



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

CREACIÓN DE APP PARA LA SELECCIÓN DE LOS TURNOS DE TRABAJO EN UNA EMPRESA

Alumno: Alejandro Expósito Pontiveros

Tutor: Ángel Inocencio Aguilera García
Dpto: Departamento de informática

Noviembre, 2022



Universidad de Jaén
Escuela Politécnica Superior de Jaén
Departamento de Informática

Don Ángel Inocencio Aguilera García, tutor del Trabajo Fin de Grado titulado: Creación de APP para la sección de turnos de trabajo en una empresa, que presenta Alejandro Expósito Pontiveros, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, noviembre de 2022

El alumno:

El tutor:

Alejandro Expósito Pontiveros

Ángel Inocencio Aguilera García

ÍNDICE GENERAL

ÍNDICE DE FIGURAS	6
ÍNDICE DE TABLAS	9
Agradecimientos	10
CAPÍTULO 1.- INTRODUCCIÓN	11
1.1. Introducción al proyecto	11
1.2 .Motivación	12
1.3. Objetivos	12
CAPÍTULO 2.- GESTORES DE CONTENIDOS (CMS).....	13
2.1. ¿Qué es un CMS?	13
2.2. Uso de los CMS hoy en día	14
2.3. Ventajas e inconvenientes de los CMS	14
2.4. Comparativa de CMS	15
2.4.1. WordPress	16
2.4.2. TYPO3.....	16
2.4.3. Joomla	17
2.4.4. WIX.....	17
2.5. Elección de CMS para este proyecto	18
CAPÍTULO 3.- HERRAMIENTAS UTILIZADAS.....	19
3.1. XAMPP	19
3.2. Joomla	19
3.3. GanttProject.....	20
3.4. Figma	20
3.5. Selenium IDE	21
CAPÍTULO 4.- INGENIERÍA DEL SOFTWARE.....	22
4.1. Introducción	22
4.1.1. Etapas en Ingeniería del Software	22
4.1.2. Modelos de desarrollo software	23
4.1.3. Elección del modelo	25
4.2. Definición de requisitos.....	26
4.2.1. Requisitos funcionales	26
4.2.2. Requisitos no funcionales	28
4.3. Casos de uso	29
4.3.1. Actores	29
4.3.2. Diagrama de máquina de estados	30
4.3.3. Diagramas de casos de uso	31
4.3.5. Diagramas de secuencia.....	46
4.3.6. Diagrama de clases.....	51
CAPÍTULO 5.- PLANIFICACIÓN.....	52
5.1. Estimación de tiempos.....	52
5.2. Diagrama de Gantt	53

5.3. Estimación de costes.....	54
5.3.1. Recursos Software	54
5.3.2. Recursos hardware	55
5.3.3. Costes ordenadores	56
5.3.4. Recursos humanos	56
5.3.5. Alquiler de oficina	57
5.3.6. Consumo de luz y agua.....	57
5.3.7. Internet	57
5.3.8. Coste total	57
CAPÍTULO 6.- PROTOTIPO	58
6.1. Accesibilidad Web	58
6.2. Diseño interfaz.....	59
CAPÍTULO 7.- IMPLEMENTACIÓN.....	62
7.1. Uso de extensiones	62
7.2. Implementación de cada funcionalidad	68
7.2.1. Registro de la hora de entrada.....	68
7.2.2. Registro de la hora de salida.....	71
7.2.3. Visualización de todos los registros realizados por el empleado.....	71
CAPÍTULO 8.- PRUEBAS	75
8.1. Página de inicio	75
8.2. Login	76
8.3. Página bienvenida	77
8.4. Home	79
8.5. Finalizar jornada laboral	82
8.6. Tabla registros de jornadas	84
8.7. Proyectos y tareas	85
8.8. Estadísticas	87
8.9. Bajas y permisos	89
8.10. Perfil de usuario.....	90
8.11. Modificar contraseña	92
CAPÍTULO 9.- CONCLUSIÓN	94
9.1. Conclusiones	94
9.2. Desarrollos futuros.....	94
ANEXOS.....	95
Configuración e instalación de Xampp	95
Configuración de la base de datos	98
Configuración e instalación de Joomla.....	100
Elementos de Joomla	107
Usuarios	107
Menús.....	112
Artículos.....	116
Extensiones	117

BIBLIOGRAFÍA..... 119

ÍNDICE DE FIGURAS

Ilustración 1: Logo de nuestra aplicación.....	11
Ilustración 2: CMS (Content Management System).....	13
Ilustración 3: Comparativa de CMS	15
Ilustración 4: Logo de WordPress	16
Ilustración 5: Logo de TYPO3	16
Ilustración 6: Logo de Joomla!.....	17
Ilustración 7: Logo de WIX.....	17
Ilustración 8: Logo de XAMPP	19
Ilustración 9: Logo de Joomla 3	19
Ilustración 10: Logo de GANTTProject.....	20
Ilustración 11: Logo de Figma	20
Ilustración 12: Logo de Selenium IDE.....	21
Ilustración 13: Comparativa entre modelos tradicionales.....	23
Ilustración 14: Comparativa entre modelos ágiles.....	24
Ilustración 15: Modelo incremental.....	25
Ilustración 16: Tareas principales del análisis de requisitos	26
Ilustración 17: Actor: Administrador	29
Ilustración 18: Actor: Empleado.....	30
Ilustración 19: Diagrama de máquina de estados de nuestra aplicación	30
Ilustración 20: Diagrama de caso de uso: Iniciar sesión.....	35
Ilustración 21: Diagrama de caso de uso: Cerrar sesión	35
Ilustración 22: Diagrama de caso de uso: Añadir estados al horario	39
Ilustración 23: Diagramas de caso de uso: Finalizar turno	39
Ilustración 24: Diagrama de caso de uso: Ver mi registro de turnos	40
Ilustración 25: Diagrama de caso de uso: Crear ficha de empleado	44
Ilustración 26: Diagrama de caso de uso: Añadir datos esenciales del empleado	44
Ilustración 27: Diagrama de caso de uso: Eliminar ficha de empleado	45
Ilustración 28: Diagrama general de casos de uso	¡Error! Marcador no definido.
Ilustración 29: MVC Joomla	46
Ilustración 30: Diagrama de secuencia: Iniciar sesión.....	47
Ilustración 31: Diagrama de secuencia: Modificar contraseña	48
Ilustración 32: Diagrama de secuencia: Agregar estados al calendario.....	49
Ilustración 33: Diagrama de secuencia: Añadir información adicional	50
Ilustración 34: Diagrama de clases.....	51
Ilustración 35: Diagrama de Gantt.....	53
Ilustración 36: Pc sobremesa	55
Ilustración 37: PC portátil	55
Ilustración 38: Prototipo: Interfaz.....	59
Ilustración 39: Prototipo: Entrada.....	60
Ilustración 40: Prototipo: Login	60
Ilustración 41: Prototipo: Registro de jornada laboral	61
Ilustración 42: Prototipo: Check Out.....	61
Ilustración 43:Extensiones: Limit Active Login.....	62
Ilustración 44:Extensiones: Helix Ultimate.....	63
Ilustración 45:Extensiones: SP Page Builder	64
Ilustración 46:Extensiones: TimeWorked (1).....	64
Ilustración 47:Extensiones: TimeWorked (2).....	65
Ilustración 48:Extensiones: TimeWorked (3).....	65
Ilustración 49:Extensiones: TimeWorked (4).....	66
Ilustración 50:Extensiones: TimeWorked (5).....	66
Ilustración 51:Extensiones: Sourcerer	67
Ilustración 52: Registro de la hora de entrada: Tabla de la base de datos (1).....	68
Ilustración 53: Registro de la hora de entrada: Tabla de la base de datos (2).....	68
Ilustración 54: Registro de la hora de entrada: Código	69
Ilustración 55: Nombre de la base de datos de mi proyecto	70
Ilustración 56: Ejemplo código	70
Ilustración 57: Fila de nuestra tabla.....	70

Ilustración 58: Código checkout.....	71
Ilustración 59: Código tabla básica de Bootstrap	71
Ilustración 60: COUNT de la tabla respecto al usuario actual.....	72
Ilustración 61:Código devolver nombre del usuario actual	72
Ilustración 62: Código devolver nick del usuario actual	73
Ilustración 63: Código devolver fecha y hora de entrada del usuario actual	74
Ilustración 64: Código devolver fecha y hora de salida del usuario actual.....	74
Ilustración 65: Página de Inicio	75
Ilustración 66: Login pruebas automáticas	76
Ilustración 67: Login	76
Ilustración 68: Código test login	77
Ilustración 69: Página de bienvenida pruebas automáticas.....	77
Ilustración 70: Página de bienvenida.....	78
Ilustración 71: Página de bienvenida con sesión activa.....	78
Ilustración 72:Código test página de bienvenida.....	79
Ilustración 73:Home pruebas automáticas.....	79
Ilustración 74: Home.....	80
Ilustración 75: Menú desplegable.....	81
Ilustración 76:Código test home.....	82
Ilustración 77:Finalizar jornada laboral pruebas automáticas	82
Ilustración 78: Finalizar jornada laboral.....	83
Ilustración 79:Código test finalizar jornada laboral	83
Ilustración 80:Tabla de registros de jornadas	84
Ilustración 81: Tabla de registros de jornadas antes de finalizar la jornada actual.....	84
Ilustración 82: Tabla de registros de jornadas después de finalizar la jornada actual	84
Ilustración 83:Código test finalizar jornada laboral	85
Ilustración 84:Proyectos y tareas pruebas automáticas.....	85
Ilustración 85: Proyectos y tareas.....	86
Ilustración 86:Código test proyectos y tareas.....	87
Ilustración 87:Estadísticas pruebas automáticas.....	87
Ilustración 88: Estadísticas	88
Ilustración 89:Código test cambiar estadísticas.....	88
Ilustración 90:Bajas y permisos pruebas automáticas	89
Ilustración 91: Bajas y permisos.....	89
Ilustración 92:Código test bajas y permisos	90
Ilustración 93:Perfil de usuario pruebas automáticas	90
Ilustración 94: Perfil de usuario	91
Ilustración 95: Editar perfil de usuario	91
Ilustración 96:Código test perfil de usuario.....	92
Ilustración 97:Modificar contraseña pruebas automáticas.....	92
Ilustración 98: Cambiar contraseña	93
Ilustración 99: Código test cambiar contraseña.....	93
Ilustración 100: Instalador XAMPP (1).....	95
Ilustración 101: Instalador de XAMPP (2).....	96
Ilustración 102: Instalador de XAMPP (3).....	96
Ilustración 103: Panel de control de XAMPP (1).....	97
Ilustración 104: Panel de control de XAMPP (2).....	97
Ilustración 105: Panel de control de XAMPP (3).....	98
Ilustración 106: phpMyAdmin (1).....	98
Ilustración 107: phpMyAdmin (2).....	99
Ilustración 108: phpMyAdmin (3).....	99
Ilustración 109: Descargar Joomla (1).....	100
Ilustración 110: Descargar Joomla (2).....	100
Ilustración 111: Descargar Joomla (3).....	101
Ilustración 112: Descargar Joomla (4).....	101
Ilustración 113: .zip de Joomla.....	102
Ilustración 114: Carpeta donde se aloja nuestro proyecto	102
Ilustración 115: Instalador Joomla (1).....	103
Ilustración 116: Instalador Joomla (2).....	104
Ilustración 117: Instalador Joomla (3).....	104

Ilustración 118: Instalador Joomla (3).....	105
Ilustración 119: Eliminar carpeta installation.....	105
Ilustración 120: Administrador Joomla.....	106
Ilustración 121: Página de nuestro proyecto.....	106
Ilustración 122: Usuarios Joomla (1).....	107
Ilustración 123: Usuarios Joomla (2).....	107
Ilustración 124: Usuarios Joomla (3).....	110
Ilustración 125: Usuarios Joomla (4).....	110
Ilustración 126: Usuarios Joomla (5).....	111
Ilustración 127: Usuarios Joomla (6).....	111
Ilustración 128: Menús Joomla (1).....	112
Ilustración 129: Menús Joomla (2).....	112
Ilustración 130: Menús Joomla (3).....	113
Ilustración 131: Menús Joomla (4).....	113
Ilustración 132: Menús Joomla (5).....	114
Ilustración 133: Menús Joomla (6).....	114
Ilustración 134: Menús Joomla (7).....	115
Ilustración 135: Menús Joomla (8).....	115
Ilustración 136: Artículos Joomla (1).....	116
Ilustración 137: Artículos Joomla (2).....	116
Ilustración 138: Artículos Joomla (3).....	117
Ilustración 139: Extensiones Joomla (1).....	117
Ilustración 140: Extensiones Joomla (2).....	118
Ilustración 141: Extensiones Joomla (3).....	118

ÍNDICE DE TABLAS

Tabla 1: Ventajas e inconvenientes de los CMS	14
Tabla 2: Ventajas e inconvenientes de WordPress	16
Tabla 3: Ventajas e inconvenientes de TYPO3	16
Tabla 4: Ventajas e inconvenientes de Joomla	17
Tabla 5: Ventajas e inconvenientes de los WIX	17
Tabla 6: Modelos de casos de uso: Gestión de usuarios: Registro	31
Tabla 7: Modelos de casos de uso: Gestión de usuarios: Iniciar sesión	32
Tabla 8: Modelos de casos de uso: Gestión de usuarios: Modificar contraseña por defecto	33
Tabla 9: Modelos de casos de uso: Gestión de usuarios: Cerrar sesión	34
Tabla 10: Modelos de casos de uso: Gestión de horarios: Añadir estados al horario	36
Tabla 11: Modelos de casos de uso: Gestión de turnos: Iniciar turno	37
Tabla 12: Modelos de casos de uso: Gestión de ficha de empleado: Crear ficha de empleado	41
Tabla 13: Modelos de casos de uso: Gestión de ficha de empleado: Añadir datos esenciales	42
Tabla 14: Modelos de casos de uso: Gestión de ficha de empleado: Eliminar ficha de empleado	43
Tabla 15: Estimación de tiempos	52
Tabla 16: Coste total del proyecto	57

Agradecimientos

A mi novia Julia por apoyarme y darme energías en momentos donde no era capaz de afrontar el día a día.

A mi amigo Flo por darme ánimos y ofrecerme su ayuda para entender conocimientos.

A mi tutor por su constante interés y ayuda en este proyecto.

A todas aquellas personas que confiaron en mí.

Y sobre todo a mis padres por enseñarme que con insistencia y ganas se puede lograr llegar a metas cuyos caminos están llenos de obstáculos, por ser tan pacientes y por enseñarme a nunca abandonar pase lo que pase.

CAPÍTULO 1.- INTRODUCCIÓN

1.1. Introducción al proyecto

Desde la pandemia, el mundo ha cambiado y con él nuestras vidas. Como futuro ingeniero informático estoy al tanto de lo que ha supuesto esta pandemia a nuestro mercado. Una de las cosas positivas que trajo esta pandemia es la implementación masiva del teletrabajo al sector. Aún sin el peligro de los contagios por COVID muchas empresas siguen usando este modelo.

Por otro lado, de un tiempo a esta parte, dentro de los cambios en modelos de gestión de las empresas, la jornada laboral flexible ha ido ganando su espacio.

Alcanzar una óptima gestión del tiempo es de vital importancia para nuestro rendimiento. Se trata de que el empleado tenga una mayor libertad a la hora de fijar su horario de entrada/salida a la oficina. Es el propio empleado quien en base a su experiencia se impone mayor flexibilidad para realizar su tarea. [1]

Las principales ventajas que tiene una correcta gestión del tiempo que gire en torno al horario flexible son:

- Mejora el clima laboral dentro de la empresa
- Aumenta el rendimiento y productividad de los trabajadores
- Reducen los niveles de absentismo de la compañía
- No supone un gasto extra para las empresas
- La satisfacción y motivación del empleado se ve incrementada

Este proyecto está pensado para empresas que implementan un sistema de trabajo a distancia y/o flexible. La aplicación quiere ayudar al control de asistencia y control horario en las empresas que introducen estos modelos de trabajo. Otra funcionalidad que he implementado es la de asignar proyectos y tareas a diferentes usuarios y hacer un registro de las actividades y el tiempo usado en estos proyectos y tareas.



Ilustración 1: Logo de nuestra aplicación

1.2. Motivación

Han existido dos principales motivaciones a la hora de escoger este proyecto:

1. **Aprender a usar un CMS y crear aplicaciones con un uso real:** Es muy común hoy en día ver webs de comercios pequeños que usan CMS (WordPress, Joomla...). Desde reservar cita en nuestra peluquería de confianza hasta alquilar una pista de tenis en el polideportivo de nuestra ciudad. Aprender a utilizar uno o varios CMS es una gran oportunidad de negocio, tanto para vivir de ello, creando aplicaciones para pequeños o medianos comercios, hasta para compaginarlo con un trabajo fijo.
2. **La idea del proyecto:** Como he descrito en el apartado anterior, nos encontramos en pleno auge de la jornada flexible y el teletrabajo. No son modalidades de trabajo que lleven estandarizadas demasiado tiempo. El desarrollo de este tipo de aplicaciones está aumentando a un ritmo muy rápido debido a la demanda de las empresas.

1.3. Objetivos

- Estudiar el uso actual de los CMS.
- Comparar los diferentes CMS que existen en el mercado.
- Aprender y utilizar los diferentes elementos que compondrán el CMS elegido para el proyecto.
- Crear una interfaz sencilla y clara.
- El sistema debe ser capaz guardar y enseñar el rango de horas de trabajo del usuario.
- El sistema debe ser capaz de guardar y enseñar el tiempo que el usuario ha dedicado a tareas y proyectos.
- El usuario podrá consultar estadísticas sobre las horas de trabajo que ha dedicado a lo largo de del tiempo.
- Limitar el número de sesiones abiertas por un mismo usuario para evitar problemas de seguridad.

CAPÍTULO 2.- GESTORES DE CONTENIDOS (CMS)

Hace años crear tu propia página web era impensable si no tenías conocimientos muy avanzados de HTML, JavaScript, CSS, ASP o PHP y programar tu web desde cero, pero las cosas han cambiado. Ahora es suficiente con instalar un gestor de contenidos y en mucho menos tiempo se puede conseguir una página web online y accesible desde cualquier parte del mundo. [2]

2.1. ¿Qué es un CMS?

Un CMS ayuda a las empresas a gestionar contenido digital. Equipos enteros pueden utilizar estos sistemas para crear, editar, organizar y publicar contenidos en la web. Actúa como un lugar único para almacenar todo el volumen de contenido que pueda generar y proporciona procesos automatizados para la gestión y creación de nuevos recursos digitales, utilizando herramientas ya creadas o integradas. Los individuos tienen a su disposición diferentes privilegios y responsabilidades en función de sus roles. Un ejemplo de estos roles puede ser, los autores pueden publicar y guardar su trabajo, pero los editores pueden modificarlo y publicarlo. Los administradores pueden hacer todas estas cosas, así como conceder a otras personas de la organización permiso para actualizar o revisar el contenido.

Un CMS ayuda a crear y gestionar los sitios web y el contenido de los mismos con una mínima sobrecarga técnica, para que se pueda dirigir el foco de atención a conseguir una mejor experiencia. Al proporcionar una solución fácil y rentable para la gestión de contenidos, un CMS permite a las empresas gestionar y distribuir sus contenidos sin invertir en un equipo de desarrollo especializado a tiempo completo. [3]



Ilustración 2: CMS (Content Management System)

2.2. Uso de los CMS hoy en día

Un CMS ayuda a crear, gestionar y publicar contenidos en la web. Además, ayuda a mantener los contenidos organizados y accesibles para que puedan utilizarse y reutilizarse de forma eficaz. Hay varios tipos de sistemas de gestión de contenidos disponibles, desde los basados en la nube hasta los CMS headless (donde no se genera frontend), para satisfacer todas las necesidades del público. [3]

No sólo le ofrece una forma de almacenar y gestionar toda la información en una única base de datos de fácil acceso, sino que también ayuda a:

- Ayuda a la colaboración entre equipo tanto externos e internos
- Proporciona una forma fácil y accesible de actualizar el contenido
- Aumenta la visibilidad del contenido
- Mejora la productividad
- Reducción de los costes
- Permite mantener la coherencia entre los contenidos
- Escala con tus necesidades

2.3. Ventajas e inconvenientes de los CMS

Un CMS puede ser ideal para que los pequeños negocios y las empresas gestionen el contenido. Muchos CMS proveen al desarrollador con funcionalidades para construir páginas que permiten arrastrar y soltar elementos lo que hacen la vida aún más fácil. Ahora todo lo que la empresa necesita es un tema o un diseño de interfaz de usuario. Aunque los sistemas CMS son soluciones perfectas para muchas empresas, no cumplen con los requisitos de todas ellas, a continuación, se muestran algunas de las ventajas e inconvenientes: [4]

Ventajas	Inconvenientes
Desarrollo rápido	Dependencia de plugins y widgets
Menos código en el backend	Velocidad de carga del sitio web
Fácil para una persona sin conocimientos técnicos	El mantenimiento
Seguro	Limitación en los requisitos funcionales

Tabla 1: Ventajas e inconvenientes de los CMS

2.4. Comparativa de CMS

Antes de elegir un CMS es importante hacer una comparativa entre los distintos CMS que existen en el mercado. Hay que tener en cuenta que una vez hayamos creado nuestra aplicación con un sistema, será difícil migrarse a otra solución. Por lo tanto, se debe elegir no solo en función de nuestras necesidades actuales, sino también anticipar la evolución que pueda tener nuestra aplicación. [5]

En la **ilustración 3** podemos observar una comparativo a razón de uso de algunos de los distintos CMS que nos ofrece el mercado: [6]

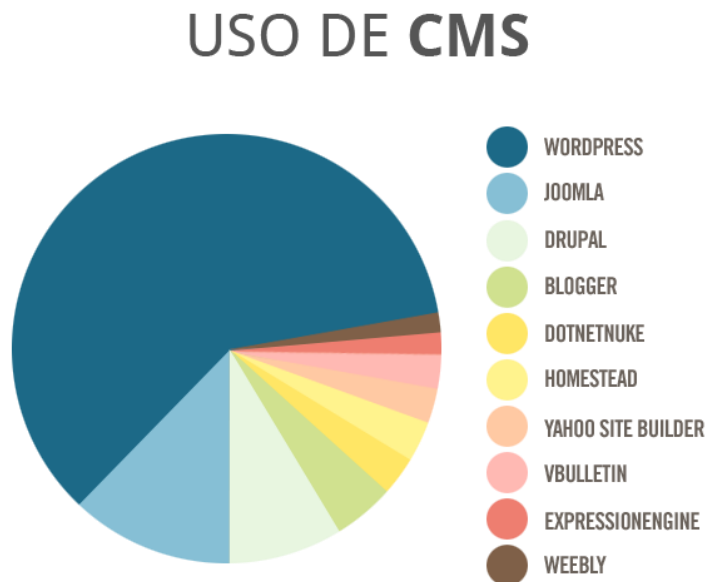


Ilustración 3: Comparativa de CMS

A continuación, vamos a estudiar las características de los CMS más populares teniendo en cuentas sus pros y sus contras y eligiendo la que mejor se pueda adaptar a nuestro proyecto:

2.4.1. WordPress

El CMS más utilizado. Aunque fue concebido al principio como un sistema de blogs, gracias a sus diversas extensiones, se convierte en un Software de gestión de contenidos muy completo.



Ilustración 4: Logo de WordPress

Ventajas	Inconvenientes
Código abierto	Consumo de recursos alto
Gratuito	CMS más atacado
Fácil instalación	Baja velocidad
Gran variedad de funciones	Plugin inestables
Gran comunidad de soporte	Las funciones de CMS requieren plugin adicionales

Tabla 2: Ventajas e inconvenientes de WordPress

2.4.2. TYPO3

Basado en PHP. Al contrario que WordPress, TYPO3 nació para ser un CMS. Cuenta con el apoyo de un equipo técnica especializado que se ocupa de su continuo desarrollo y mantenimiento. Fuerte expansión en otros países.



Ilustración 5: Logo de TYPO3

Ventajas	Inconvenientes
Comunidad de expertos	Requiere muchos requisitos del servidor para funcionar
Buena variedad de funciones	Instalación compleja
Flexible	Aprendizaje más lento que otros CMS debido a su complejidad
CMS más avanzado y potente	Rendimiento no óptimo con alojamiento compartido (contar al menos con un VPS o Servidor Cloud)

Tabla 3: Ventajas e inconvenientes de TYPO3

2.4.3. Joomla

Joomla es un CMS de código abierto, que destaca por su flexibilidad y su gran cantidad de plantillas. Es una opción muy interesante para sitios web complejos o con un nivel alto de personalización.



Ilustración 6: Logo de Joomla!

Ventajas	Inconvenientes
Instalación sencilla	Más enfocado en proyectos simples
Software libre	Considerado vulnerable, pero más seguro que WordPress
Muchas plantillas	Configuración de plantillas y módulos poco intuitivas
Puede usar https de forma sencilla	Rendimiento no óptimo con alojamiento compartido (contar al menos con un VPS o Servidor Cloud)

Tabla 4: Ventajas e inconvenientes de Joomla

2.4.4. WIX

CMS destacado por su sencillez de cara a principiantes. Uso muy simple debido a sus herramientas de arrastrar y soltar y tener muchas plantillas prediseñadas y aptas para móviles.



Ilustración 4: Logo de WIX

Ventajas	Inconvenientes
Sencillez	Requiere muchos requisitos del servidor para funcionar
Gratuita	Acceso a código muy limitado
Variedad de plantillas	No puedes cambiar de plantilla una vez elegida una
Apto para móviles	No puede utilizarse para tiendas online

Tabla 5: Ventajas e inconvenientes de los WIX

2.5. Elección de CMS para este proyecto

He decidido elegir Joomla para trabajar en este proyecto por estos motivos [6]:

- Es más seguro que WordPress.
- La oportunidad de aprender CMS que son una alternativa a WordPress.
- Joomla es uno de los tres mejores gestores de contenido de código abierto.
- Respaldado por una increíble comunidad.
- Activo proceso de desarrollo.
- Una opción muy elegida entre los gobiernos (La Junta de Andalucía ofrece muchos portales desarrollados en Joomla).
- Al ser una tecnología de código abierto los clientes no están atados a una única empresa de desarrollo.

Como he citado arriba, un motivo de peso a la hora de elegir Joomla como el CMS a usar para mi proyecto ha sido aprender y trabajar con un CMS que, aunque es uno de los más usados y conocidos, no es el habitual hoy en día. Joomla está constantemente en proceso de desarrollo y nunca podemos saber si algunas de sus versiones podrían traer novedades al mercado de los CMS.

Por otro lado, es más seguro que WordPress que podríamos decir que es su rival directo. La seguridad en aplicaciones de este perfil es algo fundamental y algo que debemos tener presente antes de realizar este tipo de proyectos.

Otro motivo es que Joomla nos permite configurar HTTPS desde el mismo administrador de manera muy sencilla y nos permite elegir entre establecer HTTPS solo en el área de administración o en todo el sitio.

CAPÍTULO 3.- HERRAMIENTAS UTILIZADAS

Mencionaré y describiré brevemente las diferentes herramientas utilizadas para la elaboración de este proyecto

3.1. XAMPP

XAMPP es uno de los servidores web multiplataforma más utilizados, que ayuda a los desarrolladores a crear y probar sus programas en un servidor web local. Fue desarrollado por Apache Friends y su código fuente nativo puede ser revisado o modificado por la audiencia.

Es una plataforma que proporciona un entorno adecuado para probar y verificar el funcionamiento de proyectos basados en Apache, Perl, base de datos MYSQL y PHP a través del sistema del propio host. [7]



Ilustración 8: Logo de XAMPP

3.2. Joomla

Joomla es un sistema de gestión de contenido dinámicos que permite crear sitios de gran relevancia con una interactividad de primera, acabado profesional y gran eficiencia. Su principal gestión como CMS es la puesta en marcha y gestión de contenidos en línea. Para este proyecto he decidido usar la versión Joomla 3 ya que hay una mayor cantidad de plugins en esta. [8]



Ilustración 9: Logo de Joomla 3

3.3. GanttProject

GanttProject es una herramienta de escritorio libre y multiplataforma que permite la realización de diagramas de GANTT a fin de planificar un proyecto y gestionar los recursos.

Es una herramienta muy completa a la hora de planificar un proyecto, ya que permite una visualización absoluta del mismo. Todo queda bajo el control del programa, desde los recursos necesarios en materia personal, así como los días festivos, hasta dividir el proyecto en un árbol de tareas y asignar a cada uno los recursos oportunos. [9]



Ilustración 10: Logo de GANTTProject

3.4. Figma

Figma es una aplicación para diseñar interfaces que se ejecuta en el navegador, pero en realidad es mucho más que eso. Me atrevería a decir que es probablemente la mejor aplicación para proyectos de diseño colaborativo en equipos. [10]



Ilustración 11: Logo de Figma

3.5. Selenium IDE

Selenium IDE es una herramienta muy útil que nos permite simular interacciones de usuario con el sistema.

Estas interacciones pueden encadenarse, modificarse, repetirse y exportarlas a formato de código.



Ilustración 12: Logo de Selenium IDE

CAPÍTULO 4.- INGENIERÍA DEL SOFTWARE

4.1. Introducción

La ingeniería del software es una disciplina que implica el uso de estructuras, herramientas y técnicas para construir y mantener software de calidad.

Ingeniería de Software es la aplicación práctica del conocimiento científico en el diseño y construcción de programas de computadora y la documentación asociada requerida para desarrollar, operar (funcionar) y mantenerlos. Se conoce también como desarrollo de software o producción de software. [11]

4.1.1. Etapas en Ingeniería del Software

El proceso de ingeniería del software consta de 7 etapas, siendo las 5 primeras fundamentales y la etapa 6 está presente en cada una de las anteriores. Mencionaremos brevemente en qué consisten cada una de las etapas: [12]

- 1. Análisis de requerimientos:** Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios.
- 2. Especificación:** Es la tarea de describir detalladamente el software a ser escrito. Se describe el comportamiento esperado del software y su interacción con los usuarios y los sistemas.
- 3. Diseño y arquitectura:** Describir cómo funcionará de forma general sin entrar en detalles. Incorporamos consideraciones de la implementación tecnológica (hardware, red...).
- 4. Programación:** Se traduce el diseño a código. Una especificación o diseño incompletos pueden exigir que tareas propias de estas etapas tengan que realizarse en esta.
- 5. Prueba:** En esta etapa comprobamos que el software realice correctamente las tareas indicadas en la especificación.
- 6. Documentación:** Realización del manual de usuario, y posiblemente un manual técnico con el propósito de un mantenimiento futuro y ampliaciones del sistema.
- 7. Mantenimiento:** En esta etapa se resuelven errores y se mejoran las funcionalidades.

4.1.2. Modelos de desarrollo software

Antes de elegir una metodología de trabajo para nuestro proyecto, es necesario hacer un repaso de las diferentes metodologías que existes y ver cuál de ellas se adaptar mejor tanto a nuestro proyecto como a nuestro ritmo de trabajo.

A continuación, pasamos a valorar cada uno de ellos:

Modelo	Ventajas	Desventajas
En cascada	<p>Es un modelo lineal por lo que es más simple de implementar</p> <p>Cantidad de recursos mínimos</p> <p>Después de una etapa importante se realizan pruebas</p>	<p>No se puede volver atrás en caso de error, es rígido a modificaciones</p> <p>Si el cliente no tiene una idea clara producirá mucha confusión</p>
Incremental	<p>Se divide en múltiples incrementos</p> <p>El cliente puede ver el producto antes de estar acabado</p>	<p>Una vez comenzado el incremento no se pueden modificar los requisitos</p> <p>Requiere de mucha planificación, tanto administrativa como técnica</p> <p>Requiere de metas claras para conocer el estado del proyecto</p>
Desarrollo Rápido de Aplicaciones (DRA)	<p>Mayor velocidad que el desarrollo en cascada al reutilizar componentes</p> <p>Entrega temprana al cliente</p> <p>Ciclos de desarrollo más pequeños</p>	<p>Tiene inconvenientes para proyectos grandes</p> <p>Progreso más difícil de medir</p> <p>Alto costo de herramientas integradas y equipo necesario</p>

Ilustración 13: Comparativa entre modelos tradicionales

Como podemos observar en la **ilustración 13** existen varios **modelos de desarrollo tradicionales** conocidos. [13]

- **Modelo en cascada:** Modelo simple y eficaz pero que no nos permite volver a etapas de desarrollo anteriores.
- **Modelo incremental:** Podemos ir entregando diferentes versiones incompletas del producto. Una vez empezado el incremento no podemos modificar sus requisitos hasta que este haya concluido.
- **Desarrollo Rápido de Aplicaciones (DRA):** Este modelo nos permite una mayor velocidad. Se basa en el modelo en cascada y en la creación de componentes reutilizables.

Metodología	Ventajas	Desventajas
Scrum	Resultados anticipados	Funciona mejor con equipos reducidos
	Flexibilidad y adaptación a los cambios en los requisitos	Requiere precisa definición de tareas y plazos Exige una alta formación
Kanban	Permite disminuir o eliminar stock entre procesos	No es una técnica específica del desarrollo software
	Facilita el control de la producción Permite un sistema de producción flexible según la demanda	
TDD	Mayor simplicidad en el diseño	Difícil de implementar la capa de presentación
	Diseño enfocado en las necesidades	El hecho de pasar pruebas no significa que no existan errores
	Menos tiempo invertido en depuración de errores	Pronunciada curva de aprendizaje

Ilustración 14: Comparativa entre modelos ágiles

La **ilustración 14** nos presenta varios de los **modelos de desarrollo ágiles** más conocidos. Quizás estos modelos sean menos interesantes para usarlos en este proyecto puesto que son modelos pensados para aplicarse en proyectos realizados por grupos, sin embargo, es interesante mencionarlas y valorarlas ya que podemos fusionar varios procesos para mitigar estas limitaciones individuales. [13]

- **Scrum:** Permite un rápido desarrollo del proyecto y otorga gran flexibilidad siendo una metodología muy tolerante a cambios durante la fase de desarrollo del proyecto.
- **Kanban:** Control de la producción usando herramientas visuales. No enfocado totalmente en desarrollo de software.
- **TDD:** Enfocado solamente en las necesidades. Es un proceso de desarrollo que consiste en **codificar pruebas, desarrollar y refactorizar** de forma continua el código construido.

4.1.3. Elección del modelo

Me he decantado por usar el **modelo incremental** por estas razones:

- Es un modelo flexible a los cambios. En caso de que se propicien cambios en los requisitos o se incorporen nuevos, este modelo es el idóneo.
- Adaptable a las necesidades que vayan surgiendo.
- El modelo de entrega por incrementos hace el desarrollo del proyecto más ameno.
- Existe un bajo riesgo de fallo total del proyecto.

El modelo incremental es un enfoque intermedio que nos ofrece una mayor flexibilidad a la hora de las entregas de los incrementos. Enfocándonos en este proyecto, al principio del mismo identificamos a grandes rasgos los servicios que proporcionará el sistema. Este modelo me ha permitido poder llevar a cabo análisis adicionales de requerimientos para incrementos posteriores (pero nunca para el que estaba desarrollando en el momento). Para el cliente se trata de un modelo que le permite integrar los incrementos tan pronto estén finalizados y poder acoplar los nuevos conforme vayamos completándolos. Es una forma de que el cliente saque provecho antes de acabar el proyecto en su totalidad. [14]

Podemos observar el ciclo de vida de este modelo en la **ilustración 15**:



Ilustración 15: Modelo incremental

4.2. Definición de requisitos

Los requisitos de un proyecto de software son las características, funciones y restricciones que debe cumplir el producto final.

Dicha labor es muy importante porque es la base del trabajo posterior de diseño e implementación del software. Una especificación pobre de requisitos, causará efectos negativos en los clientes, aun cuando todo el trabajo de diseño y programación que se realice posteriormente, sea técnicamente correcto [15].

En la **ilustración 16** podemos observar las cinco tareas principales del análisis de requisitos, las cuales son independientes de la metodología que se utilice:

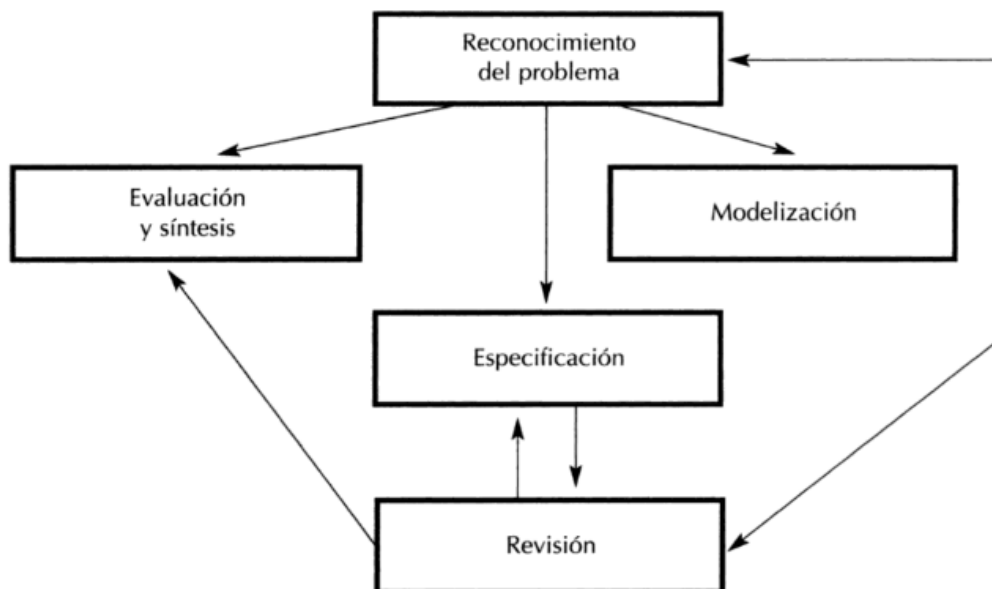


Ilustración 16: Tareas principales del análisis de requisitos

4.2.1. Requisitos funcionales

Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particulares y de cómo se debe comportar en situaciones particulares. En algunos casos, los requerimientos funcionales de los sistemas también pueden declararse explícitamente lo que el sistema no debe hacer. [14]

1. Gestión de usuarios:

-Registro de usuarios:

- El administrador registra a los empleados.

-Modificar contraseña:

- Los usuarios podrán cambiar su contraseña.

-Eliminar empleado:

- El administrador podrá eliminar todos los datos de un empleado.

-Acceder a los datos de un usuario:

- El administrador puede consultar los datos de un empleado.

2. Gestión de turnos:

-Iniciar turno:

- El empleado inicia su jornada laboral en la aplicación.

-Solicitar cambio de turno:

- El empleado solicita un cambio de turno con otro empleado.

-Aceptar cambio de turno:

- El administrador confirma el cambio de turno entre dos empleados.

-Finalizar turno:

- El empleado finaliza su jornada laboral.

3. Gestión de horarios:

-Mostrar horarios de un empleado:

- El sistema debe ser capaz de mostrar los horarios asignados a un empleado.

-Mostrar horas trabajadas por un empleado:

- El sistema debe ser capaz de mostrar las horas que lleva trabajadas un empleado.

-Añadir estados al horario:

- El sistema debe permitir a los empleados marcar en un cuadrante de horarios si el empleado se encuentra de baja, descanso, permiso o vacaciones.
-

4. Gestión ficha de empleado:

-Crear ficha de empleado:

- El sistema crea por defecto una ficha de empleado en cada registro de nuevo empleado.

-Subir foto de perfil:

- El sistema debe permitir al empleado subir una foto de perfil.

-Añadir datos del empleado:

- El empleado podrá añadir información asociada al trabajo y puesto que realiza en la empresa.

-Eliminar ficha de empleado:

- El administrador puede eliminar fichas de empleado.

5. Gestión de estadísticas de empleados

-Mostrar estadísticas de los empleados:

- El sistema debe ser capaz de mostrar las horas laborales realizadas en un periodo de tiempo frente a las horas laborales esperadas.

4.2.2. Requisitos no funcionales

Los requisitos no funcionales son aquellos requerimientos que no se refieren directamente a las funciones específicas que proporciona el sistema, sino las propiedades emergentes de éste como la fiabilidad, el tiempo de respuestas y la capacidad de almacenamiento.

Los requerimientos no funcionales no solo se refieren al sistema software a desarrollar. Alguno de estos requerimientos puede restringir el proceso que se debe utilizar para desarrollar el sistema. Es importante puntualizar que estos requerimientos nacen de las necesidades del usuario, debido a las restricciones en el presupuesto, a las políticas de la organización, a la necesidad de interoperabilidad con otros sistemas o a factores externos.
[14]

A continuación, describiré brevemente los requisitos no funcionales de nuestra aplicación:

- La interfaz debe ser lo más intuitiva posible.
- Disponibilidad en cualquier momento del día
- **Confidencialidad:** No todos los empleados pueden acceder a la información, estadísticas y cambio de horarios de otros empleados.
- **Seguridad:** La aplicación puede tener datos muy sensibles sobre la empresa por lo tanto es crucial mantener unos buenos niveles de seguridad.
- La aplicación tiene que ser portable en iOS y Android,
- **Escalabilidad:** La aplicación debe estar preparada para aguantar cualquier volumen de usuarios que se le presente.

4.3. Casos de uso

Los casos de uso forman la base para escribir los requisitos del usuario y, en particular, ayudar a los desarrolladores a escribir los diferentes puntos de vista del usuario respecto al sistema. Los casos de uso proporcionan más visión holística y dinámica de los requisitos del usuario.

Los casos de uso se pueden escribir en una amplia variedad de formas y en diferentes niveles.

La idea es dar descripciones de alto nivel de las funciones básicas y no describir al usuario detalles de la interfaz. [16]

4.3.1. Actores

Pueden existir varios actores en el sistema. Algunos actores pueden ser más importante que los demás en función de la importancia de sus roles. Después de identificar a cada actor es importante describir las actividades realizadas por cada uno de ellos.

Administrador: Se encarga de administrar el sistema. Es capaz de gestionar toda la información de un empleado. Es el encargado de aceptar o rechazar las solicitudes que haga un usuario.



Ilustración 17: Actor: Administrador

Empleado: Puede modificar su foto de perfil. Puede cambiar algunos datos personales y puede solicitar algunas acciones en su horario: cambio de turno, solicitar vacaciones, declarar en el sistema que está de permiso...



Ilustración 18: Actor: Empleado

4.3.2. Diagrama de máquina de estados

Este tipo de diagramas permite detallar el funcionamiento interno del sistema o de alguna de sus partes.

Debemos crear un diagrama de máquina de estados que represente el funcionamiento general de nuestro sistema. [17]

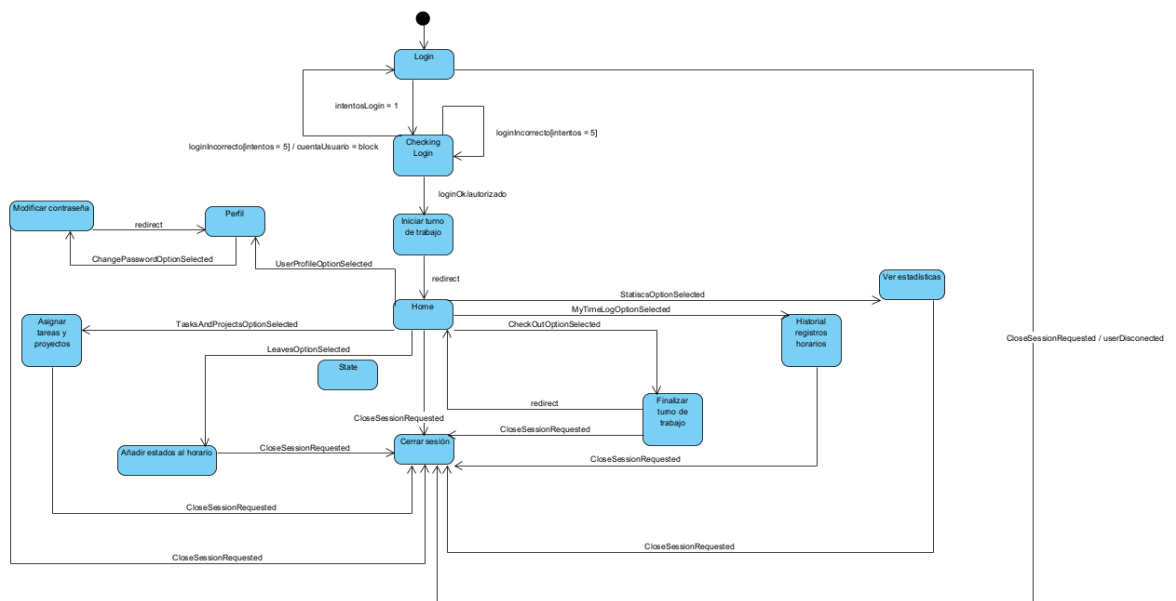


Ilustración 19: Diagrama de máquina de estados de nuestra aplicación

4.3.3. Diagramas de casos de uso

Gestión de usuarios:

CASO DE USO	Registro
Resumen	Registro de un empleado en el sistema
Actores	Administrador
Prerrequisito	El empleado de la empresa aún no está registrado en el sistema.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El moderador usa la opción 'Registro'. 2. El sistema muestra el formulario de registro. 3. El moderador rellena los datos requeridos. 4. El moderador usa la opción 'Registrar'. 5. El sistema comprueba si el formulario es correcto y completo. 6. El sistema registra al empleado.
Propósito	Registrar una cuenta en el sistema.
Alternativa 4.A El registro es incorrecto o incompleto	<ol style="list-style-type: none"> 4.A.1. El sistema muestra el formulario de registro junto al error que se detecta. 4.A.2 Volver al paso 2.
Excepciones	

Tabla 6: Modelos de casos de uso: Gestión de usuarios: Registro

CASO DE USO	Iniciar sesión
Resumen	Entrar en la aplicación
Actores	Empleado
Prerrequisito	El empleado de la empresa está registrado en el sistema.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El empleado usa la opción 'Login'. 2. El sistema lleva al usuario al formulario de acceso. 3. El empleado rellena el formulario con su usuario y su contraseña. 4. El sistema comprueba si el formulario es correcto y completo. 5. El sistema lleva al empleado a la pantalla principal.
Propósito	Acceder con una cuenta en el sistema.
Alternativa 4.A Datos de inicio de sesión incorrectos	<ol style="list-style-type: none"> 4.A.1. El sistema muestra el formulario de inicio de sesión junto al error que se detecta. 4.A.2 Volver al paso 2.
Excepciones	

Tabla 7: Modelos de casos de uso: Gestión de usuarios: Iniciar sesión

CASO DE USO	Modificar contraseña por defecto
Resumen	Cambiar contraseña
Actores	Empleado
Prerrequisito	El empleado de la empresa está registrado en el sistema.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El empleado accede a su perfil. 2. El empleado pulsa en el botón 'Cambiar de Contraseña'. 3. El usuario introduce la nueva contraseña. 4. El sistema comprueba si la contraseña Cumple los requisitos de contraseña segura. 5. El sistema recarga la pantalla de perfil.
Propósito	Cambiar contraseña asignada a un empleado.
Alternativa 4.A La contraseña no cumple los requisitos	<ol style="list-style-type: none"> 4.A.1. El sistema muestra el formulario de cambio de contraseña. 4.A.2 Volver al paso 3.
Excepciones	

Tabla 8: Modelos de casos de uso: Gestión de usuarios: Modificar contraseña por defecto

CASO DE USO	Cerrar sesión
Resumen	El usuario cierra sesión
Actores	Empleado
Prerrequisito	El empleado de la empresa tiene la sesión iniciada.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El empleado pulsa la opción 'cerrar sesión'. 2. El sistema cierra la sesión del usuario. 3. El sistema redirige a la página principal.
Propósito	Cerrar la sesión activa del usuario.
Alternativa	
Excepciones	

Tabla 9: Modelos de casos de uso: Gestión de usuarios: Cerrar sesión

En el caso de uso de la **ilustración 20** se muestra las acciones que tiene que hacer el empleado para iniciar sesión en el sistema.

Una vez dentro del sistema el usuario puede salir del sistema desde cualquier pantalla. Vemos esto en la **ilustración 21**

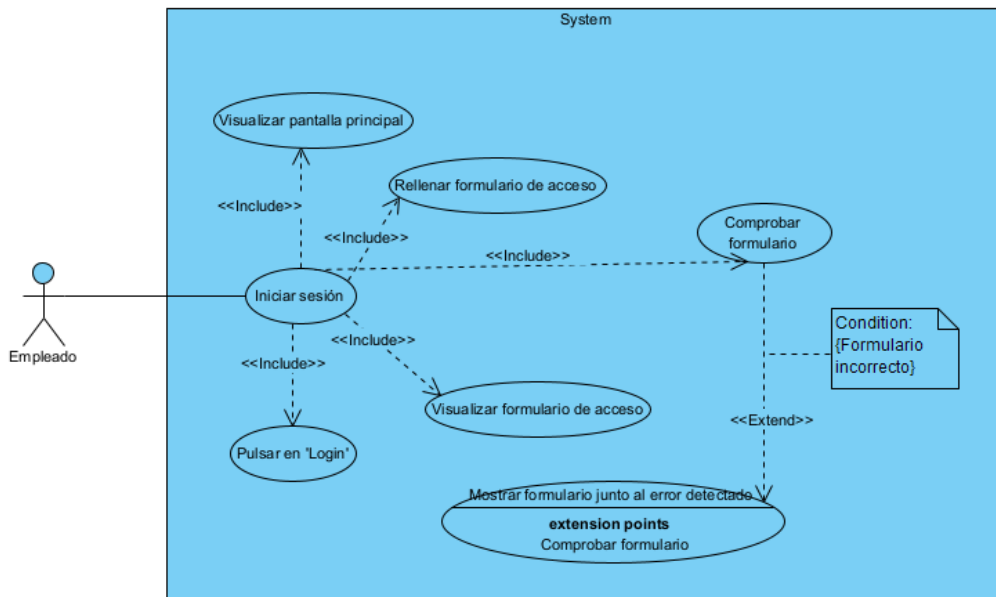


Ilustración 20: Diagrama de caso de uso: Iniciar sesión

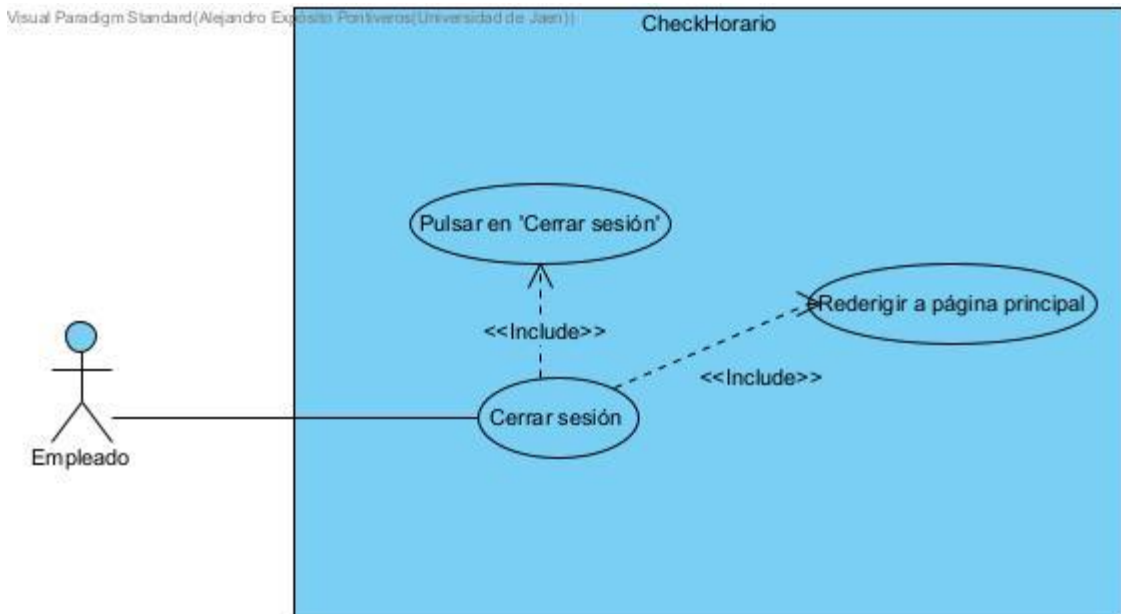


Ilustración 21: Diagrama de caso de uso: Cerrar sesión

Gestión de horarios:

CASO DE USO	Añadir estados al horario
Resumen	El empleado agrega estados en su horario laboral
Actores	Empleado
Prerrequisito	
Operaciones Básicas	<ol style="list-style-type: none"> 1. El empleado pulsa el botón 'calendario'. 2. El empleado pulsa un día en el calendario. 3. El empleado rellena el formulario con el estado y la franja horaria. 4. El empleado pulsa el botón 'aceptar'.
Propósito	Informar a la empresa de una ausencia de un empleado.
Alternativa	
Excepciones	

Tabla 10: Modelos de casos de uso: Gestión de horarios: Añadir estados al horario

CASO DE USO	Iniciar turno
Resumen	El usuario inicia el turno de trabajo
Actores	Empleado
Prerrequisito	El empleado de la empresa ha iniciado sesión en el sistema.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El empleado pulsa en el botón de inicio de jornada laboral. 2. El sistema activa el cronometro que hace correr el tiempo.
Propósito	Iniciar contador de tiempo de jornada laboral.
Alternativa	
Excepciones	

Tabla 11: Modelos de casos de uso: Gestión de turnos: Iniciar turno

CASO DE USO	Finalizar turno
Resumen	El usuario termina su turno de trabajo
Actores	Empleado
Prerrequisito	El empleado de la empresa ha iniciado el turno de trabajo.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El empleado pulsa en el botón 'Iniciar jornada laboral'. 2. El empleado pulsa en el botón 'Finalizar jornada laboral'.
Propósito	Finalizar contador de tiempo de trabajo.
Alternativa	
Excepciones	

Tabla 11: Modelos de casos de uso: Gestión de turnos: Finalizar turno

En el caso de uso de la **ilustración 22** podemos ver como el empleado puede agregar diferentes estados a su horario.

En el caso de uso de la **ilustración 23** solo necesita de una acción para finalizar su turno de trabajo.

En el caso de uso de la **ilustración 24** podemos ver como el empleado puede acceder a ver sus estadísticas.

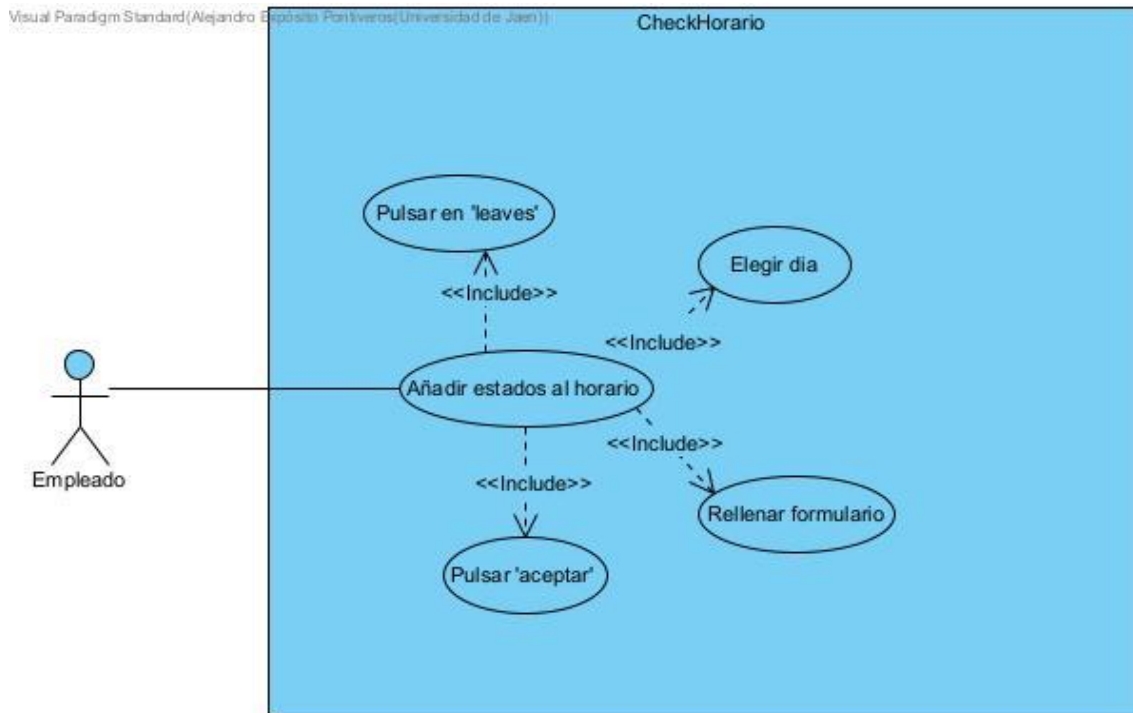


Ilustración 22: Diagrama de caso de uso: Añadir estados al horario

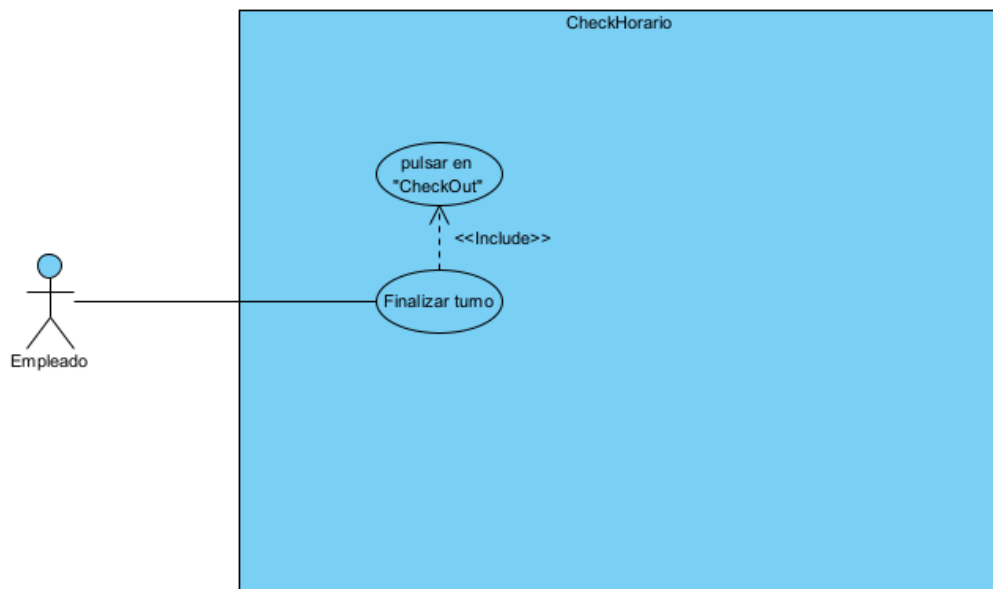


Ilustración 23: Diagramas de caso de uso: Finalizar turno

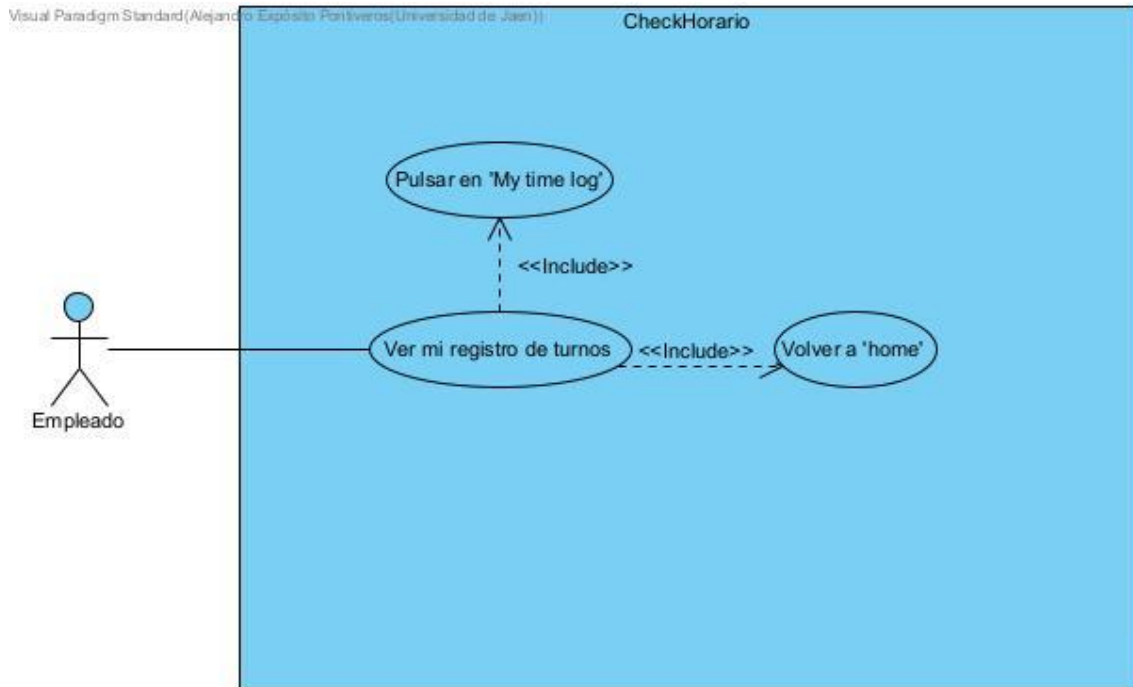


Ilustración 24: Diagrama de caso de uso: Ver mi registro de turnos

Gestión de ficha de empleado:

CASO DE USO	Crear ficha de usuario
Resumen	El administrador crea una nueva ficha de usuario
Actores	Administrador
Prerrequisito	
Operaciones Básicas	<ol style="list-style-type: none"> 1. El administrador pulsa en 'empleados'. 2. El administrador pulsa en la opción 'crear nueva ficha de empleado'. 1. El administrador rellena el formulario con los datos esenciales del empleado. 4. El administrador pulsa la opción de 'guardar'
Propósito	Crear una ficha visible con los datos del empleado.
Alternativa	
Excepciones	

Tabla 12: Modelos de casos de uso: Gestión de ficha de empleado: Crear ficha de empleado

CASO DE USO	Añadir datos esenciales del usuario
Resumen	El administrador añade datos a la ficha de empleado
Actores	Administrador
Prerrequisito	El administrador ha creado con anterioridad la ficha del empleado.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El administrador pulsa en 'empleados'. 2. El administrador pulsa en la ficha del empleado. 1. El administrador hace click en la opción 'editar'. 2. El administrador rellena y edita los distintos campos disponibles.
Propósito	Asegurarnos que la ficha del empleado tenga todos los datos esenciales y asegurarnos que cualquier dato esencial (Ej.: domicilio) esté siempre actualizado.
Alternativa	
Excepciones	

Tabla 13: Modelos de casos de uso: Gestión de ficha de empleado: Añadir datos esenciales

CASO DE USO	Eliminar ficha de empleado
Resumen	El administrador puede eliminar la ficha de un empleado
Actores	Administrador
Prerrequisito	El administrador ha creado con anterioridad la ficha del empleado.
Operaciones Básicas	<ol style="list-style-type: none"> 1. El administrador pulsa en 'empleados'. 2. El administrador pulsa en la ficha del empleado. 3. El administrador hace click en la opción 'borrar'. 4. El sistema muestra un mensaje emergente en el que indica al administrador si está seguro de realizar esta acción. 5. El administrador pulsa en 'si'.
Propósito	Eliminar fichas de empleados que ya no trabajen en la empresa.
Alternativa	<ol style="list-style-type: none"> 5.A.1. El administrador hace click en 'no'. 5.A.2. El sistema redirige al administrador a la pantalla de empleados.
Excepciones	

Tabla 14: Modelos de casos de uso: Gestión de ficha de empleado: Eliminar ficha de empleado

En el caso de uso de la **ilustración 25** podemos ver como el Administrador crea una ficha de empleado.

En el caso de uso de la **ilustración 26** podemos ver como el Administrador añade los datos esenciales del empleado.

En el caso de uso de la **ilustración 27** podemos ver que acciones debe tomar el Administrador para eliminar la ficha de un empleado.

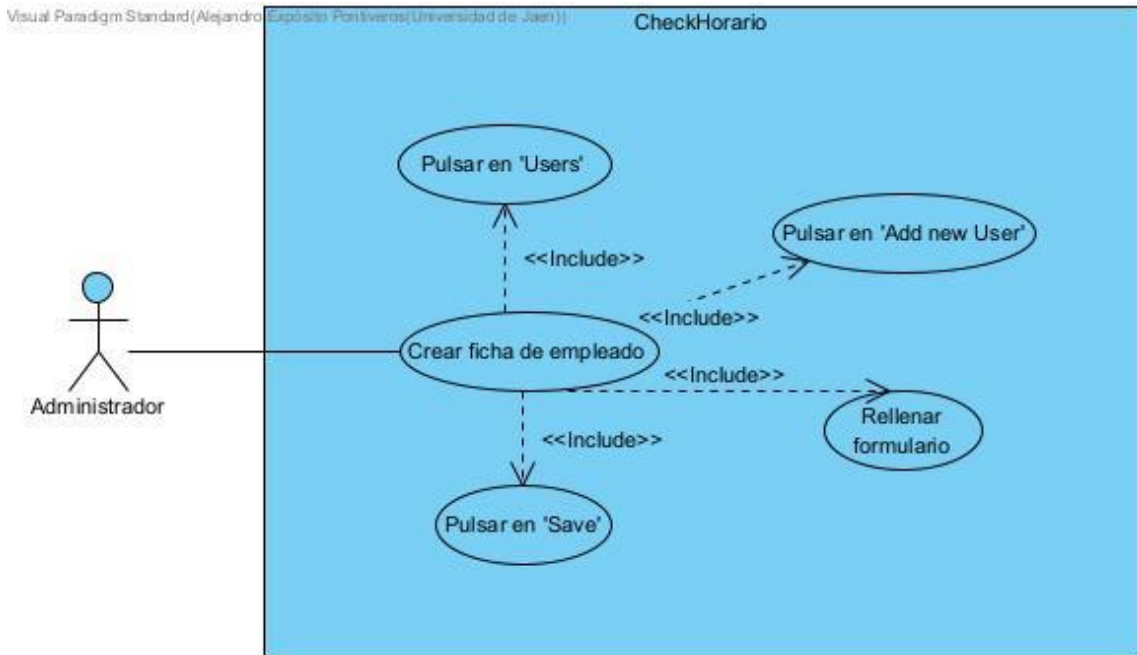


Ilustración 25: Diagrama de caso de uso: Crear ficha de empleado

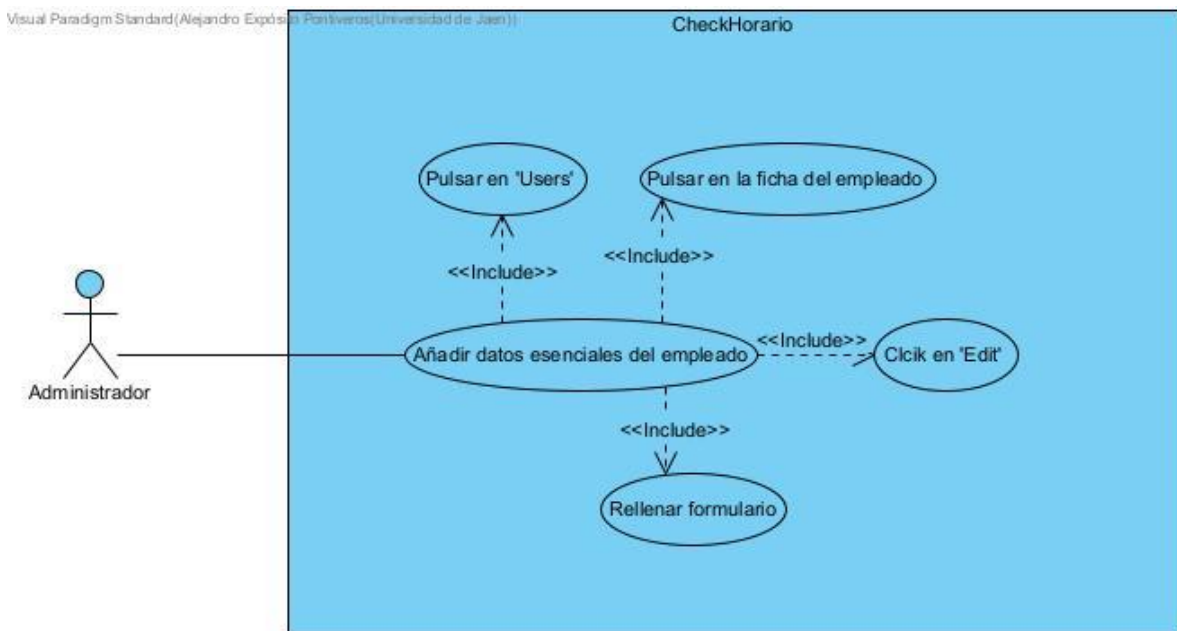


Ilustración 26: Diagrama de caso de uso: Añadir datos esenciales del empleado



Ilustración 27: Diagrama de caso de uso: Eliminar ficha de empleado

4.3.5. Diagramas de secuencia

Un diagrama de secuencia podríamos es similar a una tabla cuyas columnas son relacionadas con objetos individuales y cuyas filas están relacionadas con los mensajes que se intercambian entre los objetos. Los objetos que participan en la interacción están dispuestos en el eje x comenzando por la izquierda con los objetos que inician la interacción y procediendo a la derecha. Los mensajes que se intercambian entre los objetos se ordenan en tiempo a lo largo del eje y.

Los diagramas de secuencia tienen dos características que lo distinguen de otros diagramas:

- Línea de vida de los objetos
- Foco de control

Una línea de vida de objeto es una línea vertical discontinua que representa la existencia de un objeto durante un periodo de tiempo,

El foco de control representa el periodo de tiempo durante el cual el objeto ejecuta su acción. [18]

Joomla usa un modelo vista-controlador.

Cuando Joomla se inicia para procesar una solicitud de un usuario, como un GET para una página en particular, o un POST que contiene datos de formulario, una de las primeras cosas que hace Joomla es analizar la URL para determinar qué componente será responsable de procesar la solicitud, y entregar el control a ese componente. Podemos ver el modelo que usa Joomla en la **ilustración 29**.

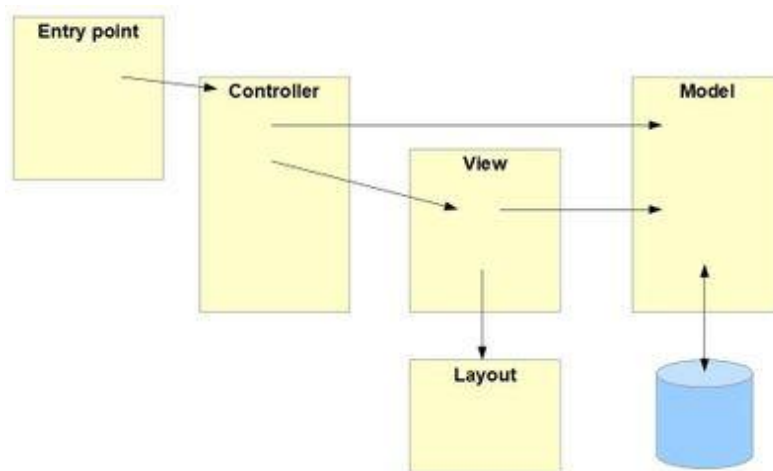


Ilustración 29: MVC Joomla

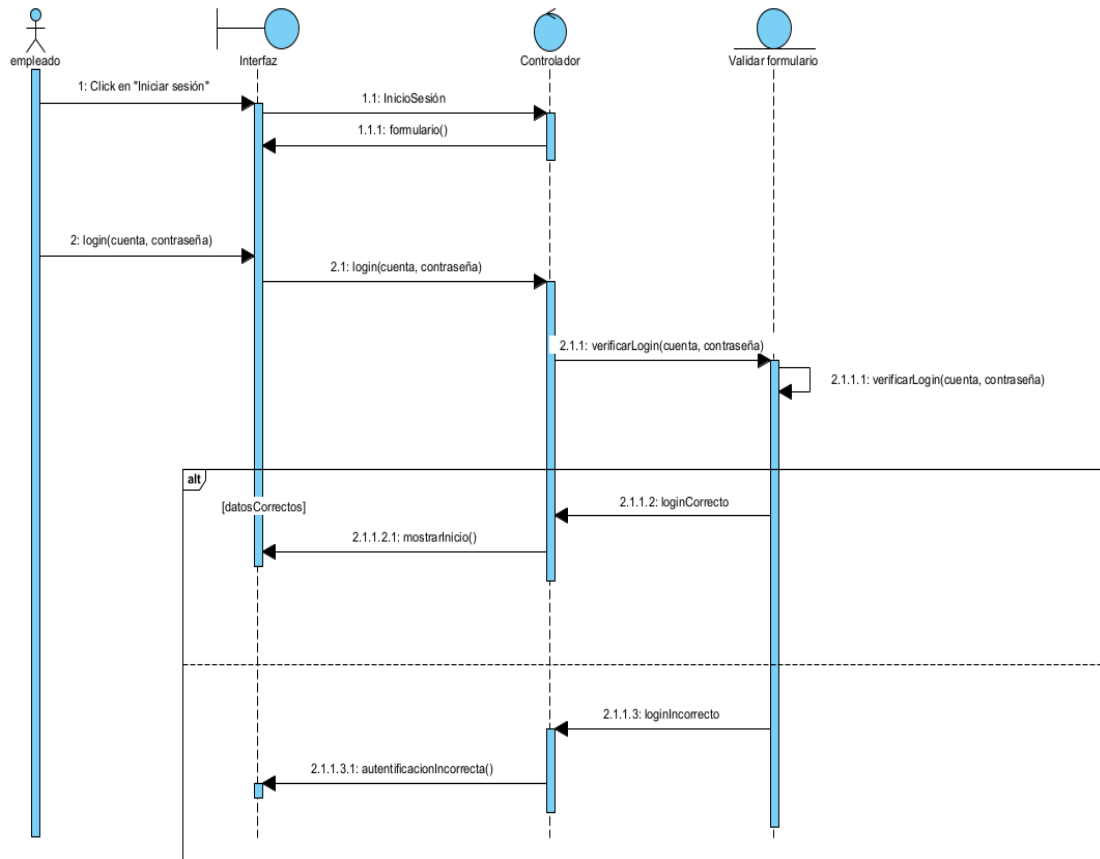
Iniciar sesión:

Ilustración 30: Diagrama de secuencia: Iniciar sesión

Modificar contraseña:

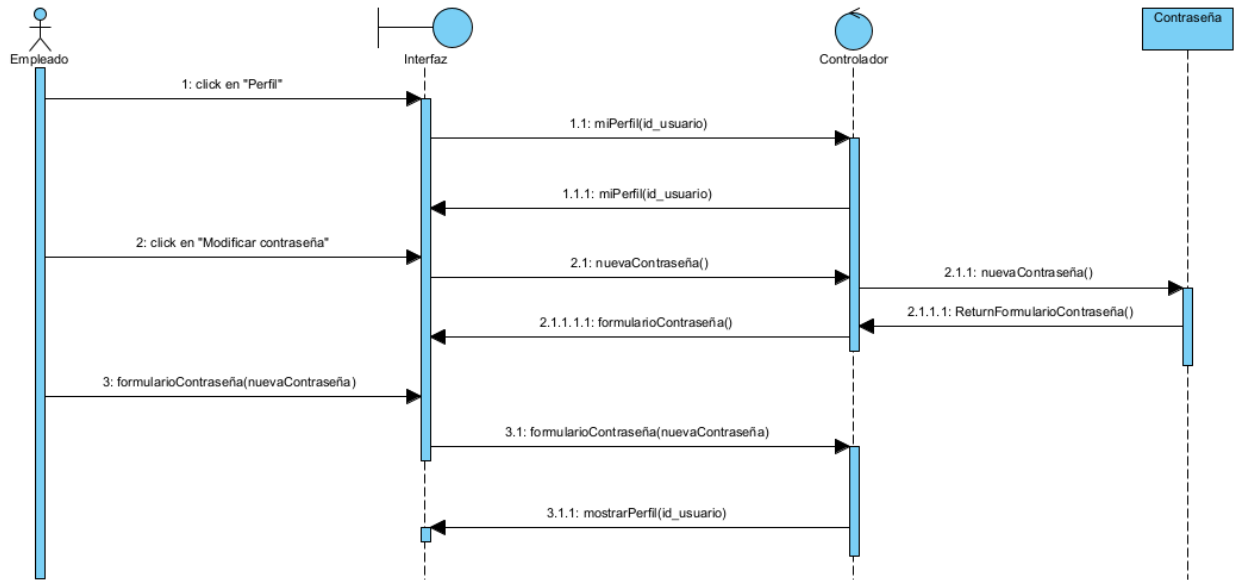
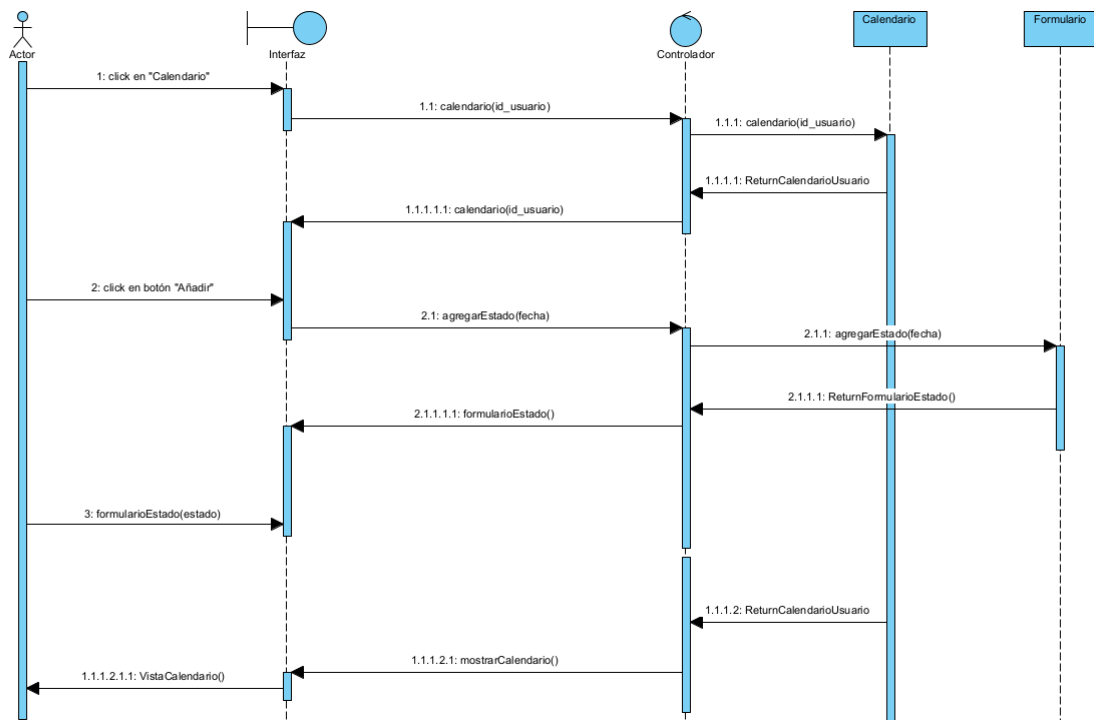


Ilustración 31: Diagrama de secuencia: Modificar contraseña

Agregar estados calendario:*Ilustración 32: Diagrama de secuencia: Agregar estados al calendario*

Agregar información adicional:

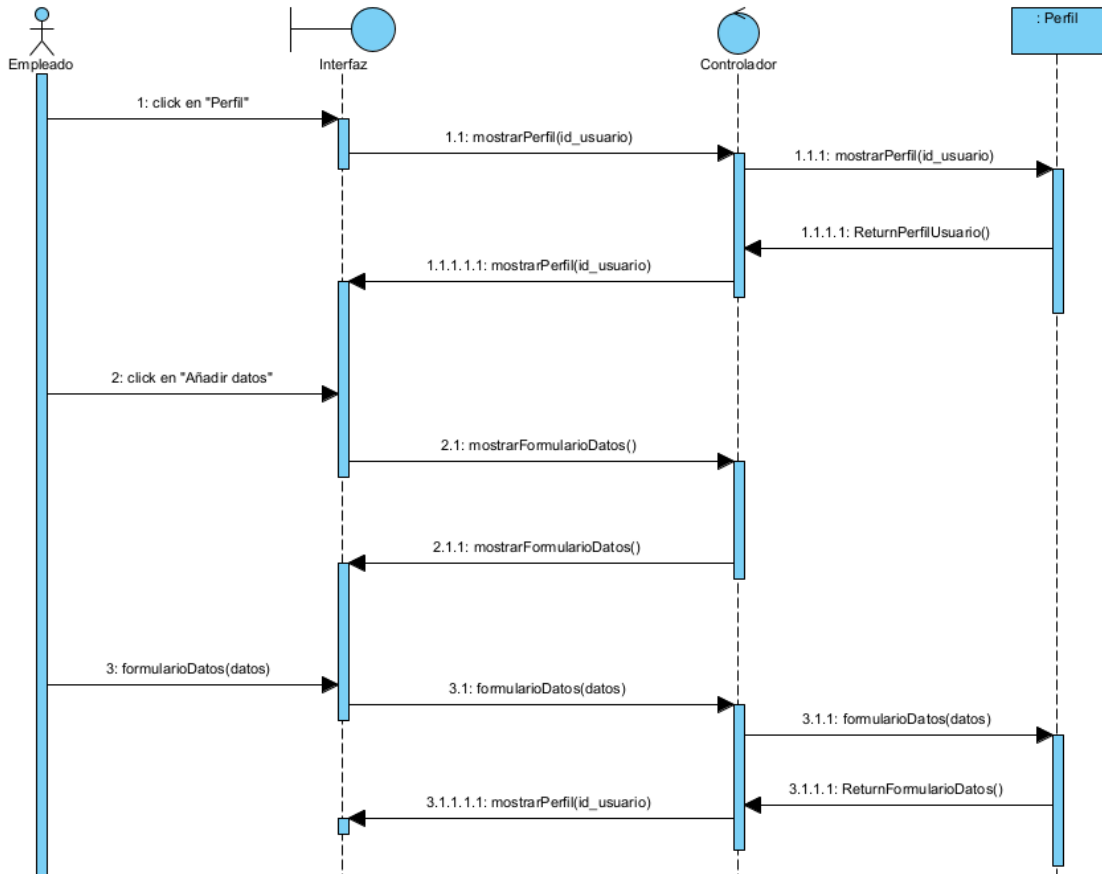


Ilustración 33: Diagrama de secuencia: Añadir información adicional

4.3.6. Diagrama de clases

El diagrama de clases muestra la estructura estática del sistema. El diagrama establece las principales entidades del sistema y sus relaciones entre ellas. [19]

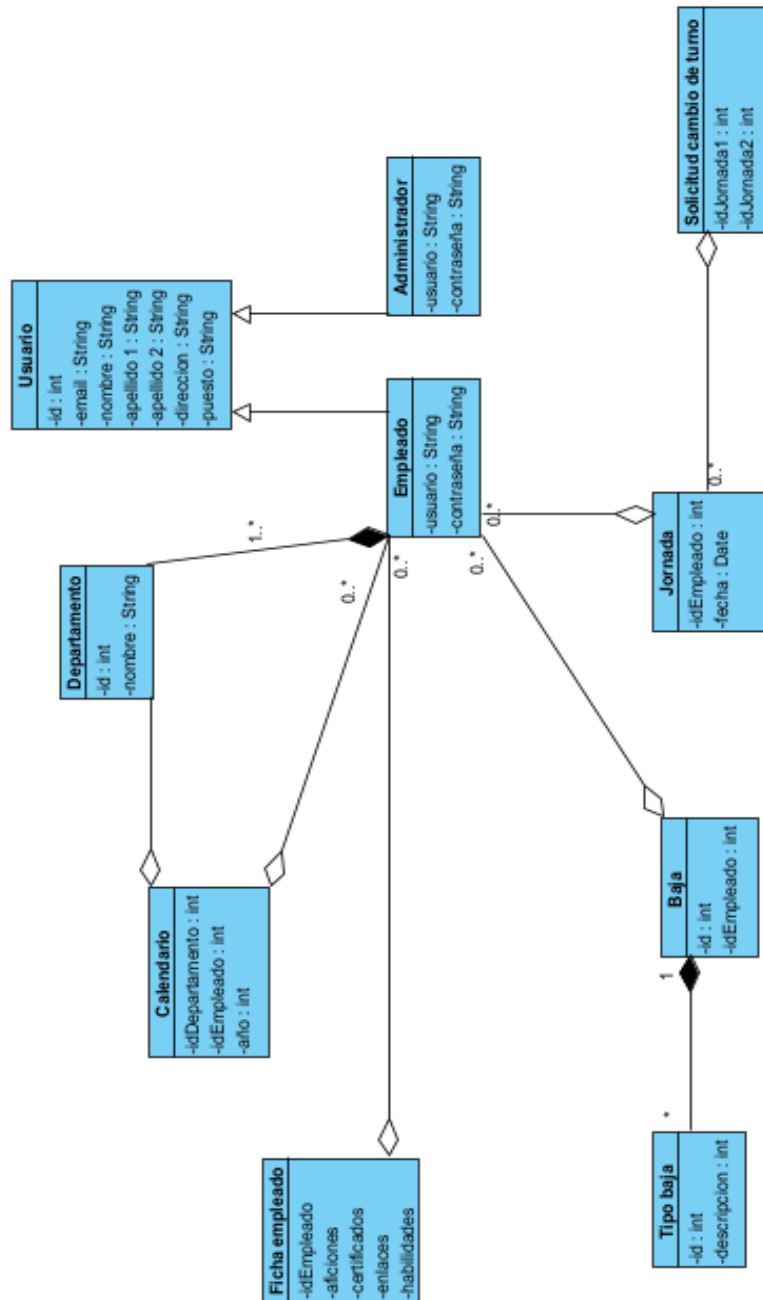


Ilustración 34: Diagrama de clases

CAPÍTULO 5.- PLANIFICACIÓN

En este capítulo realizo una estimación de tiempos y costes. Estimaremos los tiempos que nos empleará finalizar cada una de las tareas y los costes que nos supondrá realizar el proyecto.

5.1. Estimación de tiempos

Tarea	Duración
Búsqueda de información general sobre los CMS	10 horas
Búsqueda y elección del CMS que mejor se ajuste al proyecto	15 horas
Instalación del CMS, aprender cómo funciona y búsqueda de los plugin necesarios para el proyecto	40 horas
Hosting de la aplicación	20 horas
Análisis del proyecto	30 horas
Implementación del proyecto	110 horas
Pruebas	10 horas
Portar proyecto a dispositivos móviles	15 horas
Documentación	50 horas
Total	300 horas

Tabla 15: Estimación de tiempos

He decidido planificar la realización del proyecto usando la idea de jornadas laborales. He decidido que cada jornada laboral es de 4 horas diarias sin contar fines de semana y festivos. Mientras realizaba este proyecto, estaba aún cursando asignaturas de la carrera y por lo tanto las jornadas han de ser menos intensas.

5.2. Diagrama de Gantt

El diagrama de Gantt es una sencilla herramienta visual que sirve no sólo para programar el trabajo de acuerdo con las propiedades, sino también para evaluar rápidamente el estado de todas las tareas, tanto para conocer al instante su situación como para modificar el orden de prioridad según se necesite.

Los diagramas de Gantt muestran gráficamente el trabajo a realizar, una expectativa del trabajo requerido, los tiempos iniciales y finales y, por lo general, el estado del trabajo. [20]

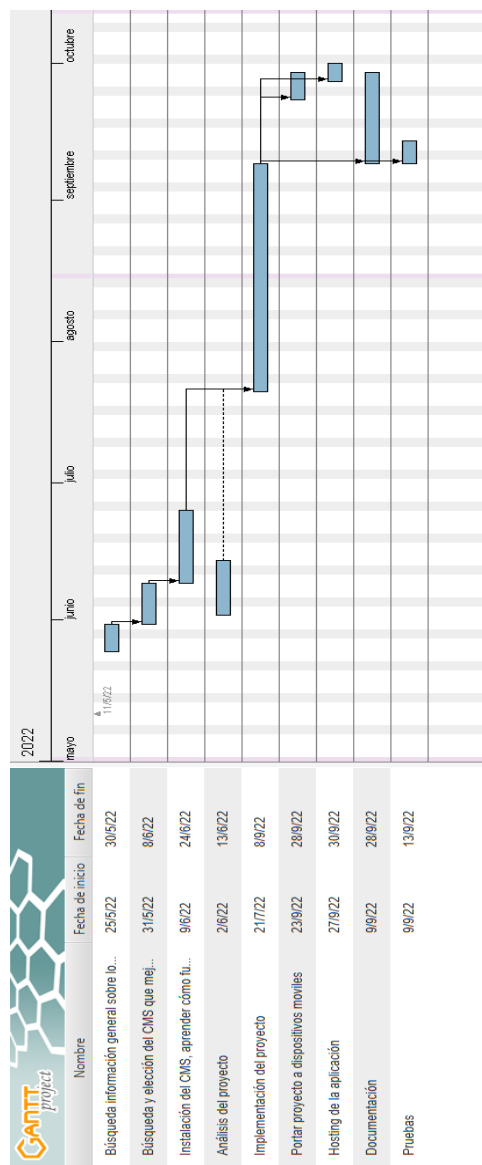


Ilustración 35: Diagrama de Gantt

5.3. Estimación de costes

Describimos y calculamos el costo de los recursos utilizados para la realización del proyecto.

5.3.1. Recursos Software

- Sistema Operativo: Windows 10
- Joomla
- XAMPP
- Visual Paradigm
- Figma

Lenguajes de programación:

- PHP
- HTML
- CSS
- SQL

Todo el Software utilizado es libre o usa licencia académica, por lo tanto, no hemos gastado dinero en estos

5.3.2. Recursos hardware

He utilizado 2 equipos para realizar el proyecto:



Sobremesa montado a piezas

1229.661€

Ilustración 36: Pc sobremesa



HP-Pavilion 15-BC450NS

668.99€

Ilustración 37: PC portátil

5.3.3. Costes ordenadores

Estos equipos los usaremos durante el desarrollo del proyecto, durante un total de 128 días. Cada equipo tiene una vida útil de 5 años, o lo que es lo mismo, 1825 días, por lo tanto, cada equipo tendrá un coste de amortización.

Coste total = SUM (Coste de Amortización por equipo)

Coste Amortización = (Precio de adquisición x días de uso) / vida útil

Coste Amortización equipo sobremesa:

$(1229.36 \text{ €} \times 128 \text{ días}) / 1825 \text{ días} = 89.20 \text{ €}$

Coste Amortización Equipo portátil:

$(668.99 \text{ €} \times 128 \text{ días}) / 1825 \text{ días} = 46.92 \text{ €}$

Coste total equipos amortizados = $89,20\text{€} + 46,92\text{€} = \underline{136.12\text{€}}$

5.3.4. Recursos humanos

Para este proyecto se miró el perfil de desarrollador Full-Stack. Un desarrollador de Full-Stack cobra de media unos 34000 € al año.

El desarrollo del proyecto ha durado un total de 5 meses a media jornada por lo tanto el salario de esos 5 meses juntos es: 7083 €.

5.3.5. Alquiler de oficina

Para este proyecto se estima necesario el alquiler de una oficina donde físicamente llevar a cabo el trabajo. La media de coste de alquiler de una oficina en Andalucía es de 420 € al mes. Por lo tanto: $420 \text{ €} \times 5 \text{ meses} = 2100 \text{ €}$

5.3.6. Consumo de luz y agua

El consumo de luz y agua de la oficina se estima en un coste de 125 € al mes.

5.3.7. Internet

Será necesaria la contratación de una línea de fibra óptica que supone un coste de 17 € al mes.

5.3.8. Coste total

Descripción	Detalles	Coste
Licencias software	Software Libre / Claves académicas	0 €
Hardware	Ordenadores	136.12 €
Desarrollador		7083 €
Alquiler de oficina		2100 €
Consumo de luz y agua		625 €
Conexión a internet		85 €
Diversos consumibles	Material de oficina	100 €
TOTAL		10130.12€

Tabla 16: Coste total del proyecto

CAPÍTULO 6.- PROTOTIPO

En este apartado, hablaremos sobre el prototipo de la aplicación y de la importancia de tener una interfaz accesible.

Es importante citar las ventajas y desventajas que puede tener un prototipo: [21]

Ventajas

- Facilita la comunicación con el cliente
- Resultados visibles al comienzo
- Disminuye el riesgo de error
- Facilita la gestión de cambios
- Aumenta la productividad

Desventajas

- No se consideran aspectos de calidad y mantenimiento
- Se toman decisiones que resultan no ser las más adecuadas

6.1. Accesibilidad Web

Un sitio Web accesible es aquel que permite el acceso efectivo a todos sus contenidos, a todas las personas, independiente de su discapacidad (visual, auditiva, física, intelectual, entre otras). Este concepto, basado en el principio de Accesibilidad Universal, hace a su vez referencia a un diseño que permitirá que estas personas puedan percibir, entender, navegar e interactuar con la Web. [22]

PRINCIPIO 1: PERCEPTIBLE

La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que ellos puedan percibirlos. El sitio Web debe ser percibido por sentidos como la visión y la audición, considerando aspectos como audio, texto y tamaño de fuente, contraste de colores, entre otros.

PRINCIPIO 2: OPERABLE

Los componentes de la interfaz de usuario y la navegación deben ser operables por teclado, permitiendo la interacción entre el usuario y la plataforma. Todas las funciones del sitio están disponibles a través del uso del teclado. Personas con discapacidad visual no usan mouse e interactúan con el sitio mediante navegación por teclado, utilizando lectores de pantalla.

PRINCIPIO 3: COMPRENSIBLE

La información y el manejo de la interfaz de usuario deben ser comprensibles, es decir, entenderse de manera fácil. Los contenidos del sitio Web deben tener un orden lógico de navegación (derecha y hacia abajo).

PRINCIPIO 4: ROBUSTO

El contenido debe ser fiable y consistente para permitir su uso y navegación a través de una amplia variedad de aplicaciones de usuario y ayudas técnicas, y soportar futuras tecnologías. Este principio hace referencia a al código fuente del sitio Web.

6.2. Diseño interfaz

INICIO:

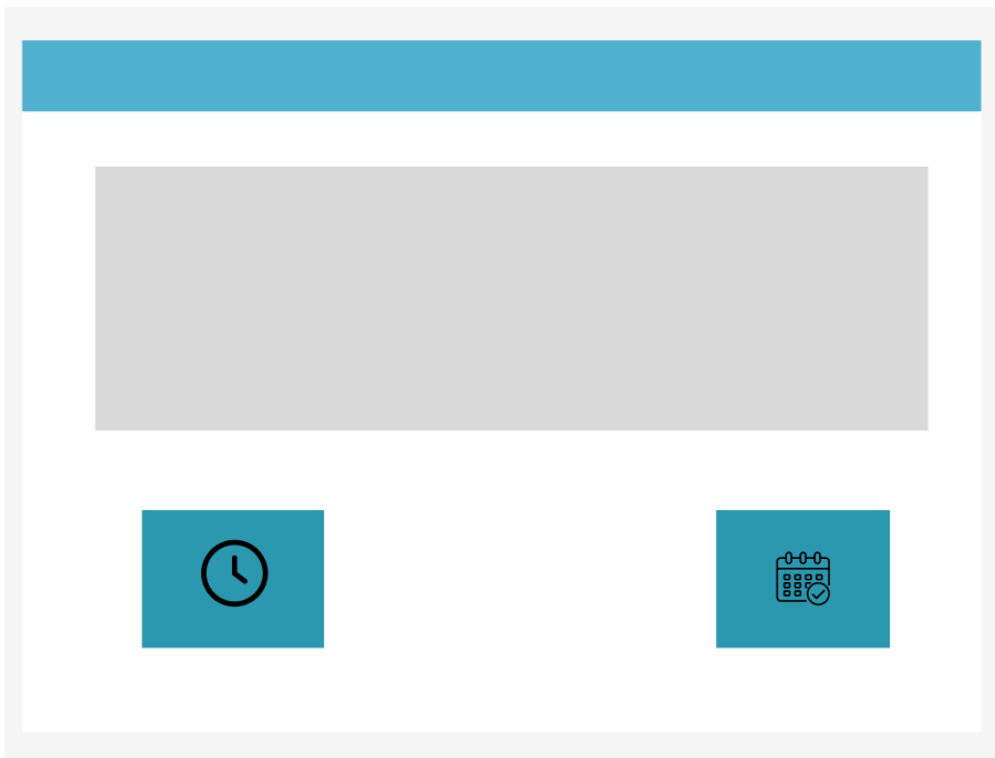


Ilustración 38: Prototipo: Interfaz

ENTRADA:

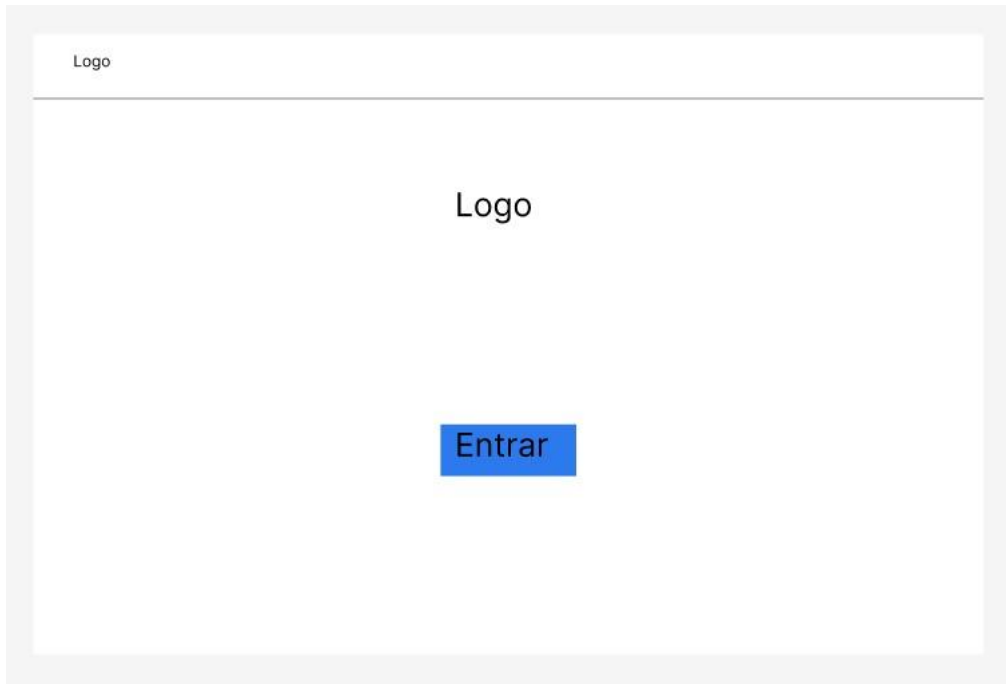


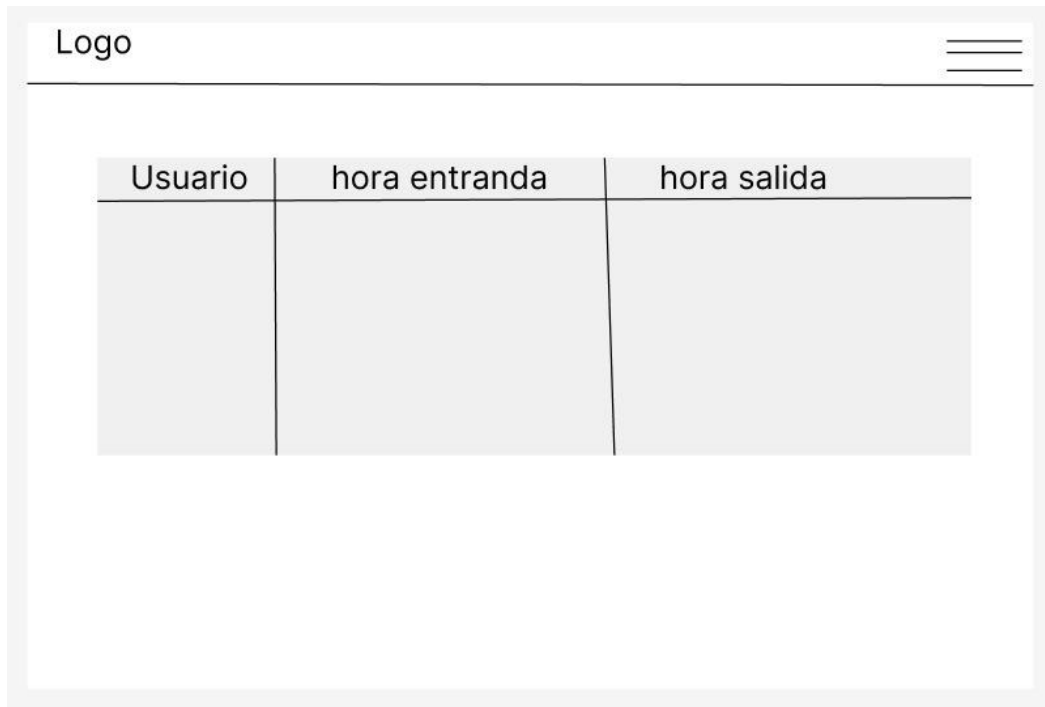
Ilustración 39: Prototipo: Entrada

LOGIN:



Ilustración 40: Prototipo: Login

REGISTRO DE JORNAD LABORAL:




Logo

Usuario	hora entrada	hora salida

The image shows a mobile application prototype for a work shift registration system. It features a header with the word 'Logo' on the left and a hamburger menu icon on the right. Below the header is a table with three columns: 'Usuario', 'hora entrada', and 'hora salida'. The table has a light gray background and a thin black border. The first row contains the column headers, and the second row is empty, representing a data entry point.

Ilustración 41: Prototipo: Registro de jornada laboral

CHECK OUT:



Logo

Logo

Has terminado tu jornada laboral a las:

0000-00-00 00:00:00

[Volver a inicio](#)

The image shows a mobile application prototype for a 'Check Out' screen. It features a header with the word 'Logo' on the left and a hamburger menu icon on the right. Below the header, the word 'Logo' is centered. Underneath, the text 'Has terminado tu jornada laboral a las:' is followed by the placeholder time '0000-00-00 00:00:00'. At the bottom, there is a blue rectangular button with the text 'Volver a inicio' centered below it.

Ilustración 42: Prototipo: Check Out

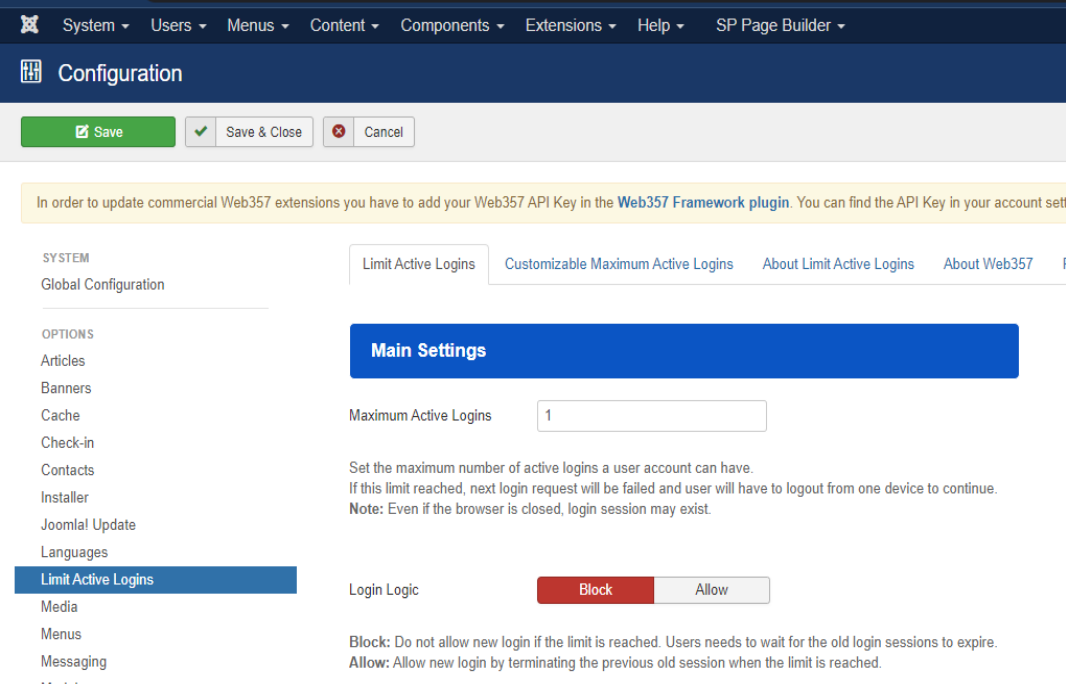
CAPÍTULO 7.- IMPLEMENTACIÓN

En este apartado explicaré que extensiones he usado en mi aplicación y capturas de todas las funcionalidades implementadas en el mismo.

7.1. Uso de extensiones

Como he comentado en la parte que hablaba sobre las extensiones, las extensiones son elementos muy diversos y situacionales. En este apartado hablaré sobre las extensiones que he usado para este proyecto y su utilidad dentro del mismo.

- Limit Active Login:** Esta extensión nos permite limitar el número de sesiones activas que puede tener activas un mismo usuario. Al tratarse esta aplicación, una aplicación para el control de jornada laboral, es esencial limitar a una única sesión activa por usuario ya que si hubiera más podrían existir conflictos a la hora de registrar las horas de entrada o de salida.



In order to update commercial Web357 extensions you have to add your Web357 API Key in the [Web357 Framework plugin](#). You can find the API Key in your account sett

SYSTEM
Global Configuration

OPTIONS
Articles
Banners
Cache
Check-in
Contacts
Installer
Joomla! Update
Languages
Limit Active Logins
Media
Menus
Messaging
Modules

Limit Active Logins Customizable Maximum Active Logins About Limit Active Logins About Web357 F

Main Settings

Maximum Active Logins

Set the maximum number of active logins a user account can have.
If this limit reached, next login request will be failed and user will have to logout from one device to continue.
Note: Even if the browser is closed, login session may exist.

Login Logic Block Allow

Block: Do not allow new login if the limit is reached. Users needs to wait for the old login sessions to expire.
Allow: Allow new login by terminating the previous old session when the limit is reached.

Ilustración 43:Extensiones: Limit Active Login

- **Helix Ultimate:** Aunque se trata de una plantilla, al ser un componente externo, es tratado como una extensión. Nos permite personalizarla a nuestro gusto además de ofrecer la versión de la aplicación para dispositivos móviles y tablets.

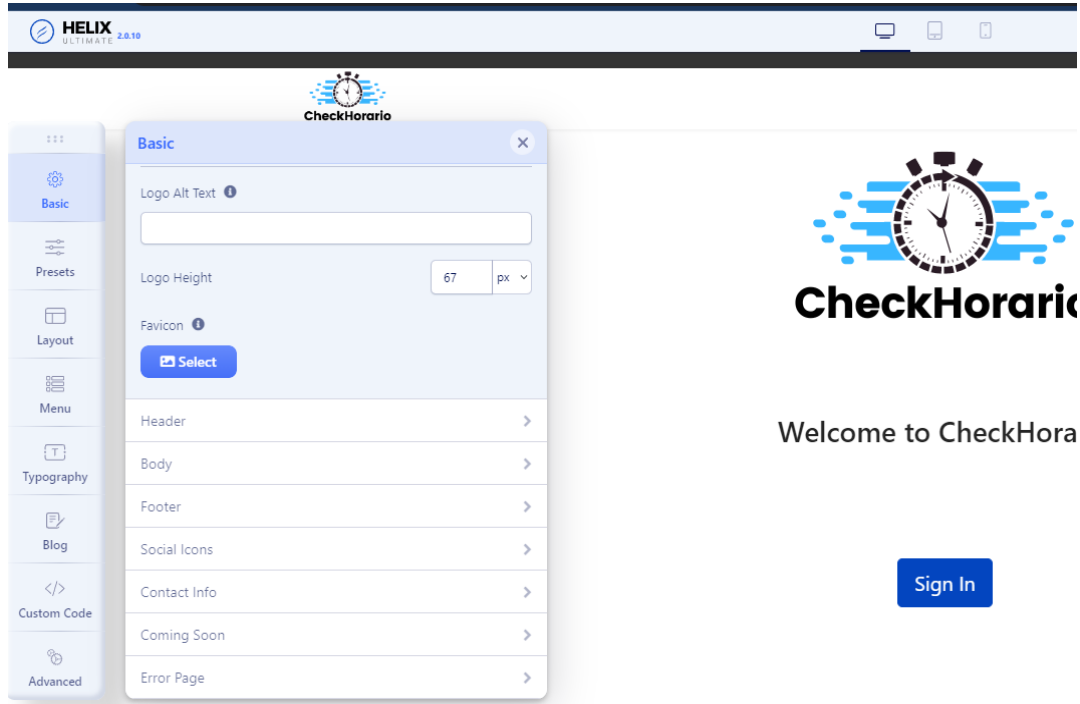


Ilustración 44: Extensiones: Helix Ultimate

- **SP Page Builder:** Nos permite crear las páginas de nuestra aplicación de una manera muy dinámica. Además, nos ofrece la posibilidad de implementar cualquier tipo de componente en cada una de sus filas.

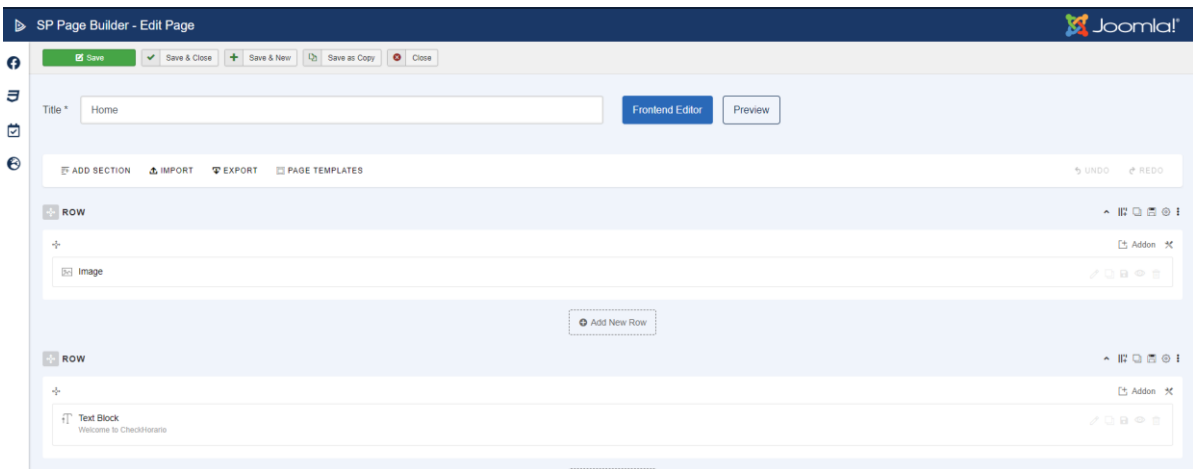


Ilustración 45: Extensiones: SP Page Builder

- **TimeWorked:** Esta extensión nos implementa varias funcionalidades. Nos incorpora más grupos de usuario, proyectos, empresas, tipo de bajas de empleo y estadísticas sobre cada uno de los trabajadores. **Esta extensión no nos permite hacer el sistema de control de hora de entrada y salida.**

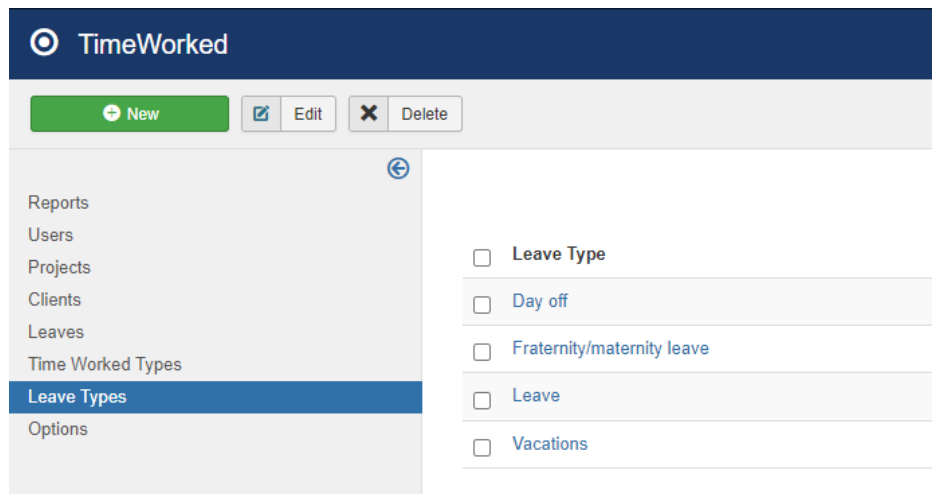


Ilustración 46: Extensiones: TimeWorked (1)

The screenshot shows the 'TimeWorked' application interface. On the left is a navigation menu with options: Reports, Users, Projects, Clients, Leaves (highlighted), Time Worked Types, Leave Types, and Options. The top bar contains buttons for New, Edit, Approve, and Reject. The main content area displays a table with the following data:

Leave Dates	Working Days	Leave Type	Employee
2022			
<input type="checkbox"/> 20 Oct 2022 — 25 Oct 2022	4	Vacations	Alberto Colomo Rueda
<input type="checkbox"/> 20 Oct 2022 — 25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador
<input type="checkbox"/> 20 Oct 2022 — 25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador
<input type="checkbox"/> 20 Oct 2022 — 25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador
<input type="checkbox"/> 20 Oct 2022 — 25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador
<input type="checkbox"/> 7 Oct 2022 — 10 Oct 2022	2	Vacations	Eduardo Sanchez Salvador

Ilustración 47: Extensiones: TimeWorked (2)

The screenshot shows the 'TimeWorked' application interface. On the left is a navigation menu with options: Reports, Users, Projects, Clients (highlighted), Leaves, Time Worked Types, Leave Types, and Options. The top bar contains buttons for New, Edit, Delete, Publish, and Unpublish. The main content area displays a table with the following data:

Name	Preview	Enabled	Contact person	Number of projects
<input type="checkbox"/> Google	Google	<input checked="" type="checkbox"/>		1
<input type="checkbox"/> Ondev	Ondev	<input checked="" type="checkbox"/>		2

Ilustración 48: Extensiones: TimeWorked (3)

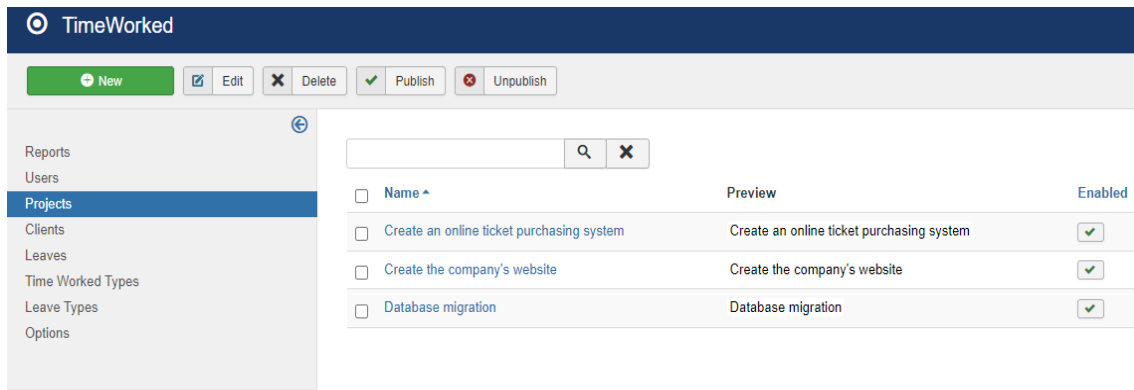


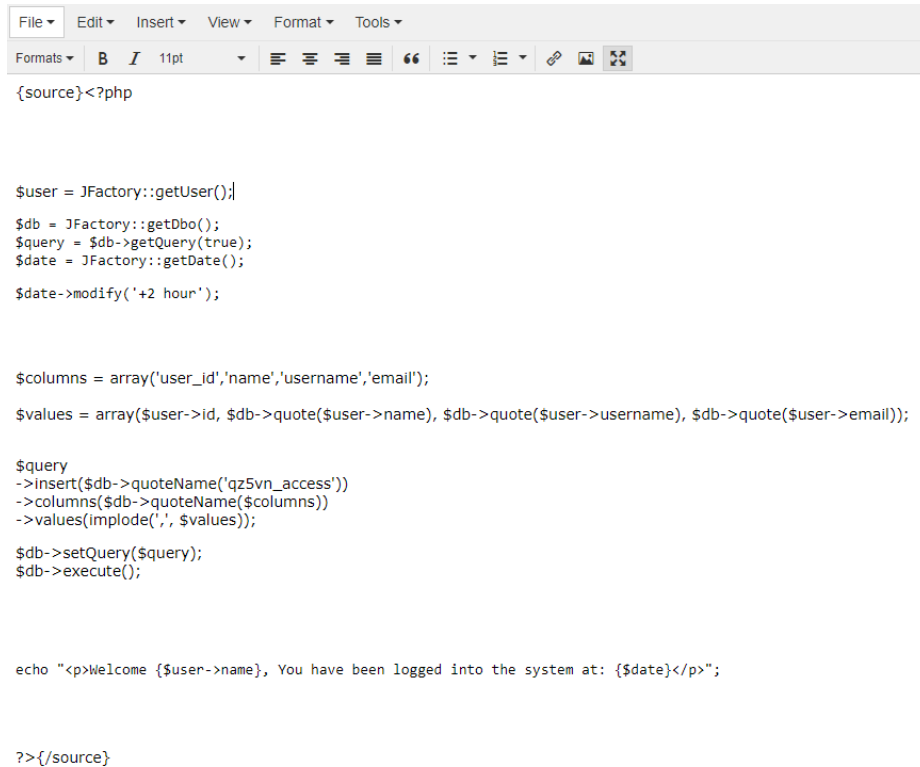
Ilustración 49: Extensiones: TimeWorked (4)

<input type="checkbox"/>	Username	Email	User Groups	Number of projects	Total hours reported	Latest activity
<input type="checkbox"/>	Alberto Colomo Rueda	albertocolru47@gmail.com	Registered, Employee	1	0:30	2022-10-05 15:56:42
<input type="checkbox"/>	Ángel Gallardo Hernández	angelgallardohermz89@gmail.com	Registered, Employee	2	N/A	N/A
<input type="checkbox"/>	Eduardo Sanchez Salvador	eduardosanchez@gmail.com	Registered, Employee	1	7:15	2022-10-07 07:03:24
<input type="checkbox"/>	Francisco Expósito Torres	franexptor@gmail.com	Registered, Project Manager	1	N/A	N/A
<input type="checkbox"/>	Jorge Fernández Chica	jorgefer87@gmail.com	Registered, Employee	1	3:03	2022-10-07 07:17:46
<input type="checkbox"/>	Jose Ángel Muñoz Camacho	axposito644@gmail.com	Manager, Administrator, Super Users	0	N/A	N/A
<input type="checkbox"/>	Pablo Molinero Sánchez	pablomoliner95@gmail.com	Registered, Project Manager	1	3:00	2022-10-09 16:02:39
<input type="checkbox"/>	Sergio Molina Montero	sergiomolina23@gmail.com	Registered, Employee	1	N/A	N/A

Ilustración 50: Extensiones: TimeWorked (5)

Este plugin implementa los roles de **Employee**, **Project Manager** y **Manager**.

- **Sourcerer:** Este plugin me permite usar HTML, CSS, JS y PHP en cualquier cuadro de texto (artículos, bloque de texto de SP Page Builder...). Necesitaremos apoyarnos de '{source}' para poder usar la extensión.



```

{source}<?php

$user = JFactory::getUser();

$db = JFactory::getDbo();
$query = $db->getQuery(true);
$date = JFactory::getDate();

$date->modify('+2 hour');

$columns = array('user_id','name','username','email');
$values = array($user->id, $db->quote($user->name), $db->quote($user->username), $db->quote($user->email));

$query
->insert($db->quoteName('qz5vn_access'))
->columns($db->quoteName($columns))
->values(implode(',', $values));

$db->setQuery($query);
$db->execute();

echo "<p>Welcome {$user->name}, You have been logged into the system at: {$date}</p>";

?>{/source}

```

Ilustración 51: Extensiones: Sourcerer

7.2. Implementación de cada funcionalidad

7.2.1. Registro de la hora de entrada

Después de que el usuario entre en el sistema mediante el formulario de entrada, el usuario será redirigido a la página de bienvenida la cual crea una fila en la tabla 'access' que es la tabla donde guardo la información del usuario junto con su hora de entrada y salida de jornada laboral tal y como vemos en la **ilustración 52**.

	id	user_id	creation_time	name	username	email	date_out
<input type="checkbox"/>	142	290	2022-10-07 07:28:55	Eduardo Sanchez Salvador	ess232	eduardosanchez@gmail.com	2022-10-07 07:31:58
<input type="checkbox"/>	143	294	2022-10-07 07:43:20	Pablo Molinero Sánchez	pms493	pablomoliner095@gmail.com	2022-10-07 07:43:35
<input type="checkbox"/>	144	292	2022-10-07 07:51:58	Sergio Molina Montero	smm928	sergiomolina23@gmail.com	2022-10-07 07:52:41
<input type="checkbox"/>	145	292	2022-10-07 08:02:56	Sergio Molina Montero	smm928	sergiomolina23@gmail.com	2022-10-07 08:03:18
<input type="checkbox"/>	146	290	2022-10-07 09:01:16	Eduardo Sanchez Salvador	ess232	eduardosanchez@gmail.com	2022-10-07 09:02:14
<input type="checkbox"/>	147	290	2022-10-07 09:05:39	Eduardo Sanchez Salvador	ess232	eduardosanchez@gmail.com	2022-10-07 09:06:23
<input type="checkbox"/>	148	295	2022-10-07 09:17:15	Jorge Fernández Chica	jfc749	jorgefer87@gmail.com	2022-10-07 09:32:23
<input type="checkbox"/>	149	294	2022-10-07 09:18:36	Pablo Molinero Sánchez	pms493	pablomoliner095@gmail.com	2022-10-09 18:03:49
<input type="checkbox"/>	150	291	2022-10-07 09:25:41	Francisco Expósito Torres	fet432	franexptor@gmail.com	9999-00-00 00:00:00
<input type="checkbox"/>	151	295	2022-10-07 09:32:07	Jorge Fernández Chica	jfc749	jorgefer87@gmail.com	2022-10-07 09:32:23
<input type="checkbox"/>	152	289	2022-10-07 09:38:43	Jose Ángel Muñoz Camacho	jamc392	axposito644@gmail.com	2022-10-07 09:40:18
<input type="checkbox"/>	153	294	2022-10-09 18:01:03	Pablo Molinero Sánchez	pms493	pablomoliner095@gmail.com	2022-10-09 18:03:49

Ilustración 52: Registro de la hora de entrada: Tabla de la base de datos (1)

La tabla 'access' es una tabla creada por mí mismo. No la crea Joomla automáticamente. El motivo para crear esta tabla es porque Joomla no tiene ninguna tabla que registre la hora de entrada y salida de un empleado, y aunque la hubiera, la idea de la aplicación es que el checkout lo haga el empleado de forma manual tal y como vemos en la **ilustración 53**.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Comentarios	Extra	Acción
1	id	int(10)		UNSIGNED	No	Ninguna		AUTO_INCREMENT	Cambiar Eliminar Más
2	user_id	int(11)			No	Ninguna			Cambiar Eliminar Más
3	creation_time	datetime			Si	current_timestamp()			Cambiar Eliminar Más
4	name	varchar(400)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
5	username	varchar(400)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
6	email	varchar(100)	utf8_general_ci		No	Ninguna			Cambiar Eliminar Más
7	date_out	datetime			Si	9999-00-00 00:00:00			Cambiar Eliminar Más

Ilustración 53: Registro de la hora de entrada: Tabla de la base de datos (2)

La hora de entrada al sistema será igual a la hora que sea en ese momento y a su vez, esta será la hora en la que empieza la jornada laboral del empleado. Esto lo logro dándole por defecto el valor 'current_timestamp ()' al atributo 'creation_time'.

Veamos el código que utilizo para realizar esto en la **ilustración 54**:

```
{source}<?php

$user = JFactory::getUser();
$db = JFactory::getDbo();
$query = $db->getQuery(true);
$date = JFactory::getDate();
$date->modify('+2 hour');

$columns = array('user_id','name','username','email');
$values = array($user->id, $db->quote($user->name), $db->quote($user->username), $db->quote($user->email));

$query
->insert($db->quoteName('qz5vn_access'))
->columns($db->quoteName($columns))
->values(implode(',', $values));

$db->setQuery($query);
$db->execute();

echo "<p>Welcome {$user->name}, You have been logged into the system at: {$date}</p>";

?>{/source}
```

Ilustración 54: Registro de la hora de entrada: Código

JFactory es la clase con la que podemos usar PHP en Joomla. Con **JFactory::getUser()** obtenemos una instancia del usuario que estemos usando en ese momento en nuestra aplicación Joomla. **JFactory::getDate()** nos muestra la fecha y la hora actual, pero, muestra siempre el tiempo con dos horas de retraso, por eso, tengo que hacer un 'modify +2 hour' para sacar por pantalla la hora correcta (esta no es la hora que guardo más adelante en la tabla, la que guardo es el valor predeterminado, 'current_timestamp() el cual saca la hora actual correcta). Con **JFactory::getDbo()** conectamos con la base de datos que usa nuestra aplicación. En mi proyecto la base de datos se llama 'joomla3esp'.

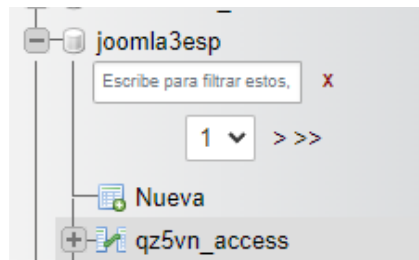


Ilustración 55: Nombre de la base de datos de mi proyecto

Una vez conectados con la base de datos del proyecto, podremos hacer consultas, insertar, borrar y actualizar cualquier tabla de nuestra aplicación tal como vemos en la **ilustración 56**.

```
$columns = array('user_id','name','username','email');
$values = array($user->id, $db->quote($user->name), $db->quote($user->username), $db->quote($user->email));

$query
->insert($db->quoteName('qz5vn_access'))
->columns($db->quoteName($columns))
->values(implode(',', $values));

$db->setQuery($query);
$db->execute();
```

Ilustración 56: Ejemplo código

Para introducir valores nulos (recordad que la fecha de entrada y deben serlo) simplemente no usamos estos atributos dentro de la variable 'columns'. Al hacer esto, la base de datos crea una nueva fila en nuestra tabla con los valores por defecto que nosotros hemos puesto ('current_time ()' para la fecha de entrada y '9999-00-00 00:00:00' para la de salida).

□	✎ Editar	📄 Copiar	🗑️ Borrar	150	291	2022-10-07 09:25:41	Francisco Expósito Torres	fe1432	franexptor@gmail.com	9999-00-00 00:00:00
---	----------	----------	-----------	-----	-----	---------------------	---------------------------	--------	----------------------	---------------------

Ilustración 57: Fila de nuestra tabla

De esta manera ya tenemos la hora en la que un usuario ha empezado la jornada laboral.

7.2.2. Registro de la hora de salida

En la página principal de la aplicación podemos pulsar en el botón de acabar la jornada laboral, el cual nos redirigirá a una página donde se ejecutará el código PHP necesario para guardar en nuestra tabla 'access' la hora en la cual ha terminado nuestra jornada laboral.

```
{source}<?php
$user = JFactory::getUser();
$db = JFactory::getDbo();
$query = $db->getQuery(true);
$date = JFactory::getDate();
$date->modify('+2 hours');

$fields = array(
$db->quoteName('date_out') . ' = ' . $db->quote($date)
);

$conditions = array(
$db->quoteName('user_id') . ' = ' . $user->id,
$db->quoteName('date_out') . ' = ' . $db->quote('9999-00-00 00:00:00')
);

$query->update($db->quoteName('qz5vn_access'))->set($fields)->where($conditions);
$db->setQuery($query);
$result = $db->execute();

?>{/source}
```

Ilustración 58: Código checkout

7.2.3. Visualización de todos los registros realizados por el empleado

Tabla básica de Bootstrap (*ilustración 59*):

```
{source}<table class="table">
----><thead>
----><tr>
---->>><th scope="col">Name</th>
---->>><th scope="col">User</th>
---->>><th scope="col">Check in</th>
---->>><th scope="col">Check out</th>
----></tr>
----></thead>
----><tbody>
----><tr>
---->>><td>
```

Ilustración 59: Código tabla básica de Bootstrap

Guardamos el resultado de la query que devuelve el número de registros que tiene el usuario tal y como vemos en la *ilustración 60*:

```

<?php
$db = JFactory::getDbo();
$query = $db->getQuery(true);
$queryCOUNT = $db->getQuery(true);

$queryCOUNT->select('COUNT(*)');
$queryCOUNT->from($db->quoteName('qz5vn_access'));
$queryCOUNT->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));

$db->setQuery($queryCOUNT);
$count = $db->loadResult();

```

Ilustración 60: COUNT de la tabla respecto al usuario actual

Mostramos por pantalla todas las columnas con un while donde la *i* es igual a la query anterior -1 (empezamos a sacar columnas desde la columna *i=0*) (**ilustración 61**):

Nombre:

```

$query->select($db->quoteName(array('name')));
$query->from($db->quoteName('qz5vn_access'));

$query->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));

$db->setQuery($query);
$column= $db->loadColumn();

$i = 0;
while ($i <= $count-1) {

print_r($column[$i]);
$i++;
echo "<br>";

}

```

Ilustración 61: Código devolver nombre del usuario actual

Tanto en Usuario, como en Hora y fecha de entrada y en Hora y fecha de salida el código utilizado es el mismo. Solo cambiamos el resultado de la query.

Para conseguir un buen efecto en mi tabla dinámica he optado por usar la función loadColumn() ya que así consigo acoplar mejor los datos en la tabla.

Usuario:

```
$db = JFactory::getDbo();
$query = $db->getQuery(true);
$queryCOUNT = $db->getQuery(true);

$queryCOUNT->select('COUNT(*)');
$queryCOUNT->from($db->quoteName('qz5vn_access'));
$queryCOUNT->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));

$db->setQuery($queryCOUNT);
$count = $db->loadResult();

$query->select($db->quoteName(array('username')));
$query->from($db->quoteName('qz5vn_access'));

$query->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));

$db->setQuery($query);
$column= $db->loadColumn();

$i = 0;
while ($i <= $count-1) {

    print_r($column[$i]);
    $i++;
    echo "<br>";

}
```

Ilustración 62: Código devolver nick del usuario actual

Hora y fecha de entrada:

```

$db = JFactory::getDbo();
$query = $db->getQuery(true);
$queryCOUNT = $db->getQuery(true);
$queryCOUNT->select('COUNT(*)');
$queryCOUNT->from($db->quoteName('qz5vn_access'));
$queryCOUNT->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));
$db->setQuery($queryCOUNT);
$count = $db->loadResult();

$query->select($db->quoteName(array('creation_time')));
$query->from($db->quoteName('qz5vn_access'));
$query->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));$query->order('creation_time DESC');
$db->setQuery($query);
$column= $db->loadColumn();

$i = 0;
while ($i <= $count-1) {

print_r($column[$i]);
$i++;
echo "<br>";
}

```

Ilustración 63: Código devolver fecha y hora de entrada del usuario actual

Fecha de entrada descendiente porque queremos mostrar primero los más recientes.

Fecha y hora de salida:

```

$db = JFactory::getDbo();
$query = $db->getQuery(true);
$queryCOUNT = $db->getQuery(true);
$queryCOUNT->select('COUNT(*)');
$queryCOUNT->from($db->quoteName('qz5vn_access'));
$queryCOUNT->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));
$db->setQuery($queryCOUNT);
$count = $db->loadResult();

$query->select($db->quoteName(array('date_out')));
$query->from($db->quoteName('qz5vn_access'));
$query->where($db->quoteName('username') . ' LIKE ' . $db->quote($user->username));
$query->order('date_out DESC');
$db->setQuery($query);
$column= $db->loadColumn();

$i = 0;
while ($i <= $count-1) {

print_r($column[$i]);

$i++;
echo "<br>";
}

```

Ilustración 64: Código devolver fecha y hora de salida del usuario actual

CAPÍTULO 8.- PRUEBAS

Para este apartado he decidido usar pruebas automáticas mediante la herramienta Selenium IDE. Estas pruebas nos ayudarán a comprobar el correcto funcionamiento de nuestra aplicación.

Las pruebas automatizadas tienen como objetivo detectar fallas en el software evitando que una persona tenga que ejecutar las pruebas manualmente. En este caso, el experto en pruebas genera un caso a probar, utilizando una herramienta para construir un robot de software que realice automáticamente las pruebas.

He decidido usar pruebas automáticas por los siguientes motivos:

- Existen procesos repetitivos
- Es necesario correr múltiples pruebas
- Minimizar errores humanos
- Aprender a manejar una herramienta de autorización de pruebas

8.1. Página de inicio

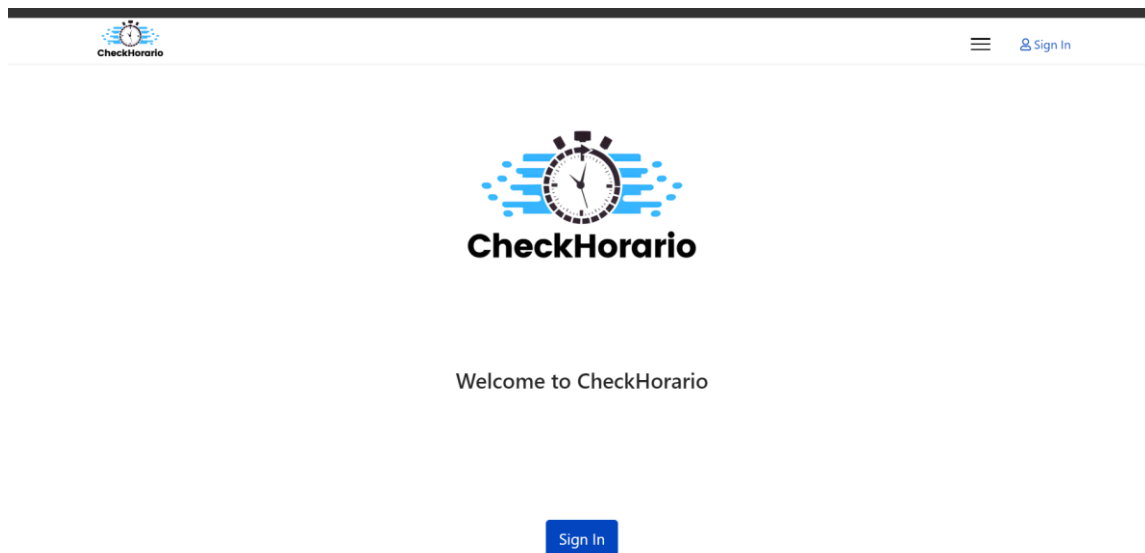


Ilustración 65: Página de Inicio

La página de 'Inicio' y 'Home' son realmente la misma. Con 'SP page builder' puedo hacer ciertos elementos de la página ocultos para según que usuario. En este caso, la imagen de 'CheckHorario', el cuadro de texto de debajo y el botón de entrar, son elementos que solo están visibles para usuarios 'Guest',

que son todos aquellos usuarios visitantes o que aún no han entrado al sistema. El motivo por el que he hecho esto es porque el logo de check horario de la esquina superior izquierda, es un link hacia 'Home'. Supongamos que el usuario aún no ha entrado en el sistema: si el usuario pulsa el logo, le redirigirá a 'Home' la cual es una página a la que no tiene acceso aún.

8.2. Login

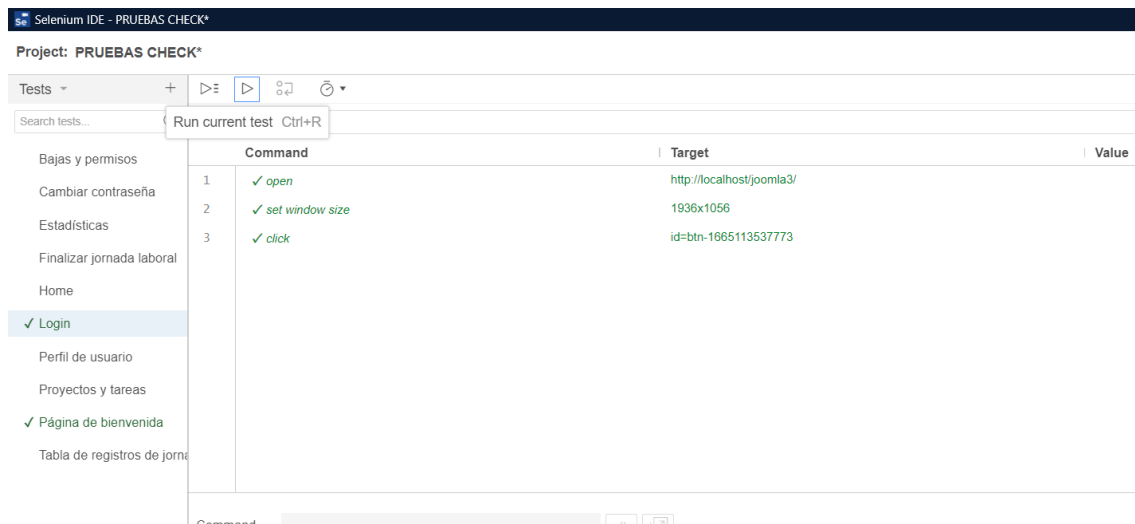


Ilustración 65: Login pruebas automáticas

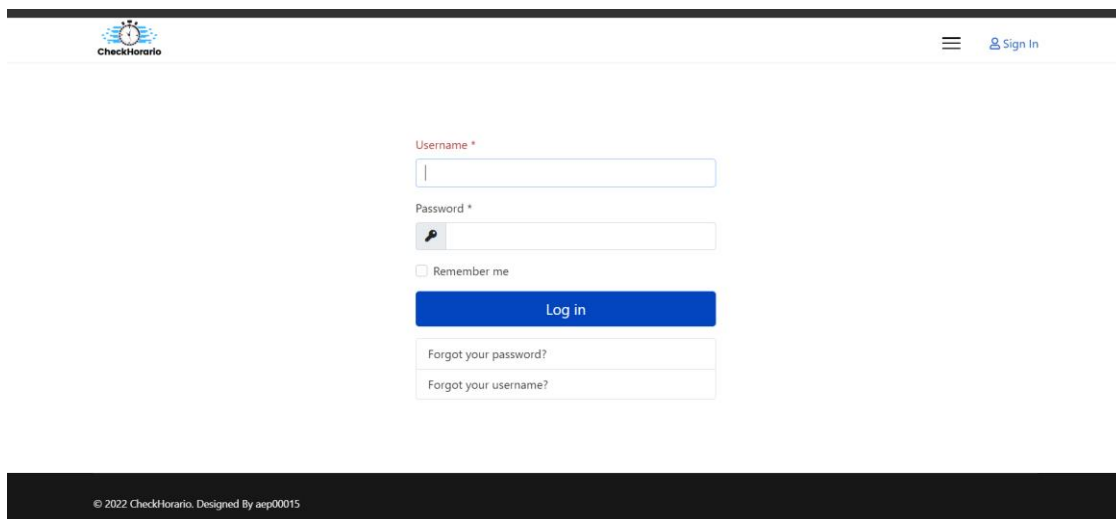


Ilustración 66: Login

Esta página es accesible desde el botón 'Sign In' de la pantalla anterior o desde el link con el mismo nombre que se encuentra en la esquina superior derecha.

```
public class LoginTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void login() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1665113537773")).click();
    }
}
```

Ilustración 68: Código test login

8.3. Página bienvenida

The screenshot shows the Selenium IDE interface with a project named 'PRUEBAS CHECK*'. The test suite 'Página de bienvenida' is selected. The test steps are as follows:

Step	Command	Target	Value
1	open	http://localhost/joomla3/	
2	set window size	1936x1056	
3	click	id=btn-1665113537773	
4	click	id=username	
5	type	id=username	fel432
6	click	id=password	
7	type	id=password	[REDACTED]
8	send keys	id=password	\$(KEY_ENTER)

Ilustración 69: Página de bienvenida pruebas automáticas



Welcome Francisco Expósito Torres, You have been logged into the system at: 2022-10-19 00:31:17

Home

Ilustración 70: Página de bienvenida

Como he explicado en el apartado anterior, solo puede haber una sesión de usuario activa. Esto es muy importante para la funcionalidad de la aplicación. En el caso de que ya haya una sesión activa del usuario se mostrará la siguiente pantalla:



Your devices

Login failed!

You have reached the maximum number of active logins for this account.
The limit of logins are 1 per device. Please, logout from another device to continue.
You have already logged in to the below devices.

Ilustración 71: Página de bienvenida con sesión activa

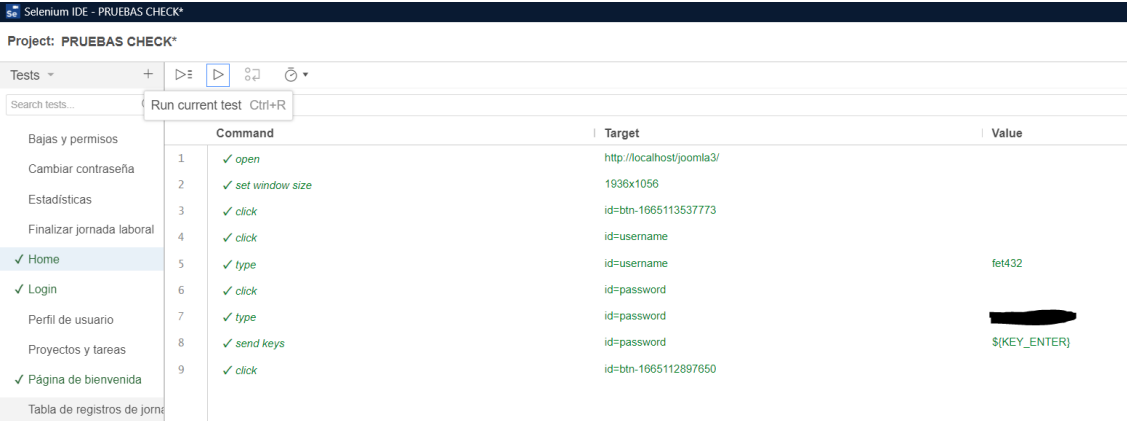
```

25 public class PginadebienvenidaTest {
26     private WebDriver driver;
27     private Map<String, Object> vars;
28     JavascriptExecutor js;
29     @Before
30     public void setUp() {
31         driver = new ChromeDriver();
32         js = (JavascriptExecutor) driver;
33         vars = new HashMap<String, Object>();
34     }
35     @After
36     public void tearDown() {
37         driver.quit();
38     }
39     @Test
40     public void pginadebienvenida() {
41         driver.get("http://localhost/joomla3/");
42         driver.manage().window().setSize(new Dimension(1936, 1056));
43         driver.findElement(By.id("btn-1665113537773")).click();
44         driver.findElement(By.id("username")).click();
45         driver.findElement(By.id("username")).sendKeys("fet432");
46         driver.findElement(By.id("password")).click();
47         driver.findElement(By.id("password")).sendKeys("██████████");
48         driver.findElement(By.id("password")).sendKeys(Keys.ENTER);
49     }
50 }
51

```

Ilustración 72: Código test página de bienvenida

8.4. Home



Command	Target	Value
1 open	http://localhost/joomla3/	
2 set window size	1936x1056	
3 click	id=btn-1665113537773	
4 click	id=username	
5 type	id=username	fet432
6 click	id=password	
7 type	id=password	██████████
8 send keys	id=password	\$(KEY_ENTER)
9 click	id=btn-1665112897650	

Ilustración 73: Home pruebas automáticas

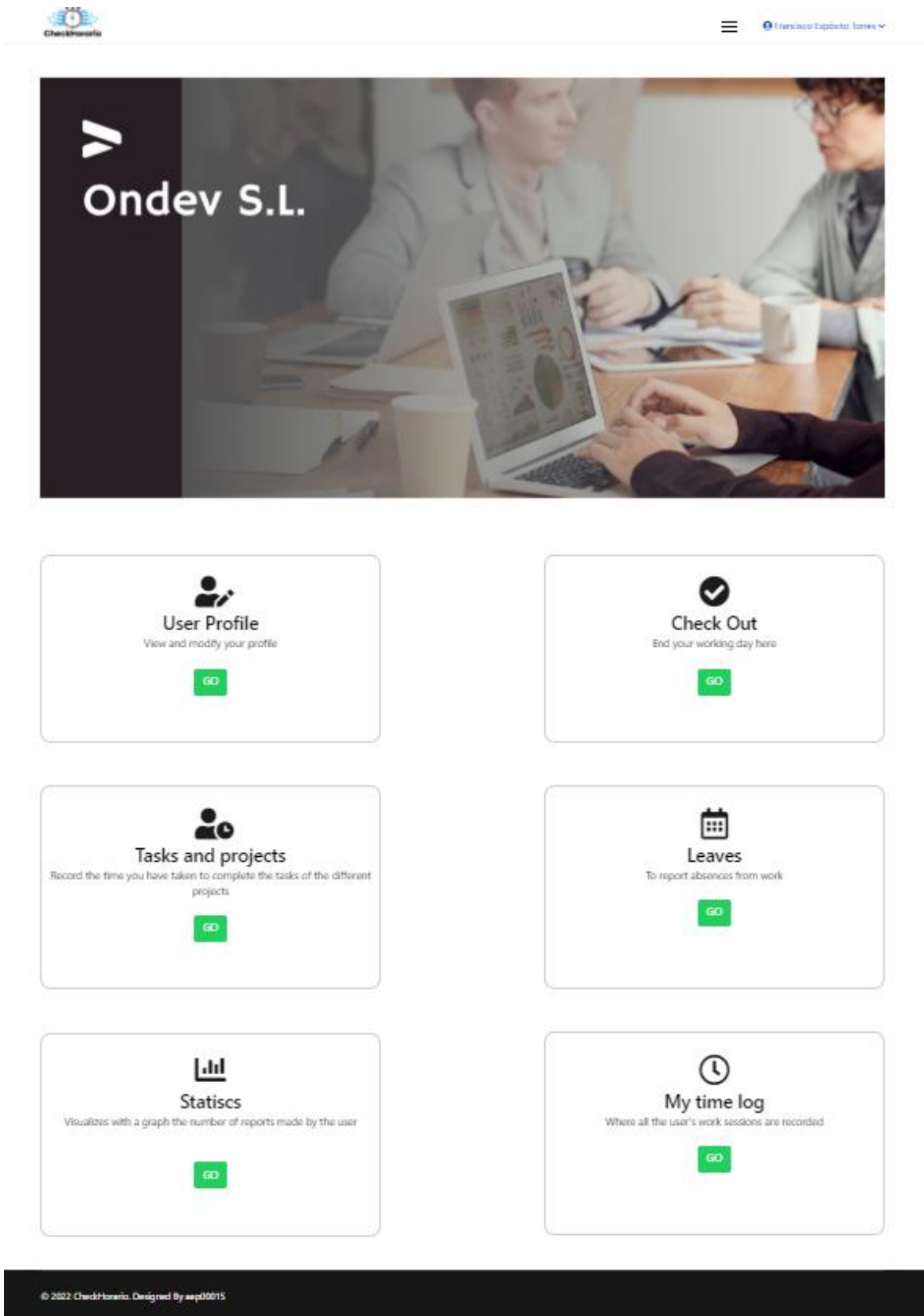


Ilustración 74: Home

Como he explicado en la pantalla de 'Inicio', 'Home' e 'Inicio' son la misma pantalla. En este caso los elementos que podemos ver en la imagen, solo son visibles para usuarios 'Registered' que son todos aquellos que han iniciado sesión.

En esta pantalla se mostrará el nombre de la empresa o compañía que está usando nuestro sistema (en este caso una compañía inventada que nos sirve como ejemplo).

Los elementos de esta pantalla serán accesibles desde las 'box' que vemos en la pantalla o desde el menú desplegable que se encuentra en la parte superior derecha:

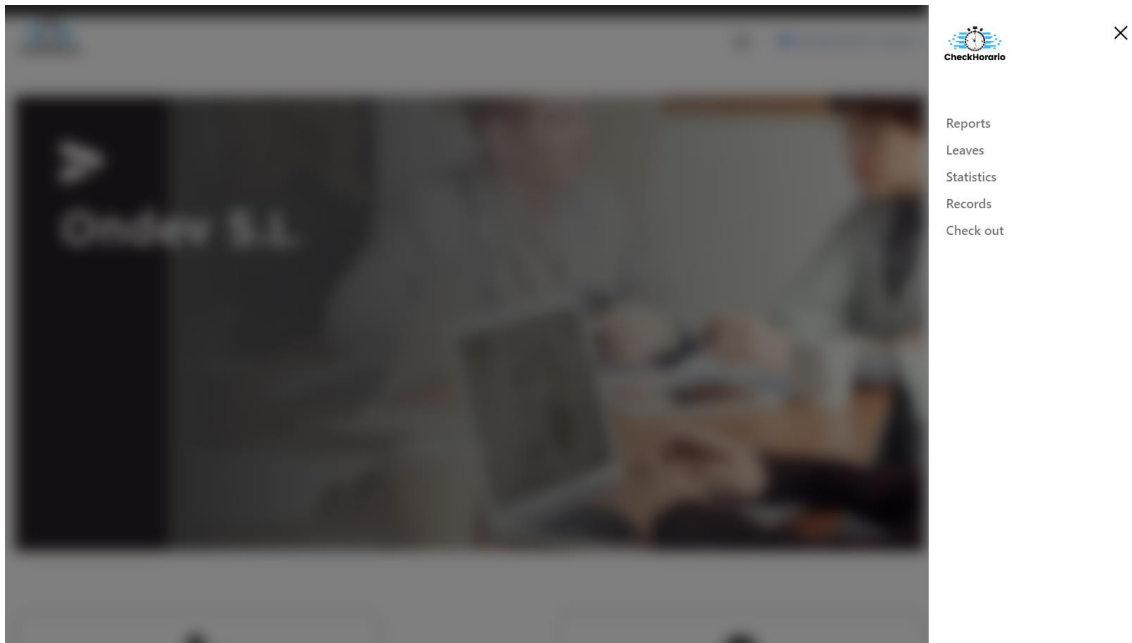


Ilustración 75: Menú desplegable

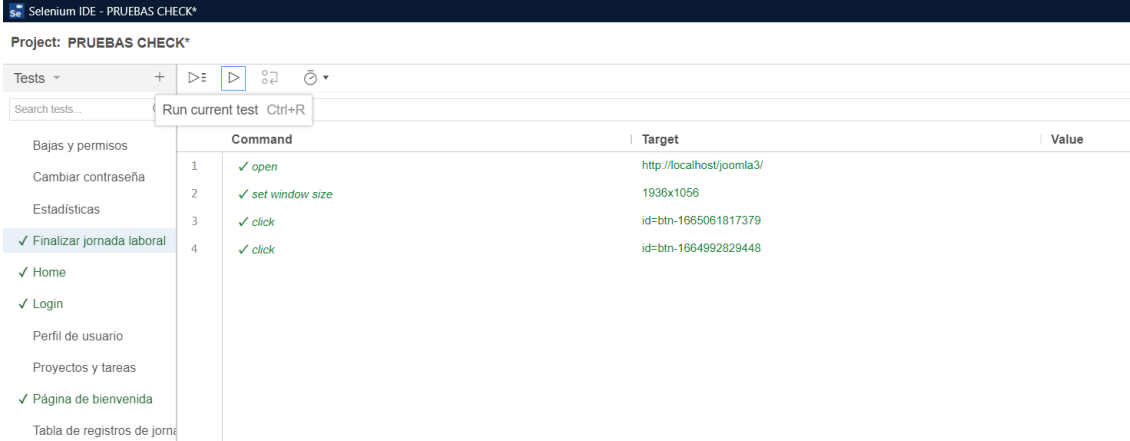
```

public class HomeTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void home() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1665113537773")).click();
        driver.findElement(By.id("username")).click();
        driver.findElement(By.id("username")).sendKeys("fet432");
        driver.findElement(By.id("password")).click();
        driver.findElement(By.id("password")).sendKeys("contraseñaunica");
        driver.findElement(By.id("password")).sendKeys(Keys.ENTER);
        driver.findElement(By.id("btn-1665112897650")).click();
    }
}

```

Ilustración 76: Código test home

8.5. Finalizar jornada laboral



The screenshot shows the Selenium IDE interface for a project named 'PRUEBAS CHECK'. The 'Tests' pane on the left lists several test cases, with 'Finalizar jornada laboral' selected and checked. The main area displays the commands for this test case:

Command	Target	Value
1 ✓ open	http://localhost/joomla3/	
2 ✓ set window size	1936x1056	
3 ✓ click	id=btn-1665061817379	
4 ✓ click	id=btn-1664992829448	

Ilustración 77: Finalizar jornada laboral pruebas automáticas



Workday successfully registered

Home

Ilustración 78: Finalizar jornada laboral

```
public class FinalizarjornadalaboralTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void finalizarjornadalaboral() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1665061817379")).click();
        driver.findElement(By.id("btn-1664992829448")).click();
    }
}
```

Ilustración 79: Código test finalizar jornada laboral

8.6. Tabla registros de jornadas

Antes de finalizar la jornada laboral actual:

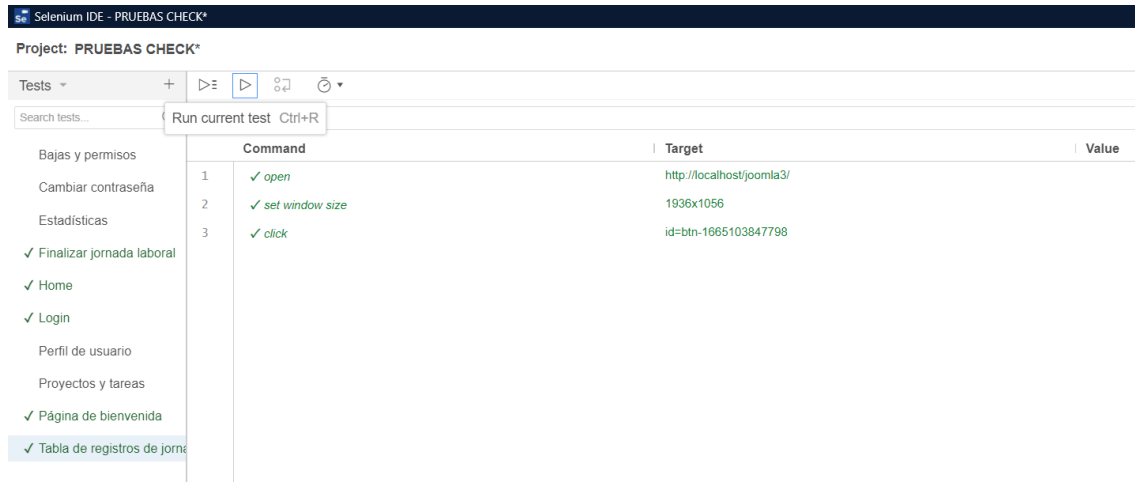


Ilustración 80: Tabla de registros de jornadas

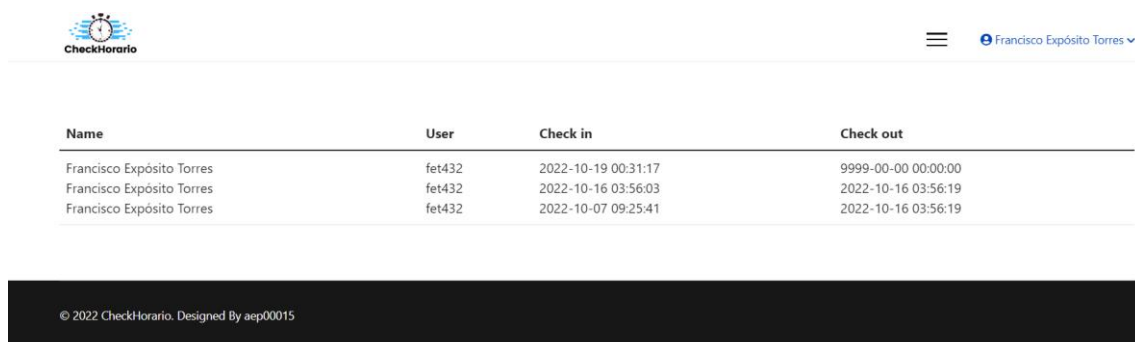


Ilustración 81: Tabla de registros de jornadas antes de finalizar la jornada actual

Después de finalizar la jornada laboral actual:

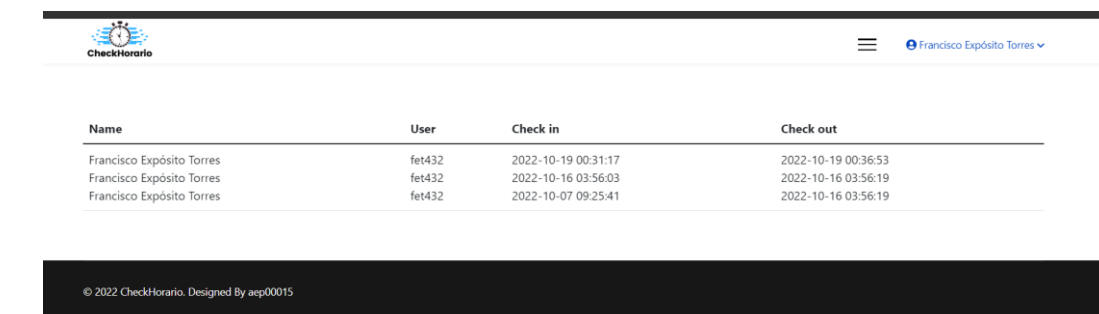


Ilustración 82: Tabla de registros de jornadas después de finalizar la jornada actual

```

public class FinalizarjornadalaboralTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void finalizarjornadalaboral() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1665061817379")).click();
        driver.findElement(By.id("btn-1664992829448")).click();
    }
}

```

Ilustración 83: Código test finalizar jornada laboral

8.7. Proyectos y tareas

Como el usuario se trata de un Project Manager, puede visualizar las tareas que han realizado los empleados de su equipo. Un 'Employee' solo puede ver los suyos propios.

The screenshot shows the Selenium IDE interface for a test suite named 'PRUEBAS CHECK*'. The URL is 'http://localhost/joomla3/'. A table lists the test steps with their commands and targets.

Step	Command	Target	Value
15	✓ click	css=#filter_project_chzn span	
16	✓ click	css=#filter_project_chzn .result-selected	
17	✓ click	css=#filter_staff_chzn span	
18	✓ click	css=.active-result:nth-child(4)	
19	✓ click	id=sp-main-body	
20	✓ click	css=.tw-btn-primary:nth-child(1)	
21	✓ mouse over	css=.tw-btn-primary:nth-child(1)	
22	✓ mouse out	css=.tw-btn-primary:nth-child(1)	
23	✓ click	css= sp-column .logo-image	
24	✓ click	id=btn-1664926185510	

Ilustración 84: Proyectos y tareas pruebas automáticas

The screenshot displays the 'CheckHorario' application interface. At the top left, there is a calendar for October 2022, with the 19th highlighted as 'Today'. The main form area includes fields for 'Project *' (with a dropdown menu), 'Task *' (text input), 'Done' (text area), 'Tickets' (text input), 'Time *' (hh:mm input), and 'Time type *' (dropdown). There are also checkboxes for 'Preserve project' and 'Preserve task name', and 'Submit' and 'Reset' buttons. Below the form is a 'Filters' section with dropdowns for 'Client', 'Employee', 'Project', and 'Date', along with a 'Filter by keyword' field. A 'Table Settings' section shows various columns checked. The main data table has columns: Client, Project, Employee, Date, Task, Done, Tickets, Time, Rejected, Not Billable, and Manage. The table contains five rows of work logs. At the bottom right, it shows 'Total: 13:48 hours'.

Client	Project	Employee	Date	Task	Done	Tickets	Time	Rejected	Not Billable	Manage
Google	Create an online ticket purchasing system	Pablo Molinero Sánchez	9 Oct 2022	Web project			3:00	▲	⊙	🗑️ 📄 📄
Google	Create an online ticket purchasing system	Jorge Fernández Chica	7 Oct 2022	frontend			3:03	▲	⊙	🗑️ 📄 📄
Google	Create an online ticket purchasing system	Eduardo Sanchez Salvador	7 Oct 2022	backend			3:32	▲	⊙	🗑️ 📄 📄
Google	Create an online ticket purchasing system	Eduardo Sanchez Salvador	7 Oct 2022	Backend			3:43	▲	⊙	🗑️ 📄 📄
Oracle	Create the company's website	Alberto Colomo Rueda	5 Oct 2022	Add drop down menu			0:30	▲	⊙	🗑️ 📄 📄

Ilustración 85: Proyectos y tareas

```

public class ProyectostareasTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void proyectosyareas() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1664926185510")).click();
        driver.findElement(By.id("workentry_task")).click();
        driver.findElement(By.id("workentry_task")).sendKeys("frontend");
        driver.findElement(By.cssSelector(".calendar-day-2022-11-18 .day-content")).click();
        js.executeScript("window.scrollTo(0,200)");
        driver.findElement(By.id("workentry_time")).click();
        driver.findElement(By.id("workentry_time")).click();
        driver.findElement(By.id("workentry_time")).sendKeys("8:00");
        driver.findElement(By.id("tw-add-form")).click();
        driver.findElement(By.cssSelector("#filter_client_chzn span")).click();
        driver.findElement(By.cssSelector(".result-selected")).click();
        driver.findElement(By.cssSelector(".span6:nth-child(2)")).click();
        driver.findElement(By.cssSelector("#filter_project_chzn span")).click();
        driver.findElement(By.cssSelector("#filter_project_chzn .result-selected")).click();
        driver.findElement(By.cssSelector("#filter_staff_chzn span")).click();
        driver.findElement(By.cssSelector(".active-result:nth-child(4)")).click();
        driver.findElement(By.id("sp-main-body")).click();
        driver.findElement(By.cssSelector(".tw-btn-primary:nth-child(1)")).click();

        WebElement element = driver.findElement(By.cssSelector(".tw-btn-primary:nth-child(1)"));
        Actions builder = new Actions(driver);
        builder.moveToElement(element).perform();

        WebElement element = driver.findElement(By.tagName("body"));
        Actions builder = new Actions(driver);
        builder.moveToElement(element, 0, 0).perform();

        driver.findElement(By.cssSelector(".sp-column .logo-image")).click();
        driver.findElement(By.id("btn-1664926185510")).click();
    }
}

```

Ilustración 86: Código test proyectos y tareas

8.8. Estadísticas

The screenshot shows the Selenium IDE interface with the project name 'PRUEBAS CHECK*'. The 'Tests' panel on the left lists several test cases, with 'Estadísticas' selected. The main area displays a table of test statistics for the selected test case.

Command	Target	Value
1 ✓ open	http://localhost/joomla3/	
2 ✓ set window size	1936x1056	
3 ✓ click	id=btn-1665056271056	

Ilustración 87: Estadísticas pruebas automáticas



Time Worked Dashboard

EmployeesProjectsMy time

Employee	Today	Yesterday	This Week	Previous Week	This Month	Previous Month
Alberto Colomo Rueda	00:00	00:00	00:00	00:00	00:30	00:00
Eduardo Sanchez Salvador	00:00	00:00	00:00	00:00	07:15	00:00
Jorge Fernández Chica	00:00	00:00	00:00	00:00	03:03	00:00
Pablo Molinero Sánchez	00:00	00:00	00:00	00:00	03:00	00:00

Ilustración 88: Estadísticas

```

public class EstadsticTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void estadstic() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1665056271056")).click();
    }
}

```

Ilustración 89: Código test cambiar estadísticas

8.9. Bajas y permisos

Selenium IDE - PRUEBAS CHECK*

Project: PRUEBAS CHECK*

Tests +

Search tests...

Command	Target	Value
1 ✓ open	http://localhost/joomla3/	
2 ✓ set window size	1936x1056	
3 ✓ click	id=btn-1664930671641	
4 ✓ click	id=leaves_daterange_img	
5 ✓ click	css= daterangepicker.nth-child(4) .calendar.nth-child(2) tr.nth-child(3) > .available.nth-child(3)	
6 ✓ click	css= daterangepicker.nth-child(4) .calendar.nth-child(2) tr.nth-child(3) > .available.nth-child(5)	
7 ✓ click	css= daterangepicker.nth-child(4) tr.nth-child(3) > .available.nth-child(3)	
8 ✓ click	css= daterangepicker.nth-child(4) .applyBtn	
9 ✓ click	css=#leaves_leave_type_chzn span	
10 ✓ click	css=.active-result.nth-child(5)	
11 ✓ click	id=leaves_work_days	

Ilustración 90: Bajas y permisos pruebas automáticas

Leaves

Leave Form

Leave Dates * Number of Working Days * Leave Type *

Comments

Filters

Employee

Hide previous years leaves

19 Oct 20; 19 Oct 20;

Date Interval [Precise Dates](#)

Table Settings

Start Date End Date Working Days Leave Type Employee Comments Status Approve Manage

Start Date	End Date	Working Days	Leave Type	Employee	Comments	Status	Approve	Manage
2022								
7 Oct 2022	10 Oct 2022	2	Vacations	Eduardo Sanchez Salvador		P	<input type="checkbox"/> <input type="checkbox"/>	
20 Oct 2022	25 Oct 2022	4	Vacations	Alberto Colomo Rueda		A	<input checked="" type="checkbox"/> <input type="checkbox"/>	
20 Oct 2022	25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador		R	<input type="checkbox"/> <input type="checkbox"/>	
20 Oct 2022	25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador		R	<input type="checkbox"/> <input type="checkbox"/>	
20 Oct 2022	25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador		R	<input type="checkbox"/> <input type="checkbox"/>	
20 Oct 2022	25 Oct 2022	4	Vacations	Eduardo Sanchez Salvador		A	<input checked="" type="checkbox"/> <input type="checkbox"/>	

Display #

Ilustración 91: Bajas y permisos

```

public class BajasypermisosTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void bajasypermisos() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1664930671641")).click();
        driver.findElement(By.id("leaves_daterange_img")).click();
        driver.findElement(By.cssSelector(".daterangepicker:nth-child(4) .calendar:nth-child(2) tr:nth-child(3) > .available:nth-child(3)")).click();
        driver.findElement(By.cssSelector(".daterangepicker:nth-child(4) .calendar:nth-child(2) tr:nth-child(3) > .available:nth-child(5)")).click();
        driver.findElement(By.cssSelector(".daterangepicker:nth-child(4) tr:nth-child(3) > .available:nth-child(3)")).click();
        driver.findElement(By.cssSelector(".daterangepicker:nth-child(4) .applyBtn")).click();
        driver.findElement(By.cssSelector("#leaves_leave_type_chzn span")).click();
        driver.findElement(By.cssSelector(".active-result:nth-child(5)")).click();
        driver.findElement(By.id("leaves_work_days")).click();
        driver.findElement(By.id("leaves_work_days")).sendKeys("3");
        driver.findElement(By.cssSelector(".tw-form")).click();
        driver.findElement(By.cssSelector("#filter_user_chzn span")).click();
        driver.findElement(By.cssSelector(".result-selected:nth-child(4)")).click();
        driver.findElement(By.cssSelector(".tw-btn-primary:nth-child(1)")).click();
        driver.findElement(By.cssSelector(".sp-column .logo-image")).click();
        driver.findElement(By.id("btn-1664930671641")).click();
        driver.findElement(By.cssSelector("#filter_user_chzn span")).click();
        driver.findElement(By.cssSelector(".result-selected")).click();
    }
}

```

Ilustración 92: Código test bajas y permisos

8.10. Perfil de usuario

Selenium IDE - PRUEBAS CHECK*

Project: PRUEBAS CHECK*

Tests + [Run current test Ctrl+R]

Test	Command	Target	Value
✓ Bajas y permisos			
Cambiar contraseña	1 ✓ open	http://localhost/joomla3/	
	2 ✓ set window size	1936x1056	
✓ Estadísticas	3 ✓ click	id=btn-1665056271059	
✓ Finalizar jornada laboral			
✓ Home			
✓ Login			
✓ Perfil de usuario			
✓ Proyectos y tareas			
✓ Página de bienvenida			
✓ Tabla de registros de jornada			

Ilustración 93: Perfil de usuario pruebas automáticas



The screenshot shows the 'Profile' section of the CheckHorario application. At the top left is the 'CheckHorario' logo, and at the top right is the user's name 'Francisco Expósito Torres' with a dropdown arrow. Below the header, there is a 'Profile' section with an 'Edit Profile' link. The profile information is displayed in a table-like structure:

Name: Francisco Expósito Torres
Username: fet432
Registered Date: Wednesday, 05 October 2022
Last Visited Date: Wednesday, 19 October 2022

Below the profile information is the 'Basic Settings' section, also in a table-like structure:

Editor: No Information Entered
Time Zone: No Information Entered
Frontend Language: No Information Entered

Ilustración 94: Perfil de usuario

Edit Your Profile

Name *

Username

Password



Confirm Password



Email Address *

Confirm Email Address *

Basic Settings

Editor



Time Zone



Frontend Language

*Ilustración 95: Editar perfil de usuario*

```

public class PerfildeusuarioTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void perfildeusuario() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1665056271059")).click();
    }
}

```

Ilustración 96: Código test perfil de usuario

8.11. Modificar contraseña

The screenshot shows the Selenium IDE interface for a project named 'PRUEBAS CHECK'. The test script is titled 'Cambiar contraseña' and is located at 'http://localhost/joomla3/'. The script consists of 9 steps:

Step	Command	Target	Value
1	open	http://localhost/joomla3/	
2	set window size	1936x1056	
3	click	id=btn-1665056271059	
4	click	linkText=Edit Profile	
5	click	id=form_password1	
6	type	id=form_password1	contraseñaunica
7	click	id=form_password2	
8	type	id=form_password2	contraseñaunica
9	send keys	id=form_password2	\$(KEY_ENTER)

Ilustración 97: Modificar contraseña pruebas automáticas

Edit Your Profile

Name *

Francisco Expósito Torres

Username

fet432

Password

.....

Confirm Password

.....

Email Address *

franexptor@gmail.com

Confirm Email Address *

franexptor@gmail.com

Ilustración 98: Cambiar contraseña

```

public class CambiarcontraseaTest {
    private WebDriver driver;
    private Map<String, Object> vars;
    JavascriptExecutor js;
    @Before
    public void setUp() {
        driver = new ChromeDriver();
        js = (JavascriptExecutor) driver;
        vars = new HashMap<String, Object>();
    }
    @After
    public void tearDown() {
        driver.quit();
    }
    @Test
    public void cambiarcontrasea() {
        driver.get("http://localhost/joomla3/");
        driver.manage().window().setSize(new Dimension(1936, 1056));
        driver.findElement(By.id("btn-1665056271059")).click();
        driver.findElement(By.linkText("Edit Profile")).click();
        driver.findElement(By.id("jform_password1")).click();
        driver.findElement(By.id("jform_password1")).sendKeys(".....");
        driver.findElement(By.id("jform_password2")).click();
        driver.findElement(By.id("jform_password2")).sendKeys(".....");
        driver.findElement(By.id("jform_password2")).sendKeys(Keys.ENTER);
    }
}

```

Ilustración 99: Código test cambiar contraseña

CAPÍTULO 9.- CONCLUSIÓN

9.1. Conclusiones

He logrado finalizar con éxito la aplicación, logrando implementar cada uno de los requisitos y obteniendo una aplicación funcional. Una de mis mayores metas era crear una función intuitiva con una interfaz básica y que se adaptara en varios dispositivos. Otra meta importante era lograr hacer una aplicación segura. Para lograrlo he trabajado con una extensión que me ha permitido limitar el número de sesiones activas de un usuario, así como el tiempo y las veces que un usuario intenta entrar a la aplicación.

Una de las mayores dificultades que he encontrado es la zona horaria. Tras varios intentos he logrado arreglar este problema y adaptar la zona horaria a la localización actual. Aunque en teoría se trata de algo sencillo, en la práctica no era posible cambiar con éxito la zona horaria. Para ello tuve que convertir en varias partes del código las String a Date, sumarle las horas necesarias y volverlas a convertir.

Para finalizar, decir que este proyecto ha sido muy útil tanto para aprender PHP como para aprender acerca de los CMS y saber usar Joomla, además de, conocer muchas extensiones y elementos que nos ofrece este CMS.

9.2. Desarrollos futuros

Con respecto a este apartado, la aplicación puede ser mejorada en varios aspectos. Uno de ellos sería usar un hosting junto a un dominio web para poder usar esta aplicación en cualquier momento. Para ello una opción sería hostear la aplicación en cualquier servicio de hosting (aunque la mayoría son de pago) y comprar un dominio. La adaptación de las URLs sería algo relativamente sencillo gracias al back gráfico que nos ofrece Joomla. Además, Joomla, nos permite implementar de una manera bastante intuitiva y sencilla HTTPS.

Otra funcionalidad que podríamos implementar sería la opción de parar el tiempo de trabajo varias veces (Ej. Hora del almuerzo).

En mi proyecto me decanté por iniciar el turno de trabajo una vez entres en el sistema, pero otra opción sería iniciar el turno de trabajo manualmente, al igual que pasa con finalizar turno de trabajo.

Me siento agradecido por haber aprendido este tipo de tecnologías ya que realmente hay mucho mercado en el mundo de los CMS gracias a plataformas FreeLancer.

ANEXOS

Anexo 1. Instalación de Joomla

Configuración e instalación de Xampp

En este apartado mostraré como he realizado la instalación y la configuración de nuestro servidor web. Para realizar esta tarea he utilizado XAMPP.

Lo primero que tenemos que hacer es dirigirnos a la página oficial de XAMPP y descargar el instalador a través del siguiente enlace:

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

Elegir la versión que corresponda a nuestro sistema operativo. En mi caso he elegido la versión de Windows.

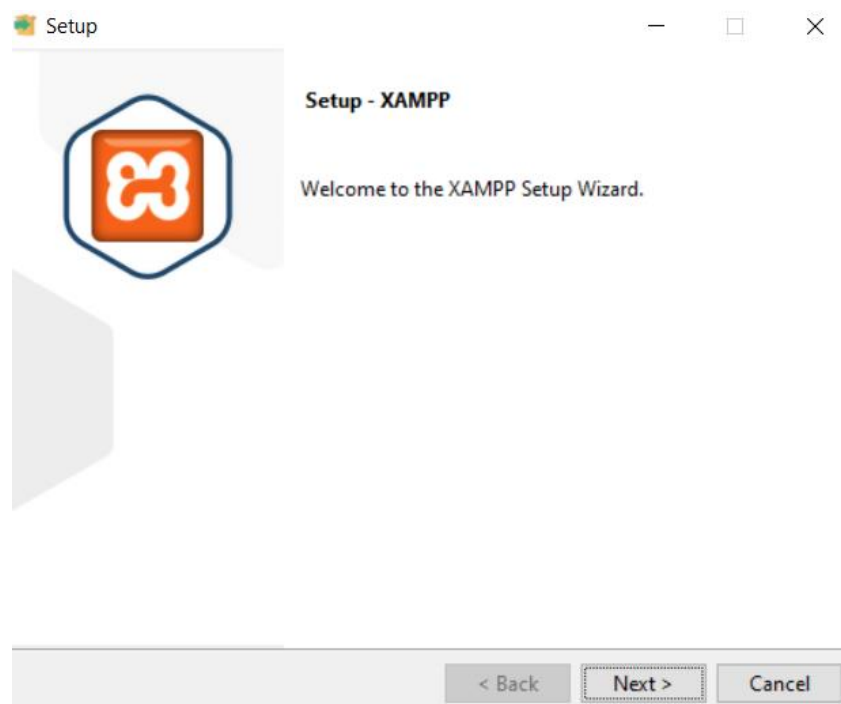


Ilustración 100: Instalador XAMPP (1)

Elegiremos los componentes que queremos instalar junto a XAMPP. Podemos dejar las casillas marcadas por defecto o escoger solo las que vayamos a necesitar.

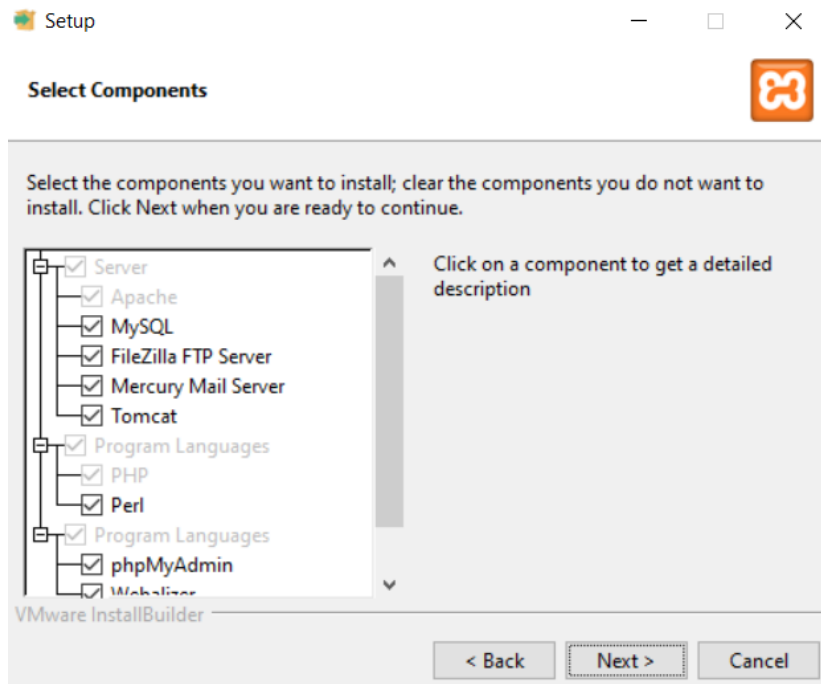


Ilustración 101: Instalador de XAMPP (2)

Elegimos la carpeta donde queremos instalar XAMPP:

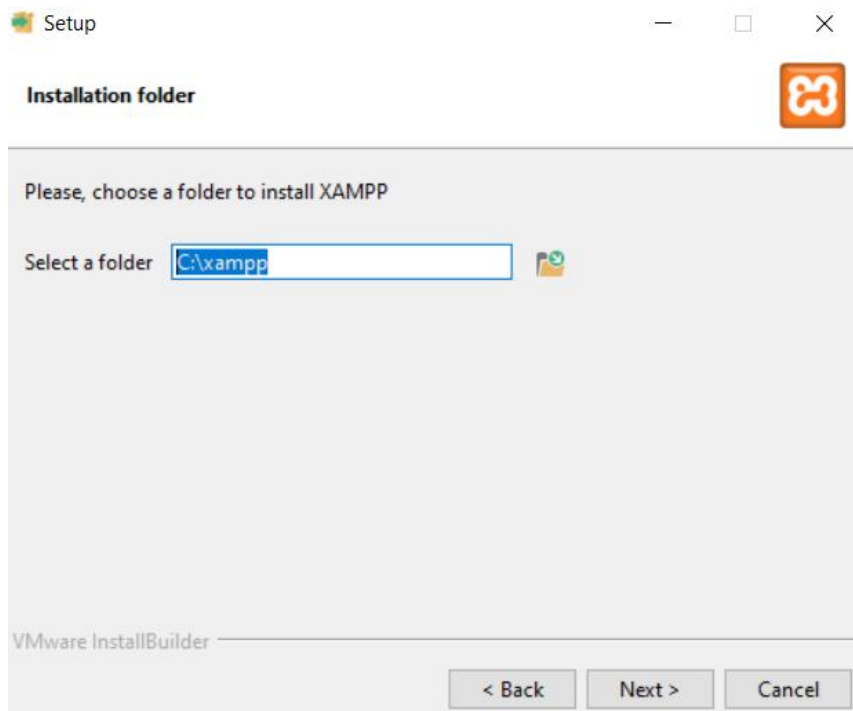


Ilustración 102: Instalador de XAMPP (3)

Una vez instalado, podremos acceder al panel de control de XAMPP. Para instalar y usar nuestro proyecto deberemos pulsar en los botones de 'Start' señalados:

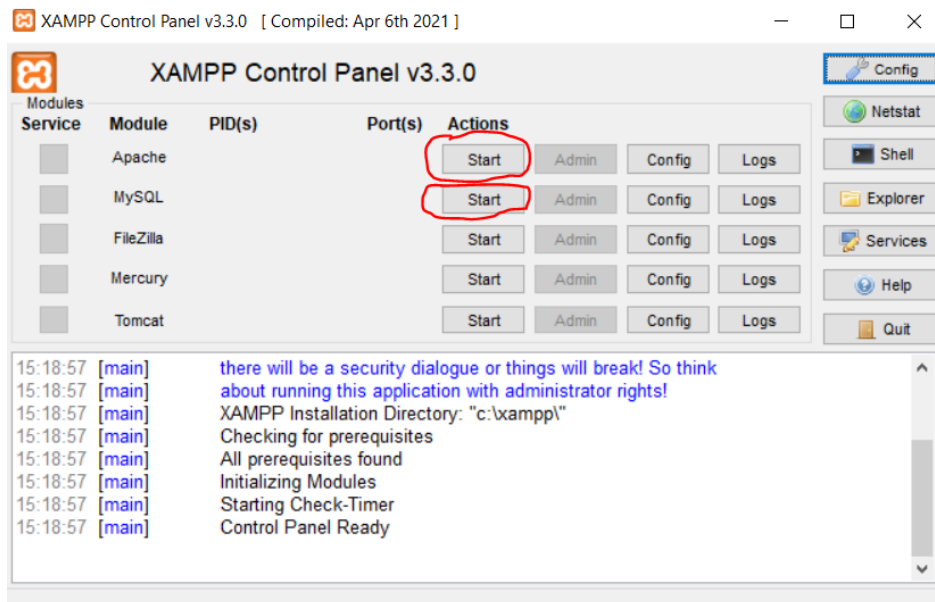


Ilustración 103: Panel de control de XAMPP (1)

Podemos cambiar los puertos tanto de Apache como de MySQL:

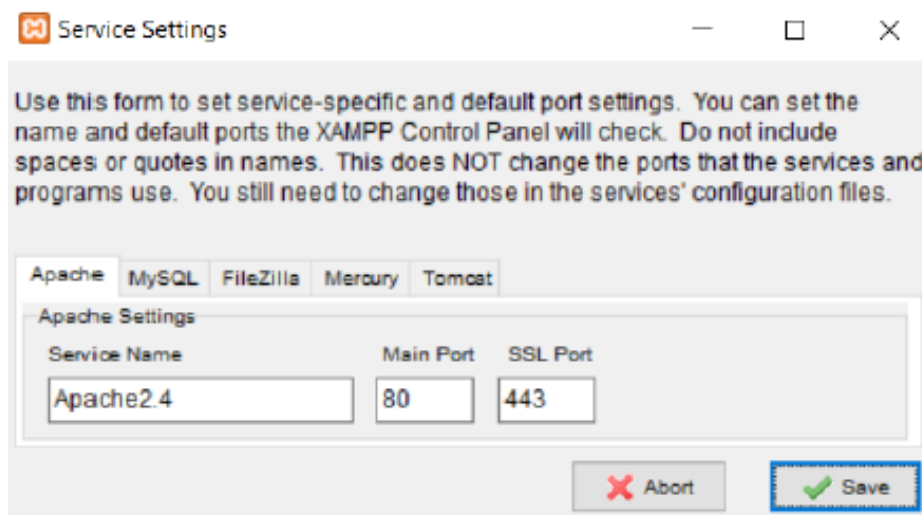


Ilustración 104: Panel de control de XAMPP (2)

Configuración de la base de datos

Una vez iniciados ambos servicios, empezaremos por crear la base de datos de nuestro proyecto. Para ello pulsaremos el botón 'Admin':

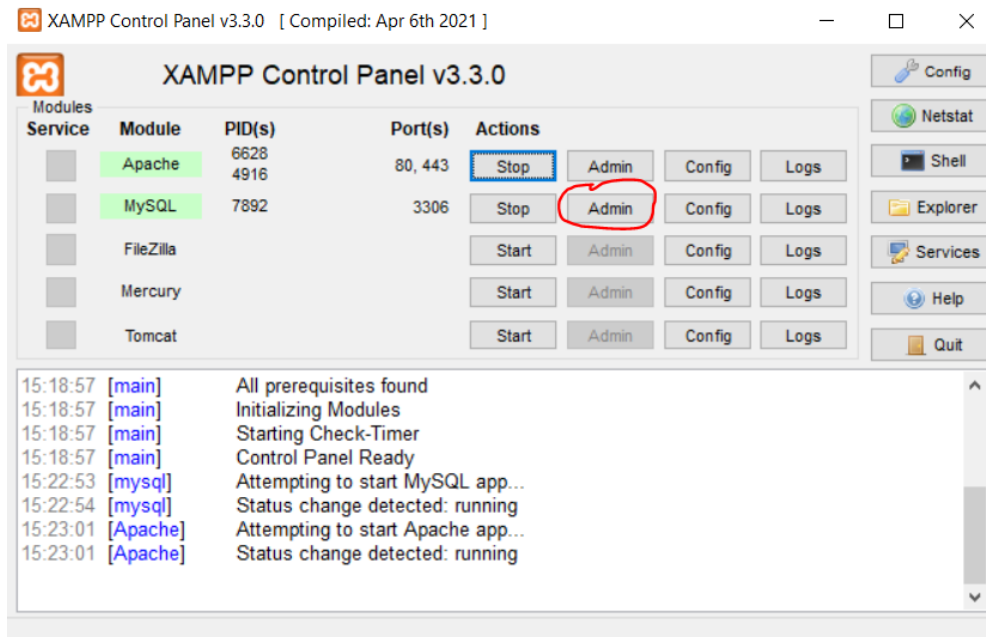


Ilustración 105: Panel de control de XAMPP (3)

Entraremos en **phpMyAdmin** donde podremos crear y configurar nuestras bases de datos. Para crear una nueva base de datos pulsaremos en 'Nuevo':

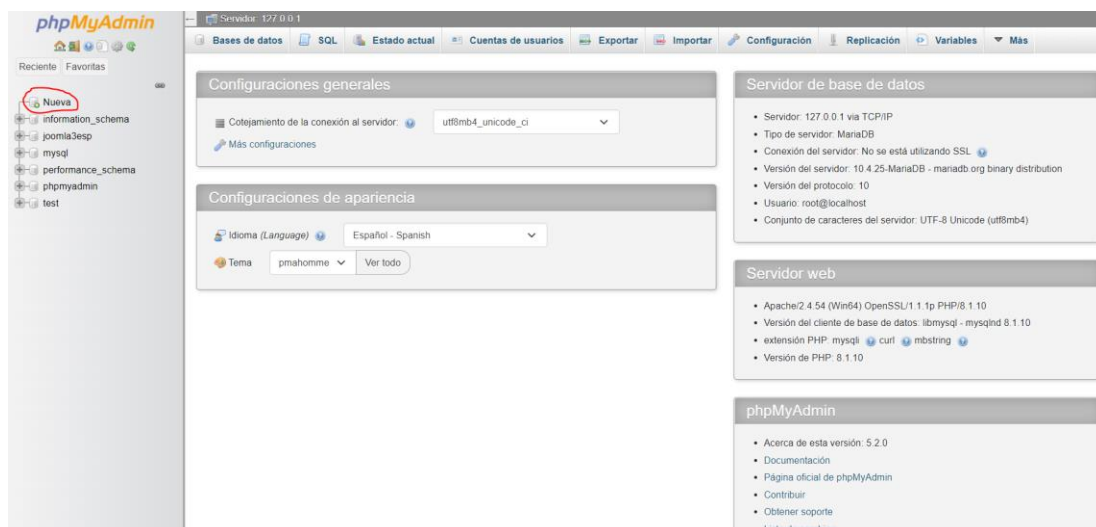


Ilustración 106: phpMyAdmin (1)

Le daremos un nombre a nuestra base de datos y elegiremos el cotejamiento (recomiendo dejar el que sale por defecto):

Bases de datos

Crear base de datos

MiProyectoJoomla utf8mb4_general_ci **Crear**

Seleccionar todo Eliminar

	Base de datos	Cotejamiento	Acción
<input type="checkbox"/>	information_schema	utf8_general_ci	Seleccionar privilegios
<input type="checkbox"/>	joomla3esp	utf8mb4_general_ci	Seleccionar privilegios
<input type="checkbox"/>	mysql	utf8mb4_general_ci	Seleccionar privilegios
<input type="checkbox"/>	performance_schema	utf8_general_ci	Seleccionar privilegios
<input type="checkbox"/>	phpmyadmin	utf8_bin	Seleccionar privilegios
<input type="checkbox"/>	test	latin1_swedish_ci	Seleccionar privilegios
Total: 6			

Ilustración 107: phpMyAdmin (2)

Después de crearla veremos que la base de datos está vacía y que podremos añadir nosotros mismo columnas. Por ahora no tocaremos nada más. Joomla durante su instalación se encargará de introducir todas las columnas necesarias para el proyecto.

phpMyAdmin

Servidor: 127.0.0.1 Base de datos: miproyectojoomla

Estructura SQL Buscar Generar una consulta Exportar Importar Operaciones Privilegios Rutinas Eventos Disparadores

No se han encontrado tablas en la base de datos.

Crear nueva tabla

Nombre de la tabla Número de columnas

4 **Crear**

Ilustración 108: phpMyAdmin (3)

Configuración e instalación de Joomla

Una vez hayamos hecho estos pasos procederemos a instalar nuestro proyecto Joomla. Para ello iremos a su página de descarga a través del enlace <https://downloads.joomla.org/es>.



Ilustración 109: Descargar Joomla (1)

Una vez aquí pulsaremos en el apartado de descargas para poder escoger la versión que queramos. En este caso escogeré la versión Joomla 3 ya que tiene mayor número de plugins que su versión Joomla 4:



Ilustración 110: Descargar Joomla (2)

Pulsamos en 'Ver lanzamientos en esta categoría'. Después pulsaremos en 'Ver archivos'.

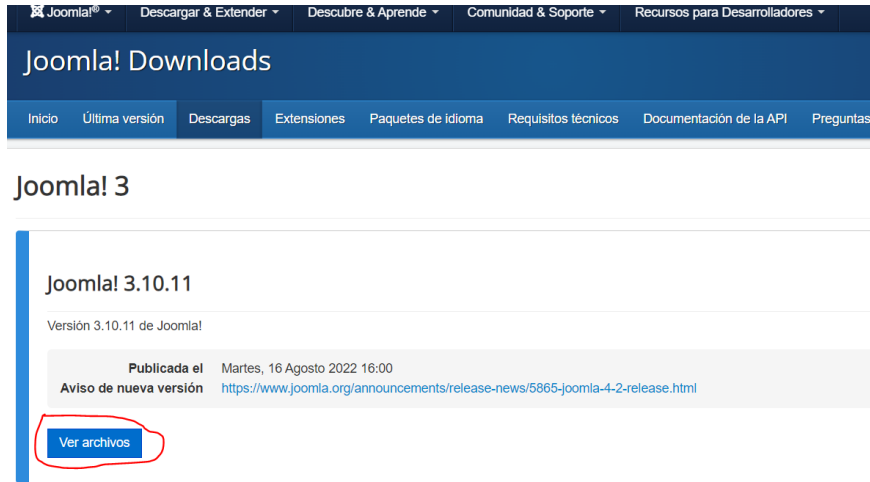


Ilustración 111: Descargar Joomla (3)

En esta ventana veremos los diferentes formatos en los que podremos descargar Joomla. En este caso escogeremos el formato de .zip



Ilustración 112: Descargar Joomla (4)

Una vez descargado el .zip, procedemos a extraerlo (recomiendo extraerlo en la propia carpeta ya que si usas la opción de 'Extraer aquí' descomprime todos los archivos sin carpeta):

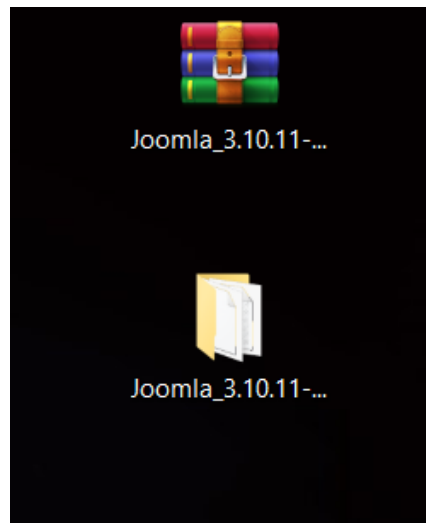


Ilustración 113: .zip de Joomla

Ahora deberemos ir a la carpeta donde hemos instalado XAMPP anteriormente. Después de localizar la carpeta, accederemos a su subcarpeta 'htdocs' y aquí pegaremos el archivo Joomla que hemos descomprimido (recomiendo renombrar la carpeta de Joomla antes de pegarla en este directorio).

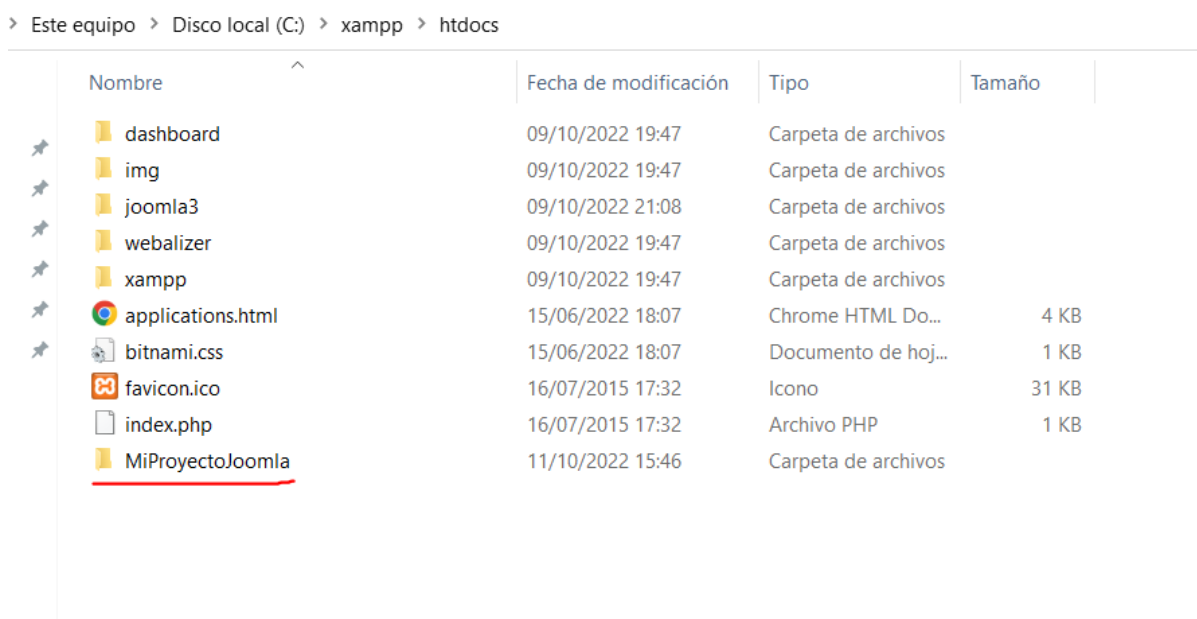


Ilustración 114: Carpeta donde se aloja nuestro proyecto

Una vez realizado estos pasos debemos acceder al instalador de Joomla. Para ello accederemos a nuestro localhost por el puerto que hayamos elegido más arriba (por defecto 80). El formato sería el siguiente: localhost/nombredemiproyecto. Una vez hayamos accedido al instalador rellenaremos los diferentes campos como queremos (importante acordarse).

The image shows the Joomla! installation configuration page in a browser. The URL is http://localhost/MProyectoJoomla/installation/index.php. The Joomla! logo is at the top, followed by the text "Joomla! es software libre liberado bajo la GNU General Public License." Below this, there are three tabs: "Configuración" (selected), "Base de datos", and "Visión general". Under "Configuración", there is a dropdown menu for "Seleccionar el idioma" set to "Español (España)" and a "Siguiente" button. The main section is titled "Configuración principal" and is divided into two columns. The left column has "Nombre del sitio" (set to "Checklozato"), "Descripción" (set to "Mi aplicación"), and a "Sitio fuera de línea" checkbox (checked). The right column is titled "Detalles de la cuenta de súper usuario" and includes fields for "El correo electrónico" (ajp00015@red.ujem.es), "Número de usuario" (ajp00015), "Contraseña", and "Confirmar la contraseña". A "Siguiente" button is at the bottom right.

Ilustración 115: Instalador Joomla (1)

Después pasaremos a la parte de la base de datos. En usuario elegiremos 'root' si previamente no hemos creado otro usuario. Lo mismo pasará en el campo de 'contraseña' el cual dejaremos vacío. Importante elegir el mismo nombre de base de datos que el que hemos creado anteriormente en **phpMyAdmin**. El prefijo de las tablas será único para cada proyecto y siempre irá antes que el nombre de la tabla. Ej. jnyf2_access.

Joomla! es software libre liberado bajo la GNU General Public License.

1 Configuración 2 Base de datos 3 Visión general

Configuración de la base de datos

← Anterior → Siguiente

Tipo de base de datos * MySQL
Probablemente sea "mysql"

Hospedaje * localhost
Normalmente es "localhost" o el nombre proporcionado por su hospedaje.

Usuario * root
El nombre de usuario que haya elegido o el facilitado por quien le sirva el hospedaje.

Contraseña
Por cuestiones de seguridad, es primordial usar una contraseña para la cuenta de su base de datos.

Base de datos * miproyectojoomla
En algunos hospedajes solo se permite el nombre específico de una base de datos por sitio. En esos casos, si le interesa instalar más de un sitio, puede usar el prefijo de las tablas para distinguir entre los sitios de Joomla! que usen la misma base de datos.

Prefijo de las tablas * jnyf2_
Cree un prefijo para la base de datos o use el generado aleatoriamente. Lo óptimo es que sea de cuatro o cinco caracteres de largo y que tenga solo caracteres alfanuméricos, y DEBE acabar con un guión bajo. Asegúrese de que el prefijo elegido no esté siendo usado por otras tablas.

Proceso para una base de datos antigua * Respaldar Borrar
"Respaldar" o "Eliminar" cualquier respaldo existente de tablas pertenecientes a Joomla! que usen el mismo "prefijo de la tabla".

← Anterior → Siguiente

Ilustración 116: Instalador Joomla (2)

Finalizamos el proceso:

Joomla! es software libre liberado bajo la GNU General Public License.

1 Configuración 2 Base de datos 3 Visión general

Configuración de la base de datos

← Anterior → Siguiente

Tipo de base de datos * MySQL
Probablemente sea "mysql"

Hospedaje * localhost
Normalmente es "localhost" o el nombre proporcionado por su hospedaje.

Usuario * root
El nombre de usuario que haya elegido o el facilitado por quien le sirva el hospedaje.

Contraseña
Por cuestiones de seguridad, es primordial usar una contraseña para la cuenta de su base de datos.

Base de datos * miproyectojoomla
En algunos hospedajes solo se permite el nombre específico de una base de datos por sitio. En esos casos, si le interesa instalar más de un sitio, puede usar el prefijo de las tablas para distinguir entre los sitios de Joomla! que usen la misma base de datos.

Prefijo de las tablas * jnyf2_
Cree un prefijo para la base de datos o use el generado aleatoriamente. Lo óptimo es que sea de cuatro o cinco caracteres de largo y que tenga solo caracteres alfanuméricos, y DEBE acabar con un guión bajo. Asegúrese de que el prefijo elegido no esté siendo usado por otras tablas.

Proceso para una base de datos antigua * Respaldar Borrar
"Respaldar" o "Eliminar" cualquier respaldo existente de tablas pertenecientes a Joomla! que usen el mismo "prefijo de la tabla".

← Anterior → Siguiente

Ilustración 117: Instalador Joomla (3)

Ya tenemos Joomla instalado. Como pone en el mensaje. Deberemos ir a la carpeta donde está nuestro proyecto y dentro del mismo borrar la carpeta 'installation':



Joomla!

Joomla! es software libre liberado bajo la GNU General Public License.

¡Felicidades! Ahora Joomla! ya está instalado.

Joomla! en su propio idioma o creación de un sitio multiidioma básico

Antes de borrar la carpeta "installation" puede instalar más idiomas. Si desea añadir más idiomas, seleccione el siguiente botón.

[→ Pasos extra: Instalar idiomas](#)

Nota: necesitará conexión a internet para que Joomla pueda [descargar e instalar los nuevos idiomas](#). Algunas configuraciones del servidor no permiten que Joomla pueda instalar los idiomas. Si este fuera su caso, no se preocupe, los podrá instalar después desde la administración del CMS.

POR FAVOR, ACUÉRDESE DE ELIMINAR COMPLETAMENTE EL DIRECTORIO DE INSTALACIÓN.
No podrá continuar usando Joomla! con normalidad hasta que la carpeta "installation" sea eliminada. Es una característica de seguridad de Joomla!

[Eliminar carpeta](#)

[Sitio](#) [Administración](#)

Detalles de acceso a la administración

Correo electrónico:

Usuario:

Ilustración 118: Instalador Joomla (3)

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

Nombre	Fecha de modificación	Tipo	Tamaño
★ administrador	14/08/2022 15:21	Carpeta de archivos	
★ bin	14/08/2022 15:21	Carpeta de archivos	
★ cache	14/08/2022 15:21	Carpeta de archivos	
★ cli	14/08/2022 15:21	Carpeta de archivos	
★ components	14/08/2022 15:21	Carpeta de archivos	
★ images	14/08/2022 15:21	Carpeta de archivos	
★ includes	14/08/2022 15:21	Carpeta de archivos	
★ installation	14/08/2022 15:21	Carpeta de archivos	
language	14/08/2022 15:21	Carpeta de archivos	
layouts	14/08/2022 15:21	Carpeta de archivos	
libraries	14/08/2022 15:21	Carpeta de archivos	
media	14/08/2022 15:21	Carpeta de archivos	
modules	14/08/2022 15:21	Carpeta de archivos	
plugins	14/08/2022 15:21	Carpeta de archivos	
templates	14/08/2022 15:21	Carpeta de archivos	
tmp	14/08/2022 15:21	Carpeta de archivos	
configuration.php	11/10/2022 15:51	Archivo PHP	3 KB
htaccess.txt	14/08/2022 15:21	Documento de tex...	4 KB
index.php	14/08/2022 15:21	Archivo PHP	2 KB
LICENSE.txt	14/08/2022 15:21	Documento de tex...	18 KB
README.txt	14/08/2022 15:21	Documento de tex...	5 KB
robots.txt.dist	14/08/2022 15:21	Archivo DIST	1 KB
web.config.txt	14/08/2022 15:21	Documento de tex...	2 KB

Ilustración 119: Eliminar carpeta installation

Si pulsamos en el botón de administración accederemos al administrador de nuestro proyecto en Joomla:

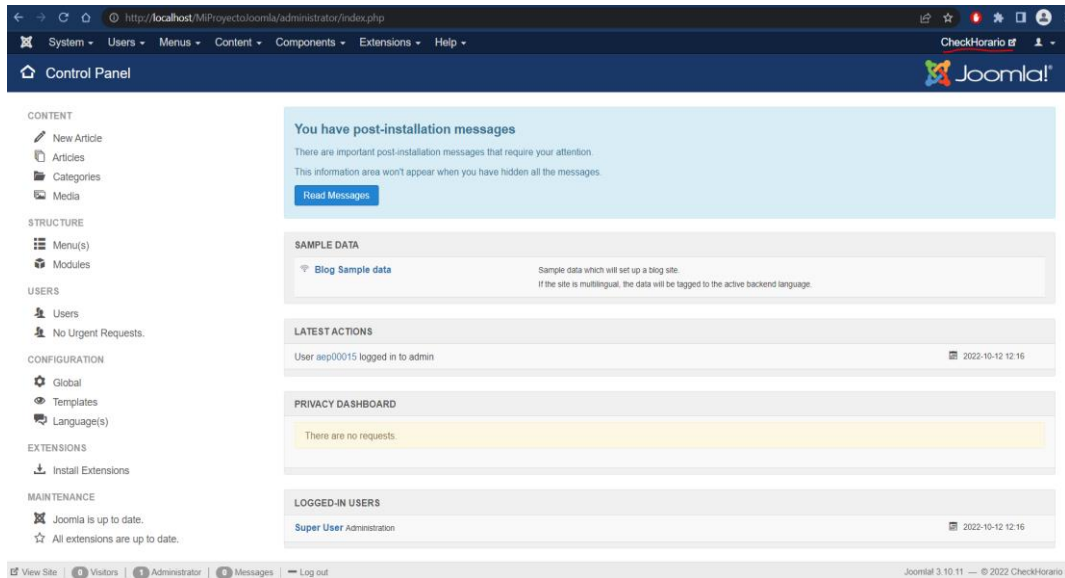


Ilustración 120: Administrador Joomla

Pulsaremos al nombre de nuestro proyecto que se encuentra en la esquina para ver la aplicación:

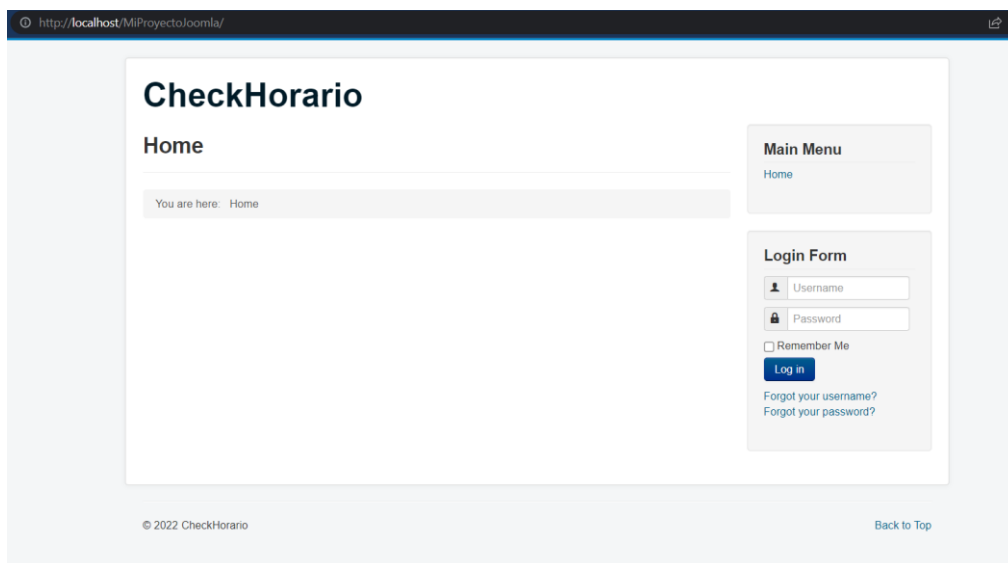


Ilustración 121: Página de nuestro proyecto

Elementos de Joomla

Usuarios

Primero mostraré como crear un usuario nuevo. Para ello desde el administrador de nuestro proyecto pulsaremos en el desplegable de 'Users':

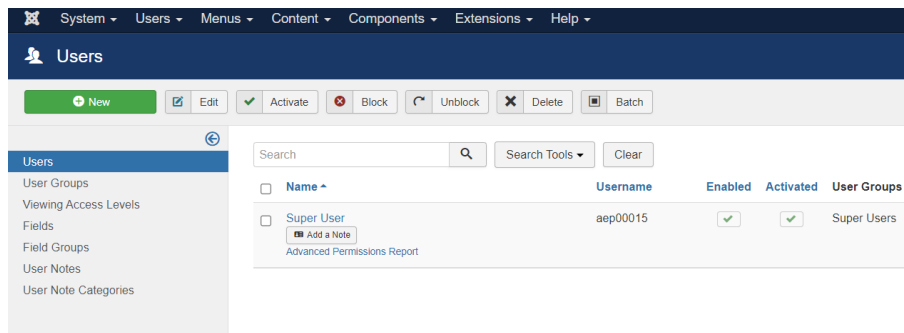


Ilustración 122: Usuarios Joomla (1)

Vemos que nuestro usuario tiene por defecto el rol de 'Super User' que es el que puede modificar la aplicación desde el administrador.

Pulsaremos en 'New'. Nos aparecerá un formulario el cual debemos rellenar.

Ilustración 123: Usuarios Joomla (2)

Si pulsamos en la pestaña de 'Assigned User Groups' podremos ver los diferentes roles que podemos asignarle a un usuario. Por defecto nos selecciona el de 'Registered'.

A continuación, hablaré brevemente de los roles básicos de Joomla:

Hay cuatro (4) grupos de frontend disponibles: [23]

Registered - Este grupo permite al usuario acceder a la interfaz del Frontend. Los usuarios registrados no pueden contribuir con contenido, pero esto puede permitirles acceder a otras áreas, como un foro o una sección de descargas si su sitio tiene una.

Author - Este grupo permite a un usuario publicar contenido, normalmente a través de un enlace en el Menú de Usuario. Pueden enviar nuevos contenidos, seleccionar las opciones para mostrar el artículo en la página principal y seleccionar las fechas de publicación, pero no pueden publicar directamente ningún contenido. Cuando un usuario de nivel Autor envía un contenido, recibe el mensaje 'Gracias por su envío. Su envío será revisado antes de ser publicado en el sitio'. Sólo pueden editar sus propios artículos, pero sólo cuando ese artículo ha sido publicado y es visible.

Editor - Este grupo permite a un usuario publicar y editar cualquier artículo de contenido (no sólo el suyo) desde el Frontend. También pueden editar contenido que no haya sido publicado. Si su sitio utiliza la opción de menú 'Noticias' de la instalación por defecto, que es de tipo Lista de Tablas - Sección de Contenido, los Editores verán los artículos no publicados en la lista que pueden seleccionar para editar, mientras que un usuario Autor o Público (no registrado) ni siquiera verá los artículos no publicados en la lista. Aun así, los usuarios de Editor no pueden, publicar o cambiar el estado de publicación de ningún artículo, ni siquiera el suyo propio.

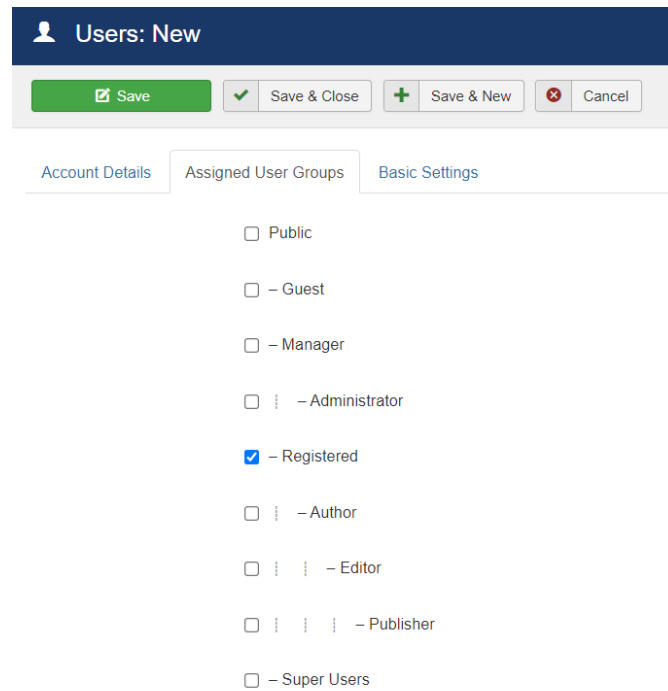
Publisher - Este grupo permite a un usuario enviar, editar y publicar cualquier artículo (no sólo el suyo) desde el frontend. Los editores pueden revisar todos los artículos, editar y cambiar las opciones de publicación, pero también pueden determinar cuándo un artículo está listo para ser publicado, haciéndolo visible para el público registrado, el autor y el público no registrado (dependiendo de la visibilidad elegida en el artículo, por supuesto).

Hay tres (3) grupos de la sección de Administración que permiten el acceso a Joomla:

Manager - Este grupo permite el acceso a la creación de contenidos y otra información del sistema desde el Backend. Piensa en los usuarios del Gestor como Editores, con acceso al Backend. Pueden iniciar sesión a través de la interfaz del Administrador, pero sus derechos y acceso están generalmente restringidos a la gestión de contenidos. Pueden crear o editar cualquier contenido, acceder a algunas características del Backend como añadir, eliminar y editar Secciones y Categorías, editar la Página Principal y los Menús, pero no tienen ningún acceso a la 'Mecánica' de Joomla, como la gestión de usuarios o la capacidad de instalar componentes o módulos. Tenga en cuenta que, si un Administrador se conecta a través de la interfaz de Frontend, son tratados igual que un Editor, con los mismos derechos y acceso.

Administrator - Este grupo permite el acceso a la mayoría de las funciones de administración. Un usuario Administrador tiene todos los privilegios en el backend de un Gestor, pero también tiene acceso a establecer opciones en, e instalar/borrar componentes, módulos y plugins, acceso al Gestor de Usuarios y puede ver las estadísticas del sitio. Lo que no pueden hacer, sin embargo, es cambiar, editar o instalar Plantillas de Sitio o hacer cualquier cambio en las opciones de configuración Global de los sitios. En el acceso a través del Frontend, son tratados como Editores, al igual que los usuarios Gestores. Es interesante notar que cuando un Administrador accede a la lista del Gestor de Usuarios, verá todos los usuarios de su nivel de acceso o inferior; en otras palabras, puede modificar cualquier usuario EXCEPTO un Súper Administrador - de hecho, ¡ni siquiera verá las cuentas de Súper Administrador en la lista! Además, no pueden crear cuentas adicionales de nivel de superadministrador, sólo un superadministrador puede hacerlo.

Súper Administrador - Este grupo permite el acceso a todas las funciones de administración. Sólo otro Súper Administrador puede crear o editar una cuenta de usuario de Súper Administrador. Los Super Administradores tienen acceso completo a TODAS LAS ÁREAS, y una vez creadas no pueden ser eliminadas tan fácilmente. Puede que no sea obvio, pero primero tienes que cambiar el grupo del usuario a algo distinto de Super Administrador. Luego puedes eliminarlos.

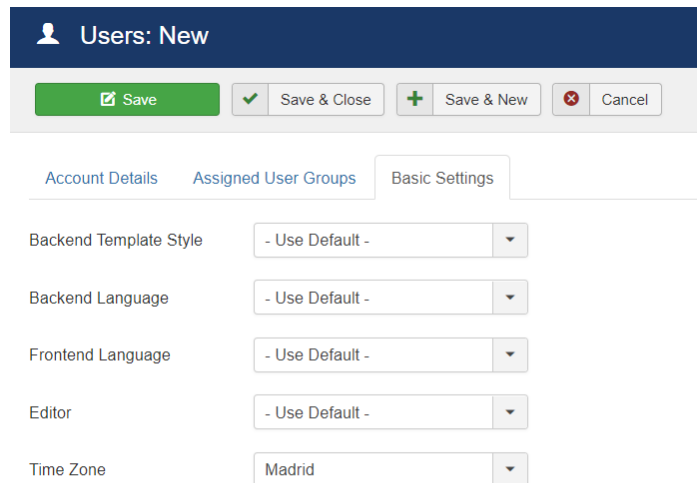


The screenshot shows the Joomla! 'Users: New' form with the 'Assigned User Groups' tab selected. The form has a dark blue header with a user icon and the text 'Users: New'. Below the header is a toolbar with four buttons: 'Save' (green), 'Save & Close' (green checkmark), 'Save & New' (green plus), and 'Cancel' (red X). Below the toolbar are three tabs: 'Account Details', 'Assigned User Groups' (active), and 'Basic Settings'. The 'Assigned User Groups' tab contains a list of user groups with checkboxes:

- Public
- Guest
- Manager
- Administrator
- Registered
- Author
- Editor
- Publisher
- Super Users

Ilustración 124: Usuarios Joomla (3)

Además de los roles podemos acceder a la pestaña de 'Basic settings' para configurar algunos ajustes sobre el usuario:



The screenshot shows the Joomla! 'Users: New' form with the 'Basic Settings' tab selected. The form has a dark blue header with a user icon and the text 'Users: New'. Below the header is a toolbar with four buttons: 'Save' (green), 'Save & Close' (green checkmark), 'Save & New' (green plus), and 'Cancel' (red X). Below the toolbar are three tabs: 'Account Details', 'Assigned User Groups', and 'Basic Settings' (active). The 'Basic Settings' tab contains five configuration options, each with a dropdown menu:

- Backend Template Style: - Use Default -
- Backend Language: - Use Default -
- Frontend Language: - Use Default -
- Editor: - Use Default -
- Time Zone: Madrid

Ilustración 125: Usuarios Joomla (4)

Una vez rellenado el formulario, escogido el rol que queremos asignarle al usuario y configurados sus ajustes básicos, pulsaremos 'Save'.

Ilustración 126: Usuarios Joomla (5)

Tras guardar el usuario podremos verlo en la pestaña de 'Users'

Name	Username	Enabled	Activated	User Groups
Francisco Garzón	pepeg	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Registered
Super User	aep00015	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Super Users

Ilustración 127: Usuarios Joomla (6)

Menús

Los menús son donde veremos los elementos que forman la aplicación web. Estos elementos pueden ser desde artículos hasta plugins. Pueden existir varios menús a la vez y utilizar los que queramos en nuestra aplicación, aunque, lo más común es tener un menú principal por el cual navegar los diferentes elementos de nuestra aplicación.

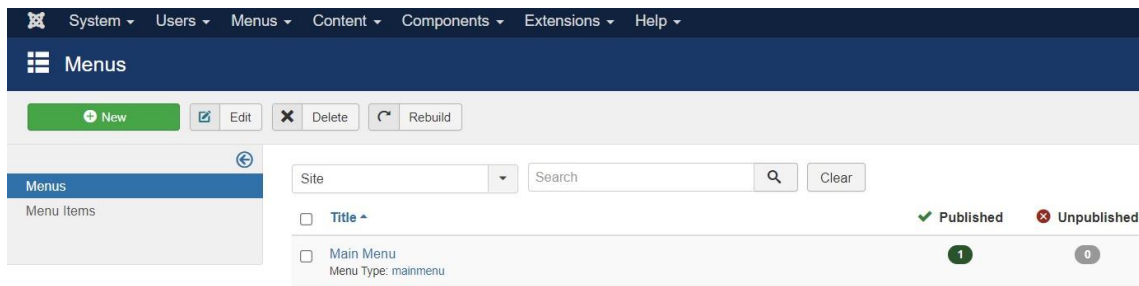


Ilustración 128: Menús Joomla (1)

Una vez pulsemos en el menú podemos ver los elementos que contiene.

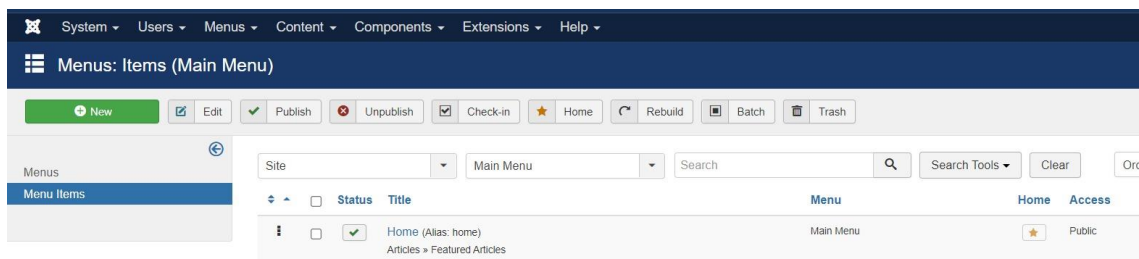


Ilustración 129: Menús Joomla (2)

Ahora vamos a añadir un nuevo elemento a nuestro menú:

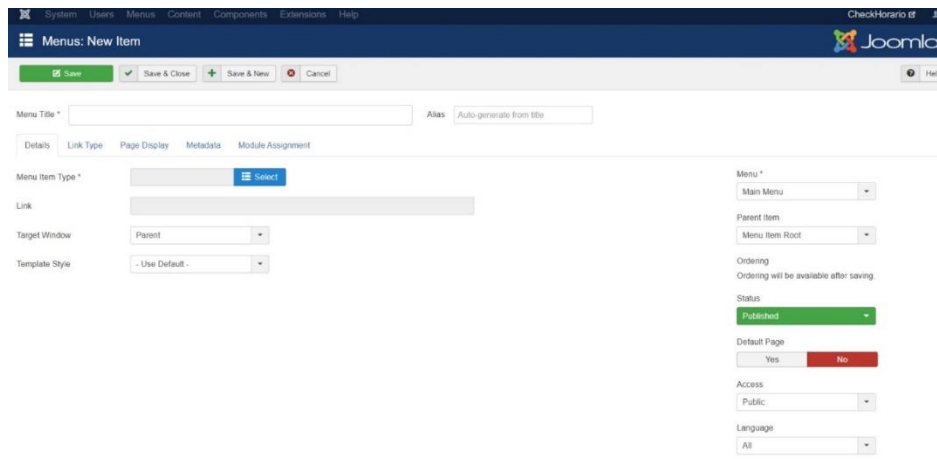
The image shows the Joomla! administration interface for creating a new menu item. The title bar reads 'Menus: New Item'. Below the title bar are buttons for 'Save', 'Save & Close', 'Save & New', and 'Cancel'. The form is divided into several sections: 'Menu Title' with an input field and an 'Alias' dropdown set to 'Auto-generate from title'; 'Details' (selected), 'Link Type', 'Page Display', 'Metadata', and 'Module Assignment'. The 'Menu Item Type' dropdown is set to 'Form' with a 'Select' button. The 'Link' field is empty. 'Target Window' is set to 'Parent' and 'Template Style' is set to '- Use Default -'. On the right side, there are dropdowns for 'Menu' (Main Menu), 'Parent Item' (Menu Item Root), 'Status' (Published), 'Default Page' (Yes/No), 'Access' (Public), and 'Language' (All).

Ilustración 130: Menús Joomla (3)

Elegimos el tipo de elemento que queremos introducir (en este caso un formulario de login):

The image shows a 'Menu Item Type' selection dialog box. It contains a list of menu item types: Articles, Configuration, Contacts, News Feeds, Privacy, Search, Smart Search, System Links, Tags, and Users. The 'Users' category is expanded, showing sub-items: 'Edit User Profile' (Edit a user profile), 'Login Form' (Displays a login form), 'Logout' (Direct logout and redirect to page), and 'Password Reset' (Displays a request to reset password). A 'Close' button is located at the bottom right of the dialog.

Ilustración 131: Menús Joomla (4)

System Users Menu Content Components Extensions Help

CheckHorario Joomla!

Menus: New Item

Save Save & Close Save & New Cancel Help

Menu Title * Login form Alias Auto-generate from title

Details Options Link Type Page Display Metadata Module Assignment

Menu Item Type * Login Form Select

Link index.php?option=com_users&view=login

Target Window Parent

Template Style - Use Default -

Menu * Main Menu

Parent Item Menu Item Root

Ordering Ordering will be available after saving.

Status Published

Default Page Yes No

Access Public

Ilustración 132: Menús Joomla (5)

Hacemos click en 'Save' y nuestro elemento ya formará parte de nuestro menú.

System Users Menu Content Components Extensions Help

Menus: Items (Main Menu)

New Edit Publish Unpublish Check-in Home Rebuild Batch Trash

Site Main Menu Search Search Tools Clear Order

	Status	Title	Menu	Home	Access
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Home (Alias: home) Articles » Featured Articles	Main Menu	<input checked="" type="checkbox"/>	Public
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Login form (Alias: login-form) Users » Login Form	Main Menu	<input type="checkbox"/>	Public

Ilustración 133: Menús Joomla (6)

Aquí podemos verlo en el front de nuestra aplicación:

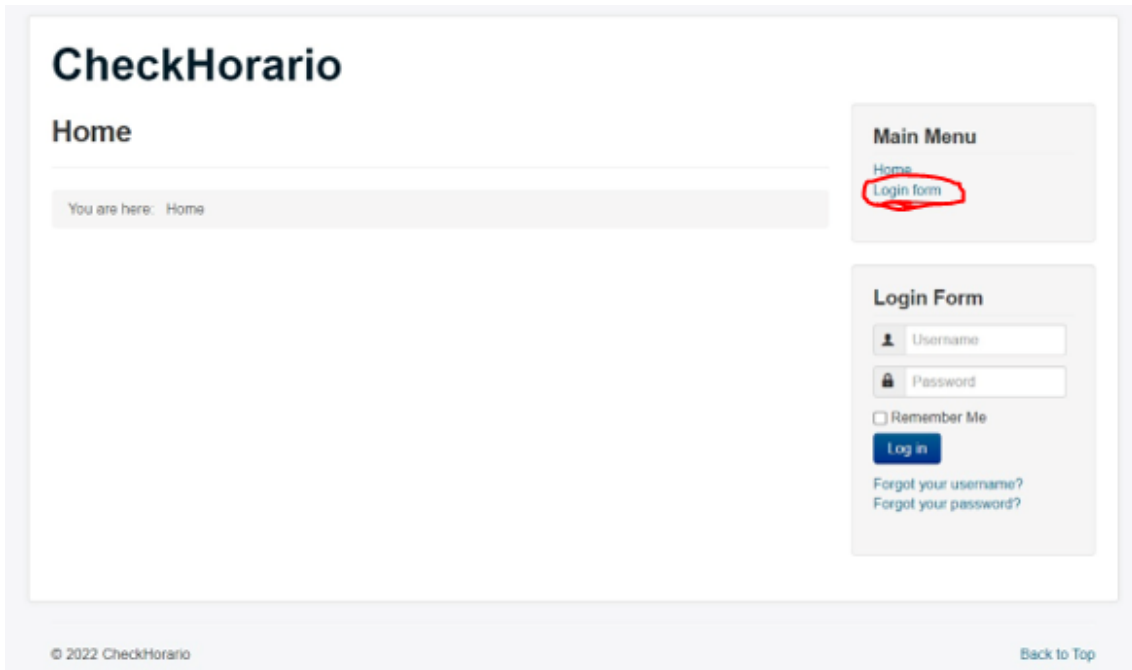


Ilustración 134: Menús Joomla (7)

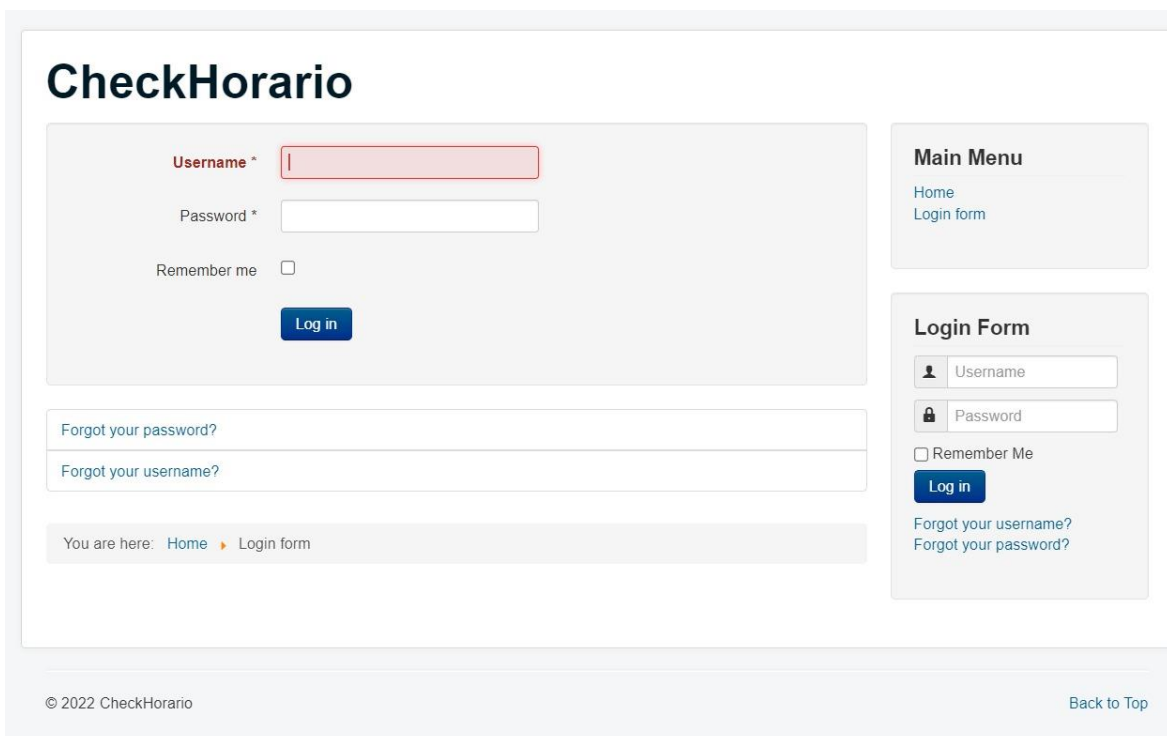


Ilustración 135: Menús Joomla (8)

Artículos

Los artículos son elementos generalmente formados por textos (aunque pueden ser imágenes). Como su nombre indica podemos pensar en estos elementos como un artículo escrito en un periódico o publicado en una página web de noticias.

Para crear un nuevo artículo iremos a: Content > Articles. Para crear un nuevo artículo pulsaremos en el botón de 'New'.

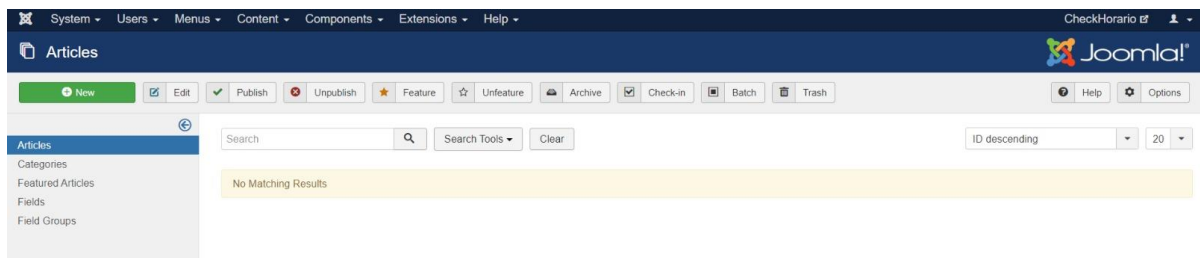


Ilustración 136: Artículos Joomla (1)

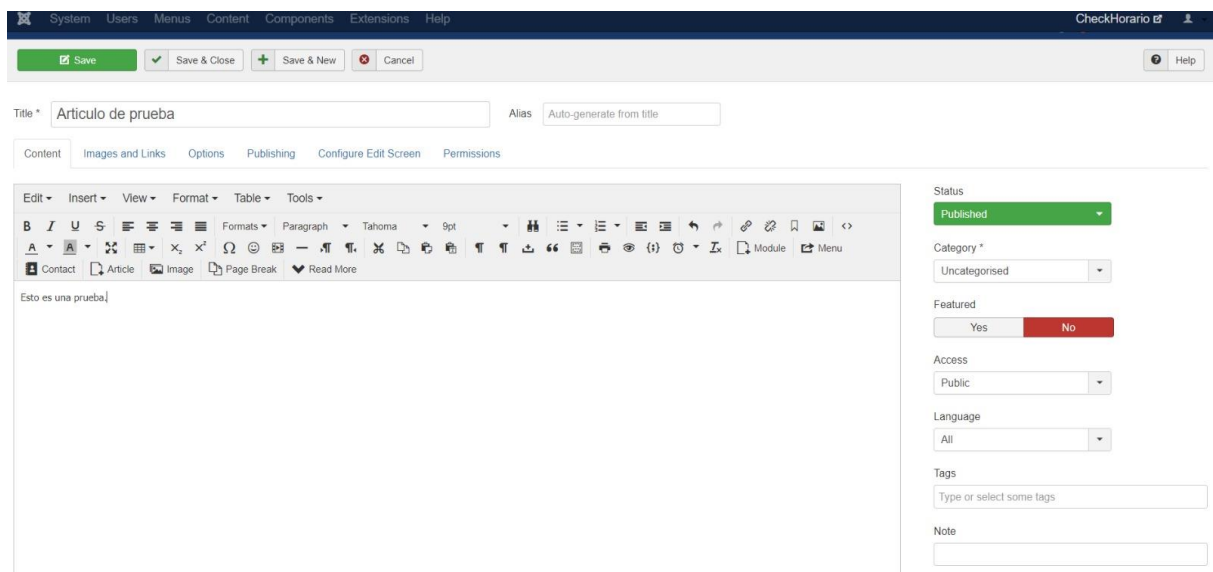


Ilustración 137: Artículos Joomla (2)

Como la mayoría de los elementos de Joomla podemos indicar que grupos de usuarios tienen acceso al mismo. Además, podemos crear categorías de artículos. Ej. Artículo médico.

Ahora veamos el artículo publicado en nuestro proyecto:

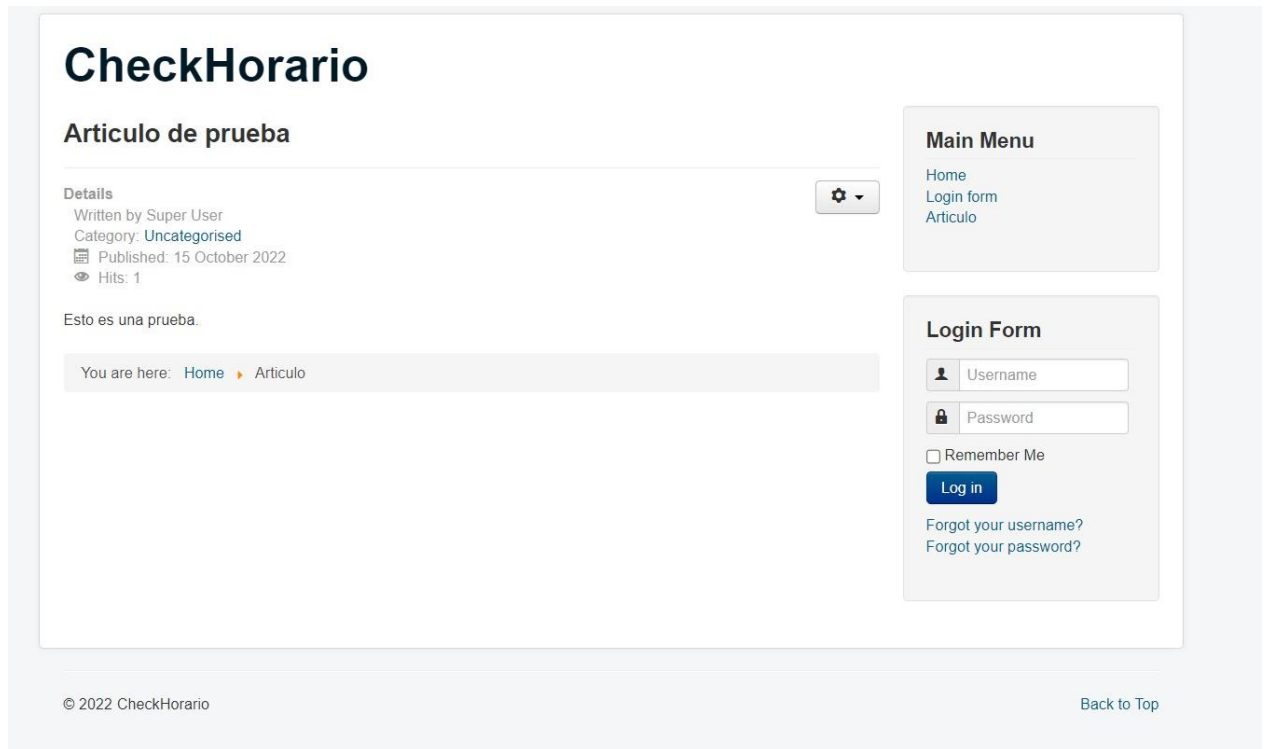


Ilustración 138: Artículos Joomla (3)

Extensiones

Joomla nos permite utilizar y modificar elementos propios del CMS. Ej. Login Form. Los módulos hacen referencia a estas utilidades predefinidas por el CMS. Estos módulos pueden ser usados libremente, aunque, presentan algunas limitaciones.

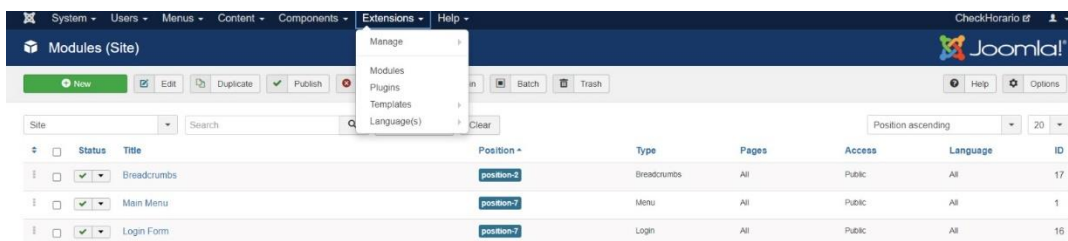


Ilustración 139: Extensiones Joomla (1)

Otro tipo de módulo son las plantillas. Las plantillas son las encargadas de darle un estilo predefinido a nuestra web. Podemos tener tantas plantillas queramos y podemos elegir la que más nos guste.

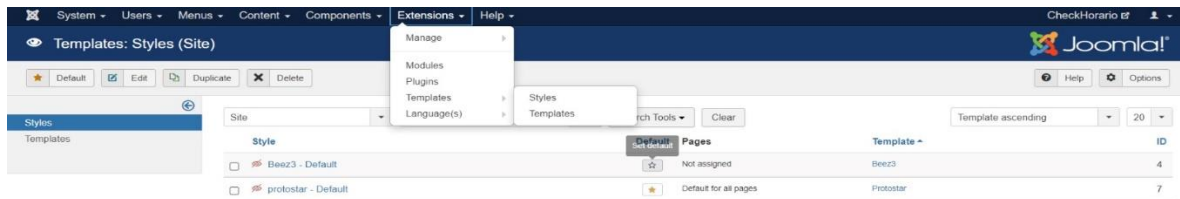


Ilustración 140: Extensiones Joomla (2)

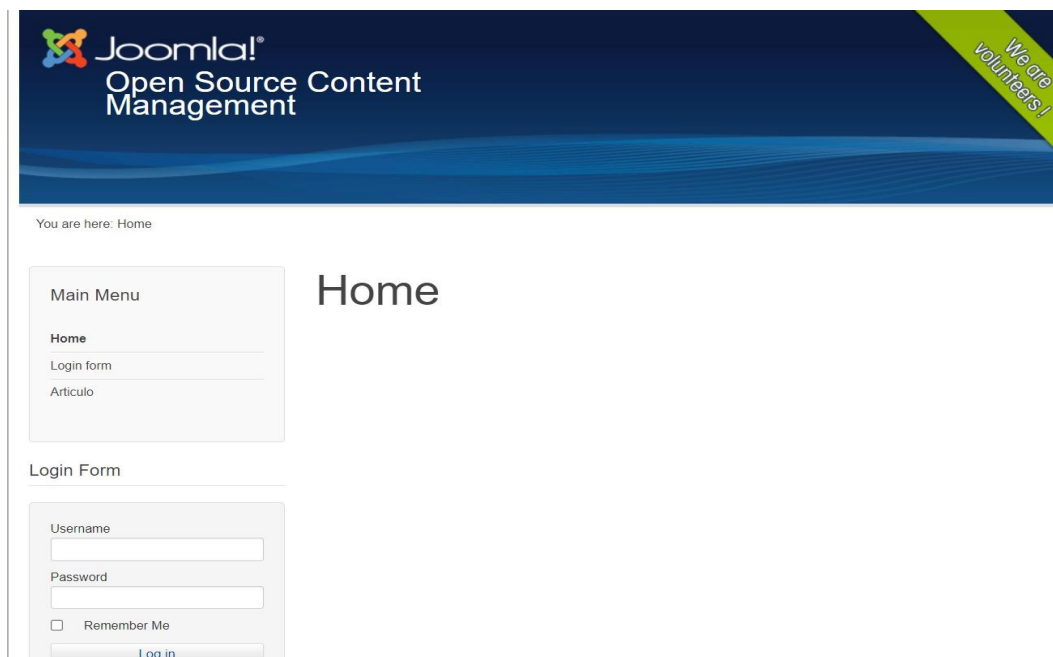


Ilustración 141: Extensiones Joomla (3)

Existen otro tipo de extensiones (o plugins) que podemos encontrar externamente e instalarlas para nuestro proyecto Joomla. Hablaré más en profundidad de ellas en el capítulo de implementación ya que son muy variadas y situacionales

BIBLIOGRAFÍA

- [1] «¿Por qué es bueno el horario flexible en las empresas?,» [En línea]. Available: <https://www.workmeter.com/blog/por-que-es-bueno-el-horario-flexible/>.
- [2] «Gestor de contenidos,» [En línea]. Available: <https://www.lucushost.com/blog/gestor-de-contenidos/>.
- [3] «What is cms?,» [En línea]. Available: <https://www.oracle.com/content-management/what-is-cms/>.
- [4] A. a. d. o. cms. [En línea]. Available: <https://rentechdigital.com/blog/b/advantages-and-disadvantages-of-cms>.
- [5] D. Mazier, de *Joomla! 3.8: cree y administre sus sitios web*.
- [6] H. Graf, Joomla! - Guía para Principiantes.
- [7] C. O. P. Qué Es Xampp Usos. [En línea]. Available: <https://mundobytes.com/xampp/>.
- [8] ¿. e. Joomla?. [En línea]. Available: <https://cei.es/que-es-joomla/>.
- [9] «GanttProject, una buena herramienta para gestionar tus proyectos, ya está disponible en nuestro catálogo de aplicaciones.,» [En línea]. Available: <https://www.bilib.es/actualidad/blog/noticia/articulo/ganttproject-una-buena-herramienta-para-gestionar/#:~:text=GanttProject%20es%20una%20herramienta%20de,una%20visualizaci%C3%B3n%20absoluta%20del%20mismo..>
- [10] «What is Figma?,» [En línea]. Available: <https://webdesign.tutsplus.com/es/articles/what-is-figma--cms-32272>.
- [11] ¿. u. t. l. i. d. software?. [En línea]. Available: <https://www.teknei.com/2021/11/09/que-es-y-que-utilidad-tiene-la-ingenieria-de-software/>.
- [12] «Etapas del proceso de desarrollo de software,» [En línea]. Available: <https://www.solbyte.com/blog/5-etapas-del-proceso-de-desarrollo-de-software/>.
- [13] «COMPARATIVAS DE LOS MODELOS DE CICLO DE VIDA,» [En línea]. Available: <http://www.repositorio.unacar.mx/jspui/bitstream/1030620191/199/1/acalan%2074-2.pdf>.
- [14] Sommerville, Ingeniería del software.
- [15] R. C. Morales, Introducción Al Análisis de Sistemas Y la Ingeniería de Software.
- [16] J. G. M. C. Ahmed Seffah, Human-Centered Software Engineering - Integrating Usability in the Software Engineering.
- [17] B. R. M. Roger S. Pressman, Software engineering : a practitioner's approach.
- [18] M. Popovic, Communication Protocol Engineering.
- [19] R. A. M. De Eric J. Naiburg, UML for Database Design.
- [20] S. N. Chapman, Planificación y control de la producción.
- [21] J. B. d. Areba, Metodología del análisis estructurado de sistemas.
- [22] «Introducción a la Accesibilidad Web,» [En línea]. Available: <https://www.w3.org/WAI/fundamentals/accessibility-intro/es>.
- [23] «Users: Group access levels explained in simple terms,» [En línea]. Available: https://docs.joomla.org/J1.5:User_Group_Access_levels_explained_in_simple_terms.
- [24] «<https://www.workmeter.com/blog/por-que-es-bueno-el-horario-flexible/>,» [En línea].
- [25] [En línea]. Available: <https://www.workmeter.com/blog/por-que-es-bueno-el-horario-flexible/>.
- [26] El sitio web, «aa,» 2 1 1998. [En línea].
- [27] «¿Qué es un cms?,» [En línea]. Available: <https://creandoblog.com/diccionario-bloguero/que-es-un-cms/>.