



UNIVERSIDAD DE VALLADOLID

E.T.S. DE INGENIERÍA INFORMÁTICA

Ingeniería Técnica en Informática de Gestión

**APLICACIÓN COMPLEMENTARIA A LA
INICIATIVA DE LA AECC DIARIO DE UN
SUPERVIVIENTE**

Autores:

Fernando Santa Olaya Rodríguez
Rubén Toquero González

Tutor:

Fernando de Prada Moraga

Agradecimientos

A mis padres Fernando y Filomena que me dieron a mí y a mis hermanos todo lo que ellos no tuvieron y por lo que se desvivieron trabajando, sin ellos no sería quien soy. Y a mis hermanos y hermana, David, Marta y Jose Ignacio, porque sin ellos mi vida hubiera sido muchísimo más aburrida y tampoco sería el hombre que hoy soy si ellos no hubiesen estado a mi lado.

También quiero agradecer a nuestro tutor Fernando de Prada Moraga por toda la paciencia que ha tenido con nosotros, la comprensión y la ayuda que nos ha brindado tantas veces con este y otros asuntos durante la etapa estudiantil.

A la familia más extensa, tíos, tías, primos etc por apoyarme y animarme cada vez que nos juntábamos.

A mis amigos y compañeros de clase, la familia que eliges, por todos los sin sabores que me han acompañado a transitar en estos años y no mandarme muy lejos casi nunca.

A Rubén por ser mi motor en esta última etapa y forzarme a empujar un poco más cada día y robar tiempo de donde no lo había.

Por último a todos aquellos profesores que me ayudaron a llegar aquí, desde EGB hasta este último proyecto, por saber despertar la curiosidad y ansia de saber y por su dedicación a mi formación no sólo como ingeniero sino también como persona.

Fernando Santa Olaya Rodríguez

Quiero agradecer a mi familia todo el apoyo que me han dado durante todos estos años, a los que están y a los que ya no están, a los que dedico este trabajo por razones obvias.

También quiero agradecer al tutor Fernando de Prada Moraga toda la paciencia que ha tenido con nosotros, nos ha comprendido y nos ha ayudado a llevar a cabo esta necesaria tarea.

A mi novia, que es la que me ha sufrido la mayor parte del tiempo, todavía me resulta incomprensible que me siga aguantando después de tantos años.

A Fer que también me ha aguantado como un jabato y ha tenido paciencia y comprensión.

Por último agradezco a los profesores de esta escuela sus enseñanzas (espero que bien aprovechadas por mi parte) y a todos aquellos con los que he compartido mi tiempo en mi paso por esta etapa de mi vida que aquí se cierra.

Rubén Toquero González

Resumen

El presente proyecto implementa una aplicación móvil como micro asistente virtual a personas que padezcan la lacra moderna del cáncer y una aplicación en servidor con la que se comunica la app y recopila estadísticas de actividades y estados de animo de los usuarios para su posibles estudios relacionados con la enfermedad. Además provee de una plataforma con la que se puede interactuar fácilmente con los usuarios de la misma por un operador de la misma. Todo esto se incluye dentro de la iniciativa "Diario de un superviviente" de la Asociación Española contra el Cáncer.



Figura 1: Portada del folleto.

Abstract

This project implements a mobile application as micro virtual assistant to people suffering from the modern scourge of cancer and an application server with the app communicates and gathers statistics activities and moods of users for possible studies related to the disease. Also provides a platform that can easily interact with users of the same for the same operator. All this is included in the "Diary of a Survivor" initiative of the Spanish Association Against Cancer. This project implements a mobile application as micro virtual assistant to people suffering from the modern scourge of cancer and an application server with the app communicates and gathers statistics activities and moods of users for possible studies related to the disease. Also provides a platform that can easily interact with users of the same for the same operator. All this is included in the "Diary of a Survivor" initiative of the Spanish Association Against Cancer.[3]

Índice general

Índice de figuras	11
Índice de cuadros	13
1. Introducción	15
1.1. Antecedentes y Motivación del proyecto	15
1.2. Ámbito de trabajo	16
1.3. Objetivos del proyecto	17
1.4. Estructura del proyecto	17
1.5. Estructura de la memoria	17
2. Visión general del Proyecto	19
2.1. Estado del arte de la movilidad	19
2.1.1. Smartphones	19
2.1.2. Sistemas operativos Móviles	20
2.2. ¿Porqué Android?	21
2.3. Parte App móvil	21
2.4. Parte Servidora	22
3. Planificación y Metodología	25
3.1. Plan de desarrollo de software	25
3.2. Propósito general de la planificación	25
3.3. Metodología	25
3.3.1. SCRUM	26
3.3.2. Roles y responsabilidades	27
3.3.3. Eventos y rituales	28
3.4. Planificación completa	30
3.5. Recursos necesarios	30
4. Análisis	33
4.1. ¿El porqué?	33
4.2. Explicación detallada de las funcionalidades de la aplicación	34
4.3. Documento de Análisis	36
4.3.1. Descripción de objetivos de manera detallada	36
4.3.2. Captura de Requisitos	38
4.3.3. Identificación de actores	41
4.3.4. Diagrama de casos de uso	41
4.3.5. Descripción de Casos de Uso	41
4.3.6. Diagrama de clases de análisis	44
4.3.7. Diagrama Entidad-Relación de la base de datos	44

4.3.8. Modelo relacional del análisis	44
5. Diseño	45
5.1. Parte App móvil	45
5.2. Arquitectura	45
5.3. Tecnología	46
5.4. Diagrama de Clases	46
5.5. Diseño de la base de datos	46
5.6. Diagramas de estado	46
5.7. Prototipado	46
5.7.1. Material Design	46
5.7.2. Pantallas Principales	46
5.8. Flujo de navegación	48
6. Implementación y Pruebas	53
6.1. Construcción y Pruebas	53
6.2. Estructura de la aplicación	53
6.3. Plan de desarrollo	53
6.4. SVN, GIT,	55
6.5. Plan de trabajo y comunicaciones	55
6.6. Pruebas	55
6.6.1. Plan de pruebas	55
6.6.2. Tipos de pruebas	55
6.6.3. Baterías de pruebas	55
6.6.4. Pruebas en el dispositivo	55
6.7. Puesta en producción	55
7. Conclusiones y trabajo futuro	57
7.1. Conclusiones y trabajo futuro	57
7.2. Conclusiones	57
7.3. Dificultades encontradas	57
7.4. Consecución de Objetivos	57
7.5. Conocimientos adquiridos	57
7.6. Trabajo futuro	57
8. Bibliografía	59
Bibliografía	61
8.1. Conclusiones y trabajo futuro	63
8.2. Conclusiones	63
8.3. Dificultades encontradas	63
8.4. Consecución de Objetivos	63
8.5. Conocimientos adquiridos	63
8.6. Trabajo futuro	63

Índice de figuras

1.	Portada del folleto.	5
3.1.	Resumen de Scrum.	28
4.1.	Paradigma captura de requisitos método tradicional vs. ágil	36
5.1.	Drawer menu y Pantalla principal	47
5.2.	Pantallas con los horarios y ocupación del paciente	47
5.3.	Pantalla con la interfaz para las rutinas	48
5.4.	Pantalla con la actividad de las citas	49
5.5.	Pantalla con la actividad de la medicacion	49
5.6.	Pantalla con la actividad de los personajes	50
5.7.	Pantalla con la actividad de las pruebas	50
5.8.	Pantalla con la actividad de los sintomas	51
5.9.	Pantalla con la actividad de los recursos	51
5.10.	Pantalla con la actividad conf del perfil	51
5.11.	Pantalla con la actividad de las citas	52

Índice de cuadros

4.1. Menú drawer.	37
4.2. Menú drawer.	38
4.3. Creación del perfil - Inserción	41
4.4. Creación del perfil - Inserción	42
4.5. Creación del perfil - Inserción	42

CAPÍTULO 1

Introducción

1.1. Antecedentes y Motivación del proyecto

El cáncer es una enfermedad provocada por un grupo de células que se multiplican sin control y de manera autónoma, invadiendo localmente y a distancia otros tejidos. El cáncer es una de las principales causas de mortalidad en todo el mundo; en 2012 hubo unos 14 millones de nuevos casos y 8,2 millones de muertes relacionadas con el cáncer.

Se prevé que el número de nuevos casos aumente en aproximadamente un 70 % en los próximos 20 años. En 2012, los cánceres diagnosticados con más frecuencia en el hombre fueron los de pulmón, próstata, colon y recto, estómago e hígado, en la mujer fueron los de mama, colon y recto, pulmón, cuello uterino y estómago.

Aproximadamente un 30 % de las muertes por cáncer son debidas a cinco factores de riesgo conductuales y dietéticos: índice de masa corporal elevado, ingesta reducida de frutas y verduras, falta de actividad física, consumo de tabaco y consumo de alcohol.

«Cáncer» es un término genérico que designa un amplio grupo de enfermedades que pueden afectar a cualquier parte del organismo; también se habla de «tumores malignos» o «neoplasias malignas». Una característica del cáncer es la multiplicación rápida de células anormales que se extienden más allá de sus límites habituales y pueden invadir partes adyacentes del cuerpo o propagarse a otros órganos, proceso conocido como metástasis. Las metástasis son la principal causa de muerte por cáncer.

¿Cuál es la causa del cáncer?

El cáncer comienza en una célula en la que se produce la transformación de una célula normal en tumoral.

El envejecimiento es otro factor fundamental en la aparición del cáncer. La incidencia de esta enfermedad aumenta muchísimo con la edad, muy probablemente porque se van acumulando factores de riesgo de determinados tipos de cáncer. La acumulación general de factores de riesgo se combina con la tendencia que tienen los mecanismos de reparación celular a perder eficacia con la edad.

Factores de riesgo del cáncer El consumo de tabaco y alcohol, la dieta malsana y la inactividad física son los principales factores de riesgo de cáncer en todo el mundo. Algunas infecciones crónicas también constituyen factores de riesgo, y son más importantes en los países de ingresos medios y bajos.

En resumidas cuentas, el cáncer es una gran lacra en nuestros días [1] y uno de los aspectos importantes en el tratamiento de este grupo de enfermedades, es la buena disposición y estado mental positivo del enfermo. No cura la enfermedad pero ayuda a superar los duros tratamientos a los que se ven sometidos los enfermos.

Por otro lado, de un tiempo a esta parte las aplicaciones móviles o apps forman parte de nuestros smartphones y por tanto de nuestras vidas. Son tan comunes que para cada ámbito o actividad existe una app que apoya, ayuda, informa o al menos lo pretende con mayor o menor éxito.

Dentro de las apps relacionadas con la salud ninguna o muy pocas de esas tratan sobre esta enfermedad o grupo de enfermedades. Debido a que ambos fuimos tocados de cerca por esta enfermedad y que a la Asociación Española contra el Cáncer le surgió la necesidad decidimos apoyar de la mejor manera que podemos que es escuchando sus necesidades e implementándolas en una app para el sistema operativo móvil Android y aconsejándoles las mejores alternativas en ese sentido.

Mediante el desarrollo de esta aplicación pretendemos hacer más llevaderos los procesos que ocurren a continuación de pasar por el postraumático tratamiento y cicatrices visibles o no que deja esta enfermedad y devolver la normalidad a esos pacientes que quieren continuar de una manera normal con sus vidas. Haciéndoles más llevaderas algunas de las tareas de obligado cumplimiento 'posterior' que deben realizar estos verdaderos luchadores y supervivientes.

Sin ánimo de ser victimistas, nosotros mismos conocemos de primera mano todo el sufrimiento que se genera, la angustia y el dolor que causa esta lacra y una de las principales causas de muerte en el mundo.

Esta aplicación surge como complemento digital a un dossier/iniciativa de la AECC que pretende servir al enfermo como centro de datos y recolección de información que es el Diario de un superviviente, en ese sentido esta app pretende ofrecer todo lo que permite el dossier adaptado al mundo digital y aportar algún valor, como pudieran ser filtros sobre los datos, alarmas y notificaciones en las citas, y como modo más experimental el contacto directo con el agente de la propia AECC como si de una app de mensajería instantánea se tratase.

1.2. Ámbito de trabajo

En el desarrollo de este proyecto hay dos ámbitos de trabajo, cercanos, pero ligeramente diferenciados.

El primero de ellos es el de una app de lo que podríamos usuario final, para el uso que el propio usuario crea conveniente dentro de las capacidades de la misma. Es por eso que la usabilidad, el diseño, los tutoriales, ayudas etc deben ir en línea con que el espectro de usuarios potenciales es muy amplio, y que además están atravesando un trance complicado.

El segundo ámbito de trabajo se enmarca en la parte servidora, por usuarios de la misma, aunque en este caso podemos presuponer cierto entrenamiento y hábito en el uso de herramientas web, además de que desde la propia AECC se formará a quienes estén manejando esta web y que hemos denominado agentes.

En resumen el uso y ámbito de trabajo donde general es el de uso publico por usuarios no categorizados ni expertos por un lado, y usuario final con entrenamiento y conocimiento por otro.

1.3. Objetivos del proyecto

El objetivo principal del proyecto es servir de apoyo a las acciones emprendidas por la AECC en el marco de la nueva estrategia de identidad digital mediante aplicaciones útiles para cada una de sus iniciativas y las de apoyo general e las acciones de la asociación, más concreto en este primer paso con la iniciativa de Diario de un Superviviente.

Por tanto uno de los objetivos del proyecto por tanto será el desarrollo de una app móvil denominada "Diario de un Superviviente" que sirva a los intereses de la AECC en la línea de apps en los stores y presencia corporativa e imagen digital. Es por esto que no solamente será el objetivo de este proyecto la realización de la app, sino que la ayuda a la campaña en la web mediante banners, y desarrollo de la propia sección en la Play store, quedando para futuras evoluciones el desarrollo y la inclusión en la tienda del sistema operativo iOS.

En este sentido el otro gran objetivo es desarrollar una parte servidora, en forma de API REST, para realizar sincronizaciones entre dispositivos, recogida de estadísticas, seguimiento por parte del agente asignado, estos últimos puntos además presentados en una web para dicho agente aprovechando la API antes mencionada, quedando también para futuras líneas la implementación de apps que sirvan a los agentes en las tareas que ahora realizan con la web.

1.4. Estructura del proyecto

El proyecto se estructurará en tres partes bien diferenciadas la app móvil, la parte servidora y por último toda la documentación, tanto de esta memoria como los manuales de instalación y usuario del resto de partes.

1.5. Estructura de la memoria

Explicación breve del contenido de cada capítulo:

Capítulo 1 contexto general en el que se desenvuelve la aplicación, explicando el problema existente y la solución propuesta para dicho problema. Termina mostrando la estructura del presente documento.

Capítulo 2 breve explicación de las tecnologías seleccionadas resumen de la plataforma elegida, presentación general del proyecto y justificación del mismo.

Capítulo 3 desarrollo de la aplicación, planificación, plazos, técnicas utilizadas para llevar a cabo las funcionalidades de la aplicación.

Capítulo 4 fase de análisis donde podrá encontrar aspectos clave para el buen desarrollo del producto como una introducción a la aplicación y los estudios sobre la arquitectura

utilizada, casos de uso y requisitos, diagrama de clases, entidad-relación.

Capítulo 5 diseño, determina el comportamiento esperado de la aplicación mediante una serie de diagramas de secuencia y un prototipado de bajo nivel.

Capítulo 6 construcción de la aplicación, como se ha conseguido llevar a cabo la implementación de las funcionalidades de la aplicación. Además incluye las pruebas realizadas para determinar el buen funcionamiento de la misma.

Capítulo 7 presenta las conclusiones obtenidas una vez finalizado el desarrollo del proyecto, y una serie de trabajos futuros que podrían desarrollarse como mejoras de la aplicación.

Capítulo 8 bibliografía y referencias web, que incluyen toda documentación consultada para la elaboración del proyecto.

Anexo I contiene el manual de usuario, así como el manual de instalación para aquellos usuarios que no estén tan familiarizados con la plataforma Android.

CAPÍTULO 2

Visión general del Proyecto

El concepto de aplicación es un asistente-diario para un enfermo de cáncer en la parte de la app móvil y una parte servidora que se encarga de sincronizar datos entre diferentes dispositivo, una base datos para dar apoyo a esto y que ademas reciba datos estadísticos de la app, por si alguna vez pueden extraer conclusiones tras un tratamiento estadístico de los mismos. Además se incluye un servicio de comunicación con los usuarios mediante notificaciones push a los dispositivos

2.1. Estado del arte de la movilidad

Dentro del concepto Movilidad se agrupan, tanto el hardware como el software de pequeños dispositivos portables o dispositivos móviles. Podríamos definir a estos últimos como “aparatos de pequeño tamaño, con algunas capacidades de procesamiento, con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.”

De entre los elementos que pueden llegar a ser un dispositivo móvil (una PDA, un teléfono móvil, un lector de libros electrónico o un ordenador portátil) destaca, sin lugar a dudas, el teléfono móvil como el dispositivo más utilizado de entre todos. Y entre los teléfonos móviles destacan los denominados “SmartPhones y Tabletas” por ser tener unas capacidades y potencia de ejecución de aplicaciones complejas de consumo de lo más variadas, desde reproductores de todo tipo de media, hasta mensajería instantáneas, llamadas apps. Es el desarrollo de estas apps y su comercialización por parte de grandes compañías tecnológicas como Google, Apple y Microsoft lo que ha devenido en todo un nuevo mercado de desarrollo de software de todo tipo, no solo para grandes compañías sino también para pequeños grupos de desarrollo o incluso de desarrolladores indies.

2.1.1. Smartphones

Un Smartphone (cuya traducción sería “teléfono inteligente”) “es una evolución del teléfono móvil tradicional que cuenta con ciertas características y prestaciones que lo acercan más a un ordenador personal que a un teléfono tradicional.” Entre dichas prestaciones y características, se encuentran una amplia mejora del almacenamiento de datos, conexión a Internet mediante una tarifa contratada o haciendo uso de redes WIFI, acelerómetro, pantalla táctil, teclado QWERTY...y un sinnfin de aplicaciones de usuario, además de la posibilidad de descarga de nuevas aplicaciones.

Todas estas prestaciones y características de los Smartphone estarían desaprovechadas sin

software que las saque partido. Por ello los Smartphone llevan un SO (Sistema Operativo) que les permite realizar todas estas tareas de una forma rápida y sencilla, además de optimizando el consumo energético ya que dependen de una batería para ello la mayoría de las veces. En el siguiente apartado, realizaremos una breve descripción de los SO para dispositivos móviles y Smartphone más importantes que se encuentran en el mercado a día de hoy.

2.1.2. Sistemas operativos Móviles

Partiendo de una definición de sistema operativo: “Capa compleja entre el hardware y el usuario, concebible también como una máquina virtual, que facilita al usuario o al programador las herramientas e interfaces adecuadas para realizar sus tareas informáticas, abstrayéndole de los complicados procesos necesarios para llevarlas a cabo.”

Por ende, un sistema operativo móvil es un sistema operativo que controla un dispositivo móvil. Sin embargo, estos son mucho muy diferentes a los tradicionales SO de equipos de escritorio, para empezar hay mucha mas variedad, pero es que la interacción con el usuario, la gestión de la memoria, las tareas de llamadas que deben ser prioritarias, etc

El sistema operativo destinado a controlar un dispositivo móvil necesita ser fiable y tener una gran estabilidad, ya que incidencias habituales y toleradas en ordenadores personales como reinicios o caídas no tienen cabida en un dispositivo de estas características. Además, a de adaptarse adecuadamente a las consabidas limitaciones de memoria y procesamiento de datos, proporcionando una ejecución exacta y excepcionalmente rápida al usuario.

Estos sistemas han de estar perfectamente testeados y libres de errores antes de incorporarse definitivamente a la línea de producción. Las posibilidades que existen en un ordenador estándar de realizar actualizaciones e incluso reinstalar mejores versiones del sistema para cubrir fallos o deficiencias son más limitadas en un dispositivo móvil. Es posible incluso que un aparato de esta naturaleza deba estar funcionando ininterrumpidamente durante semanas e incluso meses antes de ser apagado y reiniciado, a diferencia de lo que ocurre con un ordenador personal. El consumo de energía es otro tema muy delicado: es importante que el sistema operativo haga un uso lo más racional y provechoso posible de la batería, ya que esta es limitada y el usuario siempre exige una mayor autonomía.

En la actualidad, existen varios sistemas operativos para toda la gama de dispositivos móviles. Más adelante veremos por qué se ha elegido Android para la realización de este proyecto fin de carrera, pero antes veamos las características más importantes de los principales sistemas operativos móviles.

En 2015 en España y el mundo dentro de este mercado de SO de la movilidad se pueden encontrar diferentes actores que proveen de lo que denominamos sistemas operativos móviles, existiendo desde los provistos por las antes mencionadas grandes compañías tecnológicas hasta aquellos provistos por fundaciones, empresas que han perdido gran parte del negocio o empresas compradas por alguna de las grandes compañías, un repaso somero de los distintos SO móviles que podemos encontrar hoy día nos arroja nombres como Android, iOS, Windows Phone, Balckberry, Symbian, Firefox OS, Ubuntu Touch, etc

2.2. ¿Porqué Android?

Dentro del mercado de SO móviles solamente Android e iOS pueden considerarse grandes actores propiamente dichos pues entre ambos copan cerca del 90 % del mercado de Smartphones en España y cerca del 80 % mundial. El resto algunas tuvieron gran peso pero hoy están en franca retirada (Blackberry), no terminan de funcionar todo lo bien que se desea entre el gran público (Windows Phone), son de dispositivos de otra era no están pensados para la movilidad moderna (Symbian), o no terminan de salir del halo de prototipo y aun deben pasar ciclos en fases beta para poder ser pasadas al gran público (Ubuntu Touch, Firefox OS).

Así pues la elección se encontraba entre Android e iOS de cara a afrontar este proyecto. Si tenemos en cuenta la cuota de mercado de cada uno de ellos ¡, Android sin duda alguna es el gran vencedor, pero es que además la gama de dispositivos que ofrece dicho SO es bastante más amplia y con mayor variabilidad en el precio de los terminales, permitiendo que más y más personas accediesen a la app.

A esto hay que añadir los costes de desarrollo, ya que para publicar aplicaciones en las tiendas de las mismas de los SO móviles se necesita de un proceso de registro como tal desarrollador en las tiendas, y este proceso no es gratuito. Para ser claros diremos que el coste en iOS es de 99 dólares anuales, además de disponer de un hardware específico para realizar el desarrollo y compilación de la app. Mientras que en Android el coste de 25 dólares en un pago único y el desarrollo y compilación puede llevarse a cabo con casi cualquier infraestructura de SO, hardware etc.

Es por eso que para este piloto de la iniciativa de la Asociación Española contra el Cáncer se haya optado por un desarrollo más económico, y tras analizar los resultados si evalúa la iniciativa como positiva y avanza del estadio de piloto, se planteará por parte de la AECC futuros desarrollos en ese sentido, pero eso queda fuera del alcance y ámbito de este proyecto.

2.3. Parte App móvil

En la parte de la app móvil hay 5 funcionalidades, aparte de los ajustes y preferencias, Personas Implicadas, Calendario de Citas y Posología, Seguimiento de Análisis, Rutina Diaria y Hablar con un agente, además de notificaciones locales preguntando por diferentes cosas, para mantener al usuario tanto en la app como con la moral lo más alta posible.

Personas Implicadas es una mini agenda personal, con el número de teléfono o correo electrónico u otras formas de contacto de las personas que estén más implicadas para el enfermo, como pueden ser su oncólogo, el agente de la AECC, un psicólogo propio, familiares o amigos de confianza con los que el agente de la AECC pueda contactar, previo consentimiento de estos y del paciente. En fin personas con interés en apoyar al enfermo en su duro trance de la enfermedad.

Calendario de Citas y Posología tiene como fin registrar las citas importantes relacionadas con el proceso del enfermo, como pueda ser revisiones, sesiones de quimio, entrega de análisis, ingresos, operaciones, etc. Para que el enfermo disponga en un lugar de toda la información referente a su caso. Si toma algún medicamento, analgésico etc también se reflejar aquí como parte importante del proceso puramente médico en sí.

Seguimiento de análisis permite tener un pequeño histórico para el usuario de los análisis que le hayan sido realizados, permitiendo fotografiar los mismos para poder consultarlo siempre y además introducir que parámetros quiere obtener un especial seguimiento y gráfico de los mismos, para ir comprobando su evolución

Rutina Diaria tiene como finalidad la de ser un horario semanal de actividades, tanto dentro de la AECC como fuera con la finalidad de que mediante la rutina y la realización de actividades el enfermo se encuentre mejor psicológicamente, además de físicamente en el caso de actividades físicas, y que mediante la rutina y realización activa de actividades consiga apartar de la mente la enfermedad.

Hablar con un agente permite durante ciertas horas del día que el enfermo pueda consultar o incluso simplemente charlar con el agente asignado de la AECC, para que sienta que siempre dispone de alguien que lo apoye desde el lado de la asociación

Además de esto la app dispone de preferencias tanto de sonidos y notificaciones como aspecto, así como justes de usuario borrar cuenta, etc

Por último la app funcionará mucho en base a la información recolectada en esta parte para lanzar notificaciones locales preguntando por diferentes cosas al enfermo, desde como te encuentras esta mañana? a la hora aproximada que en Rutina Diaria el usuario haya establecido como hora de despertar a has ido hoy a bailes de salón? el día que tenga marcado que tiene que ir a bailes pasando que animo tienes? tras haber salido de una sesión de quimioterapia. Todos estos datos desagregados del usuario se envían a la parte servidora para tratarlos con fines estadísticos y ponerlos a disposición de investigadores en el campo de la enfermedad.

2.4. Parte Servidora

La parte servidor lo primero que proporciona es una API REST(posible referencia) para interactuar con los recursos que ofrecerá.

Por una parte permite la sincronización de la información entre los diferentes dispositivos que un mismo enfermo pueda disponer. Esto lo hace de manera silenciosa enviando cada cambio al servidor, y este en cada conexión de un dispositivo pregunta por su estado de sincronización.

Por otro ofrece servicios para que cada dispositivo envíe la información estadística pertinente, además la almacenará en una base de datos, haciendo anónimos esos datos en el proceso por confidencialidad hacia el usuario, y permitiendo después su consulta mediante servicio web o en la web donde este alojada la parte servidora, para la monitorización de los mismos.

Dispondrá de control de usuarios, esto es que existirá un usuario encargado de ir asignando a los distintos agentes, generalmente por proximidad geográfica, para que este sea el encargado de monitorizar la actividad de sus enfermos

Se le ofrecerá al agente los contactos de cada enfermo que tenga asignado para en el caso de que necesitare conversare de alguna manera con alguno de ellos, bien sean familiares o el

oncólogo si por ejemplo se encuentra en algún ensayo poder contrastar información

En relación a esto ultimo la parte servidora dispondrá de un chat que permita comunicar directamente al agente con el enfermo, bajo ciertas premisas

CAPÍTULO 3

Planificación y Metodología

3.1. Plan de desarrollo de software

Para el desarrollo de este aplicativo se seguirán los principios del Agile Manifesto [4], y de entre todas las metodologías que lo implementan utilizaremos Scrum, ya que nos permite desarrollar siempre sobre algo ejecutable y tiene una buena "pelea contra el tiempo." "timeboxing"za que al final de cada sprint hay que hacer una retrospectiva sobre lo que ya se ha construido y entregado.

3.2. Propósito general de la planificación

La planificación nos debe dar una cifra orientativa del esfuerzo a comprometer para acometer un desarrollo de un proyecto software. Pero debido a lo mencionado anteriormente sobre el manifiesto ágil, creemos que dar una cifra estimativa en tiempo es venturoso, más si queremos ceñirnos a el y más aún cuanto mas a largo plazo sea la estimación. Por eso las metodologías ágiles suelen ocultar la referencia temporal de los desarrollos y estiman la complejidad de las tareas, sabiendo por el histórico, ya que los equipos deben ser fijos en el tiempo, la complejidad aproximada que un equipo dedicado a un proyecto puede acometer.

3.3. Metodología

Tradicionalmente se denomina metodología de desarrollo de software a un marco de referencia para estructurar, planificar, controlar y medir el proceso de desarrollo de software.[2]

En los últimos años se han popularizado diversas metodologías que rompen.^{en} parte con los paradigmas anteriores a cambiar el punto sobre el que ponen el foco. Es en estas nuevas metodologías, que se han venido a denominar ágiles por su capacidad rápida de respuesta al cambio, donde vamos a encontrar la metodología que vamos a utilizar para el desarrollo del proyecto.

Como generalidades sobre las metodologías ágiles decir que se basan todas ellas en la formación de grupos auto organizados y multidisciplinarios que colaboran para llevar a cabo el proyecto. De hecho es esta colaboración y esta auto organización las piezas clave para que el equipo vea que tiene poder de decisión durante la vida del proyecto, y que de esta manera se implique mucho más en el éxito del mismo.

Para la realización del proyecto buscaremos como ya hemos comentado dentro las diferentes metodologías ágiles, hemos elegido SCRUM, por ser la que mejor se adapta a la continua

pelea contra el tiempo que el equipo debe mantener.

Lo primero que debemos decir es que SCRUM es una metodología iterativa e incremental que promueve la auto organización de los equipos de desarrollo y un esquema de colaboración con el cliente, haciéndose responsable de que las prioridades, los requisitos, etc pueden cambiar por parte de este, y que es responsabilidad del propio equipo el asumir y responder a estos cambios.

3.3.1. SCRUM

SCRUM como metodología fué propuesto por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, al analizar cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard (Nonaka & Takeuchi, *The New New Product Development Game*, 1986).

Aunque esta forma de trabajo surgió en empresas de productos tecnológicos, es apropiada para proyectos con requisitos inestables y para los que requieren rapidez y flexibilidad, situaciones frecuentes en el desarrollo de determinados sistemas de software.

SCRUM define un conjunto de prácticas y roles, que pueden tomarse como punto de partida para definir el proceso de desarrollo que se ejecutará durante un proyecto. Abordar un proyecto mediante SCRUM lleva a la repetición o iteración de la unidad básica del proyecto denominada Sprint. Durante cada sprint, un periodo entre una y cuatro semanas (la magnitud es definida por el equipo y debe ser lo mas corta posible), el equipo crea un incremento de software potencialmente entregable (utilizable). El conjunto de características que forma parte de cada sprint viene del Product Backlog, que es un conjunto de requisitos de alto nivel priorizados que definen el trabajo a realizar a menudo denominadas Historias de usuario. Los elementos del Product Backlog que forman parte del sprint se determinan durante la reunión de Sprint Planning. Durante esta reunión, el Product Owner identifica los elementos del Product Backlog que quiere ver completados y los hace del conocimiento del equipo. Entonces, el equipo conversa con el Product Owner buscando claridad y magnitud adecuadas para luego determinar la cantidad de ese trabajo que puede comprometerse a completar durante el siguiente sprint. Durante el sprint, nadie puede cambiar el Sprint Backlog, lo que significa que los requisitos del sprint están congelados durante el propio sprint, pero podrían modificarse aquellos que aun estuviesen en el Product Backlog.

SCRUM permite la creación de equipos autoorganizados impulsando la co-localización de todos los miembros del equipo, y la comunicación verbal entre todos los miembros y disciplinas involucrados en el proyecto, ya que una de la bases de SCRUM es la colaboración y comunicación entre todos los roles de un proyecto.

Un principio clave de SCRUM es el reconocimiento de que durante un proyecto los clientes pueden cambiar de idea sobre lo que quieren y necesitan, y que los desafíos impredecibles no pueden ser fácilmente enfrentados de una forma predictiva y planificada. Por lo tanto, SCRUM adopta una aproximación pragmática, aceptando que el problema no puede ser completamente entendido o definido, y centrándose en maximizar la capacidad del equipo de entregar rápidamente y responder a requisitos emergentes.

Las características más marcadas que se logran notar en SCRUM serían: gestión regular de las expectativas del cliente, resultados anticipados, flexibilidad y adaptación, retorno de inversión, mitigación de riesgos, productividad y calidad, alineamiento entre cliente y equipo, por último equipo motivado. Cada uno de estos puntos mencionados hacen que el SCRUM sea utilizado de manera regular en un conjunto de buenas prácticas para el trabajo en equipo

y de esa manera obtener resultados posibles.

Los beneficios para las empresas más notables al utilizar SCRUM son

- Flexibilidad a cambios. Gran capacidad de reacción ante los cambiantes requerimientos generados por las necesidades del cliente o la evolución del mercado. El marco de trabajo está diseñado para adecuarse a las nuevas exigencias que implican proyectos complejos. Reducción del Time to Market. El cliente puede empezar a utilizar las características más importantes del proyecto antes de que esté completamente terminado.
- Mayor calidad del software. El trabajo metódico y la necesidad de obtener una versión de trabajo funcional después de cada iteración, ayuda a la obtención de un software de alta calidad. Mayor productividad. Se logra, entre otras razones, debido a la eliminación de la burocracia y la motivación del equipo proporcionado por el hecho de que pueden estructurarse de manera autónoma.
- Maximiza el retorno de la inversión (ROI). Creación de software solamente con las prestaciones que contribuyen a un mayor valor de negocio gracias a la priorización por retorno de inversión.
- Predicciones de tiempos. A través de este marco de trabajo se conoce la velocidad media del equipo por sprint, con lo que es posible estimar de manera fácil cuando se podrá hacer uso de una determinada funcionalidad que todavía está en el Backlog.
- Reducción de riesgos El hecho de llevar a cabo las funcionalidades de mayor valor en primer lugar y de saber la velocidad a la que el equipo avanza en el proyecto, permite despejar riesgos efectivamente de manera anticipada

Como puntos flacos tanto de SCRUM como de muchas de las metodologías ágiles podemos observar:

- Fuerte dependencia de las personas y en sus habilidades comunicativas y de auto organización
- Participación fuerte e intensa de todos los implicados en el proceso, con los riesgos por parte de un cliente poco participativo
- Falta de documentación o mucha laxitud en la que existe.

Aunque más adelante en este documento se profundiza en la metodología aquí se muestra una imagen que puede servir como resumen de todos los conceptos roles y rituales de SCRUM.

3.3.2. Roles y responsabilidades

Los roles implicados en el proceso de desarrollo y sus responsabilidades de la app son:

Product Owner

El Product Owner representa la voz del cliente, que puede ser el cliente mismo o alguien en su nombre. Se asegura de que el equipo trabaje de forma adecuada desde la perspectiva del negocio. El Product Owner escribe historias de usuario, las prioriza, y las coloca en el Product Backlog, de hecho se suele decir que el Product Owner es el dueño del Product Backlog. En nuestro caso este rol lo llevaban a cabo varias personas con las que hemos tenido contacto en la AECC.

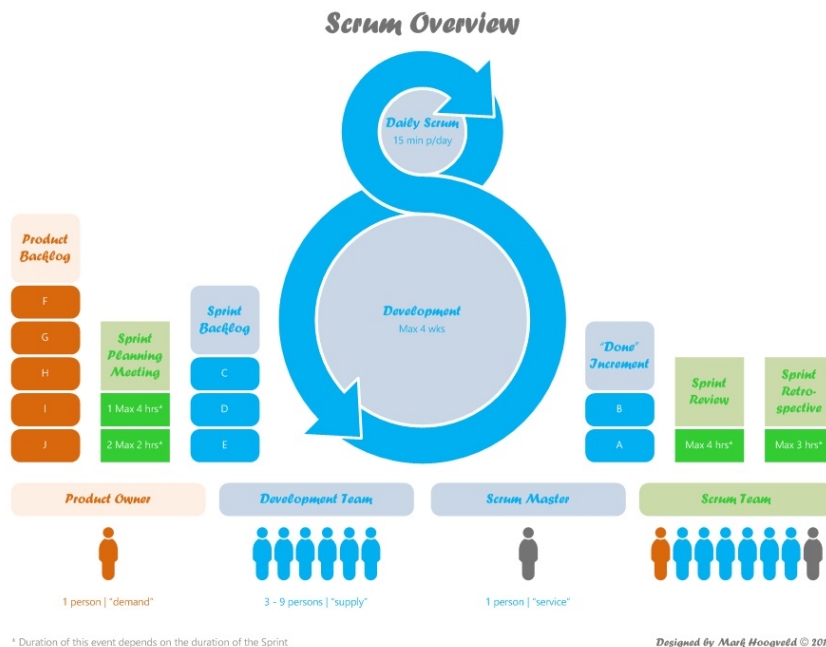


Figura 3.1: Resumen de Scrum.

ScrumMaster (o Facilitador)

El Scrum es facilitado por un ScrumMaster, cuyo trabajo primario es eliminar los obstáculos que impiden que el equipo alcance el objetivo del sprint. El ScrumMaster no es el líder del equipo (porque ellos se auto-organizan), sino que actúa como una protección entre el equipo y cualquier influencia que le distraiga. El ScrumMaster se asegura de que el proceso Scrum se utiliza como es debido. El ScrumMaster es el que hace que las reglas se cumplan. Quizá este es el puesto clave de Scrum ya que se encarga de que el equipo se centre en producir aquello que el Product Owner ha marcado. En nuestro caso este rol ha sido una suerte de Frankenstein ya que para algunas tareas ha sido el personal de la AECC, para otras ha sido el tutor y en algún caso nosotros mismos saliendo del rol de equipo.

Equipo de desarrollo

El equipo tiene la responsabilidad de entregar el producto. Es recomendable un pequeño equipo de 3 a 9 personas con las habilidades transversales necesarias para realizar el trabajo (análisis, diseño, desarrollo, pruebas, documentación, etc).

A veces se definen más roles que pueden complementar en alguna fase a estos, o ayudar en tareas administrativas, pero formalmente quedan un poco fuera de la metodología por mas que sean muy necesarios en las empresas hoy día.

3.3.3. Eventos y rituales

Dentro de la metodología de SCRUM existen varias reuniones o rituales, tras las que se esconde alguno de los principios o practicas de las metodologías ágiles[4], dentro de que

cada equipo debe adaptar la metodología a su idiosincrasia y condiciones particulares, se recomienda en la medida de lo posible ajustarse a las mismas ya que la ganancia que reportan está comprobado que refuerza todos los esfuerzos de la metodología.

Sprint

No es en sí un evento o un ritual, simplemente es la medida de tiempo en los que se dividen los proyectos a acometer por el equipo. Así el Sprint es el período en el cual se lleva a cabo el trabajo en sí. Se recomienda que la duración de los sprints sea constante y definida por el equipo con base en su propia experiencia. Se puede comenzar con una duración de sprint en particular (2 o 3 semanas) e ir ajustándolo con base en el ritmo del equipo, aunque sin relajarlo demasiado. Al final de cada sprint, el equipo deberá presentar los avances logrados, y el resultado obtenido es un producto potencialmente entregable al cliente. Asimismo, se recomienda no agregar objetivos al sprint o sprint backlog a menos que la falta de estos objetivos amenace al éxito del proyecto. La constancia permite la concentración y mejora la productividad del equipo de trabajo.

Daily Scrum o Stand-up meeting

Cada día de un sprint, se realiza la reunión sobre el estado de un proyecto. Esto se llama daily standup o Stand-up meeting. El scrum tiene unas guías específicas:

- La reunión comienza puntualmente a su hora.
- Todos son bienvenidos, pero sólo los involucrados en el proyecto pueden hablar. (Esto es el equipo y el scrum master).
- La reunión tiene una duración fija de 15 minutos, de forma independiente del tamaño del equipo.
- La reunión debe ocurrir en la misma ubicación y a la misma hora todos los días.
- Durante la reunión, cada miembro del equipo contesta a tres preguntas:
 1. ¿Qué has hecho desde ayer?
 2. ¿Qué es lo que harás para mañana?
 3. ¿Has tenido algún problema que te haya impedido alcanzar tu objetivo? (Es el papel del ScrumMaster recordar y anotar estos impedimentos para su posterior análisis y subsanación si procede).

Reunión de Planificación del Sprint

Al inicio de cada ciclo de Sprint, se lleva a cabo una reunión de planificación del Sprint. en esta reunión debe estar el equipo al completo, el scrum master y el product owner Se pretende:

- Seleccionar qué trabajo se llevará a cabo por el equipo durante el sprint
- Preparar, con el equipo completo, el Sprint Backlog o sea las tareas del sprint clarificando el Product Owner las posibles dudas o inconsistencias que el equipo detecte en la redacción de las tareas, este se debe extraer de las tareas más prioritarias del Product Backlog, de acuerdo con el Product Owner.
- Identificar y comunicar cuánto del trabajo es probable que se realice durante el actual Sprint.

Realizarse esta planificación en ocho horas como tiempo límite.

Reunión de Revisión del Sprint

Al finalizar cada sprint se organiza esta reunión en la que el equipo presenta el trabajo desarrollado durante el sprint en lo que se denomina Incremento de Producto Potencialmente Entregable. Durante esta reunión tiene lugar la presentación del ejecutable en lo que se denomina Demo. Así pues las principales actividades de esta reunión serían:

- Revisar el trabajo que fue completado y no completado
- Presentar el trabajo completado a los interesados, entre ellos el product owner y que estos de su aprobación al mismo o presenten sus quejas o matices (alias “demo”), pero teniendo en cuenta que los estados del trabajo son binarios, está acabado o no está acabado, no hay porcentajes intermedios.
- El trabajo incompleto no puede ser demostrado

Para esta reunión y dependiendo del tamaño del sprint se dispondrá de cuatro horas como límite

Retrospectiva del Sprint

Después de cada sprint, se lleva a cabo una retrospectiva del sprint, en la cual todos los miembros del equipo dejan sus impresiones sobre el sprint recién superado. El propósito de la retrospectiva es realizar una mejora continua del proceso. Esta reunión tiene un tiempo fijo de cuatro horas.

3.4. Planificación completa

Para realizar la planificación completa de la app debemos antes introducir algunos conceptos a la estimación en el contexto de las metodologías ágiles, como son velocidad del equipo, historias de usuario, puntos de historia, etc.

3.5. Recursos necesarios

Los recursos designados para la realización completa de este proyecto serán los siguientes:

Recursos humanos

- El equipo, en este caso los alumnos que defienden este proyecto.
- El scrum master, como se ha mencionado este papel esta repartido entre el tutor, personas involucradas desde la AAEC y en ocasiones alguno de los alumnos.
- El product owner, que ya se ha dicho antes en este documento son varias personas dentro de la AECC.

Determinar el esfuerzo del rol equipo en este proyecto es sencillo, pero el resto derroles es más complejo el computar las horas de dedicación a este proyecto de cara a realizar una planificación, de tal manera que se estimará que el rol scrum master sea un 30 % del tiempo consumido por el equipo y por contra el rol de product owner se estimará en un 25 % de la anterior medida.

Recursos Software

Los recursos software para esta iteración serán:

- Microsoft Project 2003 como herramienta de planificación
- TexStudio + Sharelatex herramienta de elaboración de documentos PDF
- Adobe Photoshop CS6 para realizar ilustraciones e imágenes
- StarUML y Enterprise Architect para modelado y diagramas
- Dropbox y github como herramientas para compartir código y archivos
- Android Studio como IDE (Integrated Development Enviroment o entorno integrado de desarrollo) para el desarrollo del código Android necesario
- SQLite como sistema gestor de bases de datos en el dispositivo móvil

Recursos Hardware

Los recursos hardware para esta iteración serán:

- Dos ordenadores portátiles personales
- Un ordenador Personal de sobremesa
- Varios móviles con SO Android
- Memoria USB para el intercambio de datos
- Cables USB - Micro USB (Debug)

CAPÍTULO 4

Análisis

La fase de análisis y el proceso de Ingeniería de Requisitos, parten básicamente de dos contextos bien diferenciados:

-Por una parte de las reuniones mantenidas con el cliente, en este caso la AECC.

-Por otra parte del libro que utiliza la AECC llamado **MIS CUIDADOS, DIARIO DE SALUD DE UN SUPERVIVIENTE DE CANCER** y que se utiliza de manera general sobre enfermos de cáncer que han superado esta enfermedad y sobre otros que todavía la siguen padeciendo.

4.1. ¿El porqué?

Desde el punto de vista de la definición clásica:

La ingeniería del software es el proceso formal de desarrollo de software en el que las necesidades del usuario se traducen en requisitos, estos se transforman en diseño que se implementa en código que se prueba, documenta y se certifica para su uso operativo. Según la definición del IEEE la ingeniería del software se define como “(1) la aplicación de un método sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, esto es, la aplicación de la ingeniería al software” y “(2) el estudio de los métodos de (1)”

El proceso requiere una metodología con 5 etapas:

Análisis de requisitos: Se extraen los requisitos del producto de software. En esta etapa la habilidad y experiencia en la ingeniería del software es crítica para reconocer requisitos incompletos, ambiguos o contradictorios.

Usualmente el cliente/usuario tiene una visión incompleta/inexacta de lo que necesita y es necesario ayudarlo para obtener la visión completa de los requerimientos. El contenido de comunicación en esta etapa es muy intenso ya que el objetivo es eliminar la ambigüedad en la medida de lo posible.

Partiendo de un exhaustivo estudio de las funcionalidades que podrían resultar útiles para los pacientes de Cáncer, y de acuerdo con las especificaciones de la AECC hemos decidido incluir en la misma las siguientes:

4.2. Explicación detallada de las funcionalidades de la aplicación

AVATAR

Editar el perfil, Foto, nombre, apellidos, edad, cumpleaños... (1 pantalla), ¿Grupo sanguíneo y alérgenos tampoco estaría mal pero es información de otro nivel LOPD?

PROYECTO AMPLIACIÓN (asignar un agente personalizado para poder atender de manera personalizada al paciente Superviviente)

PRINCIPAL

Vista de Rutina próxima, citas, Se debería ver lo que realmente interesa ¿Noticias AECC tipo RSS, Twitter, Facebook? (1 pantalla)

HORARIO

Vista día, mes, con las actividades, citas, ¿Cumpleaños, aniversario de enfermedad...? al pinchar en una hora, día, se podrá añadir rutina o cita, iría a la correspondiente lista de creación de una nueva (1 pantalla)

Deberían aparecer las citas y rutinas ocupando el espacio destinado a la duración de las mismas.

Revisar esta página <https://code.google.com/p/yadview/>

RUTINA Lista de rutinas, desde la propia vista se podrán añadir rutinas, al pulsar sobre la opción correspondiente nos llevará al listado de estas opciones añadiéndose así.

Se podrá ordenar por fecha, asc y desc, duración de la actividad asc y desc pero sobre todo se podrá ordenar por satisfacción de la misma.

Añadir rutina, , en esta pantalla se permite ver al pinchar sobre el propio ítem de la lista ,editar, o pulsando el botón añadir del final de la lista crear una nueva, Valorar si deberíamos gestionar la repetición de los elementos dentro de la rutina (añadir) se podrán añadir personajes y asignar una alarma personalizable (avisar) con antelación, se puede ver sin editar(2 pantallas) con una hora de aviso, hora de empiece de la actividad. Duración de las mismas en horas, satisfacción de 0 a 10

Dentro se va a valorar también la satisfacción personal que se produce al hacer esto, podrá ser de 0 a 10

AMPLIACIÓN DE PROYECTO: Poder variar los avisos para que se hagan de manera repetitiva, por ejemplo repetir de manera semanal los jueves al estilo TICK TICK, llamarlo actividades en vez de Rutina

CITAS

Lista de citas Nombre de la cita, fecha y hora. Desde el propio ítem de la lista se podrán añadir medicamentos, personas, síntomas y/o pruebas, al pulsar sobre la opción correspondiente nos llevará al listado de estas opciones añadiéndose así.

Añadir cita, en esta pantalla se permite ver al pinchar sobre el propio ítem de la lista ,editar, o pulsando el botón añadir del final de la lista crear una nueva, dentro de la cita (añadir) se podrán añadir ubicación, personajes, medicamentos, pruebas y síntomas y asignar una alarma personalizable con fecha y hora (avisar) con antelación. se puede ver sin editar

Deberíamos añadir duración?

Valorar si deberíamos gestionar la repetición de los eventos

¿En la ubicación se podría comunicar con google maps?

¿Deberíamos mostrar en el ítem de la cita si ya posee personas, pruebas, síntomas o medicación?

PROYECTO AMPLIACION Poder exportar e importar a google maps, conectar la ubicacion con google maps para poder ir

MEDICACION Lista de medicamentos, lista de los medicamentos que tengamos con el boton de añadir al final y en la barra la posibilidad de ordenar en principio por orden ascendente o descendente, al pinchar sobre el elemento se visualiza, en las opciones de cada elemento se puede añadir a la cita, editar y borrar (confirmacion para borrar)

Añadir medicamento, , en esta pantalla se permite ver al pinchar sobre el propio item de la lista ,editar, o pulsando el botón añadir del final de la lista crear una nueva, dentro del medicamento (añadir) se podrá asignar una foto personalizable, o bien hacerla o añadirla de la galeria, viene una por defecto nombre, descripcion, una alarma personalizable que avisar con antelacion, fecha inicio, hora inicio, fecha fin, hora fin, repetir cada tantas horas, dias, semanas, (esto se valorará)

REVISAR LA FUNCION DE ALARMA EN ANDROID <http://developer.android.com/reference/android> boton guardar se puede ver sin editar, dentro de la edición y visionado se podra añadir a una cita Se puede añadir el numero de medicamentos al drawer (se estudiará)

PROYECTO AMPLIACION Llevar el stock de lo que lleva consumido el paciente y generar alertas para cuando se le acabe, duplicar tratamiento, para ahorrar trabajo, opciones de dosis.

PERSONAJES

Lista de personajes que tengamos con el boton de añadir al final y en la barra la posibilidad de ordenar en principio por orden ascendente o descendente, al pinchar sobre el elemento se visualiza, en las opciones de cada elemento se puede añadir a la cita, a la rutina, editar y borrar confirmacion para borrar

Añadir persona, , en esta pantalla se permite ver al pinchar sobre el propio item de la lista ,editar, o pulsando el botón añadir del final de la lista crear una nueva, dentro de la persona(añadir) se podrá asignar una foto personalizable, o bien hacerla o añadirla de la galeria, viene una por defecto nombre apellidos, puesto / tipo (doctor, oncologo, cirujano, farmaceutico, enfermero, medico cabecera, familiar, amigo,AECC, otro), telefono (posibilidad de llamar desde la visualizacion) se puede ver sin editar, dentro de la edición y visionado se podra añadir a una cita, o a una rutina ejemplo (paseo por la playa) o a cita con el oncólogo Se puede añadir el numero al drawer

PROYECTO AMPLIACION llevarnos ese contacto a la agenda o traernosle de ella

PRUEBAS

Lista de personajes que tengamos con el boton de añadir al final y en la barra la posibilidad de ordenar en principio por orden de fecha ascendente o descendente, al pinchar sobre el elemento se visualiza, en las opciones de cada elemento se puede añadir a la cita, editar y borrar confirmacion para borrar

Añadir prueba, , en esta pantalla se permite ver al pinchar sobre el propio item de la lista ,editar, o pulsando el botón añadir del final de la lista crear una nueva, dentro de la prueba (añadir) se podrá asignar una foto personalizable, o bien hacerla o añadirla de la galeria, viene una por defecto nombre, descripcion, fecha y hora de la misma se puede ver sin editar, dentro de la edición y visionado se podra añadir a una cita por ejemplo visita al oncólogo Se puede añadir el numero al drawer

PROYECTO AMPLIACION poder subir documentos tipo pdf para no depender de la fotografia solo, llevar estadísticas

SINTOMAS

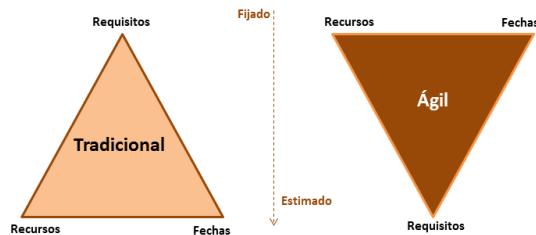


Figura 4.1: Paradigma captura de requisitos método tradicional vs. ágil

Lista de síntomas que tengamos con el boton de añadir al final y en la barra la posibilidad de ordenar en principio por orden de fecha ascendente o descendente, al pinchar sobre el elemento se visualiza, en las opciones de cada elemento se puede añadir a la cita, editar y borrar confirmacion para borrar

Añadir sintoma, , en esta pantalla se permite ver al pinchar sobre el propio item de la lista ,editar, o pulsando el botón añadir del final de la lista crear una nueva, dentro de la prueba (añadir) se podrá asignar una foto personalizable, o bien hacerla o añadirla de la galeria, viene una por defecto nombre, descripcion de los sintomas, fecha y hora de la misma se puede ver sin editar, dentro de la edición y visionado se podra añadir a una cita por ejemplo visita al oncólogo Se puede añadir el numero al drawer

PROYECTO AMPLIACION poder subir documentos tipo pdf para no depender de la fotografia solo, llevar estadísticas, poder conectar con los centros de salud, hospitales,

RECURSOS Dentro de los recursos aparece meditacion con musica relajante e instrucciones para ello consejos generales, de alimentacion, de vida, de salud... telefonos de interés generales (DEBEMOS DECIDIR SI SE PUEDEN PERSONALIZAR) Noticias (podemos añadir el blog de la AECC o la cuenta de twitter que no esta nada mal)

AJUSTES Editar el perfil, tamaño de letra, tono para alertas de citas y rutinas, color del led de notificación, posibilidad de feedback, quienes somos o motivaciones (¿? pantallas)

4.3. Documento de Análisis

Aunque utilizamos una metodología ágil, hay puntos en los que no nos podemos separar de la metodología tradicional.

Es difícil capturar todos los requisitos a partir de unas cuantas reuniones, para nuestro caso, pudimos contar con la ayuda del libro "Mis Cuidados, diario de salud para supervivientes de cáncer"

Una de las ventajas de utilizar este primer tipo de metodología, es el de poder adaptar de una manera mucho más eficiente los recursos a las fechas y a los requisitos, y poder modificar estos requisitos de manera "ágil" debido al panorama cambiante que podemos tener de cara al cliente.

4.3.1. Descripción de objetivos de manera detallada

OBJ-001 La aplicacion debera poder administrar un perfil de usuario

OBJ-001	PERFIL DE USUARIO
Versión	1.0
Autores	Fernando Santa Olaya Rodríguez Rubén Toquero González
Descripción	La aplicacion debera poder administrar un perfil de usuario
Subobjetivos	El usuario podrá introducir una foto, nombre, fecha de nacimiento
Importancia	Importante
Urgencia	Inmediatamente
Estado	En construcción
Estabilidad	Estable
Comentarios	En la cabecera del drawer aparecerá el nombre y la foto

Cuadro 4.1: Menú drawer.

OBJ-002 La aplicacion será facil de navegar para usuarios no avanzados

OBJ-003 La aplicacion facilitará la gestion de las actividades diarias de un usuario (medicas/rutinarias)

OBJ-004 La aplicacion permitirá gestionar de manera organizada las citas médicas de un usuario

OBJ-005 La aplicacion permitirá gestionar de manera organizada las rutinas diarias de un usuario

OBJ-006 La aplicacion permitirá gestionar de manera organizada los medicamentos asignados a un usuario

OBJ-007 La aplicacion permitirá gestionar de manera organizada los personajes que interactuan en las actividades de un usuario

OBJ-008 La aplicacion permitirá gestionar de manera organizada los síntomas que acontecen a un usuario

OBJ-009 La aplicacion permitirá gestionar de manera organizada las pruebas médicas a las que se somete un usuario

OBJ-010 La aplicacion proporcionará recursos de ayuda para el bienestar del usuario

OBJ-011 La aplicación permitirá ajustes que mejoren la experiencia del usuario

OBJ-012 La aplicación deberá tener una base de datos para almacenar toda la informacion del usuario

OBJ-013 La aplicación deberá contar con un manual de usuario para facilitar su comprension y uso

OBJ-014 La aplicación soportará el mayor número de dispositivos android posibles

OBJ-015 La aplicación será de verdadera ayuda para usuarios incluidos dentro de la

iniciativa de la AECC

OBJ-016 La aplicación se integrará perfectamente dentro de la iniciativa de la AECC

4.3.2. Captura de Requisitos

Como consecuencia de las diferentes reuniones realizadas con el cliente se elabora un documento de especificación de requisitos con el propósito de describir las funcionalidades necesarias para poder validar el producto final.

4.3.2.1. Requisitos de información

4.3.2.2. Requisitos Funcionales

A continuación se describirán los requisitos para este proyecto. Se tendrán dos tipos de requisitos: requisitos funcionales, que serán aquellos obtenidos a partir de los objetivos principales y secundarios del apartado anterior; y requisitos no funcionales, obtenidos de las especificaciones citadas a lo largo de la memoria y que precisen ser descritas como requisitos.

Este tipo de requisitos declaran los servicios que debe proporcionar el sistema, como debe reaccionar a una entrada particular y cómo se debe de comportar ante situaciones particulares. Describen el funcionamiento del sistema.

Los requisitos de información llevarán un nombre descriptivo y un número para identificarlos. Este número tendrá el formato **IRQ** - *jid*, donde *id* será el número del requisito de información. Además se indicarán los requisitos asociados, una descripción y cualquier información que sea relevante para una mayor claridad.

Los requisitos funcionales que debe cumplir el sistema son los siguientes

FRQ-001 El usuario podra acceder a un menú de navegación **OBJ-002**

FRQ-001	DRAWER MENÚ
Versión	1.0
Autores	Fernando Santa Olaya Rodríguez Rubén Toquero González
Dependencias	OBJ-002
Descripción	El usuario podrá acceder a un menú de navegación
Importancia	Vital
Urgencia	Inmediatamente
Estado	En construcción
Estabilidad	Estable
Comentarios	El menú será visible al desplazar el dedo de izquierda a derecha de la pantalla

Cuadro 4.2: Menú drawer.

FRQ-002 El usuario podrá introducir sus datos personales en la aplicación **OBJ-001**

FRQ-003 El usuario podra visualizar sus datos personales **OBJ-001**

FRQ-004 Es usuario podrá ver/editar sus datos personales **OBJ-001**

FRQ-005 El usuario al acceder a la aplicación podra visualizar de un vistazo su actividad para hoy **OBJ-003**

FRQ-006 El usuario podrá acceder desde la pantalla principal a sus citas medicas **OBJ-003 OBJ-004**

FRQ-007 El usuario podrá acceder desde la pantalla principal a sus rutinas diarias **OBJ-003 OBJ-005**

AFFECTADAS POR EL OBJETIVO OBJ-004

FRQ-008 La aplicación mostrará una vista del dia actual con actividades del usuario de ese dia (citas y rutinas)

FRQ-009 La aplicación mostrará una vista del mes actual para las actividades del usuario de ese mes (citas y rutinas) **OBJ-003**

FRQ-010 El usuario podrá añadir una cita desde cualquiera de las vistas mensual/diaria

FRQ-011 El usuario podrá añadir una rutina para ese dia desde cualquiera de las vistas mensual/diaria

FRQ-012 El usuario podrá visualizar/editar la cita desde cualquiera de esas vistas

FRQ-013 El usuario podrá visualizar/editar la rutina desde cualquiera de esas vistas

AFFECTADOS POR EL OBJETIVO OBJ-004

FRQ-014 El usuario podrá acceder a la lista de citas medicas

FRQ-015 El usuario podrá añadir desde la lista una nueva cita medica

FRQ-016 El usuario podrá añadir alertas y notificaciones a una cita médica

FRQ-017 El usuario podrá añadir personajes a una cita medica

FRQ-018 El usuario podrá añadir pruebas a una cita medica

FRQ-019 El usuario podrá añadir medicamentos a una cita medica

FRQ-020 El usuario podrá añadir duración a una cita médica

FRQ-021 El usuario podrá añadir síntomas a una cita medica

FRQ-022 El usuario podrá visualizar/editar la cita medica

AFFECTADOS POR EL OBJETIVO OBJ-005

FRQ-023 El usuario podrá acceder a la lista de rutinas

FRQ-024 El usuario podrá añadir desde la lista una nueva rutina

FRQ-025 El usuario podrá añadir alertas y notificaciones a una rutina

FRQ-026 El usuario podrá añadir duración a una rutina

FRQ-027 El usuario podrá añadir personajes a una rutina

FRQ-028 El usuario podrá visualizar/editar la rutina

AFFECTADOS POR EL OBJETIVO OBJ-006

FRQ-029 El usuario podrá acceder a la lista de medicamentos

FRQ-030 El usuario podrá añadir desde la lista un nuevo medicamento

FRQ-031 El usuario podra añadir información importante respecto a ese medicamento

FRQ-032 El usuario podrá añadir alertas y notificaciones a medicamento

FRQ-033 El usuario podrá visualizar/editar el medicamento

AFFECTADOS POR EL OBJETIVO OBJ-007

FRQ-034 A FRQ-049 Lo mismo con personajes, sintomas, pruebas

AFECTADOS POR EL OBJETIVO OBJ-010

FRQ-050 El usuario podrá acceder a recursos relacionados con el autocontrol y la meditación

FRQ-051 El usuario tendrá acceso a telefonos de interés

FRQ-052 El usuario tendrá acceso a consejos que le proporcionen bienestar

FRQ-053 El usuario tendrá acceso a noticias relativas a la AECC

FRQ-054 La aplicación notificará al paciente a petición citas y rutinas

FRQ-055 Las listas serán ordenables según diferentes criterios

AFECTADOS POR EL OBJETIVO OBJ-011

FRQ-056 La aplicación contendrá ajustes que faciliten su manejo al usuario

4.3.2.3. Requisitos No Funcionales

Este tipo de requisitos define las propiedades emergentes del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad.

Pueden especificar también la utilización de una herramienta CASE en particular, un lenguaje de programación o un método del desarrollo.

En este apartado se detalla el catálogo de requisitos no funcionales. Estos requisitos pueden ser más críticos que los requisitos funcionales, ya que son a los que normalmente debe apuntar la arquitectura y si estos no son cumplidos, el software puede no funcionar o el cliente simplemente no acepta el producto.

Estos requisitos se puede decir que no están relacionados directamente con la funcionalidad del sistema. Con esto no se pretende decir que no sean necesarios para la funcionalidad del sistema, pero sí que no pueden asociarse a ningún caso de uso en particular. Hay distintas categorías y tipos de requisitos no funcionales: hardware, plataforma software, comunicaciones, de interfaz, de seguridad, de control de concurrencia, de persistencia... etc.

Los requisitos no funcionales llevarán un nombre descriptivo y un número para identificarlos. Este número tendrá el formato NFRQ - *id* donde *id* será el número del requisito de información. Además se indicará una descripción y cualquier información que sea relevante para una mayor claridad.

Algunos requisitos no funcionales identificados para el desarrollo del presente proyecto son los siguientes:

AFECTADOS POR EL OBJETIVO OBJ-015

NFRQ-001 La aplicación será un apoyo y una ayuda para el paciente

AFECTADOS POR EL OBJETIVO OBJ-016

NFRQ-002 La aplicación debe ser coherente con el programa de Diario de Salud de la AECC.

AFECTADOS POR EL OBJETIVO **OBJ-013 OBJ-015**

NFRQ-003 La aplicacion sera amigable para usuarios no avanzados

AFECTADOS POR EL OBJETIVO **OBJ-016**

NFRQ-004 La aplicacion cumplira las guias de estilo fijadas por la AECC

AFECTADOS POR EL OBJETIVO **OBJ-015**

NFRQ-005 La aplicación proporcionará recursos de valor para los enfermos de cancer.

NFRQ-006 La aplicación será mantenible y escalable a petición del cliente.

AFECTADOS POR EL OBJETIVO **OBJ-004**

NFRQ-007 La aplicación facilitara las visitas médicas

AFECTADOS POR EL OBJETIVO **OBJ-005**

NFRQ-008 La aplicacion incentivará la realizacion de actividades placenteras para el paciente

AFECTADOS POR EL OBJETIVO **OBJ-003**

NFRQ-009 La aplicacion facilitara la organizacion de los compromisos del usuario

NFRQ-010 La aplicación será segura y confiable

NFRQ-011 La aplicación estará finalizada antes del dia 9 de septiembre

4.3.3. Identificación de actores

4.3.4. Diagrama de casos de uso

4.3.5. Descripción de Casos de Uso

CASOS DE USO

CU-001 CREACIÓN DE PERFIL - Inserción de perfil

CU-001	CREAR PERFIL - INSERCIÓN
Versión	1.0
Fecha	23/07/2015
Actores	Usuario
Objetivo	Incluir en el sistema los datos personales del usuario
Precondición	Los datos del usuario están sin completar
Flujo normal	1.- Desplegamos el menú drawer 2.- Pulsamos el circulo que está al principio del menú 3.- Rellenamos los datos personales del usuario
Flujo alternativo	1.- Desplegamos el menú drawer 2.- Pulsamos la opcion ajustes 3.- Pulsamos la opción en pantalla "Perfil" 3.- Rellenamos los datos personales del usuario
Postcondición	El usuario posee sus datos personales en la aplicación
Comentarios	A la hora de introducir la fotografía dentro de los datos personales, podrá elegir entre las que tenga en la galeria del propio terminal o acceder a la aplicación de cámara para hacer una nueva fotografía

Cuadro 4.3: Creación del perfil - Inserción

CU-002 CREACION DE PERFIL - edicion de perfil

CU-002	CREAR PERFIL - EDICIÓN
Versión	1.0
Fecha	23/07/2015
Actores	Usuario
Objetivo	Incluir en el sistema los datos personales del usuario
Precondición	Los datos del usuario están sin completar
Flujo normal	1.- Desplegamos el menú drawer 2.- Pulsamos el círculo que está al principio del menú 3.- Rellenamos los datos personales del usuario
Flujo alternativo	1.- Desplegamos el menú drawer 2.- Pulsamos la opción ajustes 3.- Pulsamos la opción en pantalla "Perfil" 3.- Rellenamos los datos personales del usuario
Postcondición	El usuario posee sus datos personales en la aplicación
Comentarios	A la hora de introducir la fotografía dentro de los datos personales, podrá elegir entre las que tenga en la galería del propio terminal o acceder a la aplicación de cámara para hacer una nueva fotografía

Cuadro 4.4: Creación del perfil - Inserción

CU-003 CREACION DE PERFIL - visualizacion de perfil

CU-003	CREAR PERFIL - BORRADO
Versión	1.0
Fecha	23/07/2015
Actores	Usuario
Objetivo	Incluir en el sistema los datos personales del usuario
Precondición	Los datos del usuario están sin completar
Flujo normal	1.- Desplegamos el menú drawer 2.- Pulsamos el círculo que está al principio del menú 3.- Rellenamos los datos personales del usuario
Flujo alternativo	1.- Desplegamos el menú drawer 2.- Pulsamos la opción ajustes 3.- Pulsamos la opción en pantalla "Perfil" 3.- Rellenamos los datos personales del usuario
Postcondición	El usuario posee sus datos personales en la aplicación
Comentarios	A la hora de introducir la fotografía dentro de los datos personales, podrá elegir entre las que tenga en la galería del propio terminal o acceder a la aplicación de cámara para hacer una nueva fotografía

Cuadro 4.5: Creación del perfil - Inserción

CU-004 CREACION CITA MEDICA - insercion de cita médica**CU-005 CREACION CITA MEDICA - edicion de cita médica****CU-006 CREACION CITA MEDICA - visualizacion de cita médica**

- CU-007** CREACION CITA MEDICA - borrado de cita médica
- CU-008** CREACION RUTINA - insercion de rutina
- CU-009** CREACION RUTINA - edicion de rutina
- CU-010** CREACION RUTINA - visualizacion de rutina
- CU-011** CREACION RUTINA - borrado de rutina
- CU-012** CREACION MEDICAMENTO - insercion de medicamento
- CU-013** CREACION MEDICAMENTO - edicion de medicamento
- CU-014** CREACION MEDICAMENTO - visualizacion de medicamento
- CU-015** CREACION MEDICAMENTO - borrado de medicamento
- CU-016** CREACION PERSONAJE - insercion de personajes
- CU-017** CREACION PERSONAJE - edicion de personajes
- CU-018** CREACION PERSONAJE - visualizacion de personajes
- CU-019** CREACION PERSONAJE - borrado de personajes
- CU-020** CREACION SINTOMA - insercion de síntoma
- CU-021** CREACION SINTOMA - edicion de síntoma
- CU-022** CREACION SINTOMA - visualizacion de síntoma
- CU-023** CREACION SINTOMA - borrado de síntoma
- CU-024** CREACION PRUEBA -insercion de pruebas
- CU-025** CREACION PRUEBA -edicion de pruebas
- CU-026** CREACION PRUEBA -visualizacion de pruebas
- CU-027** CREACION PRUEBA -borrado de pruebas
- CU-028** ORDENAR LISTAS
- CU-029** VISUALIZAR LISTAS
- CU-030** CONFIGURAR NOTIFICACIONES
- CU-031** AJUSTAR LA APLICACION

4.3.6. Diagrama de clases de análisis

4.3.7. Diagrama Entidad-Relación de la base de datos

4.3.8. Modelo relacional del análisis

CAPÍTULO 5

Diseño

5.1. Parte App móvil

5.2. Arquitectura

Como se ha comentado antes en este documento Scrum y las tecnologías ágiles, por su forma de enfocar las tareas y el propio desarrollo, si no se introducen medidas paliativas que corrijan esto, suelen obtener diseños arquitecturales de clases un poco pobres, tendentes al acoplamiento y poco cohesionado. Para paliar esto se suelen introducir historias de usuario de reducción de deuda técnica, o de refactorización de todo el código existente, así como herramientas que detecten *code smells* (cita o algo) o malas practicas de desarrollo. Estas herramientas de análisis estático de código pueden advertirnos sobre puntuales problemas dentro de una clase, pero poco hay a nivel de las relaciones entre las mismas, fuera del uso de patrones de diseño conocidos, y aun en las relaciones entre estos patrones que son bloques más grandes de código. Es por eso que creemos que hay que establecer unas líneas base al principio del proyecto sobre la arquitectura de la propia aplicación. En principio serán ciertamente vaguedades y lugares comunes sobre el desarrollo en capas, aislamiento de las mismas etc, pero si a través de los sprints reservamos un poco de tiempo en la reunión de sprint planning para conversar sobre las historias del propio sprint, en que capa encajan, como deberían hablar las capas entre sí si son mas de una las involucradas en la historia, etc ademas de con las historias de deuda técnica si fuesen necesarias creemos que puede solventarse en gran medida esa falta de bueno diseño y arquitectura que se achaca a las metodologías ágiles.

Para hablar de arquitectura hemos tomado como referencia Clean Architecture(cita) propuesta por Robert C. Martin aunque es muy similar a la Arquitectura Cebolla de Kent Beck (comporbar) o la Arquitectura Hexagonal de Alaister Cockburn. Todas ellas se basan por hacer capas fuertemente cohesionadas dentro de cada una pero con nulo conocimiento fuera de las mismas que se comunican mediante interfaces públicos ofrecidos al resto de capas. Esto y la propagación hacia el interior, esto es que cada capa solo hablara con la que tiene arquitecturalmente en un nivel superior o inferior. Esto es la capa de UI que es la que interactúa con el usuario solamente invocará métodos de la interfaz de la capa de vista que es la que conceptualmente tiene justo encima. Esta solamente interactuará con la capa de UI antes mencionada y con la capa del presentador, y así sucesivamente.

5.3. Tecnología

5.4. Diagrama de Clases

5.5. Diseño de la base de datos

5.6. Diagramas de estado

5.7. Prototipado

En la fase de diseño, el propósito del prototipo es obtener una primera versión de la apariencia de la interfaz de usuario así como de la funcionalidad incluida (mostrar las ventanas, su navegación, interacción, controles y botones). Con esto se pretende que el cliente tenga una primera toma de contacto con la futura aplicación antes de su desarrollo final, para así reducir o eliminar todas aquellas disconformidades y cambios en fases futuras.

5.7.1. Material Design

Material Design es la nueva metáfora visual de Matias Duarte para el sistema operativo móvil Android, en su version 5.0 y posteriores, y todo el ecosistema Google en web y supone una pequeña ruptura con la anterior metáfora de Android llamada Holo.

Es un diseño donde la profundidad, las superficies, los bordes, las sombras y los colores juegan un papel principal.

Precisamente este diseño basado en objetos es una manera de intentar aproximarse a la realidad, algo que en un mundo donde todo es táctil y virtual es difícil. Material Design quiere guiarse por las leyes de la física, donde las animaciones sean lógicas, los objetos se superpongan pero no puedan atravesarse el uno al otro y demás.

Material Design es un diseño con una tipografía clara, casillas bien ordenadas, colores e imágenes llamativos para no perder el foco y un sentido del orden y la jerarquía muy marcado. Estas ideas ya se aplican en muchos diseños, pero en Material Design Google ha creado unas normas muy claras de como llevarlo a la práctica.

5.7.2. Pantallas Principales

Pantallas iniciales y flujo de la aplicación que de manera inicial hemos planteado para la misma. No es la versión final de la aplicación pero si se aproxima y nos da una idea clara de lo que debería ser o a lo que debería parecerse. Estos prototipos, sirven para orientarnos a la hora de construir la aplicación

Pantalla Principal y Drawer menú El usuario, al ejecutar la aplicación accede a una pantalla cuyo contenido es el listado de las actividades próximas a realizar, ya sea una cita médica o una rutina. El drawer nos permite navegar entre actividades, desde la pantalla principal podemos añadir citas y acceder de manera individual a las mismas.

Pantallas referentes a la programación de actividades Desde estas pantallas tenemos una vision de la acupación diaria del paciente y de la ocupación mensual, desde la vista mensual al menos se podrán añadir nuevas citas médicas o rutinas para ese paciente.

Pantallas referentes a la rutina diaria Listado de las rutinas diarias que tiene ese paciente y representación de la introducción de una nueva rutina para ese paciente. Dentro de la introducción de la rutina se podrán añadir nuevos personajes que acompañarán a ese paciente y le harán más fácil recordar con quien quedó y para qué.

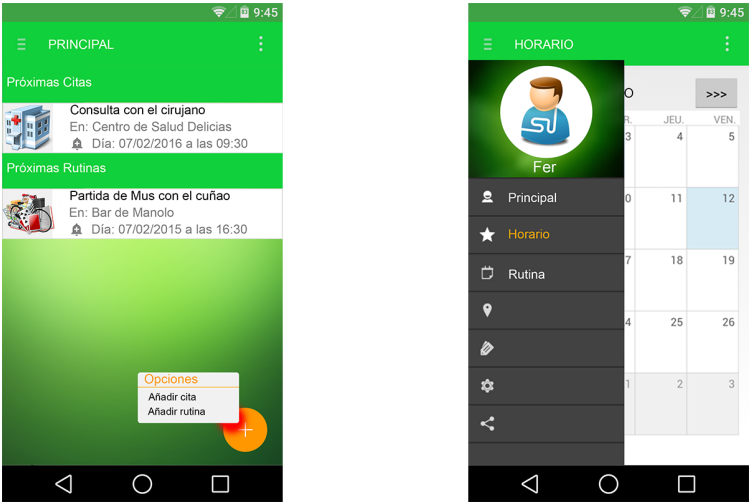


Figura 5.1: Descripción de la pantalla principal y Drawer menú

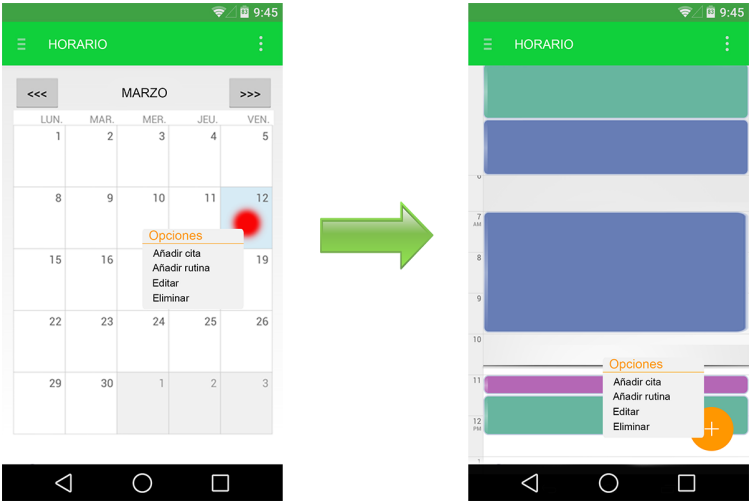


Figura 5.2: Pantallas con los horarios y ocupación del paciente

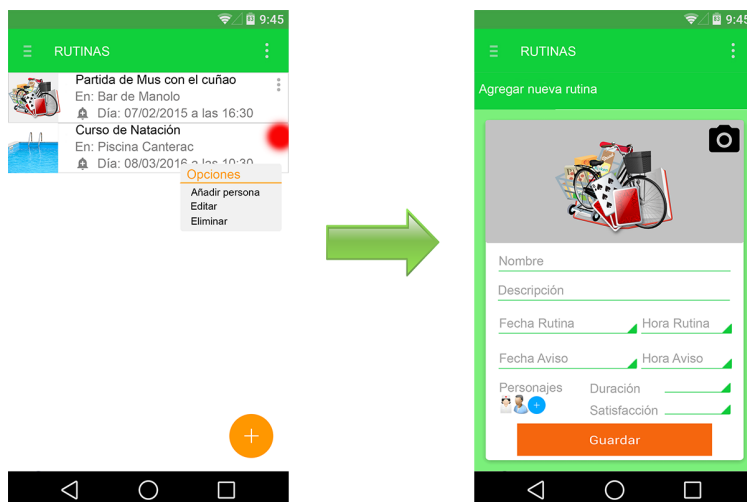


Figura 5.3: Pantalla con la interfaz para las rutinas

Pantallas referentes a las citas médicas Listado de las citas medicas de ese paciente e introducción de la cita en si de manera individual. De funcionamiento similar a las pantallas anteriores pero con diferencias sustanciales en la funcionalidad de la introducción de información, se pueden añadir personajes, síntomas, medicamentos y pruebas, de manera que el paciente cuando acuda a la cita lleve de manera ordenada toda esta información.

Pantallas referentes a la medicación El usuario introduce los medicamentos que toma, de manera que puede añadir alertas a los mismos para que se le avise de que ha llegado la hora de la dosis con una notificación. Se podrán introducir otros datos del medicamento como pueden ser los intervalos entre dosis, indicaciones...

Pantallas de gestion de personajes El usuario puede añadir a la aplicación su lista de amigos, personal médico que le atiende, asistentes, etc, con sus datos personales, de manera que a la hora de hacer una rutina (actividad) o acudir a una cita, puedan ser añadidos.

Pantallas de gestion de pruebas médicas Se facilita el poder llevar las pruebas médicas en forma de archivo fotográfico para poder consultarse en alguna de las citas medicas a las que acuda, tales como revisiones, visitas al fisio, a la enfermera. Listado del mismo.

Pantallas de gestión de los síntomas El usuario puede tomar nota de los síntomas, molestias, efectos secundarios negativos o positivos que le acontecen, para poder asignarles despues a una cita médica y poder completar de manera más eficiente las mismas, pudiendo aportar información adicional que ayude a su médico a mejorar el diagnóstico.

Pantallas de Recursos Estas pantallas están pensadas para que el paciente disponga de ayuda adicional, y una serie de recursos de fácil acceso así como una sección de noticias de la AECC para que puedan ser consultadas por el paciente. Sección de meditación, consejos generales que le puedan ser de ayuda, noticias y teléfonos de interés.

Pantalla de perfil de usuario El usuario podrá introducir sus datos personales para que la aplicación pueda tratarle de manera personalizada.

Otros: Pantalla de presentación y carga Pantalla inicial de carga, se utiliza para aportar más identidad a la aplicación y a modo de pequeña intro.

5.8. Flujo de navegación

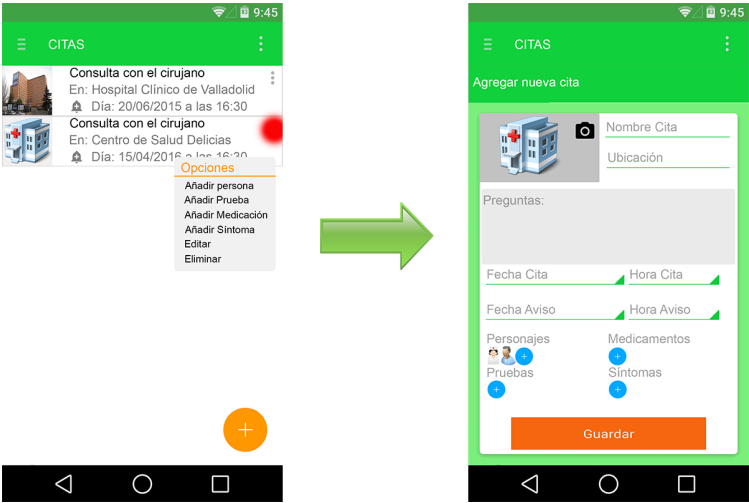


Figura 5.4: Pantalla con la actividad de las citas

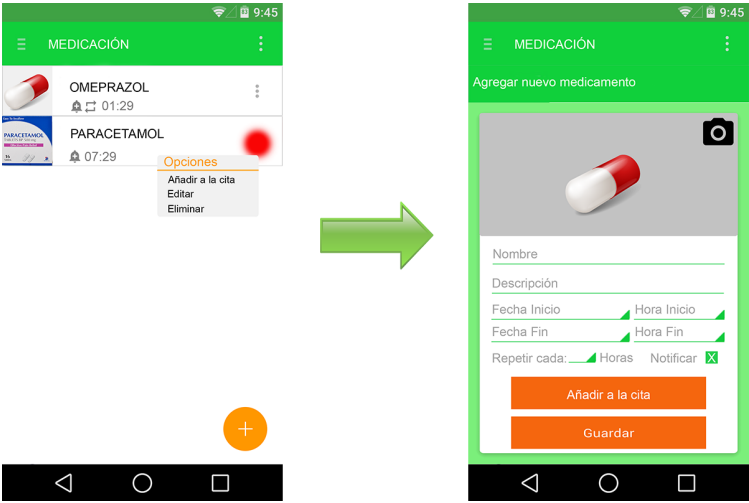


Figura 5.5: Pantalla con la actividad de la medicacion

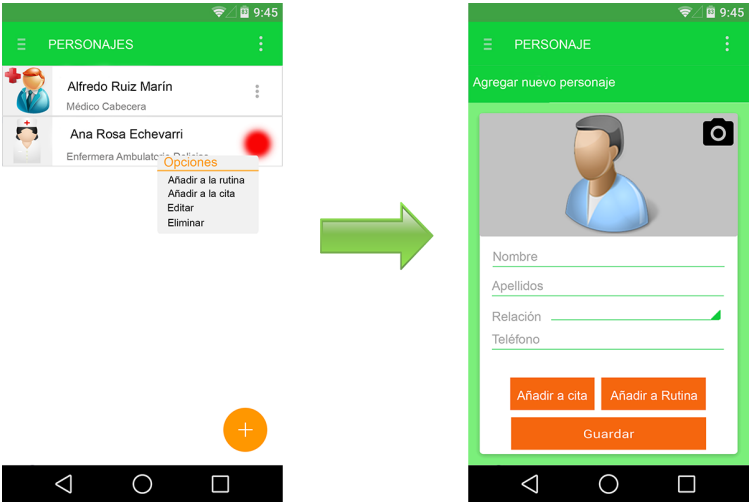


Figura 5.6: Pantalla con la actividad de los personajes

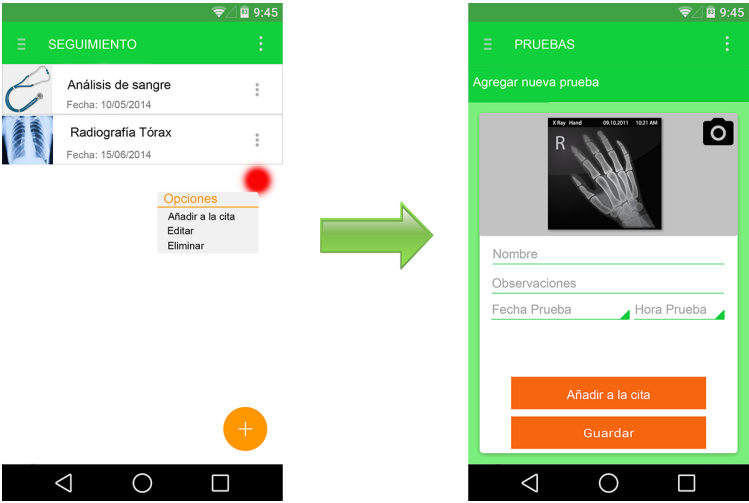


Figura 5.7: Pantalla con la actividad de las pruebas

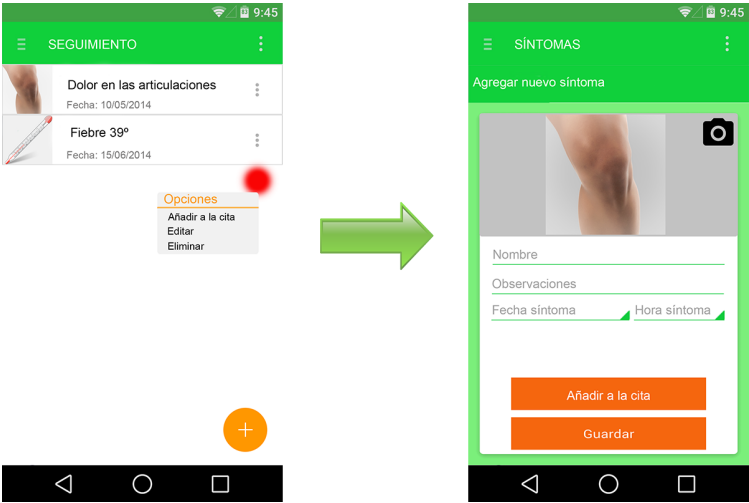


Figura 5.8: Pantalla con la actividad de los sintomas

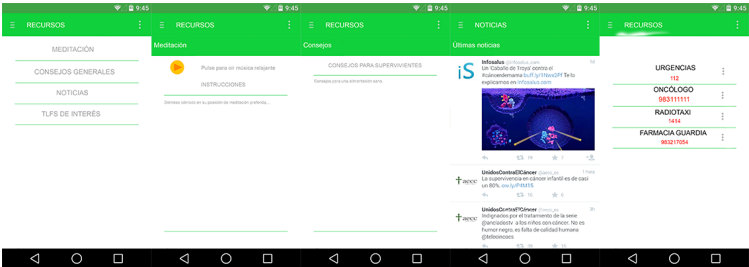


Figura 5.9: Pantalla con la actividad de los recursos

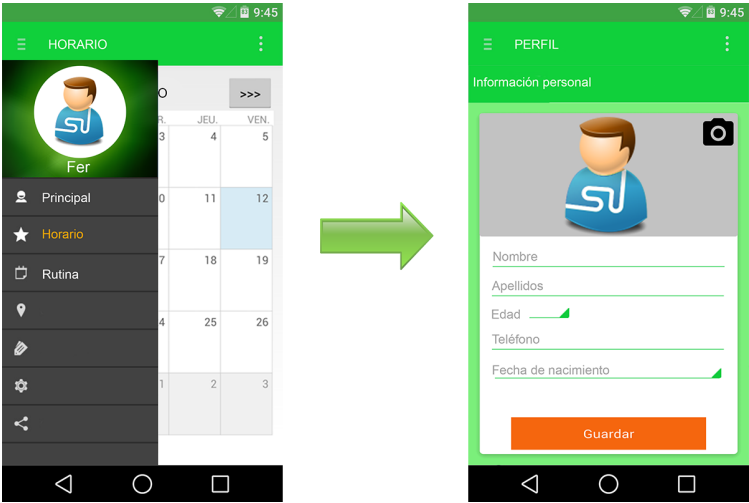


Figura 5.10: Pantalla con la actividad conf del perfil



Figura 5.11: Pantalla con la actividad de las citas

CAPÍTULO 6

Implementación y Pruebas

6.1. Construcción y Pruebas

Para la implementación en un principio las metodologías ágiles no marcaban ninguna pauta más allá de que todo lo que no está terminado al 100 % no está terminado y por otro lado que siempre hay que entregar con la máxima calidad posible, aunque quizá la más importante fue que la propiedad del código es colectiva, ya no hay más código mio o código tuyo y tenemos responsabilidades y atribuciones sobre nuestras parcelas, el código es del equipo entero y todo miembro puede y debe mejorar el mismo si observa y detecta un error.

Después de un tiempo de maduración de las mismas surgieron técnicas y disciplinas, algunas en el seno de alguna de estas metodologías (XP sobre todo) y algunas se incorporaron rápidamente desde el movimiento Craftmanship software (cita) que tiene una filosofía similar aunque más cercana y centrada en el proceso de construcción en sí que en el resto del proceso de desarrollo del proyecto o producto. Estas técnicas son Pair Programming, TDD, Code Reviews y algunas centradas en la mejora del equipo en todos los proyectos como puedan ser Code Retreats, Katas, Koans, etc(citas a gogo). todas estas técnicas y disciplinas hoy en día se consideran tan apegadas a las metodologías ágiles que precticamente se confunden con las mismas cuando hablamos del proceso de implementación.

6.2. Estructura de la aplicación

6.3. Plan de desarrollo

Pasos para el desarrollo de la aplicación móvil "Diario de un Superviviente":

Idea Inicial

Partimos de la 'idea' conformada junto a la AECC de una aplicación Android que permitiese a los enfermos de cáncer llevar de una manera mucho mas sencilla el control de las acciones que acontecen de manera rutinaria en su día a día, tales como la toma de medicamentos, asistencia a citas médicas, control de sintomas y pruebas, rutinas diarias beneficiosas.

Para ello la AECC puso a nuestra disposición un folleto que forma parte de esta iniciativa, pero que dadas sus características físicas se que da algo corto en su planteamiento.

Captura de requisitos

El proyecto debe estar bien definido, tanto sus objetivos como las funcionalidades que se requieren para que cumpla su cometido. Cuanta mejor definido esté más cerca estaremos de

cumplir sus objetivos.

Esta tarea, se ha realizado estudiando de manera pormenorizada la documentación que puso a nuestra disposición la AECC dividiendo en partes bien diferenciadas las funcionalidades de cada una de las partes de las que se compone esta documentación.

A través del cruce de correos y de alguna que otra reunión presencial se han limado distintas formas de ver algunas partes.

Con la definición del proyecto terminada, es necesario saber el tiempo que nos va a llevar en horas, por lo que hay que valorar el desarrollo. Para ello, será necesario contabilizar y estimar los plazos en horas que nos va a costar cada parte del proyecto. Tanto el plazo como el precio dependerán totalmente de las funcionalidades y del tipo de desarrollo elegido, pues no es lo mismo (ni se obtiene un proyecto de igual calidad) desarrollar apps nativas que híbridas, ni que el proyecto requiera de un complejo backend orientado a móviles o no requiera siquiera esta parte.

Planificación

Es la primera fase del desarrollo del proyecto. Consiste en tener un programa de trabajo con un desglose de todas las actividades que se van a realizar (desde el diseño hasta las pruebas finales), el plazo estimado de horas que se le va a dedicar cada una de ellas y estableciendo los medios humanos que se van a dedicar para alcanzar los objetivos que se hayan propuesto. En este proceso, que ha de ser continuo se han de reflejar

- Equipos, programas, licencias etc que se vayan a emplear.
- Requerimientos gráficos y fechas límite.
- Necesidades que dependan del cliente (AECC) y fechas para tenerlos disponibles.
- Cambios que puedan ocurrir durante el desarrollo de la app.

Una buena planificación y su actualización es clave para el correcto desarrollo de la aplicación móvil y para su puesta en funcionamiento en la fecha prevista.

Diseño UI/UX

Previo a la implementación es necesario tener totalmente definido el diseño estructural de la app y su comportamiento. Para ello hemos utilizado Photoshop para el diseño inicial, el cual nos mostrará el aspecto y la usabilidad de la aplicación.

El diseño consiste tanto en la confección del aspecto y usabilidad como en la correcta aplicación de las guidelines de diseño de google de material design[?]materialStructure), además de la correcta adaptación a todas las densidades de pantallas (recordemos que por ejemplo Android tiene MDPI(160 DPI), HDPI(240 DPI), XHDPI(320 DPI), XXHDPI(480 DPI), XXHDPI (640 DPI) y su tratamiento para que sean aptas para la programación.

Desarrollo

Es la programación del proyecto. Esta fase se hará de acuerdo a la tecnología que se haya decidido emplear para cada plataforma de programación y los entornos de desarrollo empleados serán acordes con ello (Android Studio); recordemos que se pueden desarrollar apps nativas o híbridas, y llevará mayor esfuerzo de trabajo en función de lo anterior. A la vista de lo anterior el equipo de desarrollo, de una aplicación, por muy sencilla que sea, puede llegar a estar compuesto por 5 ingenieros informáticos (Android, iOS, Windows Phone, Backend, Frontend) y un diseñador, además del director del proyecto que coordine a todos ellos. De ahí que el coste de una app sea totalmente dependiente de la tecnología que empleemos en el desarrollo y de la complejidad del proyecto en sí.

Testing

Una vez desarrollada la app es necesario hacer un testing profundo de todas las partes del mismo. El testeo se puede dividir en: -Testeo funcional: para asegurar que la aplicación trabaja como debería y sigue todos los flujos debidos. -Testeo de rendimiento: para comprobar que el comportamiento de la aplicación bajo ciertas condiciones (múltiples peticiones de acceso simultáneas, poca cobertura, poca batería...) es el correcto. -Comprobaciones de fugas de memoria, cruciales en móviles pues los recursos son mucho más limitados que en programas para ordenadores de sobremesa. Para esta tarea se utilizan habitualmente programas automatizadores de tareas y programas que reportan el código de error, además del testeo manual intensivo.

Distribución pre-lanzamiento

Previo a la subida a los markets de aplicaciones móviles se pueden hacer distribuciones de las aplicaciones móviles. En Android se puede hacer utilizando el entorno beta de desarrollo Android disponible en la consola de desarrollador.

Implantación y distribución

A la finalización del desarrollo, el último paso será subirlo a los markets correspondientes. Para este último paso habrá que firmar digitalmente las apps con la cuenta de desarrollador, compilar el paquete y subirlo a Google Play, así como preparar el resto de requisitos necesarios tales como las imágenes, logos, descripciones etc. Requeridos por los markets de apps. A partir de este momento comienza la etapa de mantenimiento de la aplicación, y su escalabilidad, dependiendo de los requerimientos y necesidades futuras de los usuarios o del propio cliente en nuestro caso de la AECC.

Esperamos que aclare las dudas que podáis tener cuando penséis en desarrollar una aplicación móvil.

6.4. SVN, GIT,

6.5. Plan de trabajo y comunicaciones

6.6. Pruebas

6.6.1. Plan de pruebas

6.6.2. Tipos de pruebas

6.6.3. Baterías de pruebas

6.6.4. Pruebas en el dispositivo

6.7. Puesta en producción

CAPÍTULO 7

Conclusiones y trabajo futuro

- 7.1. Conclusiones y trabajo futuro
- 7.2. Conclusiones
- 7.3. Dificultades encontradas
- 7.4. Consecución de Objetivos
- 7.5. Conocimientos adquiridos
- 7.6. Trabajo futuro

CAPÍTULO 8

Bibliografía

Bibliografía

- [1] O. M. de la Salud. Cáncer, nota descriptiva 297. URL <http://www.who.int/mediacentre/factsheets/fs297/es/>.
- [2] W. en español. Metodologías de desarrollo de software. URL https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software/.
- [3] D. frustrated by the restrictions of the traditional LaTeX environment. Editor online de latex. URL <https://es.sharelatex.com/>.
- [4] R. C. M. e. a. Kent Beck. Agile manifesto. URL <http://agilemanifesto.org/>.

ANEXO I: INSTALACIÓN Y MANUAL DE USUARIO

- 8.1. Conclusiones y trabajo futuro
- 8.2. Conclusiones
- 8.3. Dificultades encontradas
- 8.4. Consecución de Objetivos
- 8.5. Conocimientos adquiridos
- 8.6. Trabajo futuro

Android

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tablets; y también para relojes inteligentes, televisores y automóviles. Inicialmente fue desarrollado por Android Inc., empresa que Google respaldó económicamente y más tarde, en 2005, compró. Como curiosidad: Tanto el nombre Android (androide en español) como Nexus hacen alusión a la novela de Philip K. Dick ¿Sueñan los androides con ovejas eléctricas?, que posteriormente fue adaptada al cine como Blade Runner. Tanto el libro como la película se centran en un grupo de androides llamados replicantes del modelo Nexus-6.

Android Studio

Android Studio es un entorno de desarrollo integrado (IDE) para la plataforma Android. Fue anunciado por Ellie Powers el 16 de mayo de 2013. Android Studio esta disponible para desarrolladores para probarlo gratuitamente. Basado en IntelliJ IDEA de JetBrains, está diseñado específicamente para desarrollar para Android. Esta disponible para descargar para Windows, Mac OS X y Linux.

BBDD.

Una base de datos o banco de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso. Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente.

DAO.

Los Objetos de Acceso a Datos son un Patrón de Diseño y considerados una buena práctica. La ventaja de usar objetos de acceso a datos es que cualquier objeto de negocio (aquel que contiene detalles específicos de operación o aplicación) no requiere conocimiento directo del destino final de la información que manipula.

Dropbox.

Se trata de una herramienta de sincronización de archivos a través de un disco duro o directorio virtual. Permite disponer de un directorio de archivos de forma remota y accesible desde cualquier ordenador. Es decir, crea una carpeta en nuestro ordenador y realiza una copia a través de Internet de todos los archivos que depositemos en ella. Se ocupa de mantener la copia de nuestros archivos siempre sincronizada.

IDE (Integrated Development Environment).

Un entorno de desarrollo integrado es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios. Consiste en un editor de código, un

compilador, un depurador y un constructor de interfaz gráfica (GUI). Los IDEs pueden ser aplicaciones por sí solas o pueden ser parte de aplicaciones existentes.

Java.

Es un lenguaje de programación orientado a objetos y la primera plataforma informática creada por Sun Microsystems en 1995. Es la tecnología subyacente que permite el uso de programas punteros, como herramientas, juegos y aplicaciones de negocios. Tiene como principal característica ser un lenguaje independiente de la plataforma.

MVC.

Modelo Vista Controlador, es un patrón de arquitectura de software que separa los datos de la aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. Se ve frecuentemente en aplicaciones web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página. El modelo es el Sistema de Gestión de Base de Datos y la Lógica de negocio, y el controlador es el responsable de recibir los eventos de entrada desde la vista.

MVP.

Modelo Vista Presentador, es un patrón de arquitectura de software que se parece al MVC pero debido al gran acoplamiento existente en android y que el controlador y la vista puedan acabar fusionados en la Activity es más usado que el anterior (seguir).

MySQL.

MySQL es un sistema de administración de bases de datos para bases de datos relacionales. No es más que una aplicación que permite gestionar archivos llamados de bases de datos. Utilizado para almacenar todos los datos de interés de la aplicación y datos referentes de las simulaciones realizadas por los usuarios.

StarUML.

Es una herramienta para el modelamiento de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Arquitectura). Permite definir elementos propios para los diagramas, que no necesariamente tienen que pertenecer al estándar de UML.

SGBD (Sistema Gestor de Bases de Datos).

En inglés Data Base Management System (DBMS); es un tipo de software muy específico, dedicado a servir de interfaz entre la base de datos y el usuario y las aplicaciones que la utilizan. Se compone de un lenguaje de definición de datos, de un lenguaje de manipulación de datos y de un lenguaje de consulta.

USB (Universal Serial Bus).

Estándar de comunicaciones serie de alta velocidad. Vensim. Es una herramienta de modelaje que permite conceptualizar, documentar, simular, analizar y optimizar modelos de dinámica de sistemas. Vensim provee una forma simple y flexible de construir modelos de simulación, sean lazos causales o diagramas de stock y de flujo.

USB (Pair Programming).

Técnica (seguir).

TDD.

Tecnica.

BDD.

Tecnica.

Code Review.

tecnica.

Mockup.

Doble.

Code Smell.

mal codigo.