Aplicación Cliente del sistema de control de presencia TamTam-Home para dispositivos Android

Pablo Taboada Rodríguez
Septiembre de 2012

1. Introducción

TamTam-Home (www.tamtamhome.com, en adelante TTH) es un sistema de control de presencia basado en llamadas telefónicas para el seguimiento y monitorización de las entradas y salidas de personas en el hogar. Para su funcionamiento, las personas a controlar deben realizar dos llamadas telefónicas (una a la entrada y otra a la salida) desde un teléfono fijo del domicilio. Por medio de dichas llamadas telefónicas, la aplicación le muestra al usuario autorizado los datos relativos a asistencia, puntualidad, estadísticas, etc, todo ello en tiempo real.

De este modo, el propietario de una vivienda puede tener información puntual de las entradas y salidas de la persona que viene a limpiar, el jardinero, el profesor de los niños o el cuidador de los ancianos. Junto a los datos estadísticos de horas efectivas trabajadas, el sistema emite una serie de alertas por entrada/salida (o no entrada/salida) a la hora planificada.

En este proyecto se plantea la elaboración de una aplicación cliente para la aplicación TTH que correrá sobre dispositivos Android. Dicha aplicación permitirá a un usuario del sistema TTH acceder a sus datos y hacer un seguimiento en tiempo real de las entradas y salidas de su personal desde dispositivos Android.

1.1. Objetivo

Desarrollar una aplicación para dispositivos Android que permita el uso de la aplicación TTH para usuarios registrados en la misma.

1.2. Justificación

La empresa titular del sistema TTH busca ampliar las capacidades de su sistema de modo que pueda ofertar el acceso al mismo a sus usuarios mediante dispositivos móviles. La justificación para dicho desarrollo se fundamenta en el importante crecimiento que están experimentando los dispositivos móviles, tanto teléfonos como tabletas.

Si bien hace unos años se tendía a pensar en desarrollar aplicaciones pensando en usuarios con dispositivo móvil bien diferenciados de los usuarios con estación de trabajo

(PC/portátil), la realidad actual muestra que la gran mayoría de los usuarios usarán ambas plataformas. Si antes se pensaba en que “la mitad de tus usuarios accederán desde dispositivos móviles”, hoy debemos pensar en que “todos tus usuarios accederán la mitad del tiempo desde dispositivos móviles”. Este cambio de paradigma hace que un proyecto de este tipo esté más que justificado si se quiere ofrecer una experiencia de calidad al usuario final del sistema.

Aunque la empresa plantea inicialmente el desarrollo genérico de la aplicación para un entorno móvil, tras un primer estudio de las plataformas y sistemas disponibles, han decidido comenzar el desarrollo para terminales con sistema operativo Android, y una serie de funcionalidades limitadas.

1.3. Análisis de lo existente

La aplicación TTH existente dispone de multitud de funcionalidades para el seguimiento de entradas y salidas del personal, gestión de planificaciones, alarmas por incumplimiento, estadísticas, gestión de centros y empleados, etc, etc, etc.

Al tratarse de un sistema completo y ya en producción, este proyecto se plantea como una extensión del mismo, debiendo respetar en todo momento la funcionalidad y los parámetros de seguridad de TTH tal y como existe en la actualidad.

En la web de la aplicación (www.tamtamhome.com) se puede comprobar el funcionamiento básico del sistema y sus principales funcionalidades.

1.4. Entrevistas con el cliente

Partiendo de la solicitud del cliente para el desarrollo de una aplicación móvil para TTH, y tras un primer análisis, se le han planteado al mismo una serie de restricciones iniciales a tener en cuenta.

Dentro del abanico de los dispositivos móviles se ha optado por el desarrollo para dispositivos Android por tratarse de un sistema abierto. En cualquier caso, se debe perseguir el objetivo de un diseño sólido que permita el posterior desarrollo de la aplicación para dispositivos con sistema iOS del modo más sencillo posible.

Este proyecto se plantea como una primera aproximación al desarrollo completo de la aplicación TTH para dispositivos móviles. Como tal, servirá de testeo inicial para poder evaluar el camino a seguir para completarlo en etapas posteriores. Por este motivo, las funcionalidades del mismo deben ser acotadas, evitando plantear un proyecto que persiga disponer de las funcionalidades completas de TTH. Si este proyecto se completa con éxito y a satisfacción del cliente, se plantearán proyectos de ampliación del mismo, por lo que el desarrollo debe contemplar estas posibles ampliaciones.

Del mismo modo, al tratarse de un primer proyecto de desarrollo del sistema TTH para dispositivo móviles, se aconseja al cliente que la aplicación sea de “sólo-lectura”, de modo que se eviten problemas de seguridad añadidos en un primer desarrollo. Al igual que en el punto anterior, en proyectos posteriores se ampliarán las funcionalidades para poder editar datos del sistema desde dispositivos móviles.

Tras una serie de entrevistas con el cliente se recogen en la propuesta detallada una serie de requisitos que debe cumplir el sistema a desarrollar, además de los condicionantes de desarrollo a tener en cuenta. En la entrevista inicial el cliente ya plantea que desea que los datos a mostrar en el dispositivo móvil sean siempre actualizados, si bien muestra su preocupación por la precariedad de las conexiones móviles, de modo que desearía que esto no influyera en la velocidad de carga de la aplicación móvil.

1.5. Propuesta detallada

Se plantea el desarrollo de un sistema cliente-servidor para cubrir las funcionalidades deseadas por el cliente.

La parte servidora debe ser utilizable posteriormente si llega a desarrollarse un aplicativo TTH para dispositivos con sistema iOS. Para ello se elaborará una API simple para acceder a los datos de la aplicación TTH del modo más normalizado posible, de modo que se facilite el desarrollo posterior para otras plataformas.

La parte cliente debe ser específica para dispositivos Android, de sólo lectura, y debe mostrar los datos en tiempo real, de modo que el usuario final pueda tener la información en su dispositivo siempre actualizada con respecto a la realidad de lo que está ocurriendo en su domicilio. Para asegurar que los datos mostrados en el dispositivo

móvil sean siempre actualizados y para simplificar el desarrollo de la aplicación, ésta carecerá de almacenamiento de datos en una base de datos local. En cualquier caso, se evitará el uso duplicado de peticiones al servidor, de modo que la aplicación sea lo más ágil posible de cara al usuario final.

El esquema de infraestructura del proyecto es el siguiente:

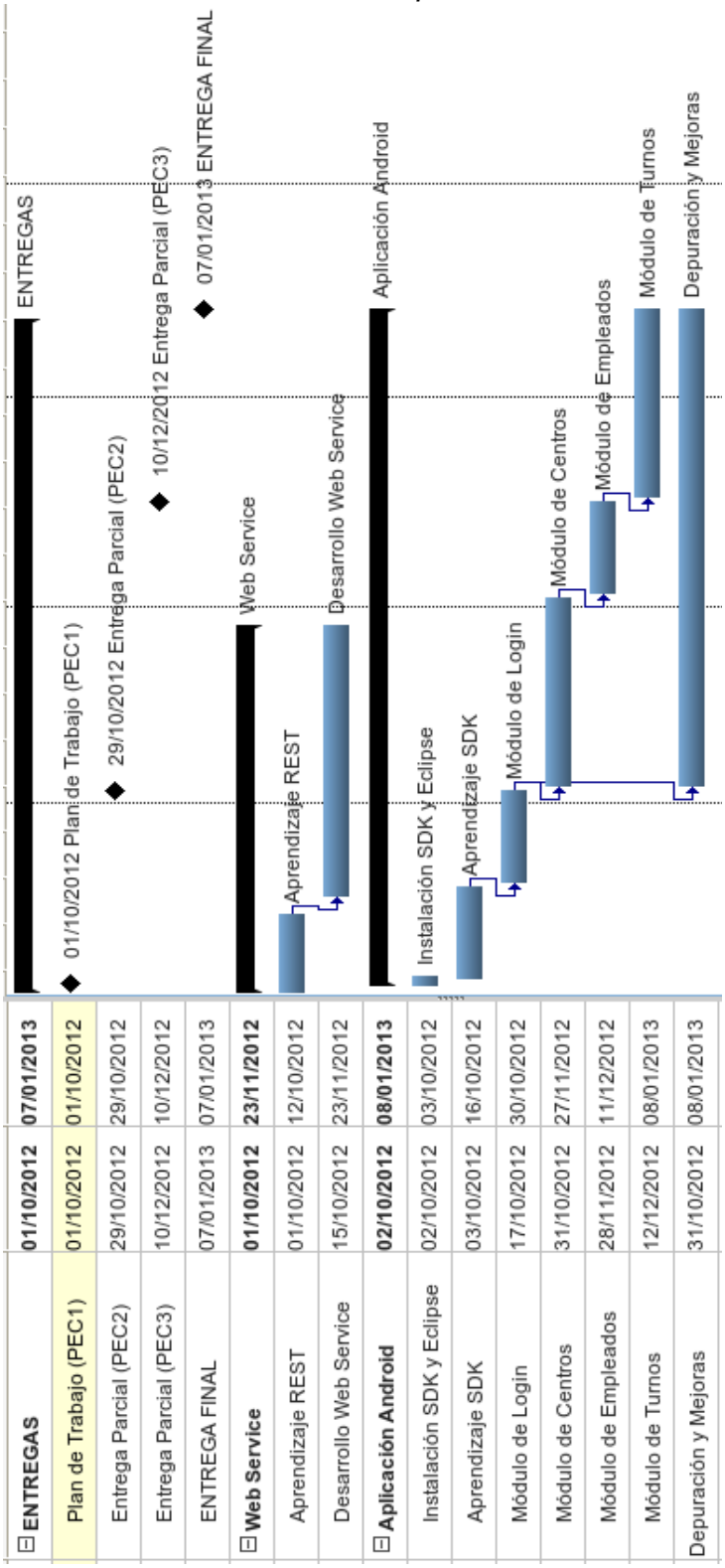
- **Parte servidora:** Web Service desarrollado en PHP, corriendo sobre un servidor Apache 2 y apoyado en un servidor de base de datos MySQL.
- **Parte cliente:** Aplicación nativa Android que se conectará al Web Service anterior para recuperar información y mostrarla en el dispositivo.

Las funcionalidades que debe proporcionar el aplicativo son las siguientes:

- Acceso al sistema con credenciales de usuario válidos
- Listado de centros a controlar y detalle de datos de un centro.
- Listado de empleados y detalle de datos de un empleado.
- Listado y detalle de turnos realizado por empleados en centros.

2. Planificación temporal

Ilustración 1: Planificación temporal



3. Análisis de requisitos

3.1. Autenticación, acceso y seguridad:

El cliente pone especial énfasis en los requisitos de seguridad que debe cumplir la aplicación, mostrando su preocupación por la posible exposición de datos a usuarios no autorizados.

Para solventar este punto sin perjudicar la usabilidad de la aplicación por parte del usuario final, se propone un sistema de autenticación basado en la autenticación por par usuario/contraseña para el inicio de sesión en el servidor. Las credenciales de autenticación se establecerán en la pantalla de opciones de la aplicación, siendo almacenadas en el dispositivo de modo que el usuario no deba introducirlas más que una vez. El nombre de usuario y contraseña de acceso deben ser los mismos que el usuario utiliza para acceder a su entorno privado en la aplicación web TTH.

Una vez que el servidor recibe una solicitud de autenticación con credenciales válidas, generará un token de caducidad limitada que se enviará al dispositivo. A partir de ese momento, todas las consultas al servidor realizadas por el dispositivo deben ir acompañadas de dicho token, que se verificará en el servidor antes de devolver un resultado con datos.

El token tendrá una caducidad de 24 horas y su función principal es evitar el envío del par usuario/clave con cada conexión al servidor. En cualquier caso, el uso del token debe ser totalmente transparente para el usuario final, de modo que, la renovación del token y las peticiones y respuestas de la aplicación móvil sean realizadas en segundo plano, sin perjudicar la experiencia del usuario.

Este planteamiento pretende unificar la forma de trabajo y lógica de la aplicación web ya en producción, ofreciendo al cliente garantías de seguridad suficientes y al usuario de la misma la máxima transparencia a la hora de usarla (cualquier usuario del sistema podrá acceder a sus datos por medio de la aplicación móvil usando sus mismos datos de usuario y clave que ya conoce).

Como mejora para posteriores evoluciones del aplicativo se le propone al cliente la posibilidad de “autorizar” cada dispositivo que vaya a tener acceso a TTH, y vincularlo con una cuenta de usuario concreta. Esto permitiría disminuir el riesgo de intentos de intrusión

desde dispositivos no autorizados y facilitaría la autenticación. De momento, el cliente rechaza esta propuesta para facilitar el uso de cualquier dispositivo por parte de sus usuarios.

Por otro lado, tal y como se exponía inicialmente, este proyecto contempla el desarrollo de una aplicación de “solo lectura”, de modo que el usuario podrá consultar sus datos de la aplicación TTH en su dispositivo móvil, pero NO podrá editar dichos datos. La edición de datos supone un riesgo añadido en el desarrollo de la aplicación que recomendamos vincular al esquema de autenticación propuesto en el párrafo anterior, quedando ambos desarrollos postergados para evoluciones posteriores del producto.

3.2. Disponibilidad de la aplicación móvil:

La aplicación desarrolla estará disponible desde el Android Market (actualmente conocido como Google Play), de modo que cualquier persona con un dispositivo android pueda descargarla. Esta aplicación será gratuita y carente de publicidad, si bien, para su uso con datos reales de TTH será necesario estar registrado como usuario en TTH y disponer del par usuario/clave de acceso al mismo.

Además, desde el propio entorno privado de TTH, cada usuario del sistema podrá descargarse el archivo .apk para la instalación directa de la aplicación en su dispositivo móvil.

3.3. Acceso y uso del Web-Service:

La parte servidora de la aplicación correrá en un subdominio específico para el mismo (android.tamtamhome.com), separado del resto de la aplicación TTH. Este servicio aceptará peticiones no autenticadas, si bien sólo responderá con datos válidos cuando reciba un token correcto, o bien un par usuario/clave válidos.

Este servicio sólo aceptará dos tipos de **peticiones**:

- **De autenticación:** permiten la validación de:
    - Par **usuario/clave**: si es correcto se generará y devolverá un token válido.
    - **Token**: si es correcto se devolverá el mismo token.

- **De datos:** las peticiones siempre deben ir acompañadas de un token válido para mostrar datos. Junto al token se pueden solicitar datos de:
  - **Centros:** listado de centros existentes en TTH para el usuario, junto con sus datos.
  - **Empleados:** listado de empleados existentes en TTH para el usuario, junto con sus datos.
  - **Turnos:** listado de turnos realizados por empleados del usuario en centros del usuario, con horas de entrada y salida y estado de cada turno.

Todas las **respuestas** de la parte servidora serán en formato JSON para minimizar la cantidad de información transmitida por la red. La aplicación móvil será la encargada de procesar dichas respuestas y mostrarlas correctamente en la interfaz de usuario.

3.4. Requisitos de la aplicación móvil:

La aplicación cliente debe ser lo más sencilla posible y visualmente auto-explicativa, escapando de la necesidad de lectura de manuales de uso ni instrucciones complejas por parte del usuario. Toda complejidad que implique la necesidad de formación específica al usuario debe desecharse o bien justificarse adecuadamente.

La terminología a usar en la misma debe ser la misma que se usa en TTH, a la cual los usuarios ya están acostumbrados, evitando el uso de términos nuevos o poco descriptivos que aumenten la complejidad de uso de la misma.

La aplicación cliente dispondrá de las siguientes pantallas:

a) **Pantalla inicial:** será la pantalla que se muestre al arrancar la aplicación. En la misma se realizarán, de modo transparente para el usuario, las tareas de validación de usuario y/o token. Mostrará las opciones disponibles en la aplicación (ver centros, ver empleados, ver turnos y acceder a las opciones de configuración de la aplicación) desde el menú de la aplicación.

b) **Pantalla de configuración:** permitirá guardar el nombre de usuario y clave de acceso a TTH.

c) **Pantalla con datos:** mostrará el listado de datos solicitado (centros, empleados o

turnos). Pulsando en cada elemento del listado se accederá al detalle del elemento pulsado.

Todas las pantallas dispondrán de un menú siempre visible desde el que se podrá acceder a las distintas opciones de la aplicación. El menú debe mostrar el título de la pantalla en la que se encuentra el usuario y los iconos de acceso a cada una de las opciones.

3.5. Peticiones y almacenamiento local de datos:

Con cada arranque de la aplicación se realizará una petición al servidor para validar los datos de acceso (usuario/clave). Si esta autenticación falla no se permitirá el acceso a los datos y se mostrará un mensaje indicando que se debe de configurar las opciones de la aplicación con un par usuario/clave válidos. Desde el propio mensaje se dará acceso directo a la pantalla de configuración.

A parte de la petición de autenticación mencionada, sólo se permite otro tipo de petición (petición de datos), para traer al dispositivo datos de centros, empleados o turnos. La primera vez que el usuario solicite datos de un tipo (por ejemplo, centros), se enviará una petición al servidor solicitando dichos datos. La respuesta (la lista de centros con sus detalles) se almacenará localmente en una variable de aplicación que será válida mientras la aplicación no se cierre. Todas las peticiones posteriores del listado de centros o de detalles de un centro, deben mostrar los datos a partir de los datos guardados localmente, evitando nuevas peticiones al servidor. De este modo, la aplicación ganará en agilidad y se minimizarán las peticiones.

Los datos almacenados en variables locales serán eliminados al cerrar la aplicación, de modo que, para cada uso de la aplicación, se garantiza que el usuario trabaja con un conjunto de datos “frescos”.

Según este esquema, el máximo de peticiones que la aplicación realizará contra el servidor cada vez que se lance será de tres (una por cada tipo de datos), junto con la petición inicial de validación. El resto de operaciones que el usuario realice con la aplicación serán realizada en “local”, sin necesidad de acceso al servidor.

4. Diseño

4.1. Diseño arquitectónico

Parte servidora: servidor Apache2 estándar que ejecutará PHP para atender las peticiones de los clientes apoyado por una base de datos MySQL.

Parte cliente: aplicación nativa Android. Realizará las peticiones contra el servidor por medio de peticiones HTTP recibiendo las respuestas en formato JSON.

4.2. Diseño de datos

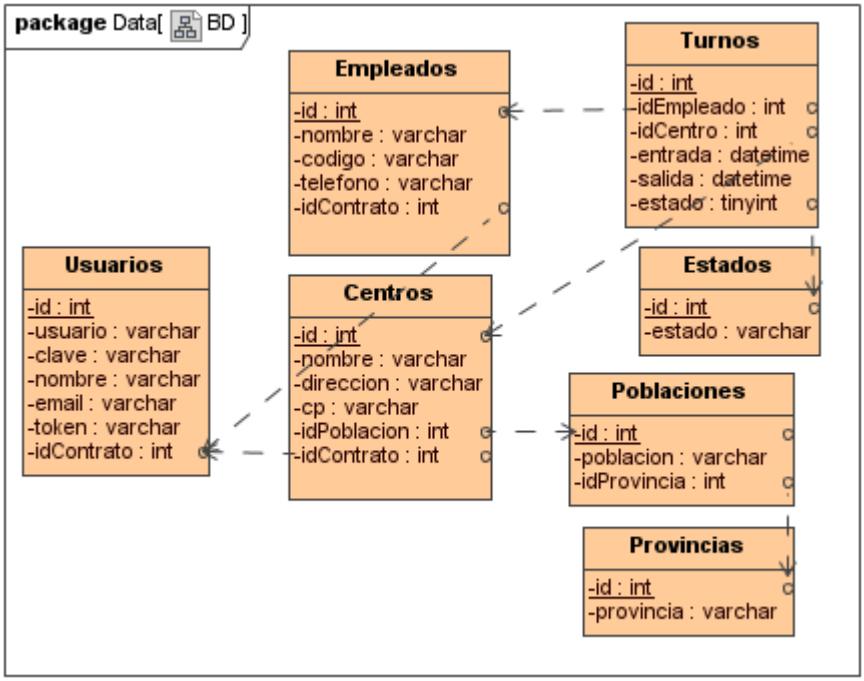


Imagen 2: Modelo de datos

El modelo de datos mostrado es auto-explicativo, y se trata de un conjunto mínimo de datos del modelo sobre el que se apoya la aplicación TTH, contemplando únicamente los datos necesarios para el desarrollo del servicio que recibirá las peticiones de la aplicación móvil.

En la ilustración, figuran subrayados los campos clave del modelo y enlazados con flechas los campos vinculados con otras tablas por integridad referencial.

Junto a esta memoria se entregan las fuentes de la aplicación. En la carpeta “servidor” se incluye un archivo de texto (creacionBBDD.sql) con todas las sentencias SQL que permiten la generación de la base de datos MySQL según esta estructura, así como una serie de datos de ejemplo para poder comprobar el correcto funcionamiento de la misma.

A continuación se presenta la definición de cada una de las tablas del modelo de datos:

Tabla usuarios

Field	Type	Null	Key
id	int(11)	NO	PRI
usuario	varchar(50)	YES	
clave	varchar(50)	YES	
nombre	varchar(50)	YES	UNI
email	varchar(100)	YES	
idContrato	int(11)	YES	MUL
token	varchar(32)	YES	

Tabla estados

Field	Type	Null	Key
id	tinyint(4)	NO	PRI
estado	varchar(11)	YES	

Tabla provincias

Field	Type	Null	Key
id	int(11)	NO	PRI
provincia	varchar(50)	YES	

Tabla poblaciones

Field	Type	Null	Key
id	int(11)	NO	PRI
poblacion	varchar(50)	YES	
idProvincia	int(11)	YES	MUL

Tabla centros

Field	Type	Null	Key

id	int(11)	NO	PRI
nombre	varchar(50)	YES	
direccion	varchar(60)	YES	
cp	varchar(8)	YES	
idPoblacion	int(11)	YES	MUL
idContrato	int(11)	YES	MUL

Tabla empleados

Field	Type	Null	Key
id	int(11)	NO	PRI
nombre	varchar(50)	YES	
codigo	varchar(8)	YES	UNI
telefono	varchar(12)	YES	
idContrato	int(11)	YES	MUL

Tabla turnos

Field	Type	Null	Key
id	int(11)	NO	PRI
idEmpleado	int(11)	YES	MUL
idCentro	int(11)	YES	MUL
entrada	datetime	YES	
salida	datetime	YES	
idEstado	tinyint(4)	YES	MUL

4.3. Diseño de la interfaz de usuario

Se ha intentado que todo el diseño de la interfaz de usuario sea lo más simple y limpio posible, evitando el uso de colores y elementos gráficos superfluos, pero permitiendo que la interfaz sea intuitiva, especialmente para los usuarios de TTH que ya están habituados a la terminología propia del sistema.

La interfaz se compone de un menú de opciones superior en el que figura el nombre de la pantalla en cuestión (o el nombre de la aplicación si todavía no se ha elegido ninguna), junto con cuatro iconos para acceder a los datos de Centros, Empleados, Turnos y Configuración. Este menú se adapta al tamaño de la pantalla del dispositivo móvil utilizado, así como a la rotación de la misma, mostrando tantos iconos como espacio disponible exista. En caso de no poder mostrarse todos los iconos en la barra de menús superior, al resto de las opciones del menú se accederá por medio de la tecla “menú” del dispositivo (si la tuviera) o por medio del control de menú que aparecerá a la derecha de la barra (si el dispositivo no tuviera tecla física de menú).

Las pantallas de listado de datos son simples listas con los nombres de cada elemento, cambiando el título de la barra de menú para mostrar el nombre de la pantalla (Empleados, Centros o Turnos).

Al pulsar en un elemento de la lista se accede a los detalles de dicho elemento. En ese momento, en el título de la barra de menús se mostrará únicamente el nombre del elemento en cuestión (nombre del Centro, nombre del Empleado, etc), pudiendo volver a la lista con la tecla “retroceso” del dispositivo.

A continuación se presentan una serie de capturas de pantalla de la aplicación según el esquema mencionado, mostrando ejemplos de cada una de las pantallas en un dispositivo android estándar.

Pantalla inicial y opciones de configuración:

Imagen 3: Pantalla inicial

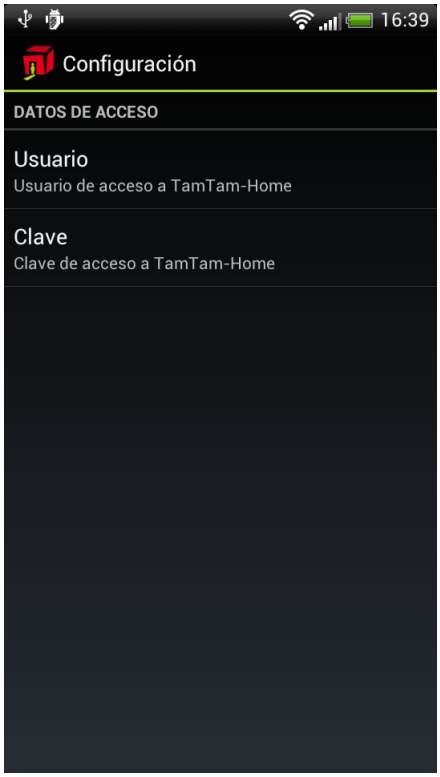


Imagen 4: Configuración

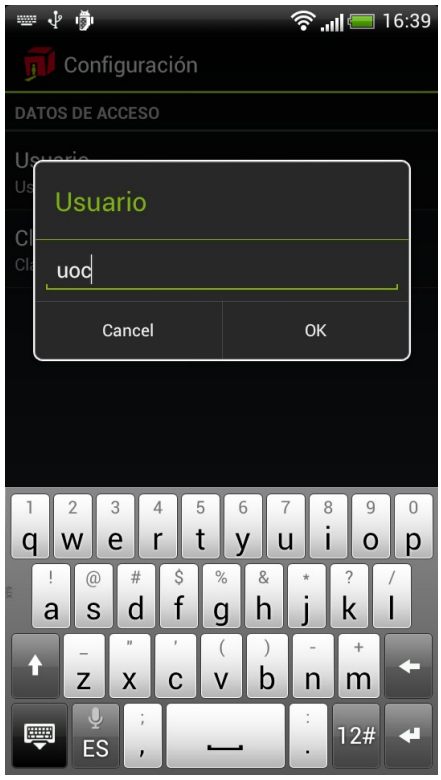


Imagen 5: Usuario y clave

Pantallas de datos:

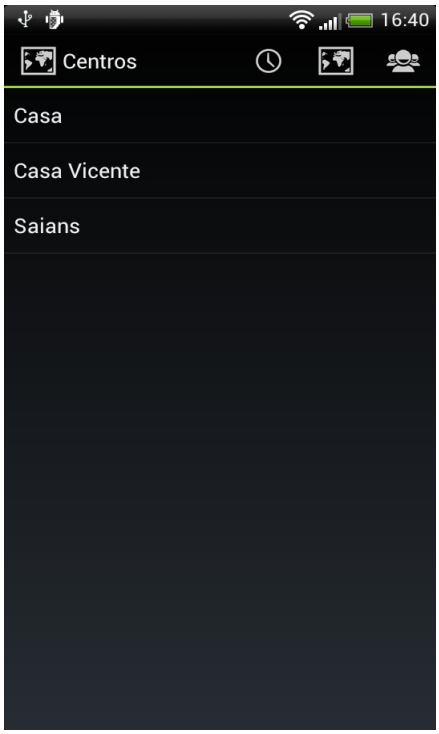


Imagen 6: Listado de centros

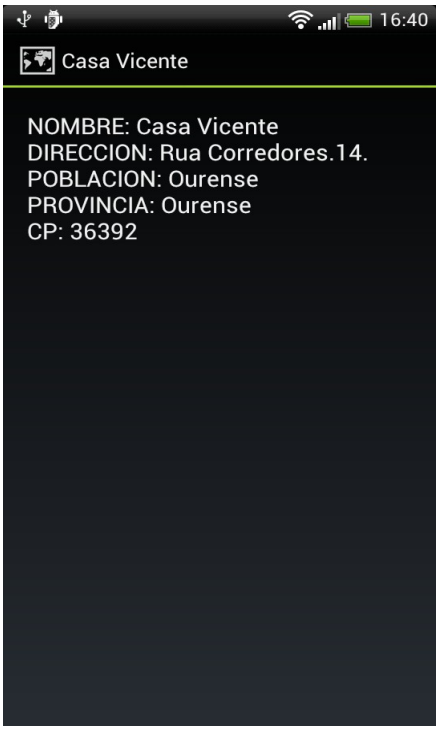


Imagen 7: Detalle de centro

5. Codificación

5.1. Entorno de programación

Para las tareas de programación se utilizó el IDE Eclipse, tanto para la parte servidora (**Eclipse Helios**, para PHP), como para la parte cliente (**Eclipse Indigo**, para java). En este último caso, además del propio IDE, se han instalado los componentes necesarios para el desarrollo: **Android SDK**, el plugin de **Android Development Tools** de Google y el gestor de máquinas virtuales Dalvik.

5.2. Código fuente

Junto a esta memoria se entrega el código fuente completo de la aplicación, tanto de la aplicación cliente como del servicio web. El esquema de directorios entregado es el siguiente:

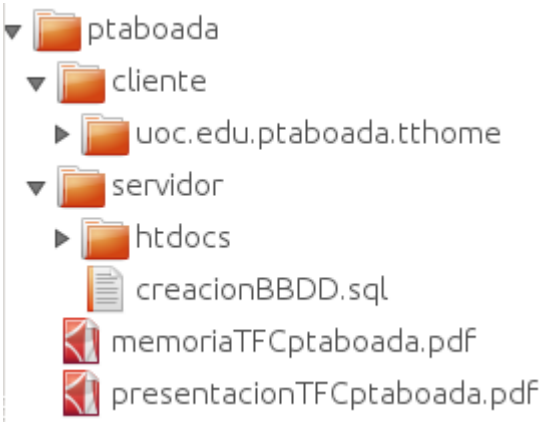


Imagen 8: Directorios del proyecto

+ **`ptaboada:`** directorio raíz del proyecto.
  + **`cliente:`** directorio de la aplicación cliente.
    + **`uoc.edu.ptaboada.tthome:`** raíz del paquete java-android.
  + **`servidor:`** directorio de la parte servidora.
    + **`htdocs:`** raíz de publicación web del servicio web.
    - **`creacionBBDD.sql:`** script de creación de la base de datos.
  - **`memoriaTFCptaboada.pdf:`** memoria del proyecto (este documento).
  - **`presentacionTFCptaboada.pdf:`** presentación de defensa del proyecto.

El conjunto de la aplicación entregada es totalmente funcional pudiendo ser probada directamente utilizando el archivo .apk para instalarlo en un dispositivo Android. Para las pruebas se ha habilitado un usuario con nombre “uoc” y clave “uoc”.

A continuación se comentan los puntos más relevantes y la toma de decisiones adoptadas para cada una de las partes de la aplicación:

5.3. Servidor:

Toda la parte servidora se ha desarrollado en PHP cumpliendo con las funcionalidades expresadas en la toma de requisitos. Básicamente consiste en un servicio web corriendo en un servidor Apache que acepta peticiones HTTP estándar por puerto 80. Para atender cada petición se apoya en un servidor de datos MySQL contra el que realiza las consultas de selección o actualización necesarias para la lógica del servicio. Dicho servidor MySQL puede situarse en la misma máquina que el servidor web o en otra diferente.

Toda la lógica de conexión a la base de datos y consultas de la misma se ha incluido en el archivo **`/inc/datos.php`**, de modo que se centraliza la conectividad de datos para permitir modificar esta capa de la aplicación sin necesidad de tocar el resto de archivos. En este mismo archivo se incluye el manejo de errores de la aplicación y el modo de compatibilidad de JSON (ver párrafo siguiente).

Todas las respuestas del servidor se entregan en formato JSON. Para ello se utiliza la función **`json_encode()`** propia de PHP para versiones de PHP superiores a la 5.2. Para versiones previas se incluye el archivo **`/inc/json.php`**, que se trata de una librería que implementa la misma funcionalidad para compatibilidad con versiones previas de PHP. Esta librería no ha sido desarrollada por mi, tal y como se especifica en la licencia de la misma.

El servicio web sólo admite dos tipos de peticiones: validación y datos. Todas las peticiones reciben los parámetros por GET y entregan las respuestas en formato JSON.

Las **peticiones de validación** se realizan contra el archivo **/valida.php** y permiten validar el token temporal (recibido en el parámetro "**t**") o el par usuario/clave (recibidos en los parámetros "**u**" y "**c**", respectivamente). Todas las respuestas de esta página son en formato JSON con dos únicos campos:

- **res:** resultado de la validación (**OK** si es correcta y **KO** si es incorrecta)
- **val:** valor a devolver (el **token** si es correcta, o el **motivo de error** si es incorrecta)

El algoritmo de la lógica de validación y el paso de mensajes entre cliente y servidor es el siguiente:

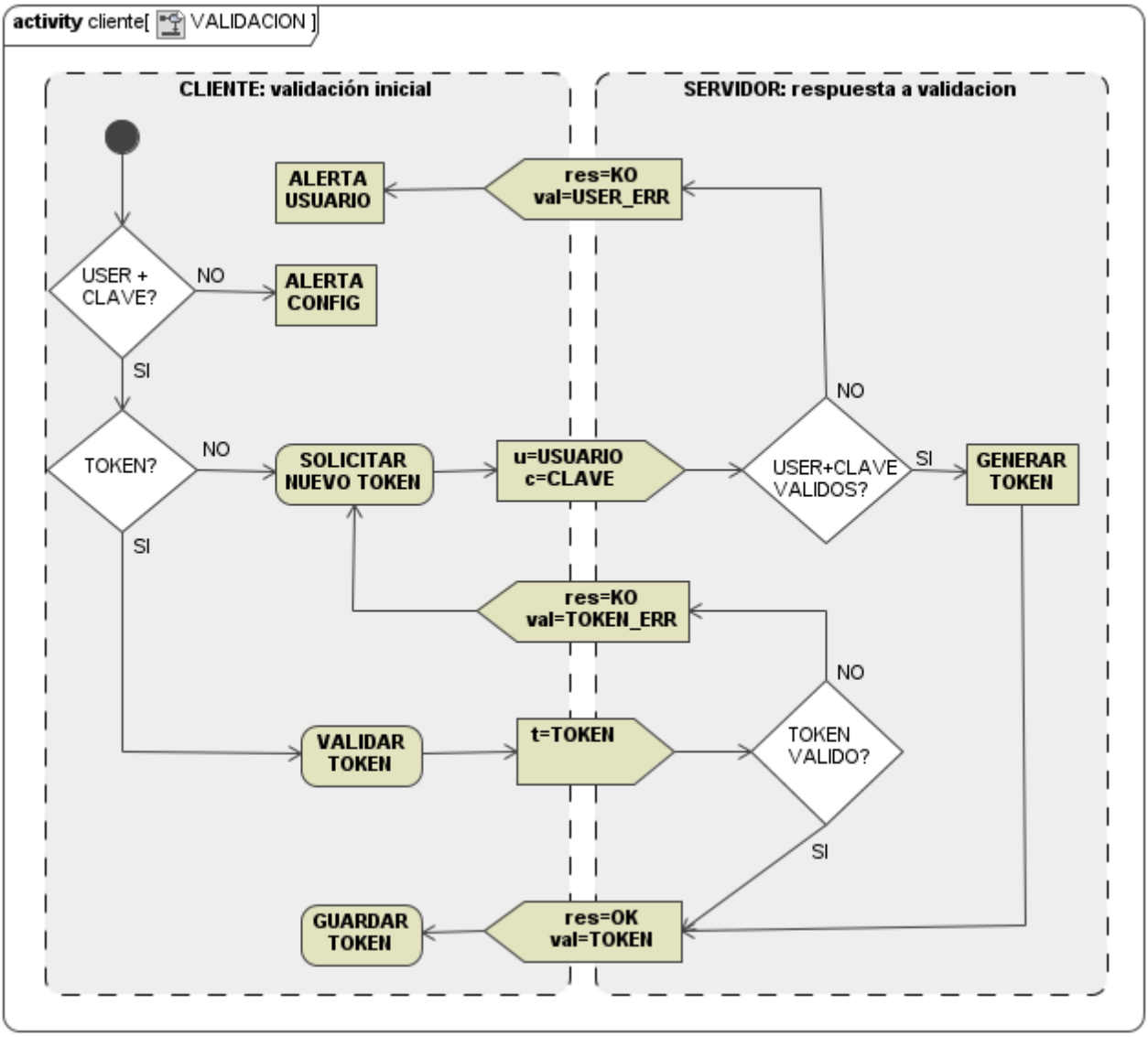


Imagen 9: Algoritmo de validación

Las **peticiones de datos** se realizan contra el archivo **`/index.php`**, y deben contener siempre dos parámetros GET: "**t**" (con el token de sesión) y "**tbl**" (con la inicial de la tabla de datos a consultar: "**c**" para Centros, "**e**" para Empleados o "**t**" para Turnos).

Las repuestas as las peticiones de datos serán un JSON con el conjunto de datos solicitados (si la petición tiene datos), o con dos campos "res/val" como los indicados en el apartado de validación. En este último caso res siempre tendrá el valor "KO" y val el motivo del error o "0" si no hay datos para mostrar.

El algoritmo y paso de mensajes de las peticiones estándar es el siguiente:

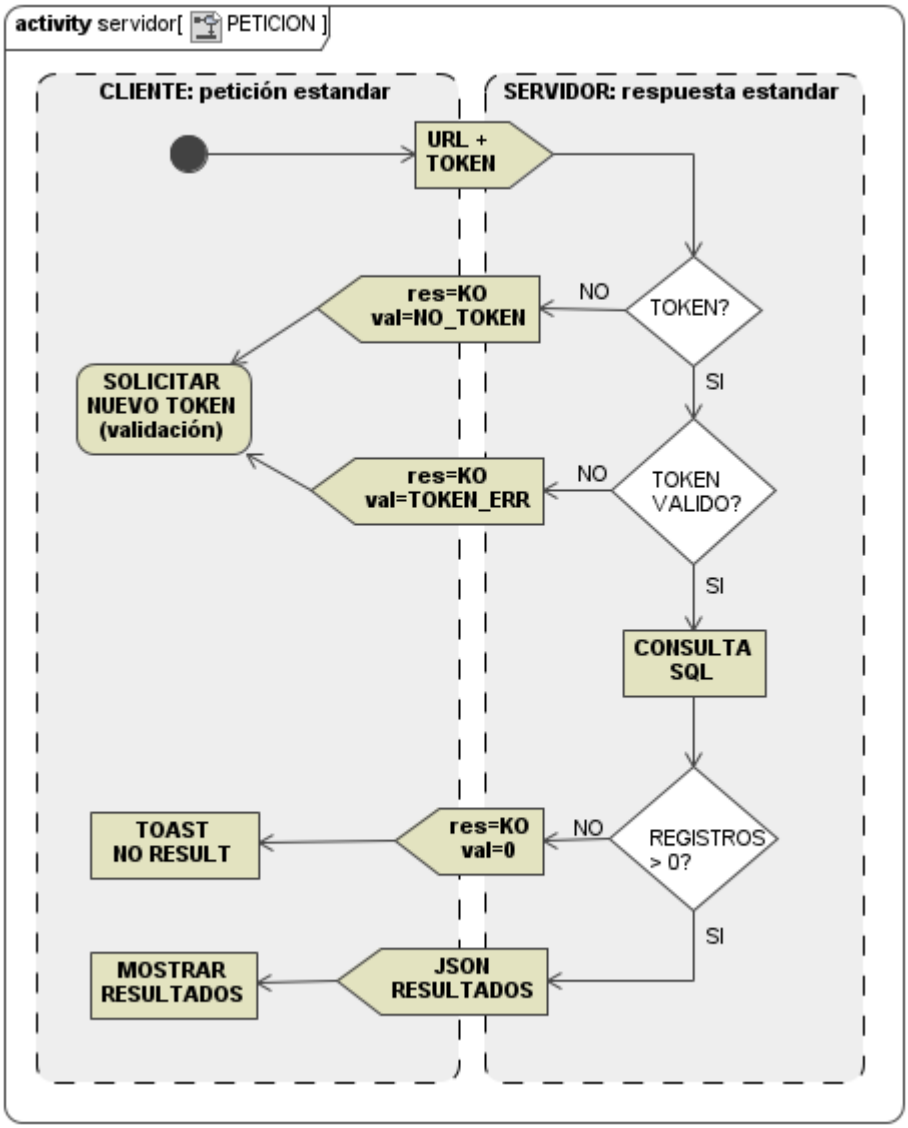


Imagen 10: Algoritmo de petición de datos

5.4. Cliente:

La parte cliente de la aplicación consiste en una aplicación nativa para dispositivos Android. Todo el código de la parte cliente se encuentra en la carpeta del proyecto **`/ptaboada/cliente/uoc.edu.ptaboada.tthome`**, según la estructura estándar de un proyecto android. Los principales directorios a tener en cuenta son los siguientes:

+ `uoc.edu.ptaboada.tthome:` raíz del proyecto android.
  + `res:` directorio de recursos.
    + `drawable:` iconos para la barra de menús.
    + `layout:` archivos .xml de disposición de elementos en las pantallas.
    + `menu:` archivos .xml con las opciones del menú.
    + `values:` archivo .xml con los literales que se mostrarán al usuario.
    + `xml:` archivo .xml con las opciones de configuración de la aplicación.
  + `src:` archivos .java con las clases java del proyecto.
  - `AndroidManifest.xml:` opciones de generación del proyecto.

Todo el código java de las clases se encuentra en la carpeta **`src`** mencionada, y se ha comentado según el estilo javadoc. En cualquier caso, a continuación se comenta cada una de las clases y se resumen los puntos más importantes de las decisiones tomadas:

`Clase TTApplication`

Se trata de una clase que extiende a la clase genérica Application y que permite el almacenamiento de los datos en variables globales que serán accesibles desde cualquiera de las otras clases. En estas variables se almacenarán todos los datos de Centros, Empleados y Turnos tras la primera petición al servidor, de modo que no sea necesario volver a pedirlos de nuevo a lo largo de la vida de la aplicación.

Así mismo, en el arranque de la *Activity* principal, también se almacena en el objeto TTApplication el token para las peticiones al servidor.

Dispone de un método para cada uno de los tipos de datos (Centros, Empleados y Turnos) que devuelven un ArrayList con los nombres para poblar las listas en pantalla.

`Clase Preferences`

Permite almacenar de modo persistente el nombre de usuario y clave introducidos por el usuario en la pantalla de configuración. Al almacenar estos datos de modo persistente se evita que el usuario deba introducir dichos datos cada vez que utilice la

aplicación. Hace uso del manejo de preferencias propio de Android.

Clases Centro, Empleado y Turno

Estas tres clases son una representación de los objetos con los que trabajará la aplicación (centros, empleados y turnos). Su funcionalidad es muy simple y, básicamente, al instanciarse, permiten encontrar los datos de un elemento concreto dentro de las variables globales de la aplicación, y mostrar su detalle en pantalla.

Serán las principales clases a modificar en versiones posteriores de la aplicación, en las que se incluya la edición de datos, mejora de la interfaz de usuario, etc.

Clase TareaHttp

Está orientada a realizar las peticiones HTTP al servidor en un segundo plano, evitando que se bloquee la aplicación en espera de una respuesta por parte del servidor. Al ser creada desde el hilo principal de la aplicación, lanza un segundo hilo de ejecución en el que se realiza la petición HTTP. Al recibir la respuesta del servidor se entregan los datos recibidos y, en el hilo principal se procesan (si la respuesta ha sido correcta) o se toman las decisiones oportunas si la respuesta ha fallado.

Clase HomeActivity

Es la *Activity* principal de la aplicación (la que se muestra al ser lanzada) y sobre la que recae la mayor parte de la lógica de la misma al recibir la interacción del usuario.

Básicamente, al ser creada realiza la validación del token y/o usuario+clave contra el servidor (según el algoritmo mostrado en el apartado *5.3. Servidor*) y prepara el menú de opciones de la aplicación. Cuando el usuario accede a un elemento del menú, por ejemplo “centros”, intenta generar una lista con los nombres de los centros a partir de los datos que ya existen en local. Si no hay datos en local realiza una petición de datos al servidor por medio de la clase TareaHttp y, al recibirlos, genera la lista.

6. Propuestas de mejora y desarrollos posteriores

Tal y como se expuso en la introducción de este proyecto, este desarrollo se plantea como una primera aproximación por parte del cliente para el desarrollo de una aplicación

completa para el servicio TamTam-Home que oferta. Por este motivo, y por la limitación temporal asociada al mismo, se ha intentado reducir al máximo la complejidad del proyecto, dejando para más adelante posibles mejoras o desarrollos posteriores que formarán parte de proyectos que se elaborarán tomando éste como base.

Entre las posibles mejoras y desarrollos a plantear posteriormente se encuentran algunas simples (generalmente comentadas como líneas TODO en el propio código) y otras más complejas. Las principales son:

- Mejora en la representación visual de las pantallas de detalle de Centros y Empleados (actualmente mostradas como texto plano).
- Representación visual de los detalles de un Turno.
- Ampliar la funcionalidad de la aplicación para permitir la edición de datos desde el dispositivo móvil (vinculado a los condicionantes de seguridad mencionados anteriormente).
- Incluir en la aplicación móvil otras funcionalidades de la aplicación completa, especialmente las relacionadas con la herramientas estadísticas.
- Incluir una capa intermedia de almacenamiento de datos en local (usando SQLite) que permita el uso de la aplicación en modo off-line.
- Adaptación del código para permitir la compatibilidad con versiones anteriores de Android (actualmente escrito para SKD 11 o posteriores por el uso del ActionBar).
- Desarrollo de la aplicación móvil para dispositivos iOS.

7. Bibliografía

Libros:

- MEIER, RETO. 2012. **Professional Android 4 Application Development**. Wrox Professional Guides.
- DIMARZIO, JEROME. 2008. **Android, A Programmer's Guide**. Mc Graw Hill.

Webs:

- **Android SDK Development Site:** http://developer.android.com/sdk/index.html
- **Android Development Tutorial:** http://www.vogella.com/articles/Android/article.html
- **REST:** http://blog.dhananjaynene.com/2009/06/why-rest/
- **Android HTTP GET JSON:** http://www.accella.net/android-http-get-json/
- **Android Code Style Guidelines:** http://source.android.com/source/code-style.html
- **How to Write Doc Comments for the Javadoc Tool :** http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html