



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Desarrollo de una aplicación Web para la gestión de partes de trabajo en una empresa de ingeniería

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Julio Martí Borrás

Tutor: Félix Buendía

Curso 2015-2016

Agradecimientos

En primer lugar a mis padres, que siempre han estado conmigo. Gracias por todo el apoyo y toda la confianza que habéis tenido en mí, de no ser así no estaría donde estoy ahora.

A mis amigos, porque siempre me han apoyado y me han ayudado en todo lo que necesitaba, a mis compañeros de clase que siempre me sacaban una sonrisa en los peores momentos y siempre han sido un apoyo, y a los profesores que me han ayudado en todo momento.

A mi tutor, Félix Buendía, gracias por todos los consejos que me has facilitado a lo largo de este proyecto.

Resumen

En el presente trabajo, el objetivo principal es poner en marcha una aplicación web para que los trabajadores de una empresa puedan guardar o modificar su parte de trabajo en cualquier lugar y a cualquier hora, de modo que con su usuario puedan introducir o consultar las horas y los días trabajados.

Por otro lado la administración de la empresa podrá extraer información pertinente en base a esos partes de trabajo, como las horas trabajadas por mes, por trabajador, por proyectos o por lo que crean conveniente.

Palabras clave: Parte de trabajo, aplicación web, gestión de empresas

Resum

En el present treball, l'objectiu principal és posar en marxa una aplicació web perquè els treballadors d'una empresa puguin guardar o modificar la seva part de treball en qualsevol lloc i a qualsevol hora, de manera que amb el seu usuari puguin introduir o consultar les hores i els dies treballats.

D'altra banda l'administració de l'empresa pot extreure informació pertinent d'acord amb aquests informes de treball, com les hores treballades per mes, per treballador, per projectes o per el que creguin convenient.

Paraules clau: Part de treball, aplicació web, gestió de empreses

Abstract

In the current work, the principal goal is to start up a web application that supports the employees managing anywhere worksheets, saving or modifying them. Moreover, with their personal user ID they can save or check the days or the worked hours.

Otherwise the company management would be able to gather information from those worksheets, like the worked hours by month, by employee, by project or by any criteria.

Keywords: worksheet, web application, Enterprise management

Tabla de contenidos

1.	Introducción	8
1.1.	Contexto	9
1.2.	Motivación.....	10
1.3.	Objetivo.....	10
1.4.	Método de trabajo.....	11
2.	Propuesta de portal web.....	13
2.1.	Introducción.....	13
2.2.	Especificación de requisitos	14
2.2.1.	Propósito	14
2.2.2.	Ámbito	14
2.2.3.	Perspectiva del producto	15
2.2.4.	Funciones el producto.....	16
2.2.5.	Características del usuario	17
2.3.	Análisis y diseño	20
2.3.1.	Análisis de casos de uso.....	21
2.3.2.	Diagrama de clases.....	27
2.3.3.	Diagramas de secuencia.....	29
2.3.4.	Capa de interfaz	35
2.3.5.	Capa de persistencia	38
2.4.	Detalles de implementación.....	40
2.4.1.	Introducción	40
2.4.2.	Tecnologías y herramientas utilizadas.....	41
2.4.3.	Interfaz y funcionalidades	51
2.4.4.	Implementación de la base de datos	67
3.	Evaluación	69
3.1.1.	Análisis del sitio web	69
4.	Conclusiones	72
5.	Bibliografía	73

1. Introducción

En la actualidad, el uso de Internet es cada vez más relevante. Hace años, Internet se usaba simplemente para buscar información, publicar noticias, y no se le dedicaba demasiado tiempo. Hoy en día se ha convertido en algo habitual, algo cotidiano y lo que nos ocupa, algo que utilizan las empresas.

Antiguamente la información se almacenaba en pilas de papeles, carpetas o archivadores, teniendo el inconveniente de la degradación de los mismos, el espacio que ocupaban o lo engorroso de buscar algo concreto que podía convertirse en una pesadilla en caso de tener poco orden.

En nuestro caso la idea es crear una aplicación web con su respectiva base de datos, de este modo se hará más fácil el manejo y la manipulación de la información.

Esta aplicación web también tiene como objetivo facilitar que el trabajador pueda desde cualquier país introducir sus partes de trabajo, el trabajo realizado y las horas de dicho trabajo, mientras que por otra parte los administradores podrán consultar y sacar datos concluyentes a partir de dichos partes.

1.1. Contexto

Vivimos en una sociedad donde todo avanza muy rápido, y la tecnología es una de las cosas que más rápido avanza, todo el mundo necesita aprender muchos conceptos y muchas herramientas en un corto periodo de tiempo, y las empresas no son ajenas a estos cambios, más bien son las más afectadas.

En las empresas el tiempo empleado para realizar cualquier trabajo juega un papel fundamental en el futuro de la misma, pues en un mundo capitalista como en el que vivimos las empresas necesitan realizar el trabajo con la mayor eficiencia y con el menor coste posible para poder competir en un libre mercado.

Esto lleva a la necesidad del uso de la informática, que se basa en el tratamiento automatizado de la información, lo que antes eran pilas de papeles, carpetas o archivadores, ahora se convierte en algo virtual que no ocupa espacio, de fácil acceso, pero también requieren de algo importante, que esa información sea fácil de crear, modificar y consultar, y aquí entran en juego las bases de datos.

Por tanto las empresas se están viendo obligadas a dejar de lado incluso en algunas ocasiones hojas Excel o Word, donde ya no se utiliza un espacio físico pero se continúa teniendo un desorden y una dificultad a la hora de buscar o tratar cierta información muy importante. Aquí entra en juego este proyecto, donde veremos que la comodidad de crear, modificar o consultar grandes cantidades de información desde cualquier lugar gracias a internet, y de manera rápida y eficaz gracias a las bases de datos.

1.2. Motivación

Uno de los principales factores por los que me he decidido a realizar este tipo de proyecto, es porque tengo un especial interés en conocer cómo se organizan algunas empresas ya que en un futuro me gustaría ser empresario.

Por otro lado aunque en el grado elegí la rama de empresas, tengo también un gran interés en aprender diseño web y ha poder ser programación para saber hacer aplicaciones, entender cómo funcionan y aplicar algunos conocimientos vistos en bases de datos para crear una aplicación que no solo utilice las bases de datos, si no que yo mismo pueda ver cómo trabajan, y precisamente por eso en la medida de lo posible en este proyecto he intentado evitar el uso de frameworks.

Finalmente, el hecho de ser el responsable de crear algo así y de ver cómo funciona y saber que eso lo has hecho tú, me motiva también muchísimo, y precisamente por eso en la medida de lo posible he intentado hacer todo el código posible yo mismo.

1.3. Objetivo

El objetivo principal de este proyecto es realizar una aplicación web para facilitar a una empresa de ingeniería que trabajadores que pueden estar en cualquier país puedan conectarse e introducir fácilmente información relativa al trabajo que están realizando, mientras que la propia empresa pueda consultar fácilmente esa información.

En el caso del trabajador, se autentificara con su usuario y podrá rellenar algo que denominaremos “parte de trabajo”, que consiste en introducir las horas, el proyecto y una pequeña descripción sobre el trabajo que están realizando semanalmente.

Mientras que en el caso de los administradores, podrán no solo ver cada parte de trabajo, si no también hacer pequeños informes para saber cuántas horas han trabajado en un periodo de días, y por supuesto gestionar proyectos, trabajos, y los propios usuarios de la aplicación.

1.4. Estructura de trabajo

Una vez hecha la presentación de este proyecto, lo que interesa describir ahora, es la forma en la que el trabajo va estar estructurado y como se va desarrollar a lo largo de esta memoria.

En primer lugar, este proyecto va dedicado a una empresa concreta con unas funcionalidades concretas, por lo tanto lo primero de todo es averiguar exactamente cuáles son las funcionalidades exactas, como la empresa estaba trabajando hasta este momento, y como hacer un diseño sencillo y parecido al que están utilizando actualmente para que el impacto sobre los trabajadores sea el menor posible.

En segundo lugar, qué tecnologías vamos a utilizar, qué lenguajes de programación, qué tipo de base de datos, en definitiva, que herramientas utilizaremos, para diseñar e implementar nuestra aplicación.

Seguidamente, se realizará la parte de diseño e implementación, donde nos centraremos en describir la página web de forma interna. En este punto tendremos todo lo relacionado con el funcionamiento, desde como comenzar a realizar la web, hasta el paso final, donde ya tendremos algo sólido como sería la interfaz de la web.

En nuestro cuarto punto, una vez terminada la aplicación, ésta se someterá a ciertas pruebas para comprobar que cumple con todo lo establecido y que no se da ningún tipo de problema. Para ello, haremos uso de algunas aplicaciones que se explicaran detalladamente en el punto cuatro.

Desarrollo de una aplicación Web para la gestión de partes de trabajo en una empresa de ingeniería

Por último, el trabajo terminará con un apartado dedicado a conclusiones y trabajos futuros, donde se hablará de las mejoras futuras de la aplicación.

2. Propuesta de aplicación web

2.1. Introducción

Existen aplicaciones como los ERP (enterprise resource planning) que pueden recordar a nuestra aplicación web, aunque estas herramientas ofrecen más herramientas y más genéricas, la finalidad de la creación de esta aplicación es, aunque con menos funcionalidades, tener algo más concreto diseñado específicamente para una empresa.

La aplicación en principio ofrecerá la gestión de partes de trabajo, por parte de los trabajadores poder crearlos, modificarlos y consultarlos, y por parte de los administradores obtener informes con la información de estos partes.

Más adelante se comentara con detalle los tipos de usuario, el tipo de aplicación y sus diferentes funcionalidades así como sus características.

Para llevar a cabo la creación del portal es necesario seguir un proceso cuyas etapas consistirán en:

1. Especificación de requisitos
2. Análisis y diseño de la aplicación.
3. Detalles de Implementación.

2.2. Especificación de requisitos

2.2.1. Propósito

Con la realización de este apartado se desea explicar todas las diferentes funcionalidades y los requisitos de cada una de ellas, tanto a nivel general como específico, referentes a la aplicación de partes de trabajo.

Los requisitos de esta aplicación web son poder almacenar partes de trabajo específicos para una empresa de ingeniería y poder consultarlos para extraer información sobre dichos partes de trabajo, aunque en el futuro tendrá más utilidades.

2.2.2. Ámbito

La aplicación de gestión de partes de trabajo, que a partir de ahora llamaremos EAADMIN, tiene como objetivo facilitar a trabajadores que están en diferentes lugares del mundo guardar el trabajo realizado en cada momento del día para que la empresa tenga conocimiento. El desarrollo de esta aplicación tiene fines administrativos y de organización de empresa.

La finalidad de este trabajo académico, es que los trabajadores puedan por el momento insertar partes de trabajo y los administradores consultar las horas trabajadas. Más adelante, si todo funciona bien, las búsquedas serán mucho más elaboradas, incluirán costes, planes de trabajo y otras características.

2.2.3. Perspectiva del producto

Esta aplicación web por sus características puede recordar a los ERP ya que como se ha comentado, en un futuro incluirá funcionalidades como los planes de trabajo, costes, cálculo de costes y otras características.

Un ejemplo sencillo de entender podría ser un ERP que gestione facturas en una empresa, en la imagen 2.2.3_1 se muestra una interfaz factura, donde el trabajador introduce el producto y otros datos para generar la factura, y guardarla posteriormente en la base de datos. En nuestro caso la empresa ofrece servicios, es decir, envía a ingenieros a trabajar a otras empresas.

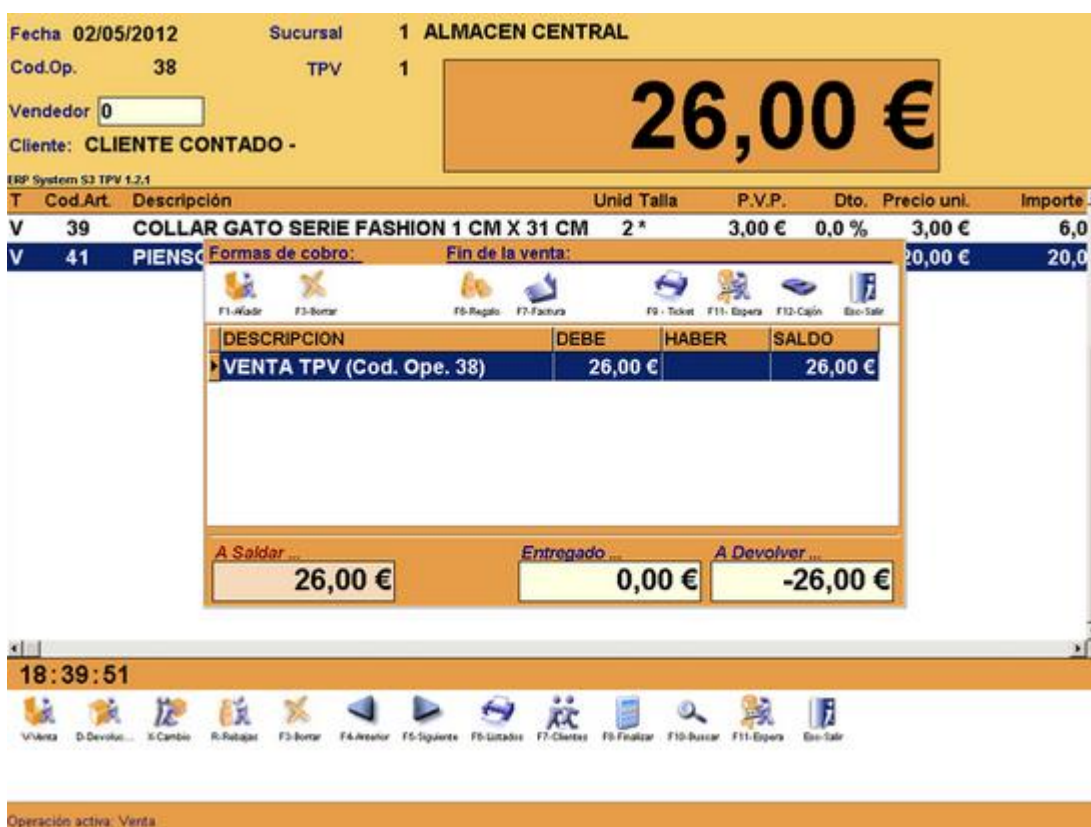


Imagen2.2.3_1. Ejemplo interfaz de un ERP

La diferencia es que el objetivo de esta aplicación es hacer algo concreto para una empresa concreta, y de ahí que se haya seguido un plan conforme a las necesidades de la empresa.

El servicio que ofrecerá como se ha comentado, será el de que los trabajadores puedan crear, modificar y consultar partes de trabajo desde cualquier lugar y de cualquier fecha, mientras que los administradores por el momento podrán consultar las horas trabajadas en un periodo de tiempo por este trabajador.

Además de esto, un aspecto muy importante de la web, es que debe permitir en función del tipo de usuario que seas, realizar unas tareas concretas u otras.

A continuación se indican las principales funcionalidades de la web y los diferentes tipos de usuarios que accederán a la página.

2.2.4. Funciones del producto

En este apartado nos centramos en distinguir las diferentes opciones que permitirá la página. Las funciones que se ofrecen se detallan a continuación:

- Visualización e interacción con la página de inicio de la aplicación: cuando un usuario quiera acceder a la web de EAADMIN se le mostrará una página inicial en la que en función de su rol podrá acceder a contenidos de administración o de trabajadores (manipulación de partes de trabajo).
- Registro: el registro se realiza únicamente desde administración. Ningún trabajador puede registrarse ni darse de baja, la administración se encarga de proporcionarles un usuario.
- Identificación: un usuario registrado podrá acceder a la aplicación mediante esta opción, proporcionando su nombre de usuario (DNI) y su contraseña.
- Generar informe: un usuario administrador podrá generar informes para saber las horas trabajadas en un periodo de tiempo por los trabajadores.
- Partes de trabajo: un usuario administrador podrá consultar y modificar partes de trabajo rellenos por los trabajadores. Un usuario trabajador

podrá crear, modificar y consultar partes de trabajo propios.

- Cerrar sesión: un usuario identificado podrá cerrar su sesión de usuario con esta opción, volviendo así a la página de inicio de la aplicación.

2.2.5. Características del usuario

Existen dos tipos de usuarios en el sistema: usuarios administradores y usuarios trabajadores. A continuación describimos las acciones que pueden realizar cada uno de ellos.

- Administradores: Trabajadores de la empresa en oficinas. Tienen la posibilidad de crear, modificar y borrar tanto trabajadores, como proyectos en los que trabaja la empresa, como partes de trabajo de los trabajadores.
- Trabajadores: Trabajadores de la empresa se mueven por diferentes países y por ello necesitan de una aplicación web. Pueden crear y modificar partes de trabajo.

2.2.6. Modelo vista controlador

El modelo–vista–controlador (MVC) es un patrón de arquitectura de software, que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello MVC propone la construcción de tres componentes

distintos que son el modelo, la vista y el controlador, es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario.^{1 2} Este patrón de arquitectura de software se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

La siguiente imagen muestra la típica colaboración entre componentes del MVC.

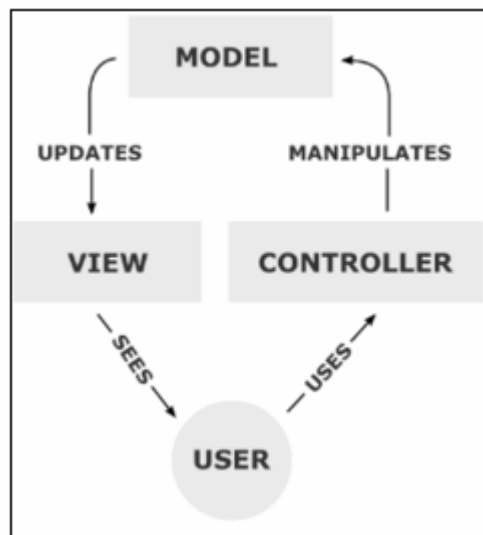


Imagen2.2.3_2. Colaboración MVC

De manera genérica, los componentes de MVC se podrían definir como sigue:

- **El Modelo**: Es la representación de la información con la cual el sistema opera, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones, implementando también los privilegios de acceso que se hayan descrito en las especificaciones de la aplicación. Envía a la 'vista' aquella parte de la información que en cada momento se le solicita para que sea mostrada.
- **El Controlador**: Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, editar un documento o un registro en una base de datos).
- **La Vista**: Presenta el 'modelo' (información y lógica de negocio) en un formato adecuado para interactuar (usualmente la interfaz de usuario) por tanto requiere de dicho 'modelo' la información que debe representar como salida.



2.3. Análisis y diseño

Una vez que ya tenemos definidos y clasificados todos los requisitos de nuestra aplicación, ahora es el turno de comenzar a analizar todos estos requisitos. Para ello, lo que vamos a necesitar es construir una serie de modelos que nos ayuden a ver poco a poco como será nuestra aplicación.

Utilizaremos el diagrama de casos de uso para mostrar como interactúa un actor con el sistema, el diagrama de clases UML para definir la estructura estática, definiremos las secuencias, capa de interfaz y capa de persistencia para las bases de datos, utilizando el modelo entidad-relación para definir la estructura de nuestra base de datos, por tanto los esquemas a seguir serán:

- Diagrama casos de uso
- Diagrama de clases
- Diagrama de secuencias
- Modelo entidad-relación para la capa de persistencia

2.3.1. Análisis de casos de uso

El diagrama de casos de uso representa la forma en cómo un actor (en nuestro caso un usuario de la web), interactúa con el sistema. En otras palabras, este diagrama lo que hace es mostrar la relación entre los actores y los casos de uso en la aplicación.

Para nuestra aplicación, los actores que van a intervenir son: usuarios administradores, y usuarios trabajadores.

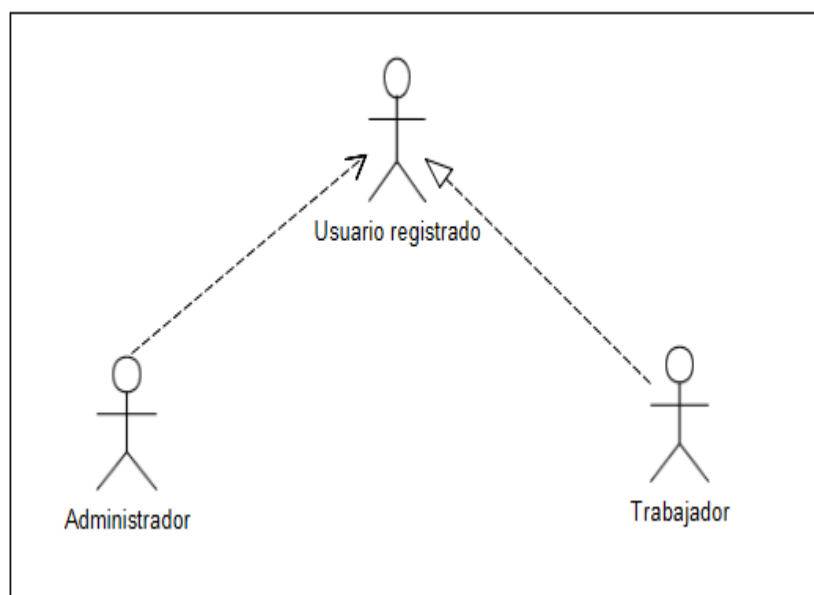


Imagen2.3.1_1. Usuarios de la aplicación

- **Usuario administrador:**

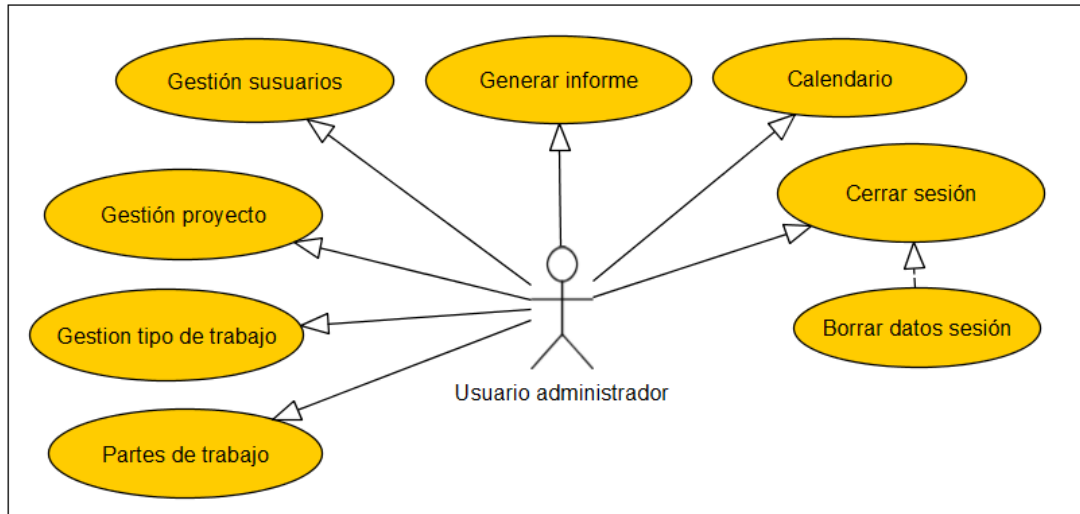


Imagen2.3.1_2. Usuario administrador

El usuario de tipo administrador tiene los casos de uso que se muestran en la imagen 2.3.1_2, cerrar sesión nos devolverá a la página de inicio y eliminara las variables de sesión, en el calendario podamos especificar días festivos, para que más adelante en el parte de trabajo las horas trabajadas esos días puedan contabilizarse como horas extras, y los otros casos e explican con más detalle a continuación.

- **Gestión de usuarios:**

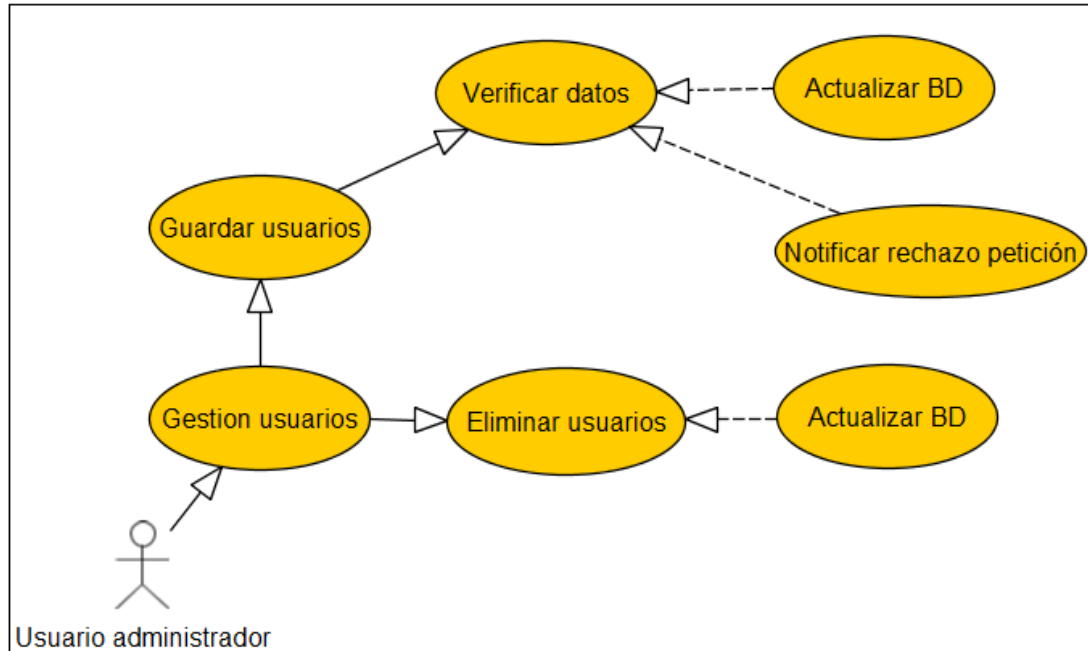


Imagen2.3.1_3. Gestión de usuarios del administrador

La gestión de usuarios por parte del administrador es la misma, que seguirán la gestión de proyectos y la gestión de trabajos, por ese motivo y por no redundar, solo incluiremos el caso de uso de la gestión de usuarios que explicaremos a continuación.

Como se puede ver en la imagen2.3.1_3, el administrador tiene dos opciones, guardar usuario o eliminar usuario, en el caso de guardar usuario, si el usuario ya existe nos preguntara si queremos modificarlo, y en el caso de que no exista si queremos crearlo, antes de preguntarnos si queremos cualquier acción, la aplicación se cerciora que los datos introducidos son correctos, y en caso de que no lo sean nos rechazara la petición de guardar.

En caso de querer eliminar algún registro, solo tendremos que seleccionarlo de la lista que nos aparecerá, y eliminarlo.

- **Gestión de partes del administrador:**

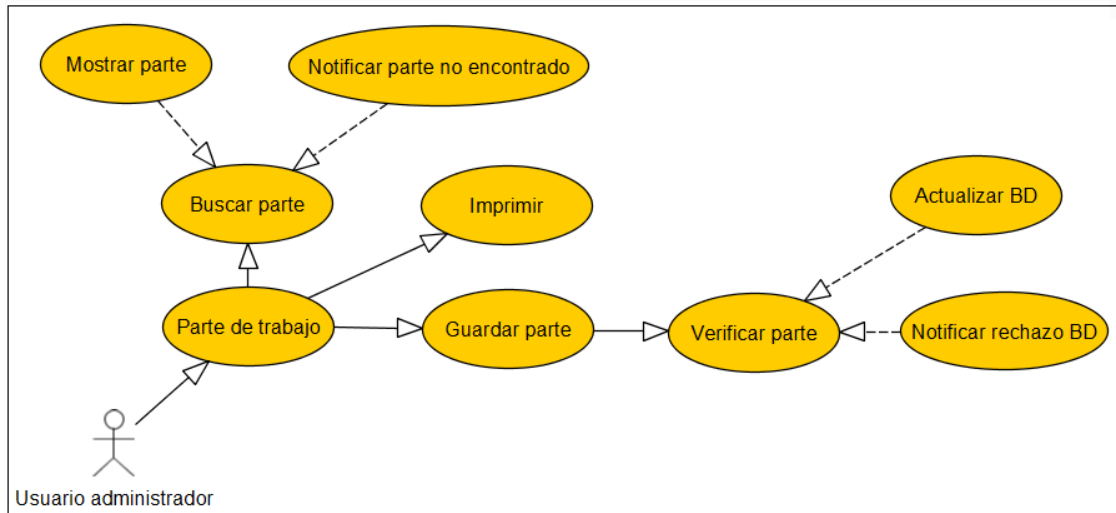


Imagen2.3.1_4. Gestión de partes del administrador

La gestión de partes de trabajo por parte del administrador es la misma que seguirán los trabajadores cuando accedan con su usuario trabajador, es decir, podrán imprimir, guardar o modificar parte, aunque no podrán crear un nuevo parte, como se muestra en la imagen2.3.1_4.

La principal diferencia con los trabajadores es que los administradores tendrán una lista con todos los trabajadores para poder filtrar por semana, año y nombre de trabajador que parte quieren visualizar, para en caso de que quieran modificarlo o imprimirlo.

- **Generar informe:**

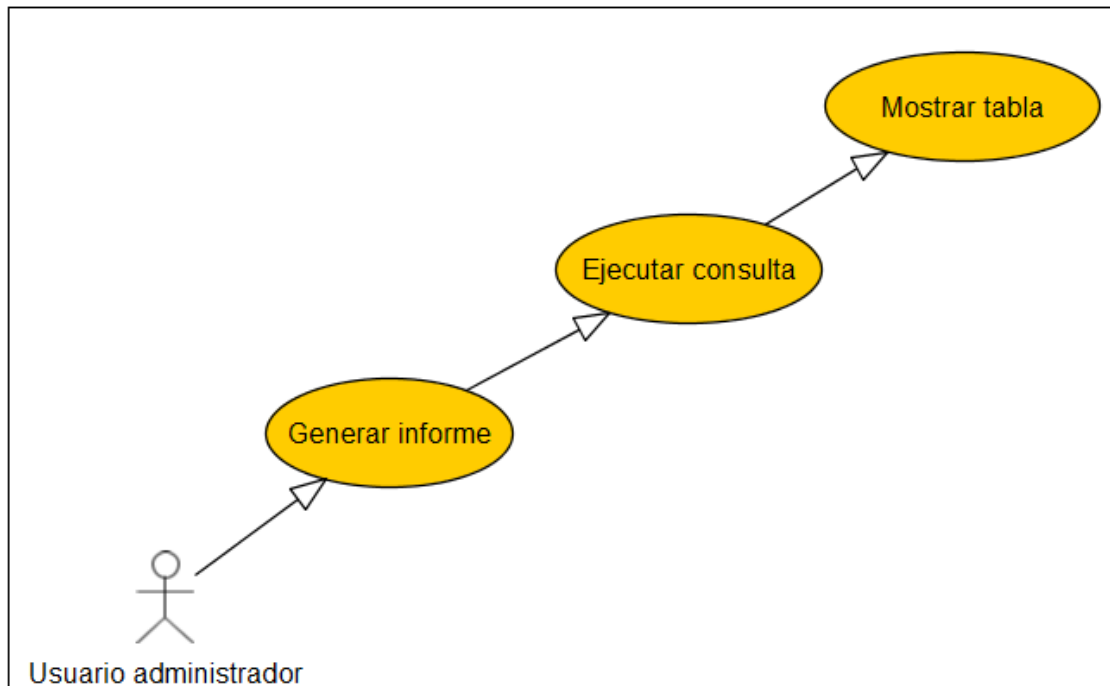


Imagen2.3.1_5. Generar informe

Generar un informe por parte del administrador, consistirá únicamente en, introducir parámetros de búsqueda, fecha de inicio, fecha de fin, nombre del trabajador, tipo de búsqueda, por parte (semanalmente) o por entrada de parte, un parte tiene 21 entradas, tres por cada día de la semana, por proyecto y por tipo de trabajo, estos serán los parámetros para generar un informe.

Tal como se muestra en la imagen2.3.1_5, una vez se accede a la sección de generar informe solo tendremos que ejecutar nuestra consulta con nuestros parámetros para poder visualizar, el informe deseado.

- **Partes de trabajo:**

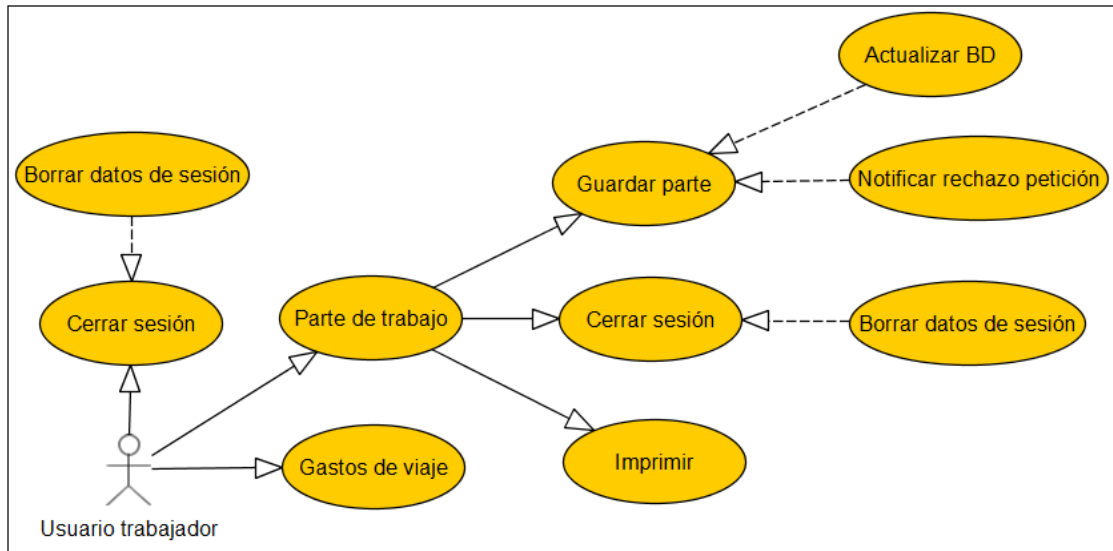


Imagen2.3.1_6. Partes de trabajo

Finalmente los casos de uso de los partes de trabajo de los trabajadores tal y como se muestran en la imagen2.3.1_6, son casi los mismos que en la gestión por parte de los administradores. Tal y como habíamos dicho la principal diferencia es que los trabajadores solo pueden ver sus partes de trabajo no los de ningún otro trabajador, y cuando inserten la información en caso de no existir, se creara un nuevo parte.

2.3.2. Diagrama de clases

Un diagrama de clases se utiliza para mostrar la estructura estática del sistema con el que se quiere trabajar, mostrando las clases y las relaciones entre ellas.

La representación de este diagrama es través de cajas rectangulares con un nombre y una serie de atributos que hacen referencia al nombre de esa caja. Estas cajas, se les denomina clases.

Una **clase** es la descripción de un grupo de objetos con estructura, comportamiento, y relaciones similares. Los atributos de estas clases pueden ser: **privados, protegidos, o públicos.**

Respecto a las **relaciones** entre cada una de las clases, existen varios tipos de relación entre ellas: **asociación, agregación, composición, especialización/generalización.**

A continuación en la imagen2.3.2_1 se muestra el diagrama de clases de EAADMIN:

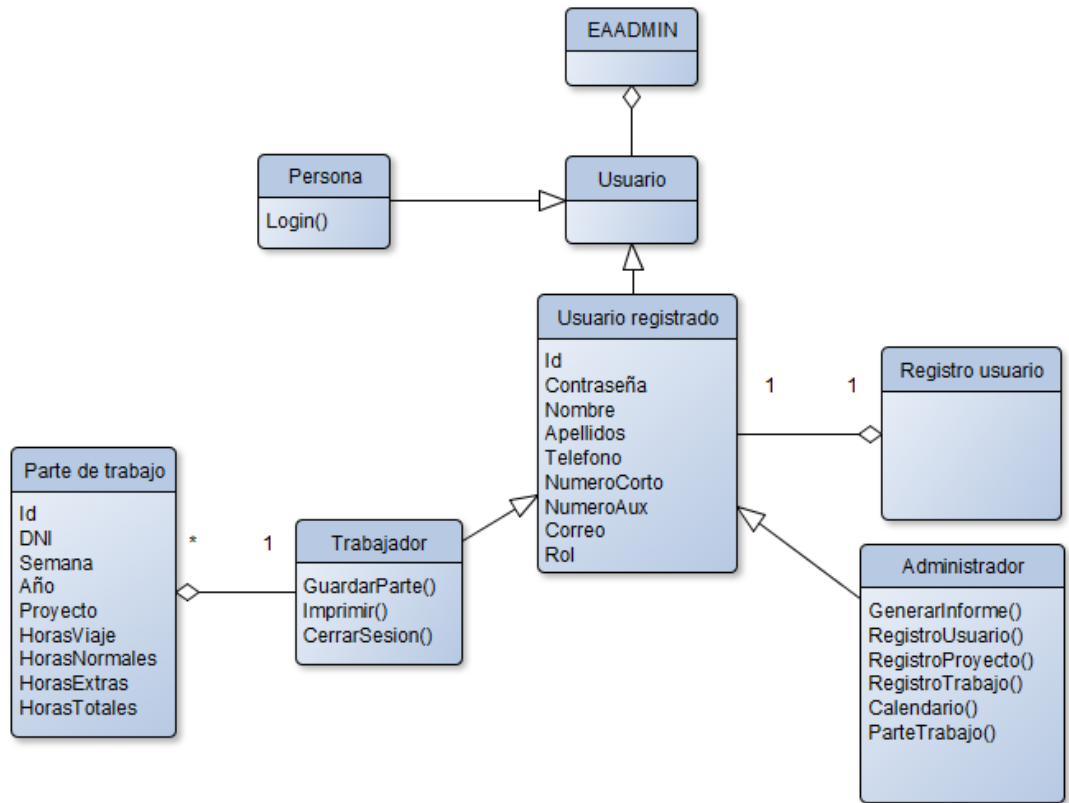


Imagen2.3.2_1. Diagrama clases EAADMIN

2.3.3. Diagramas de secuencia

Llegados a este punto, el objetivo es identificar la comunicación dentro del sistema, y las operaciones (métodos) de las clases. Para este propósito tenemos los diagramas de secuencia.

Partimos de los casos de uso, los cuales describen cómo los actores interactúan con el sistema, y a partir de esto, se identifican los eventos dirigidos hacia el sistema y, desde el sistema a los actores.

Los diagramas de secuencia muestran cada uno de estos eventos de una forma temporal. Cada objeto se representa con una línea vertical (su línea de vida), y cada mensaje con una flecha horizontal que va desde el objeto emisor hasta el receptor.

Se detallan a continuación los diagramas de secuencia que más peso tienen en la aplicación. Los escenarios de cada diagrama son estos:

- Autenticarse en el sistema
- Generar informes
- Interactuar con el parte de trabajo
- Guardar registros(Usuarios, proyectos o tipos de trabajo)
- Eliminar registros

- **Autenticarse en el sistema:**

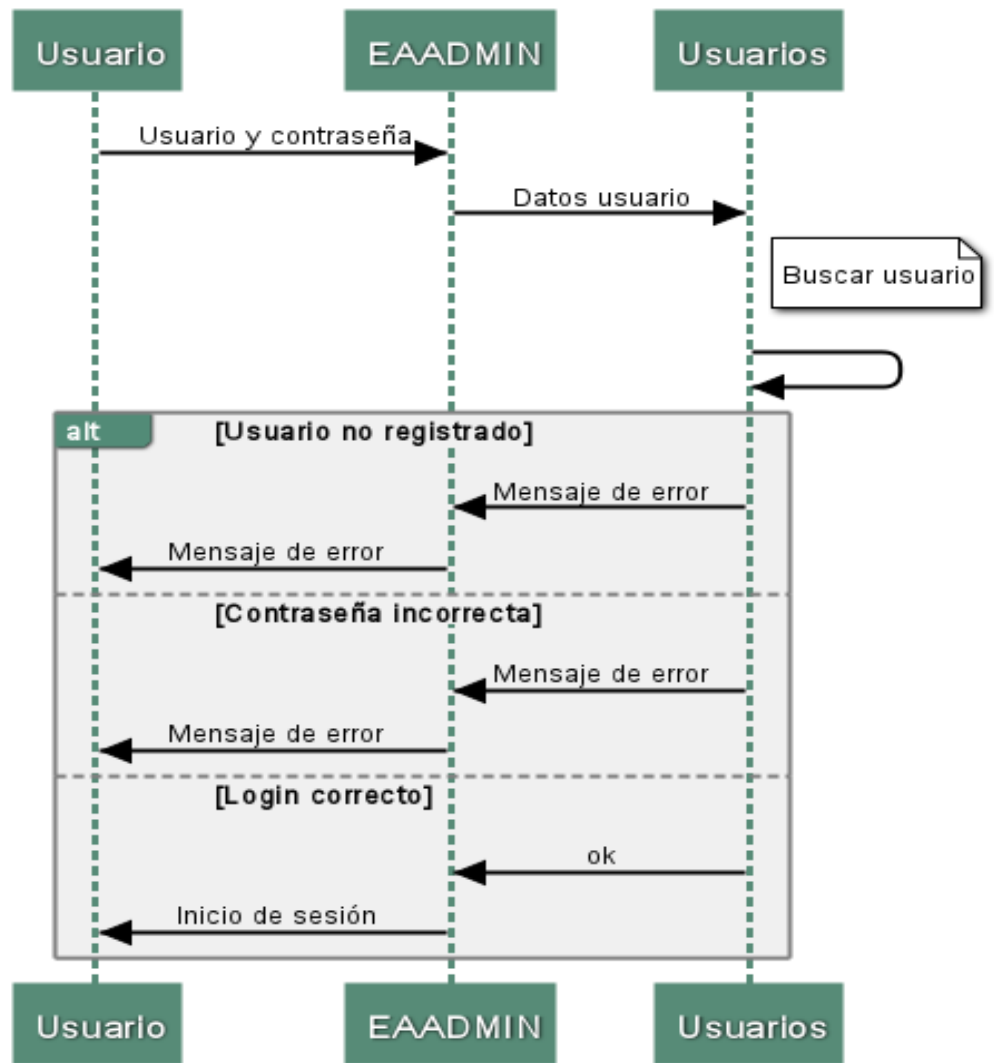


Imagen2.3.3_1. Autenticarse en el sistema

La primera opción que ofrece el portal es introducir nuestro usuario y nuestra contraseña, para poder acceder al sistema, como se observa en la imagen2.3.3_1, los usuarios introducirán sus datos y si están registrados en la base de datos y son correctos, iniciaran la sesión.

- **Generar informes:**

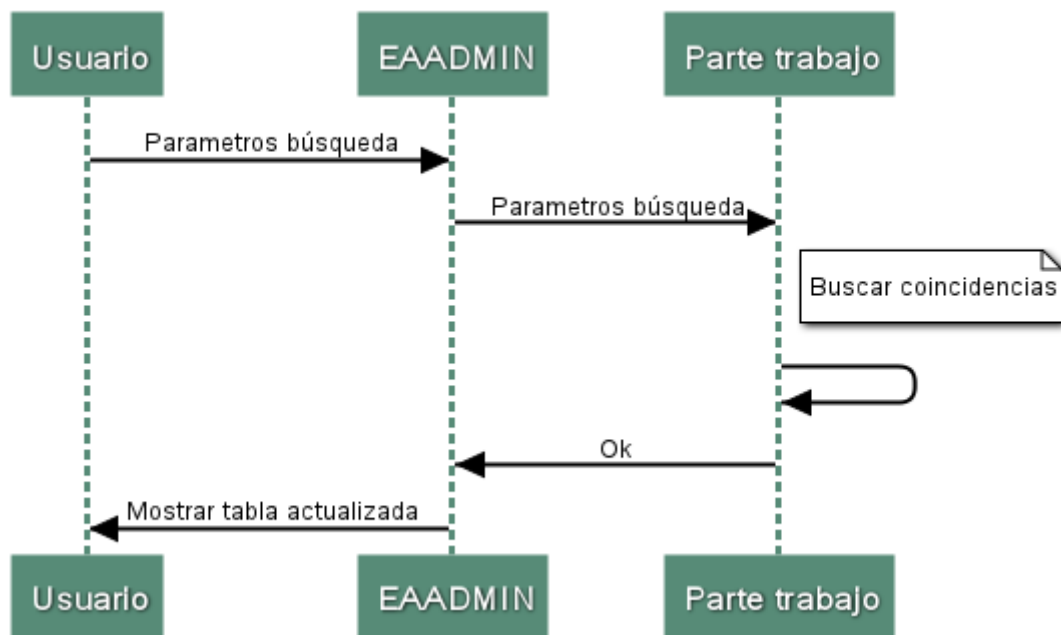


Imagen2.3.3_2. Generar informes

La principal acción de un administrador es poder realizar informes de los partes de trabajo, para ello tal y como se muestra en la imagen2.3.3_2, el usuario mediante unos parámetros de búsqueda realizara una consulta a la tabla de partes de trabajo y buscara coincidencias de dentro de la tabla, la respuesta será ofrecerle al usuario una tabla con los datos relacionados con su búsqueda, con la finalidad de obtener las horas trabajadas.

- **Interactuar con el parte de trabajo:**

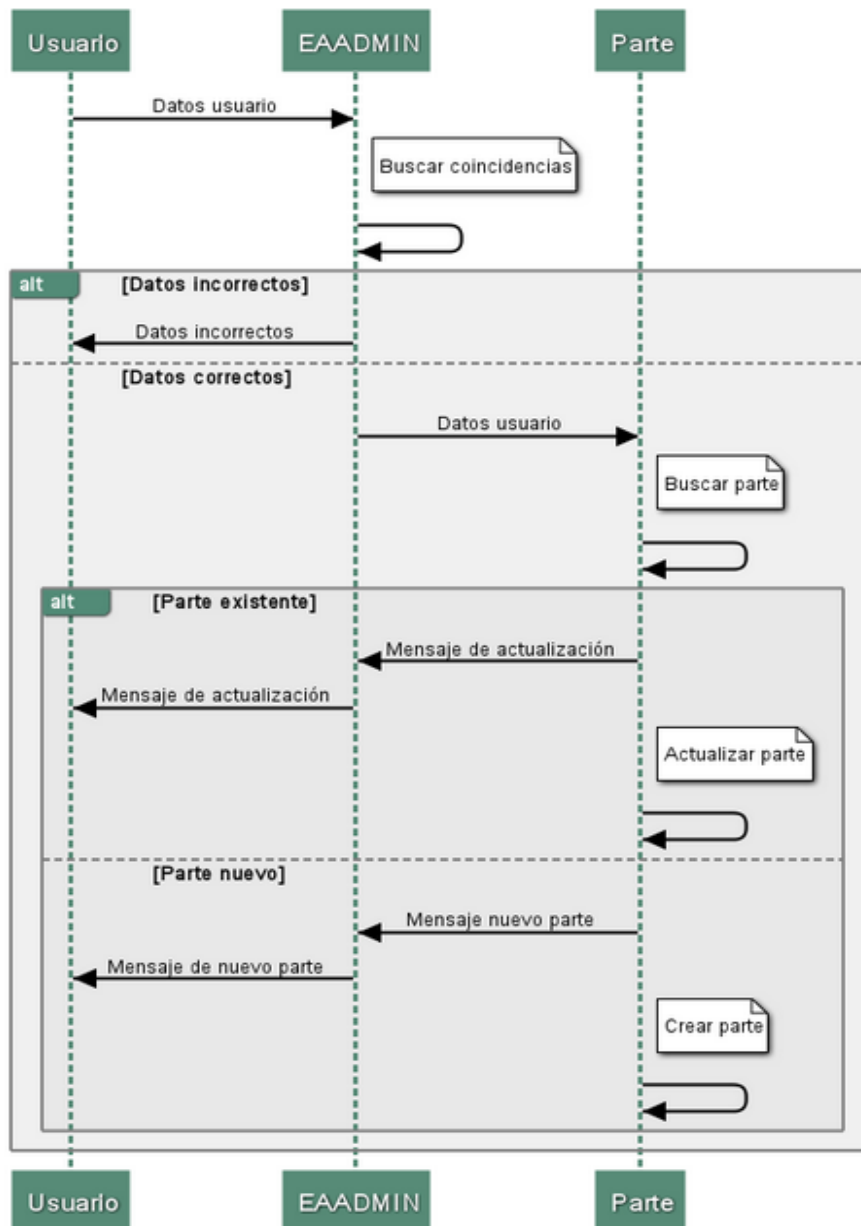


Imagen2.3.3_3. Interactuar con parte de trabajo

La principal acción de un trabajador es la de poder insertar partes de trabajo. Tal y como se muestra en la imagen2.3.3_3, cuando un usuario trate de guardar lo primero que se comprobara es que los datos estén correctos, y en caso de existir ya el parte lo modificara indicándoselo al usuario, en caso de ser un parte nuevo lo creara.

- Guardar registros:

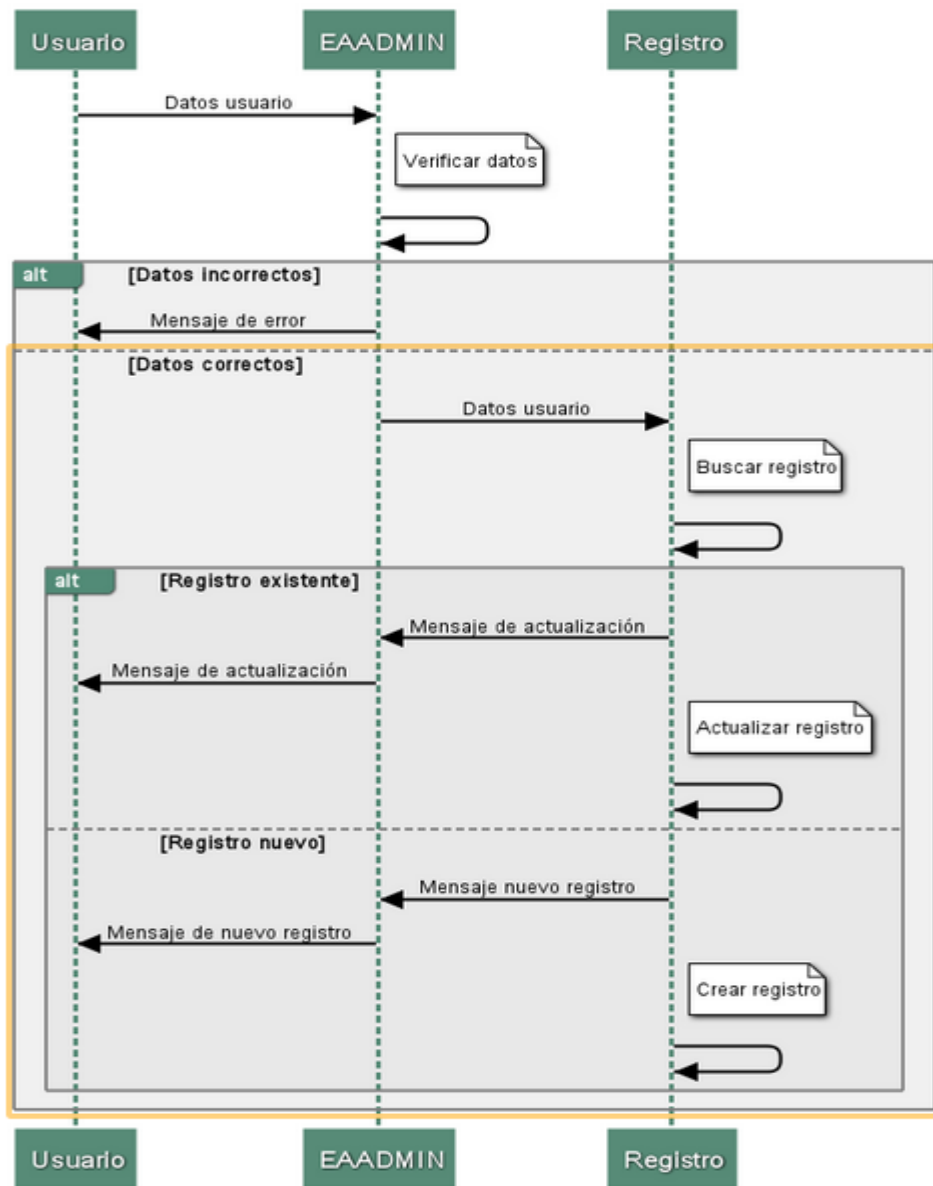


Imagen2.3.3_4. Guardar registros

Tal y como se muestra en la imagen2.3.3_4, cuando un usuario administrador trate de guardar algún registro primero verificara datos y en caso de que sean correctos las acciones a realizar serán las que se muestran en el marco amarillo, en caso de existir dicho registro preguntara si lo queremos modificar y en caso contrario si lo queremos crear.



- **Eliminar registros:**

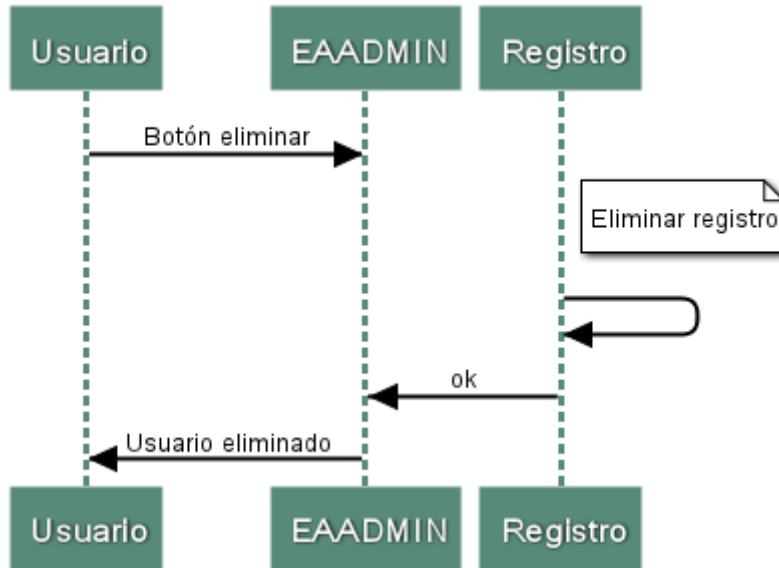


Imagen2.3.3_5. Eliminar registros

Finalmente para eliminar algún registro, tal y como muestra la imagen 2.3.3_5 se buscara en la tabla del registro, y eliminara el registro seleccionado por el usuario desde una lista de registros.

2.3.4. Capa de interfaz

Una vez comentados los requisitos funcionales, y detallados los diagramas de casos de uso, de clases, y de secuencias, ahora es el momento de describir del diseño de la interfaz web.

Lo que se desea conseguir en este paso es mostrar el diseño de nuestra aplicación mediante bocetos, áreas, etc. De esta forma conseguiremos que la perspectiva del producto sea más amplia y que todo lo que hemos estado describiendo hasta este momento se plasme de una manera más visual.

En las siguientes imágenes podemos ver cómo se van a estructurar los tres principales tipos de contenido que tendrá nuestra página web:

- **Interfaz inicio:**



Imagen2.3.4_1. Boceto interfaz inicio

Podemos observar en la imagen2.3.4_1, la típica entrada a un registro de una aplicación, logo de la empresa, campo para introducir el usuario, campo para introducir la contraseña y un botón para conectarse.

- **Interfaz administrador:**

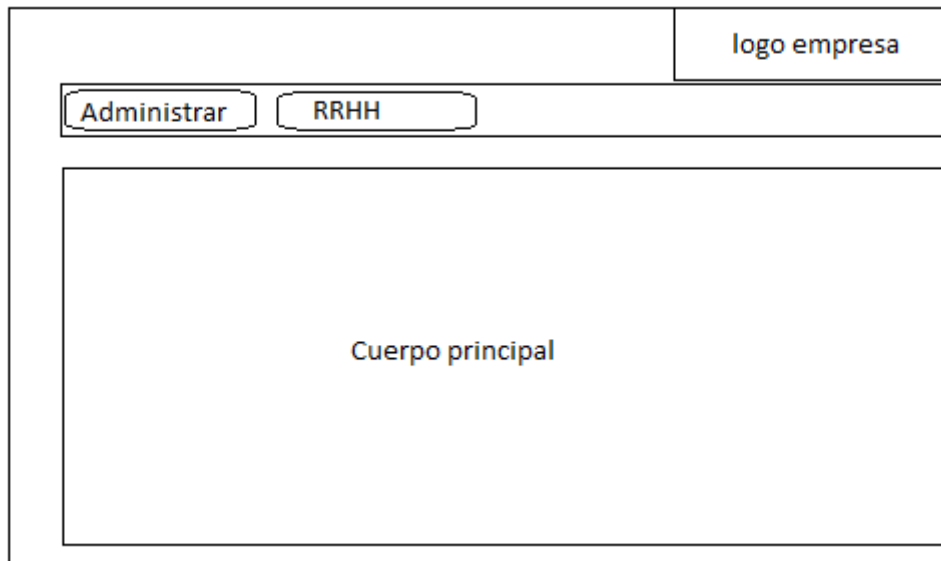


Imagen2.3.4_2. Boceto interfaz administrador

Después de conectarnos como administradores, nos aparecerá una imagen parecida a esta donde en el cuerpo principal se cargara algún tipo de imagen de bienvenida.

El botón de administrar, será un desplegable donde nos aparecerán las opciones para acceder a la interfaz de registros de usuarios, proyectos y trabajadores. También nos aparecerá la opción de cerrar sesión.

El botón de recursos humanos también será un desplegable, donde nos aparecerán las opciones de parte de trabajo, generar informe y calendario. En el parte de trabajo podremos buscar partes de trabajo de cualquier trabajador, imprimirlos y guardarlos, en generar informes podremos crear una tabla con ciertos parámetros de búsqueda para visualizar las horas trabajadas en esos parámetros, y finalmente en calendario introduciremos los días que queremos que sean festivos, para que el tipo de horas que se contabilicen sean como horas extras.

- **Interfaz parte de trabajo:**

Datos del parte				Logo empresa						
Cada uno de los siete días de la semana	Entrada	Salida	Descanso	Proyecto	Descripción	Observaciones	Viaje	Horas		
								Normal	Extra	Total

Imagen2.3.4_3. Boceto interfaz parte de trabajo

Después de conectarnos como trabajadores y acceder al parte de trabajo, nos cargara una interfaz parecida a la siguiente, donde en la parte superior izquierda nos cargara datos referentes al parte, como al año, la semana del año, nombre de usuario (este lo cargara automáticamente al entrar), ciudad, país, etc.

Como se puede observar en la imagen2.3.4_3, para cada día de la semana, nos aparecerán tres entradas diferentes por día, esto es porque en un solo día, se puede trabajar en diferentes proyectos o en diferentes trabajos.

También tendremos la opción de imprimir, cerrar sesión o guardar los datos.

2.3.5. Capa de persistencia

Este último punto describe la capa de persistencia, o también conocida como capa de base de datos.

En esta capa se encarga de almacenar los datos y los contenidos de toda la información que reside en nuestro portal web en tablas de una base de datos.

A la hora de pensar en cómo se va estructurar nuestra base de datos y todas sus tablas, es necesario pasar por una serie de procesos que van desde un modelo conceptual hasta lo que sería un modelo físico.

Para el diseño conceptual se utilizará el diagrama Entidad-Relación, el cual permite hacer una representación de entidades, relaciones y atributos.

Una **entidad** hace referencia a una “cosa” u “objeto” del mundo real, como por ejemplo, una persona, un puesto de trabajo, etc. Las **relaciones** describen la dependencia entre entidades y permiten una asociación entre las mismas. Y por último, los **atributos** son las características que definen o identifican a una identidad en concreto.

En la siguiente imagen, Figura 2.3.5_1, se muestra el diagrama formado por las entidades, relaciones y atributos de nuestra aplicación web:

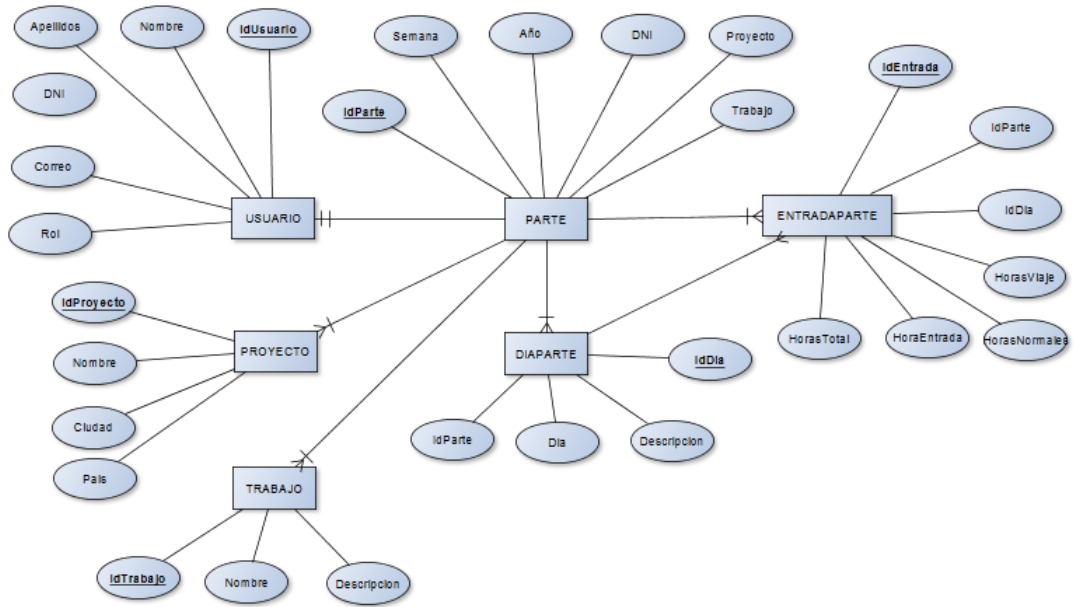


Imagen2.3.5_1. Diagrama entidad-relación EAADMIN

2.4. Detalles de implementación

2.4.1. Introducción

En los puntos anteriores de este apartado se ha ido viendo de forma paulatina toda la estructura y funcionamiento de nuestra aplicación, desde lo más básico hasta las partes más complejas que dan forma o lo que sería el resultado final de nuestra aplicación.

En este último punto del apartado de propuesta de aplicación web, solo nos quedaría por explicar las tecnologías y herramientas que se han usado para llevar a cabo todo el proceso de construcción de la página web.

A modo de resumen, las tecnologías y herramientas que se verán a continuación son las siguientes:

- HTML
- CSS
- JavaScript
- PHP
- MySQL
- phpDesigner
- JQuery
- Json
- Ajax

2.4.2. Tecnologías y herramientas utilizadas

- **HTML**

HTML es un lenguaje que se usa para programar y desarrollar páginas webs. Sus siglas corresponden a Hyper Text Markup Language, es decir, Lenguaje de Marcas de Hipertexto, que podría ser traducido como Lenguaje de Formato de Documentos para Hipertexto.

Se trata de un formato abierto que surgió a partir de las etiquetas SGML (Standard Generalized Markup Language). Concepto traducido generalmente como “Estándar de Lenguaje de Mercado Generalizado” y que se entiende como un sistema que permite ordenar y etiquetar diversos documentos dentro de una lista.

HTML se encarga de desarrollar una descripción sobre los contenidos que aparecen como textos y sobre su estructura, complementando dicho texto con diversos objetos (como fotografías, animaciones, etc.).

Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos. El texto en él, se crea a partir de etiquetas, también llamadas tags, que permiten interconectar diversos conceptos y formatos.

Para la escritura de este lenguaje, se crean etiquetas que aparecen especificadas a través de corchetes o paréntesis angulares: < y >.

A modo de ejemplo se muestra a continuación una imagen de un simple trozo de código HTML:



```
<!DOCTYPE html>
<html>
<!-- created 2010-01-01 -->
<head>
<title>sample</title>
</head>
<body>
<p>Voluptatem accusantium
totam rem aperiam.</p>
</body>
</html>
```

Imagen2.4.2_1. Ejemplo de código HTML

- **CSS**

CSS son las siglas de Cascading Style Sheets (Hojas de Estilo en Cascada), que es un lenguaje que describe la presentación de los documentos estructurados en hojas de estilo para diferentes métodos de interpretación, es decir, describe cómo se va a mostrar un documento en pantalla.

CSS es una especificación desarrollada por el W3C (World Wide Web Consortium) para permitir la separación de los contenidos de los documentos escritos en HTML, XML, XHTML, SVG, o XUL de la presentación del documento con las hojas de estilo, incluyendo elementos tales como los colores, fondos, márgenes, bordes, tipos de letra..., modificando la apariencia de una página web de una forma más sencilla, permitiendo a los desarrolladores controlar el estilo y formato de sus documentos.

Para que a una página html se le pueda dar un formato específico hay varias opciones de hacerlo.

La primera de ellas, y la que se ha utilizado, en general, para realizar todo este proyecto, es crear una hoja de estilo externa que se vinculará con el documento a través de la etiqueta `<link>`, dentro de la `<head>`.

Otra opción es poner directamente dentro del documento la etiqueta `<style>` en el elemento que se desee, y dentro de ella especificar las distintas configuraciones que se quieran crear. Esta opción no se ha usado en la creación de este proyecto.



- **JAVASCRIPT**

JavaScript (abreviado comúnmente "JS") es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

En el proyecto, JS se ha utilizado para mostrar las alertas o avisos que cada van sucediendo a cada uno de los usuarios. Más adelante se detallarán ejemplos de JS.

Para poder trabajar con JS, o bien se crea un archivo .js y se indexa dentro del código del html (igual que se hacía con los archivos .css externos), o directamente se puede insertar con la etiqueta <script>.

Se puede ver en la siguiente imagen un trozo de código escrito en lenguaje JavaScript:

```
<HTML>
<TITLE>Ejemplo03.htm</TITLE>

<SCRIPT LANGUAGE="JavaScript">
//Recoger un dato por teclado y visualizarlo
var nom;
nom=prompt("Escribe tu nombre", "NOMBRE");
alert("Mucho gusto "+ nom);

</SCRIPT>

</HTML>
```

Imagen2.4.2_2. Ejemplo de código JAVASCRIPT

- **PHP**

PHP es un lenguaje de programación de uso general de código del lado del servidor, originalmente diseñado para el desarrollo web de contenido dinámico.

Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.

Uno de los detalles que son importantes sobre este lenguaje es que su código no es visible en los navegadores. El servidor lo ejecuta y envía los resultados, pero el cliente no puede ver cómo está creado el código.

En el siguiente apartado se analizarán con mucho más detalle cada uno de los segmentos de código PHP que son relevantes para nuestra aplicación EAADMIN.

A continuación un pequeño ejemplo de código PHP:

```
<html>
<head></head>
<body>
<?php
    echo "Hola Mundo";
?>
</body>
</html>
```

Imagen2.4.2_3. Ejemplo de código PHP

- **MySQL**

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y rendimiento.

Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha.

Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales contar con un alto grado de estabilidad y un rápido desarrollo.

Gracias a la interfaz que nos proporciona, se han podido organizar todas las tablas y toda su información de una manera fácil y sobre todo segura.

The screenshot shows the phpMyAdmin interface with a table structure overview. The interface includes a navigation menu on the left with options like 'Home', 'Database', 'Table', 'Structure', 'SQL', 'Search', 'Query', 'Export', 'Import', 'Operations', 'Routines', 'Events', and 'Triggers'. The main area displays a table structure overview for the database 'eaadmin_com'. The table structure is as follows:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> CALENDARIO	Browse Structure Search Insert Empty Drop	20	InnoDB	latin1_general_ci	16 KiB	-
<input type="checkbox"/> COSTES	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8_general_ci	48 KiB	-
<input type="checkbox"/> DESC_PROYECTOS	Browse Structure Search Insert Empty Drop	10	InnoDB	utf8_general_ci	48 KiB	-
<input type="checkbox"/> DIAPARTE	Browse Structure Search Insert Empty Drop	4,893	InnoDB	utf8_general_ci	368 KiB	-
<input type="checkbox"/> ENTRADAPARTE	Browse Structure Search Insert Empty Drop	14,676	InnoDB	utf8_general_ci	1.5 MiB	-
<input type="checkbox"/> ENTRADA_PLANNING	Browse Structure Search Insert Empty Drop	208	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> GRUPO_TRABAJO	Browse Structure Search Insert Empty Drop	31	InnoDB	utf8_general_ci	16 KiB	-
<input type="checkbox"/> PARTE	Browse Structure Search Insert Empty Drop	699	InnoDB	utf8_general_ci	112 KiB	-
<input type="checkbox"/> PROYECTOS	Browse Structure Search Insert Empty Drop	25	InnoDB	utf8_general_ci	64 KiB	-
<input type="checkbox"/> USUARIOS	Browse Structure Search Insert Empty Drop	34	InnoDB	utf8_general_ci	16 KiB	-
10 tables	Sum	20,596	MyISAM	latin1_spanish_ci	2.2 MiB	0 B

Imagen2.4.2_4. Ejemplo de phpMyAdmin

- **phpDesigner**

PHPDesigner es un editor de texto para programadores de PHP, CSS, HTML, JAVASCRIPT, ETC. El programa cuenta con la posibilidad de autocompletar a medida que escribimos el código, y colorear el texto para que sea más fácil la identificación de las líneas.

También dispone de muchas más funcionalidades, la más importante para mí es la posibilidad de conectarse por ftp, y poder programar directamente sobre nuestro servidor.



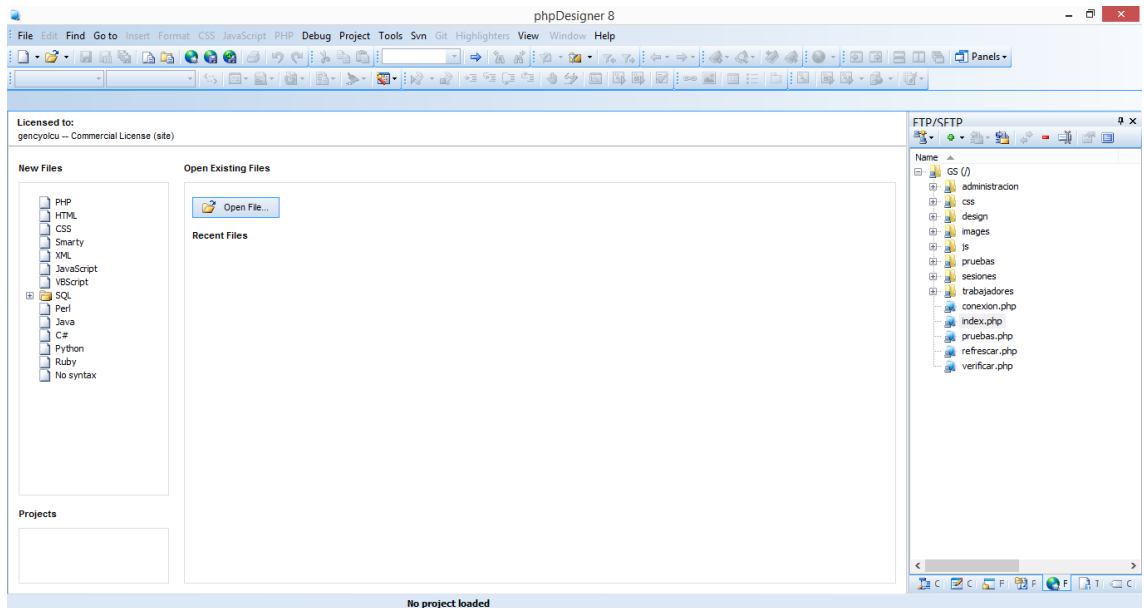


Imagen2.4.2_5. Ejemplo de phpDesigner

- **JQuery**

JQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privados.² jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

En la imagen 2.4.2_6 se muestra un pequeño fragmento de código jQuery utilizado en nuestra aplicación.


```
$(document).ready(function(){
$('#busquedaUsuario').on('change', cargarDatosUsuario);
$("#guardarUsuario").on('click', checkGuardarUsuario);
$("#eliminarUsuario").on('click', checkEliminarUsuario);
$('#admUsuarioForm').on('submit', modificarUsuario);
});
```

Imagen2.4.2_6. Ejemplo de jQuery

- **Json**

JSON, acrónimo de JavaScript Object Notation, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

En la imagen 2.4.2_7 se muestra un pequeño ejemplo de un objeto jSon.

```
- materias: [
  - {
    id: "1",
    nombre: "Dispositivos Moviles"
  },
  - {
    id: "2",
    nombre: "Computo Integrado"
  },
  - {
    id: "3",
    nombre: "Sistemas Distribuidos y paralelos"
  },
  - {
    id: "4",
    nombre: "Modelado y Simulado de sistemas Dinamicos"
  }
]
```

Imagen2.4.2_7. Ejemplo de jSon

- **Ajax**

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor.

La imagen2.4.2_8 muestra un pequeño ejemplo de una petición Ajax al servidor.

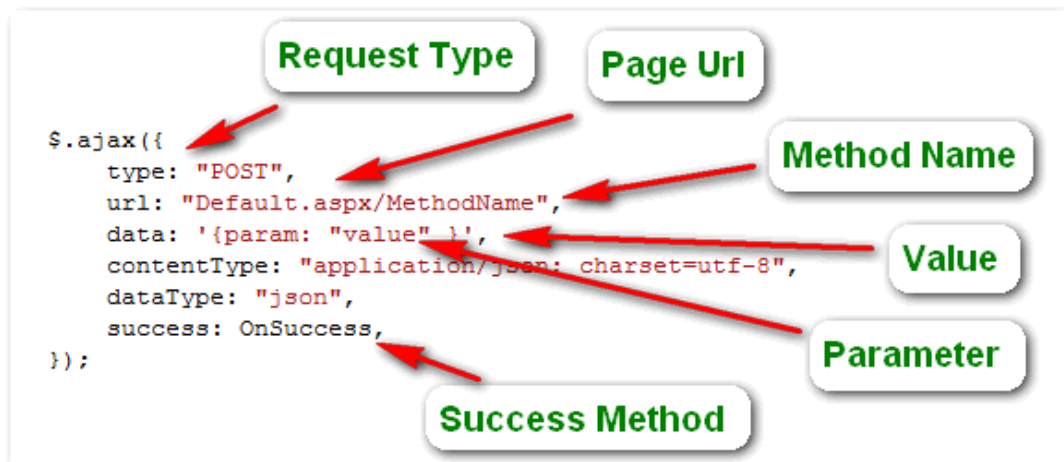


Imagen2.4.2_8. Ejemplo de petición Ajax

2.4.3. Interfaz y funcionalidades

Sin lugar a dudas, la tarea que más tiempo me ha llevado ha sido realizar la interfaz gráfica de la aplicación, ya que se partía desde cero y se ha tratado de no utilizar plantillas.

Como ya se ha mencionado en puntos anteriores, primero que nada se hicieron los borradores, sobre como tenía que quedar aproximadamente el diseño web y que necesitaríamos.

Una vez bien definidos todos los prototipos de diseño, se empezó a elaborar el primer diseño con código HTML y CSS de la página web.

A continuación se mostraran imágenes sobre las interfaces más relevantes, sobre las que ya hemos hablado en puntos anteriores, pero esta vez, se mostrara el diseño final y se comentara brevemente alguna parte importante del código.

- **Página de inicio**

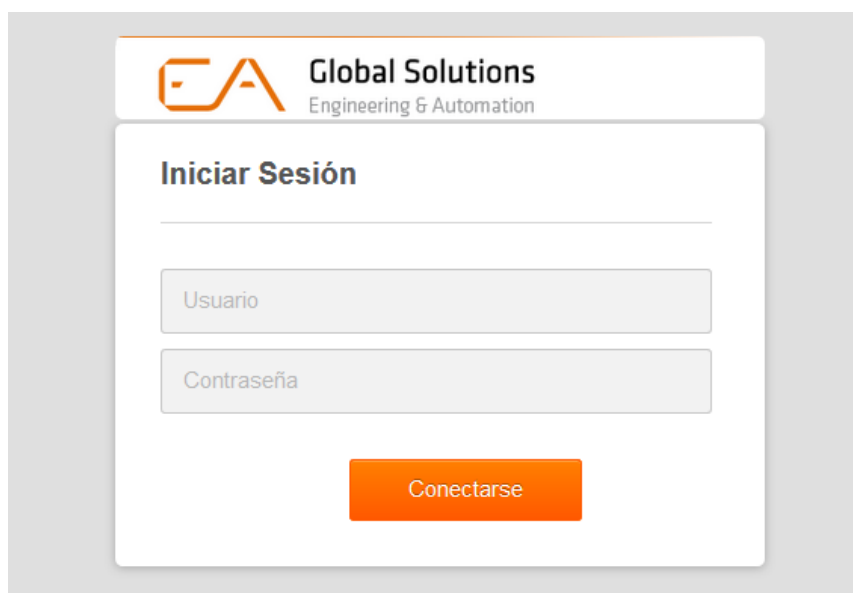


Imagen2.4.3_1. Página de inicio

En el momento de entra introduciremos nuestro usuario y nuestra contraseña, y después de darle al botón conectarse, si los datos son correctos, entraremos dentro de la aplicación o bien como administradores o bien como trabajadores, dependiendo de nuestro rol.

```
//obtenemos el número de columnas que devuelve la consulta
$Registros=mysql_num_rows($Result);
// Si devuelve mas de 0 esta en la base de datos
if($Registros){

    $Fila=mysql_fetch_row($Result);
    $NombreCompleto = $Fila[2].' '.$Fila[3];
    $_SESSION['Login']=$Log;
    $_SESSION['Rol']=$Fila[8];
    $Result2=mysql_query("Select Lider, Proyecto from GRUPO_TRABAJO where NombreCompleto='".$NombreCompleto."'",$link);
    $filaLider=mysql_fetch_row($Result2);
    $esLider = $filaLider[0];
    $idProyectoGrupo = $filaLider[1];
    //vamos al menu correspondiente
    if($Fila[8] == 'Admin'){
        header ('Location: ./administracion/administracion.php');
    }
    else{
        if($esLider){
            $_SESSION['lider']=true;
            $_SESSION['idProyecto'] = $idProyectoGrupo;
            header ('Location: ./trabajadores/trabajadorL.php');
        }
        else{
            header ('Location: ./trabajadores/trabajador.php');
        }
    }
}
}
```

Imagen2.4.3_2. Código de verificación de usuario

Aquí simplemente comprobaremos si el usuario existe, y en caso de existir, comprobaremos también si su contraseña es correcta. Una vez realizada la comprobación crearemos una sesión de usuario en PHP y su variable de sesión (sin ella no se puede acceder al contenido de la web) y dependiendo de su rol direccinaremos hacia la parte de administrador o hacia la parte de trabajador.

- **Página de registro (Usuarios)**



Imagen2.4.3_3. Interfaz registro de usuarios

Primero que nada comentar que las secciones de proyectos y descripción/trabajos funcionan igual que la de usuarios, de modo que solo explicaremos un tipo de registro como hemos hecho a lo largo de la memoria, y como siempre será el registro de usuarios.

Una vez conectados como administradores desde el botón desplegable Administrar y seleccionando la opción Usuarios se nos mostraran los campos para introducir un nuevo usuario.

Si queremos modificar o eliminar un usuario existente desde el desplegable de búsqueda que nos aparece, seleccionaremos el usuario que deseamos modificar, y nos cargara automáticamente toda la información del usuario en los campos, una vez realizado este paso podremos eliminarlo con el botón eliminar o modificarlo con el botón guardar.

Si deseamos introducir un nuevo usuario rellenáramos el formulario y, si todos los campos son correctos introduciremos un nuevo usuario en nuestra base de datos.

En primer lugar explicaremos como hacemos la validación del formulario, mediante jQuery.

Utilizaremos un plugin llamado jQuery validation, donde llamaremos a la librería validation y en nuestro código javascript introduciremos en un json la información que hace referencia a las condiciones que queremos aplicar a nuestro formulario para que sea válido, tal y como se muestra en la siguiente imagen.

```
$("#admUsuarioForm").validate({
  rules: {
    idUsuario: {
      required: true,
      dniCheck: true
    },
    Nombre: {
      required: true,
      minlength: 2
    },
    Apellidos: {
      required: true,
      minlength: 2
    },
    Password: {
      required: true,
      minlength: 4
    },
    Password2: {
      required: true,
      minlength: 4,
      equalTo: "#Password"
    },
    Correo: {
      required: true,
      email: true
    },
    Telefono: {
      required: true,
      digits: true,
      minlength: 9,
    },
  },
});
```

Imagen2.4.3_4. Código de validación

Como se puede deducir, en el json el nombre de la norma indica el nombre de nuestro campo en el html, mientras que las condiciones vienen definidas dentro de la clase.

Mientras que la siguiente imagen nos muestra la función PHP donde se realizaran las operaciones de guardado o eliminado realizadas en la petición mediante Ajax y que responderá con un json.

```
function guardarUsuario($datos, $link){
    $jsondata = array();
    //Check if user exists

    $Id = strtoupper($datos['Id']);
    $Nombre = $datos['Nombre'];
    $Apellidos = $datos['Apellidos'];
    $Password = $datos['Password'];
    $Correo = $datos['Correo'];
    $Telefono = $datos['Telefono'];
    $NumeroCorto = $datos['NumeroCorto'];
    $NumeroAuxiliar = $datos['NumeroAuxiliar'];
    $Rol = $datos['Rol'];

    $ObtenerId = "SELECT Id FROM USUARIOS WHERE Id = '$Id'";
    $usuario=$link ->query($ObtenerId);
    if ($usuario->num_rows > 0) {
        $actualizaUsuario = "UPDATE USUARIOS SET
            Id = '$Id',
            Password = '$Password',
            Nombre = '$Nombre',
            Apellidos = '$Apellidos',
            Telefono = '$Telefono',
            NumeroCorto = '$NumeroCorto',
            NumeroAuxiliar = '$NumeroAuxiliar',
            Correo = '$Correo',
            Rol = '$Rol'
            WHERE Id = '$Id'";

        $jsondata['error'] = 'False';
    }
}
```

Imagen2.4.3_5. Código de guardar usuario (actualización)

Esta función recibe por post la información de la petición javascript en este caso los campos del usuario, si el usuario ya está registrado lo actualizara y en el siguiente fragmento de código, que es la continuación de esta misma función si no existe el usuario, lo creara.

```
$jsondata['error'] = 'False';

if (!$link ->query($actualizaUsuario)) {
    $jsondata['error'] = mysqli_error($link);
}
}
else{
    $insertarUsuario = "INSERT INTO `USUARIOS`
    [(`Id`,`Password`,`Nombre`,`Apellidos`,`Telefono`,`NumeroCorto`,`NumeroAuxiliar`,`Correo`,`Rol` )
    VALUES
    (`$Id`,`$Password`,`$Nombre`,`$Apellidos`,`$Telefono`,`$NumeroCorto`,`$NumeroAuxiliar`,`$Correo`,`$Rol`)";

    $jsondata['error'] = 'False';

    if (!$link ->query($insertarUsuario)) {
        $jsondata['error'] = mysqli_error($link);
    }
}

mysqli_close ($link);
header('Content-type: application/json; charset=utf-8');
echo json_encode($jsondata, JSON_FORCE_OBJECT);
}
```

Imagen2.4.3_6. Código de guardar usuario (inserción)

En caso de no existir el usuario, la consulta sql ejecuta será un insert y no un update, de modo que creara un nuevo usuario con los datos introducidos y nos devolverá un json llamado \$jsondata con la información sobre cómo ha procedido esta solicitud, si ha habido algún error o todo se ha realizado correctamente.

- **Generar informe**

Para generar un informe accederemos a la sección como administrador de generar informe, y una vez dentro seleccionaremos los parámetros de búsqueda en el informe y ejecutaremos la consulta.

La siguiente imagen muestra la interfaz que nos encontraremos al pulsar el boto de recursos humanos y acceder a generar informes.



Imagen2.4.3_7. Interfaz de generar informe

La siguiente imagen muestra un informe generado entre las fechas “01/04/2016” y “30/04/2016”, de todos los usuarios de la base de datos, por entrada, que estén trabajando en el proyecto de VDL L560 Jaguar y de todos los tipos de trabajo.

La finalidad de estas búsquedas es principalmente y por el momento poder ver la suma de las horas totales de viaje, normales, extras y el total de las horas, aparte de poder ver en detalle la información paginada en una tabla creada con jquery con un plugin llamado dataTable jquery.

Día	Proyecto	Descrip/Trabajos	Horas Viaje	Horas normales	Horas extras	Horas en total	
2016-04-01	150017	16	0	8	1.5	9.5	
2016-04-02	150017	16	0	0	9.5	9.5	
2016-04-03	150017	16	0	0	7.5	7.5	
2016-04-04	150017	16	0	0	9	9	
2016-04-05	150017	16	0	8	1	9	
2016-04-06	150017	16	0	8	1	9	
2016-04-07	150017	16	0	8	1	9	
2016-04-08	150017	16	0	8	3	11	
2016-04-09	150017	16	0	0	10	10	
2016-04-10	150017	16	0	0	7.5	7.5	
Total horas de Viaje			Total horas normales		Total horas extras		Total horas en total
16			136		185		337
Total horas de Viaje			Total horas normales		Total horas extras		Total horas en total

Imagen2.4.3_8. Interfaz de un informe generado

Por otro lado el siguiente código muestra la información en un json del tipo de las características de la tabla que queremos crear con este plugin.

```
table = $('#tablaInf').DataTable({
  "language": {
    "lengthMenu": "",
    "zeroRecords": "No se encontro nada",
    "info": "Mostrando pagina _PAGE_ de _PAGES_",
    "infoEmpty": "",
    "infoFiltered": "(filtered from _MAX_ total records)",
    "search": "Buscar:",
    "paginate": {
      "previous": "Anterior",
      "next": "Siguiente"
    }
  }
});
```

Imagen2.4.3_9. Código json tabla.

Finalmente en la siguiente imagen veremos la función PHP que es llamada desde javascript, para obtener el json de la consulta realizada en la interfaz donde generamos el informe.

```
$consulta = "Select date(HoraEntrada), Proyecto, Descripcion, HorasViaje, HorasNormales, HorasExtras, HorasTotal
FROM ENTRADAPARTE where Proyecto not like ' ' $subUsuario $subProyecto $subDescripcion AND
HoraEntrada between $HoraInicio AND $HoraFinal";
$result=$link ->query($consulta);
$jsondata["tipo"] = "entrada";
$jsondata["DNI"] = $usuarioInf;
if($result){
  if($result->num_rows > 0){
    $jsondata["resultado"] = $result->num_rows;
    while($row = $result->fetch_assoc()){
      $jsondata["datos"][] = $row;
    }
  }
  else{
    $jsondata["resultado"] = "ninguno";
    $jsondata["datos"] = false;
  }
}
else{
  $jsondata["datos"] = false;
  $jsondata["resultado"] = "pepa";
}
}
mysqli_close ($link);
header('Content-type: application/json; charset=utf-8');
echo json_encode($jsondata, JSON_FORCE_OBJECT);
}
```

Imagen2.4.3_10. Código PHP generar informe.

Simplemente hace la consulta adecuada dependiendo de los parámetros de búsqueda, nos crea un objeto json llamado \$jsondata con la información de dicha consulta, y nos la devuelve para ser utilizada posteriormente por el cliente en la parte de javascript.

- **Calendario**

Para acceder al calendario deberemos desplegar el botón de recursos humanos y seleccionar la opción del calendario.

La finalidad del calendario es poder marcar los días que la empresa crea conveniente que sean festivos para que esos días la información introducida por los trabajadores en lo que a horas de trabajo respecta, cuenten como horas extras.

En la siguiente imagen podemos ver la interfaz del calendario con el día cuatro de abril marcado como festivo, haciendo click sobre el día del que se desea que sea festivo. En caso de volver hacer click el día ese día dejara de ser festivo.

Por supuesto, cuando se carga la interfaz en pantalla del calendario, antes se hace una consulta a la base de datos para saber qué días están marcados como festivos, y poder mostrar al usuario el calendario correctamente.

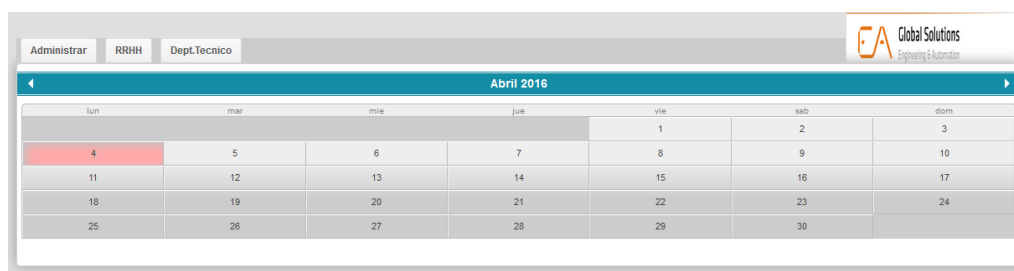


Imagen2.4.3_11. Interfaz de calendario

Para crear la interfaz del calendario se ha utilizado un plugin llamado jQuery event calendar.

En la siguiente imagen se observa un fragmento de código que llama al recibir el evento click sobre un día, a la función para guardar o eliminar el día como festivo en la base de datos.

```
flags.wrap.on('click', '.eventsCalendar-day a', function(e) {
    e.preventDefault();
    var year = flags.wrap.attr('data-current-year'),
        month = flags.wrap.attr('data-current-month'),
        day = $(this).parent().attr('rel');
    objectThis = $(this).parent();
    // GUARDAR/ELIMINAR DIA
    mes = parseInt(month) + 1;
    fechaDia = year+'-'+mes+'-'+day;
    $.post("./calendario/calendario.php",
        {
            llamada: 'guardarDia',
            dia: fechaDia
        },
        function(data, status){
            if(data.deleteSave == 'save'){
                console.log('guardado');
                $(objectThis).removeClass().addClass("eventsCalendar-day roig");
            }
            else{
                console.log('eliminado');
                $(objectThis).removeClass().addClass("eventsCalendar-day");
            }
        }
    );
});
```

Imagen2.4.3_12. Código de modificar día festivo

Finalmente el servidor recibe el día que se desea modificar comprueba si ese día está marcado en la base de datos, y en caso de estar marcado lo elimina para que deje de ser festivo, y en caso contrario lo añade para que sea festivo, tal y como se puede observar en la siguiente imagen.

```

function guardarDia( $dia, $link ) {
    $jsondata = array();

    $insertFecha = "INSERT INTO `CALENDARIO`
    (`Fecha`)
    VALUES
    ('$dia')";

    $jsondata['deleteSave'] = 'save';

    if (!$link ->query($insertFecha)) {
        $delFecha = "DELETE FROM CALENDARIO
        WHERE Fecha = '$dia'";
        $jsondata['deleteSave'] = 'delete';
        if (!$link ->query($delFecha)) {
            $jsondata['deleteSave'] = 'No se a podido ni guardar ni eliminar';
        }
    }

    mysqli_close ($link);
    header('Content-type: application/json; charset=utf-8');
    echo json_encode($jsondata, JSON_FORCE_OBJECT);
}

```

Imagen2.4.3_13. Código PHP del calendario

- **Parte de trabajo(Administrador)**

Para acceder al parte de trabajo desde administración deberemos desplegar las opciones desde el botón de recursos humanos y seleccionar la opción de parte de trabajo.

La finalidad de esta sección en administración es poder consultar cualquier parte de cualquier trabajador e incluso poder imprimirlo o modificarlo.

En la siguiente imagen se muestra un parte buscado de mí mismo en la empresa.

Desarrollo de una aplicación Web para la gestión de partes de trabajo en una empresa de ingeniería

Administrar		RRHH	Dept.Tecnico								
Julio Martí Borrás		1	2016	Buscar parte Guardar parte Imprimir							
Nombre: Julio Martí Borrás Cliente: VALIANT Proyecto: Ford (Valiant) Factoria: Ford Valencia Semana: 1 Ciudad: Valencia Año: 2016 Pais: España											
Fecha	Hora entrada	Hora Salida	Descanso	Proyecto	Descripcion	Observaciones	Viaje	Horas			
								Normales	Extras	Totales	
Lunes	04/01/2016	00:00	13:00	00:00	Ford (Valant)	PROGRAMACIÓN	---	0	8	5	13
					Elegir proyecto	Sin descripcion					
					Elegir proyecto	Sin descripcion					
Martes	05/01/2016				Elegir proyecto	Sin Descripcion	---	0	0	0	0
					Elegir proyecto	Sin descripcion					
					Elegir proyecto	Sin descripcion					
Miercoles	06/01/2016				Elegir proyecto	Sin Descripcion	---	0	0	0	0
					Elegir proyecto	Sin descripcion					
					Elegir proyecto	Sin descripcion					
Jueves	07/01/2016				Elegir proyecto	Sin Descripcion	---	0	0	0	0
					Elegir proyecto	Sin descripcion					
					Elegir proyecto	Sin descripcion					
Viernes	08/01/2016				Elegir proyecto	Sin Descripcion	---	0	0	0	0
					Elegir proyecto	Sin descripcion					
					Elegir proyecto	Sin descripcion					
Sabado	09/01/2016				Elegir proyecto	Sin Descripcion	---	0	0	0	0
					Elegir proyecto	Sin descripcion					
					Elegir proyecto	Sin descripcion					
Domingo	10/01/2016				Elegir proyecto	Sin Descripcion	---	0	0	0	0
					Elegir proyecto	Sin descripcion					
					Elegir proyecto	Sin descripcion					
Total							0	8	5	13	

Imagen2.4.3_14. Interfaz parte de trabajo (administración)

Tenemos tres opciones de búsqueda, el nombre del trabajador, la semana del año que va desde 1 al 52, y el año. Seleccionando los parámetros deseados de búsqueda y pulsando sobre el botón buscar parte podremos cargar y visualizar cualquier parte guardado.

El código y más funcionalidades del mismo vendrán explicadas en el siguiente apartado, ya que son las mismas para trabajadores y administradores.

- **Parte de trabajo(Trabajador)**

Para acceder al parte de trabajo desde un usuario con el rol de trabajador, lo primero será una vez autenticados apretar sobre el botón parte de trabajo del menú que se muestra en la siguiente imagen.

Cabe destacar que gastos de viaje es una sección en construcción que en principio no pertenece a esta memoria.

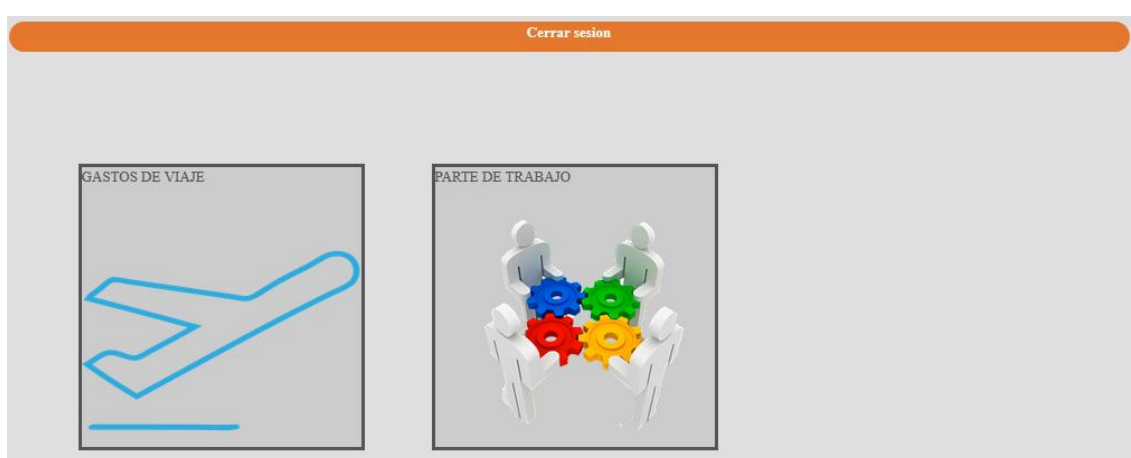


Imagen2.4.3_15. Interfaz del menú de trabajador

Una vez accedemos al parte de trabajo, nos cargara la interfaz de parte de trabajo, que es casi la misma que el parte de trabajo de administración, con la salvedad que aquí ya nos aparece nuestro nombre por defecto.

La siguiente imagen muestra el parte tal y como se carga desde la sección del rol de trabajadores.

Desarrollo de una aplicación Web para la gestión de partes de trabajo en una empresa de ingeniería


Nombre: Julio Martí Borrás										
Cliente:										
Proyecto:										
Factoria:		Semana: 25								
Ciudad:		Año: 2016								
Pais:										
Fecha	Hora entrada	Hora Salida	Descanso	Proyecto	Descripcion	Observaciones	Viaje	Horas		
								Normales	Extras	Totales
Lunes	20/06/2016			Elegir proyecto	Sin Descripción	...				
				Elegir proyecto	Sin descripción					
				Elegir proyecto	Sin descripción					
Martes	21/06/2016			Elegir proyecto	Sin Descripción	...				
				Elegir proyecto	Sin descripción					
				Elegir proyecto	Sin descripción					
Miercoles	22/06/2016			Elegir proyecto	Sin Descripción	...				
				Elegir proyecto	Sin descripción					
				Elegir proyecto	Sin descripción					
Jueves	23/06/2016			Elegir proyecto	Sin Descripción	...				
				Elegir proyecto	Sin descripción					
				Elegir proyecto	Sin descripción					
Viernes	24/06/2016			Elegir proyecto	Sin Descripción	...				
				Elegir proyecto	Sin descripción					
				Elegir proyecto	Sin descripción					
Sabado	25/06/2016			Elegir proyecto	Sin Descripción	...				
				Elegir proyecto	Sin descripción					
				Elegir proyecto	Sin descripción					
Domingo	26/06/2016			Elegir proyecto	Sin Descripción	...				
				Elegir proyecto	Sin descripción					
				Elegir proyecto	Sin descripción					
							Total			

Imagen2.4.3_16. Interfaz del parte de trabajo (trabajador)

Bien, finalmente explicaremos con más detalles las funcionalidades del parte de trabajo.

Como ya hemos comentado el nombre se carga por defecto y por supuesto no podemos modificarlo. Cabe destacar que, tanto semana como año son una lista desplegable de semanas del 1 al 52 y desde el año 2015 hasta los próximos años venideros. Cada vez que seleccionamos algún año o semana, se ejecuta internamente una función Ajax que buscara en la base de datos, esa semana y ese año de ese trabajador y, en caso de encontrarlo cargara los datos y los mostrara en pantalla.

La siguiente imagen muestra el código de activación javascript.

```

$('#semana').on('change', actualizarParte);
$('#ano').on('change', actualizarParte);
function actualizarParte () {
    buscarParte ();
    cambiar ();
}

```

Imagen2.4.3_17. Código de actualizar parte

Con jQuery cada vez que las etiquetas con el nombre de semana y año cambien, se llamara a la función actualizarParte(). Esta a su vez llamara a dos funciones, la primera buscarParte(), buscara si existe un parte de ese trabajador que coincida con la semana y el año.

Después llama a la función cambiar(), que comprueba si buscarParte() ha encontrado alguna coincidencia y en tal caso actualiza el parte, en caso contrario no hará nada más que cambiar la semana o el año.

Cuando insertemos horas de entrada, salida o descanso, deberá ser con el formato, hh:mm, y por supuesto en caso de haber algún dato incongruente como que la hora de entrada es posterior a la de salida, o directamente algún dato mal escrito, no se podrá validar el parte y nos mostrara con un recuadro rojo el parte está mal rellenado, y no nos permite que se guarde o se modifique.

A continuación explicaremos que los datos de la parte superior izquierda se cargan automáticamente al seleccionar el proyecto en el que estamos trabajando, ya que dicho proyecto tiene atributos tales como el país, la ciudad, el cliente o la factoría y por tanto estos datos se rellenan automáticamente, en caso de que seleccionemos diferentes proyectos en un mismo parte con diferentes ciudades, países o cualquier otra propiedad, simplemente aparecerá la palabra, varios. Por supuesto no se puede escribir manualmente sobre estos elementos, ya que están automatizados.

Lo siguiente son los tres botones de la parte inferior derecha, el botón de imprimir llama a una función que selecciona el div sobre el que está el parte para imprimirlo, tal y como muestra la siguiente imagen.

```
$('#imprimirParte').click(function(){
    $('#formularioEntero').printElement();
});
```

Imagen2.4.3_18. Código de imprimir parte



Guardar el parte de trabajo implica tres cosas, en primer lugar si el parte no está validado, es decir si todos los datos introducidos no son correctos, nos mostrara un mensaje de error y nos marcara en rojo donde están los fallos en el parte.

En segundo lugar, si el parte es válido comprobara si ese parte ya existe en la base de datos, es decir, si ese trabajador ya ha insertado esa semana de ese año un parte. En caso afirmativo, la aplicación nos mostrara un mensaje diciéndonos que el parte ya existe, si deseamos actualizarlo, en caso afirmativo, el cliente ordenara al servidor que actualice el parte de trabajo. En el caso de que el parte no existiera antes, nos preguntara si deseamos crear un nuevo parte, y en caso afirmativo, creara un nuevo registro sobre el parte que acabamos de crear con la información introducida.

2.4.4. Implementación de la base de datos

Para finalizar el capítulo 2, se va a detallar el modo en el que los datos con los trabaja la aplicación, es decir, datos de usuarios, partes de trabajo, búsquedas, etc., se almacenan en la base de datos.

Para procesar y almacenar los datos se ha trabajado con phpMyAdmin. Esta aplicación nos permite interactuar con una base de datos de una forma sencilla y desde una interfaz web.

El primer paso para poder comunicar la aplicación web con la base de datos es realizar una función que permita conectar con la base de datos.

```
<?php
//conexion con el servidor
$link=mysql_connect("eaadmin.com.mysql", "eaadmin_com", "*****");
mysql_query("SET NAMES 'utf8'", $link);
mysql_query("SET CHARACTER SET 'utf8'", $link);
if(!$link) echo "Error: no se a podido conectar a la base de datos.";
//seleccion de la base de datos
$db=mysql_select_db('eaadmin_com', $link);
?>
```

Imagen2.4.4_1. Código de conexión

Una vez realizada esta conexión, la aplicación web puede comunicarse con la base de datos.

A continuación se muestra en phpMyAdmin cada una de las tablas utilizadas en la base de datos para nuestro proyecto y su estructura.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	Id			No	None	AUTO_INCREMENT	Change Drop Browse distinct values Primary Unique Index More
<input type="checkbox"/>	2	Eliminado			No	None		Change Drop Browse distinct values Primary Unique Index More
<input type="checkbox"/>	3	Descripcion	utf8_spanish_ci		No	None		Change Drop Browse distinct values Primary Unique Index More

Imagen2.4.4_2. Tabla de descripción

Desarrollo de una aplicación Web para la gestión de partes de trabajo en una empresa de ingeniería

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	Id			No	None	AUTO_INCREMENT	
<input type="checkbox"/>	2	IdParte			No	None		
<input type="checkbox"/>	3	IdDia			No	None		
<input type="checkbox"/>	4	DNI	utf8_spanish_ci		No	None		
<input type="checkbox"/>	5	HoraEntrada			No	None		
<input type="checkbox"/>	6	HoraSalida			Yes	NULL		
<input type="checkbox"/>	7	HoraDescanso			Yes	NULL		
<input type="checkbox"/>	8	Proyecto	utf8_spanish_ci		Yes	NULL		
<input type="checkbox"/>	9	Descripcion	utf8_spanish_ci		Yes	NULL		
<input type="checkbox"/>	10	HorasViaje			No	None		
<input type="checkbox"/>	11	HorasNormales			No	None		
<input type="checkbox"/>	12	HorasExtras			No	None		
<input type="checkbox"/>	13	HorasTotal			No	None		

Imagen2.4.4_3. Tabla de entrada de parte

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	Id			No	None	AUTO_INCREMENT	
<input type="checkbox"/>	2	DNI	utf8_spanish_ci		No	None		
<input type="checkbox"/>	3	Validado			No	None		
<input type="checkbox"/>	4	Semana	utf8_spanish_ci		No	None		
<input type="checkbox"/>	5	Ano	utf8_spanish_ci		No	None		
<input type="checkbox"/>	6	Proyecto	utf8_spanish_ci		No	None		
<input type="checkbox"/>	7	TotalViaje			No	None		
<input type="checkbox"/>	8	TotalNormales			No	None		
<input type="checkbox"/>	9	TotalExtras			No	None		
<input type="checkbox"/>	10	TotalTotales			No	None		

Imagen2.4.4_4. Tabla parte

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	Id			No	None	AUTO_INCREMENT	
<input type="checkbox"/>	2	Proyecto	utf8_spanish_ci		No	None		
<input type="checkbox"/>	3	Factoria	utf8_spanish_ci		No	None		
<input type="checkbox"/>	4	Ciudad	utf8_spanish_ci		No	None		
<input type="checkbox"/>	5	Linea	utf8_spanish_ci		No	None		
<input type="checkbox"/>	6	Pais	utf8_spanish_ci		No	None		
<input type="checkbox"/>	7	Cliente	utf8_spanish_ci		No	None		
<input type="checkbox"/>	8	Eliminado			No	None		

Imagen2.4.4_5. Tabla de proyectos

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
<input type="checkbox"/>	1	Id	utf8_spanish_ci		No	None		
<input type="checkbox"/>	2	Password	utf8_spanish_ci		No	None		
<input type="checkbox"/>	3	Nombre	utf8_spanish_ci		No	None		
<input type="checkbox"/>	4	Apellidos	utf8_spanish_ci		No	None		
<input type="checkbox"/>	5	Telefono	utf8_spanish_ci		No	None		
<input type="checkbox"/>	6	NumeroCorto	utf8_spanish_ci		No	None		
<input type="checkbox"/>	7	NumeroAuxiliar	utf8_spanish_ci		No	None		
<input type="checkbox"/>	8	Correo	utf8_spanish_ci		No	None		
<input type="checkbox"/>	9	Rol	utf8_spanish_ci		No	None		
<input type="checkbox"/>	10	Eliminado			No	None		

Imagen2.4.4_6. Tabla de usuarios

3. Evaluación

Una vez concluida la parte de implementación y diseño del portal web, se comienza una última fase para verificar que cada función de la web se realiza de forma correcta.

Para poder llevar a cabo este proceso, se pretende, realizar una serie de pruebas con la aplicación en distintos ámbitos, lo que sería un análisis del sitio web. En los siguientes puntos se va detallando cada una de estas verificaciones.

3.1.1. Análisis del sitio web

Para analizar nuestra aplicación web nos centraremos en tres pruebas, validación de CSS, verificación de enlaces rotos y el más importante la velocidad de la aplicación.

- **Validación de CSS**

Para nuestro sitio web se han utilizado hojas de estilo para dar forma y estructura. Se necesita que una vez hechas se compruebe que todas las propiedades utilizadas son las correctas, ya que de lo contrario podrían darse problemas de visualización en algunos navegadores.

Para realizar esta prueba hemos utilizado la herramienta <http://jigsaw.w3.org> . En la imagen 3.1.1_1 se pueden ver los resultados obtenidos:

Resultados del Validador CSS del W3C para <http://www.eadmin.com> (CSS versión 3)

¡Enhorabuena! No error encontrado.

¡Este documento es [CSS versión 3](#) válido!

Puede mostrar este icono en cualquier página que valide para que los usuarios vean que se ha preocupado por crear una página Web interoperable. A continuación se encuentra el XHTML que puede usar para añadir el icono a su página Web:

Imagen3.1.1_1. Validación CSS

- **Enlaces rotos**

La segunda prueba a la que sometemos al portal web es simplemente para comprobar si todos los enlaces o vínculos que tiene el sitio web son funcionales y no hay referencias erróneas.

Se utiliza la herramienta: www.anybrowser.com/linkchecker.html

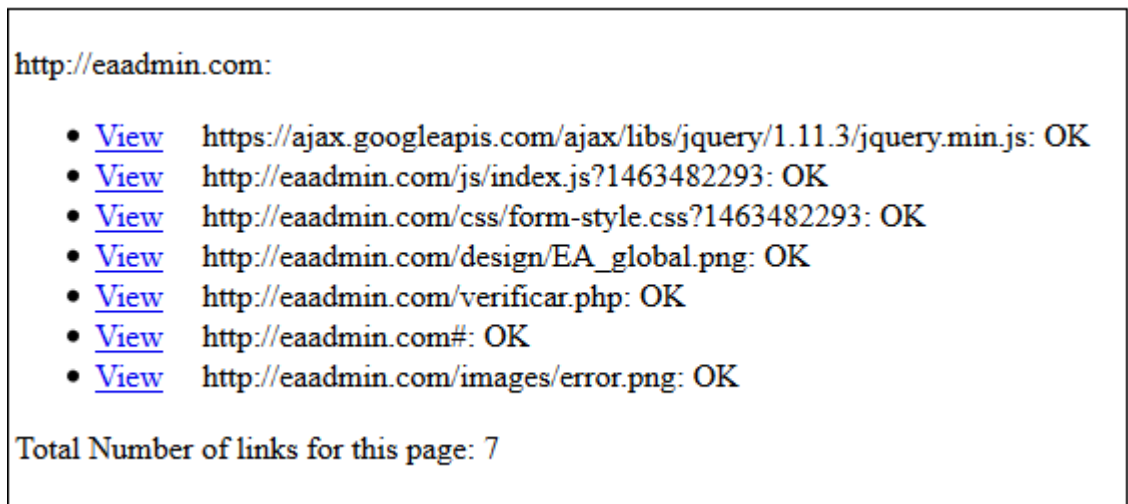


Imagen3.1.1_2. Validación enlaces rotos

- **Velocidad de aplicación**

Y lo mejor para el final, PageSpeed Insights es una de las herramientas que Google pone a disposición de los desarrolladores. Además de brindar reportes detallados, ofrece consejos útiles con los cambios que deberíamos realizar para hacer nuestras páginas mucho más veloces.

<https://developers.google.com/speed/pagespeed/insights/>

Móvil Ordenador

88 / 100 Resumen de sugerencias

! Elementos que debes corregir:

Eliminar el JavaScript que bloquea la visualización y el CSS del contenido de la mitad superior de la página
▶ [Mostrar cómo corregirlo](#)

! Elementos que puedes plantearte corregir:

Especificar caché de navegador
▶ [Mostrar cómo corregirlo](#)

Optimizar imágenes
▶ [Mostrar cómo corregirlo](#)

✓ 7 reglas aprobadas
▶ [Mostrar detalles](#)

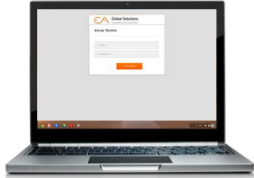


Imagen3.1.1_3. Velocidad de la web

Como no podía ser de otra manera, saca una gran puntuación en velocidad, muy por encima de webs como Hotmail, marca, y muchas otras webs conocidas.

Sin utilizar ningún tipo de framework me he encargado de hacer los balanceos de carga, separar el modelo vista controlador y tratando que por encima de una web sea una aplicación, de manera que el javascript se ocupa prácticamente de todo el funcionamiento de la web, y solo accede al servidor para consultar la base de datos, haciendo la web dinámica y que funcione como rápida.

El único pero, es que es mejorable el 88/100 y que no se ha tenido en cuenta la experiencia en el móvil, cosas que quedan para mejorar en el futuro.

4. Conclusiones

Después de terminar este proyecto he sacado muchas conclusiones, a nivel personal ha sido duro y bastante sacrificado los cuatro años de carrera, me llevo conmigo grandes recuerdos, logros personales de cosas que jamás creí que podría lograr, grandes alegrías no por aprobar exámenes, si no por entenderlos y saber porque había aprobado, una sensación que no se puede experimentar con asignaturas teóricas, y finalmente me llevo amigos y una gran experiencia personal y enriquecedora que me servirá seguro para el resto de mi vida.

A este proyecto le he dedicado muchos meses, aprovechando que trabajaba en la empresa donde lo hice, muchas horas rompiéndome la cabeza para tratar de averiguar cómo hacer ciertas cosas, pero al final, lo que me queda es saber que desde hace más de seis meses esta aplicación está siendo utilizada sin ninguna queja, y que la empresa no solo está contenta con ella, si no que quiere que continúe ampliándola y mejorándola, lo cual me deja aún más contento.

Entre estas mejoras, queda pendiente hacer que la experiencia desde móvil o tablets sea igual de buena que desde el PC, también se añadirán funcionalidades como generar planes de trabajo, poder introducir costes y generar informes de esos costes, mejorar el apartado visual de la web o generar informes más detallados y completos. También algunos apartados que están en construcción como gastos de viaje.

Como apunte final, estoy contento con el trabajo realizado, he aprendido mucho, pero sé que aún quedan me quedan muchísimas cosas por aprender y mucho que mejorar, y eso me motiva aún más para continuar esta carrera, pero ahora en mi vida profesional.

5. Bibliografía

- **Félix Buendía**

“Una guía para la realización y supervisión de proyectos finales de carrera en el ámbito de la web”.

Editorial UPV, 2008.

- **Apuntes de asignaturas**

Ingeniera de software, bases de datos, diseño y gestión de bases de datos y tecnologías de sistemas de información en la red.

- **Manuales PHP**

<http://php.net>

- **Wikipedia**

<http://es.wikipedia.org/>

- **Plugins utilizados**

<https://www.websequencediagrams.com/>

<https://jqueryvalidation.org/documentation/>

<https://datatables.net/>

<http://www.vissit.com/jquery-event-calendar-plugin/>