

Sistema de control de accesos y Gestión de Recursos

Cristian Vázquez Romero

Resumen—En este proyecto se realizará la programación de un servidor web de manera que sea capaz de identificar y autorizar a un usuario el acceso, mediante un sistema de roles y zonas. El proyecto se desarrolla en varios lenguajes de programación, siendo PHP, SQL y HTML5 los principales a lo largo del sistema, con la implementación de JavaScript para mejorar el diseño gráfico y la interactividad. Tanto la base de datos como la lógica del sistema han sido hechos a mano, así como la integridad referencial de las tablas y el diseño de la aplicación. A continuación se explicarán los objetivos planteados, el desarrollo y la metodología utilizada, así como los resultados obtenidos y las conclusiones.

Palabras clave— Servidor Web, Control de acceso, Gestión de Reservas, Seguridad web.

Abstract— This project achieves the development of a working web server able to identify and authorize user access through a system of roles and areas previously defined. The system is developed within a number of programming languages, including HTML5, PHP and SQL. The full database as well as the system inner logic has been developed from scratch, taking care of the referential integrity within tables and the design of the application. Next on this document we'll have a deeper look on the main objectives, as well as the development of the system, the methodology used and the results obtained.

Index Terms— Web Server, Access control, Resources Management, Web Security.



1 INTRODUCCIÓN

EN este documento se explicará cómo se ha llevado a cabo el proyecto “Sistema de Accesos y Control de Recursos”. Para ello se planteará a continuación tanto la motivación de este proyecto como los objetivos marcados.

A continuación se explicaran los conceptos generales sobre el estado del arte, la metodología utilizada y una descripción de las tareas realizadas del proyecto. Finalmente, una sección de conclusiones donde se expondrán los resultados, seguido de agradecimientos y la bibliografía del proyecto.

2 MOTIVACIÓN Y OBJETIVOS

Vivimos en un mundo rodeado de tecnología, donde todo está conectado a Internet y éste forma parte de nuestra vida cotidiana. Mi principal motivación fue practicar más con tecnologías y lenguajes orientados a Internet a la hora de iniciar este proyecto dado que tiene muchas posibili-

dades en nuestro día a día. Fijándome en el sistema de identificación y acceso de la empresa donde trabajo decidí realizar mi propia versión de un servidor web de control de accesos, añadiendo un módulo de gestión y reserva de recursos.

Las tecnologías orientadas a web o con funciones web tienen muchas aplicaciones utilizando otros medios o plataformas. Desde el uso de escáneres de diversas localizaciones a la implementación de sistemas con dispositivos móviles para identificar al usuario, las posibilidades para hacer un sistema de acceso web e implementarlo con otros medios son muy elevadas, y aumentan con las nuevas tecnologías. De esta manera, este proyecto permite el aprendizaje y refuerzo de lenguajes web conocidos pero útiles, a la vez sirve de práctica para proyectos, individuales o en grupo, basados en Internet.

El principal objetivo de este proyecto es desarrollar un software capaz de replicar el comportamiento de un sistema de control de accesos real, y ser una alternativa real a los demás productos de esas características. Por lo tanto, el software tiene que recoger datos de entrada y devolver la decisión de acceso o rechazo de la petición de entrada.

Por otra parte, este sistema tiene la funcionalidad adicional de, mediante un sistema de zonas definidas, permitir la reserva de esas zonas a horas que estén disponibles. Esta funcionalidad es muy útil en instalaciones donde se

- E-mail de contacto: cristian.vazquez@e-campus.uab.cat
- Mención realizada: Computación
- Trabajo tutorizado por: Marc Talló Sendra
- Curso 2015/16

realizan reuniones diariamente, como una empresa o un centro educativo.

Al margen de los objetivos funcionales, encontramos objetivos secundarios basados en el funcionamiento interno del sistema, por ejemplo una seguridad consistente con la importancia de la aplicación o una base de datos que verifique la integridad referencial de los datos.

A continuación, se exponen brevemente los objetivos principales que se definen para este proyecto en su inicio:

- Mantener un registro completo de la actividad de los usuarios
- Una herramienta en la zona de administración para crear, borrar i modificar usuarios, roles y zonas. Dentro de este apartado entra desde modificar contraseñas de usuario hasta los roles asignados al mismo, así como la relación entre un rol y una zona.
- El usuario tiene un número máximo de intentos para acceder a la aplicación hasta que salte un módulo que inhabilita al usuario y envía un correo al administrador del sistema.
- En una instalación definida con zonas y roles claros, se requiere una alta vigilancia a las intrusiones, es por eso que se registran todas las actividades de los usuarios y se avisa al administrador mediante correos electrónicos en el caso de que un usuario intente en repetidas ocasiones acceder a una zona indebida para su nivel de permisos.
- Mediante el registro de entrada y salida de los usuarios se puede conocer exactamente a la hora a la que entran y salen del edificio, permitiendo llevar el control de los movimientos de cada usuario.
- Mantener la coherencia de la base de datos haciendo especial atención en la integridad de los mismos. A la hora de crear la base de datos, diseñar eficientemente la relación entre las tablas para así poder evitar datos incoherentes que pongan en peligro la coherencia del sistema.
- Mantener los datos seguros, mediante funciones hash y otros recursos de seguridad. Mantener la base de datos segura contra ataques de SQL Injection y otros métodos maliciosos para mantener así la confianza del cliente.

En el punto 6. Resultados de este artículo se analizará si el proyecto ha cumplido todos los requerimientos expuestos en este punto. A continuación, se detalla la situación del mercado mediante un estudio del Estado del Arte.

3 ESTADO DEL ARTE

En el mercado actual se pueden encontrar muchos tipos de sistemas de acceso, dada la versatilidad de la tecnología y los dispositivos que tenemos a nuestro alcance.

Principalmente, se pueden distinguir dos grupos de sistemas, los autónomos y los que forman parte de una red. Los sistemas de control autónomos no están conectados a Internet, lo cual puede ser considerado una gran desventaja, ya que funcionan como unidades independientes del resto. Para modificar una estación de control de acceso autónoma nos tendríamos que desplazar hasta su localización y modificarla manualmente. En el caso de una instalación grande o con muchos terminales/servidores, el coste de actualización o de mantenimiento sería muy elevado, en dinero y en horas. [1]

Por esa razón se decidió utilizar un sistema enfocado al uso web, para poder sincronizar y acceder más cómodamente, a la vez que utilizar las ventajas que ello conlleva. Mediante internet podemos tener los terminales de reconocimiento de usuarios a distancia independiente, siempre que estén conectados y la comunicación sea eficaz.

Es por ese motivo que el segundo grupo es el más amplio dentro de los sistemas de acceso, aprovechando todas las facilidades que la interconectividad representa, se ha convertido en un mercado extenso y lleno de posibilidades. [2][3][4] Desde los sistemas más sencillos a los más complejos y caros, la lógica de actuación es la misma para todos; mediante un terminal se recogen los datos de entrada del usuario y el servidor tiene que contrastar esa información con los datos almacenados en la base de datos, y por último tomar la decisión de acceso del usuario.

Sin embargo, aunque la lógica sea diferente cada vez se experimenta más con los terminales que se utilizan y la forma más segura de asegurar la identidad del usuario, centrándose los estudios en zonas del cuerpo que sean realmente difíciles de replicar, como la retina del ojo humano o el patrón de capilares de un dedo.

En el mercado se han encontrado muchos sistemas de acceso que se decantan por un enfoque claro y determinado, ya sea un escáner de huellas dactilares o un lector de tarjetas magnéticas. La cuestión es que el mercado está enfocado hacia una única solución, en la cual han centrado todos sus esfuerzos y han obtenido resultados magníficos, pero en este proyecto se ha buscado la versatilidad antes que la restricción a un medio. Es por ese motivo que nuestro sistema esta encarado a la obtención de la información del usuario de manera indiferente al medio o terminal utilizado, de manera que se adapte a las exigencias del cliente y de la instalación lo mejor posible.

Por otro lado, no se ha encontrado en el mercado una alternativa a un sistema de accesos que además contenga un módulo de reserva de zonas, aprovechando los datos de usuarios, roles y zonas determinados por el adminis-

trador del sistema. Este módulo permite la organización de salas por parte de los usuarios e incluye opciones como el recordatorio por correo electrónico.

De esta manera se empezó el proyecto con la idea de hacer algo único y unificar herramientas para mejorar la experiencia de usuario y tenerlo todo accesible.

A continuación, se analizará la metodología utilizada en el desarrollo de este proyecto.

4 METODOLOGÍA

A la hora de planificar este proyecto se ha llevado a cabo de manera secuencial, la cual evita colisiones entre tareas dado que es necesario terminar una para empezar la siguiente. En el caso específico de este proyecto, hay apartados que no son correlativos a otros módulos, y se podrían haber implementado en paralelo, pero dado que es un proyecto de un solo participante se han realizado de manera iterativa.

Al inicio del proyecto se escogió como metodología a utilizar el método SCRUM. Éste método no está pensado para proyectos con pocas personas como éste, está diseñado para permitir paralelización de tareas y mantener a todo el equipo enterado de la situación global del proyecto. Sin embargo, es un sistema que me ha dado buenos resultados en proyectos pasados y se decidió utilizarlo de igual manera.

SCRUM es una metodología que se basa en *sprints*, de duración definida al gusto del usuario. En definitiva, un sprint es un periodo de tiempo para el cual te defines unos objetivos para realizar. Al acabar ese periodo se analiza el resultado, si se ha llegado al objetivo y los resultados son satisfactorios. Si todo es correcto se sigue con la planificación al siguiente sprint, pero en caso de retraso se vuelve a planificar el contenido a trabajar para el siguiente sprint. Con sprints de poca duración, se consigue un conocimiento real del estado del proyecto, lo que permite que las tareas no se descuelguen y se obtenga el resultado esperado antes de pasar a la siguiente. De la misma manera, si se encuentran problemas más adelante siempre se puede reorganizar la planificación y volver a mejorar tareas que ya estaban acabadas.

En el caso particular de este proyecto se han realizado sprints semanales, de lunes a domingo. Al inicio del sprint se definían los objetivos y al final de la semana se hacía la evaluación de los resultados y la consiguiente planificación del sprint siguiente.

Esta es la metodología utilizada y se han obtenido resultados muy favorables en cuanto a la gestión del tiempo y las tareas. A continuación, se verán las herramientas utilizadas antes de pasar al desarrollo del proyecto.

4.1 Entorno

Este proyecto se ha realizado en su totalidad en red local, mediante herramientas que permiten el uso del ordenador y el propio router como servidor local, permitiendo todo el desarrollo y las pruebas pertinentes.

He utilizado un programa llamado WampServer que consiste en una suite de herramientas que permiten el desarrollo de una página web en su totalidad.

Los archivos de código y de estilos se guardaban en local y el programa contiene la modalidad phpmyadmin que permite la creación y modificación total de la base de datos mediante menús con opciones o desde código con sentencias.

Esta herramienta ha permitido la creación de todas las tablas y las relaciones entre ellas, con claves externas y constraints para conseguir coherencia en los datos.

Se escogió esta plataforma dado que no viene con plantillas o funcionalidades añadidas, simplemente permite la ejecución de los ficheros y recrear el sistema. Eso quiere decir que en este proyecto toda la codificación se ha hecho desde cero, empezando por el código HTML y acabando por el PHP, pasando por los estilos en CSS, como veremos a continuación en el siguiente apartado.

El motivo por el cual se hizo esta decisión fue comprender y practicar con todos los lenguajes hasta el punto de conocerlos y saber manejarlos. Creo que la única forma de realmente entender un lenguaje es solucionar los problemas que te van surgiendo desde cero, aunque sea más costoso.

Seguidamente, veremos esos lenguajes de programación que constituyen el sistema de acceso web.

4.2 Lenguajes de Programación

Para este proyecto se tuvo total libertad para escoger entornos, lenguajes y demás opciones. En el caso anterior, se escogió la plataforma expuesta porque no traía código ni ayudas de serie, y era la forma más realista de empezar un proyecto desde cero. En el caso que nos ocupa, se escogieron los lenguajes necesarios para poder obtener el resultado esperado acorde con unos estándares de funcionamiento y estética.

Los lenguajes de programación web utilizados en este proyecto son cinco:

- HTML: es el lenguaje más básico y ha constituido toda la estructura de la página web. [5]
 - CSS: se complementa con HTML para otorgar estilos a las estructuras que crea. Permite dar vida a una a una página web mediante sus estilos, pero como veremos más adelante no es la única responsable de como luce la interfaz web. [6]
 - PHP: Es el lenguaje más utilizado en el sistema, ya que se encarga de toda la lógica interna y de relacionar la base de datos con el código HTML. [7]
- El código PHP está dispuesto de manera que cumple con el sistema Modelo - Vista - Controlador [8], de manera que separa el código PHP de la vista, haciendo independiente la lógica y la programación pura del estilo visual. Esta medida conlleva un gran número de ficheros, la mayoría pequeños, pero aumenta la independencia del sistema y la separación entre código y estilo.
- SQL: El lenguaje orientado a base de datos se

compenetra a la perfección con PHP, siendo eficientes a la hora de realizar consultas y obtener los datos del sistema. En SQL también se han creado todas las tablas y las relaciones entre ellas, mediante la herramienta phpmyadmin de la suite antes comentada.

- JavaScript: Este lenguaje de código cliente lo he utilizado para mejorar las funcionalidades y la apariencia del sistema. Se podría programar la plataforma sin este lenguaje pero perdería interactividad y agilidad. Utilizo scripts sencillos creados de cero, la herramienta Bootstrap [9] para facilitar algunos estilos y un plugin de calendario que permite seleccionar un día específico, además de definir días festivos que no estén disponibles. [10]

El sistema ha sido programado en un notepad++, que al igual que las herramientas anteriores no proporciona plantillas ni ayudas de código. Por último, la web se ha programado tomando como referencia Google Chrome como método de uso y visualización, ya que entre navegadores a veces difieren algunos aspectos.

5 DESARROLLO DEL PROYECTO

El proyecto ha pasado por todas las fases necesarias para su desarrollo, desde la planificación al test completo de la aplicación. A continuación se profundizará en cada uno de los pasos:

5.1 Planificación

La planificación del proyecto es donde se dejan definidos los objetivos que ha de cumplir el proyecto, así como las restricciones y la organización en el tiempo de todas las tareas. Es importante tener en cuenta posibles retrasos que influyan a todo el proyecto, dejando tiempo de holgura entre tareas.

La concepción de este sistema se produjo por las ganas de recrear un sistema familiar, con herramientas que ya había podido conocer pero no profundizar, y de esta manera conocer mejor un entorno con potencial para desarrollar.

En esta etapa se hizo un brainstorming y se planificaron las tareas y su secuencialidad. La lista a continuación muestra las tareas en el orden en el que fueron ejecutadas:

- Crear entorno y servidor web activo
 - o Representa la instalación y preparación del entorno y la base de datos, con la creación de un servidor sencillo al cual poder hacerle consultas y obtener la información necesaria.
- Diseño de la base de datos
 - o Para poder empezar a desarrollar nos hacía falta introducir usuarios activos y definir roles y zonas acorde con los permisos

que necesitemos. Para esto fue necesario crear una jerarquía de usuarios, roles y zonas de la que se hablará en más profundidad más adelante.

- Programación de las funciones del servidor
 - o Consiste en la función de acceder a una instalación y comprobar las credenciales proporcionadas. También contiene la visualización paginada del registro de actividad con todos los accesos, y el desarrollo del módulo de reserva de zonas.
- Programación de las funciones de gestión de la base de datos
 - o Se basa en la creación de las herramientas necesarias para poder modificar tanto usuarios como roles y permisos, desde el propio servidor y no tener que acceder a la herramienta externa de WampServer para hacer los cambios. También permite ver la página principal de un usuario o una zona.
- Mejorar la interfaz gráfica
 - o Las funcionalidades de la página fueron realizadas con php y el resultado era una página lenta y poco interactiva, que requería cargas constantes para simples tareas. Además de para mejorar el aspecto visual, se introdujo JavaScript para mejorar la fluidez y obtener una página *responsive* al usuario.
- Test completo
 - o A pesar de que se realizaban pruebas muy continuas para asegurar que lo que se dejaba atrás era funcional, se planificó una prueba general para comprobar el rendimiento global de la aplicación.

En el siguiente apartado veremos una breve extracción económica del proyecto.

5.2 Análisis Económico

Un proyecto como éste requiere de diferentes skills que necesitan un grupo de personas con conocimientos y habilidades diferentes. En este aspecto, el grupo de trabajo que forma este proyecto necesita de los roles de jefe de proyecto, analista, diseñador web, programador y tester.

El perfil analista será el que estudiará los objetivos y los requerimientos y realizará el diseño de la base de datos y la lógica del sistema. El rol de diseñador web será el que conciba el aspecto visual de la página y se encargará de su realización ya que, mediante la división código - aspecto que proporciona el Modelo - Vista - Controlador, el perfil programador puede avanzar en la lógica del sistema.

Por último, el tester será el que realice un estudio ex-

haustivo del sistema para asegurar su correcto funcionamiento. Todos estos apartados serán supervisados por el rol de Jefe de Proyecto, que estará informado de todos los avances y participará en mayor o menor medida dependiendo de la tarea,

Estudiando la planificación inicial del proyecto, podemos repartir las tareas en roles y calcular las horas necesarias para cada perfil, como podemos observar en la tabla siguiente:

| Perfil de trabajo | Horas Necesarias |
|-------------------|---------------------|
| Analista | 10 días = 80 horas |
| Diseñador | 20 días = 140 horas |
| Programador | 50 días = 400 horas |
| Tester | 7 días = 56 horas |
| Jefe de Proyecto | 70 días = 560 horas |

Estimamos que el jefe de Proyecto estará presente en todas las fases del desarrollo para ayudar con su visión y asegurar que los objetivos iniciales se cumplen finalmente. Se considera que el diseñador y el programador pueden empezar a la vez y así disminuimos el tiempo de desarrollo, con la idea de disminuir también costes asociados.

El sueldo de analista es de 12€ la hora, lo cual hace que su trabajo resulte en una remuneración de 960€ para este proyecto. Un diseñador gráfico cobra un sueldo de 10€ la hora, lo cual suma un total de 1400€. El sueldo de un programador oscila alrededor de los 8€ la hora, por lo tanto su trabajo en este proyecto se traduce en 3200€, repartidos en dos meses que dura su función.

El tester cobra 7€ la hora, por lo tanto su remuneración correspondiente por un trabajo de 56 horas son 392€. Por último, el Jefe de Proyecto cobra 14€ la hora y realiza un trabajo extenso a lo largo de todo el proyecto, por lo que el total de su remuneración es de 7840€.

Por lo tanto, el total de costes en horas de trabajo para el departamento es de 13792€.

A nivel de instalaciones i material, el alquiler de una instalación para trabajar es de 300€ al mes, con 50€ adicionales de electricidad, telefonía y Internet. Para los cálculos, se estima que el proyecto dura 3 meses de desarrollo. Necesitaremos 5 ordenadores de gama media/alta para poder programar la aplicación, diseñar gráficos y almacenar toda la información necesaria, además de 3 servidores para la primera fase del despliegue online de nuestra aplicación.

Los 5 ordenadores dedicados exclusivamente para programar valen 650€ cada uno, más los 3 servidores que valen 600€ por unidad, hacen un total de 5050. Si calculamos su coste de amortización, la vida útil de estos aparatos de alto uso se encuentra aproximadamente en los 4 años, con lo que el primer año se puede derivar que cada año pagamos una cuarta parte de su precio, así que de los 5050€ totales pagamos el primer año 1265,5€.

Por lo tanto, si sumamos el coste del trabajo del departamento, la instalación y el material amortizado, nos suma un total a pagar el primer año de 10155,5€.

Ese es el precio estimado necesario para desarrollar un

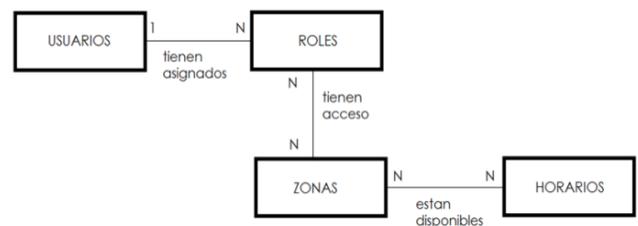
proyecto de estas dimensiones. No necesitamos adquirir software dado que todos los programas que utilizamos son de licencia libre y no es necesario ningún pago por su uso legal.

A continuación, entraremos en detalle en las diferentes etapas de desarrollo centrándonos en los momentos clave.

5.3 Diseño de la base de datos

Una vez se recreó el entorno del servidor con las herramientas antes especificadas, antes de poder introducir los datos de los usuarios fue necesario diseñar las principales características que tendría nuestro sistema. ¿Cómo determinar que un usuario tiene acceso a una zona? La opción más efectiva se basa en el uso de roles, que reflejan un nivel en la jerarquía, y que pueden ser aplicables a los usuarios que se deseen. De esta manera, en lugar de, por ejemplo, definir tablas para cada zona con los usuarios que tienen acceso, se crea una tabla con zonas, otra con roles y una tercera tabla que relaciona los identificadores de las dos. De manera que si un rol tiene relación con esa zona, todos los usuarios con ese rol asignado tendrán acceso. Este sistema hace el mantenimiento de usuarios y roles muy sencillo, ya que permite modificar los permisos de un rol para modificar a un gran grupo de usuarios con una sola acción.

Con esos tres elementos podemos diseñar toda la lógica de la base de datos. A la hora de validar el acceso a un usuario, basta con descubrir su rol en el sistema y comparar con la tabla si ese rol se relaciona con la zona deseada.



Las tablas principales del esquema son la tabla de Usuarios, la tabla de Roles, la tabla de Zonas, la tabla de Horarios y la tabla de Relación entre Roles y Zonas.

Primeramente, un usuario es asignado un rol en su creación, que se puede modificar en cualquier momento, y le permitirá acceso a zonas específicas.

Las zonas están definidas en su tabla correspondiente y se identifican por una ID que las hace cada registro de la tabla inequívoco. Las tablas contienen información relevante sobre el espacio al que representan y se relacionan con la tabla de horarios, donde podemos añadir o eliminar horarios de disponibilidad para esas zonas.

De esta manera, en la tabla Horarios tendremos registros con el identificador de la zona y el horario deseado, creando fácilmente los horarios de las zonas.

Por último, los roles y las zonas se relacionan mediante una tercera tabla que contiene registros con los identificadores de ambas entidades. Por lo tanto, la tabla de rela-

ciones tendrá un registro con el identificador del rol R1 y la zona Z1, permitiendo el acceso a esa zona por parte de todos los usuarios con ese rol asignado.

Por otra parte, se ha diseñado la base de datos de manera relacional, donde las tablas con información relevante se relacionan con otras que las complementan, mediante llaves externas. De esta manera, eliminar un usuario elimina sus reservas y se mantiene la coherencia a lo largo de toda la base.

Otra característica en este sistema es el uso de algunas tablas de log, que se contienen el registro completo de los accesos a la plataforma y los intentos de acceso a las zonas del sistema. Estas tablas sirven para monitorizar la actividad total de la aplicación y permiten controlar si un usuario intenta acceder a zonas que no le pertenecen y así tomar las acciones pertinentes.

Por último está el uso de algunas tablas de lookup, que contienen información para liberar de columnas otras tablas. Por ejemplo, una tabla tendrá un campo 'Acceso' que será un binario, y la tabla de lookup tendrá la conversión a un lenguaje más agradable, donde 1 puede ser "Aprobado" y 0 puede ser "Denegado". Esto ayuda a liberar las tablas de peso y aumenta la rapidez del sistema en insertar y eliminar registros.

5.4 Programación de las funciones del servidor

Un servidor de estas características necesita tener muchas funcionalidades y en este trabajo se han separado en el apartado que nos ocupa y el que vendrá a continuación. Este apartado verá las funcionalidades que puede realizar con el usuario, mientras que el siguiente verá los mecanismos de mantenimiento incorporados en el sistema a la hora de modificar roles, usuarios, etc.

Ahora veremos las dos principales funciones que ejecutan el sistema y la seguridad que las rodea.

En primer lugar está el acceso a la propia aplicación y el control de accesos a otras zonas.

Esta aplicación está diseñada para el uso tanto de administradores como de usuarios de la instalación, y por tanto ambos grupos pueden realizar acciones diferentes.

Para iniciar la aplicación se ha de identificar y el programa te clasifica entre usuarios con permisos administrador o usuario por defecto. Un usuario por defecto verá al conectarse a la aplicación que puede reservar una zona mediante la aplicación 'Calendario', puede acceder al 'Panel de Control' donde encontrará información sobre los usuarios del sistema y las zonas del mismo. Haciendo click en una zona podrá ver una breve descripción, acompañada de una imagen y datos sobre la misma. En la zona inferior de la pantalla podrá ver los horarios de reserva de la zona y las reservas ya efectuadas en un plazo de diez días hábiles. También puede ver la lista de usuarios y un breve perfil corporativo.

Por último, puede acceder al apartado 'Mis Reservas', donde puede ver que zonas tiene reservadas y con qué fecha, y en caso de que sea necesario puede eliminar di-

cha reserva.

De modo adicional, la plataforma incluye un módulo que permite comprobar si el usuario puede acceder a una zona seleccionada, funcionando como una representación del funcionamiento del servidor ante el usuario accediendo a una zona. Este funcionamiento se puede extrapolar a como actuaría ante una instalación real y usuarios moviéndose libremente por las múltiples zonas del complejo.

Sin embargo, menos la funcionalidad de eliminar las reservas propias del usuario, éste no tiene poder de modificar ni crear nada en la aplicación más que nuevas reservas. Ese poder les corresponde a los usuarios administrador.

Un usuario con privilegios de administrador al conectarse a la aplicación verá principalmente las mismas opciones que el usuario por defecto, pero con sus capacidades ampliadas. Las reservas continúan igual pero el panel de control cambia y se amplía drásticamente. Este aspecto del sistema lo veremos en más profundidad en el punto siguiente, donde se explican las mecánicas de seguridad y mantenimiento del sistema.

Así pues, las dos funciones más importantes son el acceso y la reserva. El acceso es un sencillo formulario con el campo de usuario y la contraseña para permitir el acceso a la aplicación. Como ya hemos comentado antes, se realiza una búsqueda de esos datos, con el sistema siendo capaz de responder que no existe el usuario especificado, o que la contraseña no es la correcta para ese perfil. Como veremos más adelante, en el sistema tenemos la opción de deshabilitar usuarios para mantenerlos inactivos sin tener que eliminar todos sus datos por si regresan en un futuro; nuestro código está diseñado para buscar solamente usuarios habilitados así que devolvería fallo de autenticación.

El calendario de reservas permite escoger un día de la semana y una zona a la que queremos acceder, y devuelve las horas disponibles para reservar la zona. Solamente permitimos que el usuario pueda seleccionar zonas a las que tiene acceso, así evitamos que se intenten reservar zonas no disponibles para el usuario o posibles errores de ese tipo.

El calendario está realizado mediante un plugin de jquery, basado principalmente en JavaScript, que se llama DatePicker. Es un plugin que dado un input te muestra un calendario al lado para escoger dinámicamente el día que se desea reservar una zona. Se escogió este plugin dado que el diseño era perfecto para la aplicación y viene con muchas opciones de modificación. Permite cambiar el orden de los días y el idioma con el que se muestran los meses y días, pero el mejor aspecto es que permite introducir los días que son festivos y por tanto no se puede reservar nada. Esto se consigue con una tabla en la base de datos que consiste de los días que no están disponibles para el usuario. La aplicación antes de mostrar los días, comprueba si ese día es seleccionable mediante una función creada desde cero que comprueba si ese día existe en la tabla mencionada anteriormente. De la misma manera, está configurado también para no mostrar disponibles los días del fin de semana, dado que lo más habitual en una

instalación de estas características, orientadas a un uso académico o empresarial, es no ser utilizados en festivos.

Con lo que hemos explicado, nos ahorramos tener que mantener una base de datos con todos los días del año, o calcular mediante bucles y condicionales los días de cada mes y si son festivos o laborables.

El único punto negativo es que se depende de software de terceras personas pero parece muy estable y por el momento no ha dado ningún problema. Este es uno de los principales puntos donde se incorporó JavaScript para mejorar la respuesta de la página y hacerla más accesible y rápida ante el usuario.

Seguidamente, se expondrán las características de las funciones de mantenimiento y seguridad del sistema.

5.5 Programación de las funciones de gestión y mantenimiento

Como ya hemos comentado antes, la segunda parte de las funcionalidades del servidor se centran en poner actualizar, modificar y mantener la base de datos de forma autónoma, sin la ayuda del programa desarrollador utilizado al crear todo el sistema. De esta forma, el poder de modificar recae completamente en los usuarios con permisos de administrador definidos en la base de datos.

La piedra angular de la gestión de este proyecto se basa en el Panel de Control. Como ya se ha avanzado con anterioridad, un usuario administrador puede utilizar más herramientas que están escondidas para el resto de usuarios.

Dentro del Panel de Control, puede crear y modificar usuarios y zonas, apartados que ya veían los otros usuarios pero sin poder hacer cambios. Un usuario administrador puede entrar por aquí al perfil de un usuario y cambiarle la contraseña, el rol en la empresa o hasta deshabilitarlo. De la misma manera puede crear una nueva zona y modificar las ya existentes.

La novedad es que ahora el administrador puede ver las pestañas de roles y de calendario dentro de este panel.

Ahora el administrador puede ver los roles, y al acceder a ellos se encuentra con las cuatro opciones básicas de customización de roles. Las más importantes son las que permiten modificar su relación con las zonas, con dos opciones que permiten añadir o eliminar relaciones con otras zonas y así permitiendo o denegando el acceso a muchos usuarios. Seguidamente, se encuentra la opción de modificar el nombre de una zona y por último de eliminarla. Eliminar una zona conlleva eliminar reservas y relaciones con otros roles, como se ha comentado anteriormente, y es donde entra en escena la base relacional creada que mantiene la coherencia entre los datos.

La segunda opción nueva en el panel es el calendario, que incluye las opciones necesarias para mantener actualizado el mecanismo de reserva de zonas por parte de los usuarios.

Primeramente, el administrador tiene la opción de añadir o eliminar las fechas que para el calendario laboral de la empresa sean festivos. Por ejemplo, el 6 de enero, el día de la Constitución y demás fechas festivas a nivel

nacional o local se pueden introducir fácilmente, así como días festivos que tenga la empresa u organización a nivel privado. De la misma manera, se pueden eliminar datos anteriormente introducidos y así hacer disponible un día deshabilitado.

Seguidamente nos encontramos con la posibilidad de eliminar una reserva, de cualquier usuario. Para ello solo hemos de seguir los pasos e primeramente introducir el nombre del usuario que hace la reserva y la fecha del día que se efectúa dicha reserva. Mediante JavaScript aparece un segundo formulario donde se especifican las zonas y horas que tiene reservadas ese usuario para las fechas especificadas, y permite la eliminación una a la vez de las reservas que se necesiten. Remarcar que estas acciones solo pueden ser realizadas por el administrador porque si no los usuarios podrían borrar reservas ajenas para conseguir las suyas y no sería una herramienta útil.

Una de las tareas que se plantearon como posibles mejoras debido al tiempo disponible para realizar este proyecto, es la recopilación de estadísticas y la posibilidad de mostrarlas por pantalla en forma de gráficas. Esta tarea se ha podido llevar a cabo en la recta final del proyecto y el resultado es muy positivo, ya que es una forma muy visual de entender el estado del sistema y poder analizar y sacar tendencias sobre los usuarios y su uso del sistema.

A nivel de seguridad, toda la aplicación intenta ser fiable y no tener brechas que permitan a usuarios básicos obtener permisos de administrador.

En ese aspecto, se podría argumentar que el primer login de la aplicación es el más importante, ya que te da acceso al índice de la página con las opciones de administrador. Para ello, hacemos HASH de las contraseñas de los usuarios para dificultar su obtención o desciframiento. Las contraseñas se modifican a un código HASH antes de ser introducidas a la base de datos y al compararlas comparamos cadenas de números y letras más complejas y aleatorias que una sencilla contraseña. [11]

Cada página de la web comprueba mediante las variables de sesión que el usuario esté identificado y sus permisos sean los correctos, de manera que si ocurre un bypass de la página de inicio cada una de las otras páginas comprueba que se tiene acceso a esa zona de administración.

La segunda medida de seguridad es evitar ataques como el SQL Injection, mediante “escapar” las cadenas para evitar la entrada de símbolos especiales que puedan alterar el funcionamiento del sistema. Para reducir el número de casos en los que puede entrar información externa, se han modificado muchos formularios que pedían entrada de datos mediante menús desplegados con las opciones especificadas que limitan en gran medida posibles brechas de seguridad. Además de para evitar brechas, si un usuario quiere reservar una zona, se le ofrece un desplegable solo con las opciones que tiene disponibles, sin zonas que no pueda acceder, para evitar también errores.

Con estas especificaciones sobre la seguridad del sistema, pasamos al siguiente apartado que se centra en cómo se mejoró el estilo visual de la página.

5.6 Mejorar el diseño visual de la aplicación

Llegados a este punto la aplicación era funcional pero se veía limitada por un amplio uso del php que requerían la carga continua de formularios para obtener un resultado. Además, el estilo visual no se había mejorado desde el inicio de la plataforma y era el momento de ponerlo al día.

Se decidió utilizar un código externo de JavaScript llamado Bootstrap, que es un conjunto de código CSS y JavaScript que permite otorgar estilos rápidamente utilizando estilos predefinidos. De esta manera, se mejoró primeramente el aspecto general de la página, dado que este módulo permite centrar mejor todo el contenido en el centro de la pantalla y dar de estilos sencillos títulos, accesos a otras páginas y demás. Donde se hizo el cambio más significativo fue en los formularios, donde Bootstrap viene con formularios en línea o en columna por defecto, y son muy sencillos de modificar en el acto. De esta manera, todos los formularios pasaron a seguir la misma tendencia estética y encajar perfectamente en el diseño global.

Si bien eso fue un gran cambio, el uso de JavaScript en la aplicación mejoró considerablemente la experiencia de uso. Anteriormente, para hacer una reserva o identificarte, había una página en php en la que completabas el formulario y al enviar se cargaba una página que te informaba del resultado, y en caso de error tenías que volver a la página anterior y volver a completar el formulario.

Cuando un sistema como éste cuenta con muchos formularios se puede hacer tediosa su navegación con tantas cargas, por eso pasamos a hacer la mayoría de POST de formularios mediante JavaScript, que permite la carga dinámica de áreas de la página sin tener que recargarla por completo. De esta manera aumentamos la fluidez y limitamos el número de ficheros de cambio, aunque también nos aumentan el número de ficheros de vista que tenemos ya que nos hacen falta al modificar dinámicamente las divisiones de la página con los resultados del formulario.

Por ejemplo, si una zona seleccionada no tiene horas disponibles para las siguientes dos semanas, la aplicación invita a cambiar la fecha y volver a probar su disponibilidad sin tener que cargar una página de error y volver a la que se estaba. Con este cambio, la página te responde en el momento cargando un mensaje de error o un mensaje



de éxito en la misma página sin tener que hacer nada más, pudiendo modificar la fecha en ese mismo momento para hacer otra reserva. Esto se nota durante el test de la aplicación y su uso repetido, ya que ahí es donde se nota el grado de fluidez conseguido.

Podemos observar en la imagen anterior el aspecto de la página para los usuarios no administrativos. Esta página solo se muestra una vez identificado en la aplicación con credenciales válidas, sino no permite el acceso y puede bloquear el usuario. Por otra parte, si nos conectamos como usuario administrador tendremos la opción adicional de ver los accesos a la página y otras funciones en el panel de control.

En el siguiente apartado nos centraremos en el test realizado a la aplicación.

5.7 Test completo

Al verificar que un software funciona, no basta con probar los casos más sencillos y seguir adelante. Se tiene que comprobar y buscar todos los casos posibles para ver cómo reacciona el sistema ante ellos.

La plataforma no es tan compleja como para hacer test de unidad de todos los apartados, pero se pueden hacer en determinadas zonas y es imprescindible para asegurar su funcionamiento.

En el desarrollo de este proyecto se ha intentado validar en profundidad cada una de las nuevas funcionalidades que se le añadían al sistema, con la idea de llegar al final con un proyecto sin fisuras. Esto ha permitido ir conociendo la aplicación de antemano y mejorar la técnica de programación a medida que se desarrollaba. Se han descubierto fallos en el diseño que de haber continuado sin detectar luego habrían sido más complicados de modificar, ya que afectarían a más partes del sistema.

6. RESULTADOS

El resultado de este proyecto es un software de control de accesos y gestión de recursos, con un servidor en php con núcleo en una base de datos relacional y presentada mediante páginas web.

El sistema tiene una interfaz gráfica sencilla y un funcionamiento eficiente. Se ha creado un código sólido y seguro que permite la modificación de los elementos del sistema de forma autónoma.

A continuación vamos a repasar los objetivos marcados en busca de conocer se ha conseguido completarlos:

Mantener un registro completo de actividad: Se cumple y se proporciona un registro total de los accesos por parte de los usuarios. Se puede filtrar por accesos fallidos o con éxito, también se puede filtrar por usuario y ordenar por fecha.

Un sistema de administración de usuarios, roles y

zonas: Como se ha comentado con anterioridad, el sistema es independiente de la herramienta con la que se ha desarrollado y permite la modificación, creación y eliminación de usuarios, roles y zonas, así como la relación entre ellos.

Los usuarios tienen intentos máximos para identificarse: Como ya hemos comentado, los usuarios se identifican en las zonas que quieren acceder y si son denegados esa información se almacena en la base de datos. Tenemos procesos que avisan cuando un usuario intenta acceder a zonas indebidas y avisa al administrador de este suceso, pudiendo deshabilitarlos si es necesario.

Control del acceso de los usuarios: Mediante nuestra aplicación se puede conocer los accesos de un usuario con total precisión, e incorpora un sistema de alerta cuando el usuario es denegado un máximo de veces, lo cual envía un correo electrónico a los usuarios administradores.

Mantener la coherencia de los datos: Se ha recreado una base de datos referencial con el esfuerzo en hacerla coherente en todo momento. Las tablas están interconectadas mediante relaciones de claves externas y claves primarias y se consigue mantener su integridad en todo momento.

Mantener los datos seguros: Como ya hemos comentado en este proyecto la seguridad en un producto como éste es importante, por eso utilizamos contraseñas hash y se tiene especial cuidado con la introducción de datos del usuario para evitar el SQL Injection.

Una vez hemos comprobado el grado de resolución de los objetivos, pasemos a las conclusiones finales.

7. CONCLUSIONES

Una vez finalizado el proyecto, se puede comprobar que se han cumplido los objetivos marcados desde el inicio. La planificación se ha podido seguir a lo largo del proyecto, efectuando pequeños cambios debido a retrasos ocasionales por problemas relacionados con el código de la aplicación. A la hora de desarrollar este proyecto ha sido un reto combinar todos los lenguajes que se utilizan porque el punto de partida era un conocimiento muy básico de todos ellos.

Cabe decir que la decisión de realizar todo el desarrollo desde cero me ha permitido conocer las bases de los lenguajes y esforzarme desde el inicio por controlarlos, pero se ha empleado mucho tiempo en estilos y en el Modelo Vista Controlador, que permite una separación clara de lógica y estilo, pero hace necesarios muchos ficheros intermedios y tiempo para conectarlos. Hay frameworks que te permiten crear controladores con pocos clicks y ayudan a un desarrollo más accesible, pero estoy contento con la decisión tomada ya que me ha permitido empezar desde cero con el desarrollo.

Por otra parte, he descubierto haciendo este proyecto

que podría mejorar el desarrollo del software con el uso de un repositorio de versiones como Github. Pienso que podría haber funcionado muy bien para mantener un control de las tareas, versiones y demás. Si hubiera formado parte de un proyecto con más personas, o más complejo, lo recomendaría sin duda.

Por último, en este breve inciso de aspectos sobre el proyecto, es importante destacar que desde un principio se pensó realizar una aplicación para móviles de manera complementaria, y así extender mis conocimientos y las funcionalidades de este proyecto, pero por motivos de tiempo al final no pudo ser posible, aunque yo lo guardo como aspectos que podría mejorar con más tiempo y como algo que me interesaría desarrollar en mi tiempo libre.

Habiendo trabajado en este proyecto, es mi opinión que puede derivar y formar parte de un sistema mayor, con otras tecnologías adheridas. Por ejemplo, para la presentación se ha preparado un complemento que se esta tecnología que utiliza códigos QR para ejemplificar la identificación de un usuario en un terminal del sistema, en el que mediante un dispositivo móvil con scanner de códigos QR se realice la comprobación en el servidor y se devuelva la respuesta esperada.

Es una forma visual y diferente de representar cómo funciona la aplicación, y ayudaría a la hora de vender el sistema a futuros clientes.

Por otra parte, recalcar que se ha logrado finalizar el modulo que muestra estadísticas sobre los usuarios y el uso de la aplicación. El punto fuerte de este concepto es conseguir, desde la misma aplicación, recoger los datos y mostrarlos mediante gráficas y diagramas, para que sea agradable visualmente y facilite la extracción de tendencias.

En el aspecto positivo, este proyecto me ha permitido meterme en la piel de un desarrollador con tareas y fechas que cumplir, así como documentación y test que realizar.

También me ha permitido la libertad de escoger los lenguajes que mejor se adaptaban a cada problema y a buscar soluciones a problemas con los que no me había enfrentado aún. Además, me podido aprender mucho más de lenguajes que conocía pero no había utilizado demasiado como PHP y JavaScript.

Un aspecto muy positivo de este proyecto es que se puede observar como una pieza dentro de un esquema mayor. Una posible línea futura para este proyecto podría ser añadir un terminal de reconocimiento facial, de retina o cualquier otro hardware de identificación de personas y con alguna pequeña modificación en el servidor y la base de datos se podría crear un sistema autónomo más complejo y avanzado. No hay que ir muy lejos para obtener resultados muy positivos, se puede hacer algo muy interesante con un teléfono móvil con tecnología NFC y un sensor o un lector de tarjetas magnéticas. Las posibilidades son muy amplias y ofrecen un gran margen para la imaginación y la tecnología para fusionarse y obtener algo nuevo.

8 AGRADECIMIENTOS

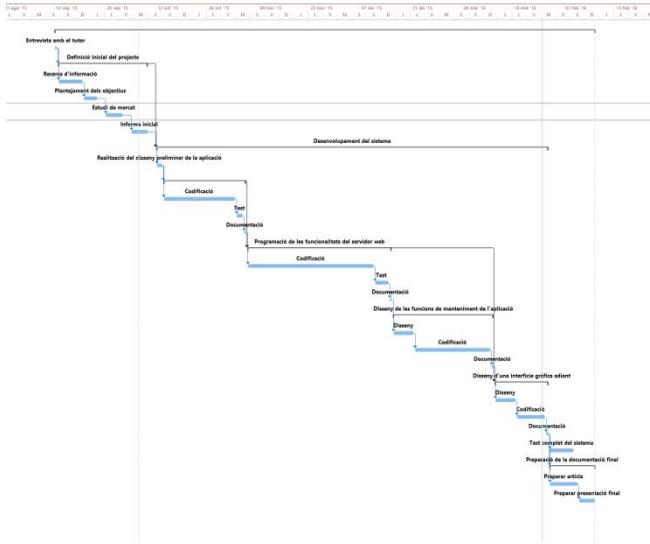
Agradecer al tutor Marc Talló Sendra por guiarme durante la concepción y la realización del proyecto, responder todas mis preguntas con intención constructiva y por su paciencia a lo largo de todo este proceso.

BIBLIOGRAFÍA

- [1] Villegas, Jaime. ¿Qué es un Sistema de Control de Acceso?
<http://www.tecnoseguro.com/faqs/control-de-acceso/%C2%BF-que-es-un-control-de-acceso.html> 22 de Febrero de 2009
- [2] Leal Sistemas Tipos de Control de Acceso
<http://www.lealsistemas.com.ar/control-de-acceso.html> 2013
- [3] Siasa. BioStar Lite 1.8.
<http://www.siasa.com/producto.php?prod=29000010> Siasa, 2011
- [4] Dorlet. Software de control Dassnet.
<http://www.dorlet.com/control/..%5Cdocs%5CSoftware%20-%20Control%20de%20Accesos.pdf> 2015.
- [5] Tutorialsonpoint. Tutorial de HTML5.
<http://www.tutorialspoint.com/html5/> 2016
- [6] BITMA. Tutorial de CSS. <http://html.net/tutorials/css/> 2016
- [7] The PHP Group. Información extensa sobre PHP5, sus clases y sus métodos. <http://php.net/manual/en/> 2016
- [8] Material didactico de la materia de Technologies i Desenvolupament Web, de la carrera de Ingeniera Informàtica. 2015
- [9] Bootstrap. Suite de código Bootstrap.
<http://getbootstrap.com/css/> 2016
- [10] LibrosWeb.es. Tutorial sobre el uso de JavaScript.
http://librosweb.es/libro/javascript/capitulo_7/validacion.html 2016.
- [11] Explicación y código de la función hash de PHP. The PHP group. <http://php.net/manual/es/function.password-hash.php> 2016

APÈNDIX 1. Diagrama de tareas

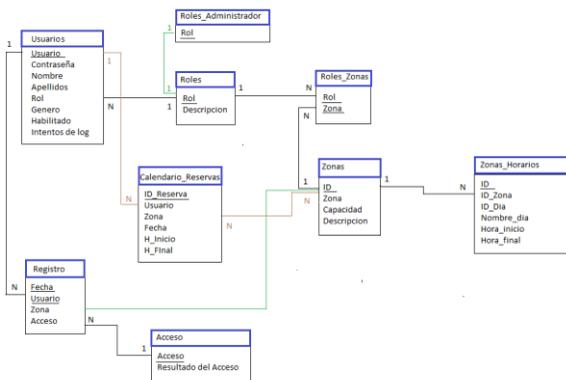
A continuación, podemos observar el diagrama de Gantt final del proyecto, con algunos cambios al respecto del inicial:



Se puede observar que dado que es un diseño iterativo y una metodología secuencial, no ha habido problemas de solapamiento de tareas ya que las hemos realizado siguiendo el orden establecido. La cual cosa no impide que al realizar una tarea, se hayan observado fallos o modificaciones necesarias en pasos anteriores y se proceda a cambiar ese concepto. Pero siempre son detalles y no requieren la modificación global del sistema o rediseño de la base de datos.

APENDIX 2. Diagrama Entidad-Relación de la Base de Datos

A continuación se mostrará un diagrama con las relaciones de las principales tablas del sistema y se explicará su funcionamiento:



Como se ha expresado en otros puntos del documento, en lo referente a la base de datos los conceptos pivotaes del proyecto son los usuarios, los roles y las zonas. Un usuario puede tener asignado un rol y ese rol tiene relaciones con las zonas que representan el acceso correcto a las mismas. Esta relación se crea mediante una tabla que une los identificadores de ambos conceptos para su fácil consulta.

De esta manera tenemos las cuatro tablas básicas relacionadas entre sí mediante foreign keys que crean un estado de coherencia entre los datos.

En el caso de las zonas y los roles, el parámetro que hace la función de primary key es un valor numérico iterativo que permite al introducir un nuevo concepto se cree con su ID propia y sea fácilmente accesible.

Algunos de estos conceptos se relacionan en las dos tablas siguientes, que aportan mucha funcionalidad al conjunto. La tabla de Registro, contiene toda la actividad del sistema. Todos los accesos se almacenan en esta estructura, que contiene relación con el usuario, la zona que accede y el resultado del intento de acceso. Aquí utilizamos una taula auxiliar de look-up para poder escribir simplemente un código en el registro y luego contrastar en esa tabla que representa. Es muy útil el uso de estas tablas para relacionar conceptos numéricos y de texto, para modificar un simple campo y que todos los elementos relacionados con ese atributo se modifiquen.

Seguidamente, la tabla de Calendario_Reservas recoge todas las reservas que se efectúan en la aplicación. Aquí guardamos información de las reservas que se relacionan con los usuarios y las zonas del sistema. Para el calendario se creó la tabla Zonas_Horarios, que contiene los horarios de disponibilidad de cada zona. Con esta relación podemos escoger la hora del día que más nos convenga para reservar la zona. Mediante el usuario podemos obtener el rol y así observar la relación. Si bien aumenta el tempo de join al tener que hacer una consulta extra, nos evita almacenar información inútil en el sistema.

Por último, tenemos las tablas auxiliares o de look-up que ayudan a simplificar el sistema. Estas tablas se han dejado del diagrama dado que no se relacionaban con las demás de manera directa, con la intención de dejar el diagrama lo más claro posible. Las tablas mencionadas son tablas que contienen información relevante para agilizar las funciones. Por ejemplo, nuestro calendario nos devuelve el día de la semana en inglés, así que tenemos una tabla con la traducción para poder tener el valor que nosotros tenemos. De la misma manera, tenemos la relación Numero del mes - Nombre del mes, Numero de día de la semana - Nombre del día de la semana, etc.

Por último, pero no por ello menos importante, la tabla Roles_Administrador contiene los roles que están definidos con permisos administrador, para un rápido acceso.