil.

- **Validación por correo electrónico con un número generado aleatoriamente**

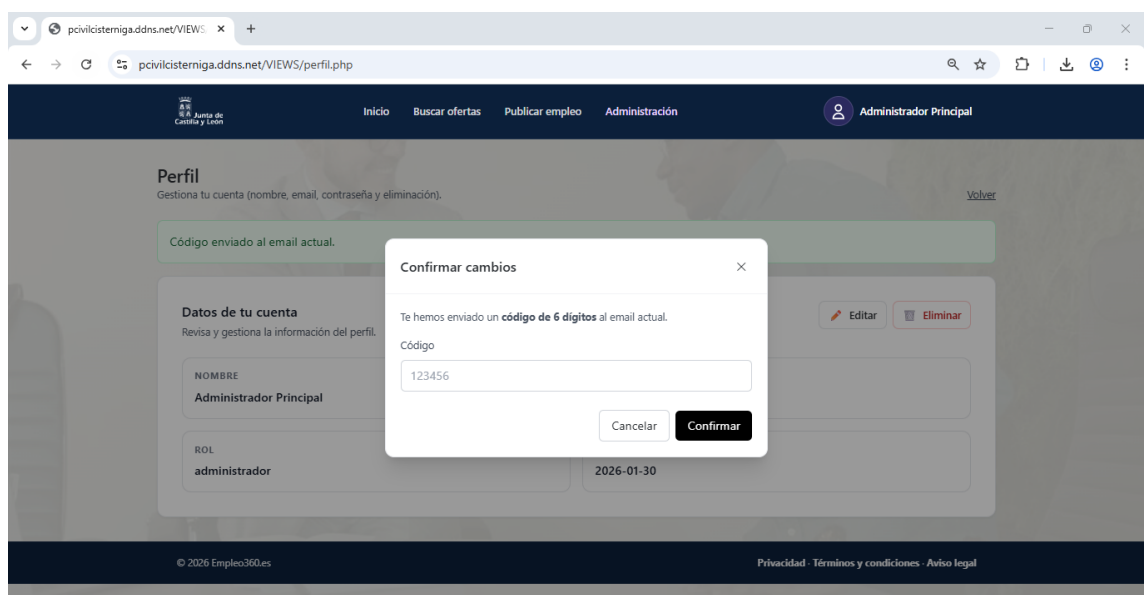


Figura 5. Captura de pantalla de validación por código.

- **Vista administrador**

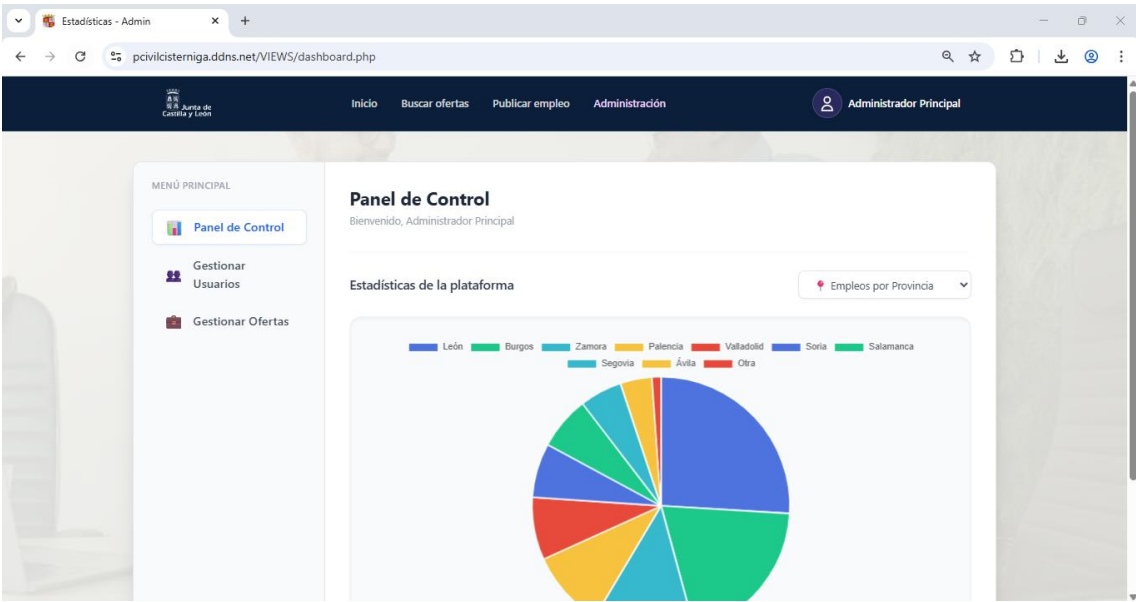


Figura 6. Captura de pantalla de inicio de vista de administrador.

- **Analíticas de administrador**

- Ofertas creadas por día



Figura 7. Captura de pantalla de las ofertas de empleo por día del mes.

Ofertas por provincia

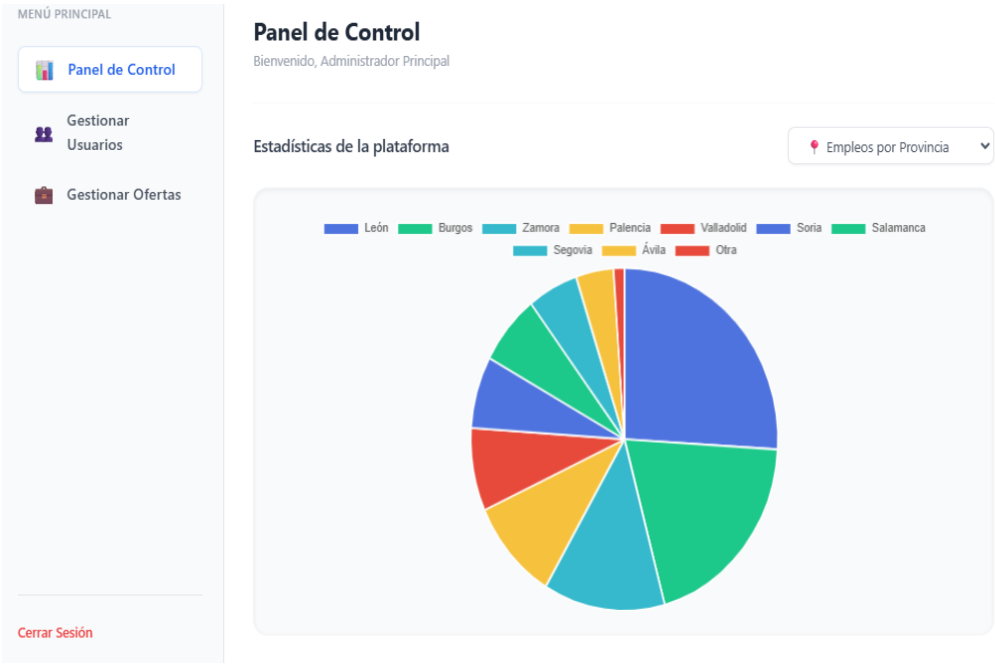


Figura 8. Captura de pantalla de las ofertas de empleo por provincia.

- **Capacidad de modificar/eliminar usuarios y ofertas (Administrador)**
 - Modificar/Eliminar usuarios

The screenshot shows the 'Gestión de Usuarios' interface. On the left is the same 'MENÚ PRINCIPAL'. The main area is titled 'Gestión de Usuarios' with a '+ Nuevo Usuario' button and 'Total: 18 registros'. Below is a table of users.

ID	USUARIO	ROL	REGISTRO	ACCIONES
#1	USUARIO 123	ADMINISTRADOR	27/01/2026	Editar Borrar
#2	alejandro	USUARIO	27/01/2026	Editar Borrar
#3	Prueba register	USUARIO	27/01/2026	Editar Borrar
#5	Funciona?	USUARIO	28/01/2026	Editar Borrar
#6	prueba captcha	USUARIO	28/01/2026	Editar Borrar
#7	cuentaPrueba	USUARIO	28/01/2026	Editar Borrar
#10	Funciona?	USUARIO	29/01/2026	Editar Borrar

Figura 9. Captura de pantalla de la gestión de los usuarios (vista administrador)

○ Modificar/Eliminar ofertas



Figura 10. Captura de pantalla de la gestión de las ofertas (vista administrador)

● Buscar en un mapa las ofertas de empleo

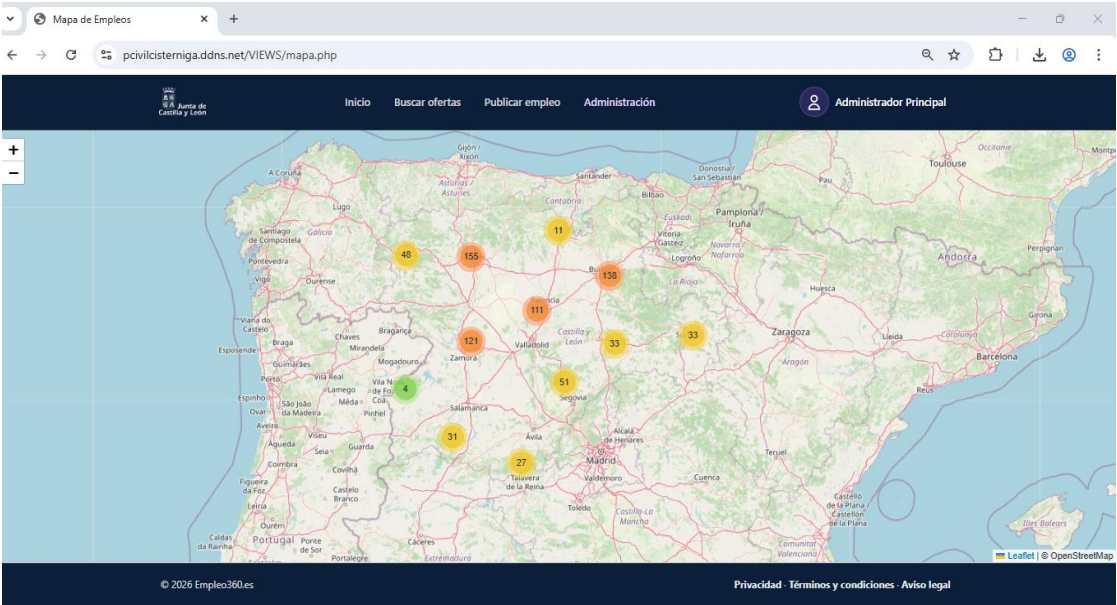


Figura 11. Captura del mapa dinámico con las ofertas de empleo

- **Filtrar ofertas**
 - Filtrar por titulo

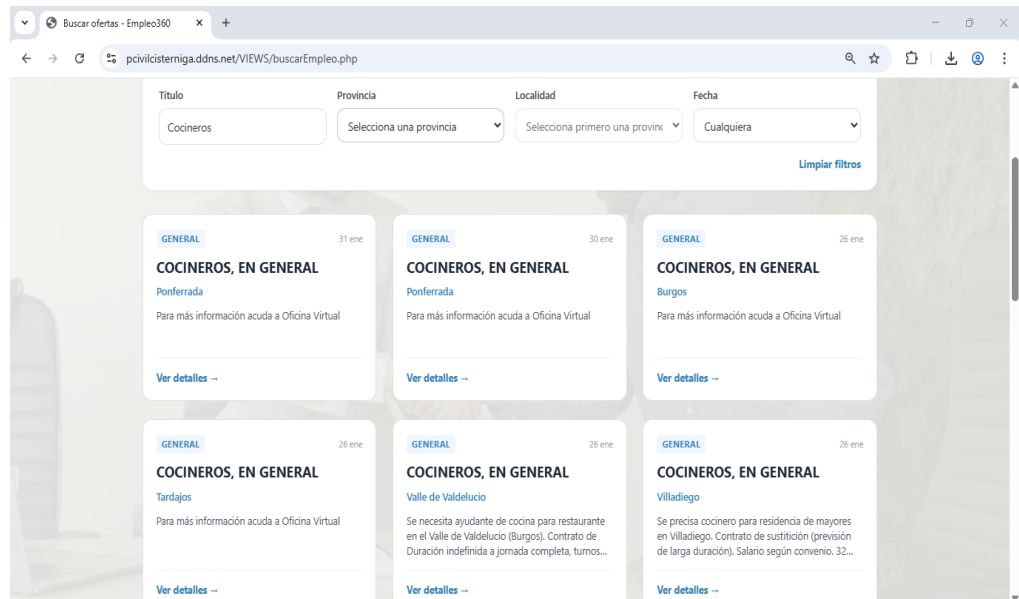


Figura 12. Captura de pantalla de las ofertas de trabajo usando filtros de texto (apartado de búsqueda de empleo)

- Filtrar por provincia

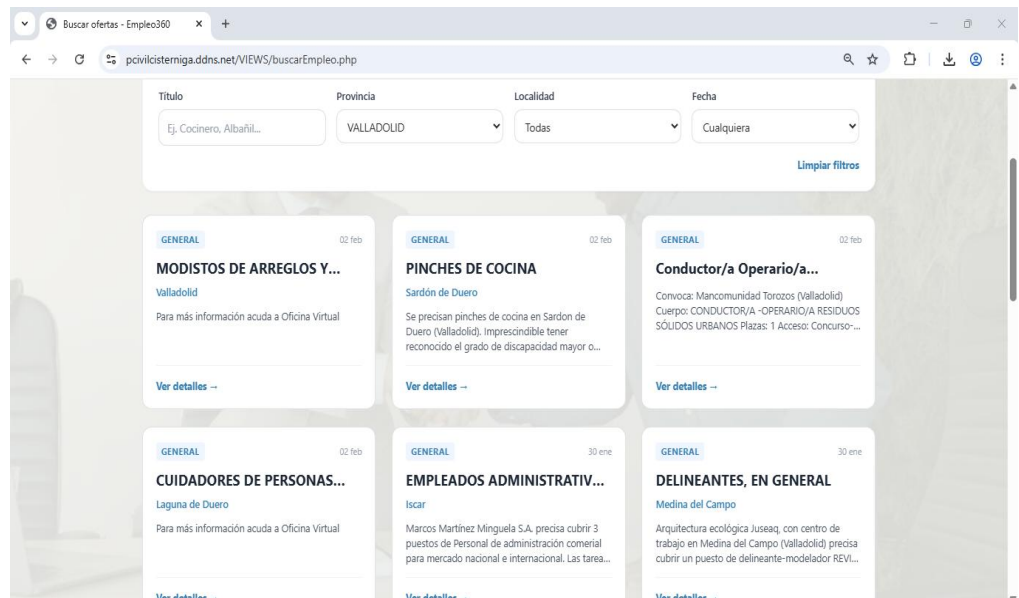


Figura 13. Captura de pantalla de las ofertas de trabajo filtrando por provincia (apartado de búsqueda de empleo)

○ Filtrar por municipio

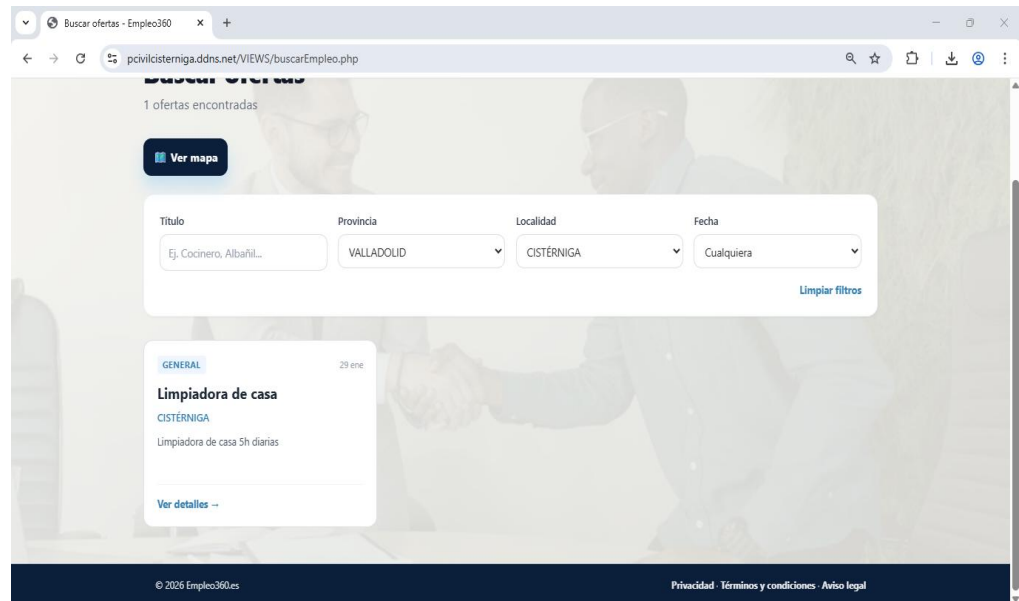


Figura 14. Captura de pantalla de las ofertas de trabajo filtrando por municipio (apartado de búsqueda de empleo)

○ Filtrar por fecha

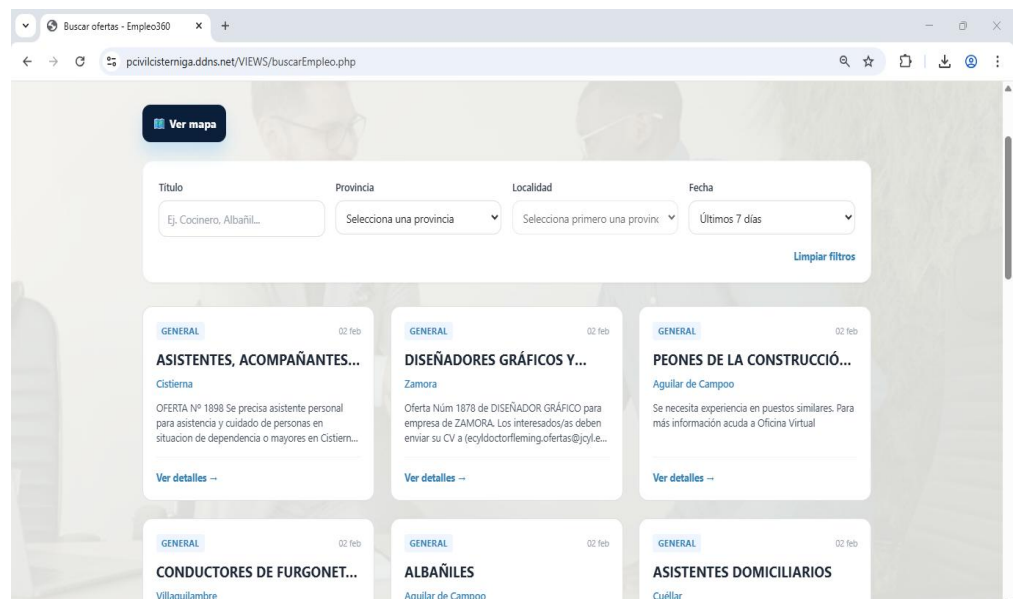


Figura 15. Captura de pantalla de las ofertas de trabajo filtrando por fecha (apartado de búsqueda de empleo)

2.3. Diagramas de casos de uso

Diagrama de caso de uso creado en Draw.io (Drawio, 2026)

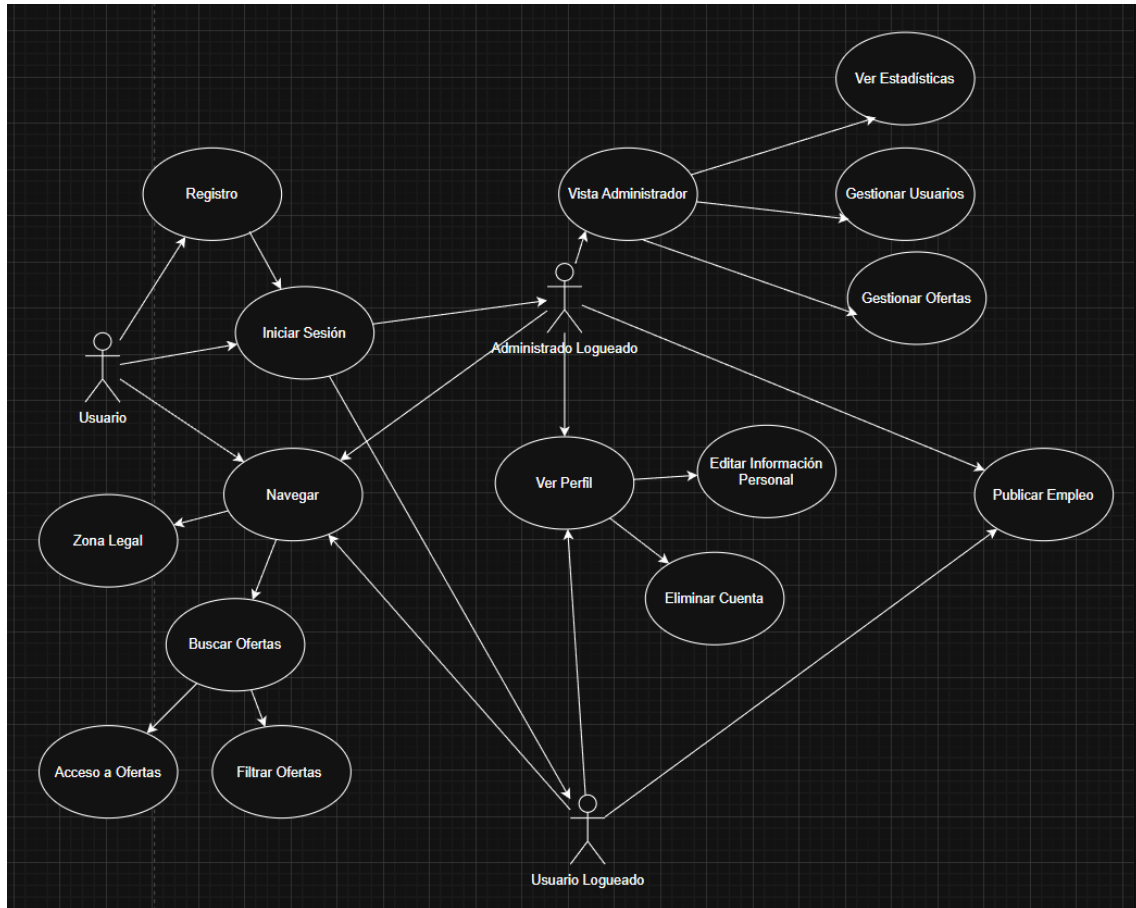


Figura 16. Diagrama de casos de usos de la aplicación web.

3. Diseño

3.1. Modelo de base de datos

- Diagrama Entidad-Relación:

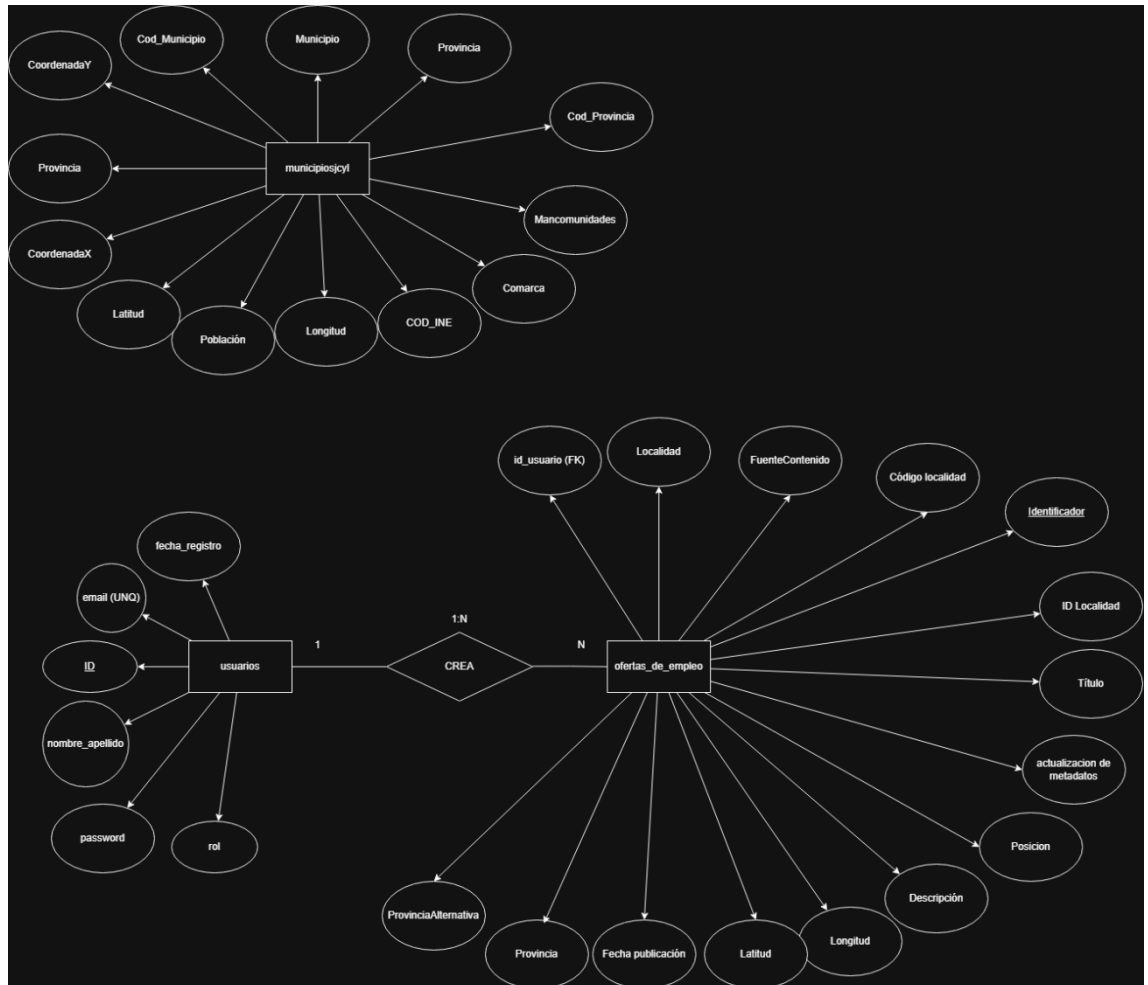


Figura 17. Captura del diagrama Entidad-Relación

- Justificación tablas:

- ofertas_de_empleo:** Tabla original del *dataset*, se relaciona con usuarios ya que presentan la capacidad de crear nuevas ofertas de empleo, las ofertas pueden ser creadas por un solo usuario, pero un solo usuario puede crear más de una oferta (1:N) presenta una *foreign key* de la id del usuario que ha creado la oferta. Su clave primaria es Identificador

- **usuarios:** Tabla creada para gestionar los usuarios, permitiendo registro e inicio de sesión, su *Primary Key* (Clave primaria) es un identificador, la relación es la misma que en ofertas_de_empleo
- **municipiosjcyL:** Dataset público de la Junta de Castilla y León utilizado para obtener las coordenadas de los municipios al crear nuevas ofertas

3.2. Diseño de la interfaz

3.2.1. Interfaz prototipo de la aplicación web en drawio.

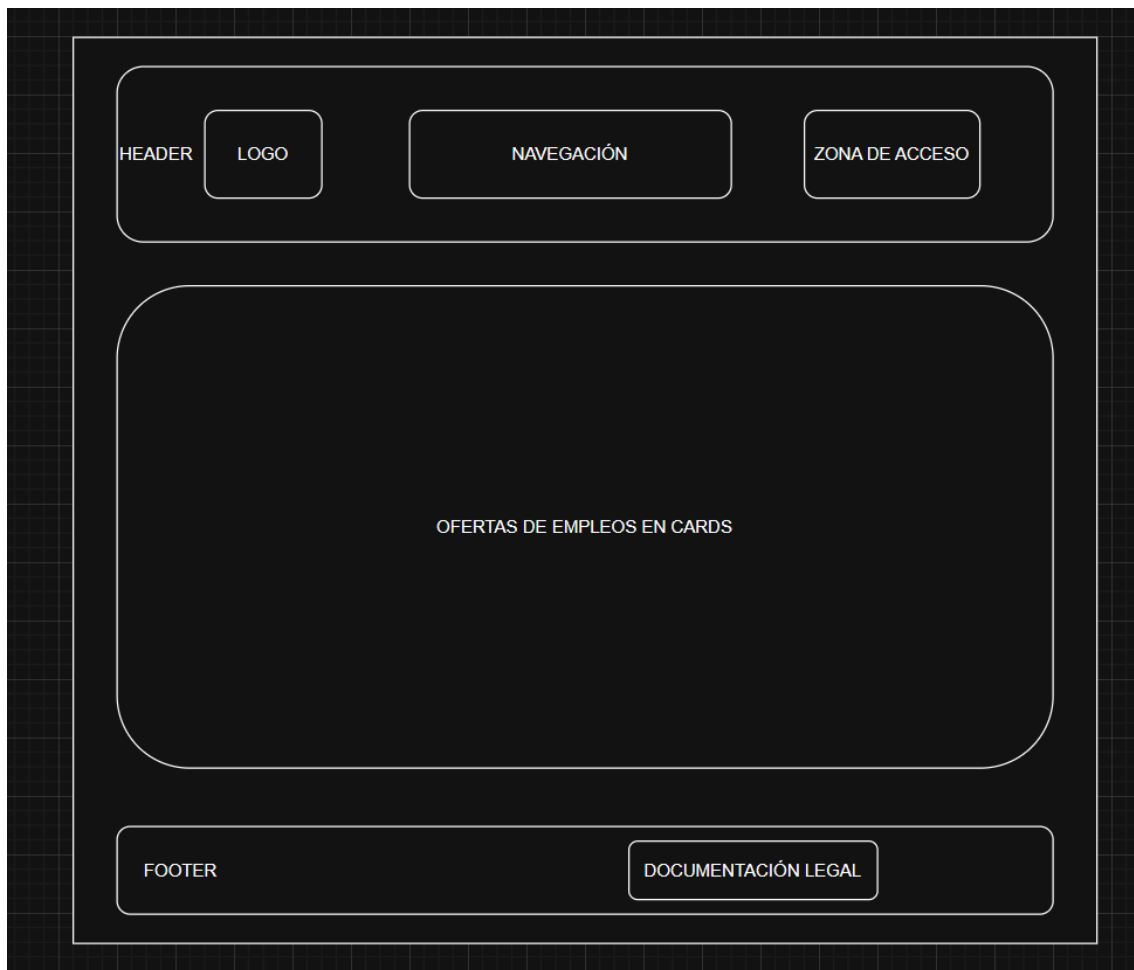


Figura 18. Prototipo de diseño principal de la aplicación (versión escritorio)

3.2.2. Interfaz prototipo de la aplicación web en vista móvil.



Figura 19. Prototipo de diseño principal de la aplicación (versión móvil)

3.3. Paleta de colores:

Paleta de colores generada en Adobe Color (Adobe Color):

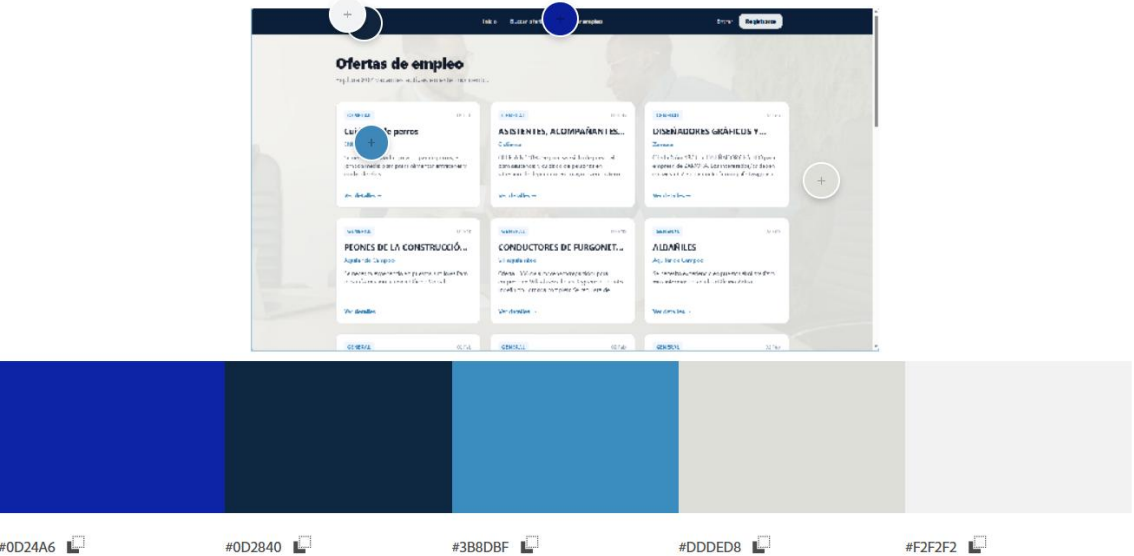


Figura 20. Paleta de colores de la página web en Adobe Color.

Guía de estilos completa mostrando fuente y paleta de colores.



Figura 21. Paleta de colores junto con la fuente.

4. Desarrollo

4.1. Stack tecnológico

- **Entorno:**
 - Para este proyecto se ha utilizado WIMP (Windows, IIS, MySQL, PHP) (IIS, 2026) (MySQL, 2026) (PHP, 2026) (Windows, 2026).
 - Backend: PHP Vanilla con patrón MVC y DAO.
 - Frontend: HTML5, CSS3, JavaScript y Tailwind. (W3Schools, 2026)
 - Base de datos: MySQL (gestionada con phpMyAdmin).
 - Control de versiones y herramientas: Entorno Git (Repositorio Github (GitHub, 2026)).
 - Despliegue: IIS con PHP.
- **Software:** Visual Studio, Gemini, XAMPP e IIS. (Gemini, 2026) (XAMPP, 2026) (VSC, 2026)

4.2. Estructura de carpetas

```
/
— ASSETS/
| — API/
| — CSS/
| — IMG/
| — JS/
— AUTH/
— CONFIG/
— DAO/
— LIBS/
| — PHPMailer/
— VIEWS/
```

La aplicación se ha organizado siguiendo una arquitectura modular que separa las responsabilidades de cada componente, facilitando el mantenimiento, la seguridad y la futura escalabilidad del proyecto.

- **ASSETS/:** Contiene los recursos estáticos y la lógica del soporte del lado del cliente. Se subdivide en directorios para hojas de estilo (**CSS**), recursos gráficos (**IMG**) y scripts de front-end (**JavaScript**). Además, incluye el directorio API, que actúa como el núcleo de procesamiento de peticiones asíncronas (**AJAX**) entre cliente-servidor .
- **AUTH/:** Directorio dedicado exclusivamente a la gestión de sesiones y control de acceso, centralizando procesos como el inicio, cierre de sesión, registro y confirmaciones de email.
- **CLASSES/:** Contiene las definiciones de las clases y objetos del sistema (**Usuario, Empleo**). Representa la lógica de negocio, definiendo propiedades y comportamientos de las entidades sin interactuar directamente con la base de datos.
- **CONFIG/:** Aloja los parámetros de configuración global, principalmente la conexión segura a la base de datos mediante PDO.
- **DAO/:** Capa de persistencia que actúa como intermediaria entre las clases y la base de datos, ejecutando las consultas SQL pertinentes.
- **DB/:** Directorio que contiene los archivos que interactúan directamente con la base de datos como, por ejemplo: llamada a la API para actualizar la base de datos, esquema de la base de datos y un script para realizar la importación de esta.
- **INCLUDES/:** Aloja fragmentos de código PHP y HTML **reutilizables** que no son vistas completas y que son comunes a varios archivos, como la cabecera, el pie de página y funciones comunes a distintos archivos
- **LIBS/:** Dependencias externas (PHPMailer).
- **VIEWS/:** Capa de presentación. Incluye las páginas principales y todas las partes visuales de la aplicación

4.3. Descripción del funcionamiento

El presente manual describe las funcionalidades disponibles para el usuario final de la aplicación Empleo360, detallando los procesos de interacción con el sistema.

4.4. Acceso al Sistema

Para utilizar las funciones personalizadas, el usuario debe identificarse:

1. **Inicio de sesión:** Introducir el correo electrónico y la contraseña desde la pantalla de *Login*.
2. **Estado de sesión:** Una vez dentro, el menú de navegación (Navbar) se actualizará para mostrar las opciones privadas y el nombre del usuario.

Gestión de Perfil

El usuario puede mantener sus datos actualizados para garantizar una comunicación efectiva:

1. **Edición de datos:** Desde la sección *Mi Perfil*, se pueden modificar el nombre, el correo electrónico o la contraseña.
2. **Verificación de seguridad:** Al solicitar un cambio de datos sensibles (como el email), el sistema enviará automáticamente un código de 6 dígitos al correo electrónico actual.
3. **Confirmación:** El usuario debe introducir dicho código en la ventana emergente para que los cambios se apliquen de forma permanente.

Búsqueda y Visualización de Empleo

La plataforma permite a los usuarios consultar las oportunidades disponibles:

1. **Listado de ofertas:** En la sección *Buscar Empleo*, el usuario encontrará una lista de vacantes activas.
2. **Detalle de la oferta:** Al hacer clic en una vacante, se redirige a la página de empleo de La Junta de Castilla y León.

Creación de Empleo

Si el usuario dispone de los permisos necesarios, podrá gestionar vacantes:

1. **Creación:** Acceder al formulario de *Crear Empleo* y completar los campos requeridos (Título, Ubicación, Descripción y Enlace).
2. **Publicación:** Al guardar, la oferta será visible de inmediato para el resto de los usuarios de la plataforma.

Cierre de Sesión

Por seguridad, el usuario debe finalizar su sesión al terminar su actividad:

1. Hacer clic en el botón **Salir** situado en el menú de navegación.
2. El sistema destruirá los tokens de sesión y redirigirá al usuario a la página de inicio pública.

5. Pruebas

Para evaluar la facilidad de uso y la claridad de la aplicación, se ha realizado una prueba de usabilidad con un usuario real. La tarea propuesta consistía en **registrarse en la plataforma, iniciar sesión y publicar una oferta de empleo**, verificando posteriormente que dicha oferta aparecía correctamente en el listado público.

5.1. Registro:

Figura 22. Captura de la prueba del usuario en la zona de registro.

El usuario accedió inicialmente a la página de registro desde el menú superior. El formulario de registro presentaba campos claros y estructurados para introducir nombre y apellidos, correo electrónico y contraseña, así como la aceptación de los términos y condiciones.

Durante este proceso no se detectaron dificultades relevantes: los campos eran fácilmente identificables, el botón de registro era visible y el sistema proporcionó retroalimentación inmediata tras el envío del formulario. Una vez completado correctamente, la aplicación mostró un mensaje de confirmación indicando que la cuenta había sido creada con éxito.

5.2. Iniciar Sesión:

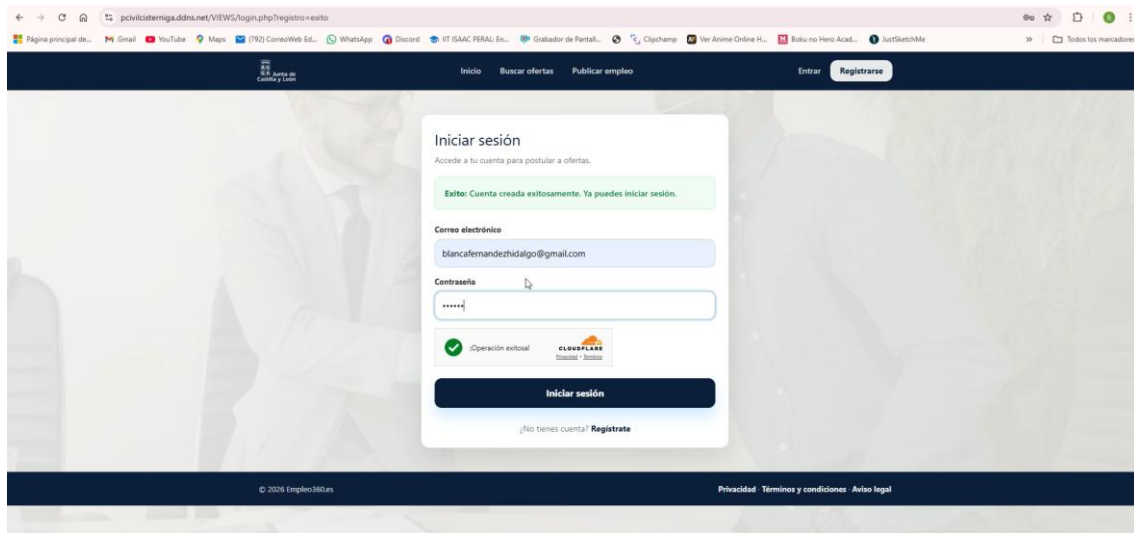


Figura 23. Captura de la prueba del usuario en la zona de inicio de sesión

Tras el registro, el usuario accedió a la pantalla de inicio de sesión. El formulario solicitaba únicamente el correo electrónico y la contraseña previamente registrados, lo que simplificó el acceso.

El proceso resultó intuitivo y rápido, y el usuario pudo iniciar sesión sin errores. Además, la interfaz mostraba de forma clara que la sesión había sido iniciada correctamente, mostrando el nombre del usuario en la barra de navegación superior.

5.3. Publicar empleo:

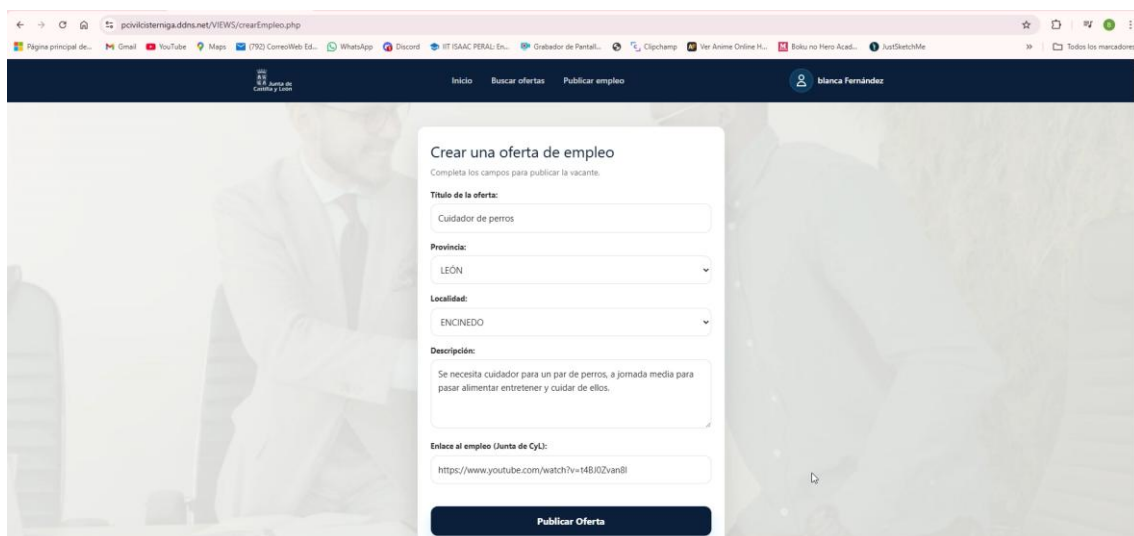


Figura 24. Captura de la prueba del usuario creando una oferta de empleo.

Una vez autenticado, el usuario seleccionó la opción “Publicar empleo” desde el menú principal. La aplicación mostraba un formulario estructurado con los siguientes campos: título de la oferta, provincia, localidad, descripción y enlace oficial a la oferta de la Junta de Castilla y León.

El diseño del formulario facilitó la comprensión de la información requerida y permitió al usuario completar la tarea sin necesidad de instrucciones adicionales. Tras rellenar todos los campos y pulsar el botón “Publicar oferta”, el sistema procesó la solicitud correctamente.

5.4. Comprobación creación de empleo:

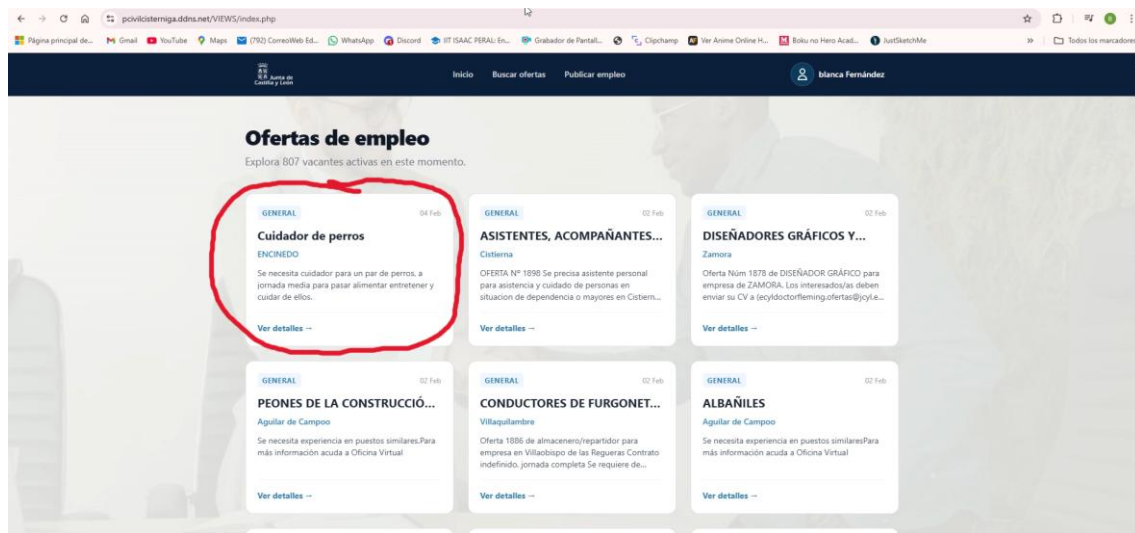


Figura 25. Prueba del usuario completada mostrando la oferta creada.

Para verificar el éxito de la operación, se revisó la página principal de ofertas de empleo. La nueva oferta publicada —“Cuidador de perros”— apareció correctamente en el listado, con su título, localidad y descripción visibles dentro de una tarjeta de empleo.

Esto confirmó que el proceso de publicación funcionaba adecuadamente y que la información introducida por el usuario se almacenaba y mostraba correctamente en la plataforma.

Análisis automático de accesibilidad con Lighthouse de Google Chrome

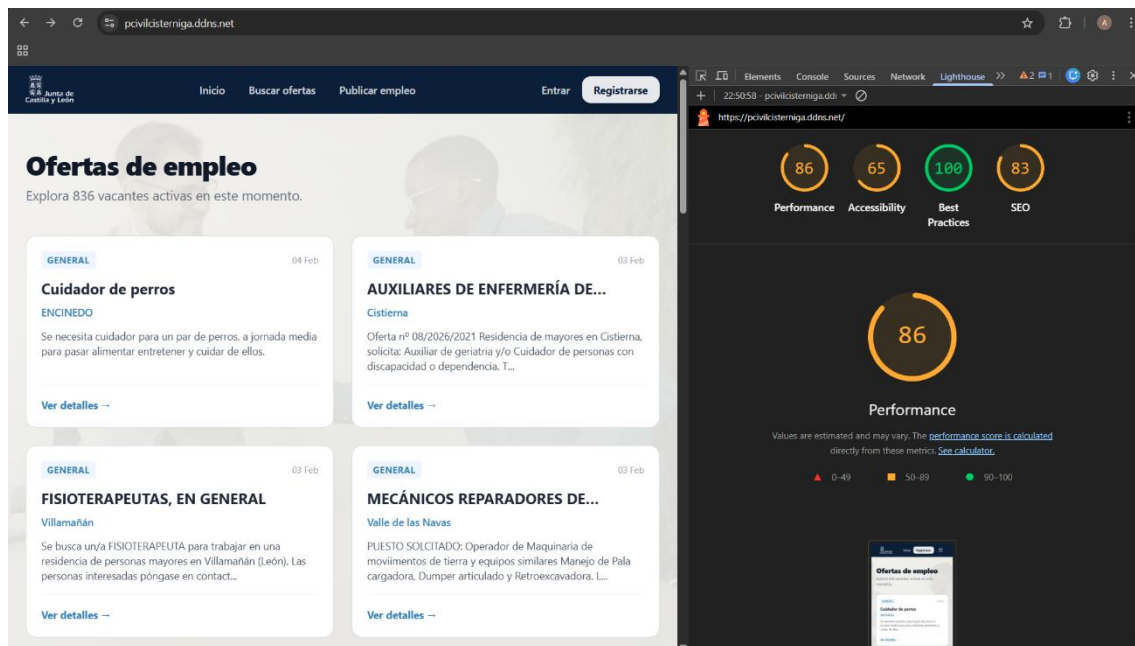


Figura 26. Captura de la valoración de la herramienta de Chrome: "Lighthouse"

6. Despliegue

- **Instrucciones de instalación en local:**

Este apartado describe el proceso técnico seguido para el despliegue de la aplicación en un entorno de servidor utilizando las herramientas nativas de Windows Server.

Dado que IIS no procesa PHP de forma nativa, se ha configurado mediante el protocolo **FastCGI**. Este permite que el servidor web delegue la ejecución de los scripts *.php* a un proceso independiente, mejorando la velocidad de respuesta y la estabilidad del sistema.

6.1. Configuración del Servidor Web (IIS)

El servidor de aplicaciones se ha implementado sobre **Internet Information Services (IIS)**, aprovechando su integración nativa con el sistema operativo Windows para ofrecer un entorno de ejecución de alto rendimiento.

6.2. Configuración de permisos de carpetas

Uno de los puntos clave en el despliegue ha sido la gestión de permisos NTFS para el usuario anónimo del servidor:

- **Identidad IUSR:** Se han asignado permisos de lectura y ejecución al usuario de sistema IUSR sobre el directorio raíz del proyecto.
- **Permisos de Escritura:** Se han restringido los permisos de escritura exclusivamente a la carpeta de recursos dinámicos (ASSETS/img/), minimizando la superficie de ataque en caso de intentos de inyección de archivos maliciosos.

6.3. Infraestructura de Red y Base de Datos

El despliegue se ha realizado mediante una arquitectura de red segmentada, separando físicamente el servidor de aplicaciones del servidor de datos para mejorar la seguridad y el rendimiento del sistema:

- **Servidor de Base de Datos (Nodo de Datos):** Se utiliza un equipo dedicado dentro de la red local que ejecuta el motor **MySQL** a través de la plataforma **XAMPP**. Este equipo tiene configurada una **IP privada estática** para garantizar que el servidor web pueda localizarlo de forma persistente. La

base de datos, denominada *proyecto_empleo_db*, se gestiona mediante la interfaz de **phpMyAdmin**.

- **Servidor de Aplicaciones (Nodo Web):** El servidor web principal corre bajo **IIS (Internet Information Services)** en un segundo ordenador independiente. Este servidor actúa como el único punto de entrada para los usuarios y es el encargado de procesar la lógica de PHP. Para obtener la información, el servidor web realiza peticiones internas a la IP estática del nodo de datos.
- **Configuración de Acceso Externo:** Para permitir que la aplicación sea accesible desde fuera de la red local, se han implementado las siguientes medidas de red:
 - **Port Forwarding (Reenvío de Puertos):** Se ha configurado el router de la red para que todo el tráfico entrante por los puertos estándar de navegación sea redirigido exclusivamente a la IP local del servidor IIS.
 - **Publicación:** Gracias a esta técnica, el servidor IIS actúa como una pasarela, manteniendo el servidor de base de datos protegido dentro de la red interna y sin exposición directa a internet.

6.4. Habilitación de Extensiones

Extensiones de PHP Habilitadas

Para garantizar la plena operatividad de la plataforma en el servidor IIS, se han activado y configurado las siguientes extensiones en el intérprete PHP:

- **Extensión *pdo_mysql*:** Es la extensión fundamental para la persistencia de datos. Permite que PHP utilice la interfaz PDO (PHP Data Objects) para conectarse de forma segura al servidor MySQL alojado en el segundo ordenador de la red local. Su uso es crítico para la prevención de ataques de inyección SQL mediante sentencias preparadas.
- **Extensión *Openssl*:** Vital para la seguridad en las comunicaciones salientes. Esta extensión permite el cifrado de datos mediante protocolos SSL/TLS, requisito indispensable para que la librería PHPMailer pueda autenticarse con los servidores de Google (SMTP) y enviar los códigos de verificación a los usuarios.
- **Extension *Curl*:** Se habilitó para permitir que el servidor realice peticiones HTTP externas para actualizar la base de datos de manera automática realizando una llamada a la API de La Junta

- **URLs:**
 - [Repositorio de Git](#)
 - [Aplicacion desplegada](#)

7. Sostenibilidad y "Green Coding"

Estas son las decisiones tomadas para reducir el impacto ambiental:

- **Estrategia de Caché:** Explicación técnica de cómo se han evitado llamadas innecesarias a la API (tiempo de caché, almacenamiento local o en BD).
- **Optimización de recursos:**
 - **Minificación de recursos:** Con el objetivo de mejorar los tiempos de carga y la eficiencia energética, en este proyecto se ha utilizado una herramienta llamada *Minify* para minificar todo lo posible el código de JavaScript y CSS, esta técnica es clave debido a que:
 - **Reduce el tráfico de red:** Ya que disminuye el tamaño de los paquetes transmitidos.
 - **Eficiencia en el cliente:** Al ser más eficiente debido a la cantidad de código cargado, es más liviano para el dispositivo y necesita un hardware menos exigente.
 - Gestión eficiente del renderizado: Para optimizar el consumo de recursos durante el acceso a la aplicación y llamadas a las APIs, se ha implementa una carga diferida de scripts
 - Para ello se han utilizado los atributos:
 - **Defer**
 - **Async**
- **Reflexión:** Desde la perspectiva del *Green Coding*, la implementación de caché supone un ahorro energético al eliminar procesos redundantes:
 - **Ahorro en servidor:** Evitando que la CPU y el disco trabajen en cada petición, sirviendo datos desde la RAM. Esto reduce el consumo eléctrico.
 - **Eficiencia de red:** Al no retransmitir datos que no han cambiado, se ahorra la energía necesaria para mover bytes a través de los nodos de Internet

- **Consumo en el cliente:** El dispositivo del usuario no gasta batería activando la antena de red ni procesando nuevas descargas, recuperando el contenido localmente
- **En resumen:** La caché transforma un gasto energético constante en un proceso de bajo consumo, siguiendo el principio de no ser redundante

8. Conclusiones

8.1. Autoevaluación

- **Dificultades encontradas:**

Durante el ciclo de vida del proyecto, surgieron diversos desafíos técnicos que requirieron investigación y adaptación. A continuación, se detallan los más significativos:

- **Conectividad de Red y Persistencia:** Inicialmente, se planteó una arquitectura sobre una **vLAN** con dos instancias de XAMPP. La principal dificultad residió en la configuración de permisos de acceso remoto de MySQL y la resolución de nombres entre nodos, lo que derivó en una reestructuración de la red local para asegurar una conexión estable entre el servidor web y el de datos.
- **Curva de Aprendizaje en IIS (Internet Information Services):** Al optar por una solución de grado empresarial sobre Windows, el equipo se enfrentó a un entorno sin preconfiguración de PHP. Los obstáculos incluyeron la gestión de dependencias como los **Visual C++ Redistributable**, la configuración del módulo **FastCGI** y la habilitación manual de extensiones críticas (`openssl`, `pdo_mysql`), lo que supuso un aprendizaje intensivo sobre administración de servidores Windows.
- **Implementación de Tailwind CSS (Tailwind CSS):** Al ser un framework basado en utilidades, la dificultad radicó en pasar de un diseño CSS tradicional a un flujo de trabajo de clases utilitarias. Se debieron testear varios métodos de integración (CDN vs CLI) hasta encontrar el equilibrio óptimo entre agilidad de desarrollo y rendimiento.
- **Tratamiento de Datos en el Dataset:** El manejo de archivos externos presentó un desafío de normalización. Los campos con caracteres especiales (tildes, espacios y signos de puntuación) en el origen de datos dificultaron el mapeo automático. Se tuvo que implementar una lógica de limpieza y tratamiento de *strings* para asegurar la integridad de la información al insertarla en la base de datos.

- **Automatización de Procesos en Entornos de Desarrollo:** La actualización automática de la base de datos presentó retos logísticos antes del despliegue final. Coordinar la sincronización de datos entre el entorno de pruebas y el servidor de producción requirió el diseño de procedimientos específicos para garantizar que la información estuviera siempre al día sin intervención manual constante.

8.2. Valoración de la Metodología y Trabajo en Equipo

La ejecución de este proyecto se ha regido por una metodología de trabajo colaborativa y estructurada, orientada a la consecución de objetivos específicos y al aprovechamiento de las capacidades técnicas de cada miembro del equipo.

Flujo de Trabajo y Herramientas de Colaboración

- **Control de Versiones mediante GIT:** Se implementó el uso de GIT como herramienta central para el control de versiones. Esto no solo facilitó la integración del código desarrollado de forma independiente, sino que permitió mantener un historial de cambios seguro, facilitando la reversión de errores y la fusión de las distintas capas del sistema (Backend, Frontend e Infraestructura).
- **Gestión por Metas Diarias:** En lugar de una planificación rígida y estática, el trabajo se estructuró mediante la definición de metas a corto plazo. Este enfoque permitió tener una hoja de ruta clara cada día, asegurando que todos los integrantes conocieran sus responsabilidades inmediatas y el estado global del proyecto.
- **Reparto Dinámico de Tareas:** La asignación de responsabilidades se realizó de manera flexible y dinámica, basándose en un análisis de los puntos fuertes y débiles de cada componente del equipo. Esto permitió que los desafíos técnicos (como la configuración de IIS o el tratamiento de datos complejos) fueran abordados por quienes tenían mayor afinidad con la materia, optimizando los tiempos de respuesta.

8.3. Comunicación y Cohesión

La coordinación humana fue el motor del desarrollo técnico:

- **Sincronización de Horarios:** Se establecieron franjas horarias comunes de trabajo conjunto. Estas sesiones fueron vitales para resolver dudas en tiempo real y avanzar en las tareas de integración, donde la interacción entre la base de datos y la interfaz requería la máxima precisión.

- **Comunicación Constante:** Se mantuvo un canal de comunicación abierto y fluido para reportar avances de manera continua. Esta transparencia permitió identificar cuellos de botella de forma temprana, como los problemas iniciales con la vLAN o las extensiones de PHP, y resolverlos mediante la colaboración del grupo.

8.4. Líneas futuras

- **Propuestas de futuro.**
 - **Sistema de Alertas Personalizadas:** Ampliar la lógica de **PHPMailer** para permitir que los usuarios se suscriban a categorías específicas, recibiendo un correo automático en el momento exacto en que se detecte una nueva entrada en el *dataset* que coincida con sus preferencias.
 - **Capacidad de guardar ofertas:** Como evolución natural de la plataforma de Empleo360, se propone la implementación de un sistema de "Ofertas Guardadas". Esta funcionalidad permitiría a los usuarios interesados en las vacantes de empleo realizar un seguimiento de aquellas oportunidades que se ajusten a su perfil sin necesidad de realizar una inscripción inmediata.

9. Bibliografía

Adobe Color. (2026). Obtenido de Adobe Color: <https://color.adobe.com/es/>

Claude. (2026). Obtenido de Claude: <https://claude.ai/>

Drawio. (2026). Obtenido de Drawio: <https://www.drawio.com/>

Gemini. (2026). Obtenido de Gemini: <https://gemini.google.com/>

GitHub. (2026). Obtenido de GitHub: <https://github.com/>

IIS. (2026). Obtenido de Windows: <https://www.microsoft.com/es-es>

JCYL. (2026). *Municipios CYL.* Obtenido de analisis.datosabiertos.jcyl: <https://analisis.datosabiertos.jcyl.es/explore/dataset/registro-de-municipios-de-castilla-y-leon/table/?flg=es-es>

JCYL. (2026). *Ofertas de Empleo.* Obtenido de analisis.datosabiertos.jcyl.es: https://analisis.datosabiertos.jcyl.es/explore/dataset/ofertas-de-empleo/table/?flg=es-es&disjunctive.provincia&sort=fecha_publicacion

Minify. (2026). Obtenido de Minify: <https://www.minifier.org/>

MySQL. (2026). Obtenido de MySQL: <https://www.mysql.com/>

PHP. (2026). Obtenido de PHP: <https://www.php.net/>

PHPMailer. (2026). Obtenido de PHPMailer:
<https://github.com/PHPMailer/PHPMailer>

Tailwind CSS. (2026). Obtenido de Tailwind CSS: <https://tailwindcss.com/>

VSC. (2026). Obtenido de Visual Studio Code: <https://code.visualstudio.com/>

W3Schools. (2026). Obtenido de W3Schools: <https://www.w3schools.com/>

Windows. (2026). Obtenido de Microsoft: <https://www.microsoft.com/es-es>

XAMPP. (2026). Obtenido de XAMPP:
<https://www.apachefriends.org/es/index.html>