



UNIVERSITAT
POLITÀCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Sistema para el control de acceso a las instalaciones de un club de tenis

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Jordi Adam Domingo

Tutor: Andrés Boza García

2019/2020

Resumen

Con el presente trabajo final de grado, se pretende diseñar e implementar, un sistema de información capaz de gestionar un seguimiento real y preciso de los accesos a las instalaciones de un club de tenis. El proyecto propone una serie de mejoras orientadas a conseguir una explotación más eficiente de las instalaciones, obteniendo como resultado, un mayor rendimiento y control sobre ellas.

Para cumplir con dicho fin se ha desarrollado una aplicación móvil Android, así como la programación de un servidor web. La aplicación, complementada con un dispositivo que notifique el acceso, y junto con un servidor que aloje la lógica de negocio y los datos, consigue la funcionalidad propuesta.

Hay que destacar el diseño del sistema, orientado para adaptarse de forma genérica a cualquier club de tenis con su web de reservas ya existente, realizando previamente un estudio del caso específico a implantar.

Palabras clave: sistema de información, aplicación móvil, Android, servidor web,

Abstract

With the present final grade work, it is intended to design and implement an information system capable of managing a real and accurate monitoring of access to the facilities of a tennis club. The project proposes a series of improvements aimed at achieving a more efficient operation of the facilities, resulting in greater performance and control over them.

To fulfill this goal an Android mobile application has been developed as well as the programming of a web server. The application, complemented by a device that notifies access, and together with a server that hosts the business logic and data, achieve the proposed functionality.

It is necessary to emphasize the design of the system, oriented to adapt in a generic way to any tennis club with its existing reservation website, previously carrying out a study of the specific case to be implemented.

Keywords : information system, mobile application, Android, web server

Resum

Amb el present treball final de grau, es pretén dissenyar e implementar, un sistema d'informació capaç de gestionar un seguiment real y precís dels accesos a les instal.lacions d'un club de tenis. El projecte proposa una sèrie de millores orientades a conseguir una explotació més eficient de les instal.lacions, obtenint així, un major rendiment i control sobre elles.

Per a complir amb aquest propòsit s'ha desenvolupat una aplicació mòbil Android, així com la programació d'un servidor web. L'aplicació , complementada amb un dispositiu que notifique l'accés, i junt amb un servidor que allotje la lògica de negoci i les dades, aconsegueix la funcionalitat proposada.

Hi ha que destacar el disseny del sistema, orientat a adaptar-se de forma genèrica a qualsevol club de tenis amb la seva web de reserves ja existent, realitzant prèviament un estudi del cas específic a implantar.

Paraules clau: sistema d'informació, aplicació mòbil, Android, servidor web,

Agradecimientos

En primer lugar, dar las gracias a todos mis familiares cercanos, que siempre han confiado y han tenido claro cuales eran mis posibilidades. Con carácter especial a mis padres, que han vivido mi peor versión debido al estrés propio de llevar dos proyectos entre manos, y aún así, han sabido sobreponerme ante las adversidades que me han surgido tanto en el proyecto como en la vida.

Como no, a mi tutor Andrés Boza García, que ha sabido comprender la idea que tenía en mente sobre el proyecto y guiarme para poder llevarlo a cabo adecuadamente.

También mencionar mis amigos, que me han ayudado a desconectar en repetidas ocasiones de la rutina diaria entre casa y trabajo.

En definitiva, a todos los que han hecho posible que hoy esté escribiendo estas líneas.

Tabla de contenidos

1	<i>Introducción</i>	10
1.1	Presentación/Motivación	10
1.2	Objetivos	10
1.3	Estructura	11
2	<i>Estudio del Arte</i>	11
2.1	Situación actual de la tecnología de localización	11
2.2	Panorama actual de los sistemas operativos para Smartphone	12
2.2.1	Android.NetBeans.	13
2.2.2	Android.Android Studio.....	13
2.3	Servidor web	14
2.3.1	Apache.....	15
2.3.2	XAMPP	15
2.3.3	PHP.....	16
2.3.4	WordPress	17
2.3.5	MySQL.....	17
2.4	Interconexión entre sistemas	18
2.4.1	Servicios Web.	18
2.5	Localización en interiores	20
2.5.1	Beacons	21
3	<i>Propuesta</i>	22
3.1	Análisis del problema	22
3.1.1	Solución propuesta.	25
3.2	Diseño de la solución	27
3.2.1	Modelado del sistema.....	28
3.2.2	Arquitectura del sistema.....	35
3.2.3	Diseño de la aplicación móvil.	35
3.2.4	Diseño del servidor.....	37
3.2.5	Diseño de localización en interiores.....	38
3.2.6	Tecnología Utilizada.	38
3.3	Implementación	39
3.3.1	Implementación de la aplicación móvil.	39
3.3.2	Implementación del servidor web.	50

4	<i>Implantación y pruebas</i>	52
4.1	Caso de implantación	53
4.2	Pruebas	57
5	<i>Conclusiones</i>	61
6	<i>Líneas Futuras</i>	62
7	<i>Bibliografía</i>	63



Índice de figuras

Fig.1: Sistemas operativos móviles en España. Imagen tomada de: www.kantarworldpanel.com	11
Fig.2: Uso de servicios web en España. Imagen tomada de: www.trends.builtwith.com	14
Fig.3: Interfaz XAMPP en macOS	15
Fig.4: Logo Tecnología NFC.....	20
Fig.5: Logo Tecnología BLE.....	20
Fig.6: Estructura del dispositivo.....	20
Fig.7: Diagrama BPMN2 del proceso de reserva actual.....	21
Fig.8: Diagrama BPMN2 del proceso actual de acceso a las instalaciones.....	22
Fig.9: Diagrama BPMN2 del inicio del proceso global actual por parte del socio.....	23
Fig.10: Diagrama BPMN2 del inicio del proceso por parte del No socio.....	23
Fig.11: Diagrama BPMN2 del proceso mejorado de acceso a las instalaciones.....	25
Fig.12: Diagrama BPMN2 del inicio del proceso global mejorado por parte del socio.....	26
Fig.13: Diagrama BPMN2 de la participación del Dueño del restaurante en el proceso.....	26
Fig.14: Diagrama BPMN2 de la participación del Conserje en el proceso.	26
Fig.15: Actores involucrados.....	27
Fig.16: Casos de uso del sistema propuesto.....	27
Fig.17: Diagrama de clases UML de un club de tenis.....	35
Fig.18: Arquitectura del sistema planteado.....	35
Fig.19: Maqueta de la pantalla de acceso.....	36
Fig.20: Maqueta de la pantalla principal para empleados.....	36
Fig.21: Maqueta de la pantalla principal para socios.....	37
Fig.22: Eddystone Development Kit Kontakt.io.....	38
Fig.23: Vista delantera y trasera del dispositivo.	38
Fig.24: Pantalla real de acceso a la aplicación.....	40
Fig.25: Segunda actividad de la aplicación.....	44
Fig.26: Panel principal de notificaciones.....	48
Fig.27: Estructura global de la implantación	52

Fig.28: Respaldo de bases de datos origen.....	53
Fig.29: Transferencia de datos desde servidor origen.....	54
Fig.30: Estado inicial de la tabla” formulario reserva” en servidor club de tenis.....	57
Fig.31: Estado inicial de la tabla “Reserva” en servidor local.....	57
Fig.32: Reserva de instalación TENIS desde Web.....	58
Fig.33: Estado final de la tabla” formulario reserva” en servidor club de tenis tras la reserva web.	58
Fig.34: Estado final de la tabla “Reserva”en servidor local tras la reserva web.....	59
Fig.35: Estado de la tabla “Reserva” en nuestro servidor tras la detección de acceso de los socios.....	59
Fig.36: Notificación: contrincante a la espera.....	60
Fig.37: Notificación tras fichaje.....	60



1 Introducción

Conseguir una explotación adecuada de una instalación, bien sea deportiva o de cualquier otra índole, no es tarea fácil. Se deberá emplear un método de reserva que sea amable y cómodo de utilizar, así como otro capaz de controlar eficazmente el acceso a los usuarios.

Hace un tiempo empezó a introducirse la tecnología en este mundo, aportando entre otras, una forma de tener acceso a la ocupación de las instalaciones vía web.

Pese a ello, hay un aspecto que aún no está resuelto de forma eficiente y es la llegada o recepción del usuario a la instalación, esto es, el conocimiento exacto de que el usuario se encuentra o no ocupando una instalación dada.

Así pues, la finalidad del presente trabajo será la de adaptar un complemento a la tecnología ya existente con el objetivo de obtener un mejor control del acceso físico a las instalaciones.

1.1 Presentación/Motivación

Practico el deporte del tenis desde los 6 años y pertenezco a un club de tenis desde entonces. Me he vuelto inmerso en este mundo compitiendo entre clubes, realizando actividades relacionadas, entrenando con mis compañeros... En definitiva, pasando horas y horas allí. Todo ello, me ha despertado bastante interés en realizar un trabajo relacionado con ese ámbito, empleando conceptos y tecnologías vistas durante el grado.

1.2 Objetivos

El objetivo principal del proyecto es tener un mayor control sobre el acceso a las instalaciones de un club de tenis, traduciéndose de diferente forma en función de la instalación en cuestión. Para cumplir con este primer objetivo propuesto, se requerirá del diseño e implementación de un sistema de información compuesto por elementos hardware y software (aplicación móvil, servidor web, base de datos, dispositivos de geolocalización, etc..). Todo ello ayudará a automatizar procesos que actualmente se hacen de forma manual al acceder de forma física a las instalaciones.

Por una parte, dicha automatización permitirá al club de tenis tener un mayor grado de detalle sobre el nivel de ocupación de sus instalaciones y así ofrecer una visión general del club en cuanto a términos de sostenibilidad.

Por otra, aportarán numeroso valor añadido al socio, ejemplo de ello es: ofrecerles descuentos y promociones personalizados, proporcionarles estadísticas sobre el nivel de asistencia diaria en cada instalación, así como también, facilitar información sobre las citas próximas y datos de ocupación en tiempo real de la misma. Todo ello se traduce en fomentar de forma proactiva el uso de las instalaciones.

1.3 Estructura

El presente proyecto se divide en los siguientes bloques:

- En primer lugar se ofrece una información introductoria con el fin de ponerse en contexto sobre el tema que se va a abordar, también se definen los objetivos a perseguir.
- Posteriormente se exponen y analizan las posibilidades tecnológicas necesarias para llevar a cabo el proyecto. Este será pues el fundamento del proyecto.
- El tercer bloque y más importante, describe a grandes rasgos cuál es la problemática actual, se procede con su diseño, desarrollo y posterior implantación.
- Finalmente se da un resumen de los resultados obtenidos, así como de las futuras ampliaciones que se pueden llevar a cabo en un futuro.

2 Estudio del Arte

2.1 Situación actual de la tecnología de localización

Hoy en día el hecho de estar localizados nos aporta numerosos beneficios, principalmente el hecho de poder llegar a cualquier sitio sin necesidad de tener que conocer previamente el camino a seguir. Ahora bien, el rango de actuación de los satélites encargados de aportarnos dicha funcionalidad tiene su límite a “nivel de calle”, y de esta forma para el caso en que nos encontremos dentro de un espacio cerrado no podremos saber en qué estancia o parte de él

estamos. Con todo ello nace la idea de “Baliza electrónica”, un dispositivo que utiliza la tecnología Bluetooth Low Energy o BLE y se encarga de emitir señales que podrán captar otros dispositivos que tengan en común dicha tecnología. Así pues, pasamos a comentar algunos ámbitos donde se esta empezando a aplicar en la actualidad:

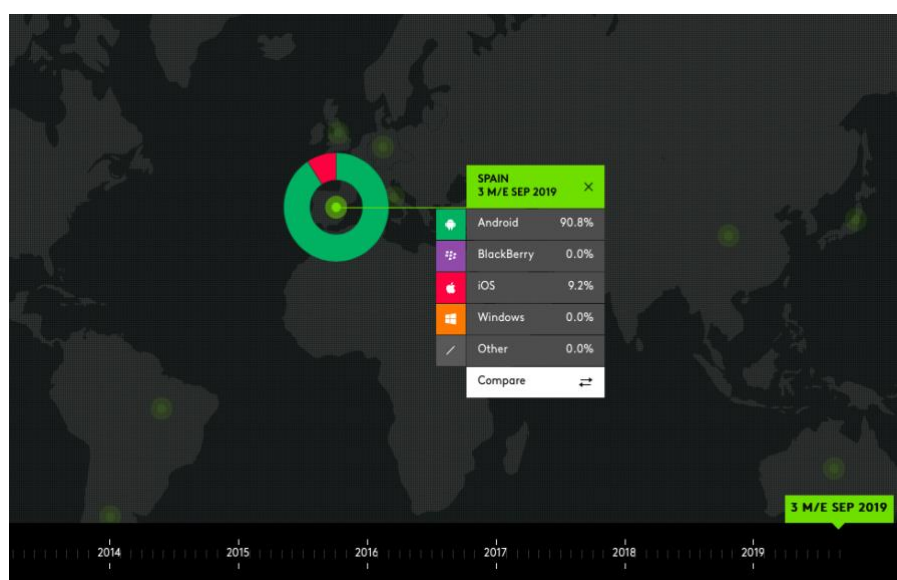
Para el ámbito de las tiendas, los dispositivos se encuentran repartidos por la tienda tanto dentro como en las inmediaciones y conforme el usuario va siendo detectado por las diferentes zonas se le envían notificaciones personalizadas a la aplicación asociada. De esta manera, el posible cliente puede enterarse de posibles ofertas o artículos en promoción que desconocía.

La inmersión de esta tecnología en el ámbito de los museos también ha sido bastante interesante. Siguiendo el mismo principio que para el caso anterior, se le proporcionará información concerniente al cuadro que se vaya encontrando delante suyo.

2.2 Panorama actual de los sistemas operativos para Smartphone

La industria de los dispositivos y las aplicaciones móviles es un entorno en constante cambio. Los sistemas operativos que han venido dando renombre al contexto de la programación han sido entre otros: BlackBerry, IOS, Windows Phone y Android, pero de entre todos ellos, hay uno que ha conseguido ponerse a la cabeza de todos ellos.

Según (Kantar,2019), el panorama actual en cuanto a sistemas operativos usados en España durante este año en curso es el siguiente:



*Fig.2: Sistemas operativos móviles en España.
Imagen tomada de: www.kantarworldpanel.com*

2.2.1 Android.NetBeans.

Según (AcademiaAndroid,2014), NetBeans es un IDE que nos permite de forma rápida y fácil desarrollar aplicaciones Java de escritorio, móviles y aplicaciones web, utilizando tecnologías y lenguajes como HTML5 con HTML, Javascript y CSS. Proporciona un conjunto de herramientas para PHP, C y C++.

Entre sus puntos fuertes al igual que Eclipse es que ambos son de código abierto y gratuitos y cuenta como añadido con una gran base de usuarios que forman parte de una comunidad de desarrolladores muy amplia en todo el mundo. Algunas de sus funcionalidades más importantes según (AcademiaAndroid,2014) son:

- Cuenta con asistentes para la configuración de distintos proyectos, así como selección de frameworks.
- Incluye intuitivo editor de código con sugerencias, coloreado y acceso a clases con tan solo un click de versiones.
- Presenta un depurador muy útil que nos permitirá entre otras monitorizar en tiempo real los valores de las propiedades y variables.
- Desde el propio IDE nos permitirá conectarnos a distintos gestores de base de datos y consultar tablas y datos. Además, se puede integrar fácilmente con diversos servidores de aplicaciones.
- Contiene una gran base de plugins con el fin de aumentar las funcionalidades que nos ofrece el propio IDE.

A la hora de utilizarlo para el desarrollo de aplicaciones para Android necesitaríamos simplemente de un JDK y agregarle algunos plugins.

2.2.2 Android.Android Studio.

Android Studio es el entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android y se basa en el software IntelliJ IDEA de JetBrains, ha sido publicado de forma gratuita a través de la Licencia Apache 2.0 y su disponibilidad para las distintas plataformas es alta, ofreciendo soporte a Microsoft Windows, macOS, y GNU/LINUX.

Según (Hébuterne, 2014), Android Studio integra varias herramientas que cubren las necesidades esenciales del desarrollador:

- Presenta un motor de producción (Build system, en inglés) Gradle, capaz de producir varias versiones de una misma aplicación.
- Una herramienta para construir las interfaces de usuario visualmente.



- Un compendio de plantillas de aplicación que integran las principales estructuras de código utilizadas en el desarrollo de Android.


A destacar también algunas de sus principales características de su versión estable actual según (Wikipedia-Android Studio, 2017):

- Integración de un potente optimizador de código como es ProGuard, así como funciones de firma de aplicaciones.
- Mayor nivel de especificación a la hora de programar.
- Incluye una consola de desarrollador, encargada de dar consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones y otros problemas.
- Soporte para programar aplicaciones para Android Wear.
- Soporte integrado para Google Cloud Platform, que permite la integración con Google Cloud Messaging y App Engine.
- Por último, cuenta también con un dispositivo virtual de Android utilizado para ejecutar y probar aplicaciones

2.3 Servidor web.

Hablamos de un elemento crucial en nuestro sistema, será el encargado de atender las peticiones enviadas por la aplicación a través de los dispositivos móviles y por tanto donde se alojará gran parte de la lógica del negocio.

Según (Wikipedia-ServidorWeb, 2017), Un servidor web o servidor HTTP es un programa informático que procesa una aplicación del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en cualquier lenguaje o aplicación del lado del cliente.

Top In Web Server Usage Distribution in Spain		
Technology	Websites	%
 Apache	185,033 *	54.74
 nginx	110,939 *	32.82
 IIS	28,132 *	8.32
 Zeus	5,818	1.72
 Phusion Passenger	1,138	0.34
 Citrix NetScaler	1,135	0.34
 Varnish	923 *	0.27

*Fig.2: Uso de servicios web en España.
Imagen tomada de: www.trends.builtwith.com*

2.3.1 Apache

Según afirma (Wikipedia-Servidor_http_apache,2019), el servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual según la normativa RFC 2616. Es utilizado principalmente para enviar páginas web estáticas y dinámicas en la World Wide Web, de hecho, muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

2.3.2 XAMPP

XAMPP es, según (Apachefriends, 2018), una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl, su paquete de instalación ha sido diseñado para ser increíblemente fácil de instalar y usar. Su objetivo es crear una distribución fácil de instalar para desarrolladores que se están iniciando en el mundo de Apache. Así pues, XAMPP viene configurado por defecto con todas las opciones activadas y es gratuito tanto para usos comerciales como no comerciales.

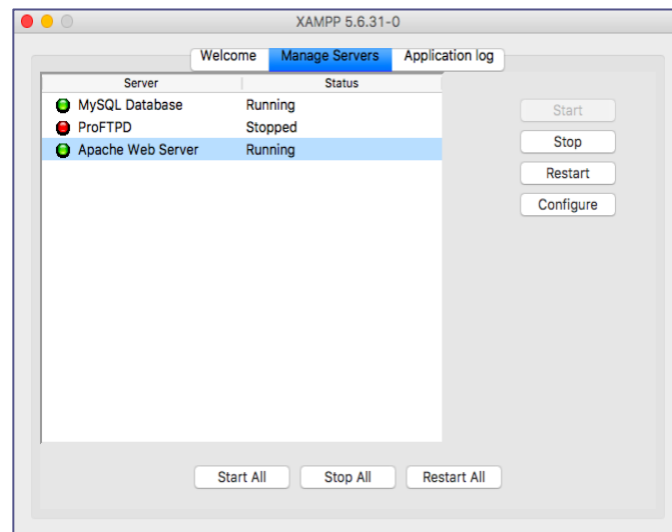


Fig.3: Interfaz XAMPP en macOS

2.3.3 PHP

Según (Php,2019), PHP (acrónimo recursivo de *PHP: Hypertext Preprocessor*) es un lenguaje de código abierto muy popular especialmente adecuado para el desarrollo web y que puede ser incrustado en HTML.

Coincidiendo con las líneas que se exponen en (Desarrolloweb,2019), PHP nos brinda la oportunidad de desarrollar sitios y aplicaciones de todo tipo. Es el motor además de los CMS más populares, como WordPress, Joomla, Drupal o Magento, entre muchos otros. En PHP también encontramos frameworks potentes y muy usados como Laravel o Symfony.

Si la comparamos con ASP, su ventaja más notoria es su carácter multiplataforma. Por otro lado, los programas en ASP resultan más lentos y pesados, así como menos estables. Hablando de su analogía con Perl, utilizado habitualmente en la programación de CGI, puede decirse que PHP fue diseñado para desarrollo de scripts orientados a web, mientras que Perl para muchos más usos cosa que lo hace más complicado.

Pasamos ahora a ver algunas de sus características más reseñables:

- Puede emplearse en todos los sistemas operativos principales, incluyendo Linux, Unix.
- Admite la mayoría de los servidores web de hoy en día, tales como Apache, IIS y muchos otros.
- Posibilidad de utilizar programación por procedimientos, programación orientada a objetos (POO), o una mezcla de ambas.

- Ofrece un amplio soporte para bases de datos.
- Cuenta con soporte para comunicarse con otros servicios usando protocolos tales como LDAP, IMAP, SNMP, NNTP, POP3, HTTP, COM (en Windows) y muchos otros.

2.3.4 WordPress

Tomando como referencia las palabras mencionadas en (Wordpress,2019), WordPress es un software diseñado para todos, enfatizando en la accesibilidad, rendimiento, seguridad y facilidad de uso. Ofrece la oportunidad a cualquiera de crear y compartir, desde anécdotas personales a movimientos que cambian el mundo.

Wordpress trabaja con temas preestablecidos, donde podemos elegir entre infinidad de ellos; el tema define un diseño para la posible web, y dentro de él se van creando diferentes páginas; estas últimas serán como plantillas que definirán la estructura estética de la página en cuestión. Pues bien, estas plantillas, de igual forma que el tema, no son más que código escrito en el lenguaje HTML y PHP, que interpreta y almacena el propio servidor.

Algunas de sus funcionalidades a nombrar según (Wikipedia-WordPress,2019) son:

- Fácil instalación, actualización y personalización.
- Capacidad de crear páginas estáticas (a partir de la versión 1.5).
- Permite ordenar artículos y páginas estáticas en categorías, subcategorías y etiquetas.
- Permite comentarios y herramientas de comunicación entre blogs (Trackback, Pingback, etc).
- Admite el uso de *permalinks* (enlaces permanentes y fáciles de recordar) mediante `mod_rewrite`.
- Distribución de los artículos mediante RDF, RSS 0.92, RSS 2.0 y Atom 1.0.
- Subida y gestión de datos adjuntos y archivos multimedia.
- Admite plantillas y widgets.
- Extensión de funcionalidades añadidas gracias a la instalación/aplicación de complementos y plugins.

2.3.5 MySQL

MySQL es un sistema gestor de bases de datos (SGBD, DBMS por sus siglas en inglés) muy conocido y ampliamente usado por su simplicidad y notable rendimiento. Aunque carece de algunas características avanzadas disponibles en otros SGBD del mercado, es una opción atractiva



tanto para aplicaciones comerciales, como de entretenimiento precisamente por su facilidad de uso y tiempo reducido de puesta en marcha. Esto y su libre distribución en Internet bajo licencia GPL le otorgan como beneficios adicionales (no menos importantes) contar con un alto grado de estabilidad y un rápido desarrollo.

MySQL dispone de dos programas principales: el servidor o motor y el cliente o motor MySQL. El primero de ellos se encarga de estar a la espera de posibles peticiones recibidas de los clientes; el segundo es el encargado de actuar.

2.4 Interconexión entre sistemas.

Para nuestro caso, será imprescindible el intercambio de datos entre dispositivos (Aplicación móvil haciendo uso del hardware y servidor). Así pues, requeriremos de un servicio o arquitectura que gestione esta comunicación logrando el paso de información.

2.4.1 Servicios Web.

Según (Wikipedia-Servicio Web, 2017), Un **servicio web** (en inglés, *web service* o *web services*) es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como internet.

Entorno a este concepto han surgido varios enfoques según (Androidcurso, 2017):

Llamadas a procedimiento remotos (RPC): Se enfoca el servicio web como una colección de operaciones o procedimientos que pueden ser invocados desde una máquina diferente de donde se ejecutan. Dado que RPC es una extensión directa del paradigma de llamadas a funciones, resulta sencillo de entender para un programador. Al ser una de las primeras alternativas que se implementó, se conocen como servicios web de primera generación.

Arquitectura orientada a servicios (SOAP): En el planteamiento anterior, RPC, la unidad básica de interacción es la operación. En este nuevo planteamiento, la unidad de interacción pasa a ser el mensaje. Cada uno de los mensajes que vamos a utilizar ha de ser definido siguiendo una estricta sintaxis expresada en XML. En la actualidad se trata de la arquitectura más extendida, soportada por la mayoría del *software* de servicios web.

Transferencia de estado representacional (REST): En los últimos años se está popularizando este nuevo planteamiento, que se caracteriza principalmente por su simplicidad. En REST se utiliza directamente el protocolo HTTP, por medio de sus operaciones GET, POST, PUT y DELETE. En consecuencia, esta arquitectura se centra en la solicitud de recursos, en lugar de las operaciones o los mensajes de las alternativas anteriores.

2.4.1.1 Rest

Los servicios web basados en la arquitectura REST (Representational State Transfer), son una alternativa más simple a SOAP y a WSDL, la cual transmite los datos sobre el protocolo estandarizado HTTP. Varias empresas como Google, Facebook y Yahoo! son casos de éxito al migrar sus servicios a esta tecnología. REST fue publicado en el año 2000 por Roy Fielding en la Universidad de California en una conferencia en la cual presentaba acerca de principios arquitectónicos de software para usar a la web como una plataforma de procesamiento distribuido (Seta, 2008). REST no está estrictamente relacionado con el protocolo HTTP, pero es con el que más comúnmente se asocia.

Repasando algunos de los principios incluidos en (Paredes y Rocha, 2019, p.311) :

- **Utiliza métodos de HTTP de manera explícita:** Cada método tiene una función específica: GET su uso es para obtener un recurso desde el servidor, POST se usa para crear un nuevo recurso en el servidor, PUT actualiza o cambia el estado de un recurso, DELETE elimina un recurso.
- **No mantiene estado:** Es necesario tener un escalamiento para cubrir la demanda constante y en crecimiento mediante balanceador, clúster y servidores con alta disponibilidad con el objetivo de distribuir las peticiones entre los equipos para disminuir el tiempo de respuesta.
- **URI en formato de directorios:** La estructura de los URI se la crea de una manera similar a directorios que es fácil de entender lo cual permite poder utilizarlos de manera intuitiva.
- **Representaciones:** Los servicios web REST emiten una respuesta en formato JSON o XML los cuales pueden ser usados por cualquier aplicación desarrollada en cualquier lenguaje.

2.4.1.2 Introducción a Volley.

Según (Developer.android,2019), Volley es una biblioteca HTTP que facilita y agiliza el uso de redes en aplicaciones para Android.

Entre los diferentes puntos fuertes que presenta, podemos destacamos los siguientes:

- Programación automática de solicitudes de red.



- Varias conexiones de red simultáneas.
- Almacenamiento de respuestas en caché y en disco transparentes con coherencia de caché en HTTP estándar.
- Compatibilidad con la priorización de solicitudes.
- API de cancelación de solicitudes (permite cancelar una única solicitud, o bien establecer bloques o grupos de solicitudes para cancelar)..
- Personalización sencilla, por ejemplo, de reintentos o retiradas.
- Ordenamiento sólido que permite completar correctamente la IU con datos recuperados de forma asíncrona de la red.
- Cuenta con herramientas de depuración y rastreo.

Se distingue por sus operaciones de tipo RPC que se usan para completar la IU, por ejemplo, obtener una página de resultados de la búsqueda como datos estructurados. Se integra fácilmente con cualquier protocolo y, además, incluye compatibilidad con strings sin procesar y JSON. Al proporcionar compatibilidad integrada con las funciones que se necesita, Volley elimina la necesidad de escribir código estándar y te permite concentrarte en la lógica que es específica de tu aplicación. Sin embargo, no es adecuado para operaciones de transmisión o descarga grandes, ya que almacena todas las respuestas en la memoria durante el análisis. Así pues, para operaciones de descarga grandes, que no es nuestro caso, podríamos utilizar como alternativa DownloadManager.

2.5 Localización en interiores.

Tomando como base el artículo introducido por (Raya,2013), vislumbramos dos claras opciones: La Tecnología NFC (Near Field Communications) o bien la tecnología BLE (Bluetooth Low Energy). Paso ahora a citar algunas de sus diferencias principales a fin de que podamos valorar qué opción se adecua más a nuestro caso:

En primera instancia hablamos del alcance que tienen estos dispositivos: La tecnología NFC presenta un alcance muy bajo (como unos 15 o 20 centímetros como máximo), mientras que la Bluetooth LE le recorta a más del doble llegando a los 50 metros.

En cuanto al aspecto energético, la primera tecnología requerirá que al menos una de las dos partes (emisor o receptor) tenga fuente propia. Para la segunda, el emisor requerirá de una simple pila de botón cuya durabilidad es bastante amplia. El receptor, al ser un dispositivo electrónico dispondrá de fuente de alimentación propia.

Si hablamos de compatibilidad, actualmente existen una mayor cantidad de dispositivos que soportan Bluetooth BLE antes que NFC.



Fig.4: Logo Tecnología NFC



Fig.5: Logo Tecnología BLE

2.5.1 Beacons

El dispositivo está compuesto por un chip radio bluetooth LE y una batería para alimentarlo. Este chip emitirá ondas que posteriormente serán captadas por dispositivos que posean la tecnología común bluetooth 4.0 o superior.



Fig.6: Estructura del dispositivo

3 Propuesta

3.1 Análisis del problema

Para conocer mejor el contexto donde se deben cumplir los objetivos propuestos en el presente TFG, se ha realizado un análisis de los procesos que se llevan a cabo actualmente para la reserva y el acceso a las instalaciones del club de tenis. Se busca pasar de un engorroso proceso de reserva-acceso a la instalación a otro más cómodo e intuitivo. Paso a continuación a detallar el proceso actual haciendo uso del modelo del mismo implementado mediante BPMN2 (Business Process Modeling Language 2):

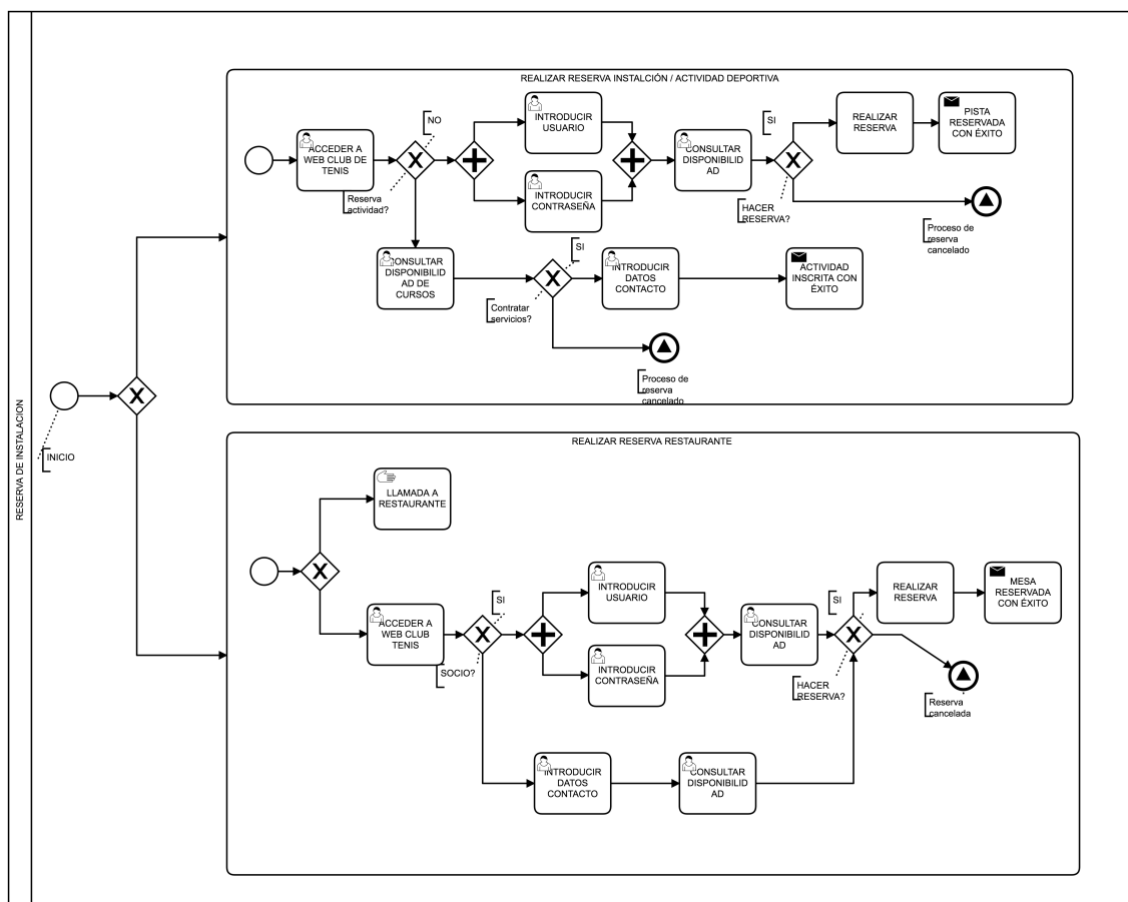


Fig.7: Diagrama BPMN2 del proceso de reserva actual

1.-El proceso empieza por parte de la persona interesada en usar la instalación (bien sea soci@ o no). Esta característica definirá el grado de limitación de esta.

1.1-Si se trata de un socio, este podrá hacer tanto reservas de instalaciones deportivas como actividades deportivas y reservas al restaurante. Para las dos primeras tendrá que hacer uso de la web , mientras que para la última podrá optar por una doble opción reservar vía telefónica o web. A la hora de reservar vía web , tendrá que previamente hacer un login con su usuario y contraseña.

1.2- Si nos centramos en el caso de los no socios, estos no podrán realizar una reserva de una instalación deportiva. Tendrán que jugar con un socio, siendo este último quien realice la reserva. Ahora bien, si que podrán reservar tanto actividades deportivas llevadas a cabo en el gimnasio como realizar reservas en el restaurante, esto es debido a que la gestión de estos dos establecimientos es gestionada por gente ajena al club de tenis que paga de forma mensual por el local en cuestión.

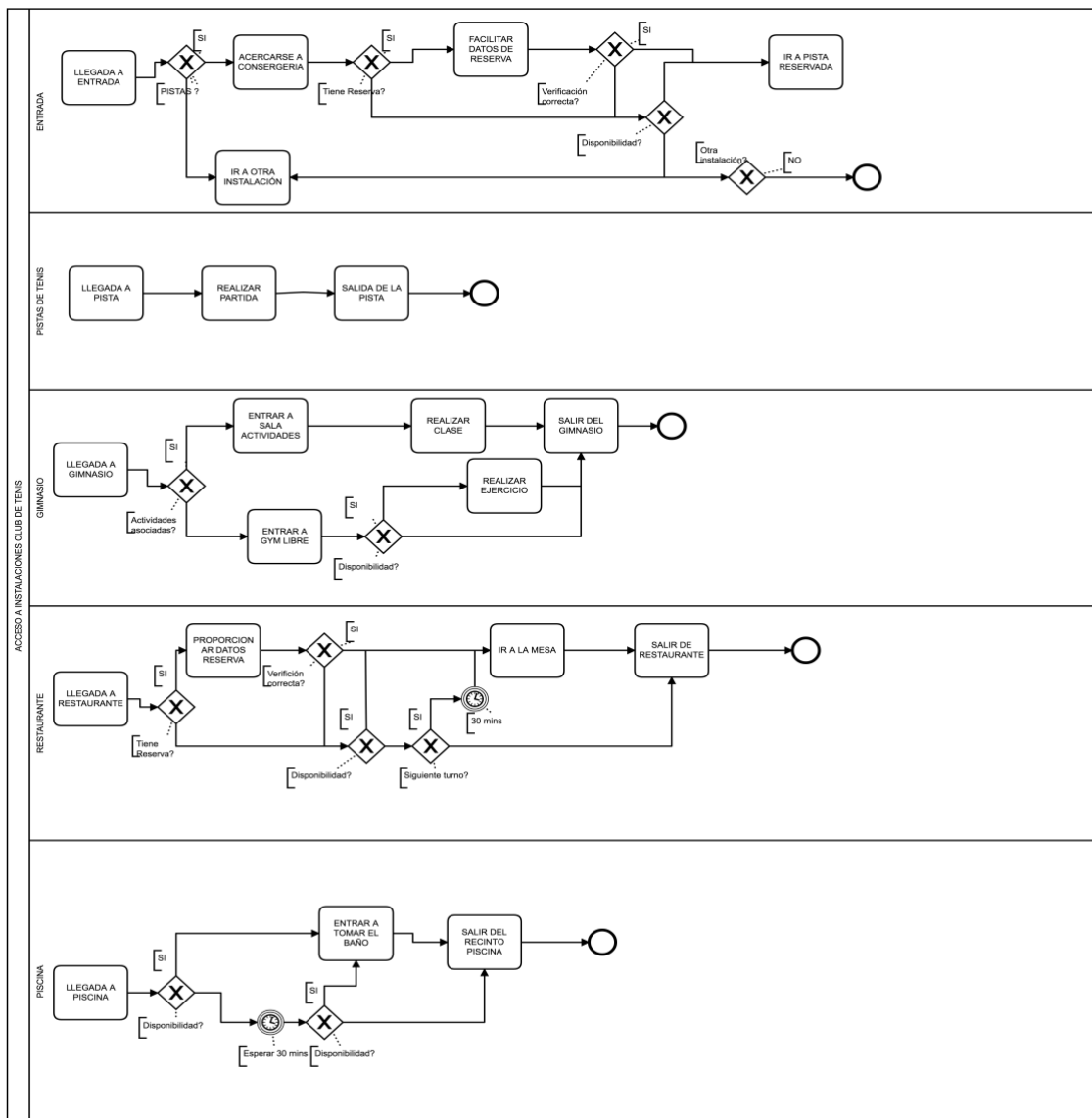


Fig.8: Diagrama BPMN2 del proceso actual de acceso a las instalaciones



2.-Vamos a comentar la segunda parte del proceso modelado anteriormente, este representa el acceso a la instalación previamente reservada. Una vez se ha llegado al club de tenis se podrá dar una doble casuística:

2.1-Si se desea ocupar una instalación deportiva, habrá que pasar en primer lugar por conserjería con el fin de validar la reserva, y en caso de no tenerla, también se podrá realizar in situ si existe disponibilidad. Por último, se accederá a la instalación en cuestión.

2.2-En caso de querer ocupar otras instalaciones como el gimnasio o piscina, no se requerirá el paso obligatorio por conserjería. Ahora bien, si se opta por ir, nadie podrá asegurar su disponibilidad.

Finalmente se expone el inicio o punto de partida de los procesos en ambas situaciones. A destacar que, para el caso del usuario no socio, sólo estará presente en la reserva. La mejora que se va a presentar irá orientada a los socios del club que pagan asiduamente por su mantenimiento.

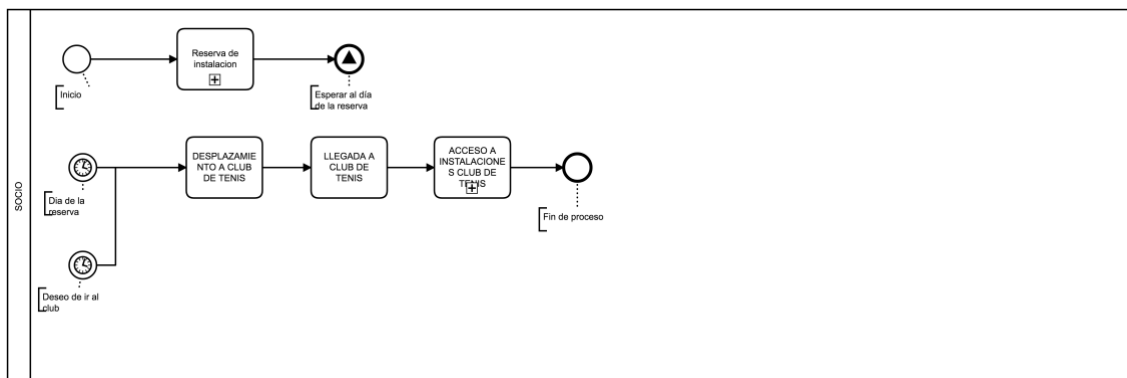


Fig.9: Diagrama BPMN2 del inicio del proceso global actual por parte del socio

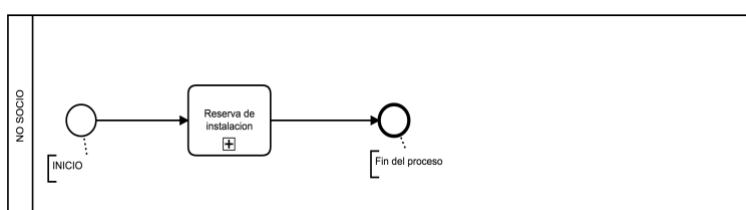


Fig.10: Diagrama BPMN2 del inicio del proceso por parte del No socio

Con el proceso actual se han conseguido mejorar muchos aspectos, aún así, el sistema empleado para gestionar instalaciones en centros deportivos (como clubs de tenis, polideportivos...) presenta bastante margen de mejora. Paso a citar algunos ejemplos:

En el ámbito de las pistas de tenis, se realiza una reserva a través de una web y posteriormente se comprueba dicha reserva antes de jugar. Dicha comprobación previa se realiza de la mano de

empleados del club y requiere de un riguroso control por parte de estos, especialmente en los casos en que haya gente jugando sin tener reserva o gente que no sea del club de tenis...etc

En cuanto al gimnasio o piscina es bastante difícil llevar un control en tiempo real de la ocupación de estos. De igual forma que para el caso anterior, requeriremos de un seguimiento por parte de los empleados del club. Si acudimos y no tenemos sitio, tendremos que esperar más tiempo o volver a casa habiendo hecho un viaje en balde.

Por último, las reservas del restaurante se pueden realizar a través de una llamada telefónica o bien mediante la web. Si bien gracias a esto el dueño del restaurante puede hacerse un cálculo de la comida que deberá comprar, le será aún de mejor utilidad saber si el cliente se encuentra llegando a las proximidades de la instalación. De esta forma podrá ultimar los preparativos del cliente/s en concreto y agilizar el proceso de servido.

3.1.1 Solución propuesta.

Una vez analizada la problemática, se vislumbra que la única solución posible consiste en hacer uso de alguna tecnología capaz de detectar el paso del usuario por la instalación. Así pues, siguiendo con las líneas descritas en el último punto del apartado dos del documento, la tecnología escogida para implementar la solución tiene que ver con los dispositivos BLE Beacons.

La función única de los Beacons es la de emitir unas señales Bluetooth que contienen varios códigos que los identifican de forma unívoca, aún así, necesitaremos de algunos elementos tecnológicos para poder lograr la funcionalidad final de estos dispositivos y conformar así el sistema:

- Primeramente, una aplicación encargada de captar y gestionar las señales emitidas por los dispositivos, así como de realizar un posterior tratamiento de la información. Esto se traducirá en peticiones a un servidor donde se alojará la mayor parte de la lógica de negocio.
- Un servidor que atienda las peticiones que trasladará el móvil a través de la aplicación. Estas peticiones HTTP contendrán en su mayoría, consultas de cualquier índole a la base de datos alojada allí.

Con lo dispuesto anteriormente, pasamos a presentar el nuevo modelo BPMN 2.0 que mejorará en gran medida el proceso de acceso a la instalación tratado:

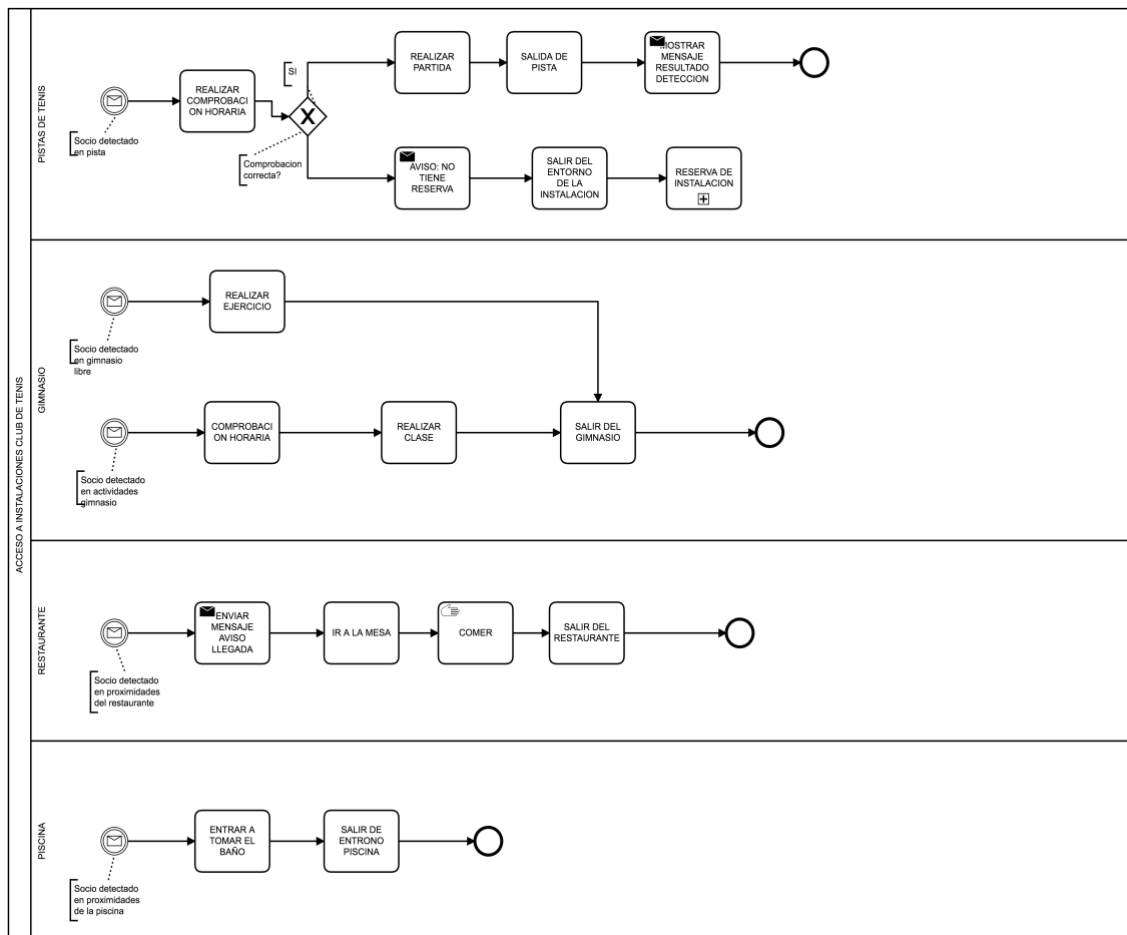


Fig.11: Diagrama BPMN2 del proceso mejorado de acceso a las instalaciones

A nivel general, podemos observar dos diferencias bastante notorias entre el diagrama anteriormente expuesto y su versión antigua:

- En primer lugar, se ha suprimido el antiguo proceso de validación, de esta forma el usuario irá directo a la instalación a ocupar y no deberá nada más.
- Por otra parte, se habrá mejorado el proceso para el caso de instalaciones no reservables (piscina, gimnasio libre...). Tal es la mejora que, gracias a la detección de los usuarios tras su paso, nos permitirá hacer un análisis de ocupación en tiempo real y ofrecerla al socio. Así, cualquiera que tenga pensado ir podrá hacer una previsión.

Algunas de las funcionalidades que aporta el tratamiento de la información con el acceso a la instalación, afectarán indirectamente al proceso de inicio del proceso por parte del socio. También provocará la actualización del modelado del dueño del restaurante y conserje/s encargado/s del mantenimiento. Pasamos a detallarlo gráficamente:

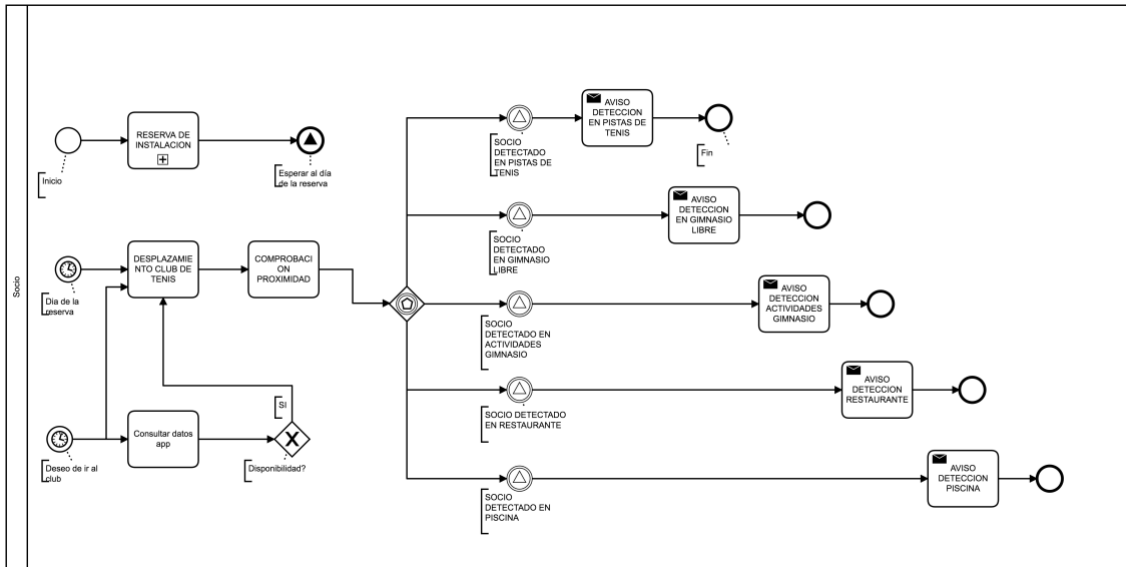


Fig.12: Diagrama BPMN2 del inicio del proceso global mejorado por parte del socio

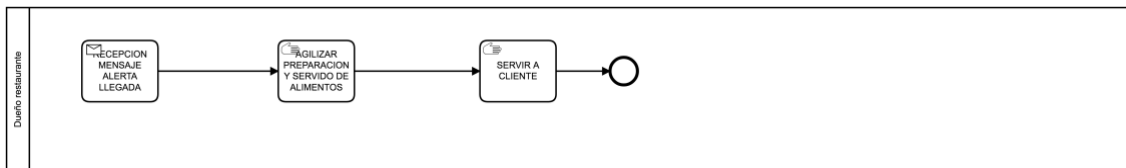


Fig.13: Diagrama BPMN2 de la participación del Dueño del restaurante en el proceso.

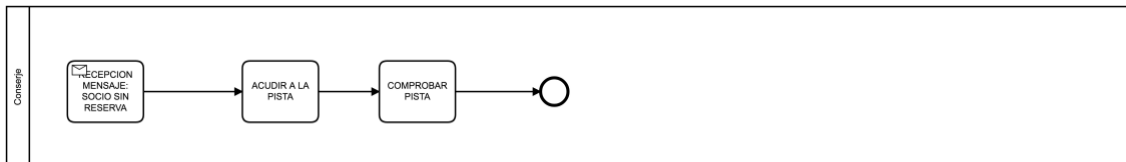


Fig.14: Diagrama BPMN2 de la participación del Conserje en el proceso.

En línea con los argumentos descritos anteriormente sobre la funcionalidad mejorada en otros modelados, hay que destacar que, gracias a esta nueva forma de acceso se procederá con mayor agilidad a la hora de usar la instalación, obteniéndose como consecuencia, una mayor explotación.

3.2 Diseño de la solución.

Vamos a exponer ahora cómo se va a llevar a cabo la propuesta planteada, modelando para ello el sistema a tratar, siguiendo con la presentación de su arquitectura, así como el diseño de sus principales componentes.



3.2.1 Modelado del sistema

3.2.1.1 Casos de uso.

Haciendo uso de la notación introducida por (Booch,2014), presentamos la comunicación existente de nuestro sistema entre los diferentes usuarios involucrados en él (actores). En primer lugar, pasamos a exponer los diferentes actores:



Fig.15: Actores involucrados

Posteriormente, definimos las comunicaciones entre ellos:

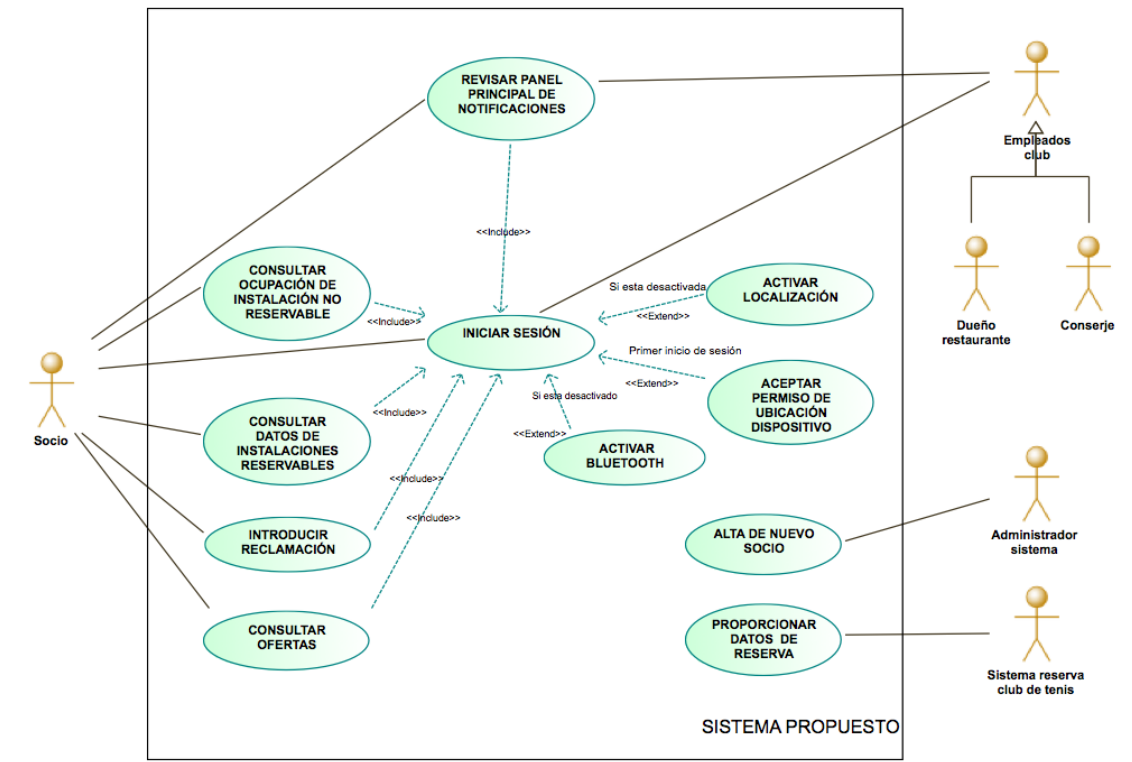


Fig.16: Casos de uso del sistema

3.2.1.2 Clases implementadas

Pasamos a ver las clases asociadas al proyecto:

1.- Clase Club de Tenis. Representará cada uno de los clubs de tenis insertados en nuestro sistema.

Entre sus atributos encontraremos:

Atributo	Descripción	Tipo
Id_ClubTenis	Ayudará a identificar de forma unívoca a un club de tenis	Numérico
Nombre		Texto
NIE		Texto
Num_Instalaciones	Número de instalaciones con las que cuenta el club de tenis	Numérico
Num_socios	Número de socios que posee el club de tenis	Numérico
Dirección	Ubicación física del club	Texto
Contacto	Dirección de correo electrónico.	Texto
Fecha_creación	Fecha de fundación del club.	Fecha
Logo		Texto

2.- Clase Tipo_Instalacion. Representará los diferentes tipos de instalación disponibles en el club.

Entre sus atributos encontraremos:

Atributo	Descripción	Tipo
Id_TipoInst	Abreviatura que representará el tipo de instalación	Numérico
Nombre		Texto
Dimensiones	Medidas de la instalación.	Texto

3.-Clase Instalacion. Representará las instalaciones del club de tenis. Sus atributos serán:

Atributo	Descripción	Tipo
Id_ClubTenis	Será necesario para identificar la instalación	Numérico
Id_TipoInst	Abreviatura asociada al tipo de instalación al que pertenecerá	Texto



Sistema para el control de acceso a las instalaciones de un club de tenis

Id_Instalacion	Valor numérico asociado	Numérico
Nombre		Texto
Superficie		Texto

4.-Clase **Socio**. Será la encargada de representar los socios que tiene un club. Sus atributos serán los siguientes:

Atributo	Descripción	Tipo
Id_ClubTenis	Identificador del club de tenis al que pertenecerá	Numérico
Id_Socio	Se tratará de un identificador único asociado	Numérico
Nombre	Valor numérico asociado	Texto
Apellidos		Texto
DNI		Texto
Fecha_nacimiento		Fecha
Fecha_inscripcion	Fecha de alta en el club	Fecha
Login	Usuario con el que accederán a la aplicación	Texto
Contraseña	Contraseña con el que accederán a la aplicación	Texto
Foto		Texto

5.- Clase **Empleado**. Representará empleados que tiene el club a su disposición. Sus atributos remarcables serán:

Atributo	Descripción	Tipo
Id_ClubTenis	Club de tenis al que representará	Numérico
Id_Empleado	Identificador único asociado	Numérico
Nombre		Texto
Apellidos		Texto
DNI		Texto
Fecha_alta		Fecha
Login		Texto
Contraseña		Texto

6.-Clase **Reserva**. Se tratará probablemente de la tabla más utilizada del sistema. Será la encargada de registrar las posibles reservas que se realizarán desde la web, así como también, almacenará

información asociada a las detecciones que se darán con el acceso a las instalaciones. Así pues, entre sus atributos podremos destacar:

Atributo	Descripción	Tipo
Id_ClubTennis	Identifica el club de tenis desde el que se realizará la reserva	Texto
Id_TipoInst	Tipo de instalación que será reservada	Texto
Id_Instalacion	Valor numérico de la instalación que será reservada	Numérico
Id_Socio	Identificador del socio que habrá hecho la reserva	Numérico
Id_Reserva	Valor numérico que, junto a otros campos, identificará la reserva	Numérico
Fecha_ReservaE	Definirá la fecha y hora respectivos a la entrada y salida a instalación	Fecha-Hora
Fecha_ReservaS	Definirá la fecha y hora correspondiente a la salida de la instalación. Será utilizado en todas las instalaciones reservables excluyendo el restaurante	Fecha-Hora
Fecha_Compr	Representará la fecha de reserva de la instalación. Será utilizado con las reservas y accesos a pistas de tenis, así como accesos a actividades de sala	Fecha
Dia_Reserva	Definirá el día de la reserva. Será usado en la reserva de actividades de gimnasio	Texto
Entrada y Salida	Hablaremos de dos campos encargados de registrar la fecha y hora real tanto de la entrada y salida del socio respectivamente a las pistas de tenis. Así pues, solamente serán usados en el acceso.	Booleanos
Resultado_E y Resultado_S	Se tratará de una cadena que representará cómo hemos realizado el fichaje en la entrada y salida respectivamente de la pista de tenis. Esto irá en función de la diferencia entre la hora real de llegada o salida y la definida en la reserva.	Textos
Resultado_Asistencia	Con el registro de la salida de la pista, se llevará a cabo un baremo entre Resultado_E y Resultado_S, cuyo resultado derivará en este campo. Se tratará pues, de una cadena que representará la resolución en forma de estado del acceso.	Texto

Mensaje_Resultado	En función del campo anterior, se le generará al usuario una notificación donde se le indicará si tiene compensación por una buena asistencia o no.	Texto
Num_Clases	Mostrará el número de clases totales	Numérico
Num_asistencia	Representará el número actual de asistencias a la actividad y serán usadas con el acceso a actividades programadas	Numérico
Mensaje_Aviso	Representará avisos a compañeros de partida para el caso del tenis, o bien, avisos de llegada de comensales para el caso de restaurantes	Texto
EnApp	Puesto que manejaremos mensajes que a su vez serán solicitados desde el panel principal de notificaciones cada cierto tiempo, será conveniente identificar en la tabla, qué avisos correspondientes a qué reservas se le habrán ya mostrado al usuario. De esta forma nos evitaremos traer toda la información cada vez.	Bit (0,1)

7.- Clase Uso. hablaremos de una clase orientada a gestionar el acceso a las instalaciones no reservables. Así pues, todos los atributos que presentaremos estarán vinculados con el acceso a las instalaciones:

Atributo	Descripción	Tipo
Id_ClubTenis, Id_TipoInst, Id_Instalacion, Id_Socio	Irán en la misma línea que lo expuesto en las clases anteriores, pero para el caso en que alguien ha hecho uso de una instalación la cual no cuenta con reserva previa.	Texto,texto, numérico, numérico
Id_Uso	Identificador numérico que representará un uso	Numérico
Entrada y salida	Se tratará de una fecha y hora de entrada y salida respectivas a la instalación en cuestión.	Fecha-Hora

8.-Clases TipoAviso_Empleado , Tipo_Oferta y Tipo_Reclamacion. Respresentarán los diferentes tipos de avisos al empleado, concretamente, ofertas y reclamaciones. Constarán de los siguientes atributos:

Atributo	Descripción	Tipo
Id_TipoAvEmp, Id_TipoOf y Id_TipoRec	Abreviaturas que mostrarán los diferentes tipos de objetos asociados.	Texto, Texto, Texto,
Nombre	Identificador numérico que representará un uso	Numérico

9.-Clase Aviso_Empleado. tratará de representar avisos a los empleados del club de tenis. Sus atributos son:

Atributo	Descripción	Tipo
Id_ClubTenis	Indicará el club de tenis al que pertenecerá el aviso	Texto
Id_Empleado	Mostrará el empleado al que irá dirigido el aviso	Numérico
Id_TipoAvEmp	Abreviación que representará el aviso asociado	Texto
Id_AvEmp	Valor numérico que identificará el aviso en cuestión	Numérico
Mensaje		Texto
Fecha_Aviso		Fecha-Hora
EnApp		Bit(0,1)

9.-Clase Oferta. su misión será la de representar una gratificación al socio. Sus atributos son:

Atributo	Descripción	Tipo
Id_ClubTenis	Club de tenis donde se ha generado la oferta.	Texto
Id_Socio	Identificador del socio al que se le ha aplicado la oferta.	Numérico
Id_TipoOf y Id_Of	De igual forma que para el caso de Aviso_empleado.	Texto, numérico
Descripcion		Texto
Descuento	Texto que indicará el porcentaje a rebajar	Texto

10.-Clase Reclamación. Representará una reclamación realizada por el socio.

Atributo	Descripción	Tipo
----------	-------------	------



3.2.2 Arquitectura del sistema

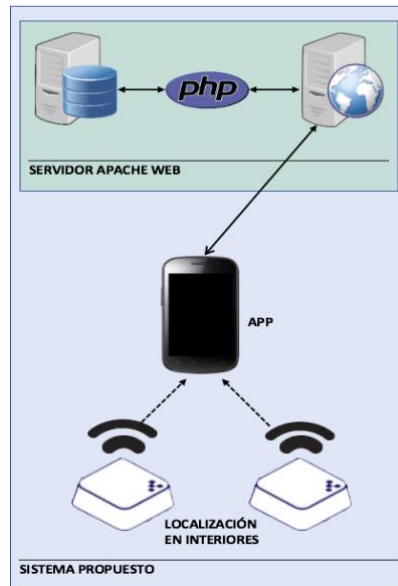


Fig.18: Arquitectura del sistema planteado

La figura anterior representará el entorno correspondiente al sistema planteado, donde inicialmente la aplicación estará abierta a la recepción de cualquier señal emitida por algún dispositivo Beacon, y tras esta, se comunicará con el servidor web con la finalidad de dejar constancia de dicho evento en la Base de datos. Otra funcionalidad consistirá en realizar peticiones sobre datos ya almacenados, obteniendo su correspondiente recepción asociada.

Tal y como veremos posteriormente, este entorno se complementará junto a otro dedicado a realizar reservas mediante una página web, seguidamente serán guardadas en el servidor local y redirigidas hacia nuestro servidor donde también serán almacenadas.

3.2.3 Diseño de la aplicación móvil.

Dado que estaremos ante una aplicación orientada a gestionar unas instalaciones privadas, a las cuales sólo accederán un número determinado de personas (socios), lo primero que deberán encontrarse estos será una pantalla de acceso dedicada a validar las credenciales del mismo

Sistema para el control de acceso a las instalaciones de un club de tenis

Tras ella encontraremos la pantalla principal, la cual variará en función de quién la vaya a utilizar, y es que no sólo irá orientada al socio, sino que también intervendrán los empleados del club. Estos últimos recibirán información sobre los accesos a las instalaciones en forma de

notificaciones, por lo que su pantalla principal contendrá únicamente el panel de inicio de notificaciones.



Fig.19: Maqueta de la pantalla de acceso



Fig.20: Maqueta de la pantalla principal para empleados

Para el caso de los socios, la pantalla principal será donde se alojará la mayor parte de la funcionalidad de la aplicación, estará conformada por diversas pestañas por las que el usuario podrá ir moviéndose con ayuda de una barra lateral. La finalidad de estas pestañas, fruto de los accesos a las instalaciones por parte de los socios, será entre otras: Dar información asociada a las instalaciones tanto de forma gráfica como de datos simples, así como mostrar notificaciones de relevancia para el socio. También contendrá una pantalla que le permitirá añadir de forma simple y rápida alguna reclamación o sugerencia que le pueda surgir.

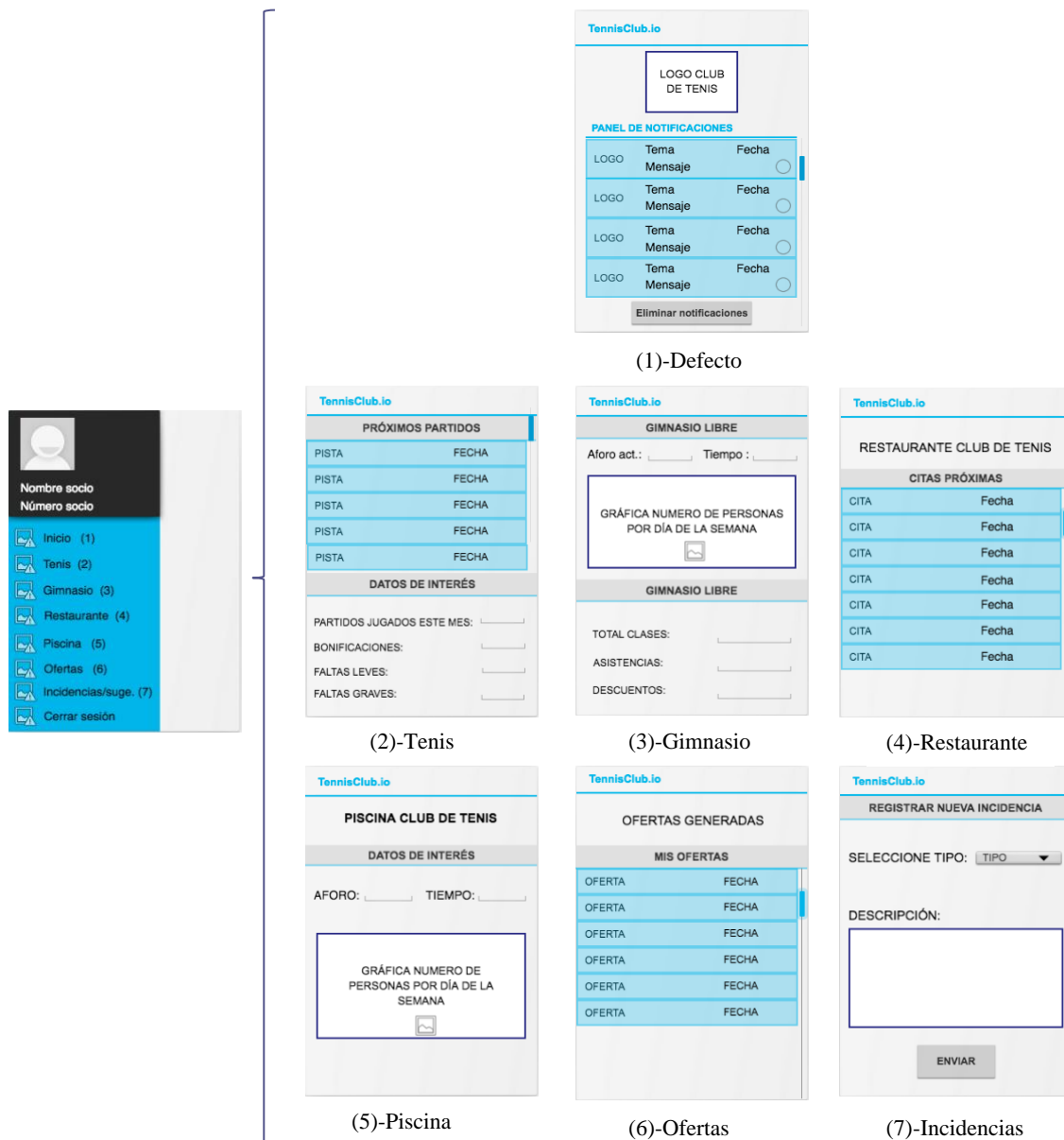


Fig.21: Maqueta de la pantalla principal para socios

3.2.4 Diseño del servidor.

En primer lugar, aquí podremos encontrar la lógica de negocio en su mayor parte, esto es, la funcionalidad asociada a la aplicación. En él, localizaremos dos archivos principales con extensión PHP: el primero llamado “index.php”, será el encargado de recibir las peticiones por parte del exterior y hacer uso de la funcionalidad propuesta, el segundo archivo llamado “funciones.php”, contendrá la funcionalidad nombrada anteriormente. Finalmente encontraremos



una base de datos, la cual conformará el núcleo de la aplicación y cuya misión será la de preservar y servir los datos contenidos en ella. Será pues totalmente necesaria para cumplir con la finalidad de las funciones.

3.2.5 Diseño de localización en interiores.

En este proyecto se utilizarán los Beacons del fabricante Kontakt.io, configurados con el protocolo Eddystone. Estos, como se ha descrito con anterioridad, serán los encargados de localizar a los usuarios tras su paso por las instalaciones.



Fig.22: Eddystone Development Kit Kontakt.io



Fig.23: Vista delantera y trasera del dispositivo.

3.2.6 Tecnología Utilizada.

A continuación, se procede con la justificación de la tecnología que se va a utilizar para el desarrollo del presente trabajo.

3.2.6.1 Aplicación Móvil.Entornos de desarrollo

A la hora de desarrollar una aplicación podemos optar por una de las siguientes opciones: O bien nos decantamos por usar lenguajes y herramientas nativos de una plataforma en concreto, o por el contrario apostar por una visión más ambiciosa usando alguna herramienta multiplataforma.

Así pues, teniendo en cuenta la complejidad que supondría la segunda opción y a la vista de los resultados obtenidos en la gráfica anteriormente expuesta, nos decantaremos por la primera.

Aunque existen algunas más, he hecho alusión a dos: la primera de ellas (Netbeans), fue la utilizada durante el transcurso del grado y tengo bastante dominio sobre ella. Para este caso, estuve revisando la importación de librerías necesarias para el proyecto y encontré bastantes dificultades a la hora de importarlas. Entre las diversas opciones posibles, encontré Android

Studio como una de las más utilizadas y recomendadas para aplicaciones Android, también a sido elegida por su alto nivel de compatibilidad.

3.2.6.2 Entorno Servidor

La plataforma XAMPP será idónea para nuestro caso, ya que nos ayudará a recrear fácilmente la implantación de un servidor en un ordenador local, empleando para ello, prácticamente la misma tecnología utilizada en el entorno real.

3.2.6.3 Peticiones Web.

En cuanto a este ámbito, habremos optado por un servicio web Rest. Con él, no es necesario realizar una programación exclusiva para cada plataforma, ni limitada por el lenguaje, sino que es suficiente con exponer el servicio web para ser consumido por los clientes.

3.2.6.4 Detección en proximidades.

Tras el análisis llevado a cabo en el punto 2 de la memoria, la opción que más se adecua a las necesidades requeridas, es la relacionada con la tecnología Bluetooth LE. Ambas son útiles, pero en cuanto a términos de funcionalidad, la segunda opción supera a la primera. Y es que para el caso que nos ocupa, es mucho más cómodo no tener que acercarse en exceso al terminal para registrar el acceso a la instalación.

3.3 Implementación

3.3.1 Implementación de la aplicación móvil.

La aplicación desarrollada esta formada por dos actividades o activities. Las librerías empleadas para su realización han sido las siguientes:

- En primer lugar, se ha optado por la librería “Android-beacon-library” para gestionar las comunicaciones entre los dispositivos BLE o Beacons. Aunque existen numerosas opciones, esta puede ser fácilmente configurada y es capaz de detectar a la gran mayoría de beacons.
- Para realizar las peticiones desde la aplicación al servidor hemos incluido la librería Volley, esta nos permitirá llevar a cabo peticiones http.
- La librería Picasso se encargará de cargar las imágenes mediante una ruta al fichero destino. Tanto la ruta como las propias imágenes estarán almacenadas en el servidor, la primera dentro de una columna de las tablas de la base de datos y la segunda en una carpeta de este.



- Por último, también hemos utilizado MPAndroidChart a la hora de representar gráficas, cuya función será la de mostrar de una forma más intuitiva los datos al usuario.

Cabe también destacar que hemos hecho uso de los conceptos expuestos por (González,2017) a la hora de implementar el servicio de los dispositivos de localización.

La **primera** que nos encontraremos será la de identificación del usuario, cuya funcionalidad consistirá en realizar una petición http al servidor con los datos introducidos por el usuario usando para ello la librería Volley. Es importante destacar que para que pueda funcionar Volley, necesitaremos incluir el permiso de los servicios de internet dentro el fichero AndroidManifest.xml del proyecto.

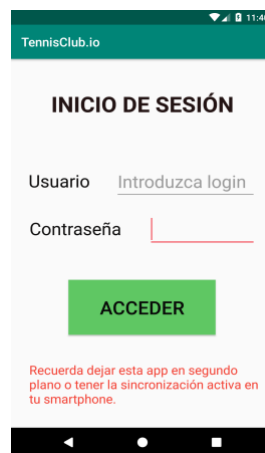


Fig.24: Pantalla real de acceso a la aplicación

La funcionalidad de esta actividad esta organizada en dos métodos:

- El primero es el “OnCreate()”, el cual es invocado nada más se crea una instancia de la actividad. Aquí recuperamos los datos introducidos por el cliente (usuario y contraseña) a la par que gestionamos el botón “Acceder”. Al hacer uso de ese botón, primeramente, se realizarán unas comprobaciones previas con el fin de asegurarse de que el usuario no haya dejado datos sin rellenar, y tras estas, se procederá con la llamada al método que tramita la operación.


```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    setContentView(R.layout.activity_login);
    Usuario = findViewById(R.id.editText2);
    Contraseña = findViewById(R.id.editText);
    Acceder = findViewById(R.id.button);
    Acceder.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(final View v) {

            //Comprobaciones previas
            if (Usuario.getText().length() == 0 || Contraseña.getText().length() == 0) {

                Toast.makeText(getApplicationContext(), "Usuario o contraseña
vacios", Toast.LENGTH_LONG).show();

            } else {

                progreso = new ProgressDialog(LoginActivity.this);
                progreso.setMessage("Un momento..."); // Setting Message
                progreso.setTitle("Iniciando sesion"); // Setting Title
                //Progress Dialog Style Spinner
                progreso.setProgressStyle(ProgressDialog.STYLE_SPINNER);
                progreso.setIndeterminate(false);
                new Thread(new Runnable() {
                    public void run() {
                        try {

                            RealizaPeticionServidor(v);

                            Thread.sleep(3000);
                            progreso.dismiss();

                        } catch (Exception e) {
                            Toast.makeText(getApplicationContext(), "Excepcion DIALOG",
                            Toast.LENGTH_LONG ).show();
                            e.printStackTrace();
                        }
                    }
                }).start();
            }
        }
    });
}

```

Método encargado de crear e inicializar la actividad principal



- Con el método “RealizaPeticiónServidor(View v)” primeramente se encapsulan los datos introducidos por el usuario en un objeto JSON, usando para ello una estructura HashMap previamente creada. Seguidamente, se gestiona un método de petición al servidor indicándole los siguientes argumentos: el tipo de petición (para este caso usaremos el POST), la dirección IP del servidor donde realizamos las peticiones, el objeto JSON donde se encuentran los datos adjuntos a la petición y finalmente un Objeto Listener encargado de tratar posibles respuestas que pueda haber por parte del servidor asociadas a dicha petición.

```
public int RealizaPeticiónServidor(final View v) {  
  
    // Cola de peticiones de la librería Volley  
    String tag_servidor = "LOGG";  
    final String login = Usuario.getText().toString();  
    String contra = Contraseña.getText().toString();  
  
    // Enviamos junto con la petición los datos introducidos  
    // por el usuario  
  
    Map<String, String> params = new HashMap<String,  
    String>();  
    //Diferenciamos la interfaz a mostrar en función del  
    //usuario.  
    if (login.equals("Conserje") || login.equals("Restaurante") ||  
    login.equals("Admin")){  
        params.put("tag","LOGINAPP_EMPLEADOS");  
    } else {  
        params.put("tag","LOGINAPP_SOCIOS");  
    }  
    params.put("login",login );  
    params.put("contra",contra);  
  
    JSONObject jsonObj = new JSONObject(params);  
  
    JsonRequest jsonObjReq = new JsonRequest(  
    Request.Method.POST, ConnectionConfig.URL_LOGIN, jsonObj, new  
    Response.Listener<JSONObject>() {  
  
        //Gestionamos la respuesta del servidor.  
        @Override  
        public void onResponse(JSONObject response) {  
            try {  
  
                switch(response.getString("Resultado")) {
```

```

case "LOGIN_OK":

    //En funcion del usuario se mostrará una app parcial y por tanto no se
    //necesitarán de los mismos parametros.
    if (login.equals("Conserje") || login.equals("Restaurante")||
    login.equals("Admin")){

        Intent intent = new Intent(v.getContext(), Main1Activity.class);
        intent.putExtra("numempleado", response.getString
        ("NumEmpleado"));
        intent.putExtra("codigoclub",response.getString("CodigoClub"));
        intent.putExtra("logo", response.getString("LogoClub"));
        intent.putExtra("login", login);
        startActivity(intent);
        finish();

    }else {

        Intent intent = new Intent(v.getContext(), Main2Activity.class);
        intent.putExtra("numsocio", response.getString("Numsocio"));
        intent.putExtra("nombresocio", response.getString("Nombre"));
        intent.putExtra("apellidos", response.getString("Apellidos"));
        intent.putExtra("foto", response.getString("Foto"));
        intent.putExtra("codigoclub", response.getString("CodigoClub"));
        intent.putExtra("logo", response.getString("LogoClub"));
        intent.putExtra("login", login);
        startActivity(intent);
        finish();
    }
    break;
case "LOGIN_INCORRECTO":
    Toast.makeText(getApplicationContext(), "Login incorrecto",
    Toast.LENGTH_LONG).show();
    break;
case "CONTRASENA_INCORRECTA":
    Toast.makeText(getApplicationContext(), "Contraseña incorrecta",
    Toast.LENGTH_LONG).show();
    break;
    }
    } catch (JSONException e) {
        e.printStackTrace();
    }
    }
    }, new Response.ErrorListener() {
        @Override
        //Captura algun error que pueda surgir usando Volley.
        }
    });
    // Añadimos la petición a la cola de peticiones de Volley
    ApplicationController.getInstance().addToRequestQueue(jsonObjReq, tag_servidor);
    return 0;
}

```



Como veremos posteriormente, todo esto a nivel del servidor, se traducirá a una comprobación en la base de datos sobre la existencia tanto del usuario como de la contraseña del empleado o socio que desea acceder.

Tras una validación correcta de usuario, se pasará a la **segunda** actividad cuya funcionalidad se pasa a describir a continuación.

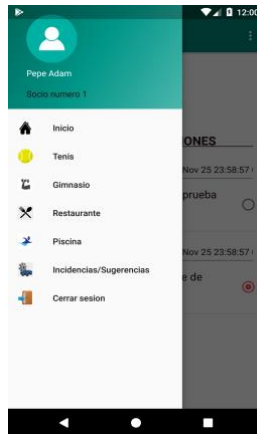


Fig.25: Segunda actividad de la aplicación

En primer lugar, se deberá de gestionar la detección de los dispositivos de localización durante un periodo definido. Tras este, se llevarán a cabo las pertinentes peticiones http teniendo en cuenta la casuística (En qué instalación se ha detectado al usuario). Para lograr este fin tendremos que realizar las siguientes modificaciones:

- Dentro del método “OnCreate()” inicializamos el gestor encargado de detectar a los dispositivos y configuramos el protocolo a usar (que para nuestro caso será Eddystone) así como una región donde serán detectados.

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    //Se inicializan los distintos elementos de la actividad.
    setContentView(R.layout.activity_main2);
    Toolbar toolbar = findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);
    DrawerLayout drawer = findViewById(R.id.drawer_layout);
    NavigationView navigationView = findViewById(R.id.nav_view);

    //Vinculamos con esta clase los elementos fisicos xml que contendrán los datos.
    ...
}
```

```

//Inicializamos el fragmento de inicio por defecto.
Fragment fragment = new InicioFragment();
getSupportFragmentManager().beginTransaction().add(R.id.content_main2,
fragment).commit();

//Gestor de beacons: configuracion previa (le indicaremos el tipo de protocolo
de beacon..)
//1-Inicializamos mBeaconManager
mBeaconManager = BeaconManager.getInstanceForApplication(this);

//2-Fijamos el protocolo beacon a usar: para este caso será EDDYSTONE
mBeaconManager.getBeaconParsers().add(new BeaconParser().
setBeaconLayout(BeaconParser.EDDYSTONE_UID_LAYOUT));
//mBeaconManager.bind(this);

//3-Creamos un array de n identificadores.Lo inicializaremos vacio puesto que
necesitamos detectar todos los beacons que estén alrededor nuestro.
ArrayList<Identifier> identifiers = new ArrayList<>();
identifiers.add(null);
//Crear region donde buscar beacons (para este caso los buscaremos todos)
miRegion = new Region(ALL_BEACONS_REGION, identifiers);

//4-Empieza la deteccion
PermisosLocalizacion();
}

```

Se inicia la instancia del servicio de detección

- Seguidamente, después de llevar a cabo las comprobaciones pertinentes sobre los requisitos previo al uso de los dispositivos, se procede con la búsqueda de estos en la región previamente creada. De dicha tarea se encargará el método “OnBeaconServiceConnect()”.Volviendo a los requisitos previos al uso de esta tecnología, se necesitará tener activada la localización o gps, encendido el Bluetooth (este debe ser 4.0 o superior), así como habilitar los permisos de localización.

```

@Override
public void onBeaconServiceConnect() {
    try {
        //Empieza a buscar beacons en la region definida anteriormente
        mBeaconManager.startRangingBeaconsInRegion(miRegion);
    } catch (RemoteException e) {
        Toast.makeText(getApplicationContext(), "Excepcion Beacon:
"+e.getMessage(), Toast.LENGTH_LONG).show();
        e.printStackTrace();
    }
}

```



```
//Hacemos que se nos notifique de alguna forma de los beacons que se han leído durante el periodo.  
mBeaconManager.addRangeNotifier(this);  
}
```

Inicio de búsqueda de beacons. Código Guía solidgargroup.com

- El método “didRangeBeaconsInRegion(Collection <Beacon> beacons,Region region)” es invocado tras el periodo de búsqueda definido y su función es informar sobre qué dispositivos se han detectado en la región. Es aquí donde diferenciamos en qué localidad del club de tenis nos encontramos y realizamos en consecuencia las acciones pertinentes. Dichas acciones consisten en llevar a cabo diversas peticiones al servidor, diferenciando para ello la instalación en la que nos ha detectado el dispositivo, y preparando los parámetros o datos necesarios. ParametrosEnvio (String instalación).

```
@Override  
public void didRangeBeaconsInRegion(Collection<Beacon> beacons, Region region) {  
  
    if (beacons.size() == 0) {  
        Toast.makeText(getApplicationContext(), "No se han detectado beacons",  
            Toast.LENGTH_LONG).show();  
    } else {  
        //Recorremos el array con los beacons detectados.  
        for (Beacon beacon : beacons) {  
  
            switch (beacon.getId2().toString()){  
  
                case "0x5a53586f7a34":  
                    ParametrosEnvio("TENIS");  
                    break;  
                ...  
                default:  
                    //Casos posteriores análogos al anterior  
                    break;  
            }  
        }  
    }  
}
```

Método encargado de identificar posibles dispositivos detectados. Código Guía: solidgargroup.com

```

public void ParametrosEnvio (String instalacion){

    //Obtención de valores necesarios. Se nos pasan de la actividad anterior.
    String numsocio = getIntent().getExtras().getString("numsocio");
    String codigoclub = getIntent().getExtras().getString("codigoclub");

    //HashMap para almacenar parametros a enviar.
    Map<String, String> params = new HashMap<String, String>();

    //Sacamos la hora con la que tendremos que tratar en algunos casos de
    instalaciones.
    java.util.Date fec = new Date();
    String mes = numeroMes(fec.toString().substring(4,7));
    String año =fec.toString().substring(30,34);
    String dia =fec.toString().substring(8,10);
    String fechacompr=año+"-"+mes+"-"+dia+"
    "+fec.toString().substring(11,19);

    switch(instalacion){

        case "TENIS":

            int horas = Integer.parseInt(fec.toString().substring(11,13));
            int minutos = Integer.parseInt(fec.toString().substring(14,16));
            // Se sacarán tres horas para realizar la comprobación horaria . Existe una
            diferencia de 30 minutos entre ellas.
            String hora1,hora2,hora3;

            //Casuística rango horario a buscar.(hora1<hora2<hora3)
            if(minutos>0 && minutos <=15){
                hora1 = Integer.toString(horas-1)+":30:00";
                hora2 = Integer.toString(horas)+":00:00";
                hora3 = Integer.toString(horas)+":30:00";

            }else if (minutos>15 && minutos <=30 || minutos>30 && minutos <=45){
                hora1= Integer.toString(horas)+":00:00";
                hora2= Integer.toString(horas)+":30:00";
                hora3= Integer.toString(horas+1)+":00:00";

            }else{
                hora1= Integer.toString(horas)+":30:00";
                hora2= Integer.toString(horas+1)+":00:00";
                hora3= Integer.toString(horas+1)+":30:00";
            }

            String fecha1,fecha2,fecha3,fechacompr;

            fecha1=año+"-"+mes+"-"+dia+" "+hora1;
            fecha2=año+"-"+mes+"-"+dia+" "+hora2;
            fecha3=año+"-"+mes+"-"+dia+" "+hora3;
            fechacompr=año+"-"+mes+"-"+dia;

```

```
params.put("tag", "TENISAPP");
params.put("num socio", num socio);
params.put("codigo club", codigo club);
params.put("fecha1", fecha1);
params.put("fecha2", fecha2);
params.put("fecha3", fecha3);
params.put("fecha compl compr", fecha compl compr);
params.put("fecha compr", fecha compr);

RealizaPeticionServidor(params);
break;
...
//Restantes casos de instalaciones.
} }
```

Método preparación de datos a enviar en aplicación

Como podemos observar, el método “ParametrosEnvio(Instalacion)” prepara los parámetros dependiendo de la instalación en cuestión, y tras ello, llama a otro encargado de realizar y gestionar la petición final. “RealizaPeticionServidor(Map<String,String>)” La funcionalidad de este último, será la misma que para la actividad anterior.

La aplicación también tendrá como objetivo mostrar información de interés sobre las instalaciones, así como gestionar un panel de notificaciones principal para el tratamiento de información asociada a las distintas detecciones. Así pues, esta segunda actividad estará organizada en diversos fragmentos o secciones, donde se alojará información relevante fruto de la explotación de los datos generados tras la reserva y acceso a cada instalación. A destacar el fragmento de inicio o principal, donde se gestionarán las notificaciones generadas con las detecciones.

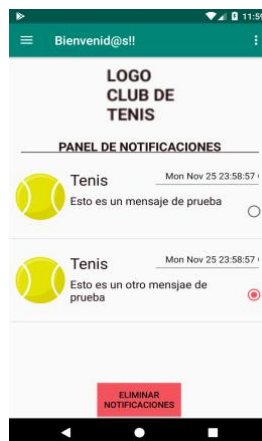


Fig.26: Panel principal de notificaciones

El método “Obtenermensajes()” será el encargado de realizar la gestión citada, empezando con la petición de los datos al servidor, siguiendo con la recepción de la respuesta asociada, y con ella el montaje e inserción de los mensajes en función de la distinción de usuarios realizada.

```
public void Obtenermensajes(){

    final java.util.Date fec = new Date();
    final Resources res = getResources();
    final RadioButton rb = new RadioButton(getContext());

    String tag_servidor = "ListaPrincipal";
    // Enviamos junto con la petición los datos introducidos por el usuario
    Map<String, String> params = new HashMap<String, String>();
    final String login = getActivity().getIntent().getExtras().getString("login");
    if (login.equals("Conserje") || login.equals("Restaurante") || login.equals("Admin") ){

        params.put("tag","LISTA_PRINCIPAL_EMPLEADOS");
        params.put("codigoclub",getActivity().getIntent().getExtras().getString("codigoclub"));
        params.put("login",login);
    }else{

        params.put("tag","LISTA_PRINCIPAL_SOCIOS");
        params.put("num socio",getActivity().getIntent().getExtras().getString("num socio"));
        params.put("codigoclub",getActivity().getIntent().getExtras().getString("codigoclub"));
    }
    JSONObject jsonObj = new JSONObject(params);

    JsonRequest jsonObjReq = new JsonRequest(Request.Method.POST,
    ConnectionConfig.URL_LOGIN, jsonObj,
        new Response.Listener<JSONObject>() {

            @Override
            public void onResponse(JSONObject response) {

                try {
                    switch(response.getString("Resultado")){

                        case "PRINCIPAL_REST":

                            for (int i =0; i< response.length()-1; i+=2){

                                mensajes.add(new Mensaje("Rest.", response.
                                getString(Integer.toString(i)), " "+ response. getString
                                (Integer.toString(i+1)).substring(0,16),
                                res.getDrawable(R.drawable.restaurante),rb));
                            }

                            Mensajes(mensajes);

                            break;
                    }
                }
            }
        }
    );
}
```

```
} catch (JSONException e) {
    e.printStackTrace();
}
}, new Response.ErrorListener() {
    @Override
    //Captura algun error que pueda surgir usando Volley.
    public void onErrorResponse(VolleyError error) {
        Toast.makeText(getApplicationContext(), error.getMessage(), Toast.LENGTH_LONG).show();
    }
});
// Añadimos la petición a la cola de peticiones de Volley
AppController.getInstance().addToRequestQueue(jsonObjReq, tag_servidor);
}
```

Método ObtenerMensajes() contenido en la actividad principal.

3.3.2 Implementación del servidor web.

Como bien se ha comentado anteriormente, el servidor contendrá una base de datos la cual será el núcleo de la aplicación y del sistema. Ahora bien, a la hora de gestionar los posibles accesos desde el exterior, bien sea desde una aplicación como desde otro servidor, necesitaremos de uno o varios ficheros que se encarguen de atender las peticiones y realizar las funcionalidades asociadas. De este modo, se introducen los ficheros principales para atender las peticiones del presente proyecto:

- “Index.php”. Hablamos del fichero principal, el cual se referenciará cuando llegue una petición. La forma de diferenciar las distintas peticiones dentro de este será gracias a un “tag”, que no es más que un identificador de la petición. Por parte de la aplicación, podremos recibir peticiones de las dos principales actividades:

Los “tags” que empiezan por “LOGINAPP_”, corresponden a las peticiones por parte de la primera actividad y cuya funcionalidad versa en realizar una verificación de los datos del usuario, así como notificarle una respuesta. En caso de ser correctos, se proporcionan además algunos datos de ese usuario en concreto, que posteriormente se usarán en la segunda actividad.

Dentro de la segunda actividad será donde se gestione la detección de los dispositivos y, en consecuencia, esta derivará en una petición al servidor. Los “tags” implicados contendrán una abreviación de la instalación a la que se refieran junto con la terminación “APP”. Todos ellos realizarán comprobaciones y operaciones de cualquier índole sobre la base de datos.

Finalmente, también podemos recibir peticiones de obtención de datos, estos serán usados para mostrarse en forma de lista en la aplicación. Estas empezarán por “LISTA_”.

```

<?php

include ('Funcionalidad.php');

//Obteniendo los parametros de la peticion APP
$json = json_decode(file_get_contents('php://input'), true);

//Cusuistica:
switch($json['tag']){
//Caso pantala LoginActivity
case 'LOGINAPP_SOCIOS':

    $resultado = logearSocio($json['login'],$json['contra'],'1234');

    if ($resultado == 'LOGIN_OK'){

        echo infoSocio($json['login'],$resultado);

    }
    else {

        $arr_inforesult = array('Resultado' => $resultado);
        header('Content-type: application/json');
        echo json_encode($arr_inforesult,JSON_FORCE_OBJECT);

    }
    break;

case 'LISTA_PRINCIPAL_SOCIOS':

//Funcionalidad asociada

break;
...
?>

```

Estructura fichero principal servidor index.php

- “Funcionalidad.php”. Como bien su nombre indica, contiene todas las funciones necesarias para conseguir la funcionalidad que se pretende. Las funciones contenidas en él son llamadas desde el fichero anteriormente descrito y su objetivo será el de interactuar con la base de datos a fin de completar operaciones. Pasamos a ver un ejemplo típico:

```

//Devuelve informacion sobre el socio que se ha logeado.
function infoSocio ($login, $resultado){

    $con = mysqli_connect("localhost", "root", "numeros1919","GESTION_CLUB_DE_TENIS") or die(mysqli_error());

    $infosocio = mysqli_fetch_row(mysqli_query($con,"SELECT Id_socio,Nombre,Apellidos,Foto,Id_ClubTenis from Socio WHERE Login ='$login'"));

    $fotoclub = mysqli_fetch_row(mysqli_query($con,"SELECT Logo from Club_De_Tenis WHERE Id_ClubTenis ='$infosocio[4]'"));

    $arr_infosocio = array ('Resultado'=>$resultado,'Numsocio'=>$infosocio[0] , 'Nombre'=> $infosocio[1], 'Apellidos'=>$infosocio[2], 'Foto'=> $infosocio[3], 'CodigoClub'=> $infosocio[4], 'LogoClub'=> $fotoclub[0], 'Login'=>$login);
    header('Content-type: application/json');
    $file= json_encode($arr_infosocio,JSON_FORCE_OBJECT);

    return $file;
}

```

Ejemplo típico en funcionalidad.php

4 Implantación y pruebas.

Tratamos con un sistema orientado a explotar el acceso a las instalaciones. Inicialmente el usuario será detectado a la hora de utilizar la instalación; se le comunicarán al servidor los datos del acceso; se procederá con un tratamiento de los datos obtenidos, y finalmente fruto de ese tratamiento, se mostrarán en la aplicación tanto avisos como resultados y estadísticas asociados a los accesos. Ahora bien, para que nuestro sistema sea de utilidad, se requerirá de un paso previo complementario, esto es, un procedimiento de reserva de la instalación. Dicho procedimiento mencionado, ya se encuentra implantado en la mayoría de los clubes mediante páginas web. Así pues, el proyecto estará enfocado hacia la aplicación genérica de nuestro sistema a cualquier club de tenis con un sistema de reserva web ya implantado.

De esta manera la estructura presentada anteriormente en el punto "3.2.2-Arquitectura del sistema", con la adaptación pertinente quedaría de la siguiente manera:

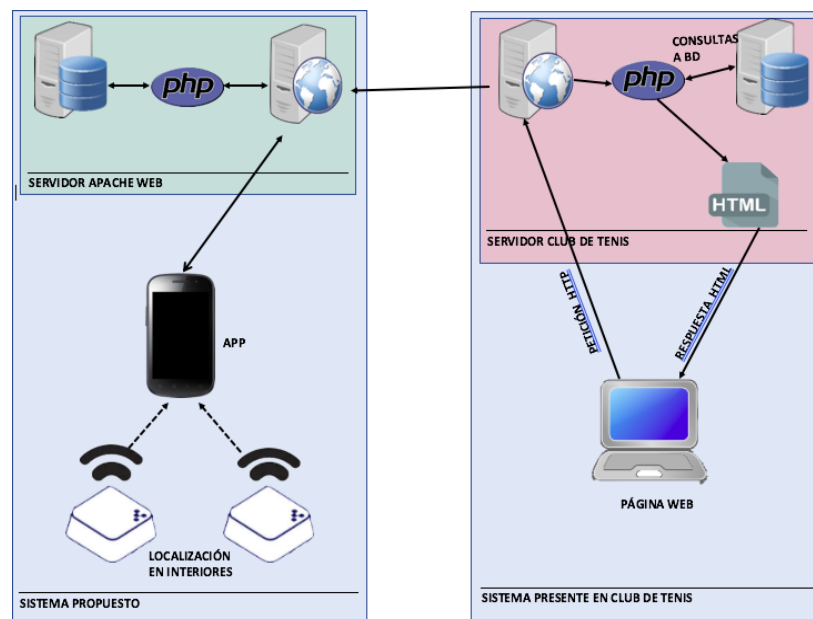


Fig.27: Estructura global de la implantación

Volviendo a la implantación, deberemos de realizar dos tareas fundamentales para completarla:

En primera instancia, la migración de los datos ya existentes sobre el club contenidos en la base de datos origen. Para este punto habrá que tener en cuenta el gestor de la base de datos con el que cuenta el servidor origen, pues en función de este, la metodología seguida para completar la

migración variará. También será necesario emprender un análisis para el caso en concreto, y de esta forma valorar qué información es necesaria traspasar.

La segunda consistirá en la redirección de los datos de reserva introducidos por el usuario hacia nuestro servidor. Para el caso del servidor origen deberemos modificar el fichero que define y gestiona la estructura del formulario, mientras que en el destino será necesario la adaptación del fichero principal para contemplar la llegada de las peticiones del exterior con su posterior tratamiento.

4.1 Caso de implantación

Las características del entorno implantado fueron las siguientes:

- Club de tenis con un sistema de información formado por un servidor encargado de alojar toda su información en bases de datos, así como una plataforma web informativa con posibilidad de realizar reservas de las instalaciones.
- Presenta una página web desarrollada mediante la herramienta WordPress.

Antes de proceder con la implantación, deberemos tener en cuenta las siguientes comprobaciones:

- Asegurarnos de tener suficiente espacio en el disco. Ello dependerá de la cantidad de tablas a migrar. De todas maneras, nunca alcanzará un tamaño considerable para nuestro caso puesto que las tablas a traspasar no serán muchas.
- Si se da el caso en que las tablas a migrar coincidan en nombre, tendremos que renombrarlas previamente.

La base de datos del club seguía una gestión mantenida por MySQL, así pues, se estudió el procedimiento a seguir con la migración entre servidores MySQL:

Se trasladaron a un fichero las tablas a migrar (básicamente las tablas “Socio_web” y “Instalacion_web”) y posteriormente, una vez internados en la ruta **C:\xampp\mysql\bin**, se procedió con la ejecución del comando `mysqldump` desde el servidor origen. Se le facilita la base de datos junto con las tablas afectadas.

```
C:\xampp\mysql\bin>mysqldump -h localhost -u root -p gestion_club_de_tenis socio
_web instalacion_web >c:\xampp\phpmyadmin\RespalDOS\respaldoClub.sql
Enter password: ****
```

Fig.28: Respaldo de bases de datos origen

Tras ello, seguimos con la transferencia del archivo de un servidor a otro. Para este caso, optando por el comando scp, se enviaron archivos entre servidores a través de un enlace encriptado y seguro ssh utilizando el puerto 22. Requiere de una ruta de directorio de los dos servidores implicados, así como tener habilitado el acceso SSH desde ambos dispositivos.

```
jordi@jordi-PC MINGW32 /c/XAMPP/phpmyadmin/Respaldos
$ scp respaldoClub.sql 192.168.0.159:/Applications/XAMPP/xamppfiles/phpmyadmin/R
espaldos
Password:
respaldoClub.sql                               100% 3105    96.2KB/s   00:00
```

Fig.29: Transferencia de datos desde servidor origen

Finalmente importamos el fichero respaldado en nuestro servidor introduciendo en **/Applications/XAMPP/bin** el siguiente comando:

```
mysql -u [root] -p GESTION_CLUB_DE_TENIS < respaldoClub.sql
```

Una vez alojados los datos, se pasó el contenido de la tabla recibida a la tabla correspondiente, usando para ello instrucciones básicas de SQL mediante PHP:

1.-Nos situamos en el directorio donde se encontraba el fichero volcado.

```
chdir('192.168.0.159/phpmyadmin/Respaldos');
```

2.-Establecimos conexión con nuestra base de datos a fin de realizar las consultas pertinentes, y completamos las adaptaciones pertinentes para algunos casos.

```
$con = mysqli_connect("localhost", "root", "numeros1919", "GESTION_CLUB_DE_TENIS ")
or die (mysqli_error());
```

```
$dinstal = mysqli_fetch_row(mysqli_query($con,"SELECT Id_instalacion from
instalacion_web WHERE Login =$login"));
```

```
mysqli_query($con,"INSERT INTO Socios ( Id_ClubTenis, Id_Socio, Nombre, Apellidos,
DNI, Fecha_nacimiento, Fecha_inscripcion, Login, Contraseña,Foto ) SELECT Id_ClubTenis,
Id_Socio, Nombre, Apellidos,DNI, Fecha_nacimiento, Fecha_inscripcion, Login,
Contraseña,Foto FROM socio_Web ");
```

```
mysqli_query($con, "INSERT INTO Instalacion (Id_ClubTenis, Id_TipoInst, Id_Instalacion,
Nombre, Superficie) SELECT Id_ClubTenis, substr(Id_instalacion,0,-1),substr(Id_Instalacion,-
1),Nombre,Dimensiones FROM instalacion_Web ");
```

```
mysqli_query($con,“DELETE FROM socio_Web”);
```

```
mysqli_query($con,“DELETE FROM instalacion_Web”);
```

Con el segundo y último paso, se estableció un canal de flujo de datos entre los dos servidores.

- En la parte del servidor del club de tenis, se alteraron los ficheros concernientes a los formularios de envío de reserva. Pasamos a mostrar un ejemplo:

```
<?php
/*
Template name: Reserva_pistas
*/
get_header();
global $wpdb;
//Obtiene los valores introducidos por el usuario en el formulario
$login1= $_POST['login1'];
$login2= $_POST['login2'];
$instalacion= $_POST['pista'];
$fechaE= $_POST['fechaE'];
$horaE= $_POST['horaE'];
$fechaS= $_POST['fechaE'];
$horaS= $_POST['horaS'];

//sql
//Guardado en la base de datos local (WORDPRESS)
$sql= "INSERT INTO `formulario_resrv` (`Login`,`Instalacion`,`FechaE`,`HoraE`,`FechaS`,`HoraS`) VALUES ('$login1','$instalacion','$fechaE','$horaE','$fechaS','$horaS')";

$sql1= "INSERT INTO `formulario_resrv` (`Login`,`Instalacion`,`FechaE`,`HoraE`,`FechaS`,`HoraS`) VALUES ('$login2','$instalacion','$fechaE','$horaE','$fechaS','$horaS')";

$wpdb->query($sql);
$wpdb->query($sql1);

?>
<div id="primary">
<div id="content" role="main">
<div id="center">
<h2>Reserva de pistas de tenis </h2>

<!-- Redirección de la información introducida por el usuario -->
<form method="post" enctype="multipart/form-data" action = "http://192.168.0.159/ProyectoTenis/">

<input type="hidden" name="tagW" value="TENISWEB"/>

<label>Introduzca login Socio 1</label>
<input type="text" name="login1"/>
</br></br>

<label>Introduzca login Socio 2</label>
<input type="text" name="login2"/>
</br></br>
...
<?php get_footer(); ?>
```

Fragmento de código ejemplo formulario html

Como podemos observar en la figura anterior, hemos modificado el encabezado de la etiqueta “form”. Aquí es donde se definirá entre otras, el método y la dirección de envío de los datos introducidos en el formulario. También será necesario el envío de una variable extra por parte del servidor origen, el “tag”, necesaria para que nuestro servidor destino pueda identificar la petición.

Introducimos ahora, los cambios realizados en los dos ficheros de nuestro servidor:

En el primero, añadimos algunas líneas para tratar la petición recibida con su posterior redirección a un método que complete la funcionalidad deseada.



```
//Recepcion reservas por parte del exterior.
switch($_POST['tagW']){
case 'TENISWEB':
//echo $_POST['fechaE'];
guardareservaTEN($_POST['login1'],$_POST['login2'],$_POST['pista'],$_POST['fechaE'],$_POST['horaE'],$_POST['fechaE'],$_POST['horaS']);
header('Location: http://192.168.0.161:8080/clubteniswordpress/');
break;
case '':
//funcionalidad asociada
..
}
}
```

Modificación fichero index.php del servidor

```
//Convierte y guarda los datos de reserva pista de tenis en la base de datos.
function guardareservaTEN($login1,$login2,$instalacion,$fechaE,$horaE,$fechaS,$horaS){
$con = mysqli_connect("localhost", "root", "numeros1919","GESTION_CLUB_DE_TENIS") or die(mysqli_error());

//Preparamos los datos externos para almacenarlos en nuestra bd.
$infosocio1 = mysqli_fetch_row(mysqli_query($con,"SELECT Id_ClubTenis,Id_Socio from Socio WHERE Login ='$login1'"));
$infosocio2 = mysqli_fetch_row(mysqli_query($con,"SELECT Id_ClubTenis,Id_Socio from Socio WHERE Login ='$login2'"));
$idInstal=substr($instalacion, -1);
$idTipo= substr($instalacion,0,-1);
$FechaEntr = mysqli_fetch_row(mysqli_query($con,"SELECT CONCAT('$fechaE',' ','$horaE')"));
$FechaSal = mysqli_fetch_row(mysqli_query($con,"SELECT CONCAT('$fechaS',' ','$horaS')"));

mysqli_query($con,"INSERT INTO `Reserva`(`Id_ClubTenis`,`Id_TipoInst`,`Id_Instalacion`,`Id_Socio`,`Fecha_Reserva_E`,`Fecha_Reserva_S`,`Fecha_Compr`) VALUES ('$infosocio1[0]','$idTipo','$idInstal','$infosocio1[1]','$FechaEntr[0]','$FechaSal[0]','$FechaEntr[0]')");

mysqli_query($con,"INSERT INTO `Reserva`(`Id_ClubTenis`,`Id_TipoInst`,`Id_Instalacion`,`Id_Socio`,`Fecha_Reserva_E`,`Fecha_Reserva_S`,`Fecha_Compr`) VALUES ('$infosocio2[0]','$idTipo','$idInstal','$infosocio2[1]','$FechaEntr[0]','$FechaSal[0]','$FechaEntr[0]')");
}
}
```

Ejemplo funcionalidad recepción de datos de reserva en fichero servidor funcionalidad.php

En el segundo fichero, adaptamos la funcionalidad para este último caso dado. Básicamente se realizó una inserción en nuestra tabla “Reserva” después de haber tratado los datos para su correcto guardado, ahora bien, dicho tratamiento podrá variar en función del club de tenis. Para nuestro caso, el tratamiento no fue excesivamente complejo:

- Consulta para sacar Id_ClubTenis como Id_Socio usando el login facilitado.
- En origen se nos facilita la instalación (tipo e identificativo), todo en un mismo campo. Tratamiento pertinente mediante le método SUBSTRING para separarlo y así poder almacenarlo en nuestro sistema.
- En cuanto las fechas de entrada y salida, nuestro sistema los almacena en un campo de tipo “DateTime” y en origen vienen separados. Así pues, lo fusionamos empleando la función de SQL CONCAT.

- Se introdujo un campo necesario que a priori no se utiliza en origen, usamos cualquiera de las Fechas recibidas. El campo en cuestión será Fecha_Compr.

4.2 Pruebas

Tras la debida implantación citada, se prosiguió con las posteriores pruebas a fin de verificar su correcto funcionamiento.

Empezamos comprobando el estado de las bases de datos al inicio de la presenta prueba:

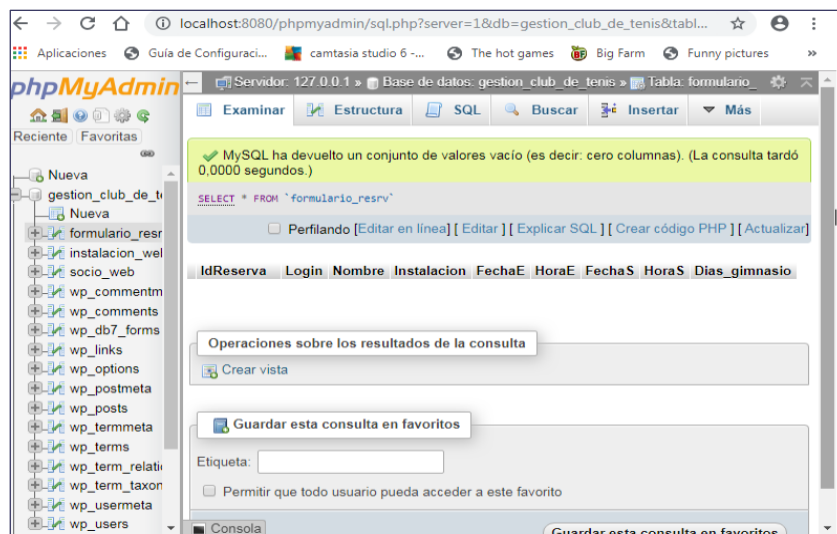


Fig.30: Estado inicial de la tabla "formulario reserva" en servidor club de tenis

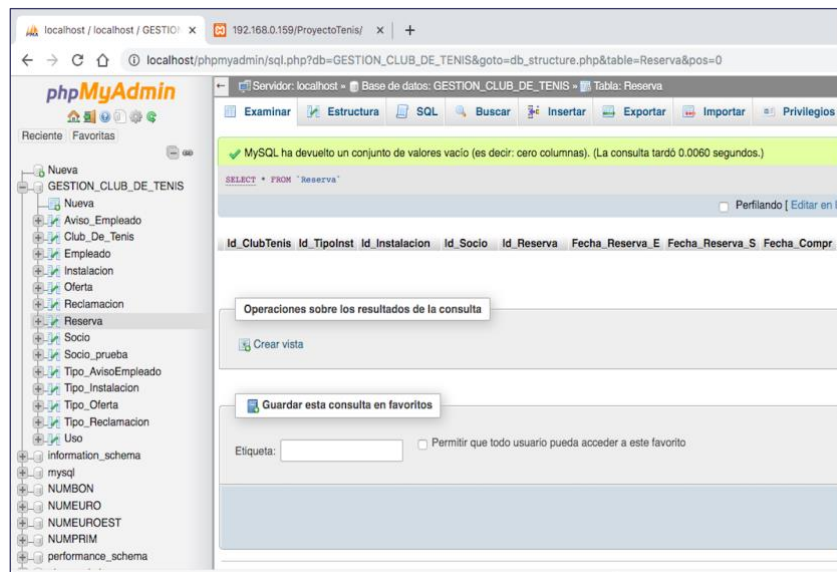


Fig.31: Estado inicial de la tabla "Reserva" en servidor local

Sistema para el control de acceso a las instalaciones de un club de tenis

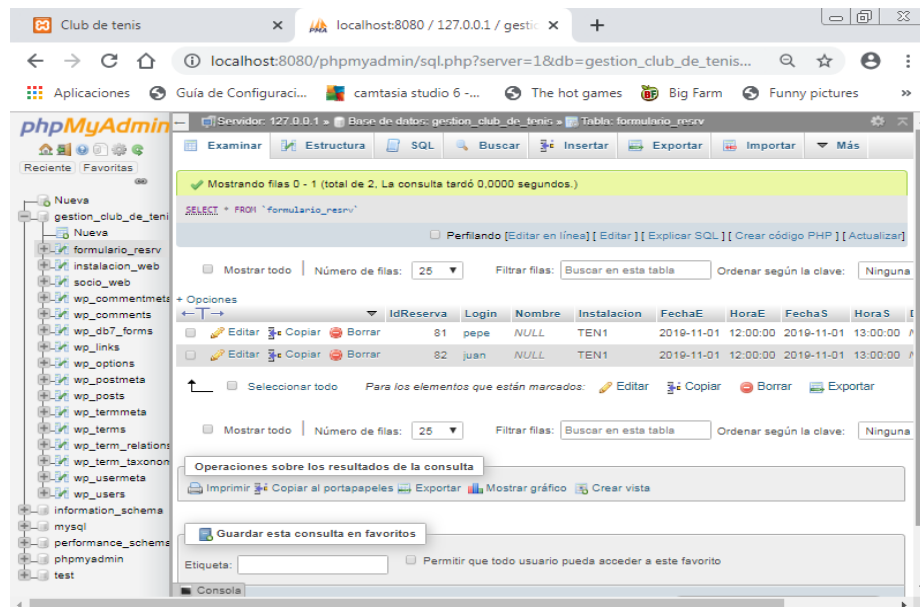
Seguidamente procedimos a realizar varias reservas desde el servidor origen, entre ellas esta:



The screenshot shows a web interface titled "Club de tenis" with a navigation menu (Inicio, Tenis, Gimnasio, Piscina, Restaurante «Nombre», Contacto). The main heading is "Reserva de pistas de tenis". The form includes fields for "Introduzca login Socio 1" (pepe), "Introduzca login Socio 2" (juan), and "Introduzca contraseña" (masked). Below is a "Seleccione pista" dropdown menu set to "Pista tierra 1". The "¿Cuándo?" section has "Fecha" (01/11/2019), "Hora inicio" (12:00), and "Hora fin" (13:00). A red "REALIZAR RESERVA" button is at the bottom.

Fig.32: Reserva de instalación TENIS desde Web

Una vez completada esta, volvimos a las tablas mencionadas anteriormente para corroborar que, en efecto, los datos introducidos se habían traspasado a los servidores pertinentes:



The screenshot shows phpMyAdmin displaying the 'formulario_reserv' table. The table contains two rows of reservation data:

IdReserva	Login	Nombre	Instalacion	FechaE	HoraE	FechaS	HoraS
81	pepe	NULL	TEN1	2019-11-01	12:00:00	2019-11-01	13:00:00
82	juan	NULL	TEN1	2019-11-01	12:00:00	2019-11-01	13:00:00

Fig.33: Estado final de la tabla "formulario reserva" en servidor club de tenis tras la reserva web

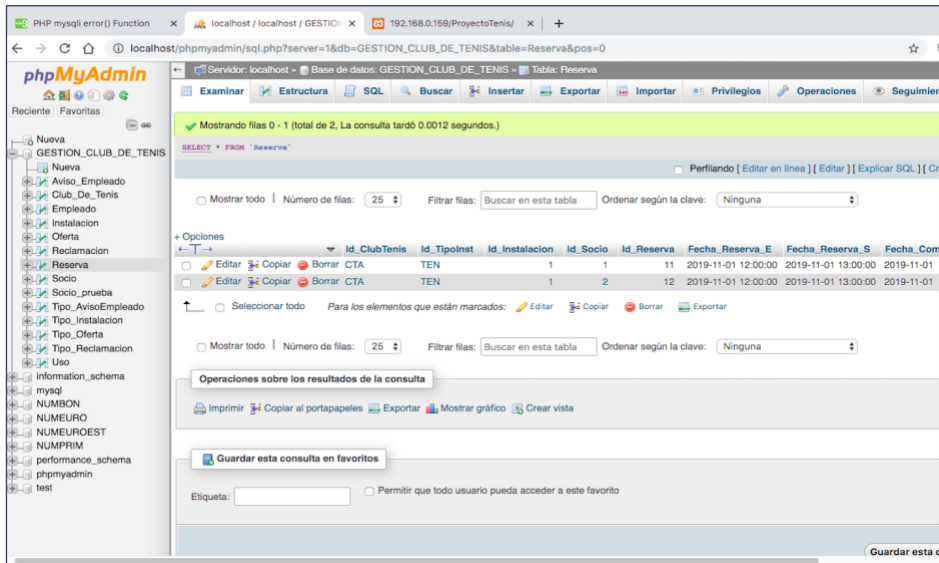


Fig.34: Estado final de la tabla “Reserva” en servidor local tras la reserva web

Una vez concluido la reserva de la instalación, llegó el proceso consecutivo de acceso a la instalación. Aquí, siguiendo con la funcionalidad propuesta para la instalación dada:

- El usuario es detectado a la entrada de la instalación, ocasionando una actualización del registro de la reserva en cuestión. Concretamente se habrán actualizado las columnas de Entrada y Resultado_E.

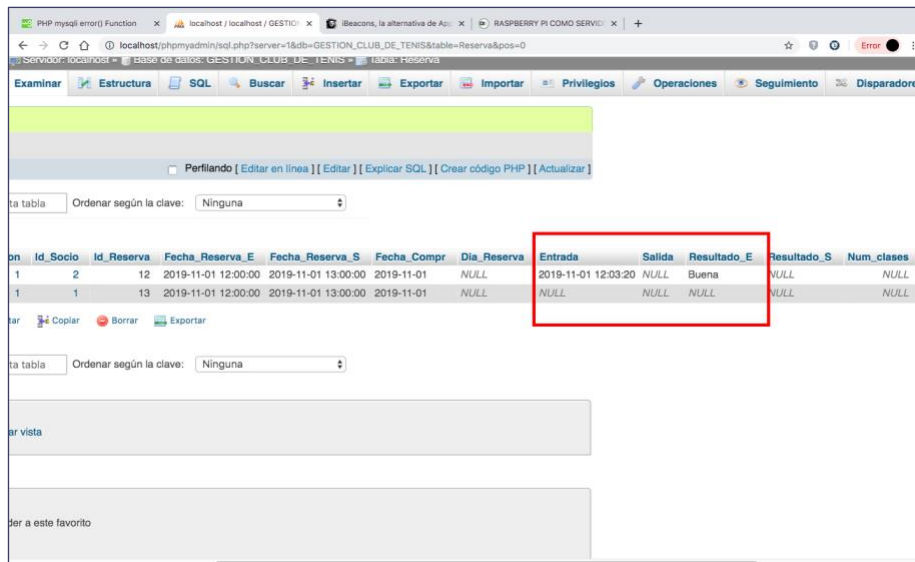


Fig.35: Estado de la tabla “Reserva” en nuestro servidor tras la detección de acceso de los socios

Sistema para el control de acceso a las instalaciones de un club de tenis

Este caso, representa la situación en que uno de los dos jugadores implicados llega con anterioridad al otro, pues el uno de los registros implicados tiene la columna “Entrada” a Null.

Así pues, se habrá actualizado el valor de la columna “Mensaje_Aviso” para el contrincante que aún no está en la instalación. Obteniéndolo a su vez, este aviso, gracias al panel de notificaciones de la aplicación.

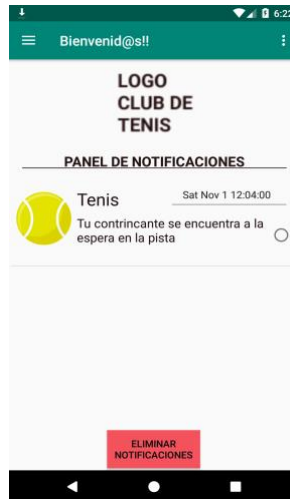


Fig.36: Notificación: contrincante a la espera.

- Posteriormente, a la salida de la instalación, con la actualización de algunas columnas como Salida, Resultado_S, Resultado_Asistencia y Mensaje_Resultado, se generó la siguiente notificación.

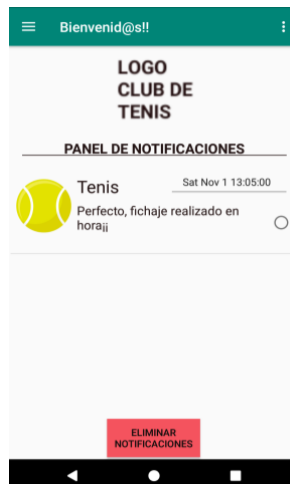


Fig.37: Notificación tras fichaje

5 Conclusiones.

Analizando el resultado aportado tras la implantación, podemos concluir las siguientes líneas:

Con el presente desarrollo se consigue alcanzar un mayor control sobre el uso de las instalaciones, y con él, una gestión más ajustada a la realidad. Prueba de ello será la importante reducción obtenida con el tiempo de “no uso” de las instalaciones. Y es que, los procesos utilizados anteriormente, no eran capaces de llevarlo a cabo a ese nivel debido a la carencia de medios físicos para mantenerlo.

Hablamos ahora del valor añadido generado al socio con el uso del sistema. El sistema trata de complementarse con el socio, haciéndolo más partícipe del club e invitándolo al uso proactivo de sus instalaciones. Todo ello se ha logrado gracias al servicio de notificaciones implementado; entre las notificaciones que soporta dicho servicio podemos destacar: resultado y recordatorio de asistencia, información de relevancia para los empleados encargados, así como otros avisos asociados.

Indirectamente, dicho valor mencionado también ha sido de gran ayuda a la hora de analizar el club desde la perspectiva de entidad que vela por su sostenibilidad.

En lo que concierne a términos tecnológicos:

A fin de abordar la problemática planteada, se procedió con la búsqueda de algún elemento tecnológico capaz de comunicar o recibir una señal unívoca asociada. Tras la conclusión de dicha búsqueda, fue notorio pensar que debía de lidiar con el desarrollo de una aplicación móvil, la cual, explotando alguna tecnología en común, interactuase con ellos. Para llevarlo a cabo, se empezó representando el sistema propuesto, haciendo uso de conceptos vistos en Ingeniería del software (ISW) y Diseños de Bases de datos (BDD). Seguidamente, con el desarrollo de la aplicación, se pusieron en práctica conocimientos sobre programación en java vistos en la asignatura de Programación (PRG), así como también nociones sobre diseño de interfaces dados en (IPC).

La segunda parte del proyecto, conformada por el desarrollo de la lógica de negocio en el servidor, fue la más desconocida para mí puesto que tuve que adentrarme en el lenguaje PHP y HTML. Así pues, es en este punto donde me han surgido un poco más de problemas, ya que no tenía noción previa alguna.

Finalmente, fue también necesario el uso de conceptos impartidos en asignaturas como Redes de computadores (RED) o (TSR) a la hora de emplear tecnologías relacionadas con la interconexión de los dispositivos y la transferencia de datos entre estos.

Para mí, este proyecto ha sido un verdadero desafío a mis capacidades. Me ha ayudado a poner en práctica muchos de los conceptos vistos durante el grado, así como a tener que empezar con otros desde cero. El estudio de estos últimos, junto con la realización a la par de prácticas en empresa, han propiciado un atraso en la fecha de entrega que tenía prevista en un principio. Sobre todo las prácticas, que ocupan prácticamente la totalidad de mis días; prácticas que ya hace meses son extracurriculares, y con las cuales me he desenvuelto bastante bien.

6 Líneas Futuras

En primer lugar, se van a presentar algunas ampliaciones o mejoras a realizar:

- Actualmente, la funcionalidad asociada a las actividades de gimnasio consiste simplemente en llevar una contabilización de las clases a las que ha asistido el socio en concreto. Pues bien, se pretenderá ampliarlo para que se generen avisos de resultado de asistencia, de igual forma que se emplean en las pistas de tenis.
- Albergar la posibilidad de reserva y detección para partidos de dobles en tenis. Así pues, tendremos que adaptar la casuística de reserva y acceso de 2 a 4 jugadores.
- Se tratará de un nuevo aviso, este se lanzará cuando de forma imprevista, una pista quede libre. Su función será: obtener el nombre de los socios que mas suelen jugar en este tipo de instalación que ha quedado vacía de forma imprevista y enviarlos una notificación para informarles de la noticia.
- Avanzar con la explotación de los datos, pero enfocado hacia datos globales o generales, mostrando sus estadísticas y valores asociados.

Finalmente, vamos a introducir una posible expansión del proyecto hacia un nuevo ámbito:

- Hoy en día, se están empezando a usar cada vez más los relojes inteligentes o SmartWatch, preparados con la más puntera tecnología. Así pues, la idea consiste en estudiar y en su caso integrar en modo de aplicación, la funcionalidad descrita en el proyecto para estos dispositivos. Serían perfectamente capaces de recibir detecciones ya que incluso los modelos más estándares poseen Bluetooth 4.0, así como también se podría

enviar la posterior información asociada a la detección mediante WiFi o la Red de datos (según sea el caso).

7 Bibliografía

Academiaandroid, 2014: *Academia Android, IDE: Entornos Integrados de Desarrollo para Android*. En : <https://academiaandroid.com/ide-entornos-integrados-de-desarrollo-para-android/> [Consulta: 10 de junio 2019]

Androidcurso, 2017: *Máster en desarrollo de aplicaciones Android, Alternativas en los servicios Web*. En: <http://www.androidcurso.com/index.php/recursos/43-unidad-10-internet-sockets-http-y-servicios-web/327-alternativas-en-los-servicios-web/> [Consulta: 31 de julio de 2019]

Apachefriends, 2018: *Acerca de Apache Friends*. En: <https://www.apachefriends.org/es/about.html/> [Consulta: 16 de junio de 2019]

Booch, G., Rumbaugh, J. y Jacobson, I., 2014. *The Unified Modeling Language User Guide*. Westfort, MA (EE. UU.): Addison-Wesley

Desarrolloweb,2019: *PHP* . En: <https://desarrolloweb.com/home/php/> [Consulta: 22 de julio de 2019]

Developer.android,2019: *Descripción general de Volley*. En: <https://developer.android.com/training/volley/> [Consulta: 25 de junio de 2019]

González, D., 2017: *Beacons en Android*. En: <https://solidgeargroup.com/beacons-en-android/?lang=es/> [Consulta: 3 de Agosto de 2019]

Hébuterne, S., 2014: *Android: guía de desarrollo de aplicaciones Java para smartphones y tabletas*. Ediciones ENI, 2014

Kantar, 2019: *Smartphone OS sales market share*. En: <https://www.kantarworldpanel.com/global/smartphone-os-market-share/> [Consulta: 10 de septiembre 2019]

Paredes, M.; Rocha, A., 2019: *Revista Ibérica de Sistemas e Tecnologias de Informação*, Tecnología de la Información y Sistemas, Lousada, nº E17

Php, 2019: *Documentation*. En : <https://www.php.net/docs.php/> [Consulta: 27 de julio de 2019]

Raya,A., 2013: *Elespañol.com*, iBeacons, la alternativa de Apple a las compras por NFC. En: https://www.lespañol.com/omicrono/tecnologia/20130912/ibeacons-alternativa-apple-compras-nfc/20248029_0.html/ [Consulta: 28 de junio de 2019]

Seta, L., 2008: *Dosideas.com*, Introducción a los servicios web RESTful. En: <https://dosideas.com/noticias/java/314-introduccion-a-los-servicios-web-restful/> [Consulta: 5 de Agosto de 2019]

Wikipedia-Android Studio, 2017: *Es.wikipedia.org*, Android_Studio. En: https://es.wikipedia.org/wiki/Android_Studio/ [Consulta: 2 de julio de 2019]

Wikipedia-Servicio Web, 2017: *Es.wikipedia.org*, Servicio_Web. En: https://es.wikipedia.org/wiki/Servicio_web/ [Consulta: 10 de mayo de 2019]

Wikipedia-Servidor_http_apache, 2019: *Es.wikipedia.org*, Servidor HTTP Apache. En: [https://es.wikipedia.org/wiki/Servidor_HTTP_Apache /](https://es.wikipedia.org/wiki/Servidor_HTTP_Apache/) [Consulta: 13 de julio de 2019]

Wikipedia-ServidorWeb, 2017: *Es.wikipedia.org*, Servidor Web. En: [https://es.wikipedia.org/wiki/Servidor_web /](https://es.wikipedia.org/wiki/Servidor_web/) [Consulta: 13 de julio de 2019]

Wikipedia-WordPress,2019: *Es.wikipedia.org*, WordPress. En: <https://es.wikipedia.org/wiki/WordPress/> [Consulta: 15 de junio de 2019]

Wordpress,2019: *Es.WordPress.org*, En: <https://es.wordpress.org/> [Consulta: 15 de junio de 2019]