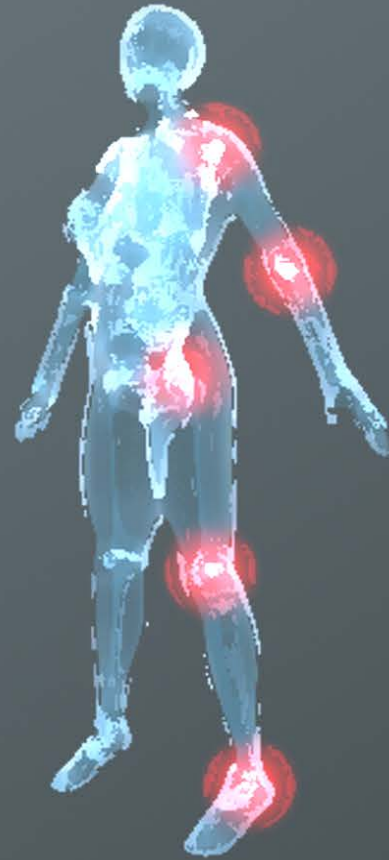




PROYECTO FINAL DE CARRERA

INGENIERIA INFORMÁTICA



Gestor Móvil de Hábitos y Rutinas en el Tratamiento del Dolor Crónico

Tutor
Roman Roset Mayals.

Estudiante
Raúl Gil Sarmiento

9 enero 2012

*A Chari, mi esposa,
y a mis hijos: Adrián, Víctor y Miriam.*

A ellos pertenecía el tiempo que he dedicado a estos estudios.

RESUMEN

Estudios recientes señalan que aproximadamente un 19% de la población europea sufre de dolor crónico, entendido como aquél que perdura más de tres meses.

Los tratamientos convencionales para estas dolencias se han demostrado mejorables mediante el uso de terapias cognitivo-conductuales. Esto es: el conocimiento por parte del paciente de todos los aspectos de su dolencia y el seguimiento de una serie de rutinas o tareas, mejora su calidad de vida. El inconveniente de estas terapias es que suelen exigir gran cantidad de recursos públicos, siempre escasos.

En este escenario, la tecnología móvil tiene mucho que aportar. Fibroline es un proyecto multidisciplinar recién iniciado, cuyo objetivo es desarrollar una aplicación para dispositivos móviles capaz de cubrir buena parte de las necesidades de apoyo, seguimiento y asistencia que tiene el paciente, haciéndolo además desde la comodidad e intimidad del hogar y evitando así en la medida de lo posible desplazamientos

del paciente o molestas irrupciones en su vida cotidiana.

El presente proyecto de final de carrera desarrolla un módulo fundamental de Fibroline: El Gestor de los Hábitos y Rutinas que el paciente debe ir incorporando a su vida durante su tratamiento.

Este módulo plantea diariamente al paciente una lista de tareas a realizar, dándole indicaciones de realización. La aplicación mantiene un registro de las tareas planteadas cada día, cuáles se realizaron y cuáles no, así como los motivos de la su no realización. Este registro sirve como feedback al equipo terapéutico y a la propia aplicación para determinar las tareas que se plantearán el día siguiente.

Aunque la aplicación funciona la mayor parte de tiempo en modo local, su fundamento web permite que el equipo médico multidisciplinar acceda remotamente a la aplicación para hacer un seguimiento del tratamiento e incluso para modificarlo.



TERMINOS CLAVE

- *Dolor Crónico*
- *Hábitos y Rutinas*
- *Asistente personal*
- *Fibroline*
- *Aplicación móvil*
- *Dispositivos móviles*
- *Tele-tratamiento*
- *Aplicación HTML5-JavaScript*
- *WAC*

ÁREA DE PROYECTO

Desarrollo de Aplicaciones Móviles mediante WAC

INDICE DE CONTENIDOS

<i>Portada</i>	0
<i>Dedicatoria</i>	1
<i>Resumen</i>	2
<i>Términos clave</i>	3
<i>Área de proyecto</i>	3
<i>Índice</i>	4
<i>Capítulo 1: Ideas Precursoras</i>	6
1. <i>Primera Idea</i>	6
2. <i>Segunda Idea</i>	9
<i>Capítulo 2: Idea Definitiva</i>	12
1. <i>Presentación de la Idea</i>	13
2. <i>Título y Objetivo del Proyecto</i>	14
3. <i>Productos a entregar</i>	14
4. <i>Motivaciones Personales</i>	15
5. <i>En qué consiste el Proyecto General Fibroline</i>	15
6. <i>Módulos funcionales que componen Fibroline</i>	16
7. <i>Historias de Usuario (requisitos) del Módulo de Gestión de Hábitos</i>	18
8. <i>Justificación Científica del Módulo de Gestión de Hábitos</i>	19
9. <i>Funcionamiento General del Módulo de Gestión de Hábitos</i>	19
10. <i>Planificación</i>	20
<i>Capítulo 3: Organización y Opciones</i>	21
1. <i>Equipo de trabajo</i>	22
2. <i>Medios de Comunicación y Colaboración</i>	23
3. <i>Metodología General del Proyecto. Kanban</i>	24
4. <i>Programación Colaborativa. GitHub</i>	25
5. <i>Control de Versiones. Git</i>	26
6. <i>Entorno de Desarrollo</i>	28
7. <i>Frameworks, Librerías y otras Tecnologías</i>	29
8. <i>Información detallada de las diferentes Tecnologías</i>	30
a) <i>HTML, CSS y JavaScript</i>	30
b) <i>Wholesale Application Community</i>	34
c) <i>PhoneGap</i>	35

d) Sencha Touch	36
e) NodeJS.....	37
Capítulo 4: Utilización del módulo de tareas.....	38
1. ¿Qué es una tarea?.....	39
2. Acceso al Módulo Gestor de Hábitos y Rutinas (Módulo de Tareas).....	40
3. Vista de Tareas Propuestas para Hoy.....	40
4. Vista de Detalles de una Tarea.....	42
5. Vista de Encuesta sobre las Tareas No Realizadas el Día Anterior.....	42
Capítulo 5: Funcionamiento Interno.....	44
1. ¿Qué función tiene el Módulo Gestor de Hábitos?.....	45
2. ¿Qué propiedades posee cada tarea?.....	45
3. ¿Dónde se almacenan las tareas?.....	47
4. Modelo-Vista-Controlador.....	47
5. Componentes de Módulo Gestor de Hábitos.....	48
6. Controlador: TaskController.js.....	49
7. Modelo: TaskModel.js y TaskStore.js.....	50
8. ¿Cómo se determinan las tareas a realizar hoy?.....	51
9. Control de la Vista de Tareas para Hoy.....	52
10. Control de la Vista de Detalles de una Tarea.....	52
11. Control de la Vista de Tareas No Realizadas el Día Anterior.....	53
Capítulo 6: Conclusiones y Futuro.....	54
1. Autoevaluación y conclusiones.....	54
2. Líneas de futuro.....	55
Anexo I: Modelo Avanzado de Tareas.....	56
1. Introducción.....	56
2. Atributos de una tarea.....	57
3. Atributos de un tipo de tarea.....	58
Ejemplos de tareas.....	59
Anexo II: Instalación del Entorno – Resumen de Instrucciones.....	64
Anexo III: Historias de Usuario. Versión completa.....	66
Bibliografía.....	72

Capítulo 1:

Ideas precursoras

Antes de llegar a la idea definitiva, aparecieron otras dos, precursoras, que merecen ser mencionadas en esta memoria. A ellas se dedica este capítulo.

PRIMERA IDEA

TÍTULO

Aplicación móvil para la comunicación escuela-familia.

INTRODUCCIÓN

Nadie discute la importancia de una comunicación fluida entre la escuela y la familia en las etapas de Educación Primaria y Secundaria. Familia y Escuela deben actuar al unísono sobre el alumno para obtener una deseada sinergia de esfuerzos. Es especialmente importante que los padres estén al tanto en todo momento de la marcha escolar de sus hijos, posibles incidencias acontecidas, tareas para casa, próximos exámenes, calificaciones obtenidas en estos, etc.

Sin embargo, con frecuencia no existen suficientes oportunidades de comunicación padres-docentes para

lograr este objetivo. Por su propia inmadurez y posición interesada, el alumno no siempre es el mejor vehículo de comunicación.

En este contexto, surge la idea de desarrollar una herramienta que facilite la comunicación entre escuela y familia.

Bien podría emplearse para ello una aplicación web convencional que fuera accedida desde un PC. Sin embargo, las ocupaciones laborales y de toda índole de los progenitores los mantienen con frecuencia alejados de los ordenadores, acentuado por una no siempre fluida relación con estos equipos.

Por el contrario, en la inmensa mayoría de los hogares actuales, al menos uno de los progenitores lleva siempre con él un móvil. Además, en poco tiempo todos los móviles y tarifas ofrecerán conexión permanente a internet.

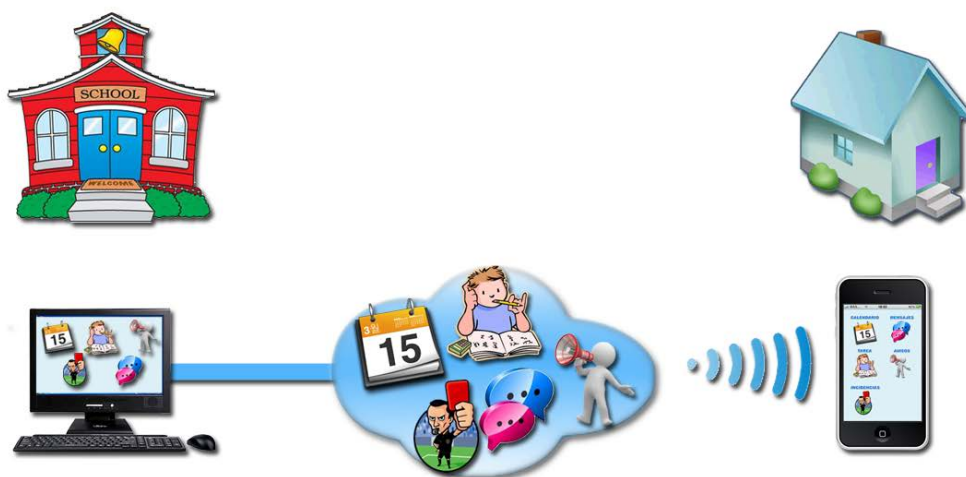
Aprovechando esta circunstancia se plantea la idoneidad de desarrollar la herramienta web para su acceso desde un terminal móvil, si bien puede ser perfectamente compatible con el acceso desde un PC.

APORTACION DEL PROYECTO - OBJETIVO

Este proyecto pretende desarrollar una aplicación web, accesible desde dispositivos móviles y posiblemente

también desde PCs, que facilite la comunicación entre la escuela y la familia.

ESQUEMA GENERAL DE FUNCIONAMIENTO



La información objeto de la comunicación, así como la aplicación web, estarán alojadas en Internet.

La escuela será responsable de generar la mayor parte de la información, empleando generalmente un PC para subir ésta a Internet valiéndose de una de las interfaces de la aplicación web, programada a tal efecto.

La familia actuará principalmente como receptor de la información subida a Internet por la escuela, accediendo a ella a través de la interfaz móvil de la aplicación.

No obstante lo anterior, parte de la información fluirá en sentido inverso, es decir, de las familias a la escuela, he incluso entre familias.

Dado el área concreta del proyecto, se dará prioridad al desarrollo del núcleo de la aplicación web y de su interfaz móvil. Si hubiera tiempo suficiente se desarrollaría también la interface de acceso desde PC; en caso contrario, se prescindiría de ella y la información sería introducida de modo manual en el servidor.

POSIBLES CATEGORÍAS DE INFORMACIÓN

Las categorías de información no están totalmente cerradas. Aquí se exponen algunas de las posibles.

Durante el desarrollo del proyecto se irán implementando una a una, pudiendo surgir alguna nueva. Se implementarán todas aquellas que los plazos existentes permitan, no necesariamente todas.

Siempre será posible ampliar funcionalidad en futuras versiones del software.

CALENDARIO:

Recoge eventos y fechas relevantes, como exámenes, fiestas, plazos de entrega de trabajo, reuniones, etc. Las genera la escuela, aunque puede plantearse la posibilidad de que la familia cree sus propios ítems para sí mismo.

Hay que plantearse si la información quedará almacenada en el calendario nativo del propio dispositivo, accesible por tanto desde la aplicación integrada de fábrica, o por el contrario quedará almacenada en el servidor web, pudiéndose acceder sólo desde esta aplicación.

TAREA:

La escuela puede mantener actualizada las tareas diarias que los alumnos deben realizar en sus casas. Se planteará la posibilidad de que los padres puedan marcar los ítems que los alumnos van realizando.



AVISOS:

Información de cualquier tipo que la escuela considere oportuno comunicar de modo general a las familias, ordenadas por fecha de inserción y/o por categoría.

MENSAJES:

Sistema de mensajería off-line, que permita la comunicación bidireccional escuela-familia, e incluso entre familias.

INCIDENCIAS:

Recogería todas las incidencias relevantes acontecidas durante la jornada escolar, relacionadas con cada alumno, como ausencias, amonestaciones, etc.

SEGUNDA IDEA

(Adaptación de la primera)

JUSTIFICACIÓN DE LA ADAPTACIÓN

La cercanía profesional del alumno al ámbito de las enseñanzas medias le llevó a plantear como escenario del proyecto la escuela. Proponía cubrir la necesidad de mayor comunicación entre la escuela y la familia.

El tutor, además de compartir esa cercanía con el mundo de la educación, trabaja como investigador-colaborador en áreas relacionadas con el dolor crónico, especialmente en la población infantil.

El tutor plantea la posibilidad de ampliar el ámbito del proyecto a este escenario terapéutico. Dada la escasez de aplicaciones en este preocupante

ámbito, y el parecido técnico con la primera idea, el alumno acepta la propuesta del tutor.

Podría comenzarse realizando una aplicación móvil con funcionalidades útiles tanto en uno como en otro ámbito para, posteriormente mediante refactorización, elaborar una librería o un framework que permita desarrollar aplicaciones móviles en cualquier escenario similar donde haya que mantener una colaboración a distancia entre un ente proveedor de servicios y sus 'clientes' o beneficiarios (Escuela-Familia-Alumno, Hospital-Familia-Paciente, etc.)

TÍTULO

Tele-Asistencia integral a niños con dolor crónico.

INTRODUCCIÓN

Son varios los escenarios en los que los dispositivos móviles pueden cubrir necesidades de comunicación y colaboración a distancia entre un ente proveedor de servicios o asistencia y sus clientes o beneficiarios.

Una de ellas es la tele-asistencia terapéutica de pacientes. Muchos de estos pacientes necesitan incorporar unas rutinas diarias para superar y/o sobrellevar la enfermedad con la mejor calidad de vida posible. Los pacientes no son totalmente autónomos en este proceso, sino que requieren

asesoramiento, apoyo y seguimiento frecuente por parte diversos profesionales.

Sin embargo, la escasez de recursos públicos suele imposibilitar un seguimiento y una asistencia presencial tan frecuente como se desearía.

En este escenario, la tecnología móvil tiene mucho que aportar. Las aplicaciones sobre dispositivos móviles no sólo pueden cubrir parte de esas necesidades de seguimiento y asistencia que tiene el paciente, sino que lo hacen desde la comodidad e intimidad del

hogar, evitando en la medida de lo posible desplazamientos del paciente o molestas irrupciones en su vida cotidiana.

Un caso particular es el relacionado con niños que padecen de dolor crónico. Así

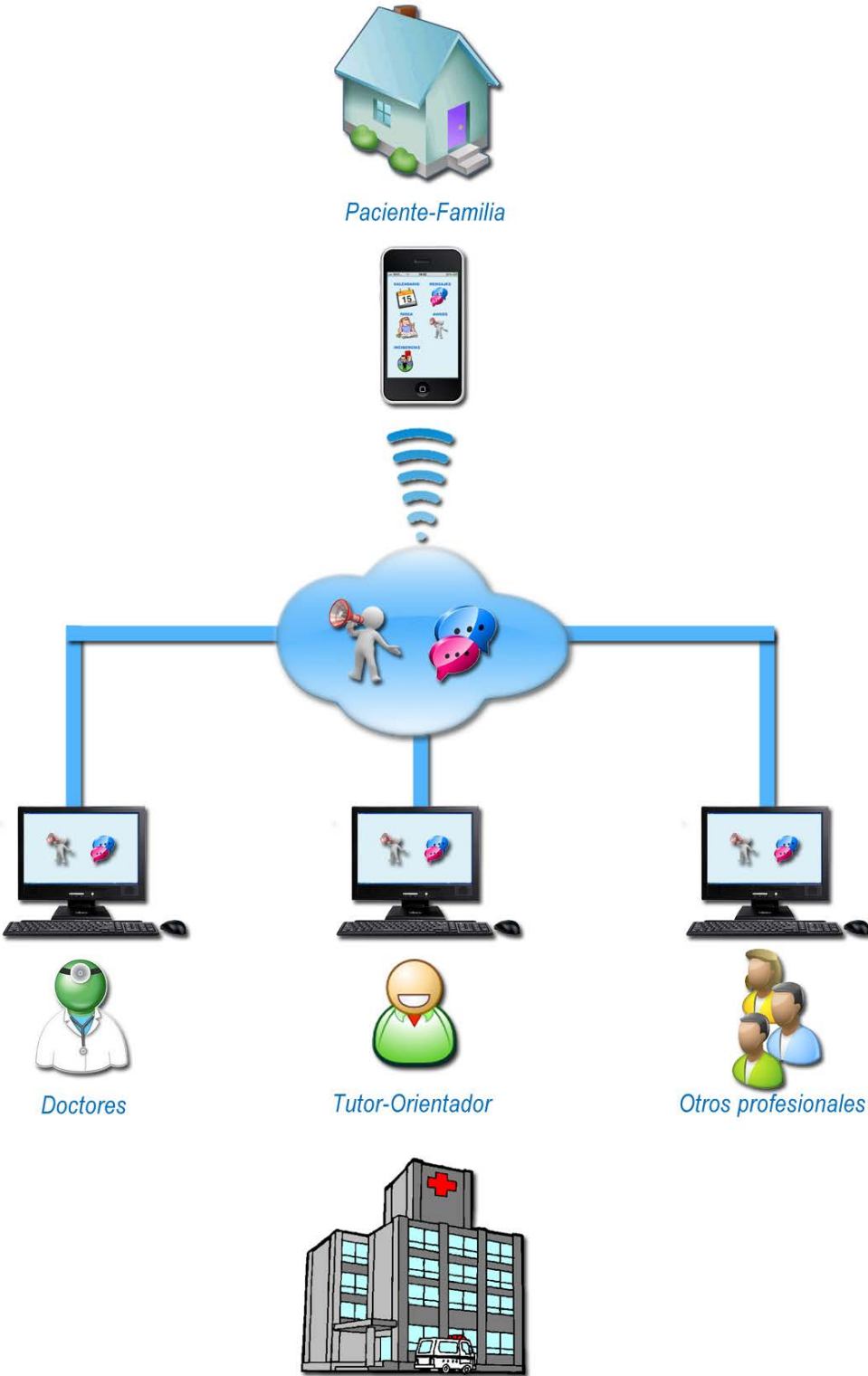
como existen abundantes protocolos de actuación, equipos y medios para tratar el dolor en la población adulta, no sucede lo mismo con la infantil. Por tanto, en este contexto resulta especialmente útil el proyecto planteado.

APORTACION DEL PROYECTO - OBJETIVO

Este proyecto pretende desarrollar una aplicación para dispositivos móviles que facilite el seguimiento y la asistencia a distancia de pacientes de dolor crónico, niños más concretamente.

La orientación del proyecto, sin embargo, apunta a que esta aplicación sea la base de un futuro framework o una librería que permita el rápido desarrollo de aplicaciones de tele-asistencia en ámbitos similares.

ESQUEMA GENERAL DE FUNCIONAMIENTO



Capítulo 2: Idea definitiva

INDICE DEL CAPÍTULO

1. Presentación de la Idea.
2. Título y Objetivo del Proyecto.
3. Productos a entregar.
4. Motivaciones Personales.
5. En qué consiste el Proyecto General Fibroline.
6. Módulos funcionales que componen Fibroline.
7. Historias de Usuario (requisitos) del Módulo de Gestión de Hábitos
8. Justificación Científica del Módulo de Gestión de Hábitos.
9. Funcionamiento General del Módulo de Gestión de Hábitos.
10. Planificación.

1. PRESENTACIÓN DE LA IDEA

En el capítulo 1 se han presentado las dos ideas precursoras:

- Aplicación web, accesible desde dispositivos móviles y posiblemente también desde PCs, que facilite la comunicación entre escuela y familia.



- Aplicación para dispositivos móviles que facilite el seguimiento y la asistencia a distancia de pacientes de dolor crónico, niños más concretamente, pero con vistas a generalizarla en última instancia para convertirla en un framework que permita el rápido desarrollo de aplicaciones de teleasistencia en ámbitos similares.

Lo delimitado del calendario de elaboración de este PFC, hizo que de los dos escenarios planteados (escolar y médico) en las primeras ideas para el proyecto, el alumno se decantara por

uno de ellos, el terapéutico. Descartando de momento un enfoque general que pretenda dar cobertura a ambos escenarios.

El tutor planteó una de las siguientes opciones.

- Que el estudiante realizara un proyecto autónomo y totalmente independiente.
- Que el estudiante inscribiera su proyecto dentro de otro más amplio, llamado Fibroline, que se ha comenzado a realizar en el marco de una tesis doctoral.

Visto lo interesante del proyecto Fibroline, el estudiante decide inscribir su proyecto en él. El título completo del Proyecto General en el que se engloba este PFC es:

“Tratamiento contra la fibromialgia mediante tecnología móvil”

Por otro lado, ante la posibilidad de trabajar individual, autónoma y o hacerlo colaborativamente con el resto del equipo de Fibroline, el alumno opta por lo segundo.

El estudiante se centrará en uno de los módulos que componen Fibroline; en concreto, en el ‘Módulo para incorporar hábitos y rutinas’, al que denominamos vulgarmente ‘Módulo de Tareas’.

2. TÍTULO Y OBJETIVO DEL PROYECTO

TÍTULO DEL PROYECTO DEL ESTUDIANTE (PFC)

Gestor móvil de hábitos y rutinas en el tratamiento del dolor crónico.

OBJETIVO DEL PROYECTO DEL ESTUDIANTE (PFC)

Desarrollar un componente software – como parte de una aplicación para dispositivos móviles - que sirva de asistente personal de pacientes de fibromialgia en la gestión y realización de las tareas establecidas en su tratamiento para incorporar hábitos y rutinas que mejoren su calidad de vida.

3. PRODUCTOS A ENTREGAR

Al finalizar el presente proyecto, el estudiante deberá entregar los siguientes productos:

- *Una aplicación para dispositivos móviles que haciendo uso de las tecnologías y estándares web cumpla con objetivos y requisitos establecidos.*
- *Una memoria de la realización del proyecto.*
- *Una presentación que exponga visual y amenamente los aspectos relevantes del proyecto.*

4. MOTIVACIONES PERSONALES

Las motivaciones personales que llevan al estudiante a escoger este proyecto son:

- Se trata de un proyecto real, no sólo académico, que puede beneficiar a un colectivo merecedor de mayor dedicación, los pacientes de dolor crónico, especialmente población infantil y juvenil.
- Representa una gran oportunidad para aprender a trabajar en equipo.
- Es un área actual que fuerza el aprendizaje de tecnologías novedosas para el estudiante y poco exploradas durante el resto de la carrera.

5. EN QUÉ CONSISTE EL PROYECTO GENERAL FIBROLINE

Fibroline tiene como objetivo desarrollar una aplicación móvil que sirva como asistente personal en el tratamiento de los pacientes de fibromialgia; enfermedad que conlleva dolor crónico y limita el desempeño y con frecuencia la calidad de vida de los pacientes.

Como ya se ha justificado, las aplicaciones sobre dispositivos móviles no sólo pueden cubrir parte de las necesidades de seguimiento y asistencia que tiene el paciente, sino que lo hacen desde la comodidad e intimidad del hogar, evitando en la medida de lo posible desplazamientos del paciente o molestas irrupciones en su vida cotidiana.

Esta tecnología es aplicable especialmente en la población infantil, adolescente y juvenil.

A los pacientes se les proporcionará un terminal móvil con el software Fibroline preinstalado y una versión inicial de su tratamiento precargada en la aplicación.



A partir de ese momento, el paciente interactúa en modo local (sin necesidad de conexión a internet) con la aplicación que le asiste en su tratamiento y esta va registrando toda la información relevante.

Eventualmente, el equipo médico interactuará con la aplicación del paciente para examinar el desarrollo del tratamiento y si fuera necesario reajustarlo.

6. MÓDULOS FUNCIONALES QUE COMPONEN FIBROLINE

Fibroline constará de varios módulos claramente diferenciables por el usuario, cada uno de los cuales ofrecerá una funcionalidad:

MÓDULO DE EVALUACIÓN

El módulo de evaluación proporciona al terapeuta y al usuario instrumentos de evaluación sobre las siguientes variables:

- Intensidad de dolor.
- Nivel de ansiedad.
- Calidad de sueño.
- Nivel de satisfacción con la actividad realizada.
- Motivos por los que no se realiza un tipo de tarea y su frecuencia.
- Cumplimiento de tareas asignadas.
- Comportamiento del usuario ante la aplicación.

MÓDULO DE HÁBITOS DE VIDA (a desarrollar por el estudiante)

El módulo de hábitos de vida se ocupa de gestionar aquellas acciones relacionadas con las actividades de la vida diaria del usuario. Dentro de este módulo se especifican los siguientes temas:

- Agenda de acciones diarias por realizar.
- El seguimiento de dichas acciones.
- Los recordatorios para realizar las acciones.

MÓDULO DE PSICOEDUCACIÓN

Está muy relacionado con el de hábitos de vida.

Gestiona las acciones de la aplicación relacionadas con las lecciones y la adquisición de competencias a conseguir una vez estudiadas las lecciones. Algunos de los temas dentro de la aplicación que administra este módulo son los siguientes:

- *Administración de las diferentes lecciones.*
- *Visualización de una lección*
- *En el futuro evaluar la adquisición de competencias de la psico-educación.*

MÓDULO DE MOTIVACIÓN

El módulo de motivación se encarga de gestionar las acciones pertinentes para aumentar la motivación del usuario para lograr alcanzar los objetivos.

MÓDULO DE INTÉRPRETE PARA LA PROGRAMACIÓN DEL TRATAMIENTO

El módulo es el responsable de interpretar un TCC expresado en términos de lenguaje artificial, para incorporar las acciones y los contenidos en el sistema. También se responsabilizará de gestionar los sistemas sobre la planificación del tratamiento y analizará el flujo del tratamiento.

MÓDULO DE MONITORIZACIÓN

El módulo de monitorización es el responsable de administrar las funcionalidades necesarias para registrar todas las acciones del usuario ante la aplicación. Así por ejemplo alguna de las acciones que debe registrar son las siguientes:

- *Pulsación de botones,*
- *Tiempo por página*
- *Etc.*

Además también recogerá aquellas acciones que son evaluadas, para que posteriormente el módulo pertinente las analice y utilice el análisis de los datos de forma perteneciente.

MÓDULO DE VISUALIZACIÓN DE DATOS

El módulo de visualización de datos es el responsable de organizar la información visualmente para que los diferentes roles del sistema puedan, a través de la visualización, cumplir determinados objetivos que van más allá de la lectura de los datos. Así por ejemplo, si uno de los objetivos en la visualización de los datos al paciente es no sólo informarle sobre sus progresos sino también motivarle, el módulo de visualización ofrecerá la información de la forma más apropiada para contribuir a lograr dicho objetivo.

7. HISTORIAS DE USUARIOS (REQUERIMIENTOS) DEL MÓD.GESTOR DE HÁBITOS

Las Historias de Usuario resumen los requerimientos o necesidades de los usuario expresados en primera persona (aunque sean redactados por el equipo de desarrollo).

Durante la planificación suelen ser desglosadas en tareas de pequeño tamaño, que son las unidades fundamentales de planificación y trabajo.

Estas tareas van siendo colocadas en el punto de partida del tablero Kanban (ver punto 3 del capítulo 3) y elaboradas una a una. Conforme estas tareas se van desarrollando, añadiendo valor al producto, avanzan por el tablero Kanban hasta ser completadas, momento en el cual salen por la derecha de tablero dando como resultado un subproducto concreto y verificable.

A continuación se exponen resumidamente las historias de usuario relacionadas con el Módulo Gestor de Hábitos y Rutinas que atenderá el estudiante. El Anexo II recoge la versión completa del documento de Historias de Usuario, en el que aparecen todos los detalles y las HH.U. del proyecto general Fibroline.

Título	Descripción
Carga de la lista de tareas	COMO terapeuta QUIERO que el sistema tenga cargada una lista de tareas, donde las tareas tienen una descripción, pertenecen a una categoría, pueden llevar una acción de usuario asociada y se inician a partir de (o bien del inicio del tratamiento o bien que una tarea haya finalizado) agregándole un número de días (que pueden ser 0), además pueden ser repetitivas o de una sola realización, y si son repetitivas tienen los días de la semana en que se van a producir (lunes, martes y viernes por ejemplo), y pueden ser de duración fijada en una fecha o para siempre PARA que se visualicen en la lista de tareas "para hoy" cuando toquen y se pueda trabajar con ellas
Ver tareas diarias	COMO usuario QUIERO ver la lista de tareas que debo realizar hoy y que se puedan distinguir aquellas que ya he completado de aquellas que me quedan pendientes, y además distinguir aquellas que se marcan como completadas de forma automática cuando realice una acción en el sistema, de aquellas que debo marcar como completadas de forma manual PARA poder completarlas cuando quiera a lo largo del día.
Marcar manualmente la tarea realizada	COMO usuario QUIERO marcar una tarea no monitorizada como completada cuando quiera a lo largo del día PARA avisar al sistema que la he completado y ser consciente de lo que he hecho y tengo pendiente
Marcar automáticamente la tarea realizada	COMO usuario QUIERO que el sistema marque por mi aquellas tareas completadas, que se realizan a través del propio sistema PARA que quede constancia de ello
Información matinal	COMO usuario QUIERO que cada día la primera vez que acceda al ToDo se me presente un cuestionario sobre las actividades del día anterior, donde contestaré por cada tarea todos aquellos datos que me haya dejado del día anterior
Explicación de incumplimiento	COMO usuario QUIERO poder elegir entre varias categorías el motivo por el cual no he cumplido una tarea PARA monitorizar mi conducta

- Se ha omitido una H.U. del que el estudiante no ha dispuesto a tiempo.

8. JUSTIFICACIÓN CIENTÍFICA DEL MÓDULO DE HÁBITOS DE VIDA

Los hábitos de vida son un aspecto fundamental a modificar en cualquier intervención psicológica. En personas con dolor es de vital importancia que algunos ámbitos queden correctamente regulados. Es el caso del sueño, el nivel de actividad y la toma de medicación, por ejemplo.

Uno de los principales problemas a la hora de obtener buenos resultados en las intervenciones dirigidas a modificar hábitos de vida es la falta de motivación intrínseca por la realización de actividades y el olvido de las tareas asignadas. La implementación de un gestor de tareas ayudaría a solventar estos problemas ya que 1) Aumenta la motivación por el cambio permitiendo observar los objetivos cumplidos para que puedan ser relacionados con los logros conseguidos y 2) permite que la persona pueda visualizar las tareas asignadas para cada día y se haga

consciente de las tareas que tiene pendientes.

Es necesario avanzar en la investigación y desarrollo de este tipo de herramienta aplicada al tratamiento de adolescentes y adultos jóvenes con dolor, ya que no hay estudios al respecto.

El extendido uso de los teléfonos móviles en esta franja de edad los convierte en el soporte ideal para implementar esta herramienta, ya que permite que la persona pueda llevarla consigo y tenga acceso a la misma en todo momento.

El desarrollo de este gestor adaptado a población con dolor ayudará a construir un tratamiento que permita atender a los pacientes a distancia y de forma autónoma. Además, permite almacenar los datos automáticamente evitándose así posibles datos missing o introducidos erróneamente.

Rocío de la Vega
Responsable del proyecto general (Tesis)

9. FUNCIONAMIENTO GENERAL DEL MÓDULO DE HÁBITOS DE VIDA

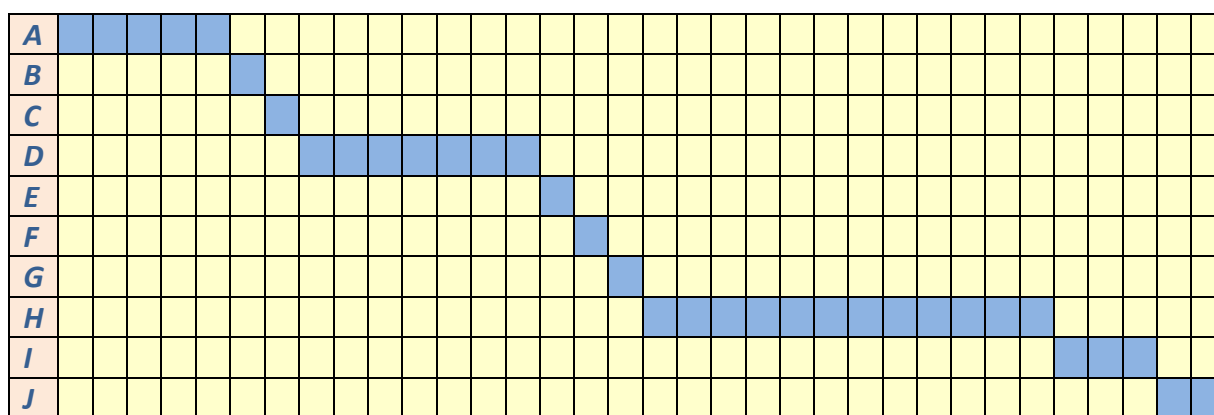


10. PLANIFICACIÓN TEMPORAL

El alumno prevé poder dedicar a la realización del proyecto unos 65 días útiles, a razón de 3 horas diarias, excluyendo del periodo total los fines de semana y los días festivos.

A. Planteamiento y establecimiento de la idea del proyecto	10 días.
B. Estudio de las metodologías disponibles y elección de la que se usará	2 días.
C. Estudio de las tecnologías disponibles y elección de las que se usarán	2 días.
D. Curva de aprendizaje de las tecnologías que se emplearán	14 días.
E. Establecimiento del modelo de tablero kanban (columnas, filas...)	2 días
F. Análisis de requisitos con Historias de usuario y tareas	2 días.
G. Establecimiento de los requisitos no funcionales	2 días.
H. Iteraciones sobre cada tarea hasta salir del tablero kanban	24 días.
I. Elaboración de la memoria	6 días
J. Elaboración de la Presentación del Proyecto	4 días.

Gráfico de Planificación Temporal



- Cada cuadro representa 2 días útiles.
- Las tareas I y J se irán realizando a lo largo de todo el periodo.

Capítulo 3:

Organización y Opciones

INDICE DEL CAPÍTULO

1. Equipo de trabajo.
2. Medios de Comunicación y Colaboración.
3. Metodología General del Proyecto. Kanban.
4. Programación Colaborativa. GitHub.
5. Control de Versiones. Git.
6. Entorno de Desarrollo.
 - Eclipse.
 - Wholesale Application Community
 - Git
 - Aptana
 - Android
7. Frameworks, Librerías y otras Tecnologías.
8. Información detallada de las diferentes Tecnologías.
 - a) HTML, CSS y JavaScript
 - b) Wholesale Application Community
 - c) PhoneGap
 - d) Sencha Touch
 - e) NodeJS

1. EQUIPO DE TRABAJO DE FIBROLINE

Rocío de la Vega

- *Es la responsable del proyecto-tesis general Fibroline.*
- *Psicóloga, está haciendo su doctorado en tratamiento contra la fibromialgia con tecnología móvil en la Universidad Rovira i Virgili (URV) de Tarragona*

Jordi Miró

- *Es el responsable del grupo y director de la tesis.*
- *Catedrático en psicología de la URV de Tarragona.*

Roman Roset Mayals

- *Tutor de proyecto de Raúl Gil Sarmiento, autor de este proyecto.*
- *Miembro del Dpto.de Psicología de la URV de Tarragona.*
- *Unitat per a l'Estudi i Tractament del Dolor - ALGOS. Research on Pain.*

Andrés Masero

- *Estudiante de Ingeniería Informática en la UOC.*
- *Realiza su PFC en el área de desarrollo de aplicaciones para móviles con WAC.*
- *Se ocupa de realizar el módulo para tomar la intensidad de dolor al paciente.*
- *Su forma de trabajar es autónoma e independiente.*

Raúl Gil Sarmiento

- *Autor del presente PFC (parte del proyecto general-tesis).*
- *Estudiante de Ingeniería Informática en la UOC.*
- *Realiza su PFC en el área de desarrollo de aplicaciones para móviles con WAC.*
- *Se ocupa de realizar el módulo gestor para incorporar hábitos y rutinas.*
- *Trabaja colaborativamente con el resto del equipo.*

2. MEDIOS DE COMUNICACIÓN Y COLABORACIÓN PREVISTOS



Email.

Soporta el grueso de la comunicación. Se incluyen como destinatarios de los mensajes a todos los miembros del equipo a los que la información pueda afectar.



Teléfono.

Se utiliza para tratar aquellos asuntos que requieran de gran interactividad y fluidez en la comunicación. Se han establecido los horarios de disponibilidad de algunos de los miembros del equipo



Skype.

Útil para mantener video-conferencias o conversaciones telefónicas sin coste económico.



Trello.com

Mediante este servicio, disponible on-line, se comparte el tablero kanban.

TeamViewer



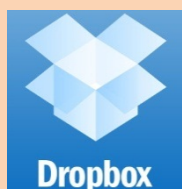
TeamViewer

Herramienta software que permite a varios usuarios operar simultáneamente sobre el mismo ordenador y escritorio. De ese modo se pueden realizar demostraciones, explicar procedimientos, etc.



GitHub

Es un sistema de control de versiones que además permite desarrollar colaborativamente a pesar de la dispersión geográfica de los miembros del equipo. Integra fácilmente las aportaciones de los distintos desarrolladores y almacena los sucesivos estados y ramas por los que discurre el desarrollo.



DropBox

Permite sincronizar automáticamente a través de internet carpetas ubicadas en diferentes ordenadores. Así, compartir archivos es tan fácil como arrastrarlos a una carpeta local. Lo utilizaremos para compartir archivos diferentes al código.

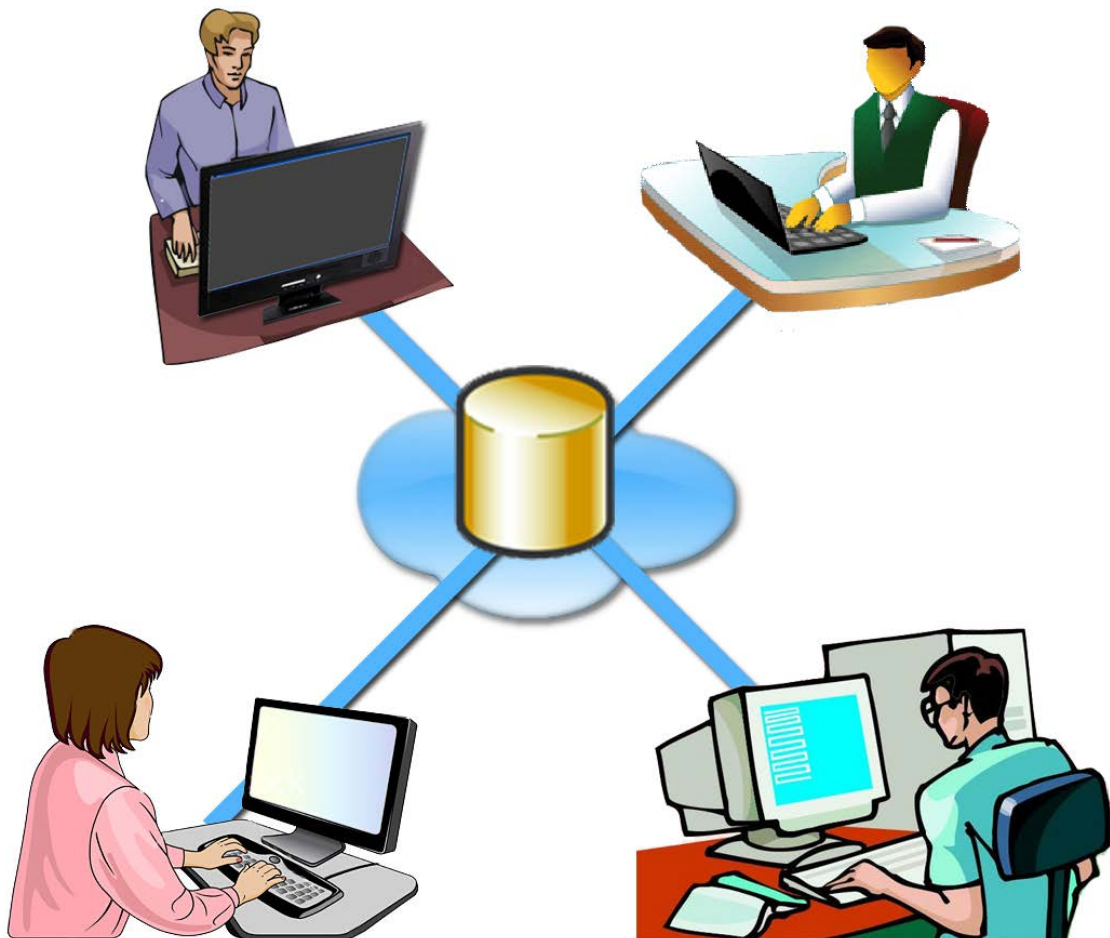
4. PROGRAMACIÓN COLABORATIVA

Para poder programar colaborativamente sobre un mismo proyecto de software se ha utilizado un repositorio centralizado en internet. Este repositorio almacenará los diversos archivos y carpetas que van conformando el proyecto en desarrollo.

Los miembros del equipo realizan inicialmente una copia local del repositorio central, trabajan sobre esta copia local, y periódicamente la sincronizan con el repositorio central. Se trata por tanto de un sistema híbrido centralizado-distribuido.

El repositorio central escogido se llama Github. Se trata de un servicio ofrecido gratuitamente para cierto tipo de proyectos. Dispone de una interface web para la creación del repositorio, su administración, etc.

Por otro lado cada desarrollador necesita instalar en su ordenador un software, denominado Git, que gestiona la copia local del repositorio y permite interactuar con el repositorio central de internet. Git suele permitir resolver con facilidad los posibles conflictos que pudieran darse al sincronizar los repositorios local y central.



5. CONTROL DEL VERSIONES

Un proyecto de software va evolucionando desde un repositorio vacío hacia una aplicación totalmente operativa. HT.

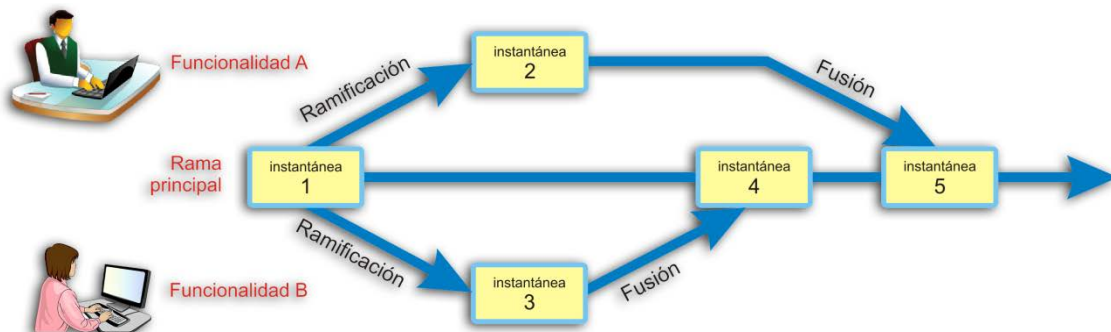


Por otro lado, desde un mismo nodo pueden partir varias ramas de desarrollo paralelo. Estas bifurcaciones pueden tener varios propósitos, como por ejemplo:

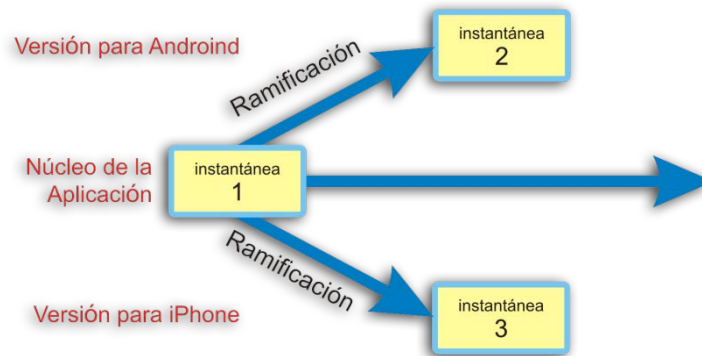
- Explorar diferentes opciones sin comprometer el desarrollo principal. Si una determinada rama exploratoria tiene éxito, se integrará a la rama principal, en caso contrario será descartada.



- Desarrollar de modo separado una funcionalidad o parte del proyecto claramente diferenciable. Una vez desarrollada y comprobada la funcionalidad, puede integrarse en la rama principal. Esto permite que la mayor parte del tiempo los desarrolladores trabajen en ramas separadas sin interferencias entre ellos.



- *Mantener diferentes variaciones del software. En ocasiones es necesario ofrecer al mercado simultáneamente varias versiones de un mismo software. Un ejemplo podría ser el de una aplicación ofrecida para diferentes sistemas operativos. Podría emplearse una rama para cada una de las variaciones del software.*



- *Diferenciar versiones en función de su estabilidad y nivel de integración de las diferentes partes. Por ejemplo:*
 - *Rama liberación. Recoge las versiones totalmente finalizadas y probadas, listas para su comercialización y explotación.*
 - *Rama desarrollo. Sirve para ir integrando las diferentes partes del proyecto y testear así la aplicación en su conjunto.*
 - *Ramas parciales. En cada rama parcial se desarrolla una parte o funcionalidad importante de la aplicación.*
 - *Ramas auxiliares. Se emplean para sub-ramificar las ramas parciales con el propósito de experimentar, como ya se ha descrito, diferentes tentativas.*

El control de versiones mencionado hasta ahora podría ser realizado de modo manual, pero es muy aconsejable emplear un software específico, ahorrando con ello tiempo y evitando errores.

Nosotros empleamos el software ya mencionado, Github como sistema central y Git en local. De hecho, Git es primordialmente un sistema de control de versiones, al que Github proporciona un carácter colaborativo y distribuido.



Eclipse.

Es el componente fundamental, que ofrece la mayor parte de las funcionalidades de desarrollo necesarias.

Mediante diferentes plugins se integran en Eclipse otras herramientas configuradas externamente.

Los plugins también permite ampliar directamente las capacidades del entorno.



WAC.

Es un completo ecosistema de desarrollo y distribución de aplicaciones para dispositivos móviles.

Sus herramientas de desarrollo pueden integrarse directamente en Eclipse mediante un plugin.



MsysGit 1.7.7.1

Se trata del sistema de control de versiones GIT para Windows.

Se instala en el sistema operativo, pero se integra en Eclipse gracias al siguiente plugin (Aptana Studio)



Aptana Studio.

Es un plugin para Eclipse que amplía las capacidades de desarrollo web de Eclipse. Entre estas capacidades está la integración de MsysGit en Eclipse.



Android SDK

Es una herramienta de desarrollo para este sistema operativo, que entre otras cosas ofrece un emulador de terminal Android. Usamos este emulador para probar la aplicación en momentos clave.

Se integra en Eclipse mediante un plugin

- *En el Anexo III se presenta una guía resumida de instalación y configuración del entorno elaborada por el estudiante.*

RESUMEN



HTML5 + CSS3 + JavaScript.

Se programará empleando estas tres tecnologías, puesto que reutilizan los conocimientos que ya se pudieran tener de ellas y sobre todo porque permiten desplegar la misma aplicación en diferentes plataformas móviles: Android, iPhone, BlackBerry, etc. sin apenas esfuerzo adicional.



WAC.

Proporciona ciertas librerías javascript, entre las que se encuentran las de acceso a las APIs de control de las funcionalidades nativas de buen número de terminales.

WAC, además permite encapsular la aplicación como nativa, para su distribución y posterior despliegue en los terminales de los usuarios. También incorpora máquina virtual para pruebas, así como múltiples ejemplos de desarrollo.



Sencha Touch

Se utilizará como Framework de programación JavaScript orientado a dispositivos móviles.



PhoneGap.

Realiza una función similar a WAC. Aunque en este PFC se empleará WAC, el proyecto general quizá se distribuya y despliegue finalmente con PhoneGap



NodeJS

Servidor web cuya principal característica es que se programa en lenguaje JavaScript. Proporcionará servicios web a la aplicación instalada en los terminales.

8. INFORMACIÓN MÁS DETALLADA SOBRE LAS DIFERENTES TECNOLOGÍAS

a) HTML, CSS y JavaScript



Estas tres tecnologías se emplean conjuntamente para programar la aplicación Fibroline.

Inicialmente se idearon para crear sitios web. Sus capacidades fueron aumentando para dotar de mayor riqueza y dinamismo las webs hasta permitir desarrollar con ellas aplicaciones completas que se ejecutan on-line.

Estas tres tecnologías no necesitan en

realidad un servidor web, puesto que los elementos correspondientes pueden almacenarse y ejecutarse localmente con la única ayuda de un navegador compatible.

Por tanto, en su versión final, Fibroline se instalará localmente en los terminales sin requerir conectividad para su operación normal. No obstante, durante el desarrollo sí podrán ejecutarse cargando desde un servidor web.



Acrónimo de HyperText Markup Language, es la más veterana de las tres tecnologías.

Aunque tradicionalmente se ocupa de proporcionar la estructura y el contenido de las webs, en nuestra aplicación Fibroline habrá un solo archivo de extensión HTML, que hará de elemento integrador todos los demás.

HTML representa no sólo el lenguaje de marcas en sí mismo, sino también el motor que proporciona funcionalidad y soporte a las otras tecnologías.

El último estándar, HTML 5, proporciona características que hasta ahora sólo podían aportar componentes externos no estándares, sobre todo en el apartado multimedia.



Acrónimo de Cascading Style Sheets. Permite controlar el aspecto estético y visual.

Inicialmente, su función era desempeñada de modo más rudimentario por el propio lenguaje HTML, pero pronto se vio las ventajas de separar el contenido de la presentación.

La última versión disponible, CSS3, aporta grandes avances que permite ofrecer aplicaciones y sitios más elegantes y visualmente atractivos con menos esfuerzo y simplicidad.

Nuestra aplicación incluye múltiples archivos CSS. La mayor parte de ellos vienen preincorporados en los frameworks y librerías que utilizamos. Estas librerías CSS incorporadas intentan emular la estética y comportamiento visual de las interfaces gráficas propias de los diferentes sistemas operativos de los dispositivos móviles.



Es el lenguaje de programación que convierte una página web en una verdadera aplicación. Se ejecuta en el navegador del cliente desde el que se accede a la aplicación, no en el servidor.

Java Script permite modificar dinámicamente los otros elementos (HTML y CSS) que en principio son estáticos.

Cada navegador entiende un dialecto diferente de este lenguaje, lo cual puede complicar mucho un desarrollo abierto al mayor número de clientes. Aunque estas diferencias también se dan en HTML y CSS, en el caso de JavaScript estas pueden tener consecuencias más importantes.

Para salvar este obstáculo, existen librerías de código JavaScript que proporciona funciones de más alto nivel que dichas librerías adaptan dinámicamente al navegador concreto en el que se estén ejecutando en un determinado momento, eximiendo de esta tarea al programador.

Por otro lado, también existen librerías de código JavaScript que encapsulan funcionalidades de alto nivel, simplificando el desarrollo. Nosotros emplearemos varias de estas librerías, que se comentarán más adelante, como WAC y Sencha Touch.

APLICACIONES CON TECNOLOGÍA WEB VS APLICACIONES NATIVAS

Cada sistema operativo móvil proporciona un lenguaje y una plataforma de desarrollo nativos para programar las aplicaciones que se ejecutarán sobre los dispositivos equipados con aquellos sistemas operativos.

Por ejemplo:

- Android emplea el lenguaje Java y unas librerías específicas.
- iPhone, por su parte, utiliza objective-c.
- BlackBerry también utiliza JavaME.

Frente a la programación de dispositivos móviles empleando los lenguajes nativos de los correspondiente sistemas operativos, se presenta recientemente como alternativa el desarrollo de aplicaciones mediante tecnologías web, esto es HTML+CSS+JavaScript. Para ello aprovechan que todos los dispositivos móviles incorporan navegadores más o menos estándares que reconocen HTML+CSS+JavaScript.

Una y otra solución tienen ventajas e inconvenientes, y pueden ser más o menos acertadas en función de varios factores, como el tipo de aplicación que se desea desarrollar.

Programación empleando Lenguajes y Plataformas Nativos.

Ventajas	Inconvenientes
Aprovecha al máximo el Hardware del dispositivo	La aplicación sólo funciona en el sistema operativo para el que fue desarrollado: <ul style="list-style-type: none">- Requiere un aprendizaje por cada S.O.- Los plazos y recursos necesarios crecen
Es más veloz en su ejecución	
Fácil distribución a través de canales propietarios bastante maduros	

Programación empleando Tecnología Web.

Ventajas	Inconvenientes
La misma aplicación es ejecutable desde la mayor parte de los sistemas operativos móviles, ahorrando enormes costes de desarrollo y mantenimiento.	Su funcionamiento es más lento, dado que se ejecuta sobre un navegador y requiere interpretación previa.
Aquellos que ya programan para la web reutilizan sus conocimientos previos.	No permite acceder a hardware específico del dispositivo (esta limitación puede solventarse con WAC o PhoneGap)
Son tecnologías estándar y abiertas, con todos los beneficios que ello conlleva.	

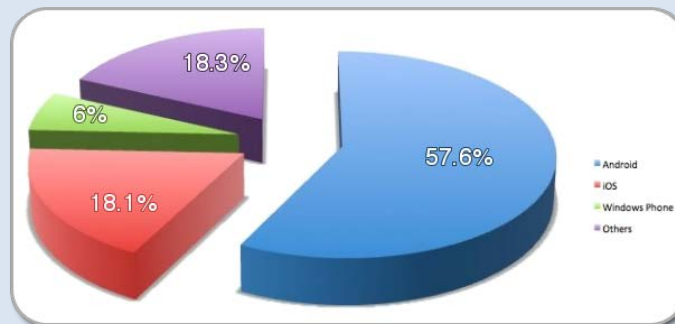
De entre las ventajas de desarrollar empleando tecnologías web, destaca el hecho de que una misma aplicación puede ejecutarse en casi todos los sistemas operativos móviles, sin apenas adaptaciones.

Esto es así porque las aplicaciones web se ejecutan sobre los navegadores incorporados en los diferentes sistemas operativos móviles, y estos utilizan los estándares HTML+CSS+JS. Además, la mayor parte de los navegadores móviles utilizan el mismo motor, llamado WebKit, garantizando aún más la compatibilidad.

Sólo será necesario un desarrollo para abarcar la mayoría de sistemas operativos móviles (y por tanto de usuarios), con el correspondiente ahorro de tiempo y recursos que ello conlleva. Y esta ventaja no se circunscribe únicamente al desarrollo inicial del producto, sino que se prolonga durante las fases de mantenimiento correctivo, perfectivo, etc.

Además, centrarse en una sola tecnología ahorra tener que contar con programadores capacitados en varios lenguajes de programación diferentes. Desarrollar de forma nativa para varios S.O. móviles requiere además contar con distintos paquetes de desarrollo e incluso tener que instalar más de un S.O. de escritorio (por ejemplo, desarrollar para iPhone OS requiere de un MAC y sus sistema operativo (no, la virtualización no funciona bien aquí).

Los desarrollos nativos pueden optar por salvar este inconveniente programando para uno sólo de los sistemas operativos, pero de ese abarcarán una menor cuota de mercado, como refleja el siguiente gráfico. Refleja las previsiones de cuotas de mercado de los diferentes S.O. para 2012 según la web Digitimes.



Por otro lado, **las dos grandes desventajas de desarrollar empleando tecnologías web**, son:

- Su funcionamiento es más lento, (especialmente en aplicaciones con grandes exigencias gráficas y de cálculo).
- De por sí solas, las tecnologías web no pueden acceder directamente a hardware específico de los dispositivos, aunque poco a poco los navegadores móviles van incorporando un mayor acceso a funciones de hardware estándares, como es el caso de la geo-localización. Además, ya existe productos como WAC y PhoneGap que salvan en buena medida estas limitaciones, como se verá más adelante.

WAC

wholesale applications community

Cada sistema operativo móvil dispone de su propio mercado o tienda de distribución de aplicaciones. Dicho Mercado supone grandes beneficios y poder de control por parte de sus propietario.

WAC es un consorcio de operadores y fabricantes de telefonía móvil (actualmente tiene 57 miembros) que sale al paso de esta fragmentación proponiendo una plataforma de desarrollo y distribución abierta y basada en estándares, que permita desarrollar una sola aplicación para todos los sistemas operativos móviles y distribuirla a los potenciales usuarios de todo el mundo. Así, WAC es tanto una plataforma de desarrollo de aplicaciones móviles como una plataforma o red de distribución de dichas aplicaciones.

La plataforma de desarrollo se basa en el estándar Widget de la W3C (ver cuadro de la derecha), que no es sino una aplicación web (HTML + CCS + JavaScript) comprimida en un solo archivo ZIP y almacenada en local para su ejecución off-line.

¿Qué añade WAC al estándar Widget?

- ❖ *Un plataforma de distribución de las aplicaciones, denominado también repositorio*
- ❖ *Un entorno especializado en el desarrollo de aplicaciones móviles.*
- ❖ *Librerías de código JavaScript, hojas de estilo, etc., que facilitan el desarrollo de aplicaciones móviles.*
- ❖ *Una API que permite accede a funciones*

¿Qué es un Widget?

En resumen, un widget es una aplicación web completa (elaborada con tecnologías web, HTML, CCS y JavaScript principalmente) almacenada localmente y lista para su ejecución.

Los elementos HTML, CCS y JavaScript, una vez en el dispositivo local, no requieren de la participación de un servidor externo. Por tanto, la idea es descargarlos una sola vez y ejecutar la aplicación tantas veces como se desee.

Esta independencia de la red hace que la aplicación pueda ejecutarse a pesar de estar sin conexión, disminuye el tráfico de red y evita una ralentización de la aplicación cuando la red está saturada o la señal es pobre. Evidentemente, si la aplicación necesita acceder a datos on-line requerirá de conexión.

La organización de estándares de Internet, W3C desarrolló un estándar para distribuir y ejecutar con facilidad estas aplicaciones web locales denominadas Widget.

Un widget listo para su distribución no es más que una aplicación web completa, mas algún archivo de información adicional específica de los widget, todo ello comprimido en un archivo zip.

Hasta ahora, se ha venido asistiendo al boom de los pequeños widgets que añaden pequeñas utilidades al escritorio, pero no hay motivos por los cuales no puedan desarrollarse bajo este estándar potentes aplicaciones exclusivas hasta ahora de lenguajes de programación local tradicionales.

avanzadas del hardware de los dispositivos, fuera del alcance de la tecnología web convencional. Este es uno de sus principales bazas, dado que las aplicaciones web se ejecutan habitualmente en una caja que impide el acceso al resto del sistema anfitrión.

Esto es posible gracias a que los grandes fabricantes de móviles, miembros del consorcio WAC, han incorporado esta funcionalidad en sus terminales.

- ❖ *Un conjunto de API de red para acceder a funcionalidades íntimamente ligadas a las redes de telefonía móvil a las que se encuentran conectados los terminales.*

Nuevamente, se trata de una característica fuera del alcance de una aplicación web convencional y que permite, acceder a servicios de tarificación, envío de SMSs, monitorización del estado de la red GSM, etc.

Tampoco sería posible si no fuera por la colaboración de los principales operadores de telefonía móvil, miembros del consorcio WAC.

c) PhoneGap

El objetivo principal de PhoneGap es lograr que aplicaciones web basadas en HTML, CSS y JavaScript puedan convertirse en aplicaciones nativas capaces de ser distribuidas por los canales convencionales de cada sistema operativo y ejecutadas como cualquier otra aplicación nativa en los dispositivos.

Realmente, lo que hace PhoneGap es envolver la aplicación web en una aplicación nativa que incorpora su propio navegador web. La aplicación nativa está programada para extraer en memoria la aplicación web que alberga en su interior y ejecutarla en el navegador web incorporado.

Es una alternativa al anteriormente mencionado WAC. No se empleará en el presente proyecto de carrera, pero se menciona porque sí está previsto su uso en el proyecto general de tesis Fibroline.



Sencha Touch fue el primer framework HTML5 para el desarrollo de aplicaciones móviles.

El nuevo estándar HTML5, entendido en sentido genérico (HTML, CSS3, etc.), significó un gran avance en el ámbito de la programación web. Sin embargo, desarrollar partiendo de cero con los componentes básicos de HTML5 aún continúa siendo una tarea ardua si la aplicación es compleja, especialmente teniendo en cuenta las manifiestas incompatibilidades entre los diferentes navegadores.

Para facilitar la labor a los desarrolladores surgieron frameworks que ofrecen todas las funcionalidades de HTML5 pero a un mayor nivel de abstracción.

Sencha Touch no sólo facilita el desarrollo de aplicaciones HTML5, sino que lo adapta a los dispositivos móviles. Además, uno de sus principales objetivos es que las aplicaciones desarrolladas emulen en todo lo posible las interfaces de usuario y los efectos multimedia de las aplicaciones nativas de los diferentes sistemas operativos móviles.

Para lograrlo, Sencha Touch integra y adapta varias librerías y frameworks preexistentes:

- *Ext. JS: Podría decirse que es el componente fundamental de Sencha. Es un completo Framework HTML5 pero no especializado en dispositivos móviles*
- *JQTouch. Se trata de un plugin para JQuery (ver cuadro de la derecha).*
- *Raphäel. Es una pequeña librería javascript que facilita el trabajo con gráficos vectoriales. Se basa en el uso del estándar SVG de la W3G y de VML.*

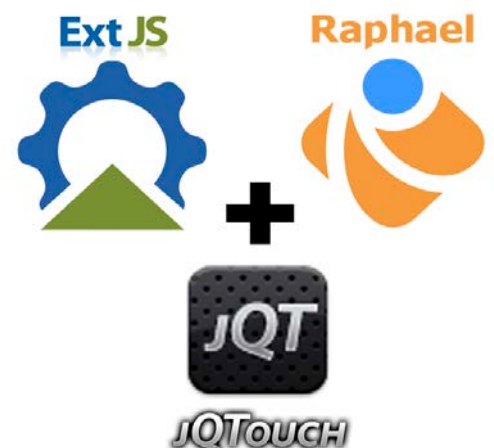
JQuery

Es una librería Javascript ampliamente conocida y utilizada que proporciona sofisticadas funcionalidades para la programación de aplicaciones web, pero con una rapidez y sencillez asombrosa. Ha simplificado enormemente el desarrollo de aplicaciones webs.

Entre los diversos componentes de JQuery caben mencionar los que se ocupan de:

- Las funciones de control del árbol DOM.
- Los componentes de manejo de llamadas AJAX.
- Los componentes visuales que sirven para interactuar con el usuario.

Además, JQuery cuenta con infinidad de plugins elaborados por desarrolladores de todo el mundo que permiten ampliar sus funcionalidades hasta el infinito.



Entre las características y funcionalidades de Sencha Touch destacan las siguientes:

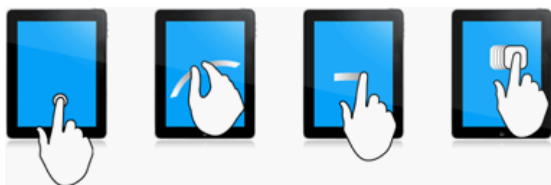
Elaborando con estándares web

En particular HTML5, lo que facilita su compatibilidad, la reutilización de los conocimientos de los desarrolladores, su mejora continua, etc.



Incorpora nuevos eventos táctiles mejorados.

Tap, doble tap, deslizar, arrastrar, etc. Estos se suman a los ya estándares, mejorando la interacción de los usuarios con sus dispositivos móviles



Con un amplio catálogo de widgets.

En la terminología de Sencha Touch, los widgets son componentes visuales que sirven de interface con los usuarios. Funcionan en los dos sentidos, puesto que permite organizar y mostrar del modo más adecuado los contenidos al usuario y recibir las entradas y eventos generados por este.

Estos widgets aprovechan al máximo el potencial de HTML5, CSS3 y JavaScript para crear ricas interfaces con el mínimo esfuerzo y complicación.

Además, imitan casi a la perfección las interfaces nativas de los 3 principales sistemas operativos móviles.



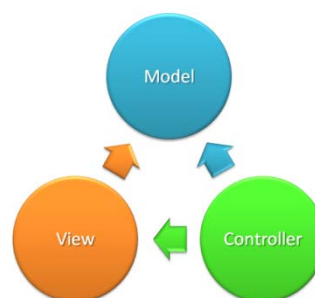
Integración de datos.

Permite acceder a diferentes fuentes de datos, como Ajax, JSONP o YQL. Además, facilita el acceso a los datos almacenados de modo local en el local storage, añadido como gran novedad en HTML5.



Incluye Modelo-Vista-Controlador.

MVC es un paradigma ampliamente reconocido y utilizado que simplifica el desarrollo de grandes proyectos.



Capítulo 4:

Utilización del módulo:

En este capítulo se muestra la utilización y funcionamiento de módulo de Gestión de Hábitos desde el punto de vista del usuario.

INDICE DEL CAPÍTULO

1. ¿Qué es una tarea?
2. Acceso al Módulo Gestor de Hábitos y Rutinas (Módulo de Tareas)
3. Vista de Tareas Propuestas para Hoy.
4. Vista de Detalles de una Tarea.
5. Vista de Encuesta sobre las Tareas No Realizadas el Día Anterior.

1. ¿QUÉ ES UNA TAREA?

Es una acción a realizar por el usuario una o varias veces durante el tratamiento.

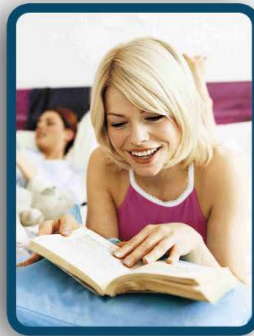
Cada tarea suele constar de varias repeticiones a lo largo del tratamiento, constituyendo así una rutina que con el tiempo se convierte en hábito. Estos hábitos suelen mejorar la calidad de vida del paciente, reduciendo el dolor y las limitaciones de movimiento propias de la enfermedad.

Ejemplos de tareas:

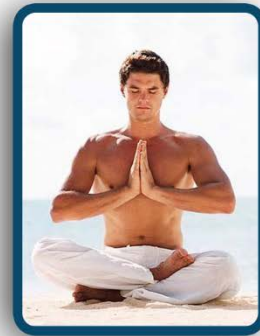
**Tomar Medicación
de la Mañana**



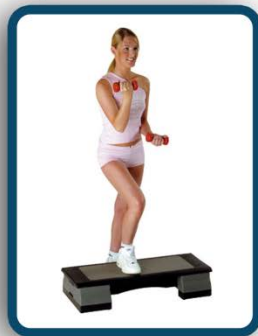
**Leer Lección 2:
"La Medicación"**



**Ejercicios de
Relajación**



**15 minutos de
Ejercicio Aeróbico**



**Autovaloración
del Dolor**



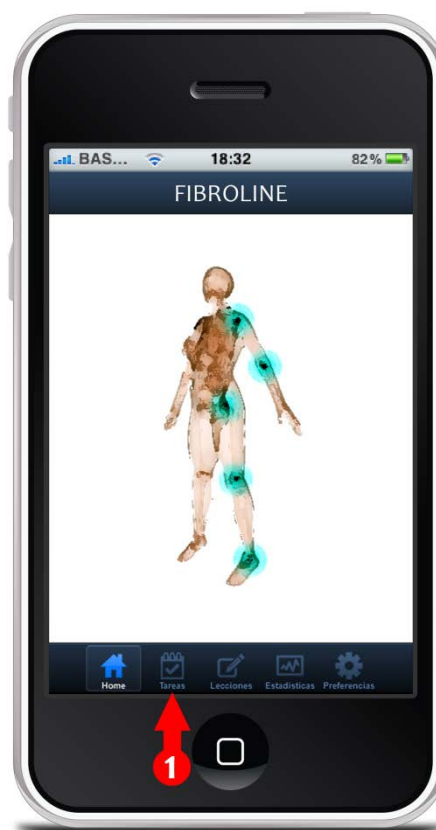
2. ACCESO AL MÓDULO GESTOR DE HÁBITOS o MÓDULO DE TAREAS

Al usuario se le proporciona la aplicación Fibroline ya instalada en su dispositivo móvil y lista para funcionar*. Por tanto, se inicia o lanza como cualquier otra aplicación local, tras lo cual se observará la pantalla de inicio.

Para acceder al módulo de tareas bastará con tocar sobre el icono correspondiente, titulado Tareas (1), de la barra inferior de la aplicación.

No obstante, la aplicación está programada de tal modo que recuerda cuál fue el último módulo y pantalla en el que estuvo, yendo a esta directamente la siguiente vez que es ejecutada.

* Durante su desarrollo podría distribuirse y/o ejecutarse on-line.



3. VISTA DE TAREAS PROPUESTAS PARA HOY



La primera vez que se accede a este módulo se muestra el listado de tareas propuestas para ese día. Cada tarea pertenece a una categoría, la cual determina entre otras cosas el color de texto con que se muestra.

Lógicamente, al principio todas las tareas del día se encuentran pendientes de hacer, como muestra el encabezado superior 'pendientes'.

Al tocar sobre el cuadro de verificación (2) que se encuentra a la izquierda de algunas tareas, dicha tarea queda marcada como realizada y pasa a la parte inferior de la pantalla, bajo el encabezado 'Realizadas'.



Una marca en el cuadro de verificación (3) que se encuentra a la izquierda de la tarea realizada indicará también su estado.

Algunas tareas no tienen cuadro de verificación a su izquierda (4). Son tareas que el usuario no puede marcar directamente como realizadas, sino que su ejecución es monitorizada por la aplicación y es esta la que la marca automáticamente.

Un ejemplo de este tipo de tareas de marcación automática es la 'valoración de la intensidad del dolor' que realiza el usuario por la mañana desde otro de los módulos de la aplicación.

Al ser una tarea que se realiza mediante el dispositivo, la aplicación es capaz de detectar su ejecución y modificar correspondientemente su estado.

Las tareas automáticas también son desplazadas a la parte inferior de la pantalla cuando son realizadas, y señaladas con la marca de verificación, aunque carecen del recuadro (5).

El usuario, a lo largo del día, puede rectificar tantas veces como quiera las acciones de marcado que haya realizado. Es decir, que si por error marcó como realizada una tarea que no lo estaba, puede tocar de nuevo sobre el cuadro de verificación que se encuentra a su izquierda, con lo que regresará a su estado de 'Pendiente', en la parte superior de la pantalla, y desaparecerá la marca de realización.



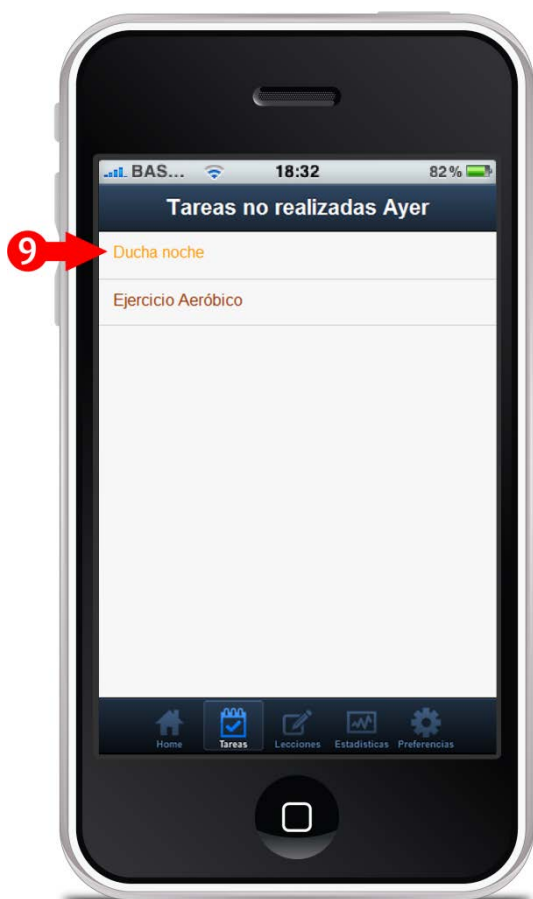
4. VISTA DE DETALLES DE UNA TAREA

Todas las tareas de la vista 'Tareas para Hoy' tienen un icono de flecha a su derecha (6). Tocando dicho icono se accede a una vista que muestra información detallada sobre la tarea.

Como se puede observar en la figura, la vista detallada contiene tanto información invariable (título, descripción, breves instrucciones de realización...) como información actualizada sobre su estado de ejecución (veces y fechas en las que se ha realizado, etc.)

La vista de detalle también puede disponer de una acción asociada, accesible tocando el enlace correspondiente (7). Por ejemplo, la tarea "Lección 6: La Medicación" llevará al módulo de lecciones.

Para regresar a la vista 'Tareas para hoy' basta tocar el botón 'Volver' que se encuentra en la esquina superior izquierda (8).



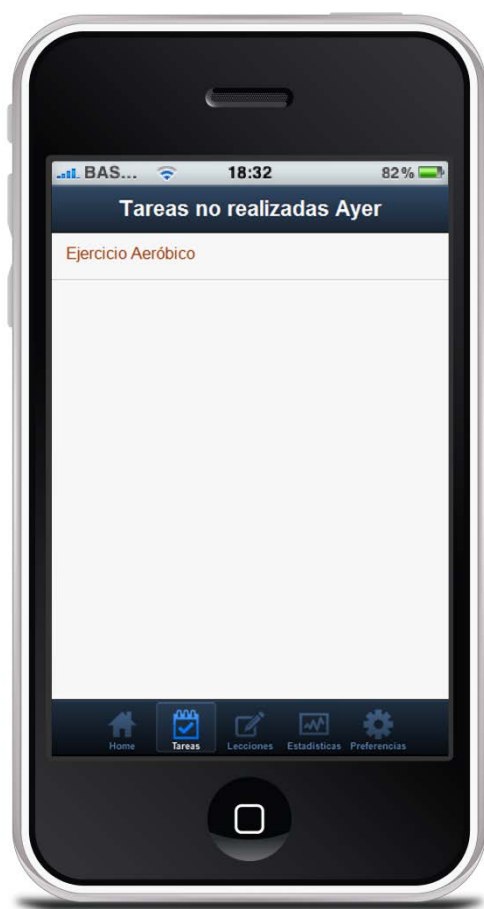
5. VISTA DE ENCUESTA SOBRE LAS TAREAS NO REALIZADAS EL DÍA ANTERIOR

Cada nuevo día, la primera vez que el usuario accede al módulo de tareas, se le presenta un listado con las tareas que dejó sin realizar el día anterior. Para cada una de esas tareas, se le preguntará al usuario el motivo por el cual no la realizó.

Para ello, el usuario deberá ir tocando sobre cada una de ellas (9).

En cada ocasión se desplegará un listado de posibles motivos, no teniendo el usuario más que tocar la respuesta que desee (10).

Las tareas van desapareciendo del listado conforme son justificadas. Cuando ya no quedan tareas sin justificar, se cierra la vista de encuesta y aparece la vista de 'Tareas para Hoy'.



Capítulo 5:

Funcionamiento Interno

En este capítulo se trata el funcionamiento interno del módulo Gestor de Hábitos.

INDICE DEL CAPÍTULO

1. ¿Qué función tiene el Módulo Gestor de Hábitos?
2. ¿Qué propiedades posee cada tarea?
3. ¿Dónde se almacenan las tareas?
4. Modelo-Vista-Controlador.
5. Componentes de Módulo Gestor de Hábitos.
6. Controlador: TaskController.js.
7. Modelo: TaskModel.js y TaskStore.js.
8. ¿Cómo se determinan las tareas a realizar hoy?
9. Control de la Vista de Tareas para Hoy.
10. Control de la Vista de Tareas No Realizadas el Día Anterior.

1. ¿QUÉ FUNCIÓN TIENE EL MÓDULO GESTOR DE HÁBITOS?

- *Carga en la aplicación la última versión del tratamiento disponible en el servidor.*
- *Determina y muestra cada día al usuario las tareas que debe realizar.*
- *Monitoriza la realización de las tareas diarias.*
- *Registra información sobre la realización de las tareas.*
- *Pregunta y registra los motivos por los que se hayan dejado de realizar ciertas tareas.*

2. ¿QUÉ PROPIEDADES POSEE CADA TAREA?

- *id* Código numérico único que sirve como clave identificativa de cada tarea.
- *tit* Título o nombre corto de la tarea.
- *des* Descripción breve de la tarea (en torno a 50 caracteres).
- *cat* Categoría a la que pertenece la tarea
- *typ* Tipo de tarea, empleada para completar propiedades ausentes en func. del tipo
- *man* Booleana, señala si el propio usuario puede marcar la tarea como realizada.
- *dep* Tarea de la que depende (0 para indicar que no tiene dependencias)
- *sta* Fecha de inicio (opcional). Formato “YYYY-mm-dd”.
- *end* Fecha de finalización (opcional). Formato “YYYY-mm-dd”.
- *dow* Días de la semana en los que se puede realizar la tarea.
0 para domingo, 1 para lunes, 2 para martes, etc.
Por ejemplo.: “135” significa lunes, miércoles y viernes.
- *tim* Número de veces que debe repetirse la tarea (opcional).
- *cnt* Contador de las veces que se ha realizado la tarea hasta el momento.
- *dlg* Date Log. Array con las fechas en las que se ha propuesto la tarea.
- *alg* Answers Log. Array con los motivos de no realización de tareas.
- *done* Indica si hoy se ha realizado la tarea propuesta o está pendiente.
- *tod* To-do. Booleana. Sirve para marcar las tareas que deben realizarse hoy

Cabe comentar algunas de las propiedades de las tareas:

- cat: Categoría a la que pertenece la tarea.

Las tareas suelen agruparse según bloques temáticos. Por ejemplo, todas las tareas relacionadas con el sueño. Las tareas pertenecientes a la misma categoría se muestran con el mismo color.

- typ: Tipo de tarea.

Sirve para agrupar tareas que tienen un comportamiento similar en cuanto a su programación de repeticiones.

Por ejemplo, las tareas que se realizan tras leer un tema formativo no tienen fecha específica de inicio (comienzan cuando se haya leído el tema), dependen de otra tarea (la lectura del tema correspondiente), y no tiene establecido un número de repeticiones, sino que se mantienen hasta el final del tratamiento.

Los posibles tipos de tarea no se encuentran prefijados, sino que se podrán crear tantos tipos de tarea como se desee, con sus características asociadas. De ese modo, podrían dejarse en blanco varias propiedades de una tarea (con la correspondiente simplificación) contando con que dichas propiedades ausentes serán completadas internamente gracias a que se conoce su tipo.

No obstante, este modelo avanzado de tareas y tipos de tareas no ha sido implementado en esta primera versión de la aplicación, aunque se encuentra desarrollado ampliamente en el.

- dlq: Date Log. Registro de fechas.

Es un array en el que quedan registradas las fechas en las que ha sido propuesta al usuario la realización de la tarea, haya sido finalmente realizada o no.

- alq: Answers Log. Registro de respuestas del usuario.

Por cada fecha en que se ha propuesto la realización de la tareas (ver propiedad anterior, dlq) se registra una entrada en este array. Dicha entrada indica si la tareas fue finalmente realizada (valor 0) o el motivo por el cual no fue realizada (valores de 0 en adelante).

- done: Hecha.

Es una propiedad auxiliar que indica si la tarea ya ha sido realizada hoy. Al comenzar cada día es reiniciada a 0.

- tod: To-do. Para hacer.

Es una propiedad auxiliar que señalan las tareas a realizar hoy. Son marcadas al comenzar cada día.

3. ¿DÓNDE SE ALMACENAN LAS TAREAS?

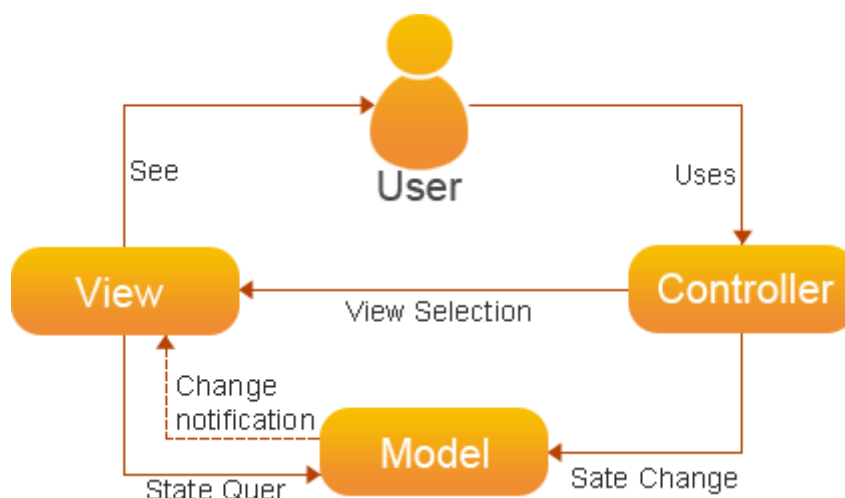
- Las tareas que forman parte del tratamiento son cargadas la primera vez, y cada vez que se encuentre una nueva versión de dicho tratamiento, desde una dirección url determinada. Esta dirección puede ser local o remota. El formato en que se recibe es Json.
- Las tareas cargadas se guardan en el Local Storage del navegador (una novedad del estándar HTML5), que mantendrá su persistencia entre ejecuciones. Las tareas no permanecen invariables en el Local Storage, sino que su estado (determinado por las propiedades vistas en el apartado anterior) van cambiando a lo largo del tratamiento.
- Cuando se requiere interacción del usuario con las tareas, estas se cachean a memoria principal mediante estructuras de datos propias del framework Sencha llamadas 'registros'. Al finalizar cada interacción con una tarea, el registro es reescrito en el Local Storage del navegador para asegurar su persistencia.

4. MODELO-VISTA-CONTROLADOR

El framework Sencha Touch tiene implementado el patrón de diseño modelo-vista-controlador.

Se trata de un paradigma ampliamente reconocido y utilizado, que simplifica el desarrollo de proyectos de mediana o gran envergadura, motivo por el cual ha sido empleado en el presente proyecto.

El patrón MVC distingue tres tipos de tareas a realizar por los componentes del software en desarrollo, e insiste en que cada componente realice una y sólo una de ellas:



- Modelo: Los componentes del modelo son los únicos en contacto directo con los datos que determinan el estado a conservar. Proporciona al resto de los componentes métodos que permiten ver o modificar indirectamente dicho estado. No deben preocuparse de la interacción directa con el usuario ni de aspectos de presentación.
- Vista: Es la interfaz que proporciona interacción con los usuarios. Captura las peticiones del usuario y le presenta del modo más adecuado los datos que éste desea ver.
- Controlador: Es el nexo de unión entre el modelo y la vista. Recibe las peticiones del usuario, capturadas en la vista, y en función de la naturaleza de la petición, del estado del modelo, y del propio estado del controlador, determina la acción a realizar. Con frecuencia, dicha acción consiste en acceder al modelo mediante uno de sus métodos predefinidos, ya sea para modificar o para leer su estado.

5. COMPONENTES DEL MÓDULO DE TAREAS

Vista	Controlador	Modelo
TaskListPanel.js	TaskController.js	TaskModel.js
TaskListPoll.js		TaskStore.js

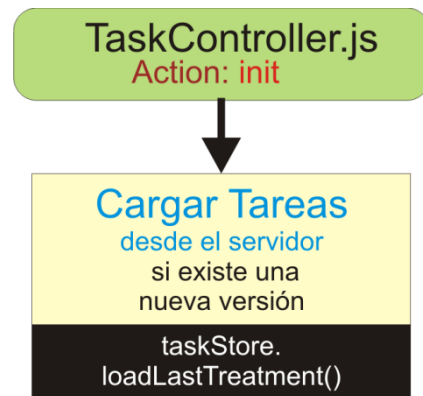
- TaskListPanel.js
Es la vista que muestra la lista de tareas a realizar hoy y permite interaccionar con ellas cambiando su estado de realización o solicitando información detallada.
- TaskListDetails.js
Es la vista que muestra los detalles una tarea.
- TaskController.js
Gestiona el funcionamiento del módulo recogiendo las acciones del usuario, realizadas sobre las vistas, y realizando una acción en consecuencia.
- TaskModel.js
Modela las propiedades que tendrá una tarea.
- TaskStore.js
Gestiona la persistencia de todas las tareas y proporciona al controlador métodos para acceder a estas y modificar su estado.

6. CONTROLADOR (TaskController.js)

INICIALIZACIÓN

Cada vez que es iniciada la aplicación, se ejecuta el método `init` del controlador del módulo de tareas.

El método `init` llama al método `loadLastTreatment` del modelo, que comprueba si existe en el servidor una versión actualizada del tratamiento y en tal caso descarga sus tareas asociadas y las almacena en el local storage del navegador.



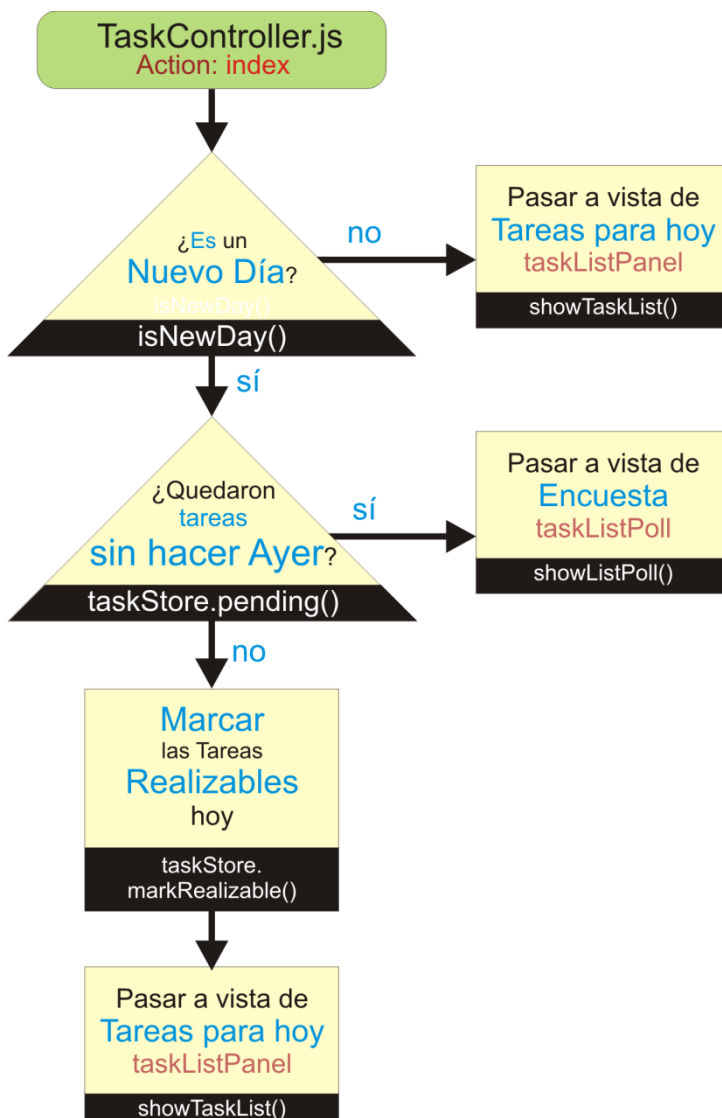
ACCIÓN PRINCIPAL DEL MÓDULO

Cada vez que se llama a la acción principal del módulo de tareas (`index`), se comprueba si se ha producido un cambio de fecha, es decir, si es un nuevo día.

Si no es así, se pasa a la vista de gestión de tareas para hoy (`taskListPanel`).

Si es un nuevo día y quedaron tareas sin hacer el día anterior (método `pending` del modelo), se pasa a la vista de encuesta (`taskListPoll`) que lista esas tareas pendientes y pregunta al usuario el motivo por el cual no fueron realizadas.

Si no quedaron tareas pendientes del día anterior, o ya se respondió a la encuesta, se establecen y marcan las tareas que deben realizarse hoy (método `markRealizable` del modelo) y se pasa a la vista de gestión de tareas del día.



7. MODELO (TaskModel.js y TaskStore.js)

TaskModel.js se limita a definir las propiedades-tipo de cualquier tarea.

TaskStore.js gestiona el resto del modelo de tareas:

- Establece la persistencia de las tareas: El Local Storage del navegador.
- Establece el criterio de ordenación por defecto: estado de realización e id.
- Establece el criterio de agrupación: pendientes de realizar y realizadas.
- Proporciona métodos para modificar el estado del modelo:
 - *filterToday*: filtra y carga en caché las tareas programadas para hoy.
 - *filterPending*: filtra y carga las tareas no realizadas el día anterior.
 - *loadLastTreatment*: averigua si hay disponible una nueva versión del tratamiento y en tal caso llama a *jsonCharge* para su descarga.
 - *jsonCharge*: carga el último tratamiento disponible desde el servidor.
 - *today*: devuelve la fecha de hoy en formato YYYY-MM-DD.
 - *newDay*: establece hoy como último día (para detectar cambio de día).
 - *pending*: devuelve true si quedan tareas sin realizar del día anterior.
 - *markRealizable* (para una explicación detallada, ver el apartado siguiente): recorre todas las tareas marcando aquellas tareas que deben realizarse hoy. Para cada tarea aplica el método *isRealizable*.
 - *isRealizable*: determina si una tarea dada es realizable hoy. Para ello utiliza, las siguientes seis funciones.
 - *isFinished*: determina si una tarea dada ya ha finalizado completamente.
 - *isDone*: determina si una tarea ya ha sido realizada el número de veces establecido.
 - *isLapsed*: determina si una tarea ha rebasado su fecha final.
 - *isAdult*: determina si ya se ha alcanzado la fecha de inicio.
 - *machDayOfWeek*: determina si hoy es uno de los días de la semana en los que está establecido que puede proponerse la realización de una tarea.
 - *isWaitingFor*: determina si la tarea está pendiente de que concluya otra tarea de la cual depende.
 - *doTask*: registra la realización de una tarea.
 - *undoTask*: deshace el último registro de realización de una tarea.
 - *regAnswer*: registra el motivo por el cual ha dejado de realizarse una tarea.

8. ¿CÓMO SE DETERMINAN LAS TAREAS A REALIZAR HOY ?

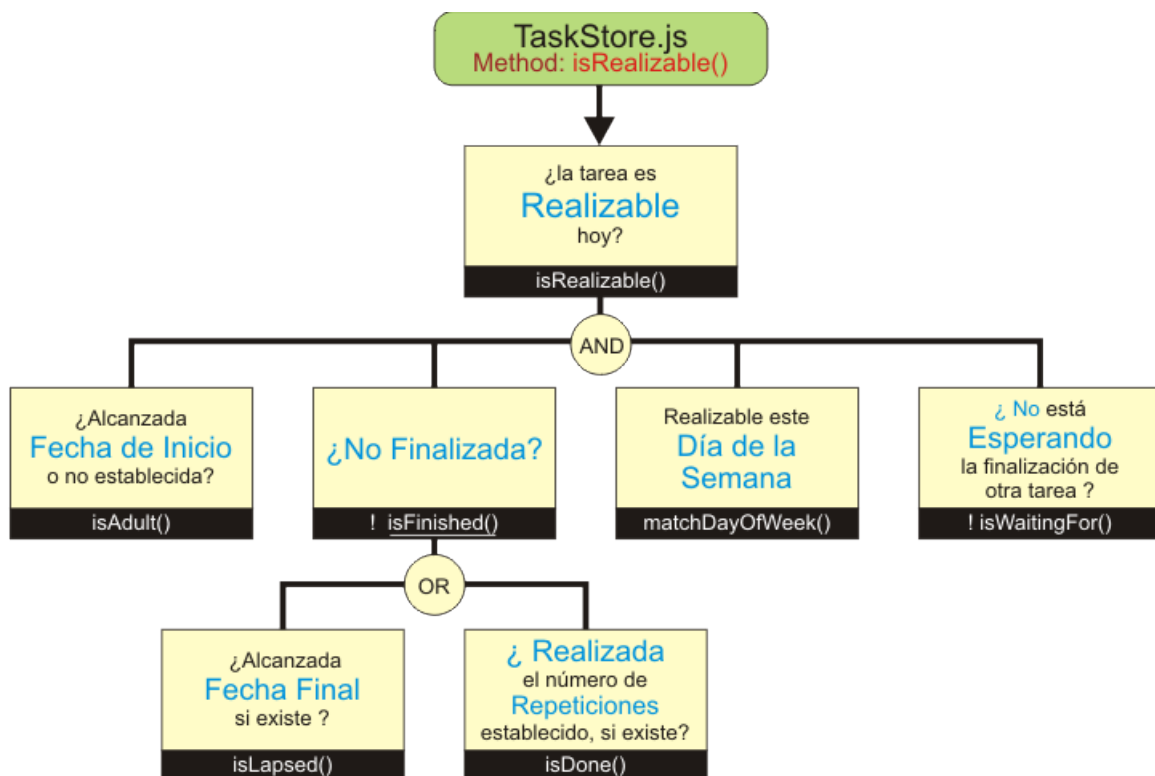
De entre todas las tareas de las que consta el tratamiento, cada día se propondrá realizar al usuario aquellas que se consideren 'Realizables', que son las que cumplan simultáneamente los siguientes requisitos:

- No tener establecida una fecha de inicio, o que ya se haya alcanzado ésta fecha.
- No depender de otra tarea que aún no haya concluido.
- No haber finalizado, ya sea por haber alcanzado una fecha tope (en caso de existir) o por haber sido realizada el número de veces establecido (si existe)
- Que sea uno de los días de la semana establecidos para esta tarea.

El método `markRealizable()` recorre todas las tareas existentes, aplicando a cada una de ellas una función que determina si dicha tarea es realizable ese día. En caso de ser realizable, la tarea es marcada poniendo a `True` el campo 'tod'.

La vista de tareas para hoy (`taskListPanel`) usará el campo 'tod' como filtro para mostrar las tareas que deben realizarse ese día.

La función que termina si una tarea dada es realizable hoy se llama `isRealizable()` y se encuentra en el modelo del módulo. `taskStore.js` devuelve `True` si la tarea es realizable hoy. Esta función se basa en otras tantas funciones auxiliares, como se muestra en la siguiente figura:



9. Control de la VISTA DE TAREAS PARA HOY

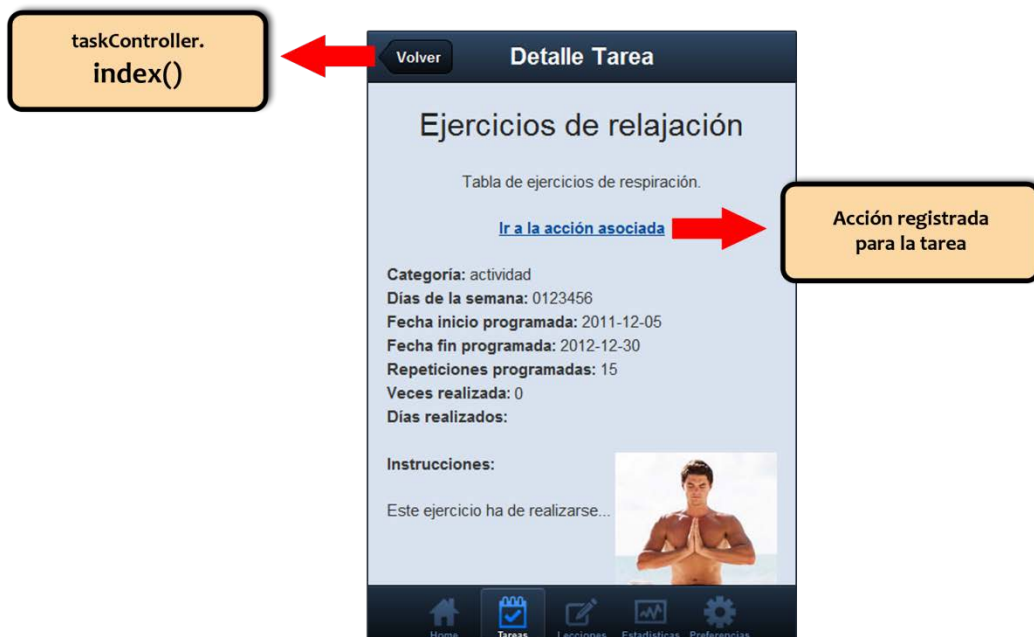
Las acciones a realizar en respuesta a los eventos producidos en la vista no están definidas en la propia vista, sino en el controlador. En este caso `taskController.js`.



10. Control de la VISTA DE DETALLES DE UNA TAREA

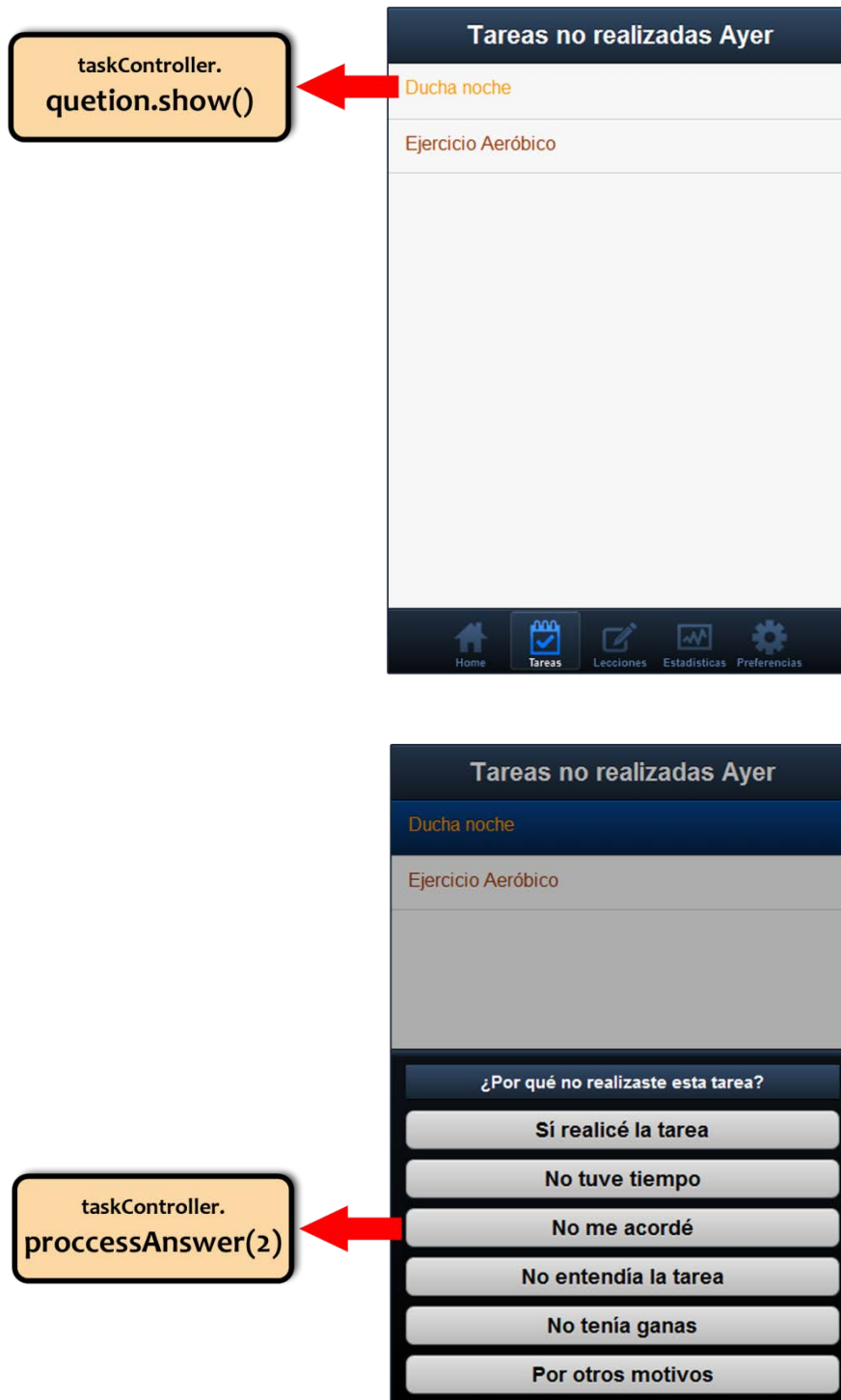
Esta vista sólo genera dos eventos a capturar por el controlador:

- Tocar el botón Regresar: que vuelve a la vista de Tareas para Hoy
- Tocar sobre "acción asociada", que ejecuta la acción que haya sido asociada a la tarea. Normalmente será una acción de uno de los otros módulos de la aplicación. Por ejemplo, la acción asociada a la tarea "Leer Lección 3" llevará al módulo de lecciones, en concreto a la lectura del tema 3.



11. Control de la VISTA DE TAREAS NO REALIZADAS EL DIA ANTERIOR

Las acciones a realizar en respuesta a los eventos producidos en la vista no están definidas en la propia vista, sino en el controlador. En este caso `taskController.js`.



Capítulo 6:

Conclusiones y Futuro

1. AUTOEVALUACIÓN Y CONCLUSIONES

Aunque corresponde al tribunal realizar una evaluación imparcial del presente proyecto, el estudiante considera que se ha cumplido en esencia el **objetivo propuesto**:

Desarrollar un componente software – como parte de una aplicación para dispositivos móviles - que sirva de asistente personal de pacientes de fibromialgia en la gestión y realización de las tareas establecidas en su tratamiento para incorporar hábitos y rutinas que mejoren su calidad de vida.

Considera además que se han cumplido básicamente los **requisitos del módulo**, expresados en el punto 7 del capítulo 2 en forma de Historias de Usuario.

No obstante, **cabén mejorías**. Una de ellas consistiría en el enriquecimiento visual y de presentación de datos de la **vista de detalles de una tarea**.

También habría sido deseable una **mayor integración con el resto** de módulos y con el núcleo de la aplicación, pero no ha sido posible debido a que el proyecto general apenas ha sido desarrollado.

En cuanto a la **planificación**, han existido desviaciones importantes. Ha sido necesario dedicar más tiempo del previsto a la evaluación de las tecnologías disponibles y al estudio de

las tecnologías utilizadas. El número de horas total de dedicación ha sido superior al previsto en aproximadamente un 30%.

El **trabajo en equipo** no ha sido todo lo fluido que se habría deseado, debido a que no se había contado con las irrupciones y dificultades propias de la vida de cualquier proyecto. Además, es justo considerar que todas las interacciones han sido no presenciales.

Respecto de las **expectativas personales**, el estudiante se encuentra muy satisfecho debido a que la experiencia de trabajo en equipo ha sido muy enriquecedora. Se encuentra igualmente complacido por todo lo aprendido en el campo del desarrollo de aplicaciones para dispositivos móviles.

2. LINEAS DE FUTURO

Dado que el proyecto Fibroline se encuentra al comienzo de su desarrollo, el alumno se ofrece a continuar su colaboración, al menos durante un semestre más, si los responsables de tomar dicha decisión lo consideran oportuno.

Si ese fuera el caso, cabría trabajar en:

- *La corrección de las deficiencias del módulo de tareas ya mencionadas*
- *La ampliación de los requisitos del módulo, como una mayor integración con el hardware de los dispositivos sobre los cuales se ejecuta (programación de alertas, etc.)*
- *La integración con el resto de la aplicación.*
- *El desarrollo de una aplicación para dispositivo de escritorio que facilite la programación, gestión y seguimiento de los tratamientos.*
- *El desarrollo de otros módulos pendientes.*

Anexo I: Modelo Avanzado de Tareas

1. INTRODUCCIÓN

En este capítulo se presenta un modelo más avanzado y sofisticado de tareas, que el estudiante elaboró inicialmente.

Ofrece más versatilidad en la programación de tareas de la que a priori era necesaria.

Por ese motivo, el equipo propuso partir de un modelo más simplificado para el desarrollo de la primera versión de la aplicación y recurrir al modelo avanzado durante el desarrollo de futuras versiones si ello se considerara necesario.

No obstante, disponer ya desde el principio de un modelo de tareas avanzado ha servido para que el modelo simplificado se haya diseñado de tal forma que sea fácil de extender.

2. ATRIBUTOS DE UNA TAREA

Id.....	Número identificador único
Nombre	Nombre o Título a emplear cuando el espacio es suficiente
Etiqueta_corta.....	Etiqueta o Título a emplear en la lista de tareas (poco espacio)
Tipo.....	Tipo de tarea
Descripción.....	Breve descripción de la tarea
Fecha inicio (FI).....	Fecha base que podrá utilizarse en la condición de inicio de la tarea
Fecha fin (FF)	Fecha base que podrá utilizarse en la condición de finalización
Repeticiones (Rp)	Número de repeticiones que podrá utilizarse en la cond. finalización
Dependencias_inicio (DI)	Evento o Tarea que podrán emplearse en la cond. de inicio
Dependencias_finalización (DF)	Eventos o Tareas que podrán emplearse en la cond. de finalización
Retardo_inicio	Días desde que se satisfacen las dependencias antes de iniciar
Retardo_finalizacion	Días desde que se satisfacen las dependencias antes de iniciar
Condición_inicio[].....	Condición lógica que determina el comienzo de una tarea
Condición_finalización[]	Condición lógica que determina la finalización de una tarea
Dias[]	Días de la semana en los que se realiza la tarea
Horario	Horario de realización
Postergar	Postergar una repetición si no se realiza un día. Si falso se pierde.
Manual	Si el usuario puede marcar/desmarcar la tarea como realizada.
Diario[].....	Diario de realización de las tareas: Día, Hora, Realizada y Motivo.
Repetido	Nº de veces que ha sido repetida la tarea.

3. ATRIBUTOS DE UN TIPO DE TAREA

Id.....	Número identificador único
Nombre	Nombre o Título a emplear cuando el espacio es suficiente
Repeticiones (Rp)	Número de repeticiones que podrá utilizarse en la cond. finalización
Dependencias_inicio (DI)	Evento o Tarea que podrán emplearse en la cond. de inicio
Dependencias_finalización (DF)	Eventos o Tareas que podrán emplearse en la cond. de finalización
Retardo_inicio	Días desde que se satisfacen las dependencias antes de iniciar
Retardo_finalizacion	Días desde que se satisfacen las dependencias antes de iniciar
Condición_inicio[].....	Condición lógica que determina el comienzo de una tarea
Condición_finalización[]	Condición lógica que determina la finalización de una tarea
Dias[]	Días de la semana en los que se realiza la tarea
Horario	Horario de realización
Postergar	Postergar una repetición si no se realiza un día. Si falso se pierde.
Manual	Si el usuario puede marcar/desmarcar la tarea como realizada.
Diario[].....	Diario de realización de las tareas: Día, Hora, Realizada y Motivo

EJEMPLO N°1 DE TAREA

Para la configuración totalmente personalizada de una tarea se establecen todos sus atributos.

Id..... 00000000234
 Nombre Realización de ejercicios matutinos de estiramiento
 Etiqueta_corta..... Estiramiento Mañana
 Tipo..... 00005
 Descripción..... Consiste en la realización de 3 tandas de estiramiento con 10 dife...
 Fecha_inicio (FI) 15-01-2012
 Fecha_fin(FF) 30-01-2012
 -Repeticiones (R) 5
 Dependencias_inicio (DI) FinTarea_220, FinTarea_221
 Dependencias_finalización (DF) IniTarea_235
 -Retardo_inicio 2
 -Retardo_finalizacion 0
 -Condición_inicio[] FA>=FI y DI
 -Condición_finalización[] FA>FF o DF o Rd>=R
 -Dias[] L, X, V
 -Horario 09:00-11:30
 -Postergar No
 -Manual Sí

Diario[]

Fecha	Hora	Realizado/Motivo
16-01-2012	10:05	Sí
18-01-2012	10:15	Sí
20-01-2012	11:10	No. Olvido
23-01-2012	09:45	Sí

Repetido (Rd) 6

*FA = Fecha Actual

EJEMPLO Nº 2 DE TAREA

Los atributos que comienzan por un guión pueden omitirse, en cuyo caso se toman los valores por defecto para ese tipo de tarea. De este modo la tarea anterior podría quedar así:

TAREA

Id..... 00000000234
Nombre Realización de ejercicios matutinos de estiramiento
Etiqueta_corta..... Estiramiento Mañana
Tipo..... 00005
Descripción..... Consiste en la realización de 3 tandas de estiramiento con 10 dife...
Fecha_inicio (FI) 15-01-2012
Fecha_fin(FF)..... 30-01-2012
Repeticiones (R) 5
Dependencias_inicio (DI) FinTarea_220, FinTarea_221
Dependencias_finalización (DF) IniTarea_235

TIPO DE TAREA

Id..... 00005
Nombre Tareas condición compleja nro.1
Descripción.....
Repeticiones..... 5
Retardo_inicio 2
Retardo_finalizacion 0
Condición_inicio[]..... FA>=FI y DI
Condición_finalización[] FA>FF o DF
Dias[] L, X, V
Horario 09:00-11:30
Postergar No

EJEMPLO N° 3 DE TAREA

También pueden omitirse los atributos que no se necesitan para una tarea dada, haciendo que la definición de una tarea simple sea también simple, como por ejemplo:

TAREA

Id..... 00000000234
Nombre Medición Mañana
Tipo..... 00001
Horario 09:00-11:30

TIPO DE TAREA

Id..... 00001
Nombre Tareas permanentes automáticas
Manual No

- Se observa que en los tipos de tarea también pueden omitirse los valores de los atributos, en cuyo caso se toman los valores globales por defecto, que podrían establecerse como se muestra a continuación.

VALORES GLOBALES POR DEFECTO

Tipo..... 00001
Etiqueta_corta..... Igual que Nombre completo
Descripción..... En blanco
Fecha_inicio (FI) Comienzo del tratamiento
Fecha_fin (FF) Fin del tratamiento
-Repeticiones (R) 0 (ilimitadas)
Dependencias_inicio (DI) Ninguna (Se cumple permanentemente)
Dependencias_finalización (DF) Ninguna (Se cumple permanentemente)
-Retardo_inicio 2
-Retardo_finalizacion 0
-Condición_inicio[] FA>=FI y DI
-Condición_finalización[] FA>FF o DF o Rd>=R
-Dias[] Todos los días de la semana
-Horario No establecido
-Postergar No
-Manual Sí

EJEMPLO N°4 DE TAREA - ESTUDIO DE UNA LECCIÓN

Este podría ser un ejemplo real de tarea “Estudio de Lección”.

Aquí se ve cómo pueden casarse los requerimientos simples establecidos para el funcionamiento inicial de la aplicación, con un modelo más flexible que permita adaptarse a futuros requerimientos, sin convertir la definición de tareas en algo farragoso:

TAREA

Id..... 00000000115
Nombre Lección 5: Los estiramientos en el tratamiento del dolor.
Tipo..... 00002
Dependencias_inicio (DI) FinTarea_99 (Lectura de la Lección 4)

TIPO DE TAREA

Id..... 00002
Nombre Estudio de las lecciones
Descripción..... Estudio de la lección
Repeticiones..... 1
Postergar Sí
Manual No

VALORES GLOBALES POR DEFECTO (Se establecen una sola vez para todas las tareas)

Tipo..... 00001
Etiqueta_corta..... Igual que Nombre completo
Fecha_inicio (FI) Comienzo del tratamiento
Fecha_fin(FF) Fin del tratamiento
-Repeticiones (R) 0 (ilimitadas)
Dependencias_inicio (DI) Ninguna (Se cumple permanentemente)
Dependencias_finalización (DF) Ninguna (Se cumple permanentemente)
-Retardo_inicio..... 2
-Retardo_finalizacion 0
-Condición_inicio[] $FA \geq FI$ y DI
-Condición_finalización[]..... $FA > FF$ o DF o $Rd \geq R$
-Dias[] Todos los días de la semana
-Horario No establecido
-Postergar..... No
-Manual Sí

EJEMPLO N°5 DE TAREA – TAREA ASOCIADA A UNA LECCIÓN

Este sería otro posible ejemplo real de tarea, asociada a la lectura de una lección.

TAREA

Id..... 00000000116
Nombre Estiramiento Matutino
Tipo..... 00006
Descripción..... Consiste en la realización de 3 tandas de estiramiento con 10 dife...
Dependencias_inicio (DI) FinTarea_115

TIPO DE TAREA

Id..... 00006
Nombre Tareas dependientes de Lecciones
Retardo_inicio 1
Horario 09:00-11:30

VALORES GLOBALES POR DEFECTO (Se establecen una sola vez para todas las tareas)

Tipo..... 00001
Etiqueta_corta..... Igual que Nombre completo
Fecha_inicio (FI) Comienzo del tratamiento
Fecha_fin(FF) Fin del tratamiento
-Repeticiones (R) 0 (ilimitadas)
Dependencias_inicio (DI) Ninguna (Se cumple permanentemente)
Dependencias_finalización (DF) Ninguna (Se cumple permanentemente)
-Retardo_inicio 2
-Retardo_finalizacion 0
-Condición_inicio[] $FA \geq FI$ y DI
-Condición_finalización[] $FA > FF$ o DF o $Rd \geq R$
-Dias[] Todos los días de la semana
-Horario No establecido
-Postergar No
-Manual Sí

Anexo III:

Instalación del Entorno

-Resumen de Instrucciones-

1. Instalar MsysGit 1.7.7.1.
 - a. Descargar
 - i. Ir a <http://code.google.com/p/msysgit>
 - ii. Clic pestaña Downloads
 - iii. Clic Git-1.7.7.1-preview20111027.exe
 - b. Instalar
 - c. Crear una Key SSH como se describe en:
<http://help.github.com/win-set-up-git/>
Si ya se dispone de ella, buscarla en el directorio .ssh
2. Crear cuenta en GitHub.
 - a. Acceder a <https://github.com>
 - b. Crear cuenta
 - i. Clic Signup and Pricing
 - ii. Clic Create a free account
 - iii. Introducir Username, Email Address y Password
 - iv. Clic sobre Create an Account
 - c. Registrar Key SSH
 - i. Clic Account Setting
 - ii. Clic SSH Public Keys~
 - iii. Clic Add another public key
 - iv. Introducir Key (y opcionalmente título)
 - v. Clic Add Key
3. Instalar Eclipse SDK 3.7 (Índigo)
 - a. Descargar
 - b. Descomprimir donde se desee
 - c. Inicial
 - d. Traducir a Español:
 - i. Clic Ayuda -> Install New Software
 - ii. Work With:
<http://download.eclipse.org/technology/babel/update-site/R0.9.0/indigo>
 - iii. Pulsar Enter
 - iv. En la lista que se genera (durante un rato aparece Pending...), marcar la casilla de verificación junto a Babel Language Packs for eclipse
 - v. Clic Next
 - vi. En la lista que se genera, marcar la casilla de verificación junto a: Babel Language Packs for eclipse in Spanish
 - vii. Next -> Aceptar terminus -> Finalizar
 - viii. Reiniciar
4. Instalar Android SDK
 - a. Descargar de <http://developer.android.com/sdk/index.html>
 - b. Instalar donde se desee. p.e. C:\Users\usuario\android-sdks
 - c. Instalar Paquetes
 - i. Ejecutar Android SDK Manager
 - ii. Seleccionar paquetes marcando casillas

- iii. Clic en Install Packets
 - d. Crear un dispositivo Virtual Android
 - i. Ventana -> Android Virtual Device Manager -> New
 - ii. Completar Name, Target
 - iii. Clic Create AVD
 - e. Instalar plugins ADT en Eclipse
 - i. Ayuda->Install New Software -> Add...
 - ii. En el campo nombre ingresa "Android Developer Tools" sin las comillas.
 - iii. En la URL escribe <https://dl-ssl.google.com/android/eclipse/>
 - iv. Clic Aceptar
 - v. En la lista, selecciona la casilla del único ítem: Developer Tools
 - vi. Siguiente -> Aceptar Licencia -> Finish -> Reiniciar
 - vii. Alternativamente instalar plugin desde ADT-15.0.0.zip.
 - f. Configurar plugin ADT automáticamente (tras reinicio)
 - i. Seleccionar la segunda opción: indicar directorio donde está instalado.
- 5. Instalar PlugIn Aptana Studio 3.0.6 para Eclipse.
 - a. Clic Ayuda -> Install New Software
 - b. Work With:
 - <http://download.aptana.com/studio3/plugin/install>
 - c. En la lista que se genera, marcar la casilla de verificación junto a:
 - Aptana Studio 3
 - d. Clic Siguiente -> Siguiente
 - e. Accept the terms
 - f. Clic Finish
 - g. Reiniciar
 - h. Copiar sencha-touch-rc1.sdocml en cualquier lugar de cada proyecto
- 6. Instalar Plugins WAC Widget para Eclipse (WAC2.0_update.zip)
 - a. Descargar
 - b. Clic Ayuda -> Install New Software -> Añadir -> Archive
 - c. Localiza y selecciona WAC2.0_update.zip
 - d. Clic Aceptar
 - e. Marca WAC 2.0
 - f. Siguiente
 - g. Marca Accept terms
 - h. Clic Finalizar
 - i. Reiniciar Eclipse
 - j. Copia los archivos de Borqs\Developer_Tools\SDK\OpenSSL\windows a A la subcarpeta tools de la carpeta donde hemos descomprimido Android SDK
 - k. Instalar BorqsWRTInstaller.pak en el AVD
 - i. Lanzar el emulador.
 - 1. Ir a C:\Users\usuario\android-sdks\tools
 - 2. Ejecutar emulator -avd MiAVD
 - ii. Ir a C:\Users\usuario\android-sdks\platform-tools
 - iii. Copiar allí BorqsWRTInstaller.apk
 - iv. Desde allí: adb install BorqsWRTInstaller.apk
- 7. Importar repositorio fibroline desde GitHub
 - a. Conseguir permiso de escritura en repositorio GitHub
 - b. Archivo -> Importar -> Git Repository As New Project
 - c. URI: <git@github.com:algos/fibroline.git>
 - d. Destination: C:\Users\usuario\workspace\fibroline

***Anexo II:
Historias de Usuario
-versión completa-***

Priority	Estimation	Topic	Title	Description	Developer
100	10	Sistema	Navegación principal	COMO product owner QUIERO implementar un módulo "core" o "backbone" que sostenga a los otros módulos del sistema PARA que las historias de usuario se puedan implementar y conetar con este de forma independiente	Roman
100	10	Sistema	Cola de Seguimiento matinal y nocturno	COMO desarrollador QUIERO que haya un Wizard con una cola de acciones, y cada acción active una vista PARA poder implementar el seguimiento nocturno y diurno	Roman
94	3	Sistema	Definir modelo conceptual (en json) para las lecciones	Como product owner QUIERO tener definido un modelo conceptual sencillo PARA definir las lecciones.	Roman
90	3	Sistema	Definir programaticamente las lecciones	COMO productowner, QUIERO tener el modelo programatico con las lecciones, en JSON PARA ????	Roman
100	3	Lecciones	Insertar leccion	Como terapeuta QUIERO que el sistema me permita insertar en un fichero independiente de la aplicación la definición de las lecciones PARA que pueda cargar las lecciones que yo defina sin entrar en el sistema.	Roman
100	3	Lecciones	Visualizar contenido de la lección	COMO usuario QUIERO que antes de empezar la lección se me informe del tiempo estiamdo para completarla PARA darme la oprotunidad de decidir cuando lo hago	Roman
100	3	Lecciones	Visualizar lecciones	COMO usuario QUIERO ver la lista de lecciones desbloqueadas por el tratamiento, con el nombre de la lección ordenadas por la fecha en que han ido apareciendo, y si esa lección es nueva que aparezca destacada (como mucho hay sólo una lección nueva) PARA poder acceder a ellas	Roman
100	3	Sistema	Registrar acciones	COMO product owner QUIERO tener un sistema de registro de acciones (log) en la aplicación donde cada entrada tenga: fecha-hora, tipo de acción, datos asociados a la acción PARA registrar determinadas acciones y avisar de cambios en el estado a los componentes que escuchen	Roman

	38				
100	4	ContenidoLeccion	Producir Lección Tratamiento	COMO Terapeuta QUIERO producir una lección sobre el tratamiento PARA incluirla en el tratamiento	Rocio
100	4	ContenidoLeccion	Producir Lección Dolor	COMO Terapeuta QUIERO producir una lección sobre el Dolor PARA incluirla en el tratamiento	Rocio
90	3	ContenidoLeccion	Producir Lección Medicación	COMO Terapeuta QUIERO producir una lección sobre la Medicación PARA incluirla en el tratamiento	Rocio
80	8	ContenidoLeccion	Producir Lección Ansiedad	COMO Terapeuta QUIERO producir una lección sobre la Ansiedad PARA incluirla en el tratamiento	Rocio
78	5	ContenidoLeccion	Producir Lección Sueño	COMO Terapeuta QUIERO producir una lección sobre el Sueño PARA incluirla en el tratamiento	Rocio
70	4	ContenidoLeccion	Producir Lección Actividad	COMO Terapeuta QUIERO producir una lección sobre las actividades PARA incluirla en el tratamiento	Rocio
100	3	ContenidoLeccion	Producir Lección prevención de recaídas	COMO Terapeuta QUIERO producir una lección sobre prevencion de recaídas PARA incluirla en el tratamiento	Rocio
100	3	Lecciones	Preparar Entorno de servidor para las lecciones	COMO productowner QUIERO Preparar el entorno en el servidor de psico para tener un github, una estructura de lección, PARA meter las lecciones	Roman
	31				
90	3	Sistema	Definir programaticamente el tratamiento	Como productowner quiero tener definido un modelo conceptual sencillo para definir el tratamiento	Roman
99	3	Sistema	Funcionamiento offline	Como terapeuta quiero que la aplicacion funcione offline para que el usuario pueda usarla sin necesidad de internet	Roman
90	4	Sistema	sincronización con el servidor		Roman
90	3	Sistema	Diseñar la pagina inicial	COMO usuario QUIERO que la pantalla inicial sea una pantalla de navegación rápida a los contenidos, que me apunte lo que debo hacer en cada estado PARA poder acceder de forma rapida	Roman

80	5	Sistema	Preparar entorno optimización	Como desarrollador, QUIERO tener un sistema automático que compile el código en desarrollo y me devuelva un código optimizado	Roman
60	3	Sistema	Preparar entorno de testing	COMO desarrollador, QUIERO que el sistema haga el test de las historias de usuario con el JAZMINE	Roman
10	14	Sistema	Definir modelo conceptual (ontología) para la programación de la tarea	Como productowner QUIERO definir un lenguaje estructurado, tan simple como estrictamente se pueda PARA que me permita definir un tratamiento con tareas	Roman
100	3	Medición del Dolor	medir con escala numérica	COMO usuario QUIERO registrar la intensidad del dolor con una escala de 0 a 10 y guardar un registro de la medición con el valor, la fecha y la hora en que lo he medido PARA observar su evolución.	Andrés
90	3	Medición del Dolor	Bloquear la aplicación hasta que no se mida el dolor	COMO usuario QUIERE QUE si se activa la franja de medir el dolor suene una alarma en el móvil cada cierto tiempo y la aplicación muestre exclusivamente la escala de dolor y no pueda realizar nada más en la aplicación que medirlo, hasta que finalice la franja o introduzca los datos, PARA QUE no se me olvide medirlo.	Andrés
75	3	Medición del Dolor	Configurar horas	COMO usuario, quiero configurar la hora en la que mediré la intensidad de dolor por la mañana y la hora para medir la intensidad de dolor por noche, PARA que salten avisos en la aplicación y no permita hacer nada más durante dos horas mientras no haya realizado la medición.	Andrés
50	4	Medición del Dolor	medir con escala FPS-R	COMO usuario QUIERO registrar la intensidad del dolor con una escala de FPS-R y guarda un registro de la medición con el valor, la fecha y la hora en que lo he medido PARA observar su evolución.	Andrés
20	2	Medición del Dolor	Configurar la escala de dolor	COMO usuario, QUIERO elegir la escala de dolor que quiero usar PARA que me sea mas agradable constestarla o por consejo de un especialista	Andrés

100	10	Hábitos diarios	Carga de la lista de tareas	<p>COMO terapeuta QUIERO que el sistema tenga cargada una lista de tareas, donde las tareas tienen una descripción, pertenecen a una categoría, pueden llevar una acción de usuario asociada y se inician a partir de (o bien del inicio del tratamiento o bien que una tarea haya finalizado) agregándole un número de días (que pueden ser 0), además pueden ser repetitivas o de una sola realización, y si son repetitivas tienen los días de la semana en que se van a producir (lunes, martes y viernes por ejemplo), y pueden ser de duración fijada en una fecha o para siempre PARA que se visualicen en la lista de tareas "para hoy" cuando toquen y se pueda trabajar con ellas</p>	Raul
100	3	Hábitos diarios	Ver tareas diarias	<p>COMO usuario QUIERO ver la lista de tareas que debo realizar hoy y que se puedan distinguir aquellas que ya he completado de aquellas que me quedan pendientes, y además distinguir aquellas que se marcan como completadas de forma automática cuando realice una acción en el sistema, de aquellas que debo marcar como completadas de forma manual PARA poder completarlas cuando quiera a lo largo del día.</p>	Raúl
100	1	Hábitos diarios	Marcar manualmente la tarea realizada	<p>COMO usuario QUIERO marcar una tarea no monitorizada como completada cuando quiera a lo largo del día PARA avisar al sistema que la he completado y ser consciente de lo que he hecho y tengo pendiente</p>	Raúl
70	2	Hábitos diarios	Información matinal	<p>COMO usuario QUIERO que cada día la primera vez que acceda al ToDo se me presente un cuestionario sobre las actividades del día anterior, donde contestaré por cada tarea todos aquellos datos que me haya dejado del día anterior</p>	Raúl
40	2	Hábitos diarios	Marcar automáticamente la tarea realizada	<p>COMO usuario QUIERO que el sistema marque por mí aquellas tareas completadas, que se realizan a través del propio sistema PARA que quede constancia de ello</p>	Raúl

30	2	Hábitos diarios	Explicación de incumplimiento	COMO usuari QUIERO poder elegir entre varias categorías el motivo por el cual no he cumplido una tarea PARA monitorizar mi conducta	Raúl
28	4	Hábitos diarios	Explicación cumplimiento de hábito	COMO usuario QUIERO poder añadir una descripción a una tarea cumplida. Por ejemplo la actividad física que he realizado y los minutos. PARA poder observar su evolución, trazar líneas base, etc.	Raúl
40	3	charts	Realizar gráfico de intensidades	COMO usuario QUIERO visualizar en un gráfico un serie temporal sobre la intensidad de 0 a 10 de las variables: intensidad de dolor diurno, intensidad del dolor nocturno, nivel de ansiedad y calidad del sueño. Además queremos ocultar y desocultar variables PARA poder observar su evolución	Roman
20	5	charts	Realizar gráfico de tareas	COMO usuario QUIERO visualizar un gráfico para las categorías de las actividades PARA poder observar su evolución.	Roman
100	3	Ansiedad	Practicar relajación	COMO usuario, QUIERO poder escuchar un audio (en el que podré controlar el flujo) PARA relajarme	Roman
70	3	Ansiedad	Registrar relajación en seguimiento	COMO usuario, QUIERO que en el seguimiento de la mañana se me pregunte si he realizado o no una relajación el día anterior, en el caso de que no la haya registrado ya, y si es afirmativo, entonces se me permita evaluar su utilidad en una escala de 0 a 10 PARA observar si es útil	Roman
20	2	Ansiedad	Preguntar utilidad relajación	COMO usuario, QUIERO que al marcar una tarea de relajación como realizada se me pregunta la utilidad con una escala de 0 a 10 PARA que no me la pregunte en el siguiente seguimiento	Roman

211
52,75
13,1875



80

??

Sistema

Seguimiento

COMO productowner,
QUIERO que si se cierra la aplicación estando en un seguimiento, entonces cuando se vuelva a abrir, si no han pasado dos horas, o se vuelve donde estaba antes del seguimiento si han pasado mas de dos horas

Sistema

Bloquear acceso de navegadores no móviles

COMO productowner,
QUIERO que si un cliente web se conecta a mozilla o IE se pare la aplicación y salga un mensaje informático PARA que no puedan acceder a la aplicación

Sistema

Gestor de actualizaciones

COMO usuario QUIERO que la App busque versiones nuevas cada vez que se conecte a internet PARA que se carguen las modificaciones aplicadas a la misma

Sistema

Cambio dinámico de css

COMO productowner,
QUIERO que la aplicación pueda cargar automáticamente un css nuevo
PARA QUE al cargar una nueva lección, pueda incorporar las imágenes y estilos relacionados con el paquete nuevo.

Bibliografía

- *Apuntes ofrecidos por el tutor en el aula virtual.*
- *Bibliografía señalada en los apuntes ofrecidos por el tutor en el aula virtual.*
- *Kanban y Scrum, obteniendo lo mejor de ambos.*
Henrik Kniberg & Mattias Skarin, 2010 C4Media Inc.
- *Presentación jQueryTouch, Mobile Web Apps with HTML, CSS & JavaScript.*
Philipp Bosch
- *Documentación oficial de WAC.*
- *Presentación sobre Sencha Touch*
James Pearse
- *Documentación Oficial de Sencha Touch.*
- *Sencha Touch In Action.*
Jesus Garcia y Anthony De Moss – Manning Publications
- *Build mobile web applications with Sencha Touch*
Brice Mason – IBM. Copyright developerWorks
- *Sencha Touch, HTML and JavaScript Mobile Web App Framework*
Akshay Prabhu
- *Presentación “Sencha Touch”*
Juan Camilo Pérez, Daniel Vargas y Daniel Veintimilla.
- *Structuring your Sencha Touch application.*
Tommy Maintz
- *Creating a Sencha Touch MVC application from scratch.*
By CAM
- *Introduction to Object-Oriented JavaScript*
Fernando Trasviña
- *Building a Sencha Touch Application.*
Jorge Ramon
- *Documentación oficial de Git.*
- *Pro Git*
Scott Chacon
- *Documentación oficial de Aptana Studio.*