



Universidad
Zaragoza

Proyecto Fin de Carrera

Evolución de una aplicación de control de presencia
en entornos dinámicos

Autor

Agustín Raluy Pirla

Director

José Ruiz Mas

ESCUELA DE INGENIERÍA Y ARQUITECTURA
2017

Evolución de una aplicación de control de presencia en entornos dinámicos

RESUMEN

El presente proyecto fin de carrera plantea la evolución en el diseño e implementación de una aplicación de control atendido de presencia en entornos dinámicos mediante el uso de diferentes tecnologías tales como redes ad hoc y *RFID* y explora su vigencia tecnológica.

La aplicación surge en el marco de las *II Jornadas InfoDIEZ* (2004), organizadas por la asociación de estudiantes *Grupo DIEZ*, a partir de la necesidad de controlar y registrar la presencia de los estudiantes en las distintas sesiones de cara a la concesión de créditos de libre elección. El resultado, después de analizar los recursos disponibles, el estado de la tecnología y las condiciones del entorno, es una aplicación web corriendo en una red de ordenadores portátiles equipados con lectores de códigos de barras.

Las limitaciones de la tecnología de lectura llevan, a principios de 2006, a la sustitución de los lectores de códigos de barras por lectores *RFID*, con la correspondiente adaptación de la aplicación. El sistema se mantiene en producción hasta el año 2010, cuando se cede a *Univolution Servicios Avanzados, spin-off* de la Universidad de Zaragoza creada en el germen de la asociación *Grupo DIEZ*.

El principal requisito de diseño es dotar al sistema de la capacidad para adaptarse a cualquier condicionante del recinto en el que tenga que llevarse a cabo el control de presencia. Así, el sistema debe ser capaz de funcionar con independencia de la geometría del recinto, del número de puntos de acceso al mismo, del número de lectores y del acceso a Internet, así como de la disponibilidad de alimentación eléctrica y del acceso a infraestructura de red. El sistema debe ser autosuficiente. Por esta última razón se plantea el uso de PCs portátiles y redes ad hoc.

Además de a la configuración espacial, el sistema también debe adaptarse a la tipología de los lectores utilizados. En su primera implementación (2004) se utilizan lectores de código de barras debido a su bajo coste y alta disponibilidad en el mercado, pasando a utilizarse lectores y etiquetas *RFID* de muy corto alcance en cuanto la tecnología empieza a estar disponible comercialmente a un precio accesible. El sistema está preparado para el uso de cualquier otro tipo de lector, así como para permitir también la identificación manual en el caso de que se produzcan errores en el funcionamiento de los lectores.

Teniendo en cuenta todos los condicionantes anteriores, se opta por la estructura más sencilla posible que permite satisfacerlos todos: una aplicación desplegada en un servidor web ubicado en el nodo principal de una red ad hoc inalámbrica, a la que puede acceder mediante un simple navegador web tanto desde el propio nodo principal como desde cualquiera de los nodos restantes equipados con un lector.

Además del desarrollo y evolución de la aplicación entre 2004 y 2006, el presente proyecto termina mostrando las limitaciones y posibles mejoras de la solución planteada para adaptarse al marco tecnológicos, a las circunstancias y a las nuevas necesidades de mercado en el ámbito del control de presencia de la actualidad, contrastando su vigencia tecnológica.

TABLA DE CONTENIDOS

1. Introducción	5
1.1. Antecedentes	5
1.2. Motivación y objetivos	7
1.3. Estructura de la memoria.....	8
2. Evolución de la aplicación (2004 – 2010)	9
2.1. II Jornadas InfoDIEZ (marzo de 2004) [Propuesta inicial]	9
2.1.1. Análisis de la situación y justificación de la solución escogida	9
2.1.2. Desarrollo e implementación.....	11
2.1.3. Puesta en producción.....	16
2.2. III Jornadas InfoDIEZ (abril de 2005) [Cambio en protocolo de control].....	16
2.2.1. Análisis de la situación	17
2.2.2. Propuesta de mejora.....	18
2.2.3. Modificaciones en el sistema	19
2.2.4. Estimación de mejora tras las jornadas	20
2.3. IV Jornadas InfoDIEZ (abril de 2006) [Introducción RFID]	21
2.3.1. Análisis de la situación	21
2.3.2. Propuesta de mejora.....	22
2.3.3. Modificaciones en el sistema	25
2.3.4. Estimación de mejora tras las jornadas	26
2.4. I Jornadas i+DIEZ y siguientes (diciembre de 2006 – abril de 2010)	26
3. Una mirada al presente.....	28
3.1. Vigencia tecnológica.....	28
3.1.1. Tecnologías de lectura e identificación.....	28
3.1.2. Conectividad.....	29
3.1.3. Arquitectura cliente-servidor	30
3.1.4. Desarrollo de la aplicación	30
3.2. Propuesta de aplicación en la actualidad.....	31
3.3. Otros usos de control de presencia	32
4. Conclusiones y líneas futuras	33
5. Bibliografía	34
Anexo I. Funcionamiento de la aplicación: manual de usuario.....	35
Anexo II. Código fuente de la aplicación y base de datos	46

TABLA DE FIGURAS

Fig. 2.1 – Servidor web + PHP + MySQL	11
Fig. 2.2 – Plantilla de acreditaciones	13
Fig. 2.3 – Ejemplo de acreditación	13
Fig. 2.4 – Estructura de red ad hoc utilizada	13
Fig. 2.5 – Configuración IP	14
Fig. 2.6 – Configuración red ad hoc	14
Fig. 2.7 – Esquema de la aplicación web (2004)	15
Fig. 2.8 – Instalación de los puntos de identificación en el recinto de las jornadas	16
Fig. 2.9 – Nuevo esquema de lectura	18
Fig. 2.10 – Lector NPR DESK II	22
Fig. 2.11 – Tags RFID	22
Fig. 2.12 – Comienzo y finalización de sesiones	23
Fig. I.1 – Carga de identificadores RFID	35
Fig. I.2 – Pantalla de arranque, antes del comienzo de las jornadas	36
Fig. I.3 – Clientes a la espera de que se abra la lectura	37
Fig. I.4 – Control de acceso, antes del comienzo de una ponencia	38
Fig. I.5 – Control de acceso, antes del comienzo de una ponencia (clientes)	39
Fig. I.6 – Lectura de una acreditación (entrada)	40
Fig. I.7 – Lectura de una acreditación (salida)	41
Fig. I.8 – Control de acceso en el transcurso de una ponencia	42
Fig. I.9 – Control de acceso al finalizar una ponencia	43
Fig. I.10 – Control de acceso al finalizar una ponencia (fin de bloque)	44
Fig. I.11 – Control de acceso al término de las jornadas	45
Fig. II.1 – Diagrama de estados de la aplicación	46

TABLA DE ACRÓNIMOS

API	Application Programming Interface
CSS	Cascading Style Sheets
DHCP	Dynamic Host Configuration Protocol
DIEZ	Desarrollo Informático Empresarial Zaragoza
DNI	Documento Nacional de Identidad
GPS	Global Positioning System
GSM	Global System for Mobile Communications
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IP	Internet Protocol
NFC	Near Field Communication
PC	Personal Computer
PDF	Portable Document Format
PHP	PHP: Hypertext Processor (acrónimo recursivo)
PVC	Polyvinyl Chloride
QR	Quick Response
REST	Representational State Transfer
RFID	Radio-Frequency Identification
SDK	Software Development Kit
SQL	Structured Query Language
SSID	Service Set Identifier
TIC	Tecnologías de la Información y Comunicaciones
USB	Universal Serial Bus
WEP	Wired Equivalent Privacy

1. Introducción

El principal objetivo de este proyecto fin de carrera es mostrar la evolución de una aplicación de control de presencia desde su nacimiento en 2004 hasta su último uso en 2010 y posteriormente contrastar su vigencia tecnológica frente a la evolución tecnológica en los últimos años hasta la actualidad. Tiene como origen el trabajo realizado por el autor del proyecto para la asociación de estudiantes *Grupo DIEZ* durante el periodo 2004 – 2006.

La asociación *Grupo DIEZ* (acrónimo de *Desarrollo Informático Empresarial de Zaragoza*) nace en marzo de 1998 en el seno de la *Facultad de Ciencias Económicas y Empresariales* (hoy *Facultad de Economía y Empresa*) de la Universidad de Zaragoza con los siguientes objetivos:

- Fomentar el uso de la informática y las nuevas tecnologías de información y comunicación (*TIC*) en todos los estamentos de la comunidad universitaria aragonesa.
- Ofrecer formación a los miembros de dicha comunidad universitaria en el uso y aprendizaje de herramientas y técnicas basadas en la información y en la informática.
- Desarrollar soportes informáticos con fines docentes e investigadores, así como apoyar tecnológicamente, en la medida de sus posibilidades, a la comunidad universitaria aragonesa.
- Promover cualquier otra iniciativa que favorezca el fomento de la informática y las nuevas tecnologías de la información.

Hasta su clausura definitiva en 2011, desarrolla sus actividades en 4 grandes áreas: área de formación, área de administración de recursos técnicos, área de proyectos con terceros (departamentos universitarios, empresas e instituciones públicas) y área de jornadas.

Dentro de esta última área se encuentra la actividad más representativa de la asociación: la coordinación anual de las *Jornadas InfoDIEZ*, desarrolladas durante 8 ediciones entre los años 2003 y 2010 en la *Facultad de Ciencias Económicas y Empresariales* y las *Jornadas i+DIEZ*, desarrolladas durante 4 ediciones entre los años 2006 y 2009 en la *Escuela Universitaria de Estudios Empresariales de Zaragoza* (hoy también *Facultad de Economía y Empresa*).

En el contexto de estas jornadas es donde se desarrolla la aplicación de control de presencia objeto de este proyecto fin de carrera. Todo el trabajo se desarrolla desde la empresa *Dataria Comunicación, S.L.*, fundada en enero de 2003 por el autor del proyecto, antiguo miembro de la asociación *Grupo DIEZ*, junto con otros dos antiguos miembros. Se llega al acuerdo de desarrollar de forma gratuita una solución de control de asistencia para las jornadas, a cambio de un patrocinio en las mismas.

1.1. Antecedentes

En marzo de 2003 se llevan a cabo las *I Jornadas InfoDIEZ*. Aunque son unas jornadas abiertas a toda la sociedad están principalmente orientadas al alumnado de la Universidad de Zaragoza. Tal y como suele ser habitual en este tipo de jornadas, el aprovechamiento académico de las mismas está retribuido con la concesión de créditos de libre elección. Por esta razón este tipo de actividades en general, y las *Jornadas InfoDIEZ* en particular, gozan de una gran aceptación entre el alumnado durante todos los años en los que se desarrollan.

Según el reglamento de la Universidad de Zaragoza, ese aprovechamiento académico para la concesión de los créditos de libre elección se traduce en la superación de una prueba de evaluación sobre los contenidos de las jornadas y en una asistencia mínima del 80% de las horas lectivas. Es necesario, por lo tanto, que exista un mecanismo para contabilizar la presencia de los estudiantes durante las distintas ponencias y mesas redondas que componen las jornadas. Tradicionalmente, este control se lleva a cabo mediante las clásicas *hojas de firmas*. Para cada una de las sesiones de las jornadas se genera un listado con el nombre y DNI de todos los inscritos en el que cada uno de los asistentes verifica su presencia mediante su firma caligráfica.

Este método tiene como principal ventaja su sencillez, puesto que el único requisito necesario para poder ponerlo en marcha es la impresión previa de las hojas. El principal inconveniente del método es que, una vez se tienen rellenas las hojas de firmas, es necesaria la puesta en común de las mismas para verificar si cada uno de los alumnos alcanza el 80% necesario. Este es un trabajo manual y muy tedioso, propenso al error humano a la hora de contar.

Existen dos métodos para trabajar con estas hojas de firmas, cada uno con sus pros y, principalmente, contras.

- a) Las hojas de firmas se distribuyen para su firma durante la ponencia.

Es el método más ágil, puesto que no requiere tiempo adicional, pero presenta algunos problemas:

- No hay control sobre las firmas: algún estudiante puede firmar la asistencia de compañeros que no han acudido a la sesión.
- Al llevarse a cabo el proceso de firma durante la sesión, la atención de los asistentes se desvía del ponente durante unos minutos. Esto redunda en un menor aprovechamiento y en mala imagen de cara al invitado, los alumnos parecen más interesados en los créditos de libre elección que en la ponencia.

- b) La asistencia se firma en un punto de control de la organización, al comienzo o a la finalización de la ponencia.

Este método resuelve los inconvenientes del método anterior, puesto que se elimina el riesgo de fraude y se consigue focalizar la atención de los estudiantes en la ponencia, pero a su vez introduce otros:

- Es necesario que al comienzo/final de una sesión todos los asistentes se desplacen al punto de control. Este es un proceso lento, sobre todo cuando el número de asistentes es elevado, y normalmente acarrea retrasos con respecto a los horarios previstos.

El método escogido para las *I Jornadas InfoDIEZ* es el de la distribución de las hojas para su firma durante las ponencias.

1.2. Motivación y objetivos

Tras la conclusión de las *I Jornadas InfoDIEZ*, realizadas en el salón de actos de la Facultad de Ciencias Económicas y Empresariales, se lleva a cabo un análisis crítico de la preparación, desarrollo y cierre de las mismas para tratar de identificar tanto los aciertos como los problemas detectados, así como los aspectos susceptibles de mejora.

Entre los últimos se encuentra precisamente el control de asistencia: se constata que la validación y puesta en común requiere demasiada dedicación e, incluso así, se detectan varios errores a posteriori. Se alcanza el acuerdo de tratar de mejorar el mecanismo de control de cara las *II Jornadas InfoDIEZ* previstas para el año siguiente. Esta decisión marca el pistoletazo de salida de la aplicación de control de asistencia.

Los objetivos de diseño son los siguientes:

- Automatizar el mecanismo de identificación durante las sesiones eliminando las hojas de firmas.
- Automatizar el mecanismo de puesta en común de todas las sesiones y generación de resultados finales eliminando los errores humanos en el conteo.
- Eliminar el fraude en la asistencia, tanto inscritos que no asisten y alguien firma en su lugar como asistentes que no han pagado la cuota de inscripción.
- Permitir múltiples puntos de lectura para agilizar el proceso.
- Dotar al sistema de movilidad para poder reutilizarlo en diferentes eventos y lugares con distintas configuraciones.
- Dotar al sistema de independencia de servidores online para evitar problemas de conectividad en entornos no controlados.
- Dotar al sistema de independencia de alimentación eléctrica que permitiese su funcionamiento en entornos no controlados.
- Minimizar la complejidad de puesta en marcha de cada punto de identificación.

Hay, además, otros tres factores que motivan el desarrollo de la aplicación:

- En línea con el espíritu y los objetivos de la asociación, existen incentivos para tratar de ser innovadores y apostar por una solución novedosa y apoyada en la tecnología.
- En el ámbito personal, se percibe como una oportunidad magnífica para poder afrontar un proyecto real como ingeniero. A partir de una situación inicial y de unos objetivos fijados, el trabajo consiste en evaluar la situación, conocer el estado del arte, diseñar la mejor solución posible dentro de la restricción presupuestaria de la asociación e implementarla en plazo y forma. Para el autor, este proyecto supone una magnífica oportunidad de aprendizaje.
- En el ámbito profesional, se acaba de constituir *Dataria Comunicación, S.L.* y la puesta en marcha de un sistema como este es una excelente oportunidad para hacer crecer la cartera de productos y dar la empresa a conocer gracias al patrocinio de las jornadas.

En cuanto a los recursos disponibles para llevar a cabo el proyecto, se cuenta con los recursos materiales de la asociación (varios PCs de sobremesa y dos PCs portátiles, acceso permanente a Internet y una impresora), así como de una partida de 150€ dentro del presupuesto de las jornadas para potenciales compras.

Para poder materializar el proyecto es necesario realizar previamente las siguientes tareas:

- Seleccionar, de entre todas las tecnologías de identificación disponibles en el mercado, la que mejor se adapte a los objetivos marcados teniendo en cuenta la restricción presupuestaria.
- Elegir, una vez se ha seleccionado la tecnología de identificación, el equipamiento concreto y la configuración necesaria de entre todas las posibilidades disponibles en el mercado.
- Desarrollar una aplicación que, a partir de los dispositivos de identificación elegidos sea capaz de gestionar de manera centralizada el control de presencia y mostrar los resultados de una forma apta para su utilización.
- Definir una arquitectura de red que permita la comunicación de los diferentes puntos de identificación entre sí, así como los mecanismos de comunicación entre los distintos nodos.
- Integrar el acceso de los datos generados por la aplicación en la intranet de la asociación.

1.3. Estructura de la memoria

Tal y como se ha indicado en la introducción, el presente proyecto recoge la evolución de un sistema a lo largo de un periodo de 3 años (2004 – 2006) para terminar dando un salto hasta la actualidad.

Los cambios en la aplicación se han llevado a cabo en distintas iteraciones entre las distintas ediciones de las *Jornadas InfoDIEZ* permitiendo mejorar y cubrir las carencias en base a lo aprendido en un entorno real de producción. Esta evolución se detalla en el capítulo 2.

El capítulo 3, por otro lado, acerca la solución adoptada hasta nuestros días y contrasta la vigencia de la misma en relación con todos los cambios y avances tecnológicos de los últimos años.

La estructura de la memoria es la siguiente:

- Capítulo 1. Introducción
- Capítulo 2. Evolución de la aplicación (2004 – 2010)
- Capítulo 3. Una mirada al presente
- Capítulo 4. Conclusiones y líneas futuras
- Capítulo 5. Bibliografía

Se incluyen, además, los siguientes anexos:

- Anexo I. Funcionamiento de la aplicación: manual de usuario
- Anexo II. Código de la aplicación y base de datos

2. Evolución de la aplicación (2004 – 2010)

A lo largo de este capítulo se va a analizar la evolución de la aplicación desde su creación, vinculada a las *II Jornadas InfoDIEZ* en 2004, hasta su retirada coincidiendo con las *VIII Jornadas InfoDIEZ* en 2010, últimas que se celebraron antes del cierre de la asociación.

2.1. II Jornadas InfoDIEZ (marzo de 2004) [Propuesta inicial]

2.1.1. Análisis de la situación y justificación de la solución escogida

De acuerdo con lo expuesto en el capítulo anterior, para poder cumplir con los objetivos previstos es necesario previamente analizar y definir sus tres elementos principales:

- El mecanismo y la tecnología de identificación para sustituir a las hojas de firmas.
- La estructura de red que permita, de una forma portátil y flexible, la instalación de varios puntos de identificación con independencia de las características del recinto.
- La propia aplicación, que permita cumplir la funcionalidad prevista uniendo los dos elementos anteriores.

Identificación de asistentes

El principal objetivo del proyecto es mejorar el sistema que se estaba utilizando para controlar la asistencia a las distintas sesiones de las jornadas. Para ello, el primer paso tiene que ser eliminar las hojas de firmas y toda la gestión manual, proclive a errores, que necesitan estas hojas y sustituir todo el proceso por otro completamente automatizado.

La solución más directa para evitar el fraude en la asistencia es la validación individual de los asistentes por parte de la organización en la entrada del recinto. De esta forma se puede evitar el acceso a los no inscritos y tener un control de quiénes de los inscritos asisten realmente a cada sesión. Por otro lado, la forma más directa de realizar una validación individual es a través de las tarjetas de acreditación que se entregan a cada inscrito al comienzo de las jornadas. El reto está, entonces, en ser capaces de leer estas acreditaciones de forma automática.

Las 2 tecnologías de lectura disponibles en el momento son la óptica (códigos de barras o bidimensionales) y la electromagnética (*RFID*). La segunda opción, *RFID*, se encuentra comercialmente en sus inicios (Zaragoza comienza pruebas piloto en autobuses en 2002, el sistema termina de implantarse en 2005) y, aunque es la opción más atractiva a priori, el alto coste de la misma (lectores y *tags*) resulta prohibitivo para el reducido presupuesto con el que se cuenta por lo que también se decide descartarla.

En cuanto a las tecnologías de lectura óptica, aunque los códigos bidimensionales (QR) son más robustos y por lo tanto ofrecen una mayor velocidad de lectura, también se descartan, esta vez por razones prácticas: por un lado, los lectores todavía son sensiblemente más caros que los unidimensionales, pero además el proceso de creación e impresión de los códigos individualizados en las etiquetas es más complejo y menos automatizable que en el caso de los códigos de barras.

Así pues, el mecanismo finalmente escogido para la identificación de los asistentes es la lectura de códigos de barras impresos en las acreditaciones de los asistentes.

Es importante destacar que estos lectores no funcionan de forma autónoma: una vez leído el código de barras, es necesario recoger la lectura desde un ordenador. Así, definimos los puntos de identificación como el conjunto formado por el lector de código de barras y el ordenador al que está conectado. Para este proyecto, los lectores se van a conectar a los equipos portátiles disponibles en la asociación.

Estructura de red

La sede única de las *I Jornadas InfoDIEZ* fue el salón de actos de la *Facultad de Ciencias Económicas y Empresariales*. La idea original es mantener la misma sede para las *II Jornadas InfoDIEZ*, aunque por problemas de disponibilidad de la sala, resulta necesario trasladar al *Paraninfo* de la Universidad de Zaragoza todas las sesiones previstas para el primer día.

Circunstancias como esta obligan a plantear la movilidad del sistema como requisito. El sistema no solo debe ser móvil, sino que tiene que ser independiente de las características físicas y de las posibilidades técnicas (conexión a Internet, enchufes cerca de los puntos de lectura...) de la sala.

En el momento en el que se plantea esta aplicación, no llegaría hasta finales de 2005, todavía no hay cobertura inalámbrica *WiFi* en los espacios de la Universidad de Zaragoza en los que se iban a desarrollar las jornadas y tampoco existe la posibilidad de disponer de conexión cableada en las puertas de los salones. Las redes de datos móviles *GSM* quedan también descartadas por su inestabilidad, baja velocidad y elevado precio.

Ante la posibilidad de que, en determinados escenarios, no haya disponibilidad de alimentación eléctrica, se decide prescindir de elementos de red externos (*routers*, puntos de acceso inalámbrico...). De esta forma, se opta finalmente por crear una red ad hoc inalámbrica entre los distintos ordenadores portátiles que, equipados con batería, van a ser usados en los puntos de lectura de acreditaciones.

Aplicación de control de asistencia

La aplicación de control de asistencia tiene que ser capaz de recoger los datos capturados por los lectores de código de barras en cada uno de los puntos de identificación y guardarlos de forma estructurada en una base de datos común, para cada estudiante inscrito y para cada sesión de las jornadas.

La forma más sencilla de guardar estos datos de forma estructurada es mediante el uso de bases de datos. Se podría trabajar con una base de datos independiente en cada uno de los ordenadores portátiles de los puntos de lectura y unificarlas al finalizar las jornadas, pero resulta mucho más eficaz (y sencillo) aprovechar la red ad hoc definida en el apartado anterior y poder así utilizar una única base de datos accesible desde todos los puntos de identificación.

Así, definimos una arquitectura cliente-servidor en la que, a través de la red ad hoc, un número indeterminado de clientes (tantos como puntos de identificación) interactúan con la base de datos de asistencia en el servidor.

De entre todas las formas posibles que puede tomar la aplicación, aprovechando la estructura cliente-servidor recién definida, se opta por realizar una aplicación web. Los puntos de identificación (clientes) ejecutan la aplicación directamente desde el navegador web sin necesitar de instalar ningún otro componente. De esta forma, para el despliegue y puesta en

funcionamiento del sistema sólo va a ser necesario preparar y configurar previamente uno de los ordenadores portátiles (servidor), mientras que el resto (clientes) no requieren de ninguna configuración: podemos integrar tantos puntos de identificación como queramos simplemente conectándolos a la red.

Para el desarrollo de la aplicación web, se escoge uno de los esquemas más estables y consolidados del momento: Servidor web + *PHP* + *MySQL*.

- Servidor web: Encargado de servir las páginas de la aplicación web (interfaz) a los clientes y de recoger las peticiones realizadas por los mismos. Dado que los equipos portátiles disponibles en la asociación funcionan con *Windows XP*, el servidor web finalmente escogido es *Sambar Server* (actualmente obsoleto) debido a su alto nivel de optimización para sistemas operativos *Windows*. Cuando se detiene su desarrollo en 2007, es sustituido por el servidor *Apache*.
- *PHP*: Lenguaje de programación del lado del servidor que permite la creación de contenido web dinámico. Además, a través de su conector *mysql* (hoy obsoleto), se puede trabajar con bases de datos *MySQL* sin necesidad de componentes externos.
- *MySQL*: motor de base de datos.

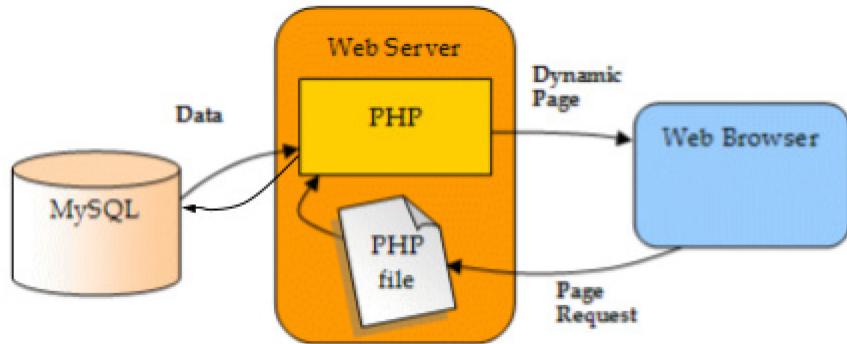


FIG. 2.1 – SERVIDOR WEB + PHP + MySQL

2.1.2. Desarrollo e implementación

Infraestructura

La única infraestructura necesaria para poder poner en marcha el sistema, además de los códigos de barras individualizados impresos en las acreditaciones, es la de los puntos de identificación que se vayan a utilizar, al menos uno en cada uno de las puertas de acceso al recinto. Cuantos más puntos de identificación haya disponibles, más rápida y ágil será la identificación de todos los asistentes. Para las *II Jornadas InfoDIEZ* se utilizan 2 puntos de identificación, situados en cada una de las 2 puertas de entrada al salón de actos.

Es importante destacar que, aunque en el apartado anterior hemos definido la necesidad de contar con un servidor (web y base de datos), su presencia no requiere de ningún equipo adicional. Los servidores, tanto *Sambar Server* como *MySQL* se ejecutan en segundo plano como servicios, por lo que el servidor puede convivir en la misma máquina que uno de los puntos de identificación clientes.

Cada punto de identificación se compone de un ordenador portátil y un lector de código de barras.

[Lectores de código de barras](#)

Aunque existen lectores de códigos de barras con diferentes interfaces de conexión (*Bluetooth, RS232, USB* y *PS/2*), rápidamente se descartan los lectores *Bluetooth* y *RS232* ya que requieren de alimentación externa. Entre *USB* y *PS/2* se opta finalmente por lectores *PS/2* y así evitar la instalación de drivers adicionales para conectarlos al PC. Se adquieren 2 lectores *Metrologic MS9520* por un importe total de 140€, dentro del presupuesto disponible.

Este tipo de lectores se comportan exactamente igual que un teclado estándar: si en un teclado cada vez que se pulsa una tecla se envía ese carácter a la aplicación en primer plano, en el caso de un lector de código de barras se envía la cadena alfanumérica codificada completa junto con un retorno de carro cada vez que se realiza una lectura.

[Ordenadores portátiles](#)

Los equipos a los que van conectados los lectores de código de barras deben cumplir las siguientes características:

- Ser portátiles, para permitir la movilidad de los puntos de identificación.
- Tener baterías en buen estado, para poder operar incluso en el caso de falta de alimentación.
- Disponer de interfaz inalámbrico para poder integrarse en la red ad hoc.
- Disponer de puerto *PS/2* para la conexión del lector de código de barras.
- Gracias a la naturaleza de la aplicación web, puede utilizarse cualquier sistema operativo que disponga de navegador. Para el servidor, es necesario un equipo con sistema operativo *Windows*.

Los equipos finalmente utilizados son los dos ordenadores portátiles disponibles en la asociación. Ambos cubren todos los requisitos enunciados y utilizan como sistema operativo *Windows XP*.

En el caso del equipo que iba a realizar la labor de servidor, se instalan los servidores *Sambar Server* (v6.01) con la extensión *PHP* (v4.3.4) y *MySQL* (v4.0.20a) necesarios para la aplicación web. También se instala un servidor *DHCP* (*DHCPServer* v1.5.2) para la configuración de la red ad hoc. El segundo equipo (cliente) no necesita ninguna preparación previa.

[Generación de códigos de barras](#)

Un código de barras es la representación de una cadena alfanumérica de caracteres utilizando un conjunto de líneas paralelas de distinto grosor y espaciado de forma biunívoca. A cada carácter le corresponde un conjunto de líneas y cada conjunto de líneas codifica un único carácter. A efectos prácticas, generar un código de barras a partir de una cadena de caracteres es escribir esa misma cadena utilizando la “fuente” de código de barras.

En cuanto al diseño de las acreditaciones, se opta por aprovechar el campo *DNI*, disponible en la base de datos de asistentes, como identificador. Así pues, se añade el *DNI* del asistente codificado como código de barras en la plantilla de la acreditación.

Para la generación e impresión de las acreditaciones se utiliza la opción de *Combinar Correspondencia* a partir de un origen de datos *MySQL* de *Microsoft Word* utilizando la base de datos de asistentes. Para los textos de la acreditación se utilizan fuentes estándar, mientras que

para el DNI se utilizó la fuente *barcode*. De esta forma el DNI aparece en el formato código de barras.

En las siguientes figuras puede verse ejemplos de la plantilla, así como de una etiqueta generada a partir de esa plantilla.

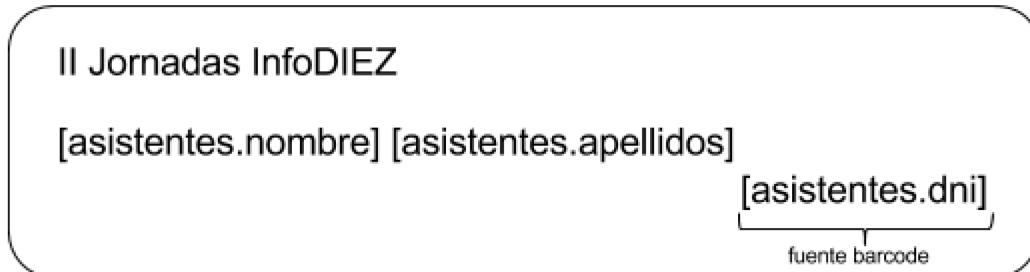


FIG. 2.2 – PLANTILLA DE ACREDITACIONES



FIG. 2.3 – EJEMPLO DE ACREDITACIÓN

Configuración de la red ad hoc

Siguiendo el objetivo marcado de minimizar la complejidad de la puesta en marcha de los distintos puntos de identificación, se propone una estructura en la que no sea necesaria ninguna configuración adicional en los clientes más allá de conectarse con la red inalámbrica generada por el servidor.

Para ello, en el mismo equipo en el que se han instalado los servidores web y MySQL, es necesario configurar el interfaz de red inalámbrica en modo ad hoc (equipo a equipo) y permitir la conexión del resto de equipos. Hay que instalar y configurar, además, un servidor *DHCP* que será el encargado de configurar las conexiones IP de los distintos nodos conectados.

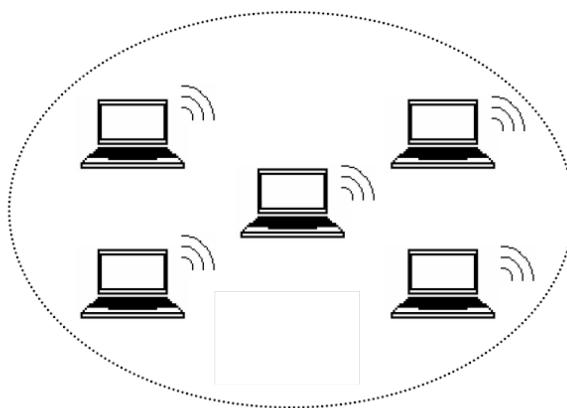


FIG. 2.4 – ESTRUCTURA DE RED AD HOC UTILIZADA

En primer lugar, es necesario establecer la configuración *IP* del interfaz inalámbrico del portátil servidor. En este caso, se define el host 192.168.10.1 dentro de la red 192.168.10.0/24 en el interfaz inalámbrico.

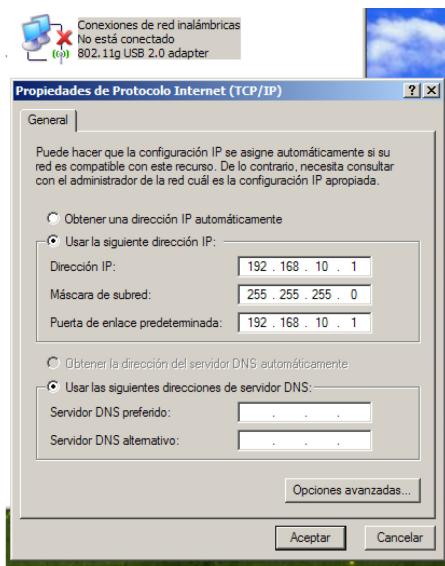


FIG. 2.5 – CONFIGURACIÓN IP

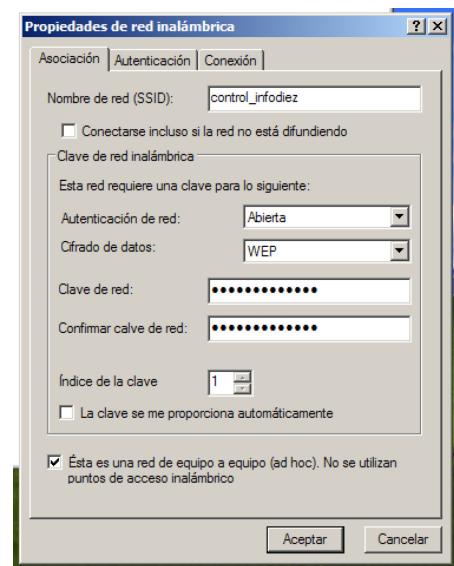


FIG. 2.6 – CONFIGURACIÓN RED AD HOC

A continuación, se crea y se pone en marcha la red ad hoc (equipo a equipo), con el *SSID* *control_infodiez* y la clave *WEP* (máximo nivel de seguridad soportado por *Windows XP* para redes ad hoc) *infodiez_2003*.

El último paso es instalar y configurar el servidor *DHCP* para que todos puntos de identificación que se conecten a la red ad hoc obtengan automáticamente la configuración *IP* y puedan acceder directamente a la aplicación web. El servidor *DHCP* escogido es el servidor gratuito *DHCP Server for Windows* (<http://www.dhcpserver.de>) y estos los parámetros necesarios de configuración para asignar direcciones *IP* en el rango 192.168.10.2 a 192.168.10.10.

```
[SETTINGS]
IPPOOL_1=192.168.10.2-10
IPBIND_1=192.168.10.1
AssociateBindsToPools=1

[GENERAL]
SUBNETMASK=255.255.255.0
```

Aplicación y base de datos

En esta primera iteración la aplicación web consta de un formulario con un único campo en el que se introduce el DNI del estudiante que está entrando en el recinto. Cuando se carga el formulario, el foco se sitúa en el campo de texto de forma que cuando se produzca la respuesta desde el lector el DNI se rellena automáticamente y se envía al servidor.

En el servidor, se comprueba que el DNI leído sea correcto y corresponda a un inscrito y, en ese caso, se registra la presencia del estudiante en la ponencia en curso y se devuelve una respuesta favorable “el usuario ha sido registrado con éxito” que se visualiza en el navegador, quedando limpio el formulario preparado para la siguiente lectura.

En el caso de que la lectura no haya sido correcta, también aparece indicado en pantalla, pudiéndose intentar una nueva lectura con el lector de código de barras o bien introducir directamente el DNI con el teclado del portátil.

En la siguiente figura puede verse un esquema de la aplicación web.

FIG. 2.7 – ESKEMA DE LA APLICACIÓN WEB (2004)

Todo lo que la aplicación web muestra en pantalla está implementado con *HTML + CSS*, preprocesado por el motor *PHP*. El aspecto visual final de la aplicación, así como su código fuente (*HTML + CSS + PHP*) están disponibles en los anexos I y II.

En cuanto a la base de datos, contiene 3 tablas: *asistentes*, con información relativa a los asistentes a las jornadas; *sesiones*, con información relativa a todas las ponencias de las jornadas y *control*, en la que se guardan todas las lecturas con su *timestamp*.

```

CREATE TABLE `asistentes` (
  `dni` varchar(10) NOT NULL default '',
  `apellidos` varchar(50) NOT NULL default '',
  `nombre` varchar(50) NOT NULL default '',
  `asistencias` tinyint(2) NOT NULL default '0',
  `apto` tinyint(2) NOT NULL default '0',
  KEY `dni` (`dni`)
) TYPE=MyISAM;

CREATE TABLE `sesiones` (
  `sesion` tinyint(2) NOT NULL auto_increment,
  `titulo` varchar(200) NOT NULL default '',
  `ponente` varchar(200) NOT NULL default '',
  `inicio` datetime NOT NULL default '0000-00-00 00:00:00',
  `estado` tinyint(2) NOT NULL default '0',
  KEY `cod_sesion` (`sesion`)
) TYPE=MyISAM;

CREATE TABLE `control` (
  `id` bigint(255) unsigned NOT NULL auto_increment,
  `dni` varchar(10) NOT NULL default '',
  `hora` datetime NOT NULL default '0000-00-00 00:00:00',
  `sesion` tinyint(2) NOT NULL default '0',
  KEY `id` (`id`)
) TYPE=MyISAM;

```

2.1.3. Puesta en producción

La misma mañana de comienzo de las jornadas se procede a vaciar las bases de datos del servidor (hasta el momento tenían datos de prueba) y a cargarlas de nuevo actualizadas con los últimos estudiantes inscritos y algunas correcciones de última hora en las sesiones. A continuación, se montan los puntos de identificación al lado de las puertas de entrada al salón de actos.

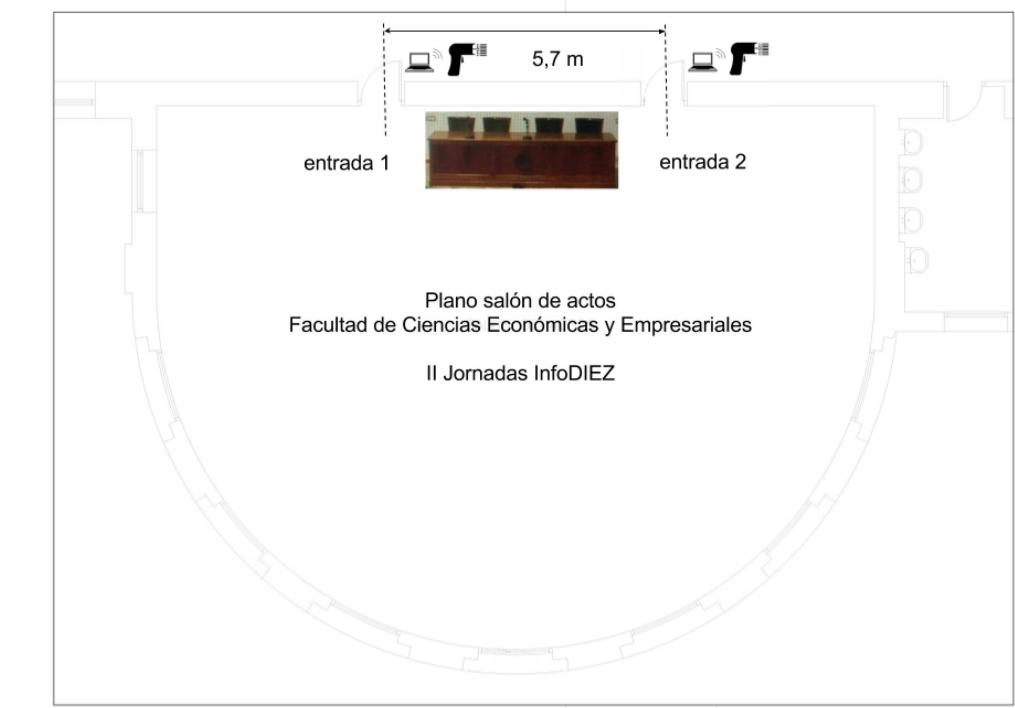


FIG. 2.8 – INSTALACIÓN DE LOS PUNTOS DE IDENTIFICACIÓN EN EL RECINTO DE LAS JORNADAS

Los puntos de identificación se montan al principio de las sesiones de mañana y de tarde y se recogen al mediodía y al finalizar la última ponencia de cada día, siempre sin incidencias de comunicación. Tampoco se registran problemas en la lectura de códigos de barras.

Una vez finalizadas las jornadas, para llevar a cabo la evaluación de asistencia, tan solo hay que consultar la tabla *asistentes*. Se considera que aquellos estudiantes cuyo campo *asistentes.asistencias* contiene un valor igual o superior al mínimo exigido (80%) son considerados aptos.

2.2. III Jornadas InfoDIEZ (abril de 2005) [Cambio en protocolo de control]

Al finalizar las *II Jornadas InfoDIEZ* comienza una nueva iteración de mejora de la aplicación. Al igual que en la anterior, el proceso parte de un nuevo análisis de la situación para verificar si se han conseguido corregir los problemas detectados en la edición anterior y alcanzar con éxito los objetivos fijados. A partir de ahí, se implementan aquellas medidas necesarias para corregir problemas e implementar las nuevas mejoras que se hayan definido.

2.2.1. Análisis de la situación

En general, la sensación es altamente satisfactoria. La aplicación web ha funcionado correctamente durante todas las jornadas y no se han encontrado problemas de lectura ni incidencias con el control de asistencia. A diferencia del año anterior, el número de errores en el control y de quejas de los estudiantes baja hasta 0. Además, al haber sido validados todos los accesos por la organización, se consigue eliminar también el fraude de asistencia. Sin duda, todo un éxito en este sentido.

Sin embargo, se detecta también que la introducción de esta nueva aplicación de control de presencia introduce un grave problema no previsto en el planteamiento inicial. Todos los test realizados durante la preparación de las jornadas habían sido realizados a pequeña escala, y en ningún caso nos detuvimos a pensar en cómo afectaría a la dinámica de las jornadas tener cerca de 150 inscritos. Debíamos incluir la escalabilidad del sistema como objetivo fundamental.

El protocolo de control de asistencia establecido requiere que, antes de cada ponencia, todos los estudiantes tienen que pasar por alguno de los puntos de identificación situados en las dos puertas de acceso al salón de actos. Esto implica que, al finalizar cada ponencia, todos los asistentes tienen que abandonar la sala, identificarse de nuevo y volver a entrar.

Todo este proceso, incluso contando con dos puntos de identificación, resulta terriblemente lento. Sobre todo, por los cuellos de botella que se forman en torno a las puertas, con estudiantes tratando de entrar y salir simultáneamente. Esta situación provoca retrasos, que se van haciendo cada vez más grandes conforme va avanzando el día. Como medida preventiva es necesario pedir a los ponentes que acortasen un poco sus charlas para compensar los retrasos en la identificación. Se trata, en cualquier caso, de una situación inaceptable que no puede repetirse en la nueva edición.

Por otro lado, de acuerdo con las estimaciones de la asociación, en la nueva edición iban a superarse los 200 inscritos. La previsión finalmente se cumple y, debido a las limitaciones de aforo del salón de actos de la facultad resulta necesario trasladar las jornadas a una nueva ubicación, el *Hotel Meliá Zaragoza*. Si el problema de los retrasos provocados por las lecturas es ya importante con 150 inscritos, ¿cómo será de grave con más de 200?

Una de las posibles soluciones para evitar los retrasos sería aumentar el número de puntos de identificación, pero se descarta porque no soluciona el problema de los cuellos de botella en las puertas, además de suponer un incremento en el presupuesto. Otra posible solución consiste en cambiar el protocolo de control, modificando a posteriori la aplicación web para recoger tales cambios. Esta última es la opción finalmente escogida.

Cabe indicar que las condiciones técnicas en el *Hotel Meliá Zaragoza* resultan extremas: no se dispone de conexión a Internet y ni de alimentación eléctrica en los puntos de identificación. Gracias a las baterías de los ordenadores portátiles y a la red ad hoc implementada consigue superarse esta limitación.

2.2.2. Propuesta de mejora

Se observó que la mayor parte del tiempo perdido se dedica a salir del recinto para a continuación volver a entrar. Esta acción, tan evidente ahora como absurda, resulta necesaria al considerar la asistencia como algo absoluto, como un binomio *asiste / no asiste* que debe ser evaluado para cada una de las ponencias. Pero, ¿qué pasaría si considerásemos la asistencia a cada ponencia de forma indirecta?

La solución planteada pasa por calcular la asistencia a cada ponencia no de forma directa, sino a partir del tiempo que el estudiante permanece dentro de la sala. De esta forma, ya no es necesario identificar a todos (n) los estudiantes en todas (m) las ponencias ($n \times m$ lecturas) sino registrar únicamente las entradas y salidas de cada uno para conocer, en todo momento si se encuentran dentro o fuera de la sala. Se observa que la mayor parte de los estudiantes no abandona el salón de actos entre ponencias, con la consiguiente reducción en el número de lecturas. Por otro lado, tampoco es necesario realizar lecturas de salida al finalizar cada bloque de mañana (al mediodía, cuando se para para comer) y de tarde (al final de la tarde, cuando finaliza la última ponencia del día) puesto que se desaloja la sala y se entiende que todos los presentes la abandonan. Por último, pero no menos importante, de esta forma se evita tener que desalojar la sala después de cada ponencia, que era el mayor lastre. En el apartado 2.2.4 se realiza la estimación del ahorro, tanto en número lecturas como en tiempo, propiciado por la modificación una vez finalizadas las jornadas.

El siguiente ejemplo muestra el comportamiento de 2 estudiantes durante el periodo en el que transcurren 2 ponencias. El primer estudiante entra antes de que comience la primera, no abandona el salón de actos en el periodo entre ponencias y abandona la sala cuando termina la segunda, al final de la jornada. El segundo estudiante entra instantes después del comienzo de la primera ponencia, pero abandona la sala cuando todavía no se ha alcanzado la mitad de la segunda.

En este ejemplo, el número de lecturas son únicamente 3 (entrada del primer estudiante, entrada del segundo estudiante y salida del segundo estudiante) frente a las 4 lecturas necesarias si se hubiese utilizado el método anterior. Comparando el tiempo que los estudiantes están dentro del recinto con la extensión de las ponencias, podemos decir que primer estudiante ha asistido a las 2 ponencias, mientras que el segundo sí ha asistido a la primera al haber permanecido en la sala más del 80% de su duración, pero no se considera que haya asistido a la segunda al haberla abandonado antes de llegar a ese mínimo.

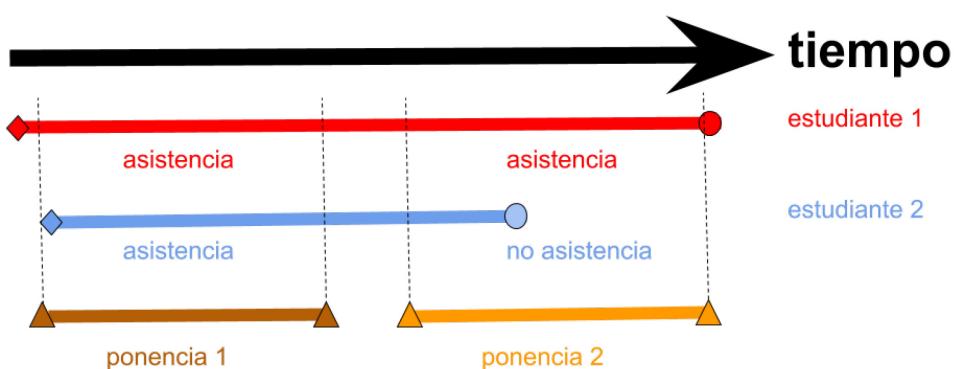


FIG. 2.9 – NUEVO ESQUEMA DE LECTURA

En resumen, este nuevo protocolo de medición es capaz de seguir controlando la asistencia individual a cada ponencia al mismo tiempo que consigue reducir el número de lecturas.

Estos cambios en el protocolo no implican modificaciones en la tecnología de lectura, en la arquitectura cliente-servidor ni en la estructura de red ad hoc. Pero sí obligan a un rediseño importante tanto en la base de datos como en la aplicación web.

2.2.3. Modificaciones en el sistema

Cambios en la base de datos

El nuevo protocolo de medición requiere recoger mucha más información que el anterior. Así, por ejemplo, ahora es necesario registrar los momentos de entrada y salida a las salas y de comienzo y fin de las conferencias. También resulta necesario agrupar las ponencias por bloques (de mañana y tarde para cada día) ya que, dentro de cada bloque, los estudiantes pueden entrar y salir en cualquier momento.

La tabla *asistentes* se amplía en un campo, *dentrofuera10*, necesario para identificar si el usuario se encuentra dentro o fuera de la sala.

```
CREATE TABLE `asistentes` (
  `dni` varchar(10) NOT NULL default '',
  `apellidos` varchar(50) NOT NULL default '',
  `nombre` varchar(50) NOT NULL default '',
  `dentrofuera10` tinyint(2) NOT NULL default '0',
  `asistencias` tinyint(2) NOT NULL default '0',
  `apto` tinyint(2) NOT NULL default '0',
  KEY `dni` (`dni`)
) TYPE=MyISAM;
```

La tabla *sesiones* también se amplía para recoger los instantes de comienzo y final de las mismas, así como otra información adicional. El campo *bloque* hace referencia al bloque en el que se ubica la ponencia y el campo *final* nos indica si se trata de la última ponencia del bloque (supone el desalojo automático de todos los estudiantes al finalizar) o no.

```
CREATE TABLE `sesiones` (
  `sesion` tinyint(2) NOT NULL auto_increment,
  `titulo` varchar(200) NOT NULL default '',
  `ponente` varchar(200) NOT NULL default '',
  `inicio_sesion` datetime NOT NULL default '0000-00-00 00:00:00',
  `inicio_ponencia` datetime NOT NULL default '0000-00-00 00:00:00',
  `final_sesion` datetime NOT NULL default '0000-00-00 00:00:00',
  `duracion` bigint(255) NOT NULL default '0',
  `estado` tinyint(2) NOT NULL default '0',
  `bloque` tinyint(2) NOT NULL default '0',
  `final` tinyint(1) NOT NULL default '0',
  KEY `sesion` (`sesion`)
) TYPE=MyISAM;
```

La tabla *control* se amplía añadiendo campos para identificar el bloque en curso en el momento de la identificación (campo *bloque*) así como para identificar si la identificación es de entrada o de salida (campo *entrasale10*).

```
CREATE TABLE `control` (
  `id` bigint(255) unsigned NOT NULL auto_increment,
  `dni` varchar(10) NOT NULL default '',
  `hora` datetime NOT NULL default '0000-00-00 00:00:00',
  `sesion` tinyint(2) NOT NULL default '0',
  `bloque` tinyint(2) NOT NULL default '0',
  `entrasale10` tinyint(2) NOT NULL default '0',
  KEY `id` (`id`)
) TYPE=MyISAM;
```

Se crean, además, 3 nuevas tablas. Por un lado, la tabla *bloques* guarda información sobre los distintos bloques horarios en los que se agrupan las ponencias. Por otro, la tabla *asistencia* guarda para cada estudiante y para cada ponencia, si el estudiante a asistido o no a la ponencia y durante cuánto tiempo. Por último, la tabla *config* guarda parámetros generales de configuración: sesión actual, porcentaje mínimo para computar asistencia, bloqueo del sistema y número total de sesiones durante las jornadas.

```
CREATE TABLE `bloques` (
  `bloque` tinyint(2) NOT NULL default '0',
  `titulo` varchar(50) NOT NULL default '',
  KEY `bloque` (`bloque`)
) TYPE=MyISAM;

CREATE TABLE `asistencia` (
  `id` bigint(255) unsigned NOT NULL auto_increment,
  `dni` varchar(10) NOT NULL default '',
  `sesion` tinyint(2) NOT NULL default '0',
  `asist` tinyint(1) NOT NULL default '0',
  `tiempo` bigint(255) NOT NULL default '0',
  KEY `id` (`id`)
) TYPE=MyISAM;

CREATE TABLE `config` (
  `id` tinyint(1) NOT NULL default '0',
  `sesion` tinyint(2) NOT NULL default '0',
  `asistencia` tinyint(4) NOT NULL default '0',
  `locked` tinyint(2) NOT NULL default '0',
  `sesiones` tinyint(2) NOT NULL default '0',
  KEY `id` (`id`)
) TYPE=MyISAM;
```

Cambios en la aplicación web

Aunque se sigue manteniendo como núcleo principal el formulario en el que se capturan las lecturas de las acreditaciones, la aplicación sufre profundos cambios. Tanto visuales, con un rediseño de las pantallas, como internos debido al incremento de complejidad a la hora de calcular la asistencia. El aspecto visual final de la aplicación, así como su código fuente (*HTML + CSS + PHP*) están disponibles en los anexos I y II.

2.2.4. Estimación de mejora tras las jornadas

Esta estimación se realiza una vez finalizadas las *III Jornadas InfoDIEZ*, a partir de los resultados registrados en la base de datos.

En primer lugar, se estima el tiempo de lectura de los códigos de barras. Para ello, se toma la tabla *control* de la base de datos y, en períodos de máxima frecuencia de lectura, se calcula el tiempo entre lecturas. Se obtiene una medida de 2,6 segundos para cada lectura.

A continuación, se calcula el ahorro en el número de lecturas. En las *III Jornadas InfoDIEZ* se registran 223 asistentes y se programan 19 ponencias. Con el sistema de control utilizado en las *II Jornadas InfoDIEZ*, todos los inscritos deben validar su presencia en cada una de las conferencias, lo que supone un total de 4.237 lecturas. Con el nuevo protocolo, el número de lecturas registradas es de 3.171, lo que supone una disminución del 25,16% del número de lecturas. Estas 1.066 lecturas no realizadas suponen un ahorro de lectura de 46 minutos y 12 segundos, que debe repartirse entre todos los puntos de identificación. Con 2 lectores de códigos de barras, el ahorro total es de aproximadamente 23 minutos.

Por otro lado, durante las *II Jornadas InfoDIEZ*, se estima que el tiempo necesario para desalojar completamente la sala (los estudiantes terminan de completar sus notas, se levantan, se mueven por los estrechos pasillos, cuello de botella en las puertas...) es de unos 7 minutos antes de poder comenzar a entrar en la sala de nuevo.

De las 19 ponencias de las *III Jornadas InfoDIEZ*, dejando fuera las últimas sesiones de mañana y tarde, es necesario desalojar la sala en 12 ocasiones, lo que supone un retraso estimado de 84 minutos.

Así, el ahorro de tiempo estimado durante las *III Jornadas InfoDIEZ* es de 107 minutos.

2.3. IV Jornadas InfoDIEZ (abril de 2006) [Introducción RFID]

2.3.1. Análisis de la situación

Al igual que en ocasiones anteriores, una vez finalizadas las *III Jornadas InfoDIEZ* se vuelve a realizar un análisis para contrastar si se han alcanzado los objetivos previstos, así como para sacar a la luz nuevos problemas y plantear posibles mejoras.

Lo primero que se observa, ya durante las jornadas, es que muchos asistentes optan por no abandonar la sala durante las pausas entre ponencias del mismo bloque. Esto representa un ahorro por dos vías distintas. Por un lado, se reduce el número de lecturas, mientras que por otro no se forman cuellos de botella en los accesos que provoquen retrasos añadidos. Tal y como se recoge en 2.2.4, se consigue reducir el número de lecturas en un 25% y los retrasos en 107 minutos.

En cuanto a la aplicación, durante el transcurso de las jornadas se detecta un problema de diseño, otra vez no previsto, que provoca que el sistema se salte una ponencia con el consiguiente descuadre. Afortunadamente, el error puede corregirse rápidamente reasignando las entradas y salidas a la ponencia correcta mediante una consulta *UPDATE* directamente en la base de datos. Inmediatamente se dan las indicaciones necesarias a los miembros de la organización responsables de la identificación para que el error no vuelva a repetirse durante el resto de las jornadas, dejándose la corrección de la aplicación pendiente hasta la siguiente iteración.

Una vez identificado el problema del salto de ponencias y cumplido el objetivo principal, el siguiente paso es preguntarse qué se puede seguir mejorando. De entre todas las posibilidades sugeridas durante el periodo de análisis se escoge la de automatizar el proceso de generación de certificados de asistencia y de superación de la prueba de evaluación escrita para la obtención de los créditos de libre elección.

Por último, nos preguntamos si todavía puede haber algún margen de mejora en cuanto a los tiempos de identificación. Manteniendo la tecnología de lectura de códigos de barras utilizada hasta el momento, no mucho. Vuelve a ponerse sobre la mesa la posibilidad de utilizar tecnología *RFID* para la identificación de asistentes. Ahora, dos años después haberlo descartado por primera vez ya nos encontramos ante una tecnología un poco más asentada y, sobre todo, bastante más asequible económicamente. Aunque todavía bastante lejos de encajar en el presupuesto de las jornadas.

Afortunadamente, después de mucho insistir se consigue el apoyo y la financiación del *Vicerrectorado de Estudiantes* y finalmente pueden adquirirse 2 lectores *NPR DESK II USB* (13.56 MHz, muy corto alcance) y 500 tags *RFID* (pasivas, formato tarjeta, *PVC*) por un importe total de 2.610€.



FIG. 2.10 – LECTOR NPR DESK II



FIG. 2.11 – TAGS RFID

2.3.2. Propuesta de mejora

En ese momento existen 3 frentes abiertos: la generación automática de certificados, la corrección del error que permitía saltar ponencias y, por último, la sustitución de los lectores de código de barras por lectores *RFID* en los puntos de identificación.

Generación automática de certificados

Anteriormente se ha indicado que una vez finalizadas las jornadas queda como resultado la tabla *asistentes* que guarda, para cada asistente, el número de ponencias a las que ha asistido y si esto representa más del 80% del total. Por otro lado, además de la asistencia, cada alumno debe realizar una prueba de evaluación objetiva, requerido por la Universidad de Zaragoza de cara a la concesión de créditos de libre elección.

Se plantea la creación de un *script* con las siguientes características:

- Para cada alumno inscrito, se comprueba si se ha superado el 80% de asistencia y la prueba de evaluación. Si se dan ambas condiciones, se deberá generar un certificado a efectos de la solicitud de créditos de libre elección. En el caso de que se haya superado la asistencia, pero no la prueba de evaluación, se deberá generar un certificado de asistencia.
- Los certificados se generan en formato *PDF* y se guardan en la base de datos, de forma que los alumnos pueden descargarlos desde su zona privada en la web de la asociación.

- En cada certificado se inserta una clave de validación única asociada al mismo que permite a terceros contrastar la validez de los mismos mediante un formulario habilitado a tal efecto.

La funcionalidad descrita se integra en la web de la asociación utilizando *PHP*, *MySQL* y la clase *FPDF*. Aunque el análisis detallado y la implementación de esta aplicación quedan fuera de la presente memoria al no ser objeto del proyecto fin de carrera, todo lo relativo a la misma queda a disposición del tribunal para posibles cuestiones y aclaraciones.

Corrección del error que permite saltar ponencias

Debido a los cambios en el protocolo de identificación introducidos en la última iteración, se hace necesario registrar el comienzo y final de cada ponencia para poder contrastarlo con la presencia del asistente en la sala.

III Jornadas InfoDIEZ

[sesiones.titulo] {título de la sesión actual}

Introduce el DNI: **Enviar**

[usuarios.nombre usuarios.apellidos] ha entrado en la sala ha abandonado la sala

Comenzar ponencia (si no está en curso) **Terminar ponencia** (si está en curso)

FIG. 2.12 – COMIENZO Y FINALIZACIÓN DE SESIONES

En la figura anterior puede verse un esquema del interfaz de la aplicación web. Además de la parte central dedicada lectura de acreditaciones, se incluye un botón en la parte inferior. Si hay una ponencia en curso, el botón toma la leyenda “Terminar ponencia”. Al pulsarlo se registra la finalización de la misma e, internamente, se pasa a la siguiente ponencia y se cambia el estado. Si no hay ponencia en curso, el botón toma la leyenda “Comenzar ponencia” y su pulsación supone registrar el comienzo de la misma y cambiar el estado.

El problema se presenta cuando, en el momento que termina una ponencia, se pulsa el botón de “Terminar ponencia” en 2 o más clientes. Con cada una de las pulsaciones se pasa a la siguiente conferencia, de forma que cada pulsación extra supone saltarse una conferencia.

La solución escogida para solucionarlo es mostrar únicamente estos botones en uno de los clientes, concretamente en el cliente situado en el mismo portátil en el que se encuentra el servidor, no deshabilitarlos en el resto de clientes. Al aparecer ahora el botón únicamente en uno de los clientes, resulta imposible pulsarlo más de una vez y desaparece el problema. Esta distinción cliente/servidor se lleva a cabo a través de la dirección *IP* asignada a cada uno de ellos.

```

$iplocal = $_SERVER["SERVER_ADDR"];
$ipremote = $_SERVER["REMOTE_ADDR"];

if ($iplocal==$ipremote)
{
    // La petición llega desde el servidor -> Mostrar botones
}
else
{
    // La petición llega desde un cliente -> No mostrar botones
}

```

Identificación mediante lectores *RFID*

La sustitución de los lectores de códigos de barras por lectores *RFID* implica realizar cambios en 3 ámbitos distintos: captura de identificadores, gestión de las acreditaciones y asignación de las mismas a los asistentes y, por último, cambios en la aplicación y en la base de datos para adaptarse al nuevo escenario.

Captura de identificadores *RFID*

Los lectores *RFID* adquiridos, *NPR DESK II*, se conectan a los ordenadores portátiles a través de un puerto *USB* y vienen acompañados de un *driver* para *Windows XP* y de una *API* en varios lenguajes de programación (C, Java...) para poder capturar las lecturas desde el PC. Desgraciadamente, entre los lenguajes de programación de la *API* no se encuentra *PHP*.

Por otro lado, como se ha visto anteriormente, la aplicación web está diseñada para recibir las lecturas a través del teclado. Cambiar el mecanismo de captura requeriría, una vez más, de cambios profundos en la aplicación.

La solución para evitar rehacer completamente la aplicación, en este caso, llega de la mano del proveedor. Nos ponemos en contacto con él y le preguntamos si sería posible modificar el *driver* para que devolviese las lecturas directamente a la entrada de teclado, emulando el comportamiento del lector de código de barras. Pocos días después nos hacen llegar el nuevo *driver* con el que ya podemos capturar los identificadores de los *tags RFID* sin necesidad de mayores cambios en la aplicación web.

Sin embargo, en el cambio de *driver* hay un compromiso implícito: a cambio de simplificar la lectura, se tiene que renunciar a la funcionalidad que ofrece la *API*. Concretamente, se pierde el control sobre la sensibilidad de la lectura, así como sobre la protección ante las lecturas múltiples. Durante el proceso de testeo se encuentra que, al ser la nuestra una lectura por proximidad, durante el proceso de acercar y alejar los *tags* del lector se pueden producir varias lecturas dependiendo de la velocidad con la que se ejecute el movimiento. De todas esas lecturas, tan solo debería ser considerada la primera. El resto tienen que ser descartadas para evitar distorsiones en la ubicación de los asistentes. Para ello, basta con añadir la estructura de control necesaria en la aplicación web, de forma que cualquier lectura del mismo usuario en un intervalo de 60 segundos es ignorada.

```

if (time(ahora) - time(último registro del mismo usuario)<60)
{
    // Si no han pasado 60'' desde la última lectura, se descarta
}
else
{
    // Si han pasado más de 60'' desde la última lectura, se procesa
}

```

Gestión de las acreditaciones

Hasta el momento las acreditaciones contenían, escrito en formato código de barras, el DNI del asistente, que actuaba como identificador único de cara al control de presencia. Con el cambio a tecnología *RFID* va a ser necesaria una modificación.

El abanico de posibilidades dentro de *RFID* es muy amplio, tanto en distancia de lectura que puede ir desde apenas un milímetro hasta cientos de metros, como en capacidades lectura/escritura y memoria de los *tags*, dependiendo de factores como la frecuencia de trabajo, el tipo y configuración de antenas o el tipo de *chip* que se use en el *tag*.

El conjunto de lectores/*tags* adquirido se ubica en el rango de muy corta distancia (pocos milímetros). Los *tags* carecen de memoria interna y la única operación que permiten es la lectura de un identificador hexadecimal único integrado en el *chip*. Por esta razón ya no es viable utilizar el DNI como identificador de los asistentes. La única posibilidad es asociar cada asistente con uno de los *tags* *RFID* a través de su identificador hexadecimal.

Para ello se añade un campo *asistentes.rfid* para almacenar estos identificadores y se crea una pequeña aplicación web auxiliar (ver Fig. I.1) que recorre la tabla *asistentes* y, para cada inscrito que todavía no tenga asignada un identificador *RFID*, se lo pide y una vez leído lo guarda en el campo del mismo nombre.

Los *tags* *RFID* de PVC no tienen ningún identificador visible por lo que resulta imprescindible, inmediatamente después de haber realizado la asociación, montar la acreditación. La acreditación está formada por el *tag* *RFID* acompañado de la parte visible de la acreditación, impresa en una hoja de papel de su mismo tamaño. Ambas partes de la acreditación se mantienen juntas metiéndolas en una funda de plástico, que es lo que finalmente se entrega a los asistentes.

2.3.3. Modificaciones en el sistema

Cambios en la base de datos

En la base de datos, el único cambio necesario es añadir el campo del identificador *RFID* a todas las tablas que requieren un identificador del asistente.

```
CREATE TABLE `asistentes` (
  `dni` varchar(10) NOT NULL default '',
  `rfid` varchar(25) NOT NULL default '',
  `apellidos` varchar(50) NOT NULL default '',
  `nombre` varchar(50) NOT NULL default '',
  `dentrofuera10` tinyint(2) NOT NULL default '0',
  `asistencias` tinyint(2) NOT NULL default '0',
  `apto` tinyint(2) NOT NULL default '0',
  KEY `dni` (`dni`)
) TYPE=MyISAM;

CREATE TABLE `asistencia` (
  `id` bigint(255) unsigned NOT NULL auto_increment,
  `dni` varchar(10) NOT NULL default '',
  `rfid` varchar(25) NOT NULL default '',
  `sesion` tinyint(2) NOT NULL default '0',
  `asist` tinyint(1) NOT NULL default '0',
  `tiempo` bigint(255) NOT NULL default '0',
  KEY `id` (`id`)
) TYPE=MyISAM;
```

```

CREATE TABLE `control` (
  `id` bigint(255) unsigned NOT NULL auto_increment,
  `dni` varchar(10) NOT NULL default '',
  `rfid` varchar(25) NOT NULL default '',
  `hora` datetime NOT NULL default '0000-00-00 00:00:00',
  `sesion` tinyint(2) NOT NULL default '0',
  `bloque` tinyint(2) NOT NULL default '0',
  `entradas10` tinyint(2) NOT NULL default '0',
  KEY `id` (`id`)
) TYPE=MyISAM;

```

Cambios en la aplicación web

Además de las modificaciones destinadas a la protección frente al cierre simultáneo de ponencias y a la protección frente lecturas múltiples reseñadas anteriormente, el único cambio que hay que realizar en la aplicación web es sustituir todas las referencias a la identificación por DNI (campos *asistentes.dni*, *asistencia.dni* y *control.dni*) por el nuevo campo destinado a la identificación por *RFID* (campos *asistentes.rfid*, *asistencia.rfid* y *control.rfid*).

El resto de la aplicación se mantiene exactamente igual que en la iteración anterior.

2.3.4. Estimación de mejora tras las jornadas

Utilizando el mismo sistema que en el apartado 2.2.4, se estima que el tiempo de lectura utilizando lectores *RFID* es de 2,0 segundos. Comparados con los 2,6 segundos del lector de código de barras, supone un total de 0,6 segundos de ahorro en cada lectura, un 23,08%.

Si se toman las mismas 3.171 lecturas que en la edición anterior, se obtiene un ahorro de lectura de 31 minutos y 42 segundos, también a repartir entre todos los puntos de identificación. Con 2 lectores de *RFID*, el ahorro total es de aproximadamente 16 minutos. Menos de 1 minuto por ponencia.

Como puede apreciarse, el ahorro propiciado por el cambio de tecnología de lectura es muy inferior al tiempo ganado al cambiar el protocolo. Aunque siempre es bienvenido, el ahorro en tiempo en este caso hubiese podido resultar insuficiente para justificar la inversión necesaria.

2.4. I Jornadas i+DIEZ y siguientes (diciembre de 2006 – abril de 2010)

Pocas semanas después de la celebración de las *IV Jornadas InfoDIEZ*, el autor de la aplicación web cesa su actividad en *Dataria Comunicación, S.L.* siguiendo nuevas oportunidades profesionales y abandonando el desarrollo activo de la aplicación de control de asistencia. Así pues, a partir de ese momento la aplicación web deja de recibir correcciones y mejoras. Se continúa, no obstante, ofreciendo soporte puntual en caso de necesidad.

La siguiente ocasión en la que se utiliza la aplicación web es en las *I Jornadas i+DIEZ*, en diciembre de 2006. Estas jornadas tienen un formato más corto, únicamente durante 2 días, y trasladan su sede a una nueva ubicación: el salón de actos de la *Escuela Universitaria de Estudios Empresariales de Zaragoza*. A partir de ahí se van alternando sucesivas ediciones de *InfoDIEZ* e *i+DIEZ* hasta llegar abril de 2010, cuando se celebra la *VIII* y última edición de las jornadas.

InfoDIEZ. Durante todo este periodo no se reportan incidencias ni casos de mal funcionamiento del sistema, lo cual confirma su robustez y capacidad de adaptación a cualquier ubicación.

Ante la falta de nueva actividad, la asociación *Grupo DIEZ* cede la aplicación, así como la infraestructura necesaria para hacerla funcionar a la empresa *Univolution Servicios Avanzados, S.L.*, *spin-off* de la Universidad de Zaragoza surgida como evolución natural de la asociación. La aplicación pasa a formar parte de la cartera de productos de la empresa. Desafortunadamente, la empresa está mucho más centrada en otras áreas de negocio (vivienda, comercio electrónico, publicidad en Internet...) y nunca se le llega a dar salida comercial.

3. Una mirada al presente

Tal y como se indica en el apartado anterior, la aplicación se mantuvo en producción al menos hasta abril de 2010. Desde entonces hasta hoy se han producido importantes avances en el entorno tecnológico que nos hacen preguntarnos sobre la validez de la propuesta a día de hoy. ¿Siguen siendo vigentes las soluciones planteadas o, por el contrario, han quedado obsoletas?

La aplicación no debe analizarse únicamente desde un punto de vista puramente tecnológico, sino también desde su capacidad para dar respuesta a situaciones que requieren control de presencia en la actualidad.

3.1. Vigencia tecnológica

Los últimos cambios introducidos en el diseño del sistema de control de presencia tratan de 2006. Desde entonces se han producido importantes avances en ámbitos directamente relacionadas con los distintos componentes del sistema (lectura automatizada, identificación, redes inalámbricas, servidores...) que debemos tener en cuenta a la hora de validar la vigencia tecnológica de la aplicación.

3.1.1. Tecnologías de lectura e identificación

Las tecnologías de lectura analizadas en el apartado 2.1.1, óptica y por radiofrecuencia, siguen plenamente vigentes en la actualidad. Potenciadas, si cabe, gracias a la irrupción de los *smartphones* en el mercado.

Identificación óptica

En el caso de identificación óptica, además de dispositivos de lectura autónomos que todavía siguen vigentes, la introducción de cámaras de alta resolución en los dispositivos móviles permite utilizar prácticamente cualquier terminal, incluso en las gamas más bajas, para la lectura de códigos de barras o bidimensionales.

Identificación por radiofrecuencia (RFID)

El salto en cuanto a tecnologías de identificación por radiofrecuencia ha sido todavía más pronunciado. Con el paso de los años la tecnología ha madurado y se ha hecho cada vez más cercana y económicamente asequible. A modo de ejemplo, adquirir hoy en día lectores autónomos comerciales y *tags* equivalentes a los que se adquirieron en 2006 tendría un coste aproximado de unos 150€, frente a los 2.610€ que costaron en su momento.

Se podría optar incluso a la creación de lectores propios a medida, utilizando otras tecnologías disponibles, como por ejemplo *Arduino*, que permiten de manera sencilla y muy económica la integración de lectores *RFID* en placas personalizadas.

Al igual que sucede con las tecnologías de identificación óptica, la identificación por radiofrecuencia también ha llegado a los *smartphones*. Aunque su uso todavía no está tan extendido como el de las cámaras, la mayor parte de los terminales móviles actuales de gama media incluyen ya tecnología *NFC*, un estándar de comunicación inalámbrica de muy corto alcance basado en *RFID* con capacidad de lectura y escritura de *tags*.

Los *tags RFID* también han evolucionado a lo largo de los últimos años. Los *tags* actuales, además de un identificador hexadecimal individual, disponen de una memoria de lectura/escritura, así como distintas posibilidades de cifrado, bloqueo de escritura y seguridad. Esta capacidad de escritura permitiría guardar distintos parámetros de la aplicación. Así, por ejemplo, podría guardarse directamente el DNI y utilizarlo como clave de identificación o, incluso, registrar en el propio *tag* los *timestamps* de entradas y salidas, así como la progresión de la asistencia durante las jornadas.

Identificación biométrica

Otro campo en el que se ha avanzado en los últimos años es el de la identificación biométrica. En lugar de utilizar códigos de barras o *tags RFID*, sería el propio asistente (huellas dactilares, rasgos faciales, voz, retina...) el que validase su propia presencia. Estas tecnologías, sin embargo, todavía no están lo suficiente maduras como para poder ser utilizadas a un precio asequible, sobre todo en comparación con las anteriores que son baratas, robustas y muy contrastadas.

3.1.2. Conectividad

La conectividad es otra de las áreas en las que el cambio percibido en los últimos años ha sido radical. La progresiva implantación de cobertura *WiFi* en espacios comunes, públicos y privados, asegura la disponibilidad de acceso a red *IP* y conexión a Internet prácticamente en cualquier espacio habilitado para conferencias.

Por si no fuera suficiente, los avances en telefonía móvil también se dejan notar en este apartado. Por un lado, las redes de datos móviles son muchísimo más rápidas y a unos precios ridículos si lo comparamos con la velocidad y precio de 2006. Esto permite que cualquier terminal móvil tenga la capacidad de conectarse a Internet desde prácticamente cualquier lugar con unas condiciones aceptables.

Por otro lado, todos los terminales móviles en el mercado disponen de conectividad inalámbrica *WiFi*, lo que permitiría convertirlos en clientes del sistema o incluso realizar las funciones de servidor gracias a su capacidad para actuar como punto de acceso inalámbrico.

Sin embargo, ninguno de estos avances es capaz de garantizar la existencia, en cualquier momento y lugar, de la estructura de red necesaria para conectar los distintos puntos de identificación. Todavía quedan muchos recintos sin cobertura *WiFi*, espacios cerrados con mala cobertura móvil, zonas fuera de las grandes ciudades sin un buen acceso a datos. Incluso cada vez es más frecuente encontrar espacios públicos con inhibidores de telefonía móvil. El uso de redes *ad hoc*, que no dependen de ningún condicionante externo, sigue resultando imprescindible.

3.1.3. Arquitectura cliente-servidor

En la propuesta original se establecían los puntos de identificación como el conjunto de 2 elementos: el dispositivo de lectura, encargado de capturar el identificador del asistente, y un ordenador portátil para ejecutar la aplicación web que recoge esa lectura, se conecta al servidor para procesarla y muestra los resultados por pantalla. Según se acaba de ver en los apartados 3.1.1 y 3.1.2, en la actualidad contamos con un dispositivo, el *smartphone*, capaz de realizar simultáneamente ambas tareas, lo cual lo convierte en un cliente ideal.

Con el mismo argumento, podríamos decir lo mismo de clientes personalizados basados en *Arduino*, o cualquier otra configuración que integrase ambos componentes.

En cuanto al servidor, la tendencia actual apunta a la reducción de los servidores locales y su sustitución por el uso de servicios en la nube permanentemente accesibles. Esta opción, sin embargo, debe descartarse puesto que va en contra del objetivo de diseño original de independencia de funcionamiento ante la posibilidad de no disponer de acceso a Internet.

Sigue siendo necesaria la utilización de un servidor local, tanto para generar la red ad hoc como para actuar como servidor de la aplicación, existiendo varias alternativas para ello:

- Seguir utilizando un ordenador portátil
- Usar un *smartphone*
- Usar microordenadores, tipo *Raspberry Pi*, alimentados por batería.

Las alternativas para motor del servidor de la aplicación y de la base de datos se han multiplicado y ramificado tanto en los últimos años que enumerarlas y compararlas podrías ser objeto de un nuevo proyecto fin de carrera. La tendencia actual invita a la implantación de un servicio web con *RESTful API* sobre una arquitectura *REST*, de forma que la comunicación cliente-servidor se realice íntegramente mediante mensajes *HTTP*.

Esta *RESTful API* puede llevarse a cabo con cualquier tecnología de servidor, pudiendo utilizarse *stacks* más tradicionales como *Apache + PHP + MySQL* como estructuras basadas en servidores asíncronos como *Node.js* apoyada en bases de datos NoSQL como *MongoDB* o *Cassandra*. Este último tipo de sistemas están optimizados para ofrecer respuestas de forma muy rápida en situaciones de alta concurrencia y ser fácilmente escalables. Para un proyecto como este, con pocas peticiones al servidor, no sería necesario llegar tan lejos en términos de eficiencia.

3.1.4. Desarrollo de la aplicación

Una vez más, el principal cambio en cuanto a aplicaciones ha venido del mundo de la telefonía móvil. Los *smartphones* actuales permiten la instalación y ejecución de aplicaciones de todo tipo. A diferencia de las aplicaciones web, limitadas por las características del navegador, estas aplicaciones nativas permiten el acceso a todas las características del terminal como pueden ser la cámara, el *GPS* o el lector *NFC*.

Además de los *SDK* oficiales para los principales sistemas operativos móviles (*Android*, *iOS*, *Windows 10*), existen múltiples *frameworks* para la creación de aplicaciones móviles. Todos

ellos tienen en común la disponibilidad de bibliotecas de desarrollo para poder acceder a estas características de los terminales móviles.

De esta forma sería factible, por ejemplo, realizar una aplicación que leyese un *tag NFC* y enviase la lectura al servidor de una forma relativamente sencilla.

3.2. Propuesta de aplicación en la actualidad

Teniendo en cuenta todas las posibilidades analizadas a lo largo del apartado 3.1, posiblemente el diseño más sencillo en términos materiales y económicos que permite seguir cumpliendo con todos los requisitos de diseño sea el que se detalla a continuación.

En cuanto a los puntos de identificación, la elección parece sencilla. Prácticamente cualquier smartphone disponible en la actualidad integra, en un único dispositivo de tamaño muy reducido, las siguientes características:

- Capacidad de identificación, a través de la cámara o de tecnología *NFC*.
- Conectividad necesaria para integrarse en, o crear, una red ad hoc inalámbrica.
- Capacidad para ejecutar aplicaciones, sea de forma nativa o a través de un navegador.
- Pantalla táctil para poder utilizar el interfaz de la aplicación
- Batería, con una duración muy superior a la de los ordenadores portátiles tradicionales.

Si comparamos estas características, tanto por tamaño como por precio como por autonomía de la batería, con el conjunto de un lector *RFID* externo conectado a un ordenador portátil rápidamente nos damos cuenta de que, escogiendo un *smartphone* como lector, disponemos de un punto de identificación equivalente pero mucho más manejable y económico.

Así pues, se propone escoger terminales móviles con sistema operativo *Android* equipados con tecnología *NFC* (*iOS* todavía permite el uso libre de *NFC*) para ser utilizados como clientes. Debido a este cambio, se propone también desvincular el servidor de los clientes y montar el servidor, que tendrá otras características, de forma independiente.

En cuanto a la estructura de la red, sigue siendo necesario el uso de una red ad hoc para poder garantizar los requisitos de diseño. Aunque se podría utilizar también un terminal móvil para generarla, no es la mejor opción debido a que el soporte tanto de *Android* como de *iOS* para software de servidor es todavía bastante limitado. Así pues, se propone seguir usando un ordenador portátil ya sea en su forma completa o de forma compacta (*Raspberry Pi* o similar) alimentada por batería.

Si ya se dispone de un ordenador portátil destinado a este propósito, puede seguir utilizándose sin ningún impedimento. Si hubiese que adquirir uno, sería más económica la segunda alternativa. En cualquiera de los 2 casos habría que configurar la red ad hoc de manera equivalente al procedimiento mostrado en el apartado 2.1.2. (*hostednetwork* a partir de *Windows 7*, *hostapd* en servidores *Linux*) de la memoria.

En cuanto a la arquitectura del servidor, la propuesta sería utilizar una arquitectura *REST* a través de *RESTful API* sobre *Apache + PHP + MySQL*, para poder aprovechar el código y las consultas desarrolladas con anterioridad. El servidor se encargará de procesar y responder las

peticiones de los clientes e interactuar con la base de datos, dejando en manos del cliente cuestiones como el interfaz o la usabilidad de la aplicación.

Por último, habría que desarrollar la aplicación móvil para los terminales *Android*. La aplicación capturará la lectura de los *tags RFID* de los asistentes y los enviará al servidor, así como las indicaciones sobre comienzo/final de las distintas ponencias.

3.3. Otros usos de control de presencia

Además de la vigencia tecnológica de la aplicación de control de presencia, cabe preguntarse también sobre la vigencia operativa: ¿resulta pertinente un sistema de control de presencia en la actualidad?

La respuesta, en mi opinión, es sí. Sobre todo, añadiendo mecanismos de verificación de identidad, lo que permitiría además poder dar el salto a un control desatendido de presencia. Todavía hay multitud de situaciones cotidianas en las que resulta necesario controlar y verificar la presencia de los agentes participantes. Por indicar algunos ejemplos representativos:

- Asistencia a clase en educación obligatoria. Se controla la presencia de los estudiantes en cada clase, incluso con la posibilidad de alertar a los padres en tiempo real.
- Validación de acceso en eventos, ferias, congresos. Una vez verificada la identidad del asistente, el sistema puede validar a qué se tiene acceso y a qué no.
- Asambleas ciudadanas, en distintos ámbitos locales.
- Registro de las horas de trabajo en empresas, sobre todo ahora que la seguridad social exige el registro diario de las horas de trabajo.
- ...

Para muchas de estas aplicaciones, sobre todo las que tienen que ver con administraciones públicas, sería interesante disponer de un tag identificador único, para evitar tener que utilizar un *tag* distinto en cada ámbito. Sería mucho más cómodo, pero al mismo tiempo la idea de tener un *tag* que permita controlar nuestra presencia e identificarnos está reñida con el concepto de privacidad en una sociedad hiperconectada. Se plantea un debate que va mucho más allá de lo tecnológico.

4. Conclusiones y líneas futuras

Tal y como se recoge en la introducción, el principal objetivo de este proyecto es contrastar la vigencia tecnológica en la actualidad de una aplicación cuyo diseño comenzó hace ya más de 13 años. Y, tras haber analizado los diferentes avances tecnológicos producidos a lo largo de los últimos años en el capítulo 3, puede decirse que la aplicación sigue siendo plenamente vigente.

Por supuesto, el paso del tiempo ha traído tecnologías mucho más potentes, eficientes y, sobre todo, mucho más económicas. Podemos decir que tecnologías más modernas han sustituido a otras que han quedado atrás, pero esta sustitución no ha cambiado las estructuras subyacentes, que siguen plenamente vigentes.

Así, un sencillo terminal móvil de gama media como el que cualquiera de nosotros tiene a escasos centímetros de su mano es capaz de sustituir y mejorar a varios componentes del sistema original. Sin embargo, este móvil realiza exactamente la misma función que los componentes originales.

Después de 13 años, la tecnología *RFID/NFC* está plenamente integrada en nuestro día a día (autobuses, pagos, ...) y es la tecnología de lectura más rápida y robusta, manteniendo un coste asequible, que puede encontrarse en la actualidad. De la misma forma, sigue siendo plenamente vigente el concepto de red ad hoc para garantizar la conectividad de los distintos nodos sin tener que depender de ningún elemento de red externo al sistema.

Desde el lado de los servidores, tampoco parece que las tecnologías propuestas hayan perdido vigencia: aunque ahora se utilicen arquitecturas *REST* para definir la comunicación entre cliente y servidor, el modelo cliente-servidor sigue manteniendo su validez. En el caso de los servidores web y de base de datos utilizados, a día de hoy *Apache* sigue siendo el servidor web más utilizado del mundo mientras *MySQL* es el segundo motor de base de datos más utilizado del mundo, a muy corta distancia del líder *Oracle*.

En relación con el desarrollo de la aplicación, más allá de haber logrado alcanzar los objetivos de diseño planteados, debo decir que supuso para mí una gran experiencia de aprendizaje. Esta fue la primera ocasión en la que me enfrentaba a un proyecto completo, desde el análisis previo a la implementación definitiva, sin guías ni supervisión de ningún tipo, lejos del paraguas de la universidad. Aprendizaje en la gestión de proyectos, pero también aprendizaje en la mayor parte de las tecnologías utilizadas. El desarrollo de esta aplicación supuso mi primera toma de contacto con servidores web y de bases de datos, mis primeras consultas en lenguaje *SQL* y mis primeras líneas de código en *HTML* y *PHP*.

Con respecto al futuro, hay dos líneas que me hubiese gustado poder explorar, pero que por distintas circunstancias nunca llegué a hacer:

- Integración de mecanismos de verificación de identidad para poder convertir el sistema en desatendido.
- Añadir un canal de comunicación interactivo en tiempo real entre la organización y los asistentes.

Tal vez ahora, que después de tantos años he vuelto a retomar este proyecto, sea un buen momento para ponerlas en marcha.

5. Bibliografía

- MS9500 User's Guide
<https://www.honeywellaidc.com/en-GB/-/media/en/files-public/technical-publications/barcode-scanners/voyager-9520-voyagercg-9540/00-02410.pdf>
- MetroSelect Single-Line Configuration Guide
<https://www.honeywellaidc.com/en-GB/-/media/en/files-public/technical-publications/multi-product/00-02544%20Rev%20K%202-11%20pdf.pdf>
- Sambar Server Documentation (accesible a través de archive.org)
<http://www.sambar.com:80/syshelp/index.htm>
- PHP Documentation
<http://php.net/manual/en/>
- MySQL Documentation
<https://dev.mysql.com/doc/>
- Apache Documentation
<https://httpd.apache.org/docs/>
- HTML Standard
<https://html.spec.whatwg.org/multipage/>
- CSS Standard
<https://www.w3.org/Style/CSS/>
- Ad Hoc Mobile Wireless Networks: Protocols and Systems
Toh, C.K.. Ad Hoc Mobile Wireless Networks: Protocols and Systems. Prentice Hall, 2001
- RFID: Radio Frequency Identification
Shepard, S. RFID: Radio Frequency Identification, New York. McGraw Hill, 2005
- La implantación de RFID en la empresa española
Sáenz, M.J. Pilot 2006, Zaragoza Logistics Center, 2006
- August 2017 Web Server Survey
<https://news.netcraft.com/archives/2017/08/29/august-2017-web-server-survey.html>
- DB-Engines Ranking
<https://db-engines.com/en/ranking>

Anexo I. Funcionamiento de la aplicación: manual de usuario

En esta sección se ilustra, a través de figuras, el funcionamiento de la versión final de la aplicación.

I. 1. Carga de identificadores RFID



FIG. I.1 – CARGA DE IDENTIFICADORES RFID

La figura I.1. ilustra la aplicación auxiliar diseñada para la carga y asociación de los identificadores *RFID* a los asistentes de las jornadas. La aplicación recorre la base de datos de asistentes y, para cada asistente que no tuviese identificador *RFID* asociado lo solicita y lo añade a la tabla correspondiente.

1.2. Pantalla de arranque, antes del comienzo las jornadas



FIG. I.2 – PANTALLA DE ARRANQUE, ANTES DEL COMIENZO DE LAS JORNADAS

La figura I.2 ilustra el estado de la aplicación en el servidor antes de comenzar las jornadas. En este momento, el sistema está cerrado y no se registran las lecturas en los puntos de identificación.

El registro de usuarios se abre al pulsar el botón “Comenzar Sesión”.

1.3. Clientes a la espera de que se abra la lectura



FIG. I.3 – CLIENTES A LA ESPERA DE QUE SE ABRA LA LECTURA

La figura I.3 ilustra el comportamiento de los clientes antes de comenzar las jornadas. Mientras el sistema permanece cerrado, no se permite la lectura de acreditaciones desde ninguno de los clientes. Tampoco se permite la opción de abrir la sesión, que debe realizarse necesariamente desde el servidor.

I.4. Control de acceso, antes del comienzo de una ponencia



FIG. I.4 – CONTROL DE ACCESO, ANTES DEL COMIENZO DE UNA PONENCIA

La figura I.4 ilustra el comportamiento de la aplicación en el servidor durante los minutos previos al comienzo de una ponencia. En este momento, la ponencia todavía no ha comenzado, pero se permite la entrada/salida de asistentes a la sala.

En la cabecera puede verse la información (número de sesión, título de la conferencia y nombre del ponente) relativa a la ponencia que está a punto de comenzar.

En el momento en que comienza la ponencia, se registra su comienzo mediante el botón "Comenzar Ponencia".

1.5. Control de acceso, antes del comienzo de una ponencia (clientes)



FIG. I.5 – CONTROL DE ACCESO, ANTES DEL COMIENZO DE UNA PONENCIA (CLIENTES)

La figura I.5 ilustra el comportamiento de los clientes durante los minutos previos al comienzo de una ponencia. Su comportamiento es exactamente igual al del servidor, con la excepción de que no aparece el botón “Comenzar ponencia”.

La ausencia de botones para controlar el flujo de las ponencias es algo que se va a repetir en el resto de estados de la aplicación.

I.6. Lectura de una acreditación (entrada)



FIG. I.6 – LECTURA DE UNA ACREDITACIÓN (ENTRADA)

La figura I.6 ilustra el comportamiento de la aplicación cada vez que se registra la lectura de la acreditación de un asistente que, en ese momento, se encontraba fuera de la sala. En la pantalla aparece el mensaje “El usuario ha entrado en la sala”.

La pantalla equivalente en los clientes no muestra el botón de cambio de estado.

I.7. Lectura de una acreditación (salida)



FIG. I.7 – LECTURA DE UNA ACREDITACIÓN (SALIDA)

La figura I.7 ilustra el comportamiento de la aplicación cada vez que se registra la lectura de la acreditación de un asistente que, en ese momento, se encontraba dentro de la sala. En la pantalla aparece el mensaje “El usuario ha salido de la sala”.

La pantalla equivalente en los clientes no muestra el botón de cambio de estado.

I.8. Control de acceso en el transcurso de una ponencia



FIG. I.8 – CONTROL DE ACCESO EN EL TRANSCURSO DE UNA PONENCIA

La figura I.8 ilustra el comportamiento de la aplicación mientras la ponencia se encuentra en curso. Durante este periodo se permite la entrada/salida de asistentes a la sala.

En la cabecera puede verse la información (número de sesión, título de la conferencia y nombre del ponente) relativa a la ponencia en curso.

En el momento en que la ponencia en curso finaliza, se registra su finalización mediante el botón “Terminar Ponencia”.

La pantalla equivalente en los clientes no muestra el botón de cambio de estado.

I.9. Control de acceso al finalizar una ponencia



FIG. I.9 – CONTROL DE ACCESO AL FINALIZAR UNA PONENCIA

La figura I.9 ilustra el comportamiento de la aplicación cuando ha finalizado una ponencia y el sistema se encuentra a la espera de la siguiente. Durante este periodo se permite la entrada/salida de asistentes a la sala.

En la cabecera puede verse la información (número de sesión, título de la conferencia y nombre del ponente) relativa a la ponencia que está a punto de comenzar. Se muestra también un texto relativo a la ponencia que acaba de terminar.

En el momento en que comienza la siguiente, se registra su comienzo mediante el botón "Comenzar Ponencia".

La pantalla equivalente en los clientes no muestra el botón de cambio de estado.

I.10. Control de acceso al finalizar una ponencia (fin de bloque)



FIG. I.10 – CONTROL DE ACCESO AL FINALIZAR UNA PONENCIA (FIN DE BLOQUE)

La figura I.10 ilustra el comportamiento de la aplicación cuando ha finalizado una ponencia con la que se finaliza un bloque (última ponencia de la mañana o de la tarde). Cuando se llega a este estado, se registra de forma automática la salida de todos los asistentes que en ese momento se encontraban dentro de la sala. En este momento no se permite la lectura de acreditaciones.

La pantalla equivalente en los clientes no muestra el botón de cambio de estado.

I.11. Control de acceso al término de las jornadas



FIG. I.11 – CONTROL DE ACCESO AL TÉRMINO DE LAS JORNADAS

La figura I.11 ilustra el comportamiento de la aplicación una vez se ha alcanzado el final de las jornadas. Cuando se llega a este estado, se evalúa la asistencia de los inscritos a las jornadas. En este momento no se permite la lectura de acreditaciones ni el cambio de estado.

Anexo II. Código fuente de la aplicación y base de datos

II.1. Diagrama de estados de la aplicación

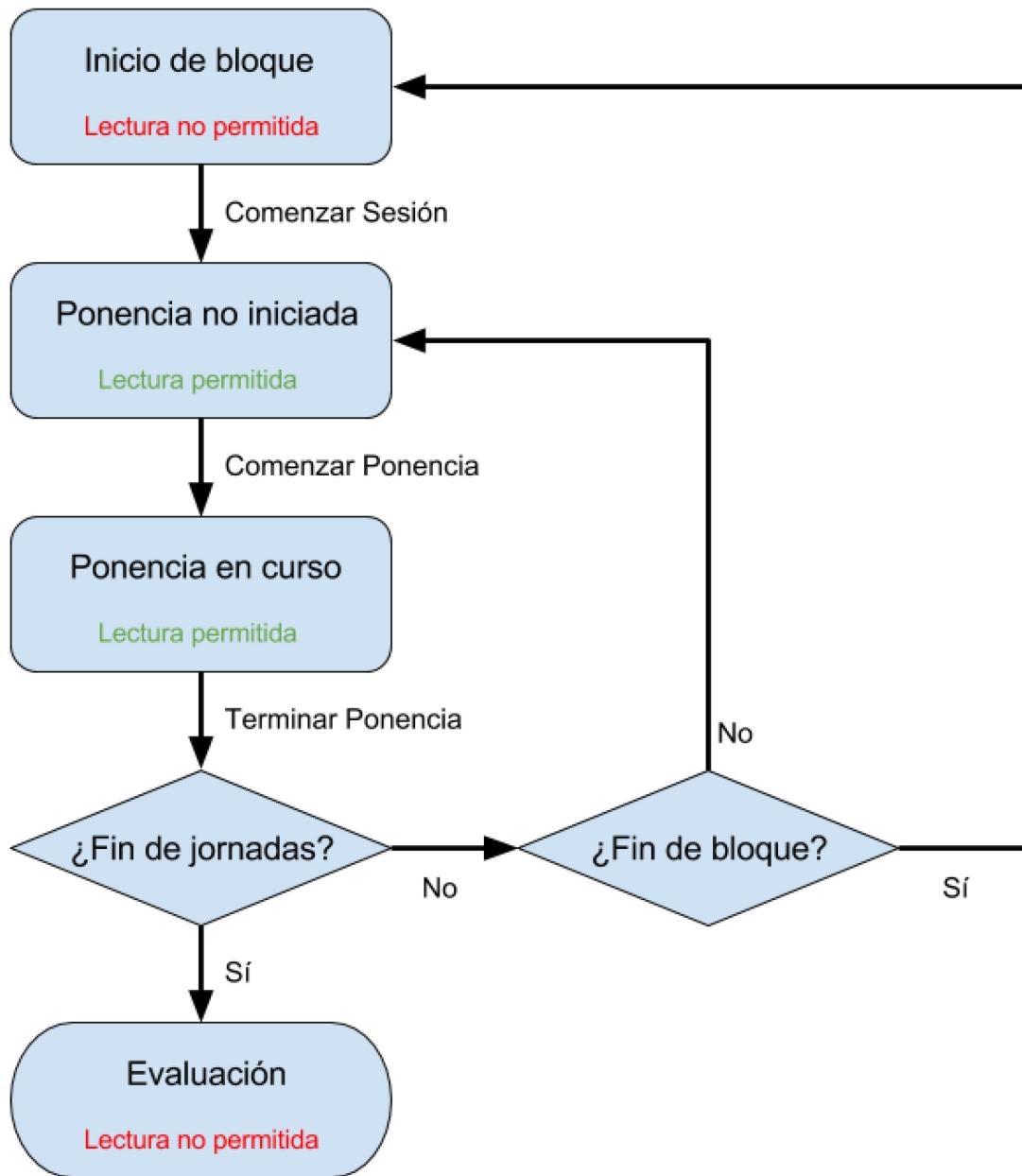


FIG. II.1 – DIAGRAMA DE ESTADOS DE LA APLICACIÓN

La figura II.1 muestra el diagrama de estados de la aplicación. El flujo se corresponde con el avance de las distintas ponencias a lo largo de las jornadas. Para cada estado, se indica si la lectura de acreditaciones está permitida o no.

II.2. Código fuente de la aplicación

La aplicación de control de presencia se compone de los siguientes ficheros:

- index.php – fichero principal de la aplicación
- config.php – fichero de configuración y conexión con base de datos
- funciones.php – fichero con funciones para generar cabecera y pie de página
- estilos.css – fichero con el CSS
- ficheros de imágenes en formatos .jpg y .gif

index.php

```

<?
//Configura el acceso a la base de datos
require("config.php");
include("funciones.php");
$db_connection=mysql_connect($host,$user,$pass) or die("Error en la conexión con la Base de
Datos");
mysql_select_db($db);

//Obtiene las variables que se usan a lo largo del script
$sql = "SELECT c.locked,c.sesion,c.asistencia,s.titulo,s.pONENTE,s.estado,s.bloque,s.final FROM
config c INNER JOIN sesiones s ON c.sesion = s.sesion";
$consulta = mysql_query($sql) or die("Error en la consulta $sql");
$res = mysql_fetch_array($consulta);
mysql_free_result($consulta);
$sesion = $res['sesion'];
$titulo = $res['titulo'];
$ponente = $res['ponente'];
$estado = $res['estado'];
$locked = $res['locked'];
$percent = $res['asistencia']/100;
$final = $res['final'];
$nbloque = $res['bloque'];
$sql = "SELECT titulo FROM bloques WHERE bloque=$nbloque";
$consulta = mysql_query($sql) or die("Error en la consulta $sql");
$res = mysql_fetch_array($consulta);
mysql_free_result($consulta);
$bloque = $res['titulo'];
$msg = "";
$iplocal = $_SERVER["SERVER_ADDR"];
$ipremote = $_SERVER["REMOTE_ADDR"];
if ($iplocal!=$ipremote && $locked)
    $msg = "Esperando al servidor";

//Cabecera HTML
cabecera();

//Código cuando se llega por un GET (inicio/final)
if (isset($_GET['op']))
{
    switch ($_GET['op'])
    {
        case '1': //Iniciar Ponencia
            if ($estado!=2)
            {
                $sql = "UPDATE sesiones SET estado=2,inicio_ponencia=NOW() WHERE sesion=$sesion";
                mysql_query($sql) or die("Error en la consulta $sql");
                $msg = "La ponencia '$titulo'<br>acaba de comenzar\n";
                $estado = 2;
            }
            break;

        case '2': //Terminar Ponencia
            if ($sesion==$_GET['ss'])
            {
                $sql = "UPDATE sesiones SET estado=3,final_sesion=NOW() WHERE sesion=$sesion";
                mysql_query($sql) or die("Error en la consulta $sql");
                $msg = "La ponencia '$titulo'<br>acaba de finalizar\n";
            }
    }
}

```

```

$sesion++;
if ($final==1)
{
    $sql = "UPDATE config SET sesion=$sesion,locked=1";
    mysql_query($sql) or die("Error en la consulta $sql");
    $msg .= "<br><br>Todos los asistentes deben abandonar la sala";
    $sql = "SELECT dni,rfid FROM asistentes WHERE dentrofuera10=1";
    $consulta = mysql_query($sql) or die ("Error en la consulta $sql");
    $res = mysql_fetch_array($consulta);
    while ($res)
    {
        $sql = "INSERT INTO control (dni,rfid,hora,sesion,bloque,entrasale10) values
        ('$res[dni]','$res[rfid]',NOW(),$_GET[ss],$nbloque,0)";
        mysql_query($sql) or die("Error en la consulta $sql");
        $res = mysql_fetch_array($consulta);
    }
    mysql_free_result($consulta);
    $sql = "UPDATE asistentes SET dentrofuera10=0";
    mysql_query($sql) or die ("Error en la consulta $sql");
}
elseif ($final==2)
{
    $sql = "UPDATE config SET locked=2";
    mysql_query($sql) or die("Error en la consulta $sql");
    $sql = "SELECT dni,rfid FROM asistentes WHERE dentrofuera10=1";
    $consulta = mysql_query($sql) or die ("Error en la consulta $sql");
    $res = mysql_fetch_array($consulta);
    while ($res)
    {
        $sql = "INSERT INTO control (dni,rfid,hora,sesion,bloque,entrasale10) values
        ('$res[dni]','$res[rfid]',NOW(),$_GET[ss],$nbloque,0)";
        mysql_query($sql) or die("Error en la consulta $sql");
        $res = mysql_fetch_array($consulta);
    }
    mysql_free_result($consulta);
    $sql = "UPDATE asistentes SET dentrofuera10=0";
    mysql_query($sql) or die ("Error en la consulta $sql");
}
else
{
    $sql = "UPDATE config SET sesion=$sesion";
    mysql_query($sql) or die("Error en la consulta $sql");
    $sql = "UPDATE sesiones SET estado=1,inicio_sesion=NOW() WHERE sesion=$sesion";
    mysql_query($sql) or die("Error en la consulta $sql");
}

//Saco las hora de inicio y final de la ponencia
$sql = "SELECT inicio_ponencia,final_sesion FROM sesiones WHERE sesion=$_GET[ss]";
$consulta = mysql_query($sql) or die ("Error en la consulta $sql");
$res = mysql_fetch_array($consulta);
$inicio = $res['inicio_ponencia'];
$stinicio = strtotime($inicio);
$final = $res['final_sesion'];
$tfinal = strtotime($final);
$duracion = $tfinal-$stinicio;
$sql = "UPDATE sesiones SET duracion=$duracion WHERE sesion=$_GET[ss]";
mysql_query($sql) or die ("Error en la consulta $sql");
mysql_free_result($consulta);

//Compruebo la asistencia para cada asistente
$sql = "SELECT dni,rfid FROM asistentes";
$consulta = mysql_query($sql) or die ("Error en la consulta $sql");
$res = mysql_fetch_array($consulta);
while ($res)
{
    $sql = "SELECT hora,sesion,entrasale10 FROM control WHERE rfid='$res[rfid]' AND
    bloque=$nbloque";
    $consulta2 = mysql_query($sql) or die("Error en la consulta $sql");
    $res2 = mysql_fetch_array($consulta2);
    $i = 0;
    $caso = 0;
    $primero = 0;
    $control[0]['entrasale10'] = -1; //Para evitar el notice
    // 0 -> No asiste
}

```

```

// 1 -> Ha entrado ANTES del comienzo de la ponencia
// 2 -> Ha entrado DESPUÉS del comienzo de la ponencia
//Se guardan entradas y salidas en el tiempo de la ponencia
while ($res2)
{
    $thora = strtotime($res2['hora']);
    if ($thora<$tinicio)
    {
        $control[0]['thora'] = $thora;
        $control[0]['entrasale10'] = $res2['entrasale10'];
    }
    elseif ($thora>$tinicio && $primero==0)
    {
        if ($control[0]['entrasale10'] == 1)
        {
            $caso==1;
            $i++;
        }
        else
            $caso = 2;
        $control[$i]['thora'] = $thora;
        $control[$i]['entrasale10'] = $res2['entrasale10'];
        $primero = 1;
    }
    else
    {
        $i++;
        $control[$i]['thora'] = $thora;
        $control[$i]['entrasale10'] = $res2['entrasale10'];
    }
    $res2 = mysql_fetch_array($consulta2);
}
mysql_free_result($consulta2);
//Se comprueba si se puede dar por válida la asistencia
if ($caso==0 && $control[0]['entrasale10']==1)
    $caso = 1;
if ($caso==0)
{
    //echo "$res[rfid] - No ha asistido<br>";
    $sql = "INSERT INTO asistencia (dni,rfid,sesion) values
('$res[dni]','$res[rfid]',$_GET[ss])";
    mysql_query($sql) or die ("Error en la consulta $sql");
}
else
{
    if ($caso==1)
        $control[0]['thora'] = $tinicio;
    if ($control[$i]['entrasale10']==1)
    {
        $i++;
        $control[$i]['thora'] = $tfinal;
    }
    $tasist = 0;
    for ($j = 0; $j<=$i; $j = $j+2)
        $tasist = $tasist + $control[$j+1]['thora'] - $control[$j]['thora'];
    if ($tasist >= $percent*$duracion)
        //echo "$res[rfid] - Ha asistido ($tasist / $duracion)<br>";
        $out = 1;
    else
        //echo "$res[rfid] - Ha asistido pero no lo suficiente ($tasist / $duracion)<br>";
        $out = 0;
    $sql = "INSERT INTO asistencia (rfid,sesion,asist,tiempo) values
('$res[rfid]',$_GET[ss],$out,$tasist)";
    mysql_query($sql) or die ("Error en la consulta $sql");
}
$res = mysql_fetch_array($consulta);
}
mysql_free_result($consulta);

//Se ajustan los parámetros para la siguiente sesión
$sql = "SELECT c.locked,c.sesion,s.titulo,s.pONENTE,s.estado,s.bLOQUE,s.FINAL FROM
config c INNER JOIN sesiones s ON c.sesion = s.sesion";
$consulta = mysql_query($sql) or die("Error en la consulta $sql");
$res = mysql_fetch_array($consulta);

```

```

mysql_free_result($consulta);
$sesion = $res['sesion'];
$titulo = $res['titulo'];
$ponente = $res['ponente'];
$estado = $res['estado'];
$locked = $res['locked'];
$final = $res['final'];
$nbloque = $res['bloque'];
$sql = "SELECT titulo FROM bloques WHERE bloque=$nbloque";
$consulta = mysql_query($sql) or die("Error en la consulta $sql");
$res = mysql_fetch_array($consulta);
mysql_free_result($consulta);
$bloque = $res['titulo'];
}
break;

case '3': //Abrir Bloque
if ($estado!=1)
{
    $sql = "UPDATE sesiones SET estado=1,inicio_sesion=NOW() WHERE sesion=$sesion";
    mysql_query($sql) or die("Error en la consulta $sql");
    $sql = "UPDATE config SET locked=0";
    mysql_query($sql) or die("Error en la consulta $sql");
    $msg = "La sesión '$titulo'<br>acaba de abrirse\n";
    $estado = 1;
    $locked = 0;
}
break;
}

//Código cuando se llega por un POST (introducción de un nuevo código de barras)
if (isset($_POST['rfid']) && $_POST['rfid']!="" && !$locked)
{
    $sql = "SELECT dni,apellidos,nombre,dentrofuer10 FROM asistentes WHERE rfid='$_POST[rfid]'";
    $consulta = mysql_query($sql) or die("Error en la consulta $sql");
    $res = mysql_fetch_array($consulta);
    mysql_free_result($consulta);
    if (!$res)
        $msg = "La acreditación no corresponde a ninguno de los asistentes\n";
    else
    {
        $sql = "SELECT hora FROM control WHERE rfid='$_POST[rfid]' ORDER BY hora DESC";
        $consulta = mysql_query($sql) or die("Error en la consulta $sql");
        $res2 = mysql_fetch_array($consulta);
        mysql_free_result($consulta);

        if (time()-strtotime($res2['hora'])<60)
        {
            $entradasalida = $res['dentrofuer10']==0?"salido de la sala":"entrado en la sala";
            $msg = "$res[nombre] $res[apellidos]<br>ha $entradasalida\n";
        }
        else
        {
            $entradasalida = $res['dentrofuer10']==1?"salido de la sala":"entrado en la sala";
            $msg = "$res[nombre] $res[apellidos]<br>ha $entradasalida\n";
            $nuevo=1-$res['dentrofuer10'];
            $sql = "UPDATE asistentes SET dentrofuer10=$nuevo WHERE rfid='$_POST[rfid]'";
            mysql_query($sql) or die("Error en la consulta $sql");
            $sql = "INSERT INTO control (dni,rfid,hora,sesion,bloque,entradasale10) values ('$res[dni]','$_POST[rfid]',NOW(),$sesion,$nbloque,$nuevo)";
            mysql_query($sql) or die("Error en la consulta $sql");
        }
    }
}

//Cuerpo HTML
echo "\t\t\t\t\t\t\t\t\t\t\t\t

IV InfoDIEZ - Control de Asistencia</p>\n";
echo "\t\t\t\t\t\t\t\t\t\t\t\t</td>\n";
echo "\t\t\t\t\t\t\t\t\t\t\t\t</tr>\n";
echo "\t\t\t\t\t\t\t\t\t\t\t\t<tr>\n";
echo "\t\t\t\t\t\t\t\t\t\t\t\t\t\t<td>&nbsp;</td>\n";
echo "\t\t\t\t\t\t\t\t\t\t\t\t\t\t</tr>\n";
echo "\t\t\t\t\t\t\t\t\t\t\t\t\t\t<tr>\n";


```


config.php

```
<?
$host = "localhost";
$user = "infodiez";
$pass = "infodiez2006";
$db = "infodiez2006";

$est[1]="pendiente de comenzar";
$est[2]="en curso";
?>
```

funciones.php

```
<?
function cabecera()
{
    echo "<html>\n";
    echo "<head>\n";
    echo "<t<title>IV InfoDIEZ - Control de Asistencia</title>\n";
    echo "<t<meta http-equiv=\"Content-Type\" content=\"text/html; charset=iso-8859-1\">\n";
    echo "<t<meta http-equiv=\"Cache-Control\" content=\"No-Cache\">\n";
    echo "<t<meta http-equiv=\"Pragma\" content=\"No-Cache\">\n";
    echo "<t<link href=\"estilos.css\" rel=\"stylesheet\" type=\"text/css\">\n";
    echo "</head>\n";
    echo "<body bgcolor=\"#FFFFFF\">\n";
    echo "<div align=\"center\">\n";
    echo "<t<table width=\"762\" border=\"0\" bordercolor=\"#FFFFFF\">\n";
    echo "<t<tr>\n";
    echo "<t<t<td width=\"752\"><div align=\"center\"><img src=\"infodiez.gif\" width=\"752\" height=\"94\"></div></td>\n";
    echo "<t</tr>\n";
    echo "<t</table>\n";
    echo "<t<br>\n";
    echo "<t<table width=\"762\" border=\"3\" bordercolor=\"0\">\n";
    echo "<t<tr>\n";
    echo "<t<t<td width=\"751\" bordercolor=\"#D8B12D\">\n";
    echo "<t<t<t<table width=\"100%\" border=\"0\" align=\"center\" cellpadding=\"0\" cellspacing=\"1\">\n";
    echo "<t<t<t<tr>\n";
    echo "<t<t<t<t<td width=\"23%\" align=\"center\" valign=\"top\">\n";
    echo "<t<t<t<t<p align=\"center\"><img src=\"jornadas.gif\" height=\"180\"></p>\n";
    echo "<t<t<t<t<t<table width=\"60\" border=\"0\" align=\"center\" cellspacing=\"0\" bgcolor=\"#C8CC0D\">\n";
    echo "<t<t<t<t<t<t<tr>\n";
    echo "<t<t<t<t<t<t<td><div align=\"center\"><b><font size=\"1\" color=\"#000000\" face=\"Verdana, Arial, Helvetica, sans-serif\">&nbsp;Organiza&nbsp;</font></b></div></td>\n";
    echo "<t<t<t<t<t<t<tr>\n";
    echo "<t<t<t<t<t<t<table>\n";
    echo "<t<t<t<t<t<t<t<p align=\"center\"><img src=\"diez.gif\" width=\"115\" height=\"53\" border=\"0\"></p>\n";
    echo "<t<t<t<t<t<t<t<table width=\"50\" border=\"0\" align=\"center\" cellspacing=\"0\" bgcolor=\"#C8CC0D\">\n";
    echo "<t<t<t<t<t<t<t<tr>\n";
    echo "<t<t<t<t<t<t<t<td><div align=\"center\"><b><font size=\"1\" color=\"#000000\" face=\"Verdana, Arial, Helvetica, sans-serif\">&nbsp;Apoya&nbsp;</font></b></div></td>\n";
    echo "<t<t<t<t<t<t<t<tr>\n";
    echo "<t<t<t<t<t<t<t<table>\n";
    echo "<t<t<t<t<t<t<t<t<p align=\"center\"><img src=\"dataria.gif\" width=\"100\" border=\"0\"></p>\n";
    echo "<t<t<t<t<t<t<t<t<td>\n";
    echo "<t<t<t<t<t<t<t<t<td width=\"54%\" align=\"center\" valign=\"top\">\n";
    echo "<t<t<t<t<t<t<t<t<table width=\"92%\" border=\"0\" align=\"center\">\n";
    echo "<t<t<t<t<t<t<t<t<t<tr>\n";
    echo "<t<t<t<t<t<t<t<t<t<t<td>&nbsp;</td>\n";
    echo "<t<t<t<t<t<t<t<t<t<t<tr>\n";
    echo "<t<t<t<t<t<t<t<t<t<t<t<tr>\n";
    echo "<t<t<t<t<t<t<t<t<t<t<t<td>\n";
}

function cierre()
{
    echo "<t<t<t<t<t<t<t<t<t<t<td>\n";
    echo "<t<t<t<t<t<t<t<t<t<t<t<tr>\n";
```

```

echo "\t\t\t\t\t\t\t\t</table>\n";
echo "\t\t\t\t\t\t\t</td>\n";
echo "\t\t\t\t\t\t\t  |
```

estilos.css

```

.info {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 10pt;
  font-weight: bold;
  color: #C8CC0D;
  text-decoration: none;
  text-align: center;
}

.negro {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 13pt;
  font-weight: bold;
  color: #000000;
  text-decoration: none;
  text-align: center;
}

.negro2 {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 10pt;
  font-weight: bold;
  color: #000000;
  text-decoration: none;
  text-align: center;
}

.azul {
  font-family: Verdana, Arial, Helvetica, sans-serif;
  font-size: 11pt;
  font-weight: bold;
  color: #0000FF;
  text-decoration: none;
  text-align: center;
}

```

Además de la aplicación principal, se han desarrollado 2 aplicaciones de apoyo. La primera de ellas se utiliza para cargar los identificadores *RFID* y asignarlos a los diferentes asistentes. La segunda aplicación, usada durante el proceso de testeo, reinicia la base de datos para poder realizar las pruebas desde cero.

```

load_rfid.php
<?
//Configura el acceso a la base de datos
require("config.php");
include("funciones.php");
$db_connection=mysql_connect($host,$user,$pass) or die("Error en la conexión con la Base de
Datos");
mysql_select_db($db);

if (isset($_POST['rfid']))
{
    $sql = "SELECT dni FROM asistentes WHERE rfid='$_POST[rfid]'";
    $consulta = mysql_query($sql) or die("Error en la consulta $sql");
    $res = mysql_fetch_array($consulta);
    mysql_free_result($consulta);

    if (!$res['dni'])
    {
        $sql = "UPDATE asistentes SET rfid='$_POST[rfid]' WHERE dni='$_POST[dni]'";
        mysql_query($sql) or die ("Error en la consulta $sql");
        $msg = "El identificador de $_POST[nombre] $_POST[apellidos] ha sido cargado con
éxito<br>\n";
    }
}

$sql = "SELECT dni,apellidos,nombre FROM asistentes WHERE rfid='' ORDER BY apellidos";
$consulta = mysql_query($sql) or die("Error en la consulta $sql");
$res = mysql_fetch_array($consulta);
mysql_free_result($consulta);

//Cabecera HTML
cabecera();

//Cuerpo HTML
echo "<t><t><t><t><t><t><t><p class=\\"negro\\">IV InfoDIEZ - Control de Asistencia</p>\n";
echo "<t><t><t><t><t><t></td>\n";
echo "<t><t><t><t><t></tr>\n";
echo "<t><t><t><t><t><t></tr>\n";
echo "<t><t><t><t><t><t><td>&nbsp;</td>\n";
echo "<t><t><t><t><t></tr>\n";
echo "<t><t><t><t><t><t></tr>\n";
echo "<t><t><t><t><t><t><td valign=\\"top\\">\n";

echo "<t><t><t><t><t><t><t><p class=\\"negro\\">Carga de identificadores</p>\n";

if (!$res['dni'])
    echo "<t><t><t><t><t><t><t><p class=\\"azul\\">Carga de identificadores concluida con
éxito</p>\n";
else
{
    if (isset($msg))
        echo "<t><t><t><t><t><t><t><p class=\\"azul\\">$msg</p>\n";

    echo "<t><t><t><t><t><t><t><form name=\\"load_rfid\\" method=\\"post\\"
action=\\"load_rfid.php\\">\n";
    echo "<t><t><t><t><t><t><t><p class=\\"azul\\">Carga la acreditación de $res[nombre]
$res[apellidos]<br><br>\n";
    echo "<t><t><t><t><t><t><t><input type=\\"text\\" name=\\"rfid\\" value=\\"\\"></p>\n";
    echo "<t><t><t><t><t><t><t><input type=\\"hidden\\" name=\\"nombre\\" value=\\"$res[nombre]\\"></p>\n";
    echo "<t><t><t><t><t><t><t><input type=\\"hidden\\" name=\\"apellidos\\"
value=\\"$res[apellidos]\\"></p>\n";
    echo "<t><t><t><t><t><t><t><input type=\\"hidden\\" name=\\"dni\\" value=\\"$res[dni]\\"></p>\n";
    echo "<t><t><t><t><t><t></form>\n";
    echo "<t><t><t><t><t><t><script language=\\"javascript\\"></p>\n";
    echo "<t><t><t><t><t><t><!--\n";
    echo "<t><t><t><t><t><t><tdocument.forms[0].rfid.focus()\n";
    echo "<t><t><t><t><t><t><!--\n";
}

```


II.3. Estructura de la base de datos

```

CREATE DATABASE `infodiez2006` DEFAULT CHARACTER SET latin1 COLLATE latin1_general_ci;
USE `infodiez2006`;

CREATE TABLE `asistencia` (
  `id` bigint(255) unsigned NOT NULL auto_increment,
  `dni` varchar(10) collate latin1_general_ci NOT NULL default '',
  `rfid` varchar(25) collate latin1_general_ci NOT NULL default '',
  `sesion` tinyint(2) NOT NULL default '0',
  `asist` tinyint(1) NOT NULL default '0',
  `tiempo` bigint(255) NOT NULL default '0',
  KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci ;

CREATE TABLE `asistentes` (
  `dni` varchar(10) collate latin1_general_ci NOT NULL default '',
  `rfid` varchar(25) collate latin1_general_ci NOT NULL default '',
  `apellidos` varchar(50) collate latin1_general_ci NOT NULL default '',
  `nombre` varchar(50) collate latin1_general_ci NOT NULL default '',
  `dentrofuera10` tinyint(2) NOT NULL default '0',
  `asistencias` tinyint(2) NOT NULL default '0',
  `apto` tinyint(2) NOT NULL default '0',
  KEY `dni` (`dni`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;

CREATE TABLE `bloques` (
  `bloque` tinyint(2) NOT NULL default '0',
  `titulo` varchar(50) collate latin1_general_ci NOT NULL default '',
  KEY `bloque` (`bloque`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;

CREATE TABLE `config` (
  `id` tinyint(1) NOT NULL default '0',
  `sesion` tinyint(2) NOT NULL default '0',
  `asistencia` tinyint(4) NOT NULL default '0',
  `locked` tinyint(2) NOT NULL default '0',
  `sesiones` tinyint(2) NOT NULL default '0',
  KEY `cod_sesion` (`sesion`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;

CREATE TABLE `control` (
  `id` bigint(255) unsigned NOT NULL auto_increment,
  `dni` varchar(10) collate latin1_general_ci NOT NULL default '',
  `rfid` varchar(25) collate latin1_general_ci NOT NULL default '',
  `hora` datetime NOT NULL default '0000-00-00 00:00:00',
  `sesion` tinyint(2) NOT NULL default '0',
  `bloque` tinyint(2) NOT NULL default '0',
  `entrasale10` tinyint(2) NOT NULL default '0',
  KEY `id` (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci ;

CREATE TABLE `sesiones` (
  `sesion` tinyint(2) NOT NULL auto_increment,
  `titulo` varchar(200) collate latin1_general_ci NOT NULL default '',
  `ponente` varchar(200) collate latin1_general_ci NOT NULL default '',
  `inicio_sesion` datetime NOT NULL default '0000-00-00 00:00:00',
  `inicio_ponencia` datetime NOT NULL default '0000-00-00 00:00:00',
  `final_sesion` datetime NOT NULL default '0000-00-00 00:00:00',
  `duracion` bigint(255) NOT NULL default '0',
  `estado` tinyint(2) NOT NULL default '0',
  `bloque` tinyint(2) NOT NULL default '0',
  `final` tinyint(1) NOT NULL default '0',
  KEY `cod_sesion` (`sesion`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci ;

```

II.4. Descarga de la aplicación

Se adjunta acceso a carpeta compartida con el siguiente contenido:

- source.7z – Contiene el código fuente de la aplicación.
- bbdd.7z – Contiene la base de datos utilizada (estructura, vacía y al finalizar las IV Jornadas InfoDIEZ).
- Servidor_InfoDIEZ_(Windows_XP)_[VirtualBox].ova – Máquina virtual, para *Virtual Box*, con el servidor configurado de la aplicación.
- RALUY_PIRLA_AGUSTÍN_430080_PFC.pdf – Esta memoria, en formato *PDF*.
- RALUY_PIRLA_AGUSTÍN_430080_PFC_PRESENTACIÓN.pdf – Presentación usada durante la defensa, en formato *PDF*.

Enlace:

<https://1drv.ms/f/s!AvxL3H5dkUh1hfIEiQ-N2Y2ifdLPJg>

Enlace corto:

<http://bit.do/controlPresencia>

