



UNIVERSIDAD DE CÓRDOBA
ESCUELA POLITÉCNICA SUPERIOR



GRADO DE INFORMÁTICA

SOFTWARE LIBRE Y COMPROMISO SOCIAL

Software Libre en el desarrollo de aplicaciones web

Aplicación web Control de asistencia

José Luis Guerrero Logroño
i92guloj@uco.es
Universidad de Córdoba – Curso 19/20

Índice de contenidos

1. INTRODUCCIÓN.....	4
2. APLICACIONES WEB.....	5
3. LENGUAJES DE PROGRAMACIÓN WEB: historia, actualidad y tendencias.....	6
3.1. Contexto: el nacimiento de Internet y guerra de navegadores.....	6
3.2. Guerra de lenguajes de programación web.....	7
3.3. Lenguajes de programación web.....	7
Perl (1987).....	8
Python (1991).....	8
Java (1991-1995).....	8
Javascript (1995).....	9
PHP (1995).....	9
JSP (Java Server Pages, 1999).....	9
VBScript (1996).....	10
Jscript (1996).....	10
ASP (Active Server Pages, 1996).....	10
ASP.NET (2002).....	10
4. SERVIDORES WEB.....	13
4.1. Definición y origen.....	13
4.2. Principales servidores web.....	14
Apache Web Server (1995).....	14
Internet Information Services (IIS, 1995).....	14
LiteSpeed (2003).....	15
NGINX (2004).....	15
Google Web Server (GWS, 2008).....	16
4.3. Otros servidores web.....	18
Sun Java System Web Server (1994).....	18
Apache Tomcat (1999).....	18
Cherokee (2001).....	19
Lighttpd (2003).....	19
Microsoft Kestrel (2016).....	19
5. SISTEMAS DE GESTIÓN DE BASE DE DATOS.....	20
5.1. Definición de sistema de gestión de base de datos.....	20
5.2. Componentes de los SGBD.....	20
5.3. Tipos de SGBD.....	21

5.4. Principales proyectos SGBD privativos o propietarios.....	22
Oracle (1977).....	22
IBM Db2 (1983).....	22
Microsoft SQL Server (1989).....	23
5.5. Principales proyectos SGBD de software libre.....	23
PostgreSQL (1989).....	23
MySQL (1995).....	24
MongoDB (2009).....	24
MariaDB (2009).....	25
6. STACKS PARA DESARROLLO DE APLICACIONES WEB.....	27
ANEXO: DESARROLLO DE APLICACIÓN WEB “CONTROL DE ASISTENCIA”.....	28
I. Introducción.....	28
II. Descripción del problema.....	28
III. Modelo conceptual.....	31
IV. Modelo relacional.....	32
V. Descripción detallada de la solución software.....	32
1. Interfaz <i>Administrador</i>	33
1.1. Interfaz panel de control <i>Administrador</i> (figura 2).....	33
1.2. Interfaz panel de control <i>Gestión de usuarios</i> (figura 3).....	33
1.3. Interfaz panel de control <i>Nuevo usuario</i> (figura 4).....	35
1.4. Interfaz panel de control <i>Gestión de contraseñas</i> (figura 5).....	35
1.5. Interfaz panel de control <i>Ver picadas de usuarios</i> (figura 6).....	36
1.6. Interfaz panel de control <i>Permisos de usuarios</i> (figura 7).....	37
2. Interfaz <i>Empleado</i>	38
2.1. Interfaz <i>Registro de picadas</i> (figura 9).....	38
2.2. Interfaz <i>Registro de picadas</i> (figura 10).....	39
2.3. Interfaz <i>Mis incidencias</i> (figura 11).....	39
2.4. Interfaz <i>Cambiar contraseña</i> (figura 12).....	40
VI. phpMyAdmin: administración de base de datos.....	40

1. INTRODUCCIÓN

Conceptos básicos: arquitectura cliente/servidor

Aunque los inicios de Internet se remontan a los años sesenta, no fue hasta la década de los 90 cuando se extendió su uso por todo el mundo gracias a la Web. De forma breve, una aplicación web se puede definir como aquella donde un usuario por medio de un navegador realiza peticiones a una aplicación remota accesible a través de Internet o una intranet, recibiendo una respuesta desde un servidor mostrada en el propio navegador del usuario. Las aplicaciones web permiten, por tanto, la generación automática de contenido haciendo uso de la denominada arquitectura cliente/servidor: un usuario solicita servicios (cliente) a través de un dispositivo y otro permanece a la espera para ofrecer la respuesta (servidor) Se distinguen tres niveles:

- Nivel superior: cliente web que interacciona con el usuario, normalmente un navegador web. Hace de interfaz entre el usuario y el servidor. También se le conoce como *front-end*
- Nivel inferior: proporciona los datos, normalmente una base de datos.
- Nivel intermedio: procesa los datos del nivel inferior para mostrarlos al cliente del nivel superior, normalmente un servidor web. También se le conoce como *back-end*.

Las características esenciales del funcionamiento en la arquitectura cliente/servidor son:

- Aplicaciones distribuidas: múltiples procesadores pueden ejecutar partes distribuidas de una misma aplicación logrando concurrencia de procesos.
- Escalabilidad horizontal: capacidad para modificar la cantidad de clientes que acceden un servicio o aplicación sin que afecte sustancialmente el rendimiento general.
- Escalabilidad vertical: capacidad para migrar servicios o aplicaciones a otros servidores de mayores prestaciones sin que afecte a los clientes.
- Acceso universal a los datos, independientemente a la ubicación del cliente y del servidor.

La arquitectura cliente/servidor permite la separación de funciones en tres niveles:

- Lógica de presentación: entrada-salida de información con el usuario, es decir, obtener información del usuario, envío de información a la lógica de negocio para su procesamiento, y muestra de resultados.
- Lógica de negocio (aplicación): gestión de datos a nivel de procesamiento, actuando como intermediario entre el usuario y los datos de forma que recibe los datos del nivel de presentación, interactúa con la lógica de datos y devuelve el resultado del procesamiento al nivel de presentación.
- Lógica de datos: gestión de datos a nivel de almacenamiento. Las principales tareas sobre los datos son: almacenamiento, recuperación, mantenimiento e integridad.

La siguiente figura muestra el esquema básico de la arquitectura cliente/servidor:

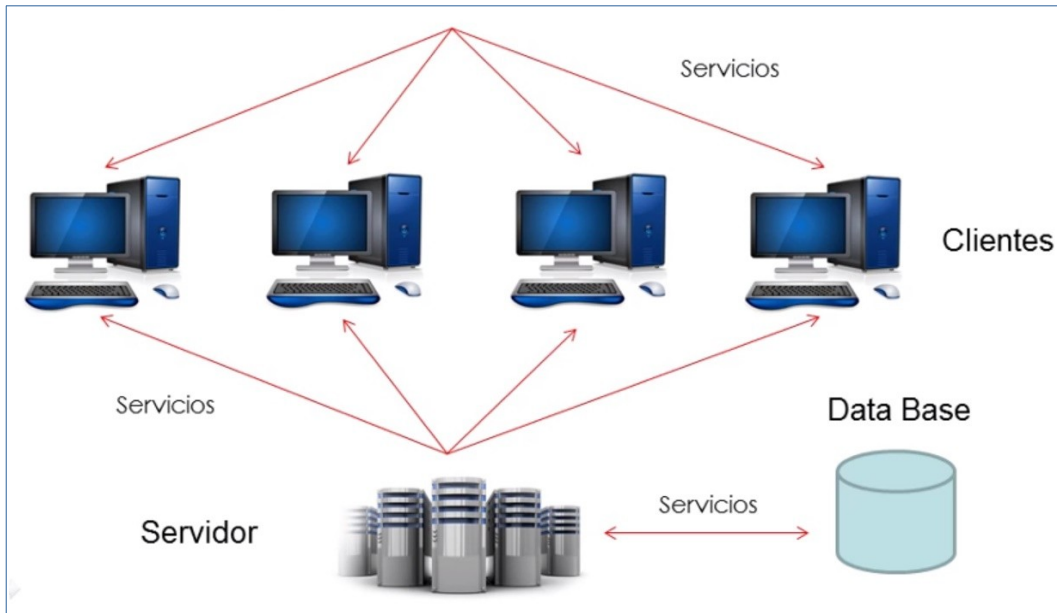


Figura 1: Esquema arquitectura cliente/servidor

2. APLICACIONES WEB

Una aplicación web es un tipo de aplicación cliente/servidor donde tanto uno como otro como el protocolo mediante el que se comunican están estandarizados. El protocolo de comunicaciones más usado en internet es HTTP, con el servicio de conexión TCP/IP que permite el intercambio de información entre distintos dispositivos heterogéneos conectados a una red, ya sea internet o intranet. Como se ha comentado, los actores implicados en una aplicación web son:

- Cliente: es un programa con el que interacciona el usuario para solicitar a un servidor web el envío de la información deseada. La parte cliente de las aplicaciones, *front-end*, suele estar formada por contenido estático desarrollado normalmente mediante código HTML y CSS, y código ejecutable realizado en lenguaje script (PHP, Javascript, VBScript...) de forma que la parte cliente puede ejecutar parte del código para aliviar la carga de ejecución en el lado servidor.
- Servidor: es un programa que se encuentra permanentemente esperando solicitudes de conexión por parte del cliente. La parte servidor de una aplicación web, *back-end*, puede estar formada también sólo por código estático, webs que siempre muestran el mismo contenido, y contenido dinámico o script que se ejecuta en el propio servidor al solicitarlo un cliente, mostrando la salida normalmente mediante una página HTML estándar.

Por lo tanto, para desarrollar una aplicación web necesitamos usar un lenguaje de programación web cliente/servidor, un servidor web, y un sistema de gestión de base de datos. A continuación se mostrarán las alternativas de cada categoría tanto para software privativo como software libre, poniendo el foco en las ventajas que proporcionan estos últimos.

3. LENGUAJES DE PROGRAMACIÓN WEB: historia, actualidad y tendencias

Teniendo en cuenta la separación de funciones en tres niveles expuestos en la descripción de la arquitectura cliente/servidor del apartado anterior, a continuación se describirán las características y algo de historia sobre los principales lenguajes de programación usados, necesarios para el desarrollo de la interfaz y de los procesos solicitud-respuesta de servicios de los clientes al servidor y viceversa que conforman la lógica de presentación.

3.1. Contexto: el nacimiento de Internet y guerra de navegadores

La primera página web del mundo se creó en diciembre de 1990, aunque no se mostró hasta agosto de 1991. Su creador fue [Tim Berners-Lee](#), científico británico conocido como el padre de la World Wide Web. Fue licenciado en Física por la Universidad de Oxford y trabajó en el CERN (Centro Europeo de Investigaciones Nucleares) donde, junto a su equipo, en 1991 implementaron el lenguaje HTML, estándar bajo el que se desarrollaban las primeras webs estáticas.

El primer navegador web que se popularizó fue el denominado *Mosaic* creado en 1993 en la NCSA (National Center for Supercomputing Applications) de la Universidad de Illinois para funcionar sobre sistemas UNIX. En 1994 Tim Berners-Lee fundó el World Wide Web Consortium ([W3C](#)) encargado de generar recomendaciones y estándares asegurando el uso universal y crecimiento de la World Wide Web a largo plazo, y garantizando el lenguaje HTML como el estándar universal.

Así, en ese año el acceso a internet se abrió a la sociedad en general, ya que hasta ese momento su uso había estado reservado sólo para universidades y organismos gubernamentales. Con celeridad Microsoft no tardó en adquirir licencias de *Mosaic* (hasta ahora sólo disponible para UNIX) para desarrollar la primera versión de Internet Explorer. A partir de ese momento, en respuesta al movimiento del gigante tecnológico de Bill Gates, los creadores del navegador *Mosaic* fundaron *Netscape Communications Corporation* (inicialmente llamada *Mosaic Communications Corporation*) con el objetivo de diseñar un nuevo navegador basado en el original al que denominaron *Netscape Navigator* con el objetivo de distribuirlo de forma gratuita entre los usuarios ya que defendían la idea de que el software para internet no debía suponer coste alguno, aunque se trataba de un proyecto propietario, no de software libre ni código abierto (su código se liberaría posteriormente en 1997 sentando los orígenes del Firefox actual)

Sin embargo, la batalla se decantó de parte de Microsoft ya que al comprobar que los usuarios podían acceder a internet usando un navegador gratuito en otros sistemas operativos, en una maniobra inteligente decidió incorporar Internet Explorer al sistema operativo Windows 95 cuya expansión era mayoritaria, lo que provocó una denuncia antimonopolio que finalmente se libró sin consecuencias relevantes para Microsoft, dejando muy tocada a *Netscape* que entonces contaba con una plantilla de sólo 21 programadores, haciendo frente al gigante de Redmond en una lucha en desigualdad de condiciones.

3.2. Guerra de lenguajes de programación web

Paralelamente a la guerra de navegadores, más visible para los usuarios, también se libraba otra batalla de carácter más técnico y profesional que perdura en la actualidad. Se trata de la batalla de los lenguajes de programación. Ambos aspectos van unidos de la mano, ya que durante la guerra de los navegadores los desarrolladores estaban obligados a crear dos versiones de sus páginas, una para Internet Explorer y otra para Netscape, ya que cada uno introdujo sus propias y abundantes extensiones incompatibles entre sí. Hasta finales de 1994 las páginas web mostraban contenido estático, por lo que únicamente se desarrollaban en lenguaje estándar HTML y las primeras versiones de CSS para las hojas de estilos en cascada. A partir de 1995 se introdujo el contenido dinámico en las páginas webs, surgiendo una variedad de técnicas y lenguajes de programación claramente diferenciadas.

Fue Brendan Eich, de *Netscape Communications*, tocada pero no hundida tras la derrota en su heroica y desigual batalla de navegadores, quien se hizo cargo del desarrollo del primer lenguaje de programación para páginas web de contenido dinámico, cuyo nombre inicial fue *Mocha*, renombrado posteriormente como *LiveScript* para quedar finalmente como *Javascript*. El cambio de nombre fue debido al dotar la nueva versión del navegador Netscape con la compatibilidad con la tecnología Java de *Sun Microsystems*, suceso que desde Microsoft tacharon de complot como parte de una estrategia comercial contra su marca, debido a que Netscape se estaba aprovechando de la alta popularidad del momento de Java para generar confusión. Como respuesta, meses más tarde publicó su propio lenguaje al que llamaron *Jscript* intentando así devolver el golpe con la misma jugada de confusión lingüística. Microsoft diseñaba sus propios lenguajes privativos orientados a la creación y diseño de páginas que funcionaran exclusivamente en su navegador Internet Explorer con la idea de monopolizar el mercado, como alternativa a aquellos de software libre compatibles con los distintos navegadores y que solían publicarse con anterioridad.

Los lenguajes de programación web se pueden dividir en dos categorías:

- Lenguajes de programación del lado cliente: se usan para la integración en páginas web, de forma que los script dinámicos se incorporan en el código HTML y se ejecuta lo interpretado no compilado. Mediante la ejecución del lado cliente se pueden validar algunos de los datos en el propio dispositivo del usuario previo a las solicitudes al servidor, aliviando así la carga de procesamiento de este último.
- Lenguajes de programación del lado servidor: son útiles en proyectos donde se tiene que acceder a información centralizada en una base de datos alojada en un servidor. Son necesarios ya que para ejecutar cualquier aplicación web se debe tener acceso a recursos externos al dispositivo del usuario, es decir, a los recursos alojados en los servidores. Al igual que los del lado cliente, se incrustan en el código HTML pero se ejecuta en el servidor.

Actualmente, la mayoría de lenguajes de programación presentan ambas características por lo que se denominan lenguajes del lado cliente/servidor. Por último, cabe resaltar que todos los lenguajes de programación son gratuitos, pero no todos son software libre. Además, los lenguajes de

Microsoft realizan muchas tareas avanzadas apoyándose en otras herramientas especializadas de terceros que no se distribuyen como software libre. Aunque existe un amplio abanico de lenguajes, sólo los más usados históricamente son objeto de mención en el presente documento, por lo que no están todos los que son, pero sí son todos los que están.

3.3. Lenguajes de programación web

- **Perl (1987):** creado por Larry Wall, programador de Unisys, se considera un lenguaje imperativo del lado del servidor derivado de C, bajo licencia GPL y Artistic License, licencia propia propuesta por su creador y aprobada por la Free Software Foundation. Se trata de un lenguaje multiparadigma, tanto para software de escritorio para lo que se creó inicialmente, como aplicaciones web usado desde los principios del www en 1995. Es usado en aplicaciones web tan importantes como Amazon o Ticketmaster. Es multiplataforma, por lo que se puede usar y compilar desde los principales sistemas operativos, viniendo instalado en las principales distribuciones GNU/Linux.



- **Python (1991):** lenguaje de programación interpretado, posee una licencia de código abierto denominada Python Software Foundation License, compatible con la GPL de GNU. Su desarrollador, Guido van Rossum, lo creó como lenguaje privativo a finales de la década de los 80 en el Centrum Wiskunde & Informatica (CWI) en los Países Bajos, liberando el código en 1991. Es un lenguaje sencillo y rápido de aprender que desde 1995 se usa para el desarrollo del *back-end* (lado del servidor) de aplicaciones web. En la actualidad son muy populares sus principales framework, *Django* y *Flask*. Cuenta con una de las comunidades más activas en la organización de eventos y congresos.



- **Java (1991-1995):** desarrollado por *Sun Microsystems* originalmente como lenguaje de escritorio orientado a objetos. Ha ido evolucionando de forma que ha adquirido tal versatilidad que es adecuado para programación bajo prácticamente cualquier paradigma, ya sean aplicaciones de escritorio, como web de lado servidor o sistemas móviles. Sin embargo aunque por sí mismo ofrece todo lo necesario para la creación de aplicaciones web sería demasiado complejo, por lo que realmente su potencia y divulgación se produce gracias a la gran cantidad de frameworks Java disponibles como ayuda en el desarrollo web debido a su alta funcionalidad, diversidad y potencia. Casi la totalidad de frameworks de java son open source, multiplataforma y compatibles con cualquier navegador actual.



- **Javascript (1995):** desarrollado por Brendan Eich, de *Netscape Communications*, fue el primer lenguaje surgido para la creación de aplicaciones web de contenido dinámico. Se trata de un lenguaje de programación cliente-servidor (aunque destaca por su rendimiento en el lado cliente), que facilitó la incrustación de fragmentos de código Java (el más popular de la época) entre las sentencias HTML. El nombre puede llevar a confusión pero Java y Javascript poseen semánticas y propósitos diferentes. Tiene licencia Creative Commons Attribution, es multiplataforma y continúa siendo uno de los lenguajes más extendidos.



- **PHP (1995):** uno de los lenguajes de código abierto más popular para desarrollo web. Al igual que sus competidores, el código de sus scripts se incrusta en el código html para generar contenido dinámico. Se trata de un lenguaje de lado del servidor que se ha popularizado gracias a la simplicidad para los programadores principiantes, junto a las características avanzadas que ofrece para desarrolladores experimentados. Es multiparadigma, multiplataforma y disponible bajo su propia licencia PHP, aceptada por la Free Software Foundation como licencia de software libre no copyleft, y como licencia de código abierto según la Open Source Initiative.



- **JSP (Java Server Pages, 1999):** lenguaje de programación del lado servidor desarrollado por *Sun Microsystems* como alternativa a ASP para competir con Microsoft. Se trata del primer lenguaje de la compañía para aplicaciones web basado en Java, que la propia organización había lanzado años antes para entornos de escritorio. Para ejecutar los scripts se incrusta código java en html siendo compatible con cualquier sistema operativo, por lo que es un lenguaje multiplataforma que sigue vigente en la actualidad.



Por otra parte, a continuación se muestra una reseña en orden secuencial de los principales lenguajes de programación web lanzados por Microsoft como competencia a los anteriores, originalmente de software privativo y únicamente compatibles con Windows para la creación de sitios y aplicaciones web que corrieran sólo bajo Internet Explorer.

- **VBScript (1996):** lenguaje de programación del lado cliente, lanzado por Microsoft por la reciente aparición de la www y la creciente expansión de Javascript, su sintaxis surge como variación de su propio lenguaje de escritorio Visual Basic. Fue parte fundamental de la ejecución de aplicaciones de servidor programadas en el lenguaje de lado servidor ASP, también de Microsoft y lanzado paralelamente, con una cuota importante dominante en el mercado entre 1997-2003. VBScript se incluyó en Internet Explorer 3.0 permitiendo la ejecución de scripts únicamente en dicho navegador.



- **Jscript (1996):** basado en el anterior, de lado cliente, Microsoft se vio obligado a su adaptación para cumplir con la especificación ECMA Script que supuso la norma de estandarización sobre las características de los lenguajes de programación web que introdujeron contenido dinámico. Para la elaboración de esta norma [ECMA International](#) se basó en Javascript, por lo que Microsoft se vio obligado a una rápida reacción para adaptar VBScript a la nueva situación de forma que pasó a ser prácticamente una copia idéntica de Javascript a la que cambiaron el nombre para evitar problemas legales.
- **ASP (Active Server Pages, 1996):** lenguaje de programación del lado servidor, lanzado por Microsoft junto con VBScript y Jscript para generar contenido dinámico en las páginas web ejecutadas bajo Internet Explorer. La tecnología sólo puede ejecutarse en los sistemas operativos de Microsoft. Desde 2002 ASP fue reemplazado por ASP.NET.



- **ASP.NET (2002):** reemplazo del anterior, fue comercializado por Microsoft a principios de 2003. Está construido sobre el Common Language Runtime (CLR), entorno de ejecución desarrollado por Microsoft de forma que los programadores web pueden codificar sus aplicaciones en cualquier lenguaje de la compañía soportado por .NET Framework, tales como C# o Visual Basic .NET. Desde 2016 forma parte del framework Microsoft .NET Core, con la que la compañía decidió liberar la licencia tras el cambio de política con respecto al software libre. Se trata de un lenguaje cliente-servidor, multiplataforma y ejecutable en cualquier navegador, aunque tiene la desventaja de consumir gran cantidad de recursos.



La principal web de referencia para desarrolladores de todo el mundo, <https://stackoverflow.com/>, publica anualmente las estadísticas de uso tanto de lenguajes de programación y frameworks, como lo concerniente a sistemas de gestión de bases de datos, sistemas operativos, plataformas y demás componentes que emplean globalmente los profesionales del sector. Se trata por tanto de unos índices de referencia muy fehacientes, ya que abarca un amplio número de programadores encuestados teniendo en cuenta múltiples factores, tales como país, edad, situación familiar, experiencia, colaboradores de software libre, desarrolladores en compañías de distinta dimensión, estudiantes... en definitiva, un cuestionario que reúne opiniones y datos recogidos mediante encuestas dirigidas a una comunidad de más de 90.000 profesionales. Por tanto la fiabilidad de las estadísticas que publican cada año toman gran relevancia e incluso marcan tendencia.

A continuación se muestran las últimas estadísticas de uso publicadas en abril de 2019 en cuanto a lenguajes y frameworks de programación se refiere:

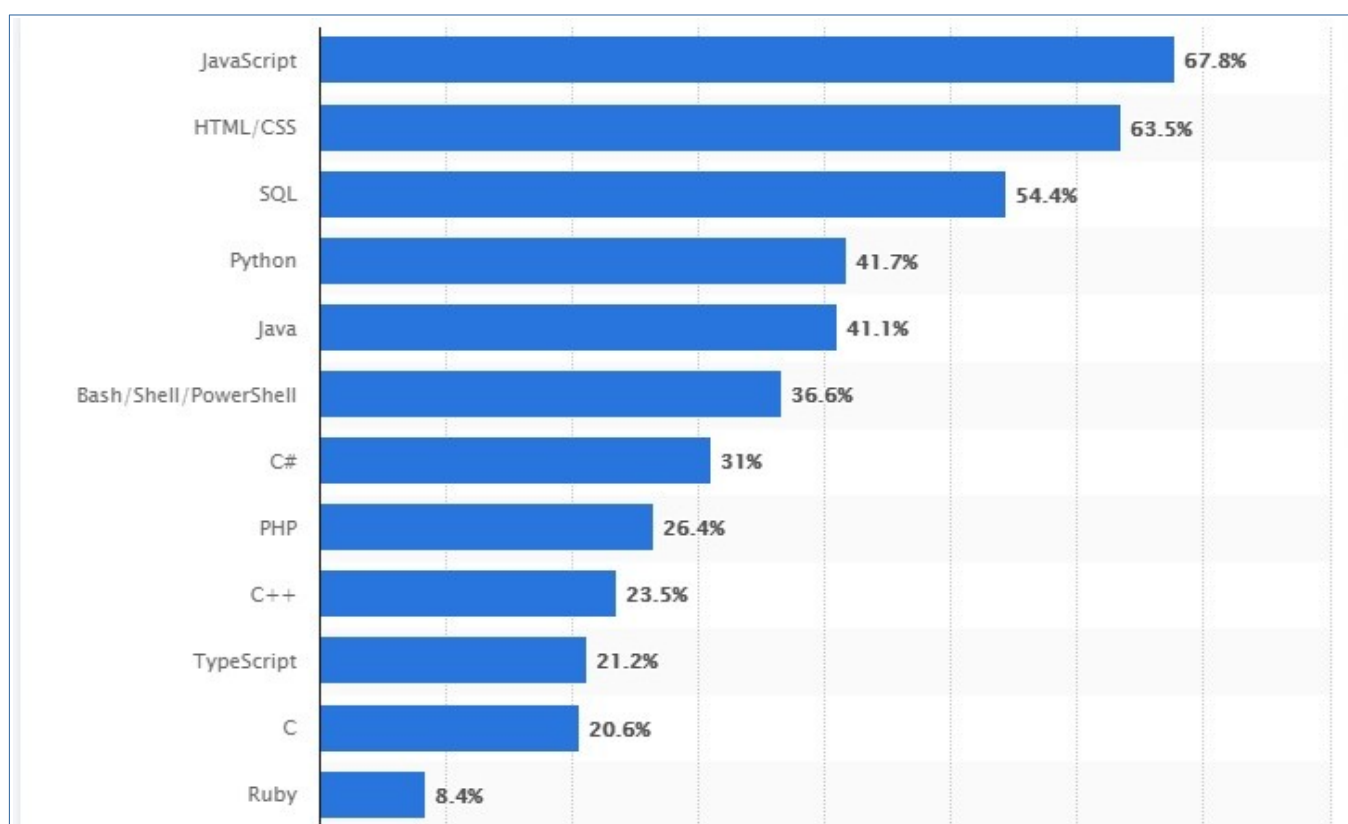


Figura 2: Estadística de lenguajes de programación más usados (Fuente: StackOverflow, abril 2019)

(*) En la estadística aparecen SQL o Bash/Shell/PowerShell que no se consideran lenguajes de programación web al uso como objeto de este estudio, ya que constituyen un lenguaje de dominio específico para sistemas de gestión de base de datos en el primer caso, y procesadores específicos de comandos para ejecutar scripts en el segundo caso.

Del gráfico se puede deducir que las herramientas que inicialmente surgieron como software libre o estándar (*JavaScript, HTML, CSS, Python, Java o PHP*) continúan siendo las más usadas para el desarrollo de aplicaciones web. Por otro lado los lenguajes más extendidos abarcados en el

framework *ASP.NET* de Microsoft (*C#*, *C++*, *TypeScript*, *C*) originalmente privativo y posteriormente agrupado en la plataforma *.NET Core* a partir de 2016 y cuyo código fue liberado, suponen una extensión de uso bastante inferior a los primeros.

En este punto es conveniente volver a insistir en un aspecto ya expuesto en un apartado anterior y es que, aunque *.NET Core*, plataforma relativamente nueva de Microsoft, surge como proyecto de código abierto, los lenguajes de Microsoft realizan muchas tareas avanzadas apoyándose en otras herramientas especializadas de terceros que no se distribuyen como proyectos de software libre, como por ejemplo *ReSharper Ultimate* un conjunto de herramientas muy galardonada y de uso popular entre los programadores del mundo *.NET*, propiedad de *JetBrains* de licencia privativa.

Por otra parte, extrapolando la información respecto a los frameworks más usados, la figura 3 muestra idéntica tendencia con respecto a las herramientas originales surgidas como software libre frente a las creadas por Microsoft:

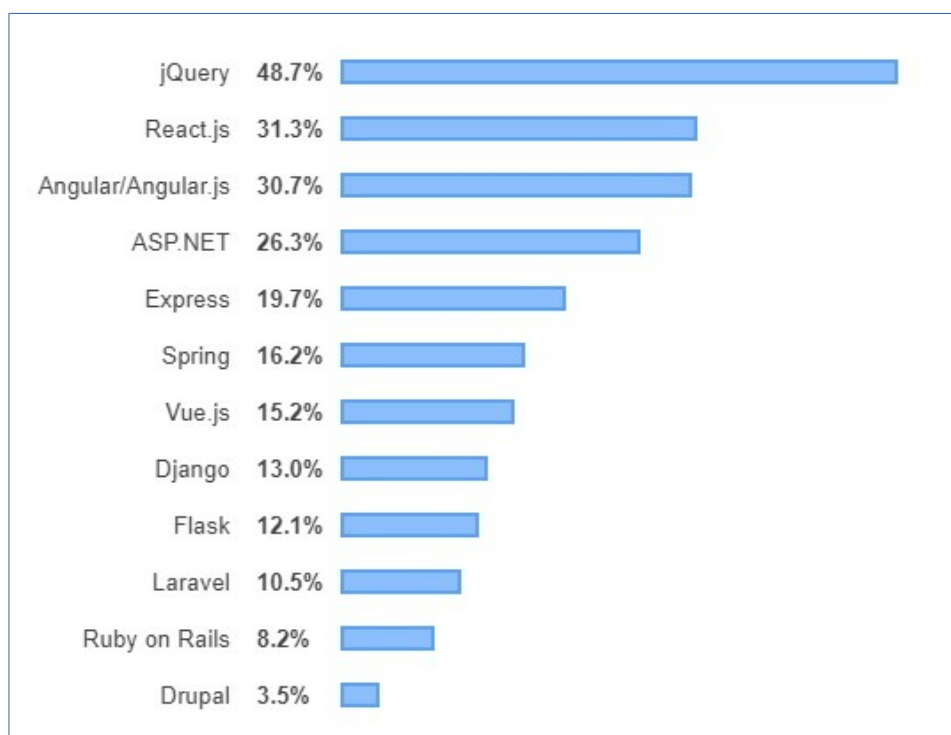


Figura 3: Estadística de frameworks más usados (Fuente: *StackOverflow*, abril 2019)

En este aspecto los frameworks basados en Javascript (*jQuery*, *React.js*, *Angular*, *Express*, *Vue.js*...) dominan sin ningún género de duda frente a *ASP.NET*, componente de *.NET Core* de Microsoft, así como por delante de *Spring* (framework de Java), *Django* y *Flask* (frameworks Python) o *Laravel* (framework PHP). Por último es digno de mención que ni *Swift* (ni su framework *Cocoa*), lenguaje de programación creado por *Apple*, ni siquiera aparecen en los rankings, aún siendo herramientas de software libre bajo licencia Apache 2.0.

4. SERVIDORES WEB

Tras los lenguajes de programación que facilitan la creación de los elementos necesarios que conforman la lógica de presentación, siguiendo con el esquema mostrado en la figura 1 correspondiente a la arquitectura cliente/servidor, a continuación se describirán algunas de las características de los servidores web tanto de software libre como privativos. Los servidores web serían el equivalente a la lógica de negocio de dicha arquitectura, ya que actúan como intermediarios entre los clientes y los datos, procesando los servicios requeridos por los primeros con la lógica de datos para devolver los resultados de dicho procesamiento.

4.1. Definición y origen

En general, la pregunta “¿qué es un servidor web?” puede referirse a dos aspectos: la solución software del servidor en sí, o bien al equipo en el que el software del servidor web se está ejecutando. En este último caso, se habla de host (puede alojar múltiples soluciones software para servidores web) El objetivo de este documento pone el foco en la primera de las definiciones, es decir, servidor como solución software. Por tanto, se define servidor web o servidor http como una solución software que se encarga de la distribución de contenido web en redes internas (intranet) o en internet. En cuanto a una aplicación web, el servidor se encargará de recibir las peticiones de los clientes, procesarlas y servir las a los clientes de forma segura y constante, por lo que debe estar accesible en todo momento. Aunque su función principal es el procesamiento y transferencia de contenido, un servidor web de confianza debe ofrecer otras características adicionales en materia de seguridad, como el cifrado de la comunicación entre cliente y servidor, o cumplir los requisitos para garantizar la autenticación de usuarios para áreas específicas de una aplicación web, así como hacer un uso efectivo del *caching* (almacenamiento en caché de documentos dinámicos para una respuesta eficiente de solicitudes evitando la sobrecarga)

Al igual que los primeros navegadores y la WWW, el desarrollo de los servidores web está estrechamente vinculado al físico e informático británico Tim Berners-Lee (conocido como el creador de internet) quien en 1989 sugirió que el intercambio de información en su compañía (CERN, Organización Europea para la Investigación Nuclear) debería realizarse a través de un sistema de hipertexto más fácil y rápido. En 1990, junto con Robert Cailliau, presentó un proyecto a partir del cual crearon el primer servidor web llamado *CERN HTTPD* o *NCSA HTTPD*



Tim Berners-Lee, *padre de internet*, en el CERN con el primer servidor de la historia NeXTcube

4.2. Principales servidores web

Existen soluciones para servidores web tanto software libre como privativas, diseñadas para diferentes aplicaciones y sistemas operativos. A continuación se estudiarán las más implantadas en la actualidad.

- **APACHE WEB SERVER (1995):** basado en *NCSA HTTPD*, y reescrito totalmente a finales de 1998 y principios de 1999. Aunque Tim Berners-Lee creó el primer servidor web de la historia (*NCSA HTTPD*), Apache puede considerarse el pionero en cuanto a su uso extenso en internet, ya que el primero de ellos sólo se usaba para páginas web de contenido estático de la propia *CERN*, compañía donde ejercía su trabajo su autor, abandonando el proyecto en 1998. Disponible desde el inicio con código abierto, históricamente ha sido el servidor web dominante por antonomasia, de mayor uso tanto entre las herramientas libres como cualquier solución privativa. Llegó a alcanzar una cuota de mercado superior al 70% del total de sitios web del mundo. Actualmente continúa en una posición relevante, aunque ha sido desbancado de la primera posición por *Nginx*, su máximo competidor en la actualidad) Apache es un servidor web multiplataforma (Windows, Unix, GNU/Linux, BSD, Mac...) desarrollado en C de cuya producción y mantenimiento se encarga la *Apache Software Foundation*, creada en 1999 específicamente para dar soporte a la herramienta, y que en la actualidad ha diversificado su actividad con otros productos de software libre. Se distribuye bajo su propia licencia Apache, aceptada por la *Free Software Foundation* y compatible con la GPLv3 de GNU, y da respuesta a sitios web escritos con cualquier lenguaje de programación web, ya sea software libre o privativo (incluso de Microsoft .NET gracias al complemento *Modmono* actuando como servidor de proxy inverso)



- **INTERNET INFORMATION SERVICES (IIS, 1995):** server de carácter privativo, fue creado por Microsoft como alternativa a Apache para ser ejecutado exclusivamente en servidores que arrancaran bajo el sistema operativo Windows NT, siendo su lanzamiento en 1995 a través de una actualización service pack 3 de Windows NT 3.51, e integrado por primera vez de forma nativa en Windows NT 4.0 en julio de 1996. Pese a que los servidores con sistema operativo GNU/Linux dominan el mercado de servidores web, IIS históricamente se sitúa en el top 3 de servidores más usados mundialmente ya que Microsoft sigue abarcando un alto porcentaje de servidores. Por otra parte, IIS resulta tremendamente restrictivo con los lenguajes de programación ya que sólo soporta código escrito con sus distintos frameworks .NET y versiones antiguas en ASP, por lo que no es compatible con sitios web desarrollados en otros lenguajes ni privativos ni de proyectos de software libre, sólo con una minoría de sitios programados en PHP o Perl.



- **LITESPEED (2003):** desarrollado por LiteSpeedTech, fue creado como una de las principales alternativas a Apache para entornos Unix y GNU/Linux. Existe una versión libre open source bajo licencia GPLv3 y gratuita (OpenLiteSpeed), y una versión comercial privativa y de pago. Si bien la primera de ellas no disfruta de una masiva popularidad, se ha decidido incluir este producto debido a la potencia de su versión comercial, que le hace situarse como el cuarto software web server más usado internacionalmente entre el millón de sitios web con mayor tráfico de red, aunque por el contrario, no alcanza ni un 5% de la cuota del mercado global. Tiene su lógica, ya que debido a su alto precio sólo un porcentaje de los sitios de mayor tráfico y nivel de ingresos deciden afrontar el gasto, al contrario que otros sitios de que pueden optar por soluciones gratuitas como Apache, Nginx u otros.



- **NGINX (2004):** posterior a sus dos competidores anteriores y de máxima competencia para Apache, Nginx es un sistema software para servidores web open source, publicado bajo los términos de la licencia BSD simplificada o FreeBSD, que alcanzó a finales de 2019 el primer puesto como servidor web más usado del mundo, arrebatando la corona que estaba en posesión de Apache. Tiene su origen en Rusia, creado por el desarrollador Igor Sysoev en 2002 para dar solución al problema que presentaba Apache con el buscador más usado del país, *Rambler.ru*, que recibía entonces más de 500 millones de solicitudes diarias y donde trabajaba Sysoev como administrador de sistemas. Posteriormente, fue en 2004 cuando se publicó internacionalmente. Sobre el origen, el jefe del proyecto cita textualmente *“Nginx fue escrito específicamente para resolver las limitaciones de desempeño de los servidores web de Apache”*. En 2011 Sysoev fundó la empresa del mismo nombre para proporcionar soporte al software y en marzo de 2019 la compañía fue adquirida por F5 Networks por 670 millones de dólares. Cuenta con una versión gratuita y otra de pago, *Nginx Plus*, es multiplataforma (Windows, Unix, Linux, Mac...) y da respuesta a sitios web escritos con cualquier lenguaje de programación web, ya sea privativo o de software libre. El código está disponible en github en <https://github.com/nginx/nginx>



La figura 4 muestra una comparativa de un estudio realizado por la prestigiosa [W3Techs.com](https://www.w3techs.com) (Web Technology Surveys) en mayo de 2019, a partir del cual se comprueba como Nginx ya se posicionaba como líder entre el millón de sitios web más activos del mundo con el 50,4% frente al 34,10% de Apache o el 9,9% de IIS. Aunque sobre el total de sitios web Nginx alcanzaba un 42,20% frente al 43,30% de Apache, a finales de 2019 y principios de 2020 se confirmó también su liderazgo global, por lo que actualmente se encuentra muy igualados, aunque Nginx se mantiene como líder destacado en cuanto a servidores del millón de sitios web con más tráfico. Ambos servidores están muy extendidos también en las compañías *Fortune 500*, las 500 mayores empresas de EEUU de capital abierto.

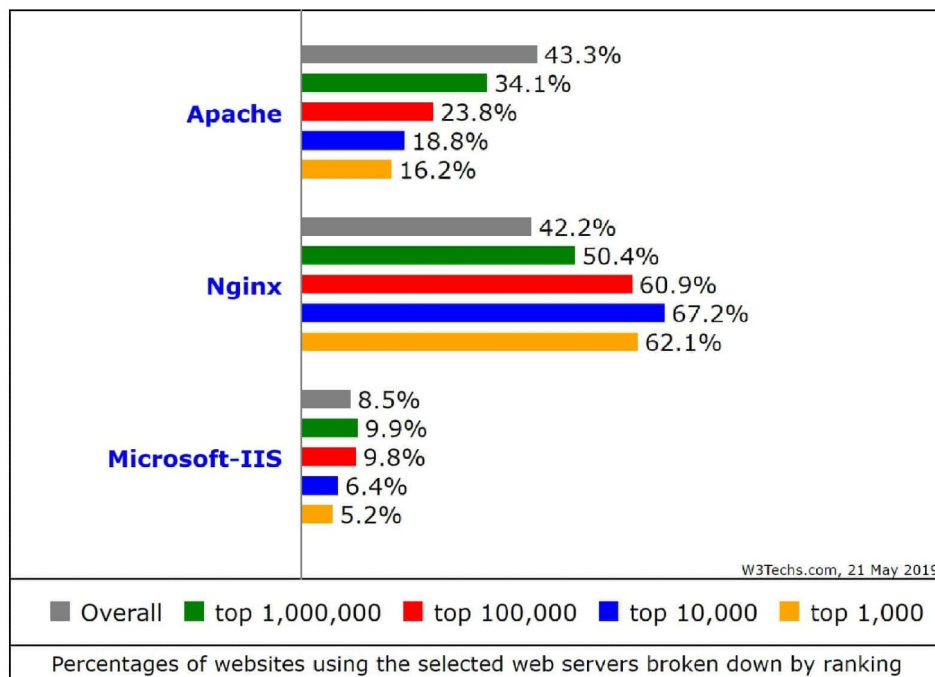


Figura 4: Ranking servidores web (fuente W3Techs.com, mayo 2019)

- GOOGLE WEB SERVER (GWS, 2008):** es el software privativo de Google para web server para su infraestructura interna. Efectivamente han leído bien, privativo y Google en una misma frase. Y es que, aunque muchos usuarios tienden a creer una vinculación entre Google y el software libre por aquello que todos los servicios de Google son gratis, nada más lejos de la realidad *The GNU Project* califica al software de Google como malware (muy recomendable la lectura del artículo del siguiente enlace <https://www.gnu.org/proprietary/malware-google.en.html>) GWS fue creado por Google en 2008 para dar soporte a su ecosistema web en sus hosts server distribuidos por todo el mundo. Debido a la vasta infraestructura que la compañía posee y la diversidad de productos y servicios que proporciona, GWS se ha convertido en el cuarto servidor web más popular, por detrás de los tres anteriores. Existe poca documentación al respecto aunque se conoce que está desarrollado en C++ y que se ejecuta sobre una distribución GNU/Linux basada en Debian.



Para finalizar este apartado, las figuras 5 y 6 muestran el ranking *Netcraft* (<https://news.netcraft.com/archives/2020/04/08/april-2020-web-server-survey.html>) actualizado a fecha de abril de 2020 donde puede verse como Nginx es el servidor web más usado en toda la red, seguido de Apache, Microsoft y Google (figura 5) y, sin embargo, como Apache vuelve a ser el líder en cuanto al millón de sitios web de mayor actividad, seguido por Nginx, Microsoft y LiteSpeed (figura 6)



Figura 5: Ranking servidores web más usados globalmente (fuente Netcraft, abril 2020)

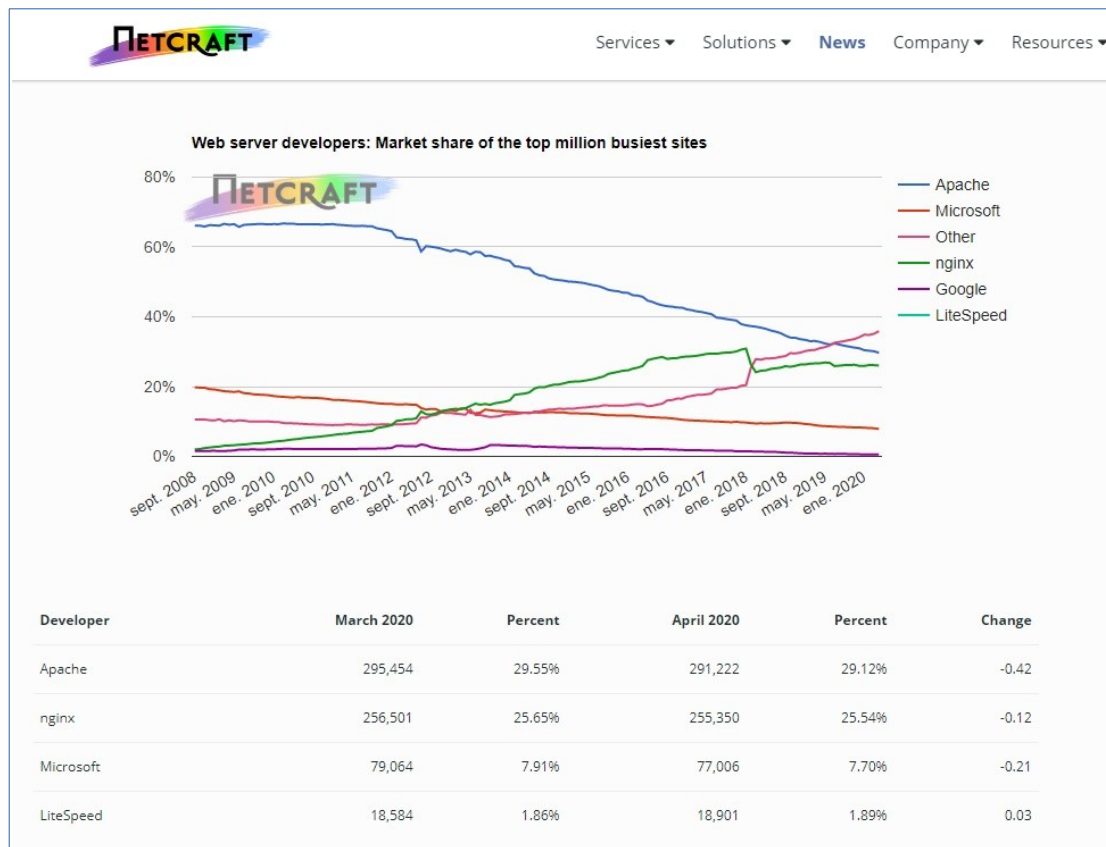


Figura 6: Ranking servidores web más usados en el millón de sitios web con más tráfico (fuente Netcraft, abril 2020)

4.3. Otros servidores web

En este apartado se mostrará una breve reseña sobre otros web server con menor impacto que los expuestos en el punto anterior pero igualmente destacables por su historia y otras características.

- **Sun Java System Web Server (1994):** originalmente **Netsite 1.0** y actualmente renombrado como **Oracle iPlanet Web Server** tras la adquisición de Sun Microsystems por Oracle Corporations en 2010, ha cambiado de nombre en distintas ocasiones, conocido también como Netscape Enterprise Server, iPlanet Web Server o Sun Web Server. Fue desarrollado originalmente por Netscape Communications Corporations, que más tarde fue adquirida a su vez por Sun Microsystems quien liberó su código bajo licencia BSD como Open Web Server. Desde que Oracle se encarga de la producción las últimas versiones se han lanzado como privativas. Por otra parte la empresa distribuye un complemento llamado Oracle Web Logic que incluye Oracle HTTP Server, la versión de su servidor web de código abierto basado en Apache (no se especifica bajo qué licencia de software libre se distribuye. Incluso existen varias preguntas sobre este aspecto en su web oficial por parte de varios usuarios que continúan sin respuesta) Tanto este último como iPlanet son multiplataforma y prestan servicio a sitios web desarrollados con tecnología Java, Active Server Pages (ASP), y, en el caso de iPlanet, PHP entre otros. Actualmente su uso es muy restringido a sitios web relacionados con Oracle y compañías afines.



- **Apache Tomcat (1999):** es una implementación de código abierto del servlet de Java , JSP y tecnologías de expresión en Java en general originalmente creadas por Sun Microsystems que más tarde adquiriría Oracle cuyo código donó a Apache Software Foundation. Un servlet es un programa de Java que amplía las capacidades de un servidor. Aunque los servlets pueden responder a cualquier tipo de solicitudes, generalmente implementan aplicaciones alojadas en servidores web. Dichos servlets web son la contrapartida de Java de otras tecnologías de contenido web dinámico, como PHP y ASP.NET. El software Apache Tomcat se desarrolla bajo el proyecto Jakarta en la Apache Software Foundation, en un entorno abierto y participativo y se publica bajo la licencia de Apache v2. Tomcat está desarrollado en Java y funciona en cualquier sistema operativo que disponga de la máquina virtual de Java. A principios de siglo estaba muy extendido en entornos Java pero actualmente apenas alcanza un 0,40% de cuota del mercado global de sitios web ya que su funcionalidad principal en estos momentos es el de servidor de aplicaciones Java.



- **Cherokee (2001):** considerado como la primera alternativa seria de software libre con respecto a Apache (LiteSpeed y Nginx son posteriores, e IIS es anterior pero de licencia propietaria) Fue creado por el español Álvaro López Ortega (Madrid) entonces ingeniero de Sun Microsystems con la idea de fabricar un producto más ligero que Apache, pues según su creador, éste se estaba quedando obsoleto y empezaba a deteriorar su rendimiento en cuanto a velocidad de respuesta ante el boom de la www. Es un servidor web programado en C, multiplataforma (Linux, MacOS, Solaris, BSD) aunque no compatible con Windows y soporta la mayoría de las tecnologías web más extendidas. Está disponible bajo licencia GNU GPL. En sus inicios tuvo una gran aceptación incluso por grandes empresas como Banesto aunque su producción parece estar algo estancada pues su última versión data de 2013 y hoy en día apenas se recuerda.



- **Lighttpd (2003):** software escrito en C por Jan Kneschke como parte de su tesis doctoral en ingeniería eléctrica y ciencias de la computación de la University of Applied Sciences de Kiel (Alemania) Software libre distribuido bajo la licencia BSD y sólo disponible para UNIX y GNU/Linux, aunque existe una versión para Windows diseñada por otra comunidad distinta de actividad mucho más reducida. Llegó a situarse como quinta solución web server del mundo en 2007 manteniendo ese puesto hasta 2011, su rápido despliegue se debió a su optimización para trabajar en hosts de bajos recursos, además de facilidad de instalación y configuración. Actualmente sigue presente con un bajo porcentaje de implementación ante los gigantes que copan la mayoría del mercado.



- **Microsoft Kestrel (2016):** es un servidor web multiplataforma (Windows, Mac y GNU/Linux) de código abierto (<https://github.com/aspnet/KestrelHttpServer>) incluido por Microsoft en ASP.NET Core, tras el cambio de política del gigante tecnológico con respecto al software libre. Distribuido bajo licencia MIT y Apache 2, es un proyecto de .NET Foundation. Puede proporcionar servicios a sitios y aplicaciones web desarrolladas en cualquier lenguaje de programación. Sin embargo, y como la propia Microsoft reconoce (<https://docs.microsoft.com/es-es/aspnet/core/fundamentals/servers/kestrel?view=aspnetcore-3.1>), no conviene exponer Kestrel como servidor web en internet ya que no cuenta con características avanzadas como IIS, Apache o Nginx, por lo que está pensado para uso en entornos de prueba para desarrolladores.

5. SISTEMAS DE GESTIÓN DE BASE DE DATOS

Tras los lenguajes de programación que facilitan la creación de los elementos necesarios que conforman la lógica de presentación, y los servidores web que serían el equivalente a la lógica de negocio actuando como intermediarios entre los clientes y los datos, a continuación se estudiarán algunos de los sistemas de gestión de base de datos, que compondrían la lógica de datos completando así los actores que intervienen en la arquitectura cliente/servidor. En esta última capa se gestionan las operaciones relativas a la información almacenada en un servidor (que podría ser el mismo servidor web u otros distintos específicos, práctica más habitual) y se devuelven a la lógica de negocio que a su vez hará lo propio con la lógica de presentación mostrando los resultados al cliente que inició la solicitud.

5.1. Definición de sistema de gestión de base de datos

Prácticamente ninguna aplicación web podría funcionar sin un sistema de base de datos, por lo que no viene mal familiarizarse con los conceptos básicos relacionados con este tipo de tecnología. El sistema de gestión de base de datos subyacente a toda base de datos es tan importante como el propio conjunto de los datos, ya que sin su intervención no sería posible administrarlos. La propia base de datos y el sistema gestor de base de datos o SGBD (también llamado DBMS, del inglés Database Management System) conforman lo que se denomina el sistema de base de datos (lo que suele denominarse simplemente como base de datos).

En términos generales, un SGBD es un software que sigue un modelo de sistema de base de datos y, por lo tanto, resulta decisivo a la hora de su configuración, uso y mantenimiento. En una aplicación web, sólo cuando el sistema gestor de base de datos está instalado y configurado, los usuarios pueden tener acceso a los datos. Los permisos de lectura y escritura, así como las funciones de administración generales, se establecen mediante las interfaces específicas de la aplicación y el lenguaje de definición de datos correspondiente. El más conocido de estos lenguajes es SQL (Structured Query Language, lenguaje de consulta estructurada)

5.2. Componentes de los SGBD

Un sistema de gestión de base de datos consta de varios componentes que contribuyen al buen funcionamiento del software. Los elementos básicos que lo conforman son tres: el diccionario de datos, el lenguaje de definición de datos y el lenguaje de manipulación de datos.

- **Diccionario de datos:** consiste en una lista de metadatos que reflejan las características de los diversos tipos de datos incluidos en la base de datos. Además, estos metadatos informan sobre los permisos de uso de cada registro y su representación física. De esta manera, el diccionario proporciona toda la información relevante sobre los datos almacenados.

- **Lenguaje de definición de datos:** el lenguaje de definición de datos, también llamado lenguaje de base de datos o DDL (Data Definition Language), sirve para estructurar el contenido de la base de datos. Gracias a este lenguaje, es posible crear, modificar y eliminar objetos individuales, como referencias, relaciones o derechos de usuario.
- **Lenguaje de manipulación de datos:** mediante el lenguaje de manipulación de datos o DML (Data Manipulation Language), se pueden introducir nuevos registros en la base de datos, así como eliminar, modificar y consultar los que ya contiene. Este lenguaje también permite comprimir y extraer los datos.

5.3. Tipos de SGBD

El objetivo de instalar un sistema gestor de base de datos es administrar los registros de la mejor manera posible. Como ya se ha mencionado, existen varios modelos para tal fin que difieren básicamente en la manera en que se estructuran los datos. Por lo tanto, decidirse por un DBMS siempre implica decantarse por un modelo de base de datos concreto. Según el modelo de base de datos, los SGBD se clasifican en:

- **Relacional:** es el más común. Los datos se estructuran mediante filas en tablas. La ventaja de este modelo radica en la posibilidad de crear diferentes relaciones entre las filas y presentarlas en columnas.
- **De red:** los datos se representan mediante colecciones de registros, y las relaciones entre los datos se representan mediante enlaces, que se pueden ver como punteros. Los registros en la base de datos se organizan como colecciones de grafos dirigidos.
- **Jerárquico:** el modelo jerárquico es similar al modelo de redes, ya que los datos y las relaciones entre los datos se representan mediante registros y enlaces, respectivamente. Este modelo se diferencia del anterior en que los registros se organizan como colecciones de árboles en lugar de grafos dirigidos.
- **Orientado a objetos:** introduce el concepto de herencia, al igual que en los lenguajes de programación orientados a objetos. En este caso los objetos pueden transferir algunos de sus atributos a otros pero regulado a través del SGBD.
- **Orientado a documentos (documental):** las bases de datos documentales almacenan cada registro y sus datos asociados en un documento. Cada documento contiene datos semiestructurados que pueden ser consultados con el uso de varias herramientas de

consulta y análisis del SGBD. En lugar de columnas con nombres y tipos de datos del modelo relacional, un documento contiene una descripción del tipo de datos y el valor de dicha descripción.

Existen sistemas de gestión de base de datos de alto rendimiento tanto privativos como distribuidos bajo alguna de las licencias de software libre. En ambos casos, las herramientas han alcanzado un alto porcentaje de cuota de mercado entre los sitios web de todo el mundo. A continuación se estudiarán los más populares, ordenados cronológicamente según fecha de publicación.

5.4. Principales proyectos SGBD privativos o propietarios

- **Oracle (1977):** SGBD relacional de carácter privativo desarrollado por Oracle Corporation. Es considerada como la compañía que introdujo la filosofía relacional en los SGBD. Tradicionalmente ha sido el más popular por excelencia entre el tejido empresarial, considerado históricamente como el más completo y robusto, destacando sobre todo por su soporte de transacciones, estabilidad, escalabilidad y multiplataforma. Es compatible con todo tipo de lenguaje de programación. Sin embargo su principal desventaja es su alto coste, aunque existe una opción gratuita *Express Edition* con muchas limitaciones respecto a la versión comercial. Es tal la confianza en el servicio que ofrece que la compañía compensa económicamente a sus clientes con millones de euros o dólares en caso de caídas de sus servidores de datos. Oracle es el dominador del mercado tanto en internet como en intranet, pero con menor diferencia en el primero de los casos con respecto a sus competidores, mientras que en servidores de datos de grandes compañías es el líder indiscutible.



- **IBM Db2 (1983):** SGBD relacional de carácter privativo desarrollado por IBM Corporation en 1970 para uso interno, aunque empezó a comercializarse en 1983. Multiplataforma y de soporte a la mayoría de lenguajes de programación. Es un SGBD que proporciona características muy avanzadas, de alto coste usado para base de datos a gran escala para grandes compañías. Históricamente siempre ha estado entre los 5 SGBD más usados en la web y en los tres primeros puestos en cuanto a calidad e implantación en grandes multinacionales.



- **Microsoft SQL Server (1989):** SGBD relacional de carácter privativo, lanzado por Microsoft para competir frente a Oracle e IBM. Es multiplataforma, aunque hasta el año 2017 sólo era compatible con sistemas Windows. Sin embargo, tras el cambio de estrategia de la compañía con respecto al mundo del software libre derivó en la compatibilidad con sistemas GNU/Linux, para intentar aumentar su presencia y reducir distancias con Oracle y MySQL (otro de sus principales competidores como se verá más adelante). Va destinado principalmente para desarrolladores que usan programación en .NET, aunque también es compatible con otros lenguajes como PHP, Javascript, etc. Está desarrollado en C/C++ sobre la base de un SGBD anterior llamado *Sybase* (creado por la compañía del mismo nombre y actualmente en desuso) algo que acarreó disputas legales por los derechos de autor. Hace uso de Transact-SQL (T-SQL, derivada del lenguaje SQL) como lenguaje de desarrollo para la manipulación de datos. Pone a disposición de sus clientes tanto versión gratuita (con limitaciones) como sus versiones comerciales, que realmente focalizan su modelo de negocio, dirigidas a grandes empresas con enormes bases de datos.



5.5. Principales proyectos SGBD de software libre

- **PostgreSQL (1989):** primer SGBD relacional publicado como código abierto, distribuido gratuitamente bajo su propia licencia PostgreSQL, similar a las licencias BSD o MIT. Su implementación comenzó en 1986 bajo el nombre de POSTGRES bajo licencia BSD, y fue dirigida por el profesor Michael Stonebraker, de la Universidad de California en Berkeley y patrocinado por grandes entidades como la Defense Advanced Research Projects Agency (DARPA), Army Research Office (ARO), o la National Science Foundation (NSF). Su primer lanzamiento data de 1989 pero restringido a unos pocos usuarios externos para evaluación de rendimiento y poniéndose a disposición global en 1991 tras su depuración y ostensible mejora. En 1994 el proyecto se dio por finalizado y el grupo original de desarrolladores se disolvió. En 1995 dos graduados de la universidad, Andrew Yu y Jolly Chen, retomaron el proyecto añadiendo soporte para lenguaje SQL y renombrándolo como Postgres95 y más tarde PostgreSQL, cuya primera versión se lanzó en 1996 como privativa y liberada a comienzos de 1997. Como su licencia no restringe su comercialización, grandes empresas lanzaron sus propios productos PostgreSQL, como Red Hat, Illustra, GreenPlum e incluso Sun Microsystems. Actualmente es el cuarto SGBD más usado del mundo y el segundo en cuanto a proyectos de software libre sólo por detrás de MySQL. Optimizado para la gestión de grandes volúmenes de datos, ha recibido gran cantidad de premios y fue considerado sistema del año tanto en 2017 como 2018 por su mayor crecimiento respecto a sus competidores.



- **MySQL (1995):** es el SGBD relacional por excelencia y el sistema open source más usado del mundo, con una distribución entre más de 100 millones de sitios web. Es líder en aplicaciones web desarrolladas como software libre y, aunque abarca empresas con grandes volúmenes de datos, su modelo de negocio principal va dirigido a pequeñas y medianas empresas que requieren la gestión de menor volúmenes de datos, aunque de dimensiones considerables. MySQL fue creado por la compañía sueca MySQL AB, con el objetivo de ofrecer opciones eficientes y fiables de gestión de datos para usuarios domésticos y profesionales. Originalmente se creó como software comercial, aunque poco después, en el año 2000 se puso el código a disposición pública siguiendo los términos de la GNU GPL (<https://github.com/mysql/mysql-server>) A partir de 2001 su popularidad creció exponencialmente. MySQL AB fue comprada en 2008 por Sun Microsystems siendo ésta su vez adquirida por Oracle en 2010. MySQL es propiedad de Oracle, por lo que se afianza como la compañía líder del sector de los SGBD reuniendo tanto su producto inicial, Oracle Database para empresas de gigantescos volúmenes de datos, como Mysql para el resto de sitios y aplicaciones web con menores requerimientos. Este aspecto posibilita el esquema de doble licenciamiento. La base de datos se distribuye en varias versiones, una Community distribuida bajo la GPLv2 de GNU, y varias versiones Enterprise para aquellas empresas que quieran incorporarlo en productos privativos. Se trata de un sistema multiplataforma, desarrollado en C/C++ y compatible con la mayoría de lenguajes de programación. Ha recibido numerosos premios siendo el más actual este mismo año 2020 cuando ha sido galardonado como el sistema que ha alcanzado un mayor índice de popularidad durante 2019. Es usado en sitios web de la dimensión de Wikipedia, Google, Facebook, Twitter o Youtube entre otros.



- **MongoDB (2009):** frente a los sistemas de modelos relacionales, MongoDB es una base de datos documental considerando su apuesta como más expresiva y potente, al ser la forma más natural y productiva de trabajar con datos según la consideración de sus creadores. Se trata de un sistema NoSQL, ya que no utiliza el lenguaje SQL para interactuar con los datos. Su desarrollo comenzó en 2007 de la mano de 10gen Inc. (renombrada como MongoDB Inc.) y fue publicado en 2009 bajo la licencia AGPL de código abierto de GNU. Los drivers para lenguajes de programación están bajo licencia Apache. Es multiplataforma y cuenta con versión gratuita y otra comercial que incluye soporte y otras características y que resulta bastante económica comparada con las versiones comerciales de sus competidores. Es usada en sitios web de la dimensión de Google, Adobe, Telefónica, ebay entre muchos otros. Aunque los SGBD relacionales satisfacen la mayoría de necesidades el nivel de popularidad de MongoDB crece considerablemente cada año ya que la mayoría de grandes empresas están optando por introducir el modelo de datos documental para muchos de sus servicios siendo MongoDB el sistema más potente NoSQL en la actualidad.



- **MariaDB (2009):** aunque no aparece en las estadísticas de los principales SGBD, MariaDB es uno de los sistemas más distribuidos en los famosos CMS o gestores de contenidos tan usados globalmente como WordPress, Drupal o Neos, y su uso crece de manera exponencial año tras año. Su importancia parte desde su propio origen, y es que MariaDB constituye prácticamente una copia exacta de MySQL ya que comparte creador, el finlandés Michael Widenius, cofundador de MySQL AB. Widenius decidió salir de Sun Microsystems donde lideraba el proyecto MySQL cuando Oracle se hizo con la compañía. El motivo no era otro que el temor a que Oracle privatizara y comercializara MySQL evitando el acceso libre y gratuito a un SGBD de tal magnitud, por lo que decidió fundar su propia organización, Monty Program AB para desarrollar su propia mini-copia (fork, miniversión estándar) de MySQL a la que denominó MariaDB en honor al nombre de su hija más joven, y la publicó bajo licencia pública general (GPL) de GNU, autoproclamándose el salvador mundial del acceso sin restricciones a un SGBD de tal rendimiento. Actualmente su temor no se ha hecho realidad y Oracle continúa con MySQL como un producto de doble licencia, una privativa y otra licenciada bajo GPLv2 como se ha visto anteriormente. Poco después de la bifurcación muchos desarrolladores de MySQL se unieron al proyecto MariaDB, y compañías como Alibaba, Tencent, IBM, Microsoft o Booking se convirtieron en grandes patrocinadores del proyecto.

Por otra parte MariaDB es uno de los sistemas predeterminados en la mayoría de distribuciones GNU/Linux (también es multiplataforma) siendo el más extendido entre los desarrolladores para entornos de prueba.



La figura 7 muestra la estadística de popularidad de los principales SGBD elaborada por la prestigiosa web especializada *Statista.com* actualizada a septiembre de 2019. El ranking se elabora con su propio sistema de puntuación basado en la valoración media de un amplio conjunto de factores como menciones en web especializadas, ofertas de empleo, opinión de profesionales, relevancia en redes sociales entre otros muchos.

Ranking of the most popular database management systems 2019*

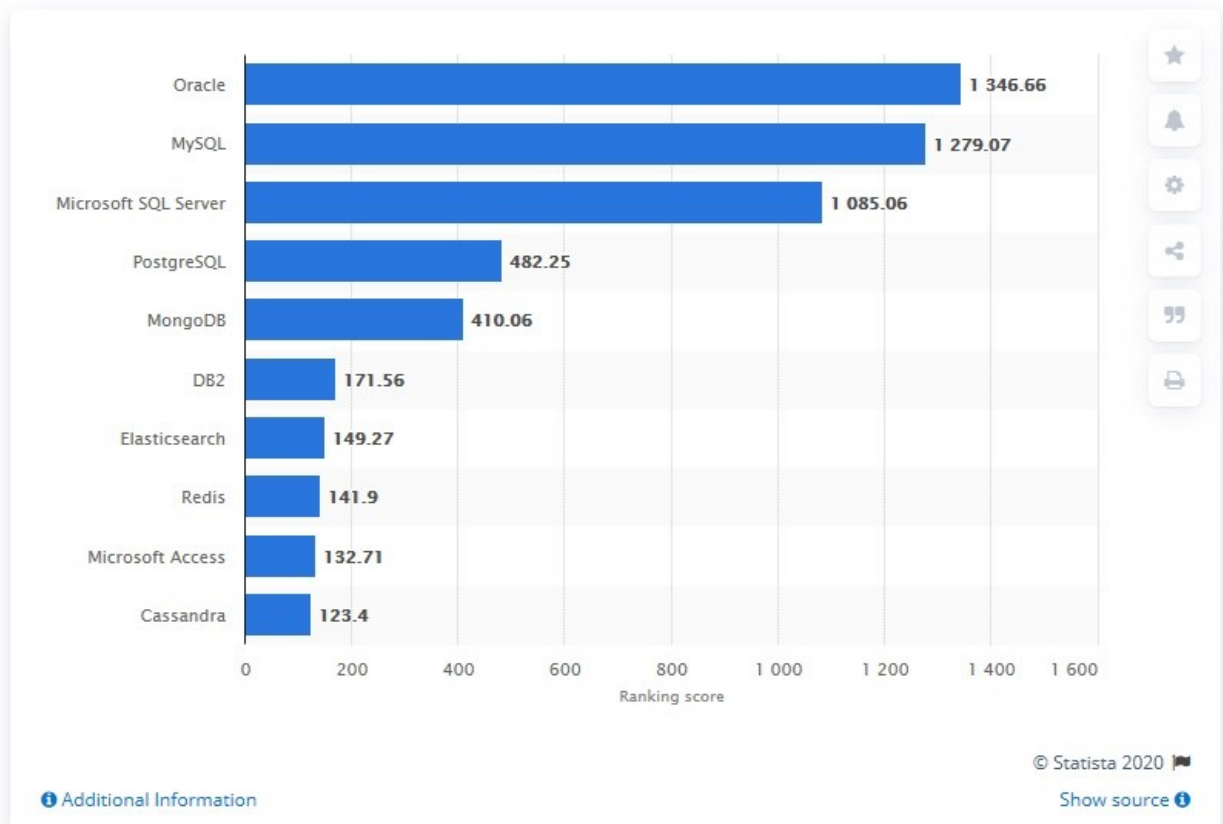


Figura 7: SGBD más populares en 2019 (fuente Statista.com)

Se puede comprobar como Oracle es el rey absoluto en cuanto a soluciones de base de datos se refiere, ya que son muchos años consecutivos de liderazgo con sus dos sistemas, Oracle (versión privativa para enormes volúmenes de datos) y MySQL, versión de software libre tanto para proyectos grandes (aunque de menor escala que los primeros) como de menores dimensiones. El software privativo de Microsoft se afianza en el podium que no ha abandonado desde su creación. Por otro lado, aunque PostgreSQL y MongoDB no son productos diseñados para manejar bases de datos de la magnitud de los gigantes tecnológicos, se puede comprobar que están por delante incluso del sistema IBM DB2, lo que da muestra de la fortaleza de estos sistemas de software libre.

6. STACKS PARA DESARROLLO DE APLICACIONES WEB

Cuando un programador se inicia en el desarrollo de un sitio o aplicación web en un equipo informático es necesario disponer de un entorno local que integre los elementos necesarios para comprobar el diseño y funcionalidad del proyecto, es decir, que integre los lenguajes de programación, el servidor web para procesar los servicios a los clientes, y el SGBD que facilite la interacción del usuario con los datos, permitiendo testear tanto el *front-end* como el *back-end*. Estos elementos pueden instalarse individualmente o bien mediante paquetes o stacks que integran todos los componentes, facilitando así su instalación y configuración. En definitiva, se trata de un software que simula la arquitectura cliente/servidor en un equipo de escritorio.

A continuación se muestran los stacks de desarrollo de software libre más extendidos, que integran los distintos componentes software libre para desarrollo de aplicaciones web:

- **LAMP:** el más popular para entorno GNU/Linux. Sus siglas indican los componentes open source que integra el stack: **L** (Linux, sistema operativo donde instalar la herramienta), **A**pache (servidor web, aunque existen versiones con Nginx, Cherokee o Lighttpd), **M**ariaDB (como sistema gestor de base de datos, aunque hay versiones que incluyen **M**ySQL) y **P**hp (lenguaje de programación, aunque hay versiones que incluyen **P**erl y/o **P**ython).
- **WAMP:** idéntico al anterior para ejecutar en entornos **W**indows.
- **MAMP:** igual que los anteriores, disponible para entornos **M**acOS.
- **XAMPP:** igual que los anteriores disponible para multiplataforma (la letra **X** indica la independencia del sistema operativo) y la última **P** indica su compatibilidad con **P**erl y/o **P**ython, junto con PHP como componente común)
- **LAPP:** igual que LAMP pero usando PostgreSQL como SGBD. Pensado para proyectos empresariales de mayor envergadura.
- **MEAN:** igual que las anteriores pero diseñada para el desarrollo con Javascript. Integra MongoDB como SGBD, Express (framework para back-end de Node.js) Angular.js (framework para front-end de código abierto desarrollado por Google para la creación de *Single-Page Applications*, aplicaciones que no necesitan recargar la página), Node.js (entorno en tiempo de ejecución basado en Javascript, multiplataforma que permite la función de servidor web entre otras)
- **MEEN:** integrando Ember.js como framework para el front-end en lugar de Angular.js.
- **MERN:** igual a los anteriores integrando React.js para el front-end.
- **JAVA EE:** el de mayor antigüedad y específico para los entornos Java, integra PostgreSQL o MongoDB, según la versión, Spring Boot como framework para back-end, JSP/HTML/CSS como framework para front-end y la plataforma Apache Tomcat como servidor web.

Por otra parte Microsoft también diseñó su propio stack de herramientas para el desarrollo de aplicaciones web, sólo disponible para Windows, llamado **WISA** que integra en **W**indows, los componentes Internet Information Services (IIS) como servidor web, **S**ql Server como SGBD, y **A**SP.net como entorno para desarrollo de front-end y back-end.

ANEXO: DESARROLLO DE APLICACIÓN WEB “CONTROL DE ASISTENCIA”

I. Introducción

En aplicación de los conocimientos adquiridos durante el trabajo de documentación para la asignatura *Software Libre y Compromiso Social* de la titulación correspondiente al Grado de Informática en la Escuela Politécnica Superior de la Universidad de Córdoba durante el presente curso 2019/2020, se ha desarrollado la siguiente aplicación web denominada **Control de asistencia**.

Para su desarrollo se ha optado por el paquete o stack XAMPP, debido a su facilidad de instalación y mantenimiento, flexibilidad de sistema operativo (multiplataforma), y la disponibilidad del componente del lenguaje de programación PHP, de importancia para el autor del presente proyecto por su facilidad de aprendizaje debido a las similitudes con el lenguaje C del que se disponen conocimientos y experiencia previa, además de ser considerado como uno de los mejores lenguajes para back-end. Por otra parte, incluye Apache como servidor web y MariaDB como SGBD, aunque para la magnitud de la aplicación es indiferente cualquier alternativa, siempre que cumplan los requisitos de software libre.

El autor pretende demostrar como se puede desarrollar una aplicación web eficiente en un entorno laboral real haciendo uso exclusivamente de herramientas de software libre como son GNU/Linux como sistema operativo, Apache como servidor web, MariaDB como SGBD, y los lenguajes de programación PHP, HTML y CSS, además de algunos scripts del framework jQuery de Javascript. Por otra parte se ha decidido no hacer uso de los habituales framework de trabajo, ya que la dimensión de la aplicación no es grande y se ha optado por programar desde cero con el fin de entender y aplicar el código del lenguaje estrictamente necesario y asimilar su aprendizaje. Tampoco se usan otro tipo de elementos como Bootstrap u otros que complementan el aspecto visual de una aplicación, por lo que se ha usado el lenguaje CSS como hoja de estilos desde cero.

II. Descripción del problema

LA EMPRESA ha decidido introducir el servicio de control de presencia con el fin de optimizar la transparencia entre el departamento de recursos humanos y los empleados en cuanto al cumplimiento del calendario laboral se refiere.

Para ello LA EMPRESA ha requerido el desarrollo de un sistema software que permita el almacenamiento en formato electrónico de los registros de presencia de cada empleado para la mejora de su rendimiento interno y optimización de recursos. El objetivo principal en cuanto a la funcionalidad del sistema software será el de proporcionar los servicios requeridos, que serán descritos en apartados posteriores del presente documento, de manera que cada empleado podrá acceder desde cualquier equipo informático conectado a la red local, a una herramienta donde

poder registrar el horario de entrada y salida de su jornada laboral diaria, así como cualquier incidencia, permiso o licencia que tenga relación con el saldo horario. Así mismo, LA EMPRESA tendrá acceso a dicho registro y podrá comprobar el cumplimiento del saldo horario de cada empleado con respecto al calendario laboral previsto.

La herramienta contará con dos perfiles de usuario:

- Perfil “ADMINISTRADOR”
- Perfil “EMPLEADO”

Inicialmente el sistema se entrega al personal responsable de recursos humanos designado por LA EMPRESA con un usuario con perfil administrador por defecto, que tendrá acceso a la total funcionalidad del sistema. Desde dicho perfil administrador se deberán registrar inicialmente al resto de usuarios, tanto empleados como otros posibles administradores, facilitándoles las credenciales de acceso a la herramienta y las indicaciones básicas de su funcionamiento en el momento de alta en el sistema. Cada usuario desde cualquier equipo informático conectado a la red local de LA EMPRESA al que tenga derecho de acceso, tendrá la obligación de conectarse a la herramienta diariamente a la hora de comienzo y fin de su jornada laboral para registrar sus correspondientes horas de inicio y finalización. La herramienta estará disponible ininterrumpidamente para que cualquier empleado pueda consultar la adecuación o desviación de su saldo horario con respecto al calendario laboral establecido por LA EMPRESA, así como para solicitar y consultar posibles incidencias por permisos y licencias que afecten a la duración de las jornadas laborales (vacaciones, enfermedad, etc) La herramienta software sólo será accesible desde cualquier equipo conectado a la red local de LA EMPRESA, y al que cada uno de los empleados tenga permitido el acceso. Se ofrecerá la siguiente funcionalidad:

(1) Usuario perfil “Administrador”

- Registro de presencia

Cada usuario con perfil “Administrador”, al igual que el resto de empleados, podrá registrar la hora de entrada y salida de su jornada laboral diaria.

- Identificación

Cada usuario con perfil “Administrador” se podrá identificar en el sistema con unas credenciales facilitadas por LA EMPRESA y que serán de carácter personal e intransferible, con la posibilidad de cambiar la contraseña de acceso cuando el usuario desee.

- Acceso al panel de control

Cualquier usuario con perfil “Administrador”, tras ser identificado satisfactoriamente, accederá automáticamente al panel de control de la aplicación, donde tendrá disponible las diferentes opciones para operar sobre usuarios.

- Acceso al panel gestión de usuarios

Desde el panel de control anterior, cualquier usuario con perfil “Administrador” tendrá acceso a un conjunto de opciones para operar sobre los usuarios (creación y edición de usuarios, acceso a registros...)

- Alta de usuarios

Cualquier usuario con perfil “Administrador” designado por LA EMPRESA registrará en el sistema a los usuarios, facilitando las credenciales de acceso que serán de carácter personal e intransferible, con la posibilidad de cambiar la contraseña de acceso cada vez que los usuarios lo estimen oportuno.

- Consulta de empleados registrados

Cada usuario con perfil “Administrador” podrá consultar los datos de los empleados registrados, así como los respectivos registros de entrada y salida diarios.

- Consulta y gestión de incidencias o permisos de los empleados

Cada usuario con perfil “Administrador” podrá consultar cualquier incidencia correspondiente a cualquiera de los empleados del sistema.

- Restablecimiento de contraseñas

Cada usuario con perfil “Administrador” podrá restablecer su contraseña y la de cualquiera de los usuarios registrados que lo soliciten.

- Cerrar sesión

Cada usuario con perfil “Administrador” podrá desconectar su sesión.

(2) Usuario perfil “Empleado”

- Registro de presencia

Cada usuario con perfil “Empleado” podrá registrar la hora de entrada y salida de su jornada laboral diaria.

- Identificación

Cada usuario con perfil “Empleado” se podrá identificar en el sistema con unas credenciales de carácter personal e intransferible facilitadas por cualquier usuario administrador en el momento de alta en el sistema. La contraseña de acceso podrá modificarla cada vez que estime oportuno.

- Comunicación de incidencias o permisos

Cada usuario con perfil “Empleado” podrá comunicar cualquier incidencia o permiso que le impida asistir a su puesto de trabajo durante un periodo determinado de tiempo.

- Consulta de incidencias

Cada usuario con perfil “Empleado” podrá consultar sus respectivas incidencias.

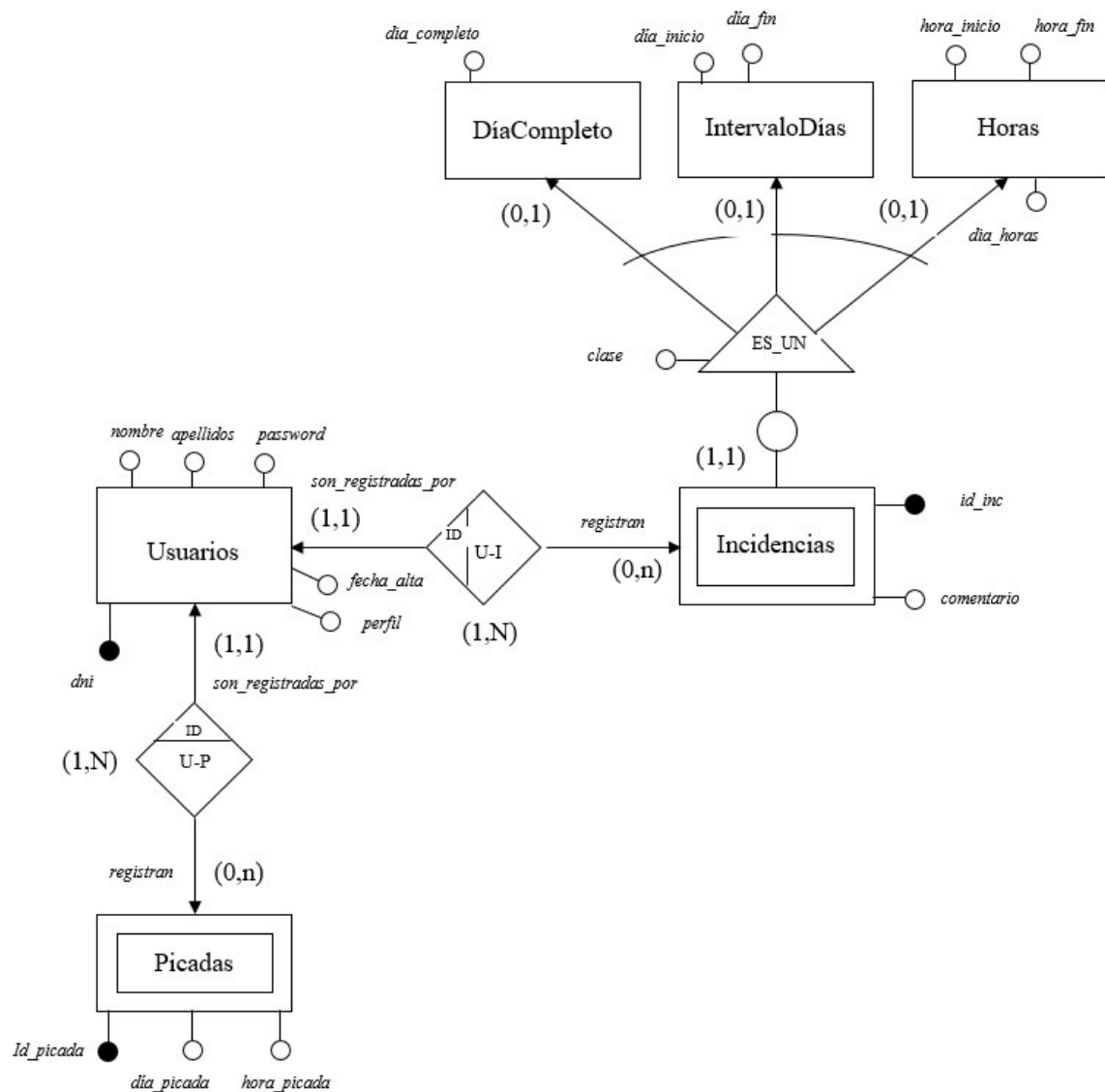
- Cambiar contraseña

Cada usuario con perfil “Empleado” podrá modificar su contraseña de acceso al sistema.

- Cerrar sesión

Cada usuario con perfil “Empleado” podrá desconectar su sesión.

III. Modelo conceptual



IV. Modelo relacional

Según el modelo conceptual anterior a continuación se muestra el modelo relacional:

Usuarios (dni, nombre, apellidos, password, fecha_alta, perfil)

Picadas (id_picada, **dni**, dia_picada, hora_picada)

Para la representación del modelo relacional de las relaciones jerárquicas se ha optado por la aplicación de la regla PRTECAR-4, por lo que desaparecerían del modelo conceptual los subtipos *DiaCompleto*, *IntervaloDias*, y *Horas*, y se transfieren al supertipo *Incidencias* tanto los atributos de los subtipos como el de la interrelación, quedando como sigue:

Incidencias (id_incidencia, comentario, **dni**, clase, dia_completo, dia_inicio, dia_fin, dia_horas, hora_inicio, hora_fin)

V. Descripción detallada de la solución software

Interfaz de inicio: Registro de presencia – Inicio de sesión (figura 1)

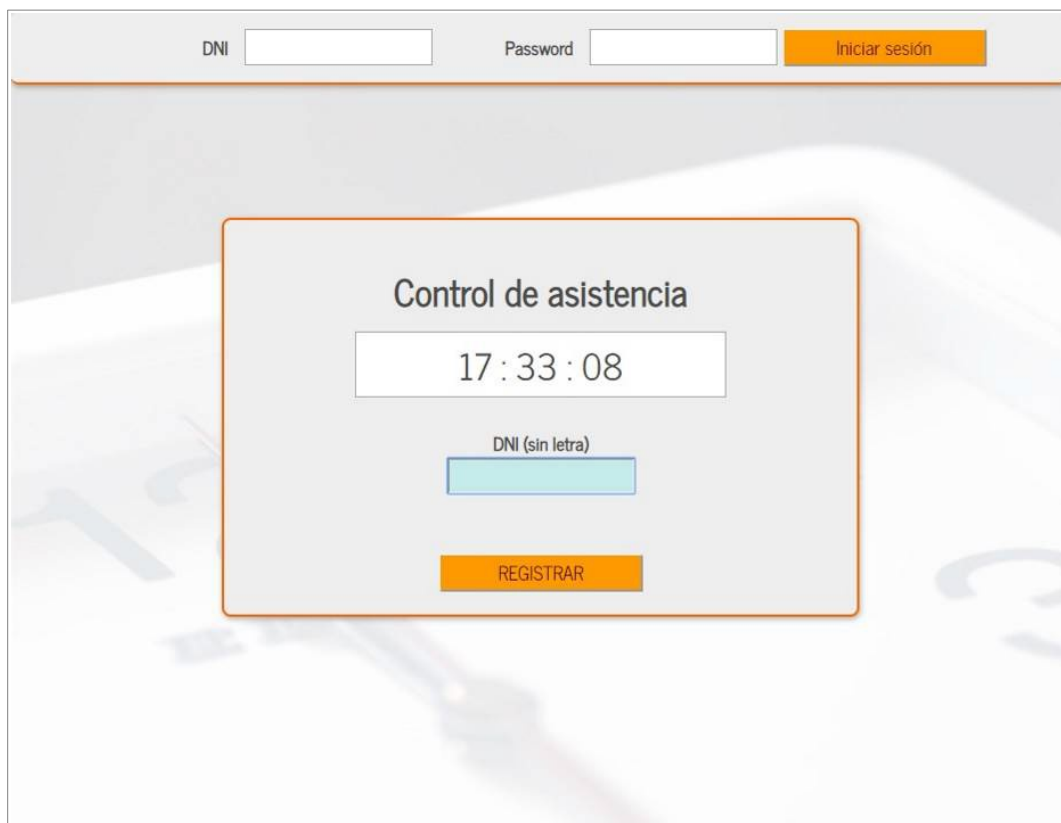


Figura 1: Interfaz de inicio: Registro de presencia – Inicio de sesión

La interfaz anterior es común tanto para los usuarios con perfil *Administrador* como *Empleado*. Como se indica en el apartado anterior, el sistema se entregará con un usuario con perfil *Administrador* habilitado por defecto, con unas credenciales de acceso de carácter personal e intransferible. Inicialmente dicho usuario se identificará en el sistema para la creación del resto de usuarios en el sistema.

1. Interfaz *Administrador*

1.1. Interfaz panel de control *Administrador* (figura 2)

Tras la correcta validación a través de la página de inicio anterior, cualquier usuario administrador accede directamente al panel de control. Desde dicho panel tendrá disponible el acceso a la gestión de los datos disponibles en el sistema sobre el resto de usuarios registrados. Por otra parte, se muestran otras opciones y cuadros informativos comunes a los usuarios con perfil empleado y que se detallarán en apartados posteriores.



DNI	Nombre	Día	Hora
123456789	JUAN CARLOS GARCIA	2023-10-27	10:30
987654321	MARIA LOPEZ	2023-10-27	11:15
456789123	JOSE ANTONIO MARTIN	2023-10-27	12:00
789123456	ANITA GARCIA	2023-10-27	13:45
321654987	CARLOS RODRIGUEZ	2023-10-27	14:30
654321098	ISABEL MARTIN	2023-10-27	15:15
098765432	ANTONIO LOPEZ	2023-10-27	16:00
210987654	JOSEFA GARCIA	2023-10-27	16:45
543210987	JOSE MANUEL MARTIN	2023-10-27	17:30
876543210	ANITA RODRIGUEZ	2023-10-27	18:15
109876543	JOSE ANTONIO GARCIA	2023-10-27	19:00
432109876	MARIA ANTONIA MARTIN	2023-10-27	19:45
765432109	JOSE ANTONIO LOPEZ	2023-10-27	20:30
012345678	ANITA MARTIN	2023-10-27	21:15
345678901	JOSE ANTONIO GARCIA	2023-10-27	22:00
678901234	MARIA ANTONIA MARTIN	2023-10-27	22:45
901234567	JOSE ANTONIO LOPEZ	2023-10-27	23:30
234567890	ANITA MARTIN	2023-10-27	00:15
567890123	JOSE ANTONIO GARCIA	2023-10-27	01:00
890123456	MARIA ANTONIA MARTIN	2023-10-27	01:45
123456789	JOSE ANTONIO LOPEZ	2023-10-27	02:30
456789123	ANITA MARTIN	2023-10-27	03:15
789123456	JOSE ANTONIO GARCIA	2023-10-27	04:00
012345678	MARIA ANTONIA MARTIN	2023-10-27	04:45
345678901	JOSE ANTONIO LOPEZ	2023-10-27	05:30
678901234	ANITA MARTIN	2023-10-27	06:15
901234567	JOSE ANTONIO GARCIA	2023-10-27	07:00
234567890	MARIA ANTONIA MARTIN	2023-10-27	07:45
567890123	JOSE ANTONIO LOPEZ	2023-10-27	08:30
890123456	ANITA MARTIN	2023-10-27	09:15

Figura 2: *Panel de control Administrador*

1.2. Interfaz panel de control *Gestión de usuarios* (figura 3)

Tras pulsar en el botón *Gestión de usuarios* del panel anterior, cualquier usuario con perfil administrador accederá al panel de control *Gestión de usuarios*, que a su vez, mostrará por defecto una tabla ordenada alfabéticamente según el primer apellido con los usuarios

registrados actualmente en el sistema. Los datos que se muestran son: ID (DNI sin letra), apellidos y nombre, fecha de alta en el sistema y perfil (administrador o empleado)

Además se facilita el acceso a la edición o borrado de usuarios y la posibilidad de buscar algún usuario en particular para mostrar su información.

Bienvenido: Usuario Administrador

Volver Cerrar sesión

GESTIÓN DE USUARIOS

Ver usuarios... Gestión de contraseñas... Ver picadas de usuarios Permisos de usuarios

Buscar usuario: (Mostrar todos) Nuevo usuario...

(Listado de usuarios ordenados alfabéticamente)

ID	Nombre	Perfil	
11111111	Administrador, Usuario	Administrador	 
44444444	Gómez Ramírez, Isabel	Empleado	 
55555555	Jaén Paredes, Alicia	Empleado	 
22222222	Paredes Jiménez, Rafael	Empleado	 
33333333	Vázquez de la Rosa, Julián	Empleado	 

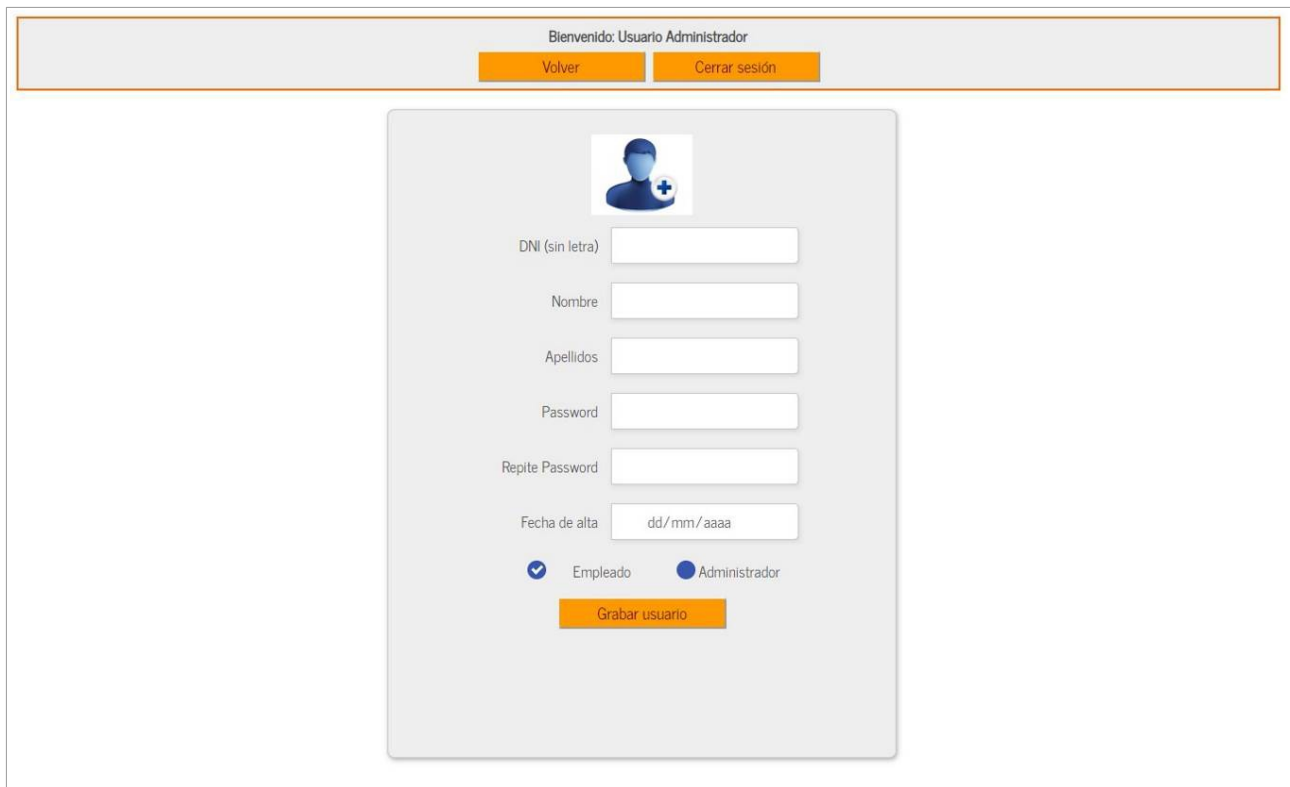
Figura 3: *Panel de control Gestión de usuarios*

A través del panel de control Gestión de usuarios, cualquier usuario administrador del sistema podrá realizar las siguientes acciones:

- Nuevo usuario: se abrirá un formulario para el registro de nuevos usuarios (figura 4).
- Ver usuarios: se muestra el listado ordenado de usuarios registrados. Es la opción por defecto al cargar el panel.
- Gestión de contraseñas: se abrirá un formulario para establecer nueva contraseña a todo aquel usuario que lo solicite, ya sea por haber olvidado su contraseña o por cualquier otro motivo (figura 5).
- Ver picadas de usuarios: se mostrarán los registros de acceso y salida al puesto de trabajo registrados por los usuarios (figura 6).
- Permisos de usuarios: se mostrarán las incidencias y solicitudes de permisos de los usuarios, así como otras opciones indicadas posteriormente (figura 7).

1.3. Interfaz panel de control *Nuevo usuario* (figura 4)

A través del panel de control *Gestión de usuarios* anterior, cualquier usuario con perfil administrador tendrá la opción de dar de alta usuarios en el sistema.



The image shows a web interface for creating a new user. At the top, a grey header bar contains the text "Bienvenido: Usuario Administrador" and two orange buttons: "Volver" and "Cerrar sesión". Below this, a central grey box contains a user profile icon (a person with a plus sign). Under the icon are several input fields: "DNI (sin letra)", "Nombre", "Apellidos", "Password", "Repite Password", and "Fecha de alta" (with a placeholder "dd/mm/aaaa"). Below these fields are two radio buttons: "Empleado" (selected with a blue checkmark) and "Administrador". At the bottom of the form is an orange button labeled "Grabar usuario".

Figura 4: Formulario *Nuevo usuario*

Para el registro de un usuario se deberán completar los siguientes campos obligatorios: DNI sin letra (compuesto por 8 caracteres numéricos, será el identificador de cada usuario para acceder al sistema), nombre, apellidos, contraseña (cadena de caracteres establecida inicialmente por defecto con las reglas de complejidad establecidas por LA EMPRESA y que el usuario podrá modificar en el momento que desee una vez validado en el sistema) fecha de alta, y perfil (empleado o administrador)

1.4. Interfaz panel de control *Gestión de contraseñas* (figura 5)

Al igual que la interfaz anterior, a partir del panel de control *Gestión de usuarios* (figura 3) un usuario administrador tendrá acceso a un formulario para restablecer la contraseña de cualquier empleado.

Bienvenido: Usuario Administrador

Volver
Cerrar sesión

Buscar usuario:

Cantero Delgado, Alicia
(Mostrar todos)
Administrador, Usuario
Cantero Delgado, Alicia
Rodríguez Camacho, Luis
Romero Peralta, María Jesús

Password:
Repite Password:

Cambiar contraseña

Figura 5: Interfaz *Gestión de contraseñas*

1.5. Interfaz panel de control *Ver picadas de usuarios* (figura 6)

A partir del panel de control *Gestión de usuarios* (figura 3) un usuario administrador tendrá acceso al listado de registros de entrada-salida del puesto de trabajo de cualquier empleado, con la opción de filtrar por empleado mediante su selección.

Bienvenido: Usuario Administrador

Volver
Cerrar sesión

GESTIÓN DE USUARIOS

Ver usuarios...
Gestión de contraseñas...
Ver picadas de usuarios
Permisos de usuarios

Buscar usuario:
(Mostrar todos)

DNI	Nombre	Día	Hora	
33333333	Julián Vázquez de la Rosa	28-04-2020	18:15	
11111111	Usuario Administrador	28-04-2020	09:01	
11111111	Usuario Administrador	28-04-2020	14:18	
33333333	Julián Vázquez de la Rosa	28-04-2020	18:14	
22222222	Rafael Paredes Jiménez	27-04-2020	07:59	
11111111	Usuario Administrador	27-04-2020	08:03	
11111111	Usuario Administrador	27-04-2020	14:13	
22222222	Rafael Paredes Jiménez	27-04-2020	14:57	
22222222	Rafael Paredes Jiménez	26-01-2018	08:13	
22222222	Rafael Paredes Jiménez	26-01-2018	14:40	
11111111	Usuario Administrador	25-01-2018	08:00	
11111111	Usuario Administrador	25-01-2018	15:05	

Figura 6: Interfaz *Ver picadas de usuarios*

1.6. Interfaz panel de control *Permisos de usuarios* (figura 7)

Cualquier usuario que se ausente del centro de trabajo durante su jornada laboral deberá registrar una solicitud de incidencia o permiso obligatoriamente como se verá en un apartado posterior. Mediante el botón *Permisos de usuarios* del panel de control *Gestión de usuarios* (figura 3) cualquier usuario administrador tendrá acceso al listado con la información general de dichas incidencias: id de la incidencia, nombre de usuario, descripción del tipo de incidencia y el botón *Ver +* para acceder a la información detallada (figura 8). Contará también con la opción de filtrar la búsqueda por usuario.

Bienvenido: Usuario Administrador

VolverCerrar sesión

GESTIÓN DE USUARIOS

Ver usuarios...Gestión de contraseñas...Ver picadas de usuariosPermisos de usuarios

Buscar usuario: (Mostrar todos) ▼

ID	Empleado	Descripción	
6	Rafael Paredes Jiménez	Reunión comité de empresa	ver +
5	Rafael Paredes Jiménez	Día libre disposición	ver +
4	Usuario Administrador	Vacaciones	ver +
3	Julián Vázquez de la Rosa	Desplazamiento por reconocimiento médico	ver +
2	Rafael Paredes Jiménez	Cuarentena por estado de alarma	ver +
1	Usuario Administrador	Visita médica	ver +

Figura 7: Interfaz *Permisos de usuarios*

Bienvenido: Usuario Administrador

VolverCerrar sesión

ID incidencia	3
ID empleado	33333333
Nombre empleado	Julián Vázquez de la Rosa
Día	25-04-2020
Hora de inicio	10:00
Hora de fin	12:00
Comentario	Desplazamiento por reconocimiento médico

Figura 8: Interfaz *Ver detalle de incidencia*

2. Interfaz Empleado

2.1. Interfaz Registro de picadas (figura 9)

A través de la pantalla de inicio de la aplicación (figura 1) los usuarios dados de alta en el sistema podrán registrar sus accesos y salidas del centro de trabajo. Para ello deberán introducir su identificador en el sistema compuesto por los 8 caracteres numéricos de su DNI (excluyendo la letra). El sistema les devolverá un mensaje de confirmación si se ha procesado con éxito, o de error si ha introducido un identificador no válido o cualquier otro fallo del sistema.

DNI Password Iniciar sesión

Control de asistencia

08 : 07 : 10

DNI (sin letra)

REGISTRAR

Registro de usuario con dni 22 **** 22 a las 08:07

Figura 9: Interfaz Picada de entrada-salida del centro de trabajo

El sistema almacenará cada registro en la tabla correspondiente en la base de datos con la fecha y hora establecidas en el servidor, que a su vez mantiene actualizada dicha información desde los servidores de internet, por lo que, aunque los empleados modificaran dichos ajustes en su equipo local, el sistema se asegura que el registro se almacena con la fecha y hora real.

2.2. Interfaz *Registro de picadas* (figura 10)

Tras la validación de un empleado en la pantalla inicial de la aplicación (figura 1), accederá al panel de control *Empleado* donde se mostrarán los últimos 30 registros o picadas del usuario, así como la opción de filtrar por mes y año. También se pone a disposición del usuario un botón *Mis incidencias* para acceder al registro de sus incidencias, y el botón *Cambiar contraseña* para, si lo desea, modificar el password de acceso al sistema.

DNI	Nombre	Día	Hora
22222222	Rafael Paredes Jiménez	27-04-2020	07:59
22222222	Rafael Paredes Jiménez	27-04-2020	14:57
22222222	Rafael Paredes Jiménez	26-01-2018	08:13
22222222	Rafael Paredes Jiménez	26-01-2018	14:40

Figura 10: Interfaz *Panel de control Empleado*

2.3. Interfaz *Mis incidencias* (figura 11)

A partir del panel anterior el empleado accederá a *Mis incidencias* para ver las incidencias que tiene registradas en el sistema, así como el formulario para crear nuevas incidencias.

ID	Empleado	Descripción	
6	Rafael Paredes Jiménez	Reunión comité de empresa	ver +
5	Rafael Paredes Jiménez	Día libre disposición	ver +
2	Rafael Paredes Jiménez	Cuarentena por estado de alarma	ver +

Figura 11: Interfaz de empleado *Mis incidencias*

Como ya se ha comentado, cuando un empleado se ausente de su centro de trabajo durante su jornada laboral, deberá registrar una incidencia en el sistema. Éstas se clasifican en incidencias de ausencia durante un día completo, durante un periodo de días (vacaciones, enfermedad...) o durante un intervalo de horas en un día concreto (consulta médica, asuntos propios...) Por otro lado, el empleado podrá ver información detallada de sus incidencias mediante el botón *Ver +* como ya se ha explicado anteriormente (*figura 8*).

2.4. Interfaz *Cambiar contraseña* (figura 12)

Mediante el panel control *Empleado* de la figura 10 el usuario podrá modificar su contraseña de acceso al sistema.

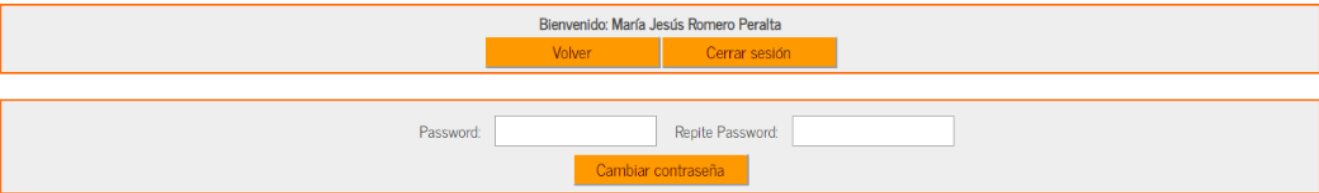


Figura 12: Interfaz *Cambiar contraseña*

VI. phpMyAdmin: administración de base de datos

Como ya se ha detallado, los stacks estudiados en apartados anteriores del presente documento se caracterizan por reunir los componentes necesarios para el desarrollo de una aplicación web completa. Recordamos que dichos componentes eran: intérprete de lenguaje de programación, servidor web y sistema de gestión de base de datos.

Para el proyecto actual se ha usado el stack XAMPP, que comprendía intérprete PHP, servidor web Apache y MariaDB como SGBD. Habitualmente, los stacks suelen incluir una herramienta adicional a modo de panel de control para el administrador del sistema, accesible desde cualquier navegador del equipo local mediante la que se posibilita la opción de manejar y administrar la base de datos del proyecto. En el stack XAMPP dicha herramienta se denomina ***phpMyAdmin***. Desde el navegador se permitirá el acceso a todas las bases de datos de los distintos proyectos locales, así como a cada una de sus tablas, columnas y registros, facilitando la visualización y manipulación de los datos.

La figura 13 muestra la base de datos del proyecto actual mediante un navegador con phpMyAdmin.

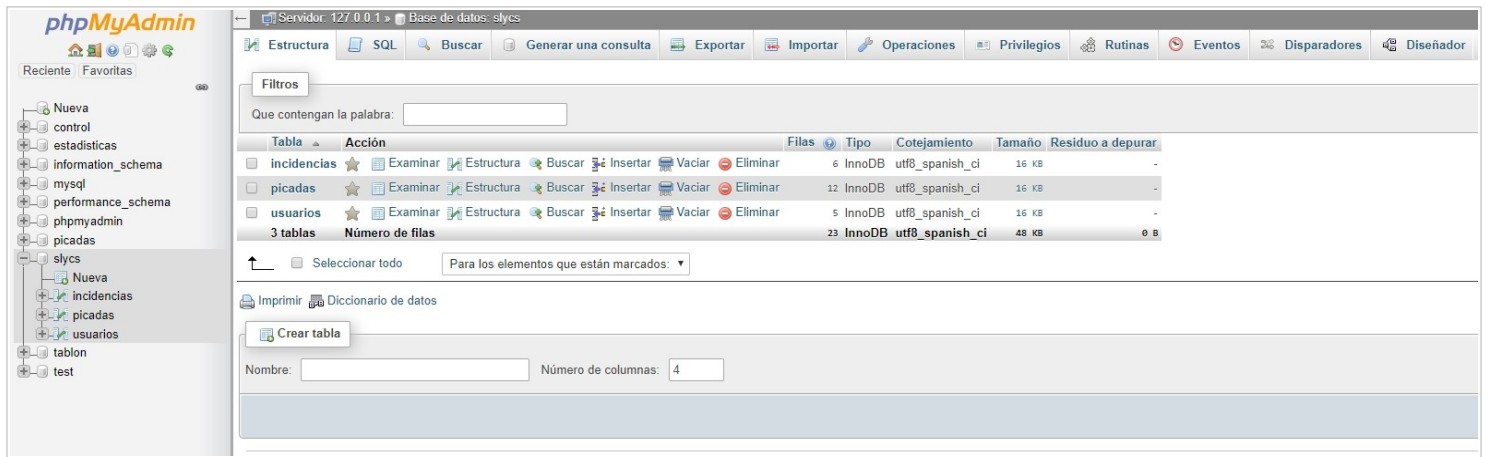


Figura 13: Interfaz *phpMyAdmin*

Entre otras bases de datos de distintos proyectos del equipo local aparece resaltado en gris la base de datos del proyecto actual denominada ***slycs***. En su interfaz inicial, al seleccionarla se muestra el listado de tablas, algunas características y todas las opciones que se pueden ejecutar tanto en la base de datos como en cada una de sus tablas. En el caso actual se comprueba como la base de datos *slycs* consta de tres tablas: *incidencias*, *picadas* y *usuarios*. En ellas se irá añadiendo, editando o eliminando la información requerida durante la ejecución de la aplicación web. De la misma forma, el administrador podrá manipular la información directamente desde la interfaz, lo que constituye una herramienta de gran valor para el desarrollo y mantenimiento de un proyecto.

Por ejemplo la figura 14 muestra la estructura y datos contenidos en la tabla *usuarios* en un momento dado, donde el administrador podrá ejecutar las acciones que desee sobre la estructura o conjunto de datos directamente.

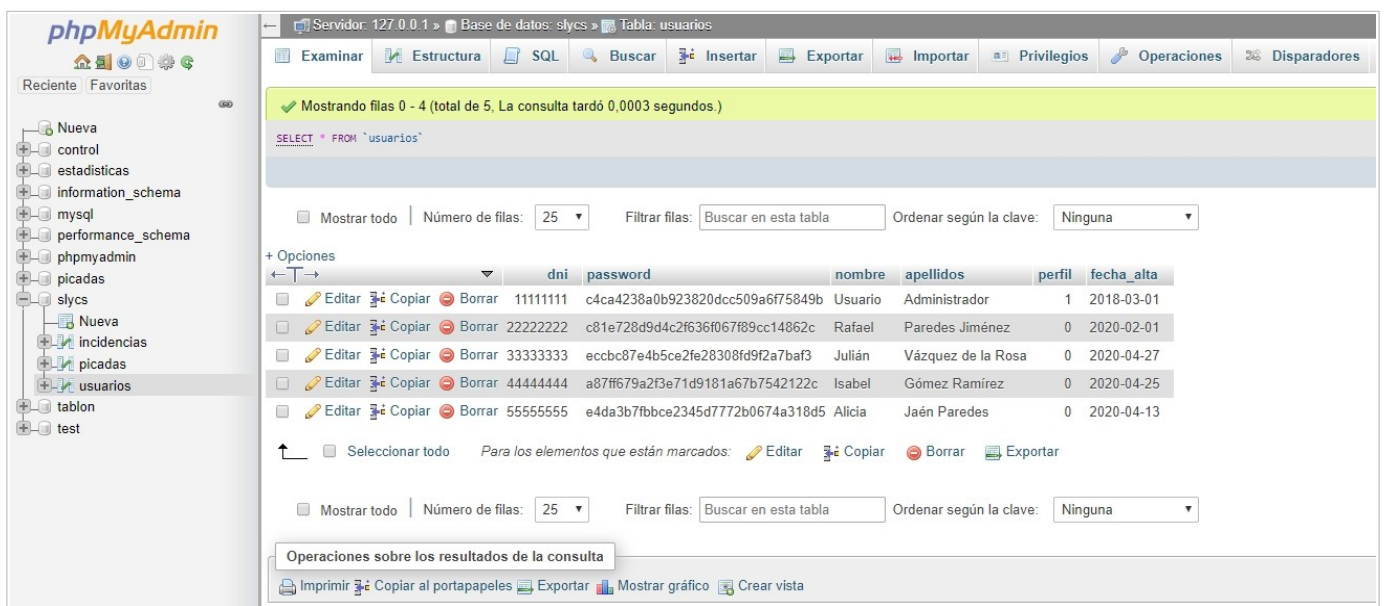


Figura 14: Interfaz *phpMyAdmin*. Tabla *usuarios* de la base de datos *slycs*