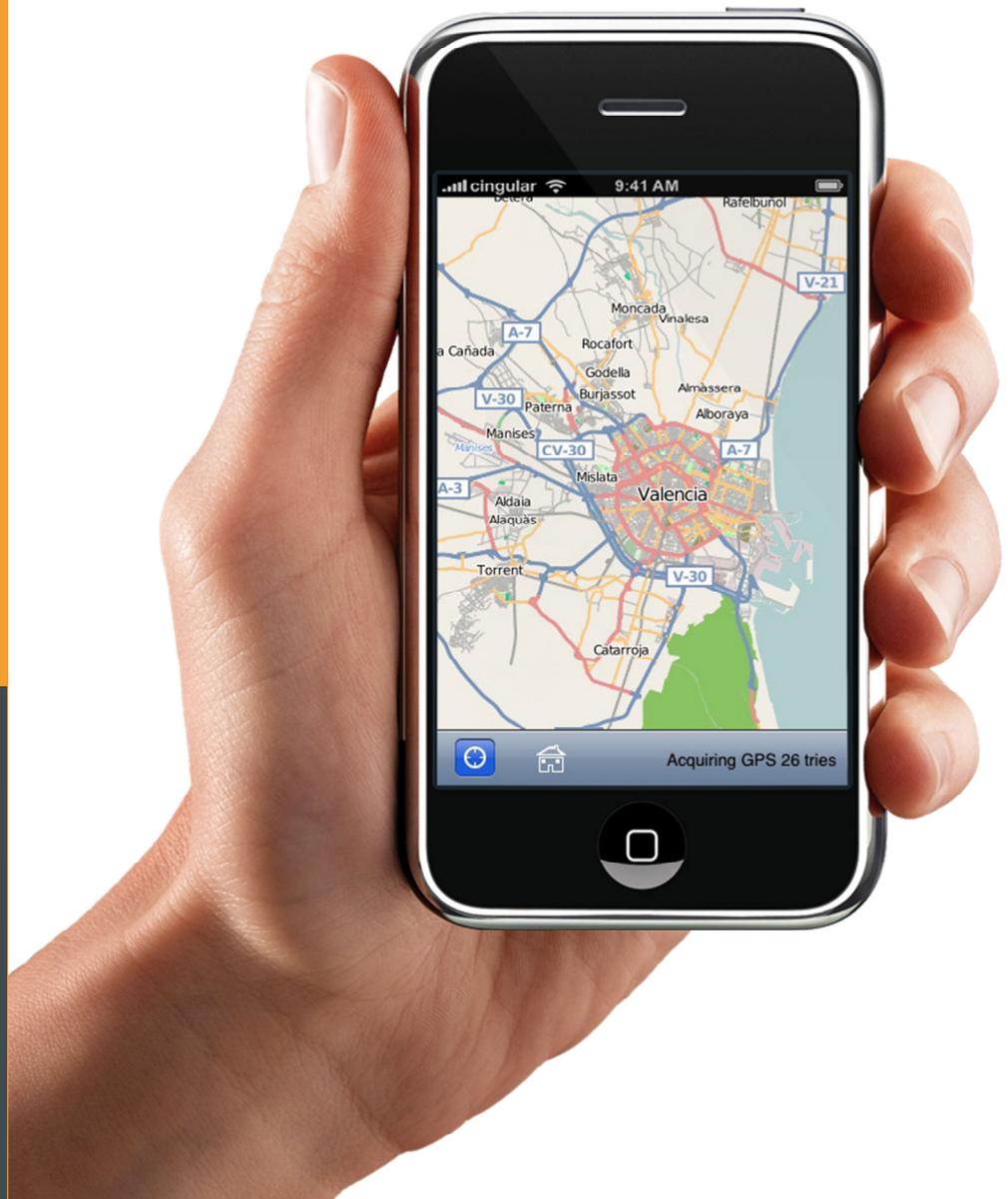




UNIVERSIDAD
POLITECNICA
DE VALENCIA



Un Enfoque Basado en Simplicidad para el Diseño de Aplicaciones Móviles Sensibles al Contexto



AUTOR:

Pablo Muñoz

SUPERVISOR:

Dr. Vicente Pelechano

“We made a big mistake 300 years ago when we separated technology and humanism. ... It's time to put the two back together.”

Michael Dertouzos

Resumen

La sólida implantación de los dispositivos móviles dentro de la vida cotidiana de las personas ha incrementado el consumo de una gran diversidad de aplicaciones móviles y servicios digitales, los cuales, se introducen en multitud de situaciones y actividades. De esta forma, el uso de estas aplicaciones y servicios provoca que en muchas ocasiones los usuarios desvíen su atención de la actividad que están realizando con el fin de interactuar con su dispositivo. Este hecho, sumado a las limitaciones de interacción introducidas por este tipo de dispositivos (ej. pantallas reducidas, mecanismos de introducción de datos limitados, etc.) muestra la necesidad de que las aplicaciones móviles proporcionen interfaces de usuario y mecanismos de interacción que faciliten el cumplimiento de los objetivos de los usuarios con el fin de que éstos puedan centrarse en sus actividades.

En este trabajo se presenta una propuesta que trata de hacer frente a esta necesidad. En particular, en este trabajo se proporcionan mecanismos que permiten refactorizar diseños de interfaz de usuario complejos en soluciones que tratan de simplificar las complejidades encontradas. Así, se define un conjunto de guías centradas en evitar complejidades en diseños de interfaz de usuario de aplicaciones móviles. Además, se propone la introducción de un rol con el que se persigue evitar posibles parcialidades de los diseñadores en el análisis de diseños de interfaz y en la detección de problemas de complejidad en la interfaz. De esta forma, la misión de este rol es aplicar el conjunto de guías provisto para detectar complejidades en los diseños provistos por los diseñadores y proponer soluciones simplificadas que resuelvan estos problemas. Asimismo, dada la heterogeneidad de aplicaciones móviles existentes y la dificultad de que un conjunto de guías pueda proveer soluciones concretas que simplifiquen cualquier diseño de interfaz, este trabajo instancia la propuesta de simplificación de interfaces al dominio de aplicaciones concreto de las aplicaciones móviles sensibles al contexto. Con este fin, este trabajo introduce un conjunto de patrones de diseño de interfaces de usuario de aplicaciones móviles sensibles al contexto, el cual, proporciona el conocimiento necesario para afrontar la simplificación de interfaces de aplicaciones móviles de este dominio.

Finalmente, se describe la experiencia obtenida en la simplificación de interfaces de usuario de un caso de estudio concreto.

Índice de Contenido

CAPÍTULO 1: INTRODUCCIÓN	1
1.1 MOTIVACIÓN.....	2
1.2 ESTABLECIMIENTO DEL PROBLEMA	3
1.3 CONTRIBUCIÓN	4
1.4 CONTEXTO DE LA TESIS	5
1.5 ESTRUCTURA DE LA TESIS	6
CAPÍTULO 2: DISPOSITIVOS MÓVILES Y USO DEL CONTEXTO	8
2.1 DISPOSITIVOS MÓVILES	8
2.1.1 Crecimiento en el Interés del Desarrollo de Aplicaciones.....	9
2.1.2 Capacidades Hardware	10
2.1.3 Funcionalidades Actuales.....	11
2.1.4 Limitaciones.....	13
2.1.5 Tendencias y Funcionalidades Futuras.....	13
2.2 USO DEL CONTEXTO EN APLICACIONES MÓVILES	15
2.2.1 Contexto: Definición y Beneficios.....	15
2.2.2 Obtención de Información de Contexto en Aplicaciones Móviles.....	16
2.2.3 Análisis del Uso de la Localización y el Tiempo en Aplicaciones Móviles	17
2.3 CONCLUSIONES.....	24
CAPÍTULO 3: TRABAJO RELACIONADO	25
3.1 PROCESO DE DISEÑO	26
3.2 ETAPA DE ANÁLISIS	27
3.2.1 Métodos para la Recolección de Información del Usuario	27
3.2.2 Representación de la Información Capturada en Artefactos.....	28
3.3 ETAPA DE DISEÑO	30
3.3.1 Factores a Tener en Cuenta en la Creación de Diseños	31
3.3.2 Heurísticas para el Diseño de Interfaces de Aplicaciones Móviles.....	34
3.4 ETAPA DE EVALUACIÓN	41
3.4.1 Métodos de Recogida de Información.....	42
3.4.2 Evaluación en el Laboratorio y en el Entorno Real.....	45
3.5 CONCLUSIONES.....	46

CAPÍTULO 4: SIMPLIFICACIÓN DE DISEÑOS DE INTERFAZ	49
4.1 VISTA GLOBAL DE LA SOLUCIÓN PROPUESTA.....	50
4.2 INTRODUCCIÓN DE LA SIMPLICIDAD EN EL PROCESO DE DISEÑO	51
4.2.1 <i>Guías para Proveer Simplicidad.....</i>	<i>52</i>
4.2.2 <i>El Papel del Rol Simplificador.....</i>	<i>56</i>
4.2.3 <i>El Refinamiento de Interfaces de Usuario.....</i>	<i>57</i>
4.2.4 <i>Discusión.....</i>	<i>60</i>
4.3 PATRONES DE DISEÑO DE APLICACIONES MÓVILES SENSIBLES AL CONTEXTO.....	61
4.3.1 <i>Análisis Realizado.....</i>	<i>62</i>
4.3.2 <i>Patrones Detectados</i>	<i>64</i>
4.3 CONCLUSIONES.....	83
CAPÍTULO 5: CASO DE ESTUDIO.....	85
5.1 DESCRIPCIÓN GENERAL DE LA APLICACIÓN	85
5.2 DISEÑOS INICIALES PROPUESTOS.....	86
5.3 DETECCIÓN DE COMPLEJIDADES.....	90
5.3.1 <i>Componente de Interacción Sobrecargado</i>	<i>90</i>
5.3.2 <i>Refinamiento del Componente de Interacción Simplificado</i>	<i>92</i>
5.3.3 <i>Información no Relevante en la Vista Timeline View.....</i>	<i>92</i>
5.3.4 <i>Sobrecarga de Componentes en la Visualización Timeline View</i>	<i>94</i>
5.3.5 <i>Falta de Intuición en la Visualización Timeline View</i>	<i>96</i>
5.3.6 <i>Falta de Contexto en la Visualización Timeline View</i>	<i>97</i>
5.3.7 <i>Falta de Contexto en la Visualización Map View.....</i>	<i>98</i>
5.3.8 <i>Solapamiento de Puntos en la Visualización Map View.....</i>	<i>99</i>
5.3.9 <i>Reducción de Interacciones Innecesarias.....</i>	<i>101</i>
5.3.10 <i>Falta de Contexto en el Cambio de Visualización</i>	<i>102</i>
5.4 CONCLUSIONES.....	103
CAPÍTULO 6: CONCLUSIONES.....	105
6.1 CONTRIBUCIONES.....	105
6.2 PUBLICACIONES.....	106
6.3 TRABAJOS FUTUROS	107
ANEXO I.....	109

Listado de Figuras

Figura 1. Resultados del análisis de usos de la localización y el tiempo en aplicaciones móviles	19
Figura 2. Vista de la aplicación móvil “Google Latitude”	20
Figura 3. Vista de la aplicación móvil “Around Me”	20
Figura 4. Vista de la aplicación móvil “Buscar mi iPhone”	21
Figura 5. Vista de la aplicación móvil “Nike+ GPS”	21
Figura 6. Vista de la aplicación móvil “Neer”	22
Figura 7. Vista de la aplicación móvil “Trenes”	23
Figura 8. Vista de la aplicación móvil “Google Calendar Client”	23
Figura 9. Tareas y técnicas de la etapa de análisis	27
Figura 10. Ejemplo del artefacto Persona	29
Figura 11. Tareas y recursos disponibles en la etapa de diseño	30
Figura 12. Modelo de contexto propuesto por Savio y Braiterman	34
Figura 13. Tareas y técnicas de la fase de evaluación	42
Figura 14. Sistemas de grabación propuestos por Schusterich y otros	43
Figura 15. Visión global de la solución propuesta	50
Figura 16. Ejemplo de aplicación de las guías de simplicidad nº1 y 2	53
Figura 17. Ejemplo de aplicación de las guía de simplicidad nº3	54
Figura 18. Interfaz de usuario que requiere la espera del usuario	55
Figura 19. Ejemplo de aplicación de guías nº6 y 7	56
Figura 20. Paso nº1 del refinamiento de diseños de interfaz de usuario	58
Figura 21. Paso nº2 del refinamiento de diseños de interfaz de usuario	59
Figura 22. Paso nº3 del refinamiento de diseños de interfaz de usuario	60
Figura 23. Diseño inicial de la vista Timeline View	87
Figura 24. Detalle del componente gráfico mostrado tras la selección de una tarea	87
Figura 25. Diseño inicial de la vista Timeline View	88
Figura 26. Diseño inicial de la vista Map View	89
Figura 27. Filtro temporal y de prioridad	90
Figura 28. Simplificaciones realizadas en un componente gráfico	91
Figura 29. Simplificación realizada para proveer una navegación más intuitiva	92

Figura 30. Información no relevante o redundante de la vista Timeline View.....	93
Figura 31. Vista Timeline View tras eliminar información no relevante o redundante.....	94
Figura 32. Introducción de gestos en la interacción de la vista Timeline View.....	95
Figura 33. Eliminación de flechas conectoras y barra de desplazamiento horizontal.....	96
Figura 34. Cambios realizados en Timeline View para aprovechar mejor el uso del tiempo.....	98
Figura 35. Introducción de localización del usuario en la vista Map View.....	99
Figura 36. Ejemplo de agrupamiento de iconos sobre un mapa.....	100
Figura 37. Mecanismo de interacción para completar una tarea en Timeline View.....	102

Listado de Tablas

Tabla 1. Perspectiva sobre la venta de dispositivos móviles según plataforma.....	10
Tabla 2. Estadísticas sobre el uso actual de los dispositivos móviles.....	12
Tabla 3. Factores que afectan a la usabilidad de una aplicación móvil.....	32
Tabla 4. Leyes de la Simplicidad de John Maeda	51

Capítulo 1

Introducción

Con la aparición de los primeros computadores, las personas incrementaron la eficiencia de algunos procesos. De esta forma, la informatización de sistemas de información hizo más plausible y productiva la ejecución de determinadas tareas en entornos de trabajo fijos. Posteriormente, la llegada de nuevos dispositivos de menor tamaño y de carácter móvil como son por ejemplo las PDAs o los smartphones, ha abierto nuevos escenarios en los que es posible acceder a sistemas de información desde cualquier localización. Así, estos dispositivos se han convertido en los candidatos idóneos para ser la interfaz física con la que las personas interactúen con aplicaciones ubicuas [5].

La mejora en las prestaciones de estos dispositivos (ej. mejores comunicaciones, inclusión de sensores, etc.), ha dado pie al incremento de la oferta y demanda de servicios digitales y aplicaciones instalables que cubren una gran variedad de funcionalidades [7]. Como consecuencia, el uso de los dispositivos móviles se intercala en cada vez más actividades cotidianas [55], introduciéndose en muchos casos en entornos donde la atención del usuario es un recurso limitado en comparación con entornos tradicionales y fijos [45]. Este hecho, sumado a la existencia de interfaces de usuario complejas que dificultan la interacción, hace que el diseño de interfaces sea un factor determinante a tener en cuenta para alcanzar el éxito de una aplicación [66].

La obtención de una interacción natural entre los usuarios y los sistemas software se promueve desde la *Computación Ubicua*. En particular, esta área de investigación se refiere a entornos electrónicos que son receptivos a la presencia de las personas, y que se componen de pequeños dispositivos que dan soporte de una forma amigable a información ubicua, comunicación y entretenimiento con el objetivo de ayudar a las

personas en sus actividades cotidianas. Sin embargo, la introducción de este paradigma en entornos reales no es una labor sencilla, y por ello, es necesario afrontar varios desafíos. Algunos de ellos son (1) entender el contexto de personas que realizan actividades diferentes, que desempeñan diferentes roles y que son de diferente edad, sexo o cultura; (2) decidir, obtener y proporcionar aquella información y funcionalidad adecuada para que el usuario desarrolle sus actividades; y (3) proveer una interacción natural, sencilla y no intrusiva.

Centrando la visión en las aplicaciones móviles, **este trabajo enfoca sus aportaciones hacia este tercer reto**. Dadas las características particulares del acceso y uso de un sistema de información desde cualquier lugar (ej. captura y tratamiento del contexto del usuario, uso intercalado de dispositivos entre actividades, atención del usuario limitada, distracciones, etc.), nos encontramos ante un escenario en el que los usuarios valoran de forma positiva los diseños sencillos [8, 17]. Sin embargo, la obtención de interfaces de usuario de aplicaciones móviles es una labor compleja [20, 38]; y como consecuencia de esto, los resultados finales proporcionados por los diseñadores tienden a alejarse en muchas ocasiones de la visión del usuario final. Bajo este escenario, este trabajo trata de ayudar en la tarea de **obtener diseños de interfaz de usuario de aplicaciones móviles sensibles al contexto, tomando un enfoque basado en simplicidad que persigue facilitar la interacción del usuario con su dispositivo**.

El resto del capítulo se organiza de la siguiente manera: la sección 1.1 explica la motivación de este trabajo. En la sección 1.2 se establecen los problemas tratados en este trabajo. La solución propuesta para cubrir dichos problemas se resume en la sección 1.3. En la sección 1.4 se explica el contexto en el que ha sido realizado este trabajo. Finalmente, en la sección 1.5 se estructura el contenido del resto del documento.

1.1 Motivación

Las diferencias entre los tradicionales computadores de sobremesa y los dispositivos móviles actuales [27, 33, 52] deben ser tenidas en cuenta en el diseño de interfaces de usuario.

En primer lugar, los mecanismos de interacción proporcionados por los dispositivos móviles son limitados, y por tanto, dificultan el cumplimiento de los objetivos de los usuarios. Así, limitan la introducción de datos debido a los pequeños teclados, a los pocos tipos de operaciones que las pantallas táctiles permiten realizar con

los dedos, o al uso del dispositivo con una única mano; y además, limitan la representación de información debido al tamaño de las pantallas que afecta al tamaño y a la cantidad de componentes gráficos que pueden tener cabida en la interfaz.

En segundo lugar, la atención del usuario se ve afectada por el contexto móvil y cambiante en el que se usan los dispositivos móviles, el cual introduce nuevas limitaciones. Algunas de ellas se deben a los frecuentes cambios producidos por el propio usuario, quien puede cambiar de localización, de actividad o de rol. Aparte, otras se deben a cambios que tienen lugar en el entorno como pueden ser cambios en la luminosidad, en el nivel de ruido, en las personas u objetos cercanos, y que pueden dar lugar a distracciones o interrupciones.

Estos problemas muestran la necesidad de proporcionar interacciones que permitan que el uso de las aplicaciones se integre mejor dentro de otras actividades para que el usuario pueda centrarse en la actividad que está realizando en la vida real.

Para alcanzar esta finalidad, la incorporación de diversos sensores permite extraer información de contexto del usuario que da lugar a la construcción de servicios y aplicaciones más útiles [12]. Así, el uso de esta información contextual puede ser aprovechado para adecuar los diseños de interfaz para que éstos sean afines a las diversas situaciones del usuario, centrando cada interfaz en aquella información relevante para el usuario en cada contexto o situación.

Por otra parte, la creación de diseños de interfaz de usuario viene guiada en muchas ocasiones por heurísticas y recomendaciones. Sin embargo, algunas de estas guías son adaptaciones de guías de diseño de aplicaciones de escritorio, y otras, aunque han sido adaptadas para guiar la obtención de diseños de aplicaciones móviles, no hacen hincapié en la necesidad de introducir la simplicidad como factor determinante en la creación de soluciones. Además, la toma de decisiones de diseño en un equipo de diseñadores se ve afectada en ocasiones por problemas de comunicación [64], y aparte, el exceso de atención a los requerimientos del usuario puede llevar a los diseñadores a aumentar la complejidad de los diseños obtenidos y a una falta de cohesión [44].

1.2 Establecimiento del Problema

El diseño de interfaces de usuario para aplicaciones móviles es uno de los campos a tratar dentro de la *Computación Ubicua*. La discusión anterior indica que varios problemas necesitan los esfuerzos de la comunidad.

Por una parte, la aplicación informal de heurísticas en la creación de diseños de interfaz [22] sumado al hecho de que los diseñadores tienen dificultad en la detección de problemas en sus propios diseños [34] provoca que en muchas ocasiones las soluciones finales no sean fáciles de usar. Además, la variedad de limitaciones introducidas por el contexto móvil en el que se usan estos dispositivos así como las propias limitaciones de interacción que introducen complican todavía más esta labor. Así, uno de los problemas que se detecta es **la necesidad de mecanismos que ayuden a obtener diseños de interfaz sencillos.**

Por otra parte, los dispositivos actuales cuentan con la capacidad de poder hacer uso de determinada información del contexto del usuario, sin embargo, y pese a que existen algunas aplicaciones que utilizan algunos atributos como la localización o el tiempo, no existe ninguna especificación concreta que oriente a los diseñadores a sacar el mejor provecho posible de estos atributos del contexto del usuario dentro de las aplicaciones móviles. Así, es **necesario detectar y catalogar los posibles usos del contexto dentro de los diseños de interfaz de aplicaciones móviles para facilitar la creación de diseños de estas aplicaciones.**

Este trabajo trata de facilitar la obtención de diseños de interfaces de usuario de aplicaciones móviles sensibles al contexto. En el siguiente punto se describe la solución propuesta con la que este trabajo contribuye a la resolución de los problemas detectados.

1.3 Contribución

Las contribuciones de este trabajo tratan de facilitar la obtención de diseños de interfaz de usuario de aplicaciones móviles sensibles al contexto; introduciendo en dicha obtención, un enfoque basado en Simplicidad.

El principal objetivo de este trabajo es proporcionar mecanismos que permitan refactorizar diseños de interfaz de usuario complejos en soluciones que tratan de simplificar las complejidades encontradas. Así, **la primera y principal aportación** consiste en la definición de un conjunto de guías centradas en evitar complejidades en diseños de interfaz de usuario de aplicaciones móviles.

Sin embargo, dada la amplitud del espectro de aplicaciones móviles existente, la aplicación del conjunto de guías sobre una aplicación concreta requiere un conocimiento profundo del dominio de aplicación y de los diseños de interacción que suelen utilizarse en dicho dominio. Así, este conocimiento ayudará a detectar complejidades y a

simplificar los diseños provistos. Por ello, la **segunda aportación** de este trabajo es la definición de un conjunto de patrones de diseño de aplicaciones móviles sensibles al contexto. Dicho conjunto de patrones ha sido extraído a partir del análisis de aplicaciones móviles representativas que usan los siguientes atributos del contexto del usuario: localización y tiempo. Estos patrones representan el comportamiento común de los diseños de interacción de este conjunto de aplicaciones, los cuales centran la información mostrada en los atributos de contexto relevantes para el usuario. Así, dentro de nuestro dominio de aplicación, este conjunto de patrones ayuda a determinar cuándo una interfaz debe ser simplificada indicando una posible solución.

Por otra parte, otro de los objetivos de este trabajo es evitar la parcialidad de los diseñadores en el análisis de diseños de interfaz y en la detección de complejidades. Por ello, la **tercera aportación** consiste en la introducción de un rol ajeno a los requerimientos del sistema cuya misión consiste en detectar complejidades en los diseños provistos por el diseñador y proponer soluciones simplificadas a las mismas a través de la aplicación del conjunto de guías que se propone. De esta manera, y mediante la intervención iterativa de este rol dentro del proceso de diseño, los diseños iniciales se refinan de manera progresiva hasta que son completamente validados.

Igualmente, la propuesta de este trabajo ha sido puesta en práctica en un caso de estudio en el cual se ha diseñado una aplicación que permite a los usuarios gestionar una lista de tareas. A través de este caso de estudio, se describe la experiencia obtenida en la refactorización de diseños y cómo los diseños de interacción iniciales fueron cambiando a lo largo de sucesivas iteraciones hasta la definición de los diseños finales.

1.4 Contexto de la Tesis

Esta Tesis de Máster ha sido desarrollada en el contexto del Centro de Investigación en Métodos de Producción de Software de la Universidad Politécnica de Valencia. El trabajo que ha hecho posible el desarrollo de esta tesis está bajo en el marco de los siguientes proyectos de investigación del gobierno:

- **EVERYWARE:** Construcción de Software Adaptativo para la Integración de Personas, Servicios y Cosas usando Modelos en Tiempo de Ejecución. Proyecto CYCIT referenciado como TIN2010-18011.
- **SESAMO:** Construcción de Servicios Software a partir de Modelos. Proyecto CYCIT referenciado como TIN2007-62894.

1.5 Estructura de la Tesis

El resto del documento se organiza de la siguiente manera:

- El Capítulo 2 sienta los antecedentes relacionados con la implantación de los dispositivos móviles y la utilización del contexto dentro de las aplicaciones móviles actuales.
- El Capítulo 3 presenta los trabajos relacionados con la propuesta, los cuales se clasifican a lo largo del proceso de diseño de interfaces de usuario.
- El Capítulo 4 describe la solución propuesta para el diseño de interfaces de usuario de aplicaciones móviles sensibles al contexto. Así, en este apartado se presenta (1) un conjunto de guías cuyo objetivo es crear interfaces de usuario simples, (2) un rol Simplificador cuya misión es simplificar los diseños de interfaz provistos por el diseñador y (3) un conjunto de patrones de diseño de interfaces de aplicaciones móviles sensibles al contexto que sirve para acotar la propuesta a un ámbito de aplicación concreto.
- El Capítulo 5 describe la experiencia obtenida en la aplicación de la propuesta sobre un caso de estudio concreto.
- El Capítulo 6 resume las contribuciones de este trabajo así como las publicaciones obtenidas. Aparte, presenta posibles trabajos futuros.
- El Anexo I muestra el listado de aplicaciones analizado al que hace referencia el Capítulo 2, así como los resultados del mismo.

Capítulo 2

Dispositivos Móviles y Uso del Contexto

Para mejorar el entendimiento de este trabajo, este capítulo describe varios antecedentes que sirven de base para los capítulos posteriores. En concreto, este capítulo refleja el auge del desarrollo de aplicaciones móviles, las características de los dispositivos actuales, su capacidad para obtener determinada información de contexto del usuario y cómo este contexto se utiliza en varias aplicaciones actuales.

El resto del capítulo se estructura de la siguiente forma: la sección 2.1 presenta el estado los dispositivos móviles actuales refiriéndose a su implantación en la sociedad, a sus capacidades, posibilidades de uso, limitaciones y futuras tendencias. La sección 2.2 define el concepto de contexto y sintetiza aquellas características que permiten identificar a una aplicación como aplicación sensible al contexto. Además, resume las principales fuentes de información de contexto del usuario disponibles en los dispositivos móviles actuales, y el tipo de información contextual que puede derivarse a partir de ellas. Asimismo, describe un análisis de aplicaciones móviles de aplicaciones que se basan en la localización del usuario y el tiempo, el cual tiene como objetivo detectar cómo se utilizan estos atributos dentro de las aplicaciones. Finalmente, la sección 2.3 concluye el capítulo.

2.1 Dispositivos Móviles

Los dispositivos móviles se han implantado en la vida diaria como nunca antes había ocurrido con ningún otro dispositivo [16]. Así, con una tendencia de ventas en alza, se

estima que actualmente ya son más de 5 billones los dispositivos móviles que se utilizan en todo el mundo¹.

En sus inicios, los dispositivos móviles comenzaron siendo simples aparatos que permitían realizar llamadas y enviar mensajes. Sin embargo, con el paso del tiempo, estos dispositivos han añadido otras utilidades a su abanico de funcionalidades. De esta forma, la inclusión de agendas, calendarios, alarmas, y otras funciones en un único dispositivo ha hecho que progresivamente las personas den cada vez más una mayor utilidad a su dispositivo móvil en detrimento de a otros objetos o aparatos que paulatinamente parecen perder utilidad (ej. relojes, despertadores, radios, agendas,...). Aun así, los dispositivos actuales, acuñados comercialmente con el término inglés “*smartphone*”, continúan incluyendo nuevas funcionalidades más complejas y completas. De este modo, es común que éstos den soporte al uso de clientes de correo electrónico, a la conexión con redes sociales, a la navegación web, o incluso a la instalación de aplicaciones realizadas por terceros con el fin de cubrir todo tipo de necesidades.

2.1.1 Crecimiento en el Interés del Desarrollo de Aplicaciones

La sólida implantación de los dispositivos móviles ha dado lugar a un mercado en fase de explotación que genera interés económico en agentes de diverso tipo.

Por una parte, genera interés en las grandes compañías que se disputan el mercado de los smartphones: tanto en aquellas que fabrican dispositivos como en aquellas que proveen las principales plataformas software. En concreto, estas plataformas son según los datos publicados por la consultora Gartner², los cuales se muestran en la Tabla 1, por orden de número de dispositivos vendidos a lo largo del año 2011: Android (perteneciente a la compañía Google), Symbian (perteneciente a la compañía Nokia), iOS (perteneciente a la compañía Apple), Research in Motion (dueña de los dispositivos Blackberry) y Windows Phone (perteneciente a la compañía Microsoft).

Aparte, genera interés tanto en compañías de desarrollo de software como en programadores independientes que ven en este nicho de mercado, la posibilidad de crear

¹ http://www.itu.int/net/pressoffice/press_releases/2010/39.aspx

² <http://www.gartner.com/it/page.jsp?id=1622614>

sus propias aplicaciones móviles y de encontrar beneficios mediante su venta en tiendas de aplicaciones online o mediante la inclusión de publicidad en aplicaciones gratuitas.

Y así, también genera interés en anunciantes y compañías especializadas en marketing, que ven en este nuevo mercado de dispositivos y aplicaciones un nuevo medio donde invertir en publicidad en detrimento de otros medios tradicionales como son por ejemplo la televisión o la prensa.

OS	2010	2011	2012	2015
Symbian	111,577	89,930	32,666	661
Market Share (%)	37.6	19.2	5.2	0.1
Android	67,225	179,873	310,088	539,318
Market Share (%)	22.7	38.5	49.2	48.8
Research In Motion	47,452	62,6	79,335	122,864
Market Share (%)	16.0	13.4	12.6	11.1
iOS	46,598	90,56	118,848	189,924
Market Share (%)	15.7	19.4	18.9	17.2
Microsoft	12,378	26,346	68,156	215,998
Market Share (%)	4.2	5.6	10.8	19.5
Other Operating Systems	11,417.4	18,392.3	21,383.7	36,133.9
Market Share (%)	3.8	3.9	3.4	3.3
Total Market	296,647	467,701	630,476	1,104,898

Tabla 1. Perspectiva de la consultora Gartner sobre la venta de dispositivos móviles según plataforma

2.1.2 Capacidades Hardware

Progresivamente, los dispositivos móviles han mejorado sus capacidades hardware facilitando así su incorporación a cada vez más actividades al permitir nuevas y numerosas funcionalidades. Brevemente, entre algunas de estas mejoras encontramos nuevas capacidades de conexión y mecanismos de interacción, la adición de sensores o el aumento de la capacidad de almacenamiento.

Respeto a las nuevas **capacidades de conexión**, los dispositivos móviles han mejorado sus capacidades mediante la inclusión de la tecnología Bluetooth, Wi-Fi, y servicios de conexión de datos 3G. De esta forma, es posible conectar entre sí dos dispositivos móviles, un dispositivo móvil con un ordenador, así como también es posible realizar pequeñas transferencias de datos entre dispositivos. Aparte, también es posible el acceso a Internet desde cualquier lugar donde haya una red Wi-Fi inalámbrica

(ej. hogar, universidad, aeropuerto), o desde cualquier punto en el que haya cobertura telefónica gracias a servicios de conexión de datos 3G.

Respecto a los nuevos **mecanismos de interacción** encontramos: la inclusión de pantallas táctiles, la utilización de gestos, y sintetizadores y reconocedores de voz. Por una parte, las pantallas táctiles han sustituido en muchos dispositivos a los convencionales teclados de botones. Así, los usuarios pueden interactuar directamente a través de la utilización de sus dedos con los objetos mostrados en pantalla, o pueden interactuar mediante la utilización de gestos o movimientos asociados a diversas funcionalidades (ej. ampliar y reducir vista, arrastrar un objeto). Por otra parte, la incorporación de sintetizadores y de sistemas de reconocimiento de voz hace posible una interacción vocal en este tipo de dispositivos, dando soporte de esta forma tanto a la introducción de datos (ej. escritura de SMS) como a la invocación de acciones concretas a través de comandos de voz.

Finalmente, respecto a la **incorporación de sensores** en los dispositivos, encontramos que principalmente se han introducido sensores GPS que permiten conocer la ubicación del dispositivo, y acelerómetros que permiten la detección de la posición del dispositivo así como los movimientos realizados con éste.

2.1.3 Funcionalidades Actuales

Las progresivas mejoras en las capacidades hardware de los dispositivos móviles anteriores han modificado y ampliado los fines para los que éstos son usados. De forma detallada, en la Tabla 2 se muestran los usos más frecuentes de estos dispositivos a través de una comparación realizada por la consultora comScore³ entre usuarios de Estados Unidos, Europa y Japón. En dichos datos podemos ver como cada vez más personas están conectadas mediante sus dispositivos con redes sociales, servicios multimedia, de información, o de búsqueda.

Otras fuentes reafirman estos datos. Por una parte, las búsquedas realizadas desde dispositivos móviles en el buscador de Google se han quintuplicado en los dos

3

http://www.comscore.com/Press_Events/Press_Releases/2010/10/comScore_Release_First_Comparative_Report_on_Mobile_Usage_in_Japan_United_States_and_Europe

últimos años⁴. De esta forma, casi 530 millones de personas han utilizado su teléfono móvil para realizar búsquedas⁵. Asimismo, el creciente uso de redes sociales hace que la actual red social más importante, Facebook, cuente con más de 250 millones de usuarios que acceden a sus perfiles a través de su teléfono móvil⁶. Igualmente, otro de los usos frecuentes de estos dispositivos es el acceso a contenido multimedia (ej. canciones, videos, fotografías), el cual puede ser consultado vía “*streaming*” o puede almacenarse en el propio dispositivo gracias a las mejoras en la capacidad de almacenamiento.

Select Mobile Behaviors in: Japan, United States and EU5 (UK, DE, FR, ES and IT). June 2010 ,			
	Percent of Total Mobile Audience		
	Japan	United States	Europe
Total Audience: 13+ yrs old	100.0%	100.0%	100.0%
Used connected media (Browsed, Accessed Applications or	75.2%	43.7%	38.5%
Used browser	59.3%	34.0%	25.8%
Used application	42.3%	31.1%	24.9%
Messaging Usage			
Sent text message to another phone	40.1%	66.8%	81.7%
Used major instant messaging service	3.3%	17.2%	12.6%
Used email (work or personal)	54.0%	27.9%	18.8%
Social Media/Entertainment			
Accessed Social Networking Site or Blog	17.0%	21.3%	14.7%
Listened to music on mobile phone	12.5%	13.9%	24.2%
Took photos	63.0%	50.6%	56.8%
Captured video	15.4%	19.2%	25.8%
Watched TV and/or video on mobile phone	22.0%	4.8%	5.4%
Played games	16.3%	22.5%	24.1%
Financial Information			
Accessed bank accounts	8.0%	9.4%	7.1%
Accessed financial news or stock quotes	16.1%	10.0%	7.2%
Retail/Travel			
Accessed online retail	7.2%	5.5%	4.1%
Accessed classifieds	4.2%	6.6%	4.2%
Accessed travel Service	3.3%	4.7%	4.1%
Other			
Accessed maps	15.7%	16.0%	10.8%
Accessed traffic reports	12.6%	8.2%	5.9%
Accessed weather	34.1%	22.3%	13.7%

Tabla 2. Estadísticas de la consultora comScore sobre el uso actual de los dispositivos móviles en Japón, Estados Unidos y Europa

⁴ Jonathan Rosenberg, SVP, Google (October 2010)

⁵ <http://www.mobilemarketingwatch.com/google-mobile-search-sees-major-gains-in-two-years-10335/>

⁶ <http://www.facebook.com/press/info.php?statistics>

2.1.4 Limitaciones

Pese a las mejoras de las capacidades de los dispositivos móviles, el empleo de dispositivos móviles y de aplicaciones en cualquier lugar y entorno, así como el reducido tamaño de los dispositivos introducen limitaciones en la interacción.

En primer lugar, hay que tener en cuenta que las operaciones o acciones que realiza un usuario con su dispositivo suelen ser actividades secundarias que se desarrollan en un determinado momento en el cual se está llevando a cabo una actividad principal como puede ser conducir, estar reunido, realizar la compra, subir al autobús, etc. De esta forma, la aparición de distracciones o de interrupciones dentro de esta actividad principal desvía la atención del usuario de la interacción con el dispositivo y perjudica el cumplimiento de los objetivos del usuario dentro de la aplicación utilizada. Asimismo, otras características del entorno como el nivel de luminosidad o de ruido pueden afectar tanto a la visualización de interfaces de usuario como a la utilización de interfaces de usuario que hacen uso del reconocedor y sintetizador de voz.

En segundo lugar, respecto a los mecanismos de interacción con los dispositivos móviles, hay que tener en cuenta las limitaciones existentes para la introducción de información ya que ésta se ve afectada por pequeñas teclas provistas a través de cada vez menos teclados numéricos y teclados QWERTY o por pequeños botones mostrados en una pantalla táctil. De la misma forma, las pantallas introducen limitaciones debido a su reducido tamaño, el cual introduce problemas en la visualización y comprensión de las interfaces de usuario, ya que pueden agrupar demasiados componentes en poco espacio dificultando la interacción con las aplicaciones.

2.1.5 Tendencias y Funcionalidades Futuras

La demanda de dispositivos móviles con cada vez más servicios y la proliferación de aplicaciones para usos muy diversos indica que el uso de estos dispositivos y de las aplicaciones móviles seguirá en auge en los próximos años. Por una parte, se prevé un incremento en la venta de smartphones que dará lugar a que estos dispositivos sustituyan a los ordenadores de sobremesa como los principales puntos de acceso a Internet. Además, a su vez se prevé un aumento en la descarga de aplicaciones, el cual se estima que llegue a su pico en el año 2013 según *ABI Research*⁷.

⁷ <http://www.abiresearch.com/press/1651-Smartphone+Downloads+from+Mobile+App+Stores+to+Peak+in+2013>

Estas tendencias, junto a la inclusión de los dispositivos móviles en cada vez más escenarios dentro de las actividades de las personas, hace pensar que el uso de los mismos seguirá cambiando en un futuro próximo a medida que madure la oferta de aplicaciones y surjan nuevas necesidades e iniciativas hasta ahora no existentes. Algunos de estos posibles futuros usos de los dispositivos móviles pueden ser la realización de pagos, la vigilancia de la salud o la identificación de objetos del mundo real. Entrando en detalle, y acorde a los datos publicados por la consultora *Gartner*⁸, el top 10 de demanda de aplicaciones y servicios móviles en 2012 estará relacionado con:

1. Transferencias de dinero. Envíos de dinero persona a persona mediante el uso de SMS. Su uso ya tiene casos de éxito en países de desarrollo (ej. M-Pesa⁹).
2. Servicios basados en localización. Servicios contextuales cuya importancia radique en proporcionar al usuario determinada información de relevancia en base a su localización.
3. Búsquedas en el móvil. Servicios que proporcionen información deseada.
4. Navegación web. Navegadores, aplicaciones o herramientas mejoren la navegación en el móvil, pudiendo potenciar ventas B2C "*Business-to-Consumer*".
5. Vigilancia o monitorización de la salud. Aplicaciones de vigilancia de pacientes a distancia por medio de la utilización de sensores que transmitan información y constantes vitales.
6. Pagos mediante móvil. Aplicaciones que faciliten la realización de pagos.
7. Comunicaciones de corta distancia (NFC) entre dispositivos y terminales. Aplicaciones que den soporte a la autenticación y a la transferencia de datos a corta distancia.
8. Publicidad en el móvil. El mayor uso de dispositivos inteligentes y el acceso móvil a internet llevará a los anunciantes a realizar más esfuerzos en publicitar sus productos y servicios en estos dispositivos.
9. Mensajería instantánea. Aplicaciones que cubran la necesidad y predisposición del usuario de comunicarse otras personas.
10. Música en el móvil. Servicios que cubran la demanda de música descargada u online.

⁸ <http://www.gartner.com/it/page.jsp?id=1230413>

⁹ M-pesa: Servicio de transferencia de dinero de Kenia.

2.2 Uso del Contexto en Aplicaciones Móviles

El uso del contexto en las aplicaciones permite mejorar el enriquecimiento de la interacción hombre-máquina y crear servicios computacionales más útiles [12]. Por ello, y dada la capacidad de los dispositivos móviles de capturar determinada información de contexto del usuario, es necesario identificar y detectar los aportes concretos que el uso del contexto introduce en el diseño de interfaces de aplicaciones móviles.

El contenido de esta sección se estructura de la siguiente forma: el apartado 2.2.1 introduce una definición de contexto y una clasificación de las principales características de las aplicaciones sensibles al contexto y de los beneficios asociados a dichas características. A continuación, el apartado 2.2.2 resume los recursos incorporados en los dispositivos móviles que sirven para capturar información del contexto del usuario así como algunos atributos de contexto que pueden derivarse de los mismos. Igualmente, el apartado 2.2.3 describe los usos concretos del contexto dentro de las aplicaciones móviles actuales, los cuales han sido obtenidos a partir de un análisis de aplicaciones móviles actuales. Finalmente, el apartado 2.2.4 muestra ejemplos concretos de algunos de estos usos.

2.2.1 Contexto: Definición y Beneficios

El contexto ha sido definido en numerosas ocasiones por diversos autores, siendo la definición realizada por Dey & Abowd en [12] la que mejor aceptación ha tenido por parte de la comunidad, la cual define el contexto como:

“Cualquier información que puede ser usada para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto que se considera relevante para la interacción entre el usuario y una aplicación, incluyendo al usuario y a la aplicación propiamente dichos.”

Asimismo, aparte de definir este concepto, los autores identifican varias características que una aplicación puede introducir gracias al contexto, las cuales enriquecen y benefician a la aplicación. En concreto, estas características son:

- **La adaptación de la información y de los servicios.** Esta característica se refiere a la posibilidad de adaptar la información y los servicios disponibles en función del contexto. De la misma forma, se refiere también a la forma en la que la información se visualiza.

- **La ejecución automática de un servicio.** Esta característica se refiere a la ejecución o invocación automática de un servicio en el momento en el que una condición determinada ocurre en el contexto del usuario.
- **La etiquetación de información.** Esta característica se refiere a la etiquetación automática de elementos u objetos con información de contexto para su posterior recuperación y tratamiento.

2.2.2 Obtención de Información de Contexto en Aplicaciones Móviles

Establecida la definición y las características de una aplicación sensible al contexto, a continuación se sintetiza el conjunto de recursos incorporados en los dispositivos móviles que permite capturar información del contexto del usuario. En concreto, este conjunto de recursos son:

- **Acelerómetro:** sensor que indica al dispositivo cual es la posición del dispositivo, y que captura los movimientos y giros hechos con el dispositivo.
- **GPS:** sensor que proporciona la localización física del usuario.
- **Micrófono:** sensor que permite detectar el nivel de ruido del entorno.
- **Bluetooth:** tecnología que permite identificar y conectar dispositivos que hacen uso de esta tecnología permitiendo pequeñas transmisiones de datos entre ellos.
- **Cámara frontal:** sensor que permite capturar imágenes o grabar video de la persona mientras ésta está utilizando el dispositivo.
- **Reloj interno:** proporciona el instante actual permitiendo las mediciones de tiempo.
- **Servicios web:** permiten la recepción de información de contexto ajena al propio dispositivo, pero que también puede ser considerada como relevante. Un ejemplo de ello podría ser la recepción del estado meteorológico de la ciudad o la notificación de las fechas de cumpleaños de personas cercanas a través de consultas a una determinada red social.

Tal como muestra este conjunto de recursos, las posibilidades de enriquecer las aplicaciones móviles a partir del uso del contexto son amplias ya que es posible identificar varios atributos del contexto del usuario que pueden derivarse a partir de los mismos. Así por ejemplo, entre algunos de los atributos que podrían ser capturados encontramos: la localización del usuario, la velocidad, la distancia recorrida por el

usuario, la detección del estado de ánimo del usuario a partir de imágenes capturadas mediante la cámara frontal, la detección de personas alrededor del usuario gracias al uso de Bluetooth o del micrófono, la hora, la estación del año actual, etc.

Sin embargo, la utilización de información de contexto dentro de las aplicaciones móviles actuales todavía está en una fase temprana, y solamente algunos atributos como la localización del usuario o del tiempo se han consolidado dentro de las aplicaciones. Por una parte, las primeras funcionalidades que adquirieron los dispositivos móviles que eran distintas a las propias de los teléfonos fueron aquellas relacionadas con agendas, alarmas o calendarios. Así, éste puede ser uno de los motivos que indiquen el por qué el tiempo es uno de los atributos de contexto utilizados dentro de las aplicaciones. Por otra parte, la proliferación de servicios y aplicaciones basadas en localización, así como su perspectiva de crecimiento indican otro motivo por el que los desarrolladores dedican sus esfuerzos al desarrollo de este tipo de aplicaciones. Aparte, la complejidad del desarrollo de otro tipo de funcionalidades que capture información de contexto como podrían ser por ejemplo la detección del estado de ánimo a partir de las imágenes capturadas, o la detección de personas cercanas a través del Bluetooth o del micrófono pueden ser motivo por el que los desarrolladores prefieran optar por crear aplicaciones más sencillas con las que obtener beneficios.

2.2.3 Análisis del Uso de la Localización y el Tiempo en Aplicaciones

Móviles

Ante esta situación, y dado que para el desarrollo de este trabajo es necesario conocer el alcance real de la utilización del contexto dentro de las aplicaciones móviles actuales, se ha realizado un análisis de aplicaciones móviles reales que trata de determinar cuáles son los usos más frecuentes de los atributos del contexto del usuario que tienen mayor presencia en las aplicaciones actuales, es decir, la localización y el tiempo.

Procedimiento

En particular, el análisis realizado comprueba si cada una de las aplicaciones analizadas cumple o no un conjunto de características que se relacionan con las aplicaciones sensibles al contexto. En concreto, se parte de la base del conjunto de características identificadas por Dey & Abowd en [12]. Es decir, (1) la adaptación de la información y servicios mostrados, (2) la ejecución automática de servicios y (3) la etiquetación de

elementos con información contextual. Además, dada la posibilidad de que dicho conjunto de características no fuera completo, a medida que se analizaba una aplicación se analizaba si ésta incluía algún uso concreto de información de contexto que pudiera ser de especial interés. Así, en caso afirmativo, éste era anotado y añadido a la lista de características a comprobar dentro de cada aplicación. Así por ejemplo, se detectó la tendencia de algunas aplicaciones de compartir la información capturada con otras personas o la necesidad de otras aplicaciones de introducir determinada información como puede ser la introducción de una fecha o de una localización.

Características a analizar:

Entrando en detalle, el conjunto de características a comprobar es el siguiente:

- Captura del instante de tiempo actual.
- Captura de la localización actual.
- Representación de información teniendo en cuenta el instante actual.
- Representación de información teniendo en cuenta la localización actual.
- Activación automática en un instante de tiempo determinado.
- Activación automática en una localización determinada.
- Seguimiento del tiempo dentro de la aplicación.
- Seguimiento de la localización dentro de la aplicación.
- Etiquetación de un elemento con el tiempo actual.
- Etiquetación de un elemento con la localización actual.
- Publicación de la localización actual.
- Publicación del instante de tiempo actual.
- Necesidad de introducción manual de un instante de tiempo.
- Necesidad de introducción manual de una localización.

Resultados:

A continuación, en la Figura 1 se muestran los resultados obtenidos de dicho análisis, el cual muestra cuántas aplicaciones sobre un total de 40 aplicaciones analizadas, incluyen las características mencionadas. Como anotación, tras analizar 40 aplicaciones se decidió no aumentar la cantidad de aplicaciones analizadas al reiterarse los usos ya detectados y no aparecer ninguno nuevo.

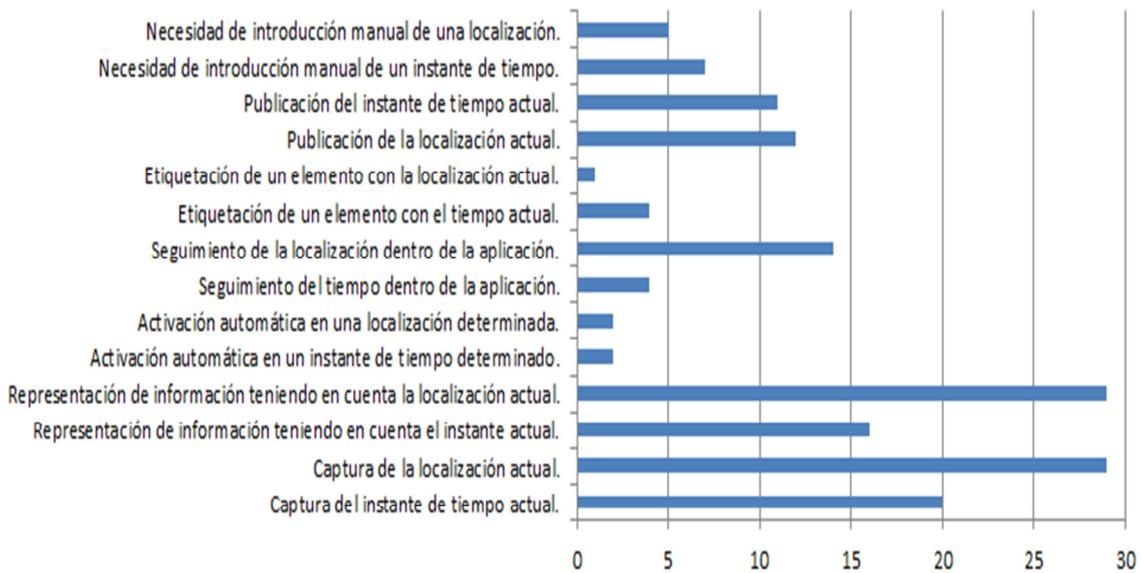


Figura 1. Resultados del análisis de usos de la localización y el tiempo en aplicaciones móviles

Como resultados a destacar, vemos que hay varios usos que sobresalen respecto al resto: captura de localización y tiempo, uso de estos factores para visualizar información, seguimiento de la localización o la publicación de dicha información. En el [Anexo I](#) de este documento puede verse el listado completo de aplicaciones sobre el que se ha realizado el análisis así como sus resultados individuales relacionados con las características que arriba se mencionan. En el siguiente apartado se muestran algunos ejemplos concretos de estos usos.

2.2.4 Ejemplos de Usos Típicos

A continuación se presentan algunos ejemplos de usos típicos encontrados en las aplicaciones analizadas.

- **Captura de la localización actual**

Finalidades: obtener puntos cercanos de interés o compartir la localización. Como ejemplos concretos de esta funcionalidad encontramos las aplicaciones “*Google Latitude*¹⁰”, o “*Around Me*¹¹”. *Google Latitude* da al usuario la posibilidad de capturar la posición actual en un momento determinado con el fin de compartir dicha información con sus contactos para que éstos vean su

¹⁰ <http://itunes.apple.com/es/app/google-latitude/id306586497?mt=8>

¹¹ <http://itunes.apple.com/es/app/aroundme/id290051590?mt=8>

localización sobre un mapa. Aparte, *Around Me* captura la localización actual para encontrar puntos de interés de diversos tipos (ej. cafés, cines, bancos, etc.).

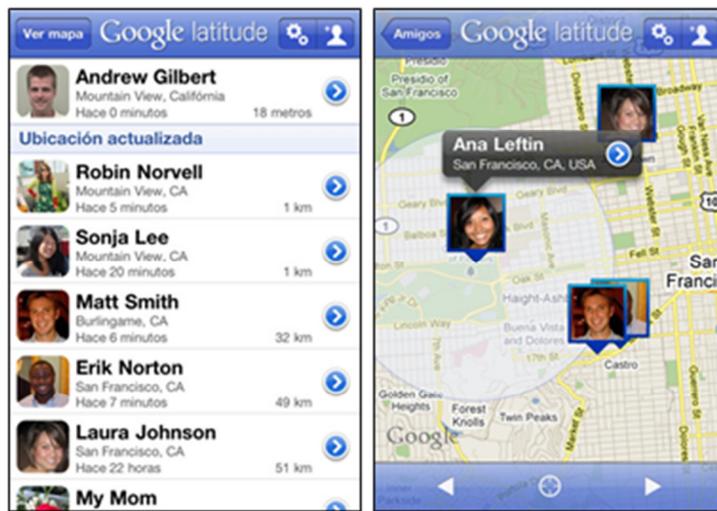


Figura 2. Vista de la aplicación móvil “Google Latitude”

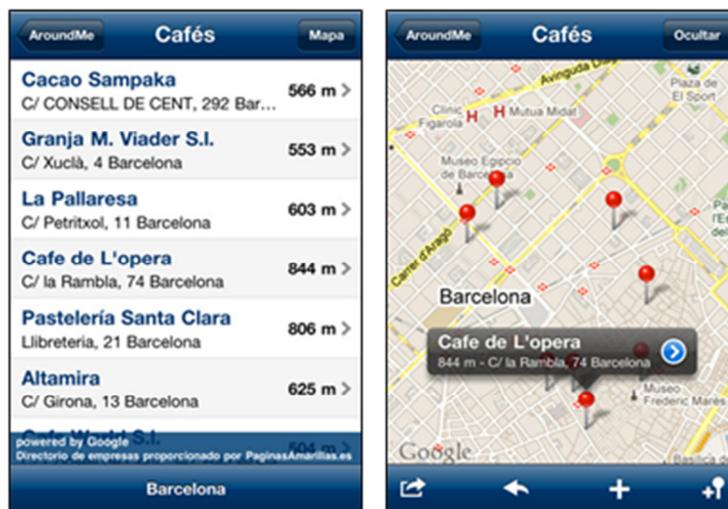


Figura 3. Vista de la aplicación móvil “Around Me”

- Seguimiento de la localización

Finalidades: controlar la ubicación de una persona (ej. transportistas, niños, etc.) o extraer información adicional sobre la ruta realizada (ej. distancia recorrida, velocidad, etc.).

Dos ejemplos concretos de aplicaciones donde se incluye esta funcionalidad típica son las aplicaciones “*Buscar mi iPhone*¹²” y “*Nike+ GPS*¹³”. *Buscar mi iPhone*

¹² <http://www.apple.com/es/iphone/features/find-my-iphone.html>

permite al usuario la posibilidad de realizar el seguimiento de la localización de su dispositivo en caso perdida. En cambio, *Nike+ GPS* permite la medición de tiempos y velocidades por tramos a partir de una ruta realizada.

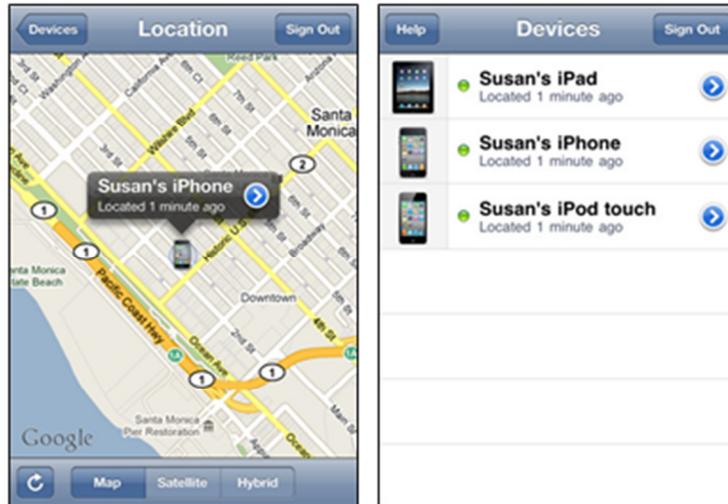


Figura 4. Vista de la aplicación móvil “Buscar mi iPhone”



Figura 5. Vista de la aplicación móvil “Nike+ GPS”

- Activación automática de la aplicación al llegar a una localización.

Finalidad: Disparo de un evento o acción a la llegada del usuario a una localización previamente establecida.

¹³ <http://itunes.apple.com/es/app/nike-gps/id387771637?mt=8>

Un ejemplo concreto de esta funcionalidad se incluye en la aplicación “*Neer*¹⁴”. Esta aplicación permite al usuario la definición de lugares típicos en los que suele ubicarse (ej. hogar, trabajo, guardería). De esta forma, a partir de estas localizaciones, la aplicación comparte de forma automática la ubicación del usuario cuando éste llega a las localizaciones establecidas.



Figura 6. Vista de la aplicación móvil “*Neer*”

- **Captura del instante actual**

Finalidades: clasificar acciones o eventos sobre la línea del tiempo, realizar una búsqueda de servicios disponibles en el día actual.

Dos ejemplos concretos de aplicaciones donde el tiempo se utiliza con estos fines son las aplicaciones “*Trenes*¹⁵” y “*Neer*¹⁶”, las cuales podemos ver respectivamente en las Figuras 7 y 6. Por una parte, *Trenes* utiliza el día actual para mostrar al usuario los horarios de trenes del día de hoy y facilitar del mismo modo los horarios de los dos días siguientes. Por otra, *Neer* utiliza el instante actual para ubicar dentro de la línea del tiempo las llegadas de contactos cercanos a las localizaciones que estos establecen previamente.

¹⁴ <http://itunes.apple.com/us/app/neer/id408503185?mt=8>

¹⁵ <http://itunes.apple.com/es/app/trenes/id312684710?mt=8>

¹⁶ <http://itunes.apple.com/us/app/neer/id408503185?mt=8>



Figura 7. Vista de la aplicación móvil “Trenes”

- **Activación automática de la aplicación en un instante de tiempo determinado**
Finalidad: invocación de un evento o acción al llegar un instante de tiempo previamente establecido.

Un ejemplo concreto de este uso dentro de una aplicación es el que realiza la aplicación “*Google Calendar Client*”¹⁷, la cual permite definir alarmas que se disparan a su llegada para lanzar un aviso recordatorio al usuario.

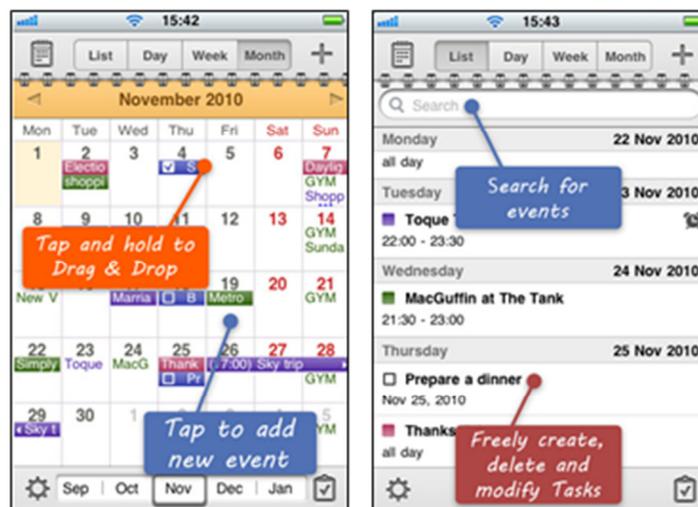


Figura 8. Vista de la aplicación móvil “Google Calendar Client”

¹⁷ <http://itunes.apple.com/us/app/calendars-google-calendar/id371434886?mt=8>

2.3 Conclusiones

En este capítulo se han introducido cuestiones básicas relacionadas con el trabajo que se presenta.

En primer lugar, se ha descrito cómo han aumentado las capacidades de los dispositivos móviles y cómo éstas pueden ser utilizadas para crear una gran diversidad de aplicaciones que cubren un gran número de funcionalidades. Así, la proliferación de aplicaciones móviles que se integran en la vida cotidiana justifica la necesidad de la propuesta que se presenta al mostrar la necesidad de que las aplicaciones se integren de una manera más desapercibida dentro de las actividades de las personas [63].

Por otro lado, en este capítulo se ha reflejado el potencial que tiene el uso del contexto para enriquecer las aplicaciones móviles y se han resumido los usos en las que algunos atributos del contexto del usuario son utilizados dentro de las aplicaciones actuales. Así, este conocimiento sienta la base para la definición del conjunto de patrones que se realiza posteriormente.

Finalmente, otras conclusiones detectadas en este capítulo son la necesidad de aprovechar mejor la información que puede obtenerse a través de sensores y servicios web para mejorar las aplicaciones; la necesidad existente de optimizar varios de los recursos de los dispositivos actuales, principalmente el del sensor GPS y la duración de la batería, ya que el uso continuado de dicho sensor dentro de una aplicación hace que la batería disminuya rápidamente; y la necesidad de tratar cuidadosamente cuestiones de privacidad y seguridad dada la tendencia existente en muchas aplicaciones de compartir información personal con otras personas.

Capítulo 3

Trabajo Relacionado

Este capítulo describe los trabajos relacionados con la propuesta que se presenta para el diseño de aplicaciones móviles sensibles al contexto. En particular, estos trabajos se clasifican a lo largo de las etapas que componen el proceso de creación de diseños de interfaz, el cual es conocido como proceso de diseño. Así, este capítulo proporciona una vista general de estas etapas y de las tareas que incluyen; una vista que permitirá identificar al final del capítulo aquellas características del proceso de diseño y aquellas carencias que justifican la propuesta que se presenta en este trabajo.

A continuación se describe la estructura del resto del capítulo. La sección 3.1 describe el objetivo del proceso de diseño y sus etapas de Análisis, Diseño y Evaluación. La sección 3.2 describe los trabajos relacionados con la fase de Análisis. La sección 3.3 presenta varios trabajos relacionados con la fase de Diseño. En particular, presenta varios conjuntos de guías y heurísticas para la creación de diseños de interfaz, y algunas propuestas que hacen uso de estas heurísticas para crear diseños de aplicaciones móviles. Después, la sección 3.4 describe los trabajos relacionados con la fase de Evaluación. Finalmente, la sección 3.5 concluye el capítulo indicando las necesidades detectadas que dan lugar a la propuesta que se presenta en este trabajo.

3.1 Proceso de Diseño

Un proceso de diseño es aquel que establece el conjunto de pasos que lleva al desarrollo de un producto o sistema¹⁸. Trasladando este concepto a la creación de interfaces de usuario, **un proceso de diseño debe establecer las etapas que trasladen los requisitos de la aplicación a los diseños de interfaz finales** que cubran dichos requisitos. Aparte, como apuntan diversos autores, estas etapas deben además tener en cuenta la necesidad de proporcionar soluciones usables que incrementen la satisfacción del usuario con la aplicación [2, 4, 66], y que incrementan como consecuencia el nivel de aceptación y éxito de la misma. Con este fin, la intervención del usuario final de la aplicación en cada etapa de desarrollo tiene un valor importante ya que es éste quien debe confirmar y validar finalmente que los diseños creados cumplen con sus necesidades y expectativas.

Un enfoque de proceso de diseño que introduce al usuario final de forma directa es el llamado “*Diseño Centrado en el Usuario*”. Este enfoque de diseño se basa en la introducción activa del usuario en cada etapa del proceso [39] de manera que desde las fases tempranas de desarrollo se centra en los usuarios y en sus tareas. Asimismo, promueve la creación de diseños de manera iterativa, es decir, el producto se diseña, y posteriormente se evalúa y modifica gracias al feedback recibido por usuarios reales.

Como describen diversos autores [2, 9, 20], un proceso de diseño UCD establece esencialmente tres etapas: *Análisis*, *Diseño* y *Evaluación*. En la fase de Análisis se introducen tareas y actividades para analizar los usuarios de la aplicación en base a sus requisitos, sus tareas, necesidades, limitaciones y expectativas frente a la aplicación. Posteriormente, en la fase de Diseño se crean bocetos y prototipos a partir de los artefactos creados en la fase de Análisis. Después, en la fase de Evaluación, los diseños creados son evaluados por usuarios reales que proporcionan el feedback necesario para modificar los diseños a lo largo de varias iteraciones. Finalmente, cuando estos diseños son validados, se entiende que cumplen con las expectativas del usuario y proporcionan una adecuada satisfacción de usuario.

A continuación se profundiza en cada una de las etapas del proceso de diseño, introduciendo dentro de cada una de ellas varios de sus trabajos relacionados.

¹⁸ http://www.nasa.gov/audience/foreducators/plantgrowth/reference/Eng_Design_5-12.html

3.2 Etapa de Análisis

El objetivo de la etapa de Análisis es entender las características del usuario y su entorno específico. De esta forma, este conocimiento permitirá desarrollar un sistema con los requerimientos apropiados, útil, con información de calidad y una interfaz de calidad [34].

Para alcanzar este objetivo, y tal como podemos ver en la Figura 9, esta etapa agrupa esencialmente dos tareas. La primera de ellas consiste en conocer el usuario de la aplicación, es decir, averiguar y descubrir sus necesidades, deseos, limitaciones y expectativas con la aplicación. Para ello, los diseñadores pueden hacer uso de diversos recursos entre los que se encuentran las entrevistas, los cuestionarios, la observación del usuario, la descripción de posibles escenarios. Aparte, la segunda de ellas consiste en reflejar toda la información capturada en artefactos que podrán ser consultados en las siguientes fases del proceso de diseño para tener presentes los deseos y requisitos del usuario en todo momento.



Figura 9. Tareas y técnicas de la etapa de análisis

3.2.1 Métodos para la Recolección de Información del Usuario

La recolección de información relacionada con el usuario puede realizarse a través de diferentes métodos que pueden utilizarse dentro de esta etapa del proceso de diseño tanto de manera individual como combinada. Por lo tanto, cada equipo de desarrollo deberá seleccionar los métodos que más se ajusten a sus capacidades y recursos para

conocer al usuario de la aplicación de la mejor forma posible. En particular, algunos de estos métodos son:

- **Entrevistas.** Las entrevistas son un buen método para recoger opiniones, pensamientos e ideas de los usuarios potenciales de una aplicación. Pueden realizarse de manera individual o en grupo. Proporcionan cierta flexibilidad al investigador que las realiza ya que en función de las respuestas recibidas del usuario puede profundizar más en determinados aspectos de la aplicación que le permitan explotar nuevas ideas y posibilidades.
- **Cuestionarios.** El pase de cuestionarios al usuario es uno de los métodos de recolección de información más utilizados. Permite recoger tanto datos cualitativos como cuantitativos en función de las preguntas que plantee. Su coste de su realización es relativamente bajo, sin embargo, tienen el inconveniente de que pueden proveer información incompleta si el encuestador no se encuentra presente ya que el usuario puede no estar comprometido con la realización del cuestionario o puede simplemente no comprender las preguntas realizadas.
- **Observaciones.** Las observaciones consisten en el análisis del comportamiento de los usuarios potenciales de la aplicación a través de la detección de la observación de sus costumbres, la detección de acciones frecuentes o de sus respuestas a determinadas situaciones.
- **Escenarios.** La descripción de escenarios es un método que consiste en la representación textual de un conjunto de acciones, eventos o situaciones que tienen lugar dentro de un determinado contexto como puede ser por ejemplo la utilización típica de una aplicación.

3.2.2 Representación de la Información Capturada en Artefactos

La representación en artefactos de la información capturada a través de los métodos comentados en el apartado anterior o de otros hace accesible la imagen obtenida del usuario en las siguientes etapas del proceso de diseño. Así, estos artefactos pueden ayudar al equipo de diseñadores y en general al equipo de desarrollo a reflexionar sobre la manera en la que piensa el usuario de la aplicación, a imaginar cómo éste se siente frente a la misma o deducir cómo se comportaría ante una determinada situación.

Una de las técnicas utilizadas para este fin es la introducida por Cooper en [11], la cual es llamada *Personas*. En particular, una *Persona* es un arquetipo o caracterización

ficticia de un usuario típico y representativo de un colectivo de usuarios [34], el cual sintetiza la imagen del usuario en una plantilla como la que podemos ver en la Figura 10. En dicho ejemplo podemos ver a su vez algunos atributos que dan forma al contenido de este artefacto entre los que podemos encontrar: nombre, fotografía, hábitos, información sobre un típico día de trabajo, conocimiento y actitud hacia las tecnologías, objetivos del usuario en la vida real, etc.

	Sarah Parker, advanced user		
	Personal Background: Sarah is a single woman who studies Economics at university. She shares a flat with two friends. Since she is a good student, she has little free time. She likes go shopping, going out, travelling, and practice tennis and running.		
Demographics: Age group: 21-34 Years online: 5 Student	Tech. Comfort: PC: high Internet: medium/high Smartphone: high	Needs: <ul style="list-style-type: none"> • Optimize time. • Organize her activities without forgetting tasks. • Non obtrusive services. 	
Motivations:		Scenarios:	
Be aware of pending tasks related to home.		<ul style="list-style-type: none"> • Introduces pending tasks. • Remembers shopping list wherever she is. • Views pending housework. • Remembers pending invoices (e.g. gas, phone, ...) 	
Be aware of pending homework related to her subjects.		<ul style="list-style-type: none"> • Introduces data about her new homework. • Remembers pending homework. • Views her timetable. 	
Be healthy and attractive.		<ul style="list-style-type: none"> • Establishes her training plan. • Tracking her activities. 	
Have fun.		<ul style="list-style-type: none"> • Be aware of special parties, concerts, social events around her. 	
Remain focused and alert.		<ul style="list-style-type: none"> • Remembers birthdays, exams dates, and deadline assignments. 	

Figura 10. Ejemplo del artefacto Persona

Los beneficios de la creación de Personas son varios [49]: centran la vista en los usuarios y en sus respectivos contextos, proporcionan el potencial necesario para captar nuevas necesidades y características, proveen una base compartida con la que mejorar la comunicación sobre los usuarios, y permiten centrarse en un público específico.

Diversos trabajos hacen uso de Personas cómo técnica para representar información sobre el usuario de una aplicación. Por ejemplo, en [13] se utiliza *Personas* para rediseñar un prototipo de aplicación con problemas de usabilidad y proveer soluciones más útiles que las iniciales; en [34] para mejorar la toma de decisiones en la fase de diseño de dispositivos para el cuidado de la salud de personas mayores.

En ambos trabajos los autores encuentran que pensar en el sistema con un usuario concreto en mente es más productivo que pensar en términos más generales o abstractos confirmando la hipótesis de Cooper que dice que es mejor diseñar con un único usuario en mente. Además, los autores encuentran útil esta técnica ya que agrupa en un único documento toda la información del usuario evitando que dicha información

esté dispersada a lo largo de diferentes artefactos; un documento que tal como se indica en [43, 50] debe proporcionar una visión del usuario fácil de recordar y de comprender para incrementar así su efectividad.

3.3 Etapa de Diseño

El objetivo de la etapa de Diseño es crear diseños de interfaces de usuario que cubran los requisitos establecidos en la fase anterior; diseños que además deben ser usables. Por este motivo, es importante que los diseñadores conozcan las limitaciones introducidas por los dispositivos móviles [20]. Así, con este conocimiento los diseñadores pueden crear o aplicar mecanismos que afronten limitaciones como pueden ser por ejemplo el uso de patrones de diseño, de guías, heurísticas o principios de diseño. De esta forma, los diseñadores pueden utilizar estos recursos en la creación de artefactos que les permitan reflexionar cómo cubrir las funcionalidades definidas previamente en la fase de Análisis; artefactos que pueden ser modificados a lo largo de varias iteraciones para alcanzar resultados usables y satisfactorios [64].

Los trabajos relacionados con esta etapa de Diseño se clasifican en los siguientes apartados. En primer lugar, los trabajos ubicados en el apartado 3.3.1 son aquellos que detectan factores que influyen en la usabilidad de una aplicación móvil. En segundo lugar, en el apartado 3.3.2 se muestran y discuten varios conjuntos de heurísticas y utilizados para crear diseños de interfaz.

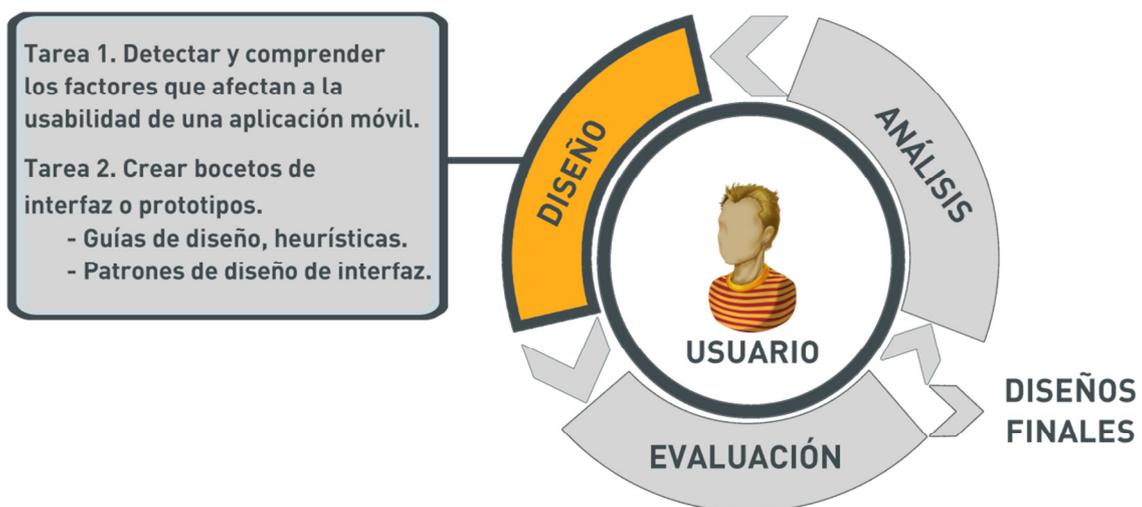


Figura 11. Tareas y recursos disponibles en la etapa de diseño

3.3.1 Factores a Tener en Cuenta en la Creación de Diseños

La interacción con una aplicación móvil se ve afectada por varios tipos de limitaciones que los diseñadores deben entender para poder crear soluciones que faciliten el cumplimiento de los objetivos de los usuarios. Para ello, si bien deben conocerse las propias restricciones hardware de estos dispositivos, también deben de tenerse en cuenta que la utilización de estos dispositivos en casi cualquier entorno y situación, limita la atención del usuario de la aplicación; un usuario que se encuentra inmerso en un entorno lleno de interrupciones y distracciones que dificultan la realización de sus tareas con el dispositivo.

Con el fin de detectar e instanciar cuáles son esas restricciones y limitaciones que afectan a la interacción con una aplicación, y por tanto a su usabilidad, a continuación se muestran dos clasificaciones diferentes que han detectado dichos factores. En particular, la primera de ellas muestra una tabla resumen que recoge algunos factores detectados y los trabajos en los que han sido detectados y comentados. La segunda de ellas es el modelo de contexto propuesto por Savio y Braiterman en [55], el cual indica gráficamente qué los factores anteriores y otros que influyen de una u otra manera en la interacción con un dispositivo móvil.

Factores Detectados en la Literatura

Los factores que dificultan y acomplejan la interacción del usuario con dispositivos y aplicaciones móviles han sido detectados y comentados en una multitud de trabajos anteriores. A continuación, en la Tabla 3 podemos ver una revisión de la literatura que muestra los trabajos que detectan y señalan diversos factores a tener en cuenta para el diseño de la interfaz de una aplicación móvil. En ella, se refleja cuáles son aquellos factores que preocupan esencialmente a la comunidad. En particular, en primer lugar se detectan aquellos factores relacionados con el hardware del dispositivo donde se encuentran: el tamaño de la pantalla, el cual afecta de manera directa al tamaño de los componentes gráficos que se incluyen en la interfaz y a la sobrecarga de información; el reducido tamaño de los teclados, que ya sean estos táctiles o no, limitan la introducción de datos por parte del usuario. Además, se encuentran factores que influyen en la eficiencia de los dispositivos como el poder computacional, la duración de la batería o la falta de conectividad son factores. Por último, también se incluye la importancia del contexto móvil en el cual se utilizan estos dispositivos que introduce problemas como el

hecho de usar el dispositivo con una única mano, el hecho de que el usuario se encuentra realizando múltiples actividades en paralelo, las distracciones o interrupciones que pueden tener en un entorno real, o los cambios producidos en dicho entorno.

Factor	Trabajos previos
Entrada de datos	[55], [51]; [25]; [52]; [66]; [20]; [15]
Duración de la batería	[55]; [51]; [52]
Tamaño de la pantalla	[55]; [66]; [51]; (Pascoe, 2000); [20]; [52]; [15]
Tamaño de los componentes gráficos.	[37]; [29]
Cambios en el contexto	[66]; [51]; [66]
Velocidad de conexión	[55]; [66]; [51]
Poder de computación	[66]; [51]
Otros	Realización de múltiples tareas simultáneas, uso del dispositivo con una única mano [55], [66]; complejidad o sobrecarga de información [55], [51], [27]; postura del usuario [37, 29]

Tabla 3. Trabajos que indican factores que afectan a la usabilidad de una aplicación móvil

Modelo de Contexto de Interacción Móvil

El modelo de contexto presentado por Savio y Braiterman en [55], el cual podemos ver en la Figura 12, muestra una multitud de factores que influyen en la interacción con un dispositivo móvil. Estos factores se presentan clasificados en diferentes niveles, los cuales se subsumen unos a otros, y los cuales hacen patente la necesidad de tener en cuenta factores que no se tenían en cuenta en el desarrollo de interfaces de usuario de aplicaciones escritorio. En particular, tiene en cuenta niveles generales que se refieren a factores culturales y niveles particulares que se refieren a limitaciones de los dispositivos. En concreto, los 5 niveles presentados son los siguientes 5 niveles:

1. Cultura: es el nivel más general. A este nivel se asocian factores de tipo general como pueden ser factores económicos, sociales, legales.
2. Entorno: es el nivel que agrupa los factores del entorno del usuario que afectan a la interacción. En este nivel podemos encontrar factores como la luminosidad, el

ruido, el hecho de encontrarse en un espacio público o privado, con personas alrededor o en un contexto donde las interrupciones son frecuentes.

3. **Actividad:** es el nivel que introduce la importancia de la actividad que está llevando a cabo el usuario, ya que, situaciones tan dispares como el hecho de estar conduciendo, trabajando, caminando por la calle, en una cola, o estar sentado en casa influyen sobre la postura del usuario o incluso las modalidades de interacción que éste puede utilizar.
4. **Usuario:** es el nivel que indica la necesidad de tener en cuenta las necesidades y objetivos generales del usuario, las tareas que este suele desarrollar, o los distintos niveles de atención que este puede dedicar en función del resto de condiciones del contexto.
5. **Dispositivo:** es el nivel que clasifica las propiedades del dispositivo con el cual el usuario realiza la interacción. Agrupa factores que se relacionan con las capacidades hardware del dispositivo (ej. nivel de batería, software instalado, nivel de conocimiento del dispositivo, etc.), o factores que se relacionan con la conectividad del dispositivo a la red (ej. velocidad, disponibilidad, etc.).



Figura 12. Modelo de contexto propuesto por Savio y Braiterman

3.3.2 Heurísticas para el Diseño de Interfaces de Aplicaciones Móviles

Para hacer frente a los factores que se describen en el punto anterior, los diseñadores suelen hacer uso de conjuntos de heurísticas o recomendaciones. Así, como apunta Goodman en [22], los diseñadores toman sus decisiones de diseño en base a guías de diseño que se usan de manera informal y de formas diferentes en cada proyecto.

Varios de estos conjuntos de heurísticas o recomendaciones provienen del diseño de interfaces de usuario de aplicaciones de escritorio. Por lo tanto, hacen referencia a algunos conceptos y atributos comunes de usabilidad que también son importantes en el diseño de interfaces de aplicaciones móviles, y por tanto, pueden ser aplicadas también a este tipo de aplicaciones. Sin embargo, debido a las diferencias entre ordenadores de sobremesa y estos dispositivos, estas guías no son completamente útiles para proveer diseños de interfaz usables, ya que su aplicación sin ningún tipo de adaptación o

matización puede llevar a la creación de soluciones que no tienen en cuenta el contexto móvil en el que se encuentra el usuario, o pueden llevar a la creación de diseños “*miniaturizados*” con interfaces de usuario sobrecargadas de componentes y de información [6].

Dentro de los conjuntos representativos de guías o heurísticas encontramos por una parte las guías para el diseño de interfaces de usuario definidas por Nielsen en [41] o las reglas propuestas por Schneiderman en [57]. Asimismo, en esta sección se incluyen también las guías proporcionadas por la compañía Apple para el diseño de aplicaciones móviles, por ser ésta una compañía que ha dado lugar a productos y dispositivos que han sido diseñados teniendo en cuenta criterios de usabilidad. A continuación se describen y discuten dichos conjuntos de guías o heurísticas.

Heurísticas de Nielsen

1. **Visibilidad del estado del sistema:** el sistema debe mantener al usuario informado sobre aquello que está ocurriendo, a través del feedback apropiado y en un tiempo razonable.
2. **Utilizar el lenguaje del usuario:** el sistema debe utilizar términos y conceptos familiares al usuario en lugar de términos orientados al sistema. Seguir convenciones del mundo real, presentando la información en un orden natural y lógico.
3. **Proporcionar al usuario libertad y control:** los usuarios frecuentemente escogen opciones erróneas de las que desean salir de una forma clara y sencilla. Dar soporte a operaciones de deshacer y rehacer.
4. **Consistencia y estándares:** los usuarios no deben preguntarse si las palabras, situaciones o acciones significan lo mismo. Seguir los estándares.
5. **Prevención de errores:** prevenir errores. Solicitar una confirmación al usuario antes de la realización de operaciones que pueden crear un resultado no deseado.
6. **Reducir la carga de memoria del usuario:** minimizar la carga de memoria del usuario a través de objetos, acciones, y opciones fácilmente visibles. Las instrucciones de uso del sistema deben ser visibles o fácilmente recuperables siempre que sea apropiado.

7. **Flexibilidad y eficiencia de uso:** el empleo de atajos puede maximizar la eficiencia de los usuarios expertos. Facilitar la realización de acciones frecuentes.
8. **Estética y diseño minimalista:** los diálogos no deben contener información irrelevante o raramente necesitados. Cada información adicional introducida disminuye la visibilidad de aquella relevante.
9. **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de los errores:** los mensajes de error deben expresarse en un lenguaje claro, expresando correctamente cómo solucionar el problema.
10. **Ayuda y documentación:** es recomendable que un sistema pueda ser usado sin documentación. Sin embargo, puede ser necesario disponer de ayuda y/o documentación. Ésta, debe ser fácil de usar, centrada en las tareas de los usuarios, listar los pasos concretos a llevar a cabo y no ser demasiado extensa.

Discusión

Este conjunto de heurísticas propuesto por Nielsen en [41] es uno de los más utilizados para para mejorar la usabilidad de productos de diverso tipos y de diverso dominio. Así por ejemplo, ha sido utilizado para el diseño de hardware de productos musicales [18], para la evaluación de una aplicación móvil [65], para la evaluación de una aplicación web e-learning [53] o para la evaluación de sitios web 2.0 [59]. Aparte, varios autores como [58] o [35] consideran este conjunto de heurísticas como la base general a utilizar para la obtención de soluciones usables; una base general que debe ser adaptada para crear un conjunto de heurísticas específico del dominio de aplicación sobre el que se pretenden aplicar tal como realiza Sim en [58] para el dominio de aplicaciones de evaluación de estudiantes, o tal como realizan Alsumait y Al-Osaimi en [1] para la evaluación de aplicaciones e-learning.

Las 8 Reglas de Oro de Shneiderman

1. **Ofrecer consistencia:** deben utilizarse las mismas secuencias de acciones ante situaciones similares; debe utilizarse idéntica terminología en menús, mensajes, ayudas, comandos.

2. **Proporcionar accesos directos para los usuarios frecuentes:** a medida que la frecuencia de uso se incrementa, los usuarios desean disminuir el tiempo dedicado a las interacciones. Así, ayudar como abreviaturas, atajos, comandos ocultos que lancen operaciones frecuentes facilita la interacción a los usuarios más avanzados.
3. **Ofrecer información de retroalimentación al usuario:** para cada operación, el sistema debe proveer información de retroalimentación. Mientras que para operaciones frecuentes, la retroalimentación puede ser modesta, para operaciones importantes o de menor frecuencia, la información proporcionada por el sistema debe ser más sustancial.
4. **Diseñar diálogos que guíen al usuario hacia la finalización de tareas:** las secuencias de acciones deben estar bien estructuradas con un principio, parte intermedia y final bien definidos. Los usuarios deben sentirse satisfechos al completar una tarea.
5. **Prevención y manejo de errores:** en la medida de lo posible, diseñar el sistema para evitar que el usuario no cometa errores. En caso de que un error tenga lugar, el sistema debe ser capaz de detectar dicho error y ofrecer al usuario mecanismos sencillos con los que manejar estos errores.
6. **Permitir una recuperación sencilla a acciones realizadas:** incluir operaciones de rehacer y deshacer para aliviar la ansiedad del usuario. Dichas opciones pueden agrupar o no un conjunto de operaciones.
7. **Dejar que el usuario sea quien controle la aplicación:** diseñar la aplicación para que sea el usuario quien controle el flujo de ejecución de la aplicación en lugar de ésta.
8. **Reducir la carga de memoria:** la limitación humana de procesamiento de información requiere que las visualizaciones se mantengan simples, se consoliden pantallas de varias páginas, la frecuencia de ventana de movimiento se reduzca y adjudique suficiente tiempo de aprendizaje para códigos, secuencias de acciones, mnemotécnicos

Discusión

Al igual que el conjunto de heurísticas anterior, el conjunto de reglas propuesto por Shneiderman en [57], también está enfocado a entornos de escritorio. Por lo tanto, **su aplicación directa sobre una aplicación móvil no es suficiente para alcanzar soluciones adecuadas.** Para cubrir esta carencia, Gong y Tarasewich mapean y adaptan este conjunto de reglas para que pueda ser utilizado en el desarrollo de

interfaces de usuario de aplicaciones móviles [21]. En particular, la adaptación realizada conserva un conjunto determinado de reglas. Asimismo, adapta otras reglas específicas e introduce otras nuevas.

Reglas aplicables directamente:

1. **Permitir accesos rápidos a usuarios experimentados.**
2. **Ofrecer información de retroalimentación.**
3. **Diseñar diálogos que guíen al usuario hacia la finalización de tareas.**
4. **Dejar que el usuario sea quien controle la aplicación.**

Reglas modificadas a las que se incluyen las siguientes directrices:

1. **Ofrecer consistencia:** ofrecer una apariencia y unos mecanismos de interacción similares en las distintas plataformas y dispositivos.
2. **Permitir una recuperación sencilla a acciones tomadas:** evitar en la medida de lo posible la dependencia de la conexión a la red ya que ésta puede impedir la recuperación de una tarea.
3. **Prevención y manejo de errores:** ninguna operación con posibles efectos perjudiciales debe ser accesible desde una manera simple ya que hay que tener en cuenta por ejemplo que varios componentes gráficos pueden encontrarse a distancias cercanas y el usuario puede equivocarse dado el reducido tamaño del dispositivo.
4. **Reducir la carga de memoria:** evitar que el usuario memorice opciones tratando de que este reconozca fácilmente las operaciones disponibles. Evitar que la aplicación móvil sea el centro de atención del usuario, ya que este se encontrara haciendo una actividad principal, la cual desea continuar en lugar de desviar toda su atención hacia el dispositivo.

Nuevas reglas introducidas:

1. **Diseñar para contextos múltiples y cambiantes:** tener en cuenta los cambios de actividad y de entorno del usuario y utilizar la información contextual del usuario para adaptar la aplicación a estos cambios de contexto (ej. proporcionando una interacción a realizar con una, dos o ninguna mano, modificando el tamaño de las fuentes en función de la actividad del usuario, etc.).

2. **Diseñar para dispositivos pequeños:** proporcionar componentes de selección en lugar de componentes de introducción manual de información. Hacer uso de la voz o el audio como modalidad de interacción.
3. **Diseñar teniendo en cuenta la limitada atención del usuario:** utilizar los sonidos o avisos por vibración como confirmación o salida de información con el objetivo de evitar que el usuario desvíe la atención de la actividad que está realizando.
4. **Diseñar fomentando un uso veloz y la recuperación de la última tarea realizada:** permitir que las aplicaciones puedan arrancarse, pararse y recuperarse con poco esfuerzo ya que los usuarios necesitan acceder rápidamente a la funcionalidad para no entorpecer la principal actividad que está realizando.
5. **Diseñar colocando las principales funcionalidades en los contextos de interacción principales:** con el objetivo de evitar sobrecarga de información en pequeñas pantallas, ser establecer diversos niveles jerárquicos estableciendo en los niveles principales aquellas funcionalidades más frecuentes y ofreciendo opciones de navegabilidad a contextos de información más detallada o precisa.
6. **Permitir la personalización:** permitir la configuración del usuario en base a sus gustos o habilidades.
7. **Cuidar la estética del diseño:** la estética de las aplicaciones deben ser agradable y llamativa a la vez de usables.

Principios y Guías de Apple para la Creación de Interfaces

- **Integridad estética.** Este principio se basa en corresponder la apariencia de una aplicación móvil a su misión, a la funcionalidad a la que da soporte. Así por ejemplo, es recomendable que un juego proporciona una apariencia atractiva o que una aplicación para mejorar la productividad proporcione componentes estándar para poder ser usada rápidamente.
- **Consistencia.** Este principio se basa en facilitar el uso de la aplicación a través del conocimiento adquirido en el uso de otras aplicaciones, para trasladar así experiencias similares a nuestra aplicación.
- **Manipulación directa.** Este principio se basa en la reducción en el número de interacciones. En detalle, consiste en la manipulación directa de los objetos mostrados en la interfaz, en lugar de por ejemplo tener que seleccionarlos y pulsar a

continuación un botón para realizar una determinada funcionalidad (ej. rotar una imagen).

- **Retroalimentación.** Este principio se basa en informar al usuario de qué es lo que está ocurriendo para que éste no se sienta desorientado en la actividad que están desarrollando.
- **Metáforas.** Este principio se basa en trasladar metáforas del mundo real a objetos u operaciones de la aplicación para facilitar la utilización de la aplicación.
- **Control del usuario.** Este principio se basa en preservar al usuario como el dueño del control de la aplicación sin que ésta tome demasiadas decisiones sin la supervisión del usuario.

Asimismo, a parte de estos principios establece varias guías concretas de usabilidad:

- Dar importancia a la tarea más importante a cubrir en cada pantalla.
- Elevar el valor del contenido mostrado removiendo, reduciendo u ocultando componentes en determinados momentos.
- Destacar la funcionalidad más usada.
- Proporcionar una navegación adecuada.
- Resaltar de forma clara la principal función de la aplicación.
- Utilizar vocabulario cercano al usuario.
- Minimiza la cantidad de información de entrada requerida.
- Utilizar consistentemente los componentes gráficos.
- Hacer uso de los gestos de una manera apropiada.
- Minimiza los tiempos de espera tanto como sea posible.

Discusión

Los principios y guías de diseño provistos por la compañía Apple para el desarrollo de aplicaciones móviles¹⁹ tratan de ayudar en la toma de decisiones de diseño persiguiendo comportamientos más intuitivos e interfaces atractivas. Pese a no constar ningún trabajo de investigación que haga uso de las mismas, se han incluido

¹⁹http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Principles/Principles.html#//apple_ref/doc/uid/TP40006556-CH5-SW2

en esta sección debido al éxito que han alcanzado muchos de sus dispositivos y productos gracias a prestar especial atención a la usabilidad.

3.4 Etapa de Evaluación

El objetivo de la etapa de Evaluación es detectar problemas de usabilidad en los diseños realizados en la fase anterior con el fin de que estos sean corregidos para mejorar la usabilidad de una aplicación [3]. Para ello, un proceso de evaluación de una aplicación se compone en líneas generales de dos piezas. La primera de ellas es el usuario que va a probar la aplicación, quien debe de realizar un conjunto de tareas concretas sobre la misma. La segunda de ellas es el conjunto de métodos que sirve para documentar y recoger información sobre la realización de las tareas definidas con el fin de utilizar esta información para modificar los problemas de usabilidad detectados.

Dentro de los métodos disponibles para la recogida de información, y cómo podemos ver gráficamente en la Figura 13 encontramos: el pase de cuestionarios a usuarios que han utilizado la aplicación, la evaluación realizada por expertos, la recolección automática de información desde la propia aplicación a través de un log, grabaciones de video, observaciones directas, etc. Así, como vemos, los diseñadores tienen a su alcance métodos de diverso tipo. Sin embargo, dadas las limitaciones existentes en las aplicaciones, hay que tener en cuenta que algunos de estos métodos de evaluación tradicionales deben ser adaptados para mejorar su eficacia [19].

Aparte, además de decidir qué métodos utilizar para la evaluación de una aplicación, los diseñadores deben decidir la localización en la que se realizará la evaluación, es decir, si se realizará dentro del laboratorio o en el entorno real de la aplicación.



Figura 13. Tareas y técnicas de la fase de evaluación

El contenido de esta subsección se organiza de la siguiente manera: el apartado 3.4.1 describe los métodos de recogida de información disponibles; el apartado 3.4.2 describe las diferencias de la realización de una evaluación dentro del laboratorio y dentro de un entorno real.

3.4.1 Métodos de Recogida de Información

Observación directa.

La observación directa consiste en la observación del usuario mientras éste interactúa con la aplicación y realiza un conjunto de tareas. Éste método permite recoger feedback general del sistema. Sin embargo, aporta menos que otros métodos como las grabaciones o el “*data logging*” ya que el hecho de basarse únicamente en el instante sin capacidad de reproducir lo ocurrido puede dejar pasar por alto determinada información. Además, este método tiene otro inconveniente importante como el hecho de no poder ver correctamente en todo momento lo que ocurre en la aplicación ya sea por el tamaño de la pantalla y el ángulo y distancia de visión del observador [56].

Cuestionarios

Al igual que en la fase de Análisis del proceso de diseño, los cuestionarios pueden ser utilizados en esta fase de Evaluación. En concreto, su utilización dentro de esta etapa tiene lugar tras la prueba de la aplicación. A través de ellos, los diseñadores plantean preguntas al usuario tratando de detectar problemas de usabilidad, opinión general, comentarios, estimación de la facilidad de uso o de la carga cognitiva, etc.

Grabaciones de video

La grabación en video del uso de la aplicación es uno de los métodos frecuentemente más utilizados para la recogida de información. Algunos autores como [14] y [26] lo introducen en sus trabajos. Sin embargo, la grabación del uso de una aplicación móvil se ve limitada por una parte por el tamaño de la pantalla y por otra por el carácter móvil del uso del dispositivo. Frente a este problema, la grabación mediante la instalación de software que permita la grabación de las acciones del usuario sobre la pantalla podría cubrir esta limitación, sin embargo, la potencia de estos dispositivos todavía no es suficiente para este tipo de software.

Como ejemplo de adaptación del método a estas limitaciones, algunos autores como [56] han propuesto sistemas de grabación alternativos que aportan algunas mejoras. En concreto, tal como podemos ver en la Figura 14, proponen por una parte la introducción de una cámara fija colocada sobre el escritorio para las evaluaciones a realizar en el laboratorio y por otra la incrustación de una cámara móvil sobre el propio dispositivo para evaluaciones sobre entornos más reales. Sin embargo, pese a las mejoras introducidas, estas propuestas todavía tienen problemas como por ejemplo el hecho de tener que requerir que el dispositivo esté en posición estática sobre la mesa o el incremento del peso del propio dispositivo

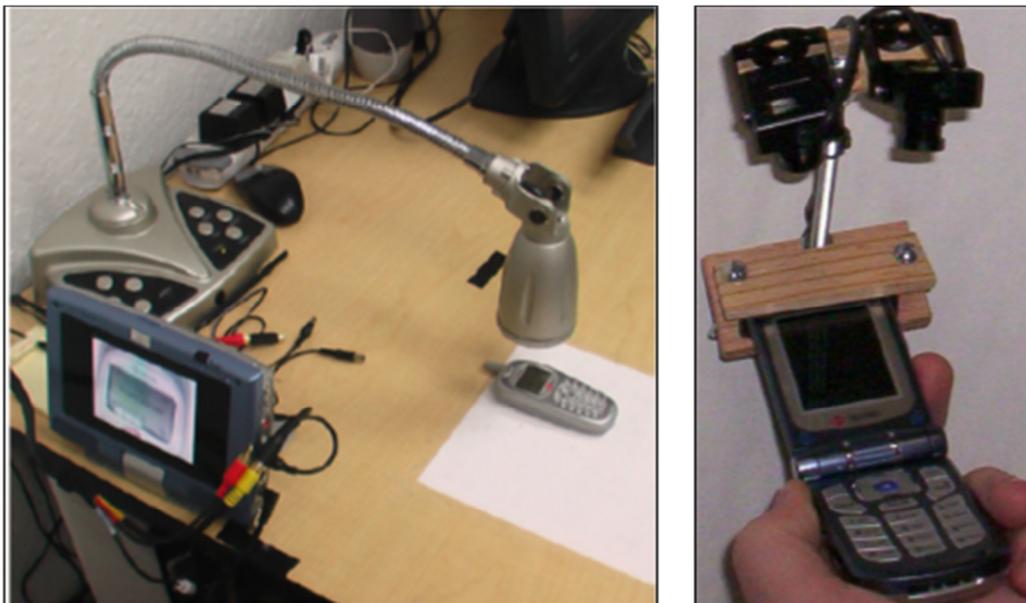


Figura 14. Sistemas de grabación propuestos por Schusterich y otros

Recolección Automática del Uso de la Aplicación (Data Logging)

La recogida automática de información desde dentro de la aplicación consiste en la monitorización del uso de la aplicación a través de modificaciones en el código fuente que permiten almacenar cada acción realizada por el usuario. Este método, según la metodología para la recolección de información propuesta en [3] contiene las siguientes 4 tareas o fases:

- **Fase de preparación:** fase consistente en la introducción de las instrucciones necesarias para la recolección de información dentro del código fuente de la aplicación.
- **Fase de recolección:** fase consistente en la propia recolección de información a través del uso de la aplicación.
- **Fase de extracción:** fase que consiste en la extracción de la información recolectada en la fase anterior.
- **Fase de análisis:** fase que tiene el objetivo de detectar partes de la aplicación con problemas de usabilidad, patrones de uso, comportamiento del usuario, etc.

La utilización de este método de recogida de información se refleja en varios trabajos: [47] lo utiliza para evaluar una aplicación móvil, [36] lo utiliza para detectar patrones de uso en una aplicación móvil, y [24] lo utilizan dentro de una aplicación basada en mapas.

Sin embargo, la utilización de este método presenta dificultades. Por una parte la preparación de la aplicación es tediosa de la misma forma que lo son las fases de extracción y análisis de los datos extraídos en aquellos casos en los que el conjunto de datos almacenado es elevado. Así, la medición de la usabilidad a partir de la recolección automática de datos no es una labor sencilla.

Evaluación realizada por expertos

Este método se centra en el hecho de que la evaluación de la aplicación la realice un experto en temas de usabilidad y no el usuario final de la aplicación. Este enfoque diferente, toma la base de que un experto puede detectar diversos problemas de usabilidad a un menor coste.

Un ejemplo de trabajo en el que tiene lugar una evaluación realizada por expertos es [19]. En este trabajo, los autores introducen varios métodos de evaluación que se aplican de manera sucesiva en una aplicación móvil: (1º) una evaluación realizada por

expertos, (2º) una evaluación junto a usuarios representativos dentro del laboratorio, y (3º) una evaluación junto a usuarios representativos de la aplicación en el entorno real.

Los resultados obtenidos indican que las diferentes evaluaciones realizadas se complementan tal como se comenta en la subsección 3.4.1. La evaluación realizada por expertos puede localizar a un bajo coste varios problemas de usabilidad relativos con la interfaz de usuario. Aparte, la evaluación de la aplicación hecha por usuarios en el laboratorio puede detectar errores que se relacionan con las actividades a realizar. Y finalmente, la evaluación realizada en el entorno real puede proporcionar datos reales sobre el uso de la aplicación de los usuarios y los efectos del entorno en la actividad a desarrollar.

Asimismo, indican que la evaluación realizada por expertos detectó problemas más críticos y serios que las demás evaluaciones. Por otra parte, la gran mayoría de los errores encontrados en la evaluación realizada en el entorno real correspondían con aspectos estéticos.

Además, respecto al coste requerido para realizar dichas evaluaciones, las mediciones de tiempo indican que la evaluación realizada por expertos requirió 32 minutos, la evaluación con usuarios en el laboratorio requirió 110 minutos, y la evaluación con usuarios en el entorno real requirió 330 minutos.

3.4.2 Evaluación en el Laboratorio y en el Entorno Real

Uno de los debates existentes en la comunidad de investigadores consiste en la decisión de elaborar la evaluación dentro del laboratorio o en el entorno real en el que la aplicación será usada. Varios autores [30, 26, 40], han realizado comparativas entre evaluaciones realizadas en el laboratorio y evaluaciones realizadas en un entorno real.

En primer lugar, las diferencias existentes entre las aplicaciones tradicionales y las aplicaciones móviles indican que es necesario que este tipo de aplicaciones se evalúen en entornos reales ya que en el laboratorio no pueden simularse las condiciones de un entorno real [28, 48, 14]. Sin embargo, el testeado de la usabilidad de una aplicación en un laboratorio es suficiente para estudiar cuestiones relacionadas con interfaces de usuario, mecanismos de introducción o selección de datos, y cuestiones de navegación [26], es decir, factores sobre los cuales no tiene influencia el contexto o entorno de uso de la aplicación.

En segundo lugar, respecto a los costes, la realización de una evaluación en un entorno real es más costosa en tiempo y en recursos necesarios [23] y más difícil de controlar que una evaluación realizada en un laboratorio ya que no pueden predecirse todos los factores que influirán sobre la usabilidad en el entorno real. Así, los estudios de usabilidad realizados en un laboratorio pueden detectar varios problemas de usabilidad a un menor coste.

3.5 Conclusiones

Los trabajos presentados y clasificados a lo largo del proceso de diseño tratan de mejorar la usabilidad de las aplicaciones ya que como apuntan una gran variedad de autores la usabilidad juega un rol importante en la aceptación de una aplicación [4, 14, 2, 66, 32]. Asimismo, queda justificada la realización de contribuciones a lo largo de las tres etapas del proceso de diseño ya que para conseguir una aplicación o un producto usable no deben introducirse tareas específicas únicamente al final del proceso de desarrollo, ya que cada decisión tomada desde las fases iniciales afecta a la usabilidad final [41].

Profundizando en el ámbito de las aplicaciones móviles, la usabilidad se convierte en una necesidad aún más importante ya que la interacción del usuario con el dispositivo se ve afectada por las limitaciones del contexto móvil en el que estos dispositivos se utilizan así como las propias limitaciones de interacción que éstos introducen.

Ante este escenario, **la simplicidad se convierte en un atributo de especial relevancia** en la creación de diseños de interfaz de aplicaciones móviles. Algunos trabajos así lo concluyen indicando la necesidad de mostrar información bajo un criterio minimal, introduciendo en las interfaces únicamente aquella información realmente importante [46, 51].

Por otra parte, otro de los problemas existentes es que los diseñadores corren el riesgo de diseñar productos para ellos mismos [34]. Por este motivo, dentro de la etapa de Evaluación **cobra importancia el hecho de que la persona que evalúe diseños de interfaz sea otra distinta al creador de la misma.**

Asimismo, como se indica en [19], la combinación de distintos métodos de evaluación tanto en el laboratorio como en el entorno real de aplicación es útil ya que ambas se complementan permitiendo detectar diversos tipos de problemas de usabilidad. De esta forma, la evaluación realizada en el laboratorio puede detectar problemas de

interfaz, de navegación o de usabilidad de componentes gráficos particulares, y para complementar el conjunto de problemas de usabilidad detectados, la evaluación en el entorno real puede detectar aquellos problemas relacionados con el contexto cambiante en el que se usa la aplicación.

Capítulo 4

Simplificación de Diseños de Interfaz

Como hemos visto en el capítulo anterior, varios problemas dificultan la creación de diseños de interfaz de usuario de aplicaciones móviles (ej. contexto móvil, limitaciones de los dispositivos, problemas de los diseñadores, etc.).

Dentro de estos problemas a tratar, la solución que se propone en este capítulo trata de afrontar esencialmente dos de ellos. Por una parte, la discusión anterior indica que **la Simplicidad es un atributo de especial relevancia en la creación de diseños de interfaz**. Varias guías, recomendaciones y heurísticas [60, 31, 62, 10, 55, 41, 2, 57] hacen referencia a este atributo, pero en ninguno de estos enfoques se introduce la simplicidad de manera ortogonal en la creación de diseños. Sin embargo, el hecho de considerar la simplicidad como aspecto independiente del sistema elimina algunas de las limitaciones que reducen el espacio del diseño [61]. Por otra parte, en la discusión anterior se comenta que los diseñadores corren el peligro de diseñar interfaces para ellos mismos, y por lo tanto, **es recomendable que un diseño se evalúe por una persona distinta a quien lo ha realizado**.

En este capítulo se describe una propuesta que trata estos dos problemas. En particular: la sección 4.1 proporciona una vista general de la solución, la sección 4.2 describe el enfoque de diseño basado en Simplicidad propuesto, el cual tiene el objetivo de refactorizar diseños de interfaz de aplicaciones móviles para reducir las complejidades encontradas. Asimismo, y para proveer soluciones concretas al dominio de las aplicaciones móviles sensibles al contexto, la sección 4.3 provee un conjunto de patrones de diseño de este tipo de aplicaciones, el cual sirve de base de conocimiento para

detectar si un diseño de interfaz de este tipo de aplicaciones necesita ser modificado. Finalmente, la sección 4.4 concluye el capítulo.

4.1 Vista Global de la Solución Propuesta

La solución propuesta para tratar los desafíos anteriores se compone de tres contribuciones, las cuales podemos ver gráficamente en la Figura 15.

La primera y más importante es la definición de un conjunto de guías de simplicidad que sirve para tratar problemas de complejidad en diseños de interfaz de usuario de aplicaciones móviles. La definición de este conjunto se inspira en las reglas de simplicidad provistas por John Maeda en [38]. Dicho conjunto de guías es una adaptación de estas reglas al ámbito del diseño de interfaces de aplicaciones móviles.

La segunda contribución es la introducción de un rol *Simplificador* ajeno al equipo de diseñadores. La misión de este rol consiste en (1) detectar complejidades en los diseños provistos por los diseñadores y en (2) aplicar el conjunto de guías anterior para proporcionar información a los diseñadores que permita la modificación de estas interfaces.

Dado que el uso de un conjunto de guías de diseño no puede proporcionar una solución concreta a cualquier diseño de interfaz, la tercera y última contribución trata de acotar la propuesta al ámbito de aplicaciones móviles sensibles al contexto. Así, esta contribución consiste en la definición de un conjunto de patrones de diseño de interfaces de usuario de este tipo de aplicaciones. De esta forma, este conjunto de patrones tiene la finalidad de proporcionar al *Simplificador* de aquel conocimiento necesario para simplificar los diseños de interfaz de este tipo de aplicaciones.



Figura 15. Visión global de la solución propuesta

4.2 Introducción de la Simplicidad en el Proceso de Diseño

El concepto de Simplicidad se relaciona con la sencillez de entender o comprender conceptos, ideas, procesos, etc. Su importancia dentro del diseño de productos, de dispositivos y también de interfaces de usuario, radica en que facilita el uso de un determinado producto o de una determinada interfaz, y por tanto, genera una experiencia de usuario más satisfactoria. Así por ejemplo, un caso de éxito de producto diseñado bajo este concepto es el reproductor de música iPod, el cual, a través de un diseño simple tuvo más éxito que otros productos similares que incluso ofrecían una mayor cantidad de funcionalidades [38].

Dentro del ámbito del diseño de interfaces de aplicaciones móviles, la simplicidad es un atributo que cobra especial relevancia. Varios autores así lo indican [2, 55]. Aparte, varios conjuntos de heurísticas para el diseño de interfaces de usuario referencian este concepto como una característica más a tener en cuenta para proporcionar resultados usables [60, 31, 62, 10, 57, 41]. Los motivos que justifican esta necesidad son de varios tipos.

Por un lado, las restricciones hardware de este tipo de dispositivos, y por otro, su uso intercalado en mitad de todo tipo de actividades del mundo real. Las personas usan sus dispositivos mientras recorren una calle, suben al autobús, realizan la compra semanal, etc. Así, la realización de la actividad que el usuario estaba haciendo justo antes de usar su dispositivo se ve perjudicada cuando el usuario desvía su atención para enfocarla sobre su dispositivo.

Por otro lado, en muchas ocasiones, las interfaces de usuario proporcionadas por las aplicaciones no son sencillas de usar, y el hecho de que estas sean usadas en un entorno cambiante y repleto de interrupciones y distracciones, hace que sea más difícil que los objetivos del usuario con la aplicación se cumplan.

Definición de Simplicidad

La definición de Simplicidad seguida en este trabajo es la proporcionada por John Maeda en [38]. Así, según el autor, la Simplicidad consiste en restar lo obvio o secundario y dar importancia a lo verdaderamente significativo e importante [38].

Dicha definición encaja con varias de las necesidades existentes en el diseño de interfaces de usuario de aplicaciones móviles. En primer lugar, este tipo de aplicaciones

rueda sobre dispositivos que tienen una pequeña pantalla, y por ello, necesitan reflejar toda la información en un pequeño conjunto de píxeles. De esta forma, cobra importancia la necesidad de que las interfaces se centren en aquello significativo. En segundo lugar, la utilización de los dispositivos dentro de cualquier actividad hace que usuario desee completar rápidamente la tarea puntual a realizar con su dispositivo para enfocar de nuevo su atención en la actividad que estaba realizando anteriormente.

4.2.1 Guías para Proveer Simplicidad

Este trabajo propone un conjunto de guías de simplicidad aplicadas al ámbito del diseño de interfaces de aplicaciones móviles. Las guías que se proponen en este punto se han definido a partir de las Leyes de la Simplicidad de John Maeda [38]. Estas leyes, las cuales podemos ver en la Tabla 4, proporcionan metáforas relacionadas con la simplicidad que son aplicables a diversas áreas como por ejemplo el diseño, los negocios, la tecnología o la vida.

Ley	Descripción
1 Reducir	La forma más simple de alcanzar simplicidad es reducir adecuadamente.
2 Organizar	La organización hace que un sistema enorme parezca menor.
3 Tiempo	El ahorro de tiempo hace sentir simplicidad.
4 Aprender	El conocimiento hace todo más sencillo.
5 Diferencias	La simplicidad y la complejidad se necesitan mutuamente.
6 Contexto	Aquello situado en la periferia de la simplicidad no es periférico.
7 Emoción	Sentir emociones es mejor que no sentir las.
8 Confianza	Confiamos en la simplicidad.
9 Fracaso	Algunas cosas nunca pueden ser sencillas.
10 La única	La Simplicidad consiste en restar lo obvio y añadir lo significativo.

Tabla 4. Leyes de la Simplicidad de John Maeda

En particular, el conjunto de guías de Simplicidad que se proporciona interpreta y adapta 7 de las leyes originales. En concreto adapta las guías: “*reducir*”, “*organizar*”, “*tiempo*”, “*aprender*”, “*contexto*”, “*emociones*”, “*la única*”.

La adaptación de estas leyes, persigue esencialmente dos objetivos. El primero de ellos, es el de **determinar y organizar la información, operaciones y componentes a incluir en la interfaz de usuario**. Así, cuatro de las siete guías se ubican bajo este objetivo. En cambio, el segundo de ellos trata de introducir mecanismos, que agrupa las guías 5, 6 y 7, trata de **transmitir buenos sentimientos** sobre esa información determinada y organizada previamente a través de mecanismos de interacción más intuitivos, mejoras que permitan ahorrar tiempo al usuario.

A continuación se detallan las guías propuestas junto a algunos ejemplos concretos que tratan de facilitar su entendimiento:

- **Guía 1: Quitar información redundante o irrelevante.**

Comprobar si el diseño contiene información no relevante y eliminarla en caso afirmativo.

- **Guía 2: Seleccionar lo esencial.**

Cuando ya no quede información irrelevante, seleccionar cual es el factor esencial, la información más importante o la que el usuario final desea ver.

Un ejemplo de la aplicación de estas dos guías es el que se muestra a continuación. Esta figura muestra un componente gráfico que permite la realización de operaciones sobre una tarea particular de una lista de tareas. En dicha figura se muestra un conjunto de botones que compite por un pequeño espacio en la interfaz gráfica. Estos botones corresponden diversas operaciones (completar tarea, eliminar tarea, editar tarea, ver localización de la tarea, o enviar tarea a contacto. Sin embargo, únicamente la operación “completar tarea” tiene especial relevancia. Así, una propuesta de simplificación sería aquella que trata de beneficiar la realización de esta tarea particular sobre la de las demás (Guía 2: seleccionar lo esencial), agrupando y trasladando el resto de operaciones a una interfaz secundaria. Aparte, el diseño inicial presenta la edición del atributo de prioridad de la tarea. Esta operación es redundante con la existencia de la operación “editar tarea”, Así, la propuesta de simplificación realizada elimina dicha operación.



Figura 16. Ejemplo de aplicación de las guías de simplicidad n°1 y 2

- **Guía 3: Complementar y contextualizar lo esencial.**

Tras seleccionar lo esencial, complementarlo teniendo en cuenta el contexto del usuario, adaptando cómo la información es mostrada.

Un ejemplo concreto de la aplicación de esta guía se muestra en la Figura 17. En particular, el componente mostrado en la parte izquierda de la figura, representa los horarios de asignaturas de un alumno de una fecha particular. Sin embargo, dicho componente no tiene en cuenta el contexto del usuario, en concreto, el instante de tiempo actual. Así, en el diseño inicial se listan todos los horarios de asignaturas del alumno por igual. En cambio, una solución propuesta que utiliza el instante de tiempo actual como información de contexto que permite adaptar la interfaz mostrada es la que se muestra en la parte derecha de la figura. En ella, se resalta la siguiente clase a la que el horario debe asistir gracias a la información de contexto capturada.

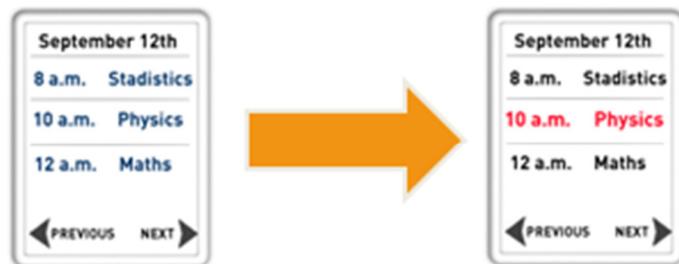


Figura 17. Ejemplo de aplicación de la guía de simplicidad nº3

- **Guía 4: Organizar lo secundario.**

Tras complementar lo esencial, agrupar y organizar el resto de funcionalidad e información que debe conservar. Posteriormente, recolocarla en un nivel secundario para no afectar la interfaz en la que se sitúa lo esencial.

- **Guía 5: Mejorar la intuición.**

Emplear metáforas o conocimiento popular de la vida real que provean una interacción más intuitiva y tratar de mejorar los mecanismos de interacción.

Un ejemplo concreto de las guías nº4 y 5 es el mostrado en la Figura 16, la cual organiza operaciones poco relevantes en una interfaz secundaria a la que se da acceso mediante un enlace que aprovecha el conocimiento popular del funcionamiento de un hipervínculo como metáfora para facilitar la navegación al usuario.

- **Guía 6: Ahorrar tiempo.**

Proporcionar mejoras que eviten malgastar el tiempo del usuario a través de esperas o interacciones lentas.

Un ejemplo concreto de interfaz de usuario que requiere la espera del usuario para realizar una determinada tarea es el que se muestra en la Figura 18. En particular, esta interfaz utiliza un diálogo de aviso que se muestra cuando el usuario completa una tarea. Este diálogo se utiliza para permitir que el usuario aborte la realización de una tarea no deseada y pueda evitar un estado no deseable. Sin embargo, introduce una espera de varios segundos cada vez que el usuario desea completar una tarea.



Figura 18. Interfaz de usuario que requiere la espera del usuario

- **Guía 7: Generar emociones.**

Lograr que los usuarios tengan buenos sentimientos sobre el uso de la aplicación para proporcionar una interacción satisfactoria.

Un ejemplo concreto de aplicación de la guía nº7 es el que se muestra en la Figura 19. En particular, el componente gráfico al que se hace referencia tiene el objetivo de informar al usuario de la necesidad de compra de varios productos. Sin embargo, la interfaz proporcionada no indica la importancia o urgencia de la necesidad de estos productos. En cambio, la solución propuesta trata de facilitar al usuario la decisión de ir o no al supermercado, y que por tanto, le hace sentir que la aplicación “*se preocupa por él*”. En concreto, la solución propuesta hace uso de una imagen más representativa que indica la urgencia de la necesidad de productos, cuando éstos sean muy necesarios, e indica al usuario la distancia entre su posición actual y la del supermercado más cercano.



Figura 19. Ejemplo de aplicación de guías nº6 y 7

4.2.2 El Papel del Rol Simplificador

En esta propuesta se propone que la aplicación del conjunto de guías de simplicidad anterior sea realizada por un rol especializado en proveer simplicidad. La introducción de este rol se justifica por la necesidad de que la simplificación de diseños debe ser objetiva, y por tanto, no debe recaer en las manos de los diseñadores ya que éstos corren el riesgo de diseñar productos para ellos mismos [34]. Este hecho, justifica la dificultad de los diseñadores de simplificar sus propias soluciones. En particular, un diseñador que ha creado un diseño de interfaz conoce a la perfección cómo funciona cada pequeña pieza de la interfaz, cada componente, y la relación existente entre ellos para alcanzar un objetivo. Así, este profundo conocimiento de la solución le impide detectar qué aspectos y complejidades deben ser tratadas.

Esencialmente, la misión de este rol Simplificador es (1) detectar problemas de complejidad, entendiendo un problema de este tipo como aquel que dificulta el uso de una aplicación, y (2) proponer soluciones que traten de simplificar las complejidades detectadas y provean mecanismos de interacción más sencillos.

Con esta finalidad, esta propuesta proporciona dos mecanismos que facilitan la labor del Simplificador. El primero de ellos es el conjunto de guías de Simplicidad provisto en el anterior, el cual es aplicable sobre diseños de interfaz de aplicaciones móviles. Estas guías permiten que el Simplificador reflexione sobre si la información mostrada, la localización de operaciones, los componentes gráficos empleados y la combinación de estos factores entre sí, permite que el usuario pueda alcanzar sus objetivos de una manera satisfactoria.

El segundo de los mecanismos, consiste en el profundo conocimiento de los diseños de interacción del tipo de aplicaciones que se desea analizar. Dado que las aplicaciones pueden tener interacciones complejas que se relacionan con el tipo específico de aplicación al que pertenecen, y no con los componentes gráficos utilizados o la información mostrada, es necesario que el Simplificador conozca las peculiaridades concretas del tipo de aplicación que va a simplificar. Así por ejemplo, si la aplicación a simplificar se integra con redes sociales y con otros usuarios, será importante que el rol Simplificador conozca en primer lugar aquellas funcionalidades típicas existentes en este tipo de aplicaciones (ej. compartir estado, etiquetar fotografía), y en segundo lugar los diseños de interacción típicos utilizados para este tipo de funcionalidades. De esta manera podrá aplicar este conocimiento a la simplificación de diseños de un tipo de aplicaciones concreto.

4.2.3 El Refinamiento de Interfaces de Usuario

En este apartado se presenta paso a paso cómo tiene lugar la intervención del Simplificador dentro del proceso de diseño. Esencialmente, el refinamiento de interfaces de usuario se resume en tres etapas: la creación de diseños, la detección de complejidades, y el acuerdo entre Diseñador y Simplificador de los cambios a realizar. A continuación se detallan cada uno de estos pasos. Nótese que como el Simplificador interviene de forma iterativa hasta que finalmente se considera que los diseños son válidos, el paso 3 puede dar lugar de nuevo al paso 1 para comenzar así a otra iteración de refinamiento de interfaces.

Paso 1. Creación de diseños

El objetivo de este primer paso consiste en la creación de los diseños a partir de los requisitos capturados previamente o bien a partir de las modificaciones a realizar sobre los diseños acordados con el simplificador.

Roles implicados:

- Diseñador.

Recursos a utilizar:

- Requisitos capturados en la fase de Análisis.
- Información provista por el Simplificador.

Descripción gráfica:



Figura 20. Paso nº1 del refinamiento de diseños de interfaz de usuario

Paso 2. Detección de Complejidades y Propuesta de Soluciones

En este segundo paso, el Simplificador recibe los diseños realizados y comprueba si en ellos se aplican las guías de Simplicidad. De esta forma, el Simplificador puede detectar problemas de complejidad que se pueden relacionar con información redundante, con la necesidad de destacar aquello realmente relevante, con la falta de información que complementa lo esencial de la interfaz, aquello para detectar complejidades, problemas de organización, de mecanismos de interacción no intuitivos, etc. Sin embargo, en determinados contextos, estas guías pueden tener una difícil aplicación porque algunas

decisiones de diseño pueden depender del dominio específico de la aplicación concreta. Por este motivo, en algunos casos será necesario un conocimiento del dominio de aplicación que complemente la aplicación de estas guías.

Roles implicados:

- Simplificador.

Recursos a utilizar:

- Diseños provistos por el Diseñador.
- Guías de simplicidad.
- Conocimiento específico del dominio de la aplicación.

Descripción gráfica:



Figura 21. Paso nº2 del refinamiento de diseños de interfaz de usuario

Paso 3. Acuerdo de Modificaciones

En este último paso se definen los cambios concretos a realizar en los diseños, es decir, las modificaciones que permitirán simplificar las complejidades detectadas. En particular, estas decisiones deben definidas mediante un acuerdo entre el diseñador y el simplificador ya que por un lado encontramos que los diseñadores deben tener en cuenta en todo momento que los requisitos de la aplicación se cumplen igualmente pese a los cambios realizados en la interfaz. Asimismo, aparte de los requisitos establecidos, pueden existir otros tipos de limitaciones que no permitan cualquier tipo de cambio

deseado por el simplificador como pueden ser por ejemplo limitaciones tecnológicas o limitaciones en los recursos disponibles. Así, es responsabilidad del simplificador la realización de propuestas que solventen las complejidades bajo las limitaciones existentes.

Roles implicados:

- Diseñador.
- Simplificador.

Recursos:

- Diseños provistos por el Diseñador.
- Guías de simplicidad.
- Conocimiento del dominio de la aplicación.
- Requisitos capturados en la fase de Análisis.

Descripción gráfica:



Figura 22. Paso nº3 del refinamiento de diseños de interfaz de usuario

4.2.4 Discusión

Tanto la definición de guías basadas en proveer simplicidad en la creación de diseños de interfaz como la introducción de un rol centrado únicamente en proveer simplicidad dentro de las mismas son dos mecanismos que tratan de ayudar a simplificar diseños de interfaz.

Uno de los puntos a tener en cuenta para la introducción de la propuesta es la relación existente entre el Diseñador y el Simplificador. En particular, se hace necesaria la validación final de los diseños por parte del Simplificador ya que en determinados equipos de trabajo pueden surgir discusiones con diseñadores extremadamente persistentes en sus decisiones y reacios a realizar cambios.

Finalmente, dada la heterogeneidad existente en los tipos de aplicaciones móviles y la imposibilidad de estas guías de dar una cobertura total a cualquier diseño de interfaz, se hace necesaria o bien la instanciación de las guías a dominios concretos tal como realizan algunos autores [1, 58], o bien que el rol Simplificador encargado de la refactorización de diseños tenga un profundo conocimiento sobre el dominio de la aplicación a evaluar. En concreto, esta segunda alternativa es la tomada en este trabajo. Para ello, en la siguiente sección se define un conjunto de patrones de diseño de interfaces de usuario de aplicaciones móviles sensibles al contexto, los cuales especifican el comportamiento de la interacción de este tipo de aplicaciones sirviendo de conocimiento al Simplificador para que éste pueda simplificar interfaces de usuario de este tipo de aplicaciones.

4.3 Patrones de Diseño de Aplicaciones Móviles Sensibles al Contexto

En esta sección se presenta un conjunto de patrones de diseño de interfaces de usuario de aplicaciones móviles. En concreto, el conjunto de patrones que se propone se acota al dominio de aplicaciones móviles sensibles al contexto, y más concretamente a aquellas aplicaciones que hacen uso de la localización y el tiempo por ser estos atributos del contexto del usuario los que mayormente se utilizan dentro de las aplicaciones móviles actuales tal como se indica en el [Capítulo 2](#) de este documento.

El objetivo que se persigue con esta tercera contribución es proporcionar al rol Simplificador el conocimiento necesario para poder simplificar diseños de interfaz de este tipo de aplicaciones. Así, este conjunto de patrones muestra el funcionamiento de los diseños de interacción típicos en las aplicaciones móviles sensibles al contexto.

Los patrones de diseño definen soluciones a problemas frecuentes dentro de un determinado contexto. Su utilización, motivada principalmente por el principio de reusabilidad introduce varias mejoras entre las que encontramos:

- La reducción del tiempo dedicado a tareas de diseño en nuevos proyectos.
- La mejora en la calidad de los diseños.
- La mejora de la comunicación entre miembros del equipo de desarrollo.

El resto de la sección se estructura de la siguiente forma: la subsección 4.3.1 introduce datos del análisis de aplicaciones realizado. La subsección 4.3.2 describe el conjunto de patrones de diseño detectados.

4.3.1 Análisis Realizado

Objetivos a detectar en el Análisis

Los objetivos concretos del análisis realizado son:

1. Detectar de qué manera concreta las aplicaciones móviles integran el tiempo y la localización en sus funcionalidades.
2. Detectar los patrones de interacción de interfaces de usuario de este tipo de aplicaciones.

Procedimiento de identificación de patrones

El procedimiento seguido para definición del conjunto de patrones ha consistido en:

- 1. Selección del tipo de aplicaciones a analizar**

Tal como se ha justificado en el [Capítulo 2](#), la propuesta se ha acotado a las aplicaciones sensibles a la localización y al tiempo.

- 2. Selección de las aplicaciones concretas a analizar**

En particular, el conjunto de aplicaciones analizado es el que se muestra en el [Anexo I](#) de este documento. Estas aplicaciones, han sido seleccionadas del mercado de aplicaciones de la compañía Apple.

- 3. Análisis de las aplicaciones**

Para cada aplicación se han analizado las interfaces provistas, detectando los objetivos que éstas persiguen y las características que permiten diferenciarlas entre sí. De esta forma, se identificó por ejemplo el criterio de clasificación de patrones de aplicaciones sensibles al contexto que separa las interfaces que muestran localizaciones sobre un mapa de las que lo hacen sobre un listado.

Asimismo, se identificó que para los patrones de aplicaciones móviles sensibles al tiempo, la longitud del rango de tiempo (ej. día, semana, mes), era la principal característica para clasificar los patrones.

Resultados del Análisis

Como resultados de estos análisis, por una parte se han extraído los usos frecuentes de los atributos citados dentro de las aplicaciones móviles actuales, los cuales han sido tratados y comentados dentro del Capítulo 2 de este documento. Por otra, se han extraído los patrones de diseño básicos de representación de información de este tipo de aplicaciones. Estos patrones son listados y especificados en la siguiente subsección.

Plantilla para la Especificación de Patrones

Para su especificación y documentación, los patrones definidos hacen uso de una estructura que trata de responder a las siguientes preguntas:

- ¿Qué problema resuelve?
- ¿Cómo lo resuelve?
- ¿Cuándo se requiere su aplicación?

Así, la estructura utiliza los siguientes atributos:

- Nombre del patrón.
- Problema: la descripción del problema que motiva el uso del patrón.
- Características: particularidades que lo identifican y permiten diferenciar de otros patrones.
- Descripción de la solución: solución que se resuelve el problema establecido.
- Beneficios e inconvenientes: indicando pros y contras relacionados con la capacidad de interacción, y haciendo referencia a otros patrones con los que son comparables.
- Representación gráfica, abstracta o concreta, de la solución o de algún ejemplo: imágenes que describen el patrón o muestran un ejemplo concreto de aplicación que utilice el patrón.

4.3.2 Patrones Detectados

El conjunto de patrones detectados sigue la taxonomía que se describe a continuación. Tal como podemos ver, en primer lugar, los patrones se diferencian en dos grupos en función del atributo de contexto al que hacen referencia: aquellos relacionados con la localización del usuario, y aquellos relacionados con el tiempo.

Por una parte, dentro de los patrones detectados en las aplicaciones analizadas que hacen uso de la localización, los patrones se clasifican en tres categorías. La primera de ellas agrupa los patrones que visualizan localizaciones concretas sobre un mapa. En cambio, la segunda de ellas muestra localizaciones concretas sobre un listado.

Por otra, los patrones detectados en las aplicaciones que hacen uso del tiempo se clasifican en función de si organizan objeto asociados a un determinado rango de fechas concreto como puede ser un único día, semana o mes, o si agrupan todos los objetos por igual. De esta forma, el conjunto global de patrones es el que se muestra a continuación:

Patrones basados en localización

Patrones de visualización de puntos sobre un mapa

1. Patrón “Mapa de puntos de un único tipo de instancia”.
2. Patrón “Mapa de puntos de un único tipo de instancia con distintos iconos”.
3. Patrón “Mapa de puntos de un único tipo de instancia con iconos personalizados”.
4. Patrón “Mapa de puntos de varios tipos de instancias”.
5. Patrón “Mapa de puntos con agrupación de iconos”.
6. Patrón “Información ampliada de un marcador simple”.
7. Patrón “Información ampliada de un marcador agrupado”.

Patrones de visualización de puntos ordenados por cercanía

8. Patrón “Lista de puntos de un único tipo de instancia”.
9. Patrón “Lista de puntos de un único tipo de instancia con atributo diferenciador”.
10. Patrón “Lista de puntos de un único tipo de instancia con iconos personalizados”.

Patrones basados en el tiempo

Patrones que no organizan objetos en ningún rango de fechas particular

11. Patrón “Carrusel ordenado de instancias”
12. Patrón “Listado ordenado de instancias”

Patrones que organizan objetos en un rango de fechas particular

13. Patrón “Vistas mensual completa”
14. Patrón “Vistas mensual maestro-detalle ”
15. Patrón “Vista semanal general”
16. Patrón “Vista semanal detallada”
17. Patrón “Vista diaria”

Patrón “Mapa de puntos de un único tipo de instancia”

Problema que resuelve: la representación de una colección de localizaciones de un único tipo de instancia (ej. restaurantes) sobre un mapa que muestra ubicación de dichas localizaciones al usuario.

Características:

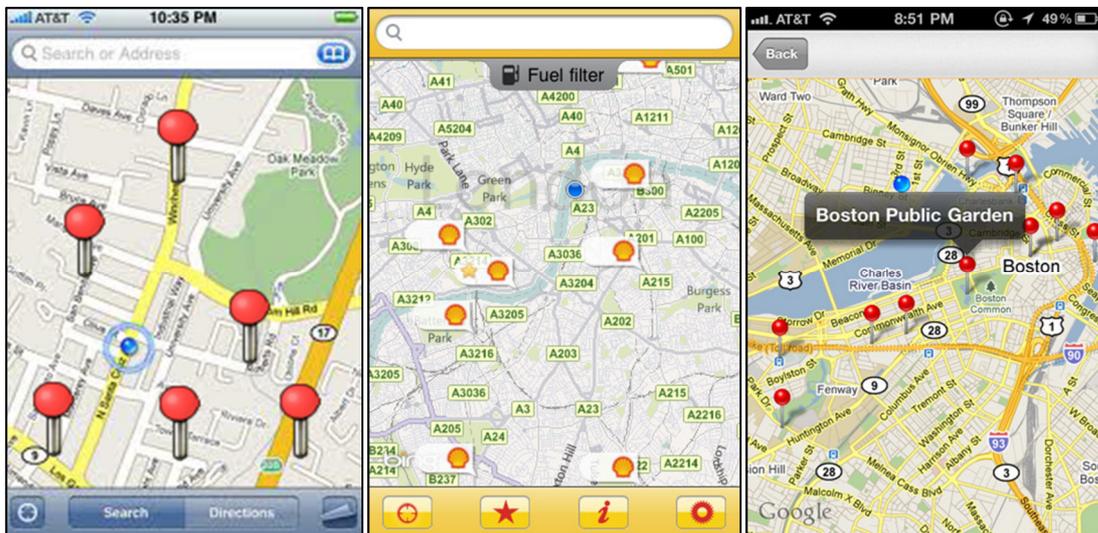
- incluye un único tipo de instancia (ej. restaurantes).

Descripción de la solución: La representación visualiza cada punto con un icono establecido por defecto o especificado por el usuario.

Beneficios: la inclusión de un único icono para la representación de puntos permite diferenciar claramente las localizaciones concretas de la ubicación del usuario.

Inconvenientes: ante situaciones en las cuales hay muchos puntos cercanos a una misma ubicación, la vista superpone los marcadores entre sí dificultando su identificación y su selección.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón: “Mapa de puntos de un único tipo de instancia con distintos iconos”

Problema que resuelve: la representación de una colección de puntos de un único tipo de instancia donde los puntos se diferencian entre sí a partir de un atributo que divide el conjunto total de instancias en varios subconjuntos.

Características:

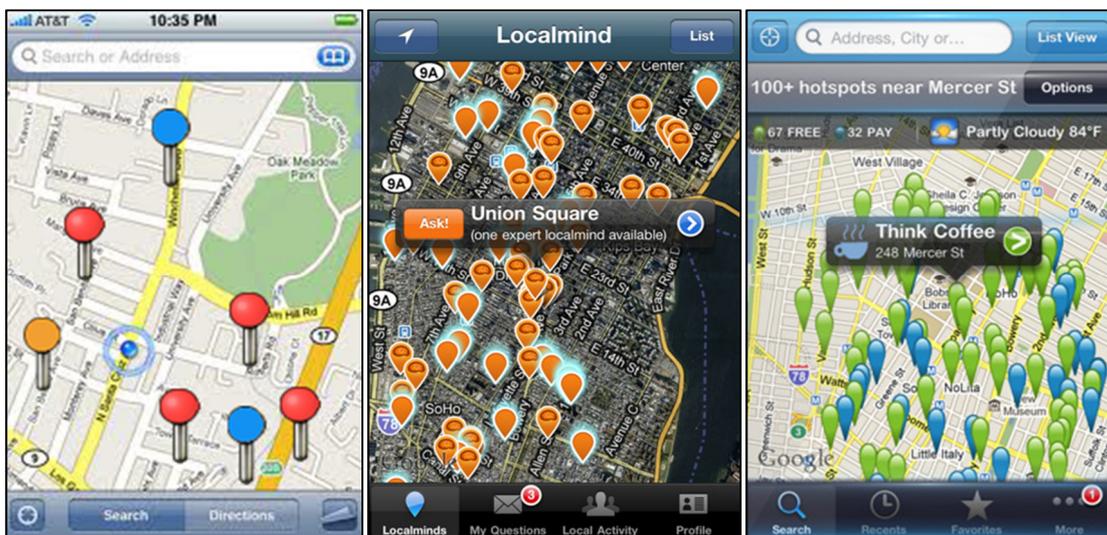
- Incluye un único tipo de punto (ej. restaurantes).
- Estos puntos se diferencian entre sí acorde al atributo diferenciador del tipo de instancia (ej. precio estimado por persona del restaurante).

Descripción de la solución: la representación visualiza cada punto con un icono establecido por defecto o especificado mediante un icono. Dicho icono emplea un color diferente acorde al valor o significado del atributo diferenciador del tipo de instancia.

Beneficios: la existencia de puntos o iconos de un mismo tipo pero de diferente color indica al usuario que dichas instancias tienen alguna diferencia entre sí.

Inconvenientes: ante situaciones en las cuales hay muchos puntos cercanos a una misma ubicación, la vista superpone los marcadores entre sí dificultando su identificación y su selección. Además, en algunos casos, la selección de los colores a emplear para la diferenciación de los iconos para que estos aporten un significado puede ser una tarea complicada.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Mapa de puntos de un único tipo de instancia con iconos personalizados”

Problema que resuelve: la representación de una colección de puntos de un único tipo de instancia donde existe una determinada característica que lo identifica completamente entre el resto de instancias.

Características:

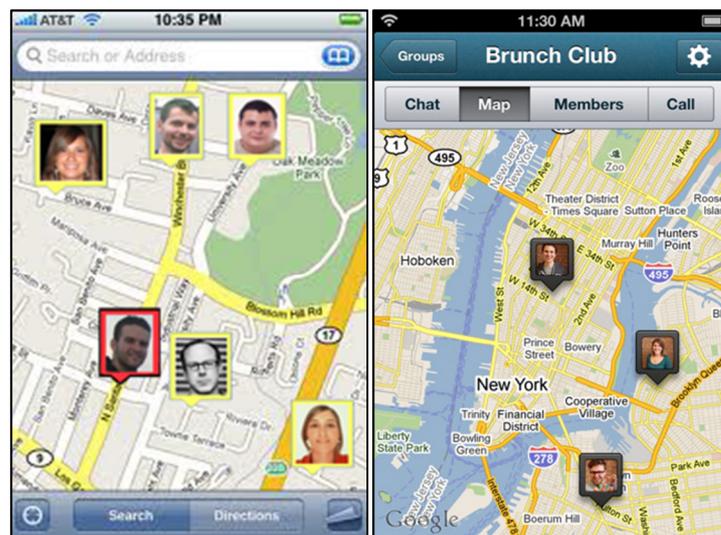
- Incluye un único tipo de punto (ej. contactos de la agenda del usuario).
- Estos puntos se diferencian entre sí acorde al atributo identificador del tipo de instancia (ej. fotografía de un contacto).

Descripción de la solución: la visualización representa cada instancia con un único tipo de icono personalizado mediante que permite la identificación clara de un punto particular.

Beneficios: la existencia de puntos con un icono totalmente personalizado permite la identificación clara del punto.

Inconvenientes: ante situaciones en las cuales hay muchos puntos cercanos a una misma ubicación, la vista superpone los marcadores entre sí dificultando su identificación y su selección.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Mapa de puntos de varios tipos de instancias”

Problema que resuelve: la representación de una colección de puntos de varios tipos de instancias.

Características:

- Incluye varios tipos de puntos (ej. restaurantes, hoteles y gasolineras)

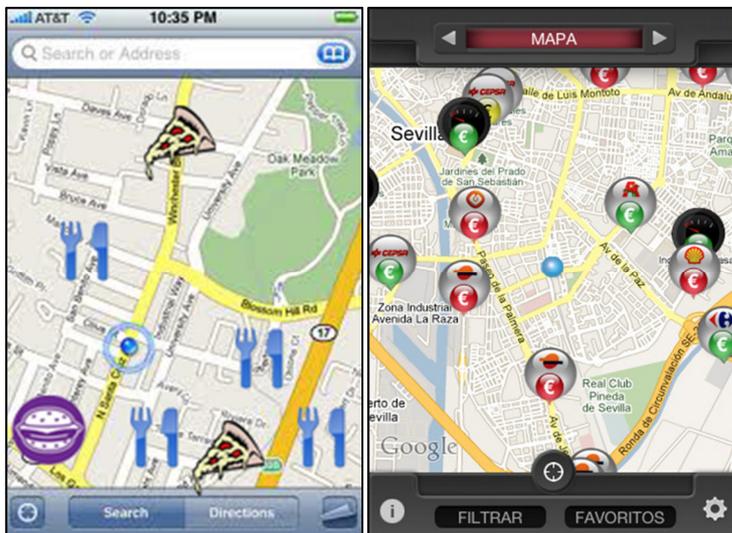
Descripción de la solución: la visualización representa con un marcador distinto cada al tipo de instancia que se representa.

Beneficios: la existencia de distintos iconos para cada tipo de instancia ayuda al usuario a diferenciar dichos tipos de instancia.

Inconvenientes: es difícil distinguir el icono que representa la ubicación del usuario en aquellos casos en que haya demasiados tipos de iconos.

Ante situaciones en las cuales hay muchos puntos cercanos a una misma ubicación, la vista superpone los marcadores entre sí dificultando su identificación y su selección.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Mapa de puntos con agrupación de iconos”

Problema que resuelve: la representación de una colección grande de puntos de un único tipo de instancia.

Características:

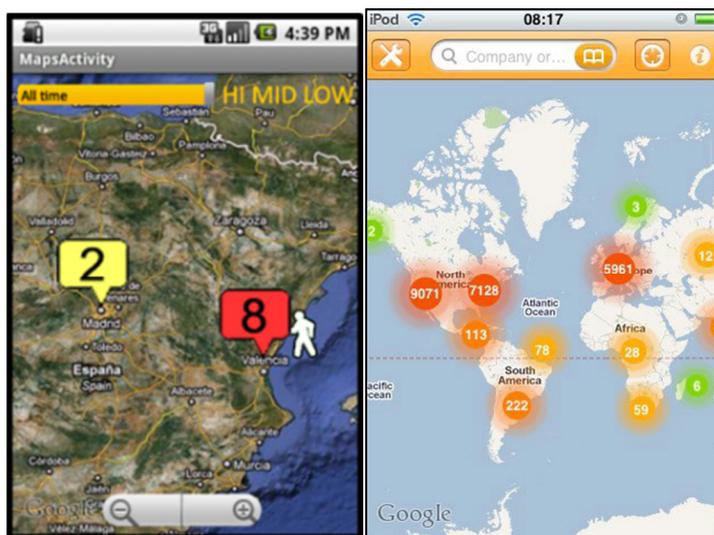
- Incluye un único tipo de punto (ej. contactos de la agenda del usuario).

Descripción de la solución: la representación visualiza la colección de puntos de forma agrupada en base a la localización en la que se ubican. De esta forma, a medida que se detalla la vista sobre una zona concreta, los marcadores agrupados de estas zona se desagrupan para permitir una mejor identificación de los puntos. Asimismo, un número y el empleo de varios colores informan al usuario de cuantos marcadores individuales se incluyen dentro de un marcador agrupado.

Beneficios: la visualización soluciona el problema de la superposición de marcadores, la cual impide una correcta identificación de un marcador particular y la interacción con el mismo.

Inconvenientes: la visualización no aporta información que permita identificar los tipos de instancias que agrupa. Por ello, se propone que se utilice esta vista únicamente a colección de puntos de un único tipo de instancia.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Información ampliada de un marcador simple”

Problema que resuelve: la obtención de información ampliada de un punto de interés.

Características:

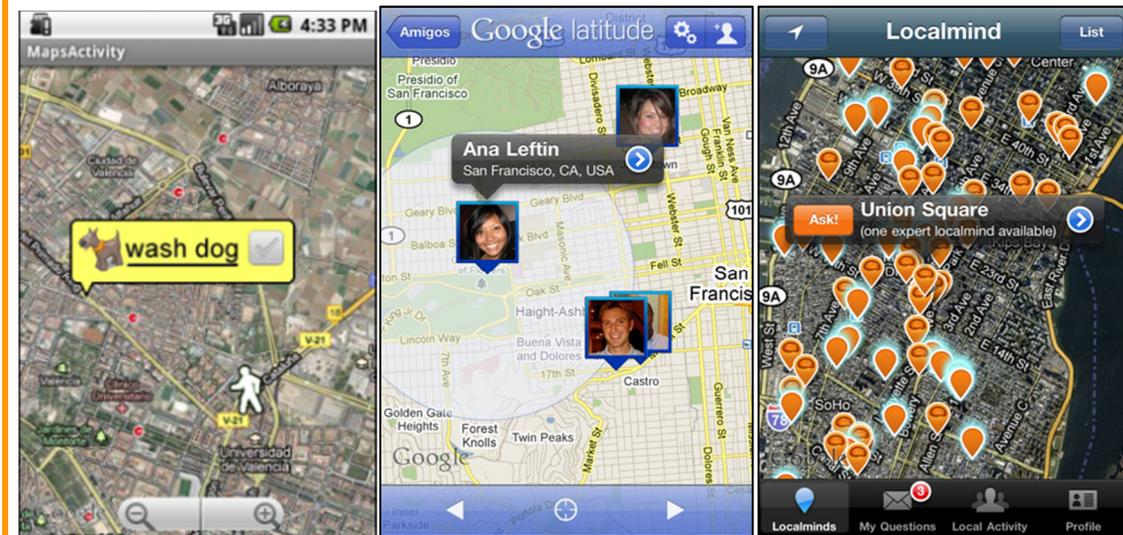
- La información ampliada se muestra cuando el usuario interactúa con un marcador particular.

Descripción de la solución: la visualización muestra capa superpuesta u “*overlay*” con información adicional sobre el marcador seleccionado. Esta capa superpuesta incluye únicamente la información más importante del punto y proporciona un enlace hacia una vista más detallada.

Beneficios: la visualización muestra el detalle básico y aquella información de relevancia relacionada con el punto.

Inconvenientes: no se han encontrado.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Mapa de puntos de un único tipo de instancia”

Problema que resuelve: la obtención de información ampliada de un marcador que agrupa varios puntos de interés.

Características:

- La información ampliada se muestra cuando el usuario interactúa con un marcador particular.

Descripción de la solución: la visualización muestra capa superpuesta u “*overlay*” con información adicional sobre el marcador seleccionado. Esta capa superpuesta incluye únicamente la información más importante de cada punto agrupado bajo el marcador proporcionando un enlace hacia una vista más detallada.

Beneficios: la visualización muestra el detalle básico y aquella información de relevancia relacionada con cada punto.

Inconvenientes: ante marcadores con un número elevado de puntos, la vista requiere de un mecanismo que mejora la visualización de los puntos que se agrupan para evitar el uso de barras de desplazamiento.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Lista de puntos de un único tipo de instancia”

Problema que resuelve: la representación de una colección de localizaciones de un único tipo de instancia de manera ordenada acorde a la cercanía con la ubicación del usuario.

Características:

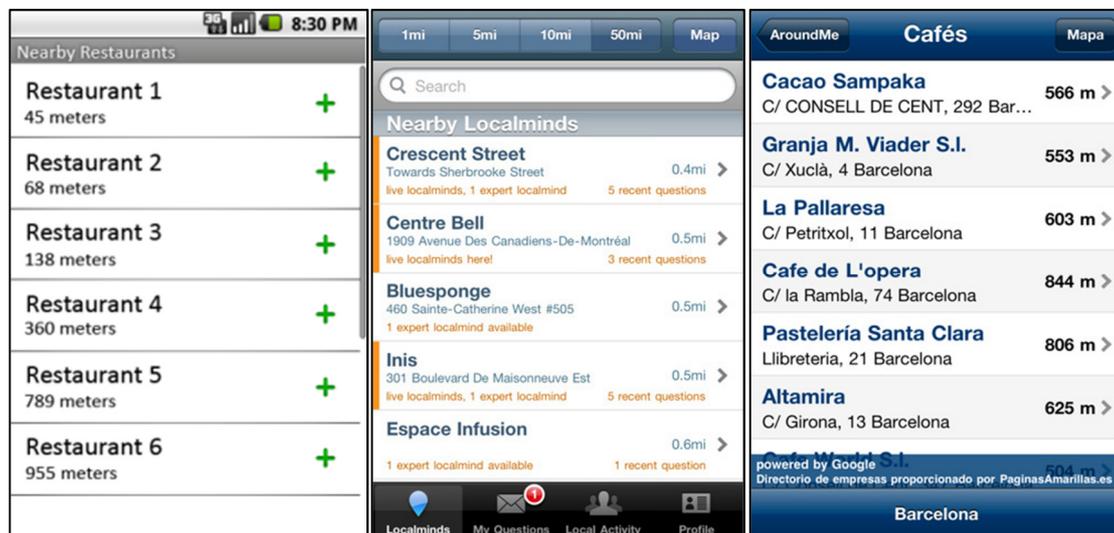
- Incluye un único tipo de punto (ej. restaurantes).

Descripción de la solución: la visualización representa un conjunto de localizaciones sobre una lista ordenada en la que los ítems se ordenan según la proximidad a la localización del usuario, la cual es mostrada.

Beneficios: esta visualización requiere menos recursos que la visualización de la vista sobre un mapa al no tener que acceder a ningún servicio de mapas.

Inconvenientes: pese a indicar la cercanía a un conjunto de ubicaciones, dicha visualización no proporciona al usuario la ubicación exacta que le permita llegar con facilidad a dicha localización.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Lista de puntos de un único tipo de instancia con atributo diferenciador”

Problema que resuelve: la representación de una colección de puntos de un único tipo de instancia donde los puntos se diferencian entre sí a partir de un atributo que divide el conjunto total de instancias en varios subconjuntos (ej. precio medio de un restaurante).

Características:

- El atributo diferenciador de cada instancia se muestra explícitamente.

Descripción de la solución: la visualización representa un conjunto de localizaciones sobre una lista ordenada en la que los ítems se ordenan según la proximidad a la localización del usuario, la cual es mostrada junto al atributo diferenciador que caracteriza cada instancia particular.

Beneficios: esta visualización requiere menos recursos que la visualización de la vista sobre un mapa al no tener que acceder a ningún servicio de mapas.

Inconvenientes: pese a indicar la cercanía a un conjunto de ubicaciones, dicha visualización no proporciona al usuario la ubicación exacta que le permita llegar con facilidad a dicha localización.

Representación gráfica abstracta y/o ejemplo concreto:



Nearby Restaurants		
Restaurant 1	expensive	+
45 meters		
Restaurant 2	expensive	+
68 meters		
Restaurant 3	cheap	+
138 meters		
Restaurant 4	cheap	+
360 meters		
Restaurant 5	expensive	+
789 meters		
Restaurant 6	cheap	+
955 meters		

Patrón “Lista de puntos de un único tipo de instancia con iconos personalizados”

Problema que resuelve: la representación de ítems de varios tipos de instancias.

Características:

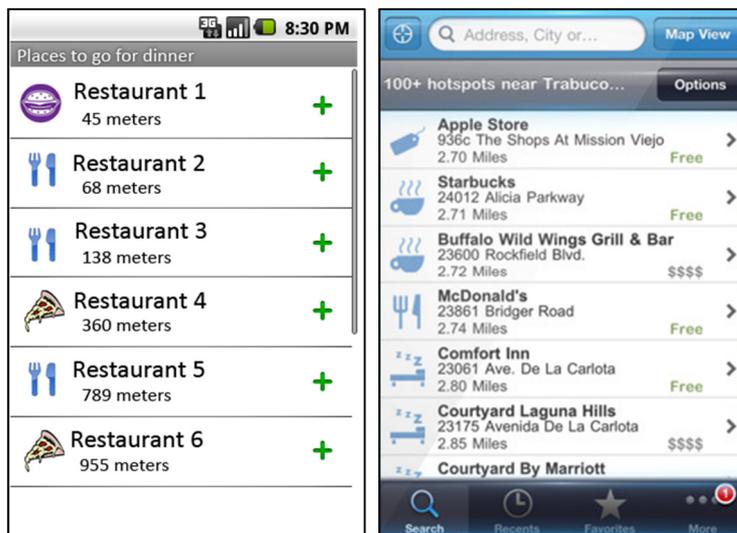
- N tipos de instancia (ej. sitios donde comer: pizzerías, restaurantes,...).

Descripción de la solución: la visualización representa un conjunto de localizaciones de varias instancias las cuales son identificadas a partir de un icono personalizo. Asimismo, dichas localizaciones se muestran sobre una lista ordenada en la que los ítems se ordenan según la proximidad a la localización del usuario.

Beneficios: esta visualización requiere menos recursos que la visualización de la vista sobre un mapa al no tener que acceder a ningún servicio de mapas. Además, la existencia de distintos iconos para cada tipo de instancia ayuda al usuario a diferenciar dichos tipos de instancia.

Inconvenientes: pese a indicar la cercanía a un conjunto de ubicaciones, dicha visualización no proporciona al usuario la ubicación exacta que le permita llegar con facilidad a dicha localización.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Carrusel ordenado de instancias”

Problema que resuelve: la representación de instancias etiquetadas con un instante de tiempo.

Características:

- Cada instancia se visualiza de manera individual.

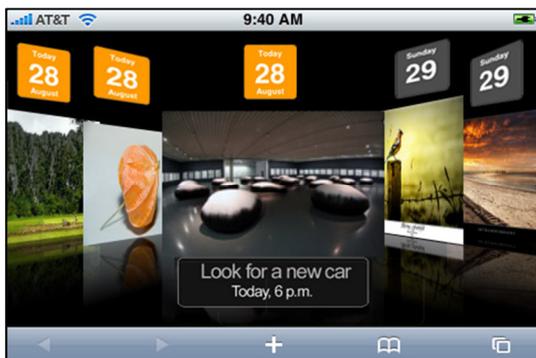
Descripción de la solución: la representación proporciona una vista en formato carrusel que ordena las instancias a lo largo de la línea del tiempo centrandolo la vista en la instancia más cercana al instante actual. Asimismo, las instancias utilizan una imagen que permite una rápida identificación de la misma, así como el instante de tiempo asociado.

Beneficios: proporciona un mecanismo de interacción táctil y más visual.

Es una vista adecuada para escenarios en los que se necesitan detalles específicos de una única instancia.

Inconvenientes: esta vista no se recomienda para la representación de un conjunto grande de instancias ya que su recorrido implicaría una excesiva interacción con el usuario.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Listado ordenado de instancias”

Problema que resuelve: La representación de instancias de diverso tipo que están etiquetados con un instante de tiempo.

Características:

- Las instancias se agrupan por días.

Descripción de la solución: descripción de la solución: Se proporciona una vista en formato lista que ordena todas las instancias a lo largo de la línea del tiempo, mostrando la siguiente tarea a realizar al principio del listado. Asimismo, para cada instancia se indica el instante de tiempo al que está asociada. Además, las instancias se agrupan por fechas.

Beneficios: esta vista proporciona una vista general de ítems que se encuentran cercanos en el tiempo ya estén estos ubicados en el día de hoy o en los sucesivos.

Inconvenientes: no proporciona una visión global del conjunto de instancias.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Vista mensual completa”

Problema que resuelve: la representación de información de todos los eventos o tareas focalizando la vista en el mes actual.

Características:

- Cubre un amplio rango de fechas.
- Emplea una vista formato calendario.

Descripción de la solución: proporciona por una parte una vista general del conjunto de eventos, tareas, avisos que tiene el usuario en el mes actual tabulando los días del mes en días de la semana, y por otra un detalle de las tareas o eventos del día seleccionado.

Beneficios: proporciona una visión global del conjunto de instancias. No requiere de ninguna interacción para acceder a la información que identifica la tarea.

Inconvenientes: la interacción con una tarea concreta es difícil dado su reducido tamaño.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Vista mensual maestro-detalle”

Problema que resuelve: la representación de información de todos los eventos o tareas focalizando la vista en el mes actual.

Características:

- Cubre un amplio rango de fechas.
- Las horas a las que están asociadas las tareas cobran importancia.
- Emplea una vista formato calendario.

Descripción de la solución: proporciona por una parte una vista general del conjunto de eventos, tareas, avisos que tiene el usuario en el mes actual tabulando los días del mes en días de la semana, y por otra, un detalle de las tareas o eventos del día seleccionado.

Beneficios: proporciona una visión global del conjunto de instancias asociadas a un mes particular sin abrumar al usuario ya que detalla únicamente las instancias a un día particular. Además, la interacción con una instancia particular no se ve limitada por el tamaño del componente en el que se incluye.

Inconvenientes: requiere una interacción extra del usuario para visualizar las instancias asociadas a una fecha asociada.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Vista semanal abstracta”

Problema que resuelve: la representación de información de todos los eventos o tareas focalizando la vista en la semana actual.

Características:

- Cubre un reducido rango de fechas.
- Las horas de las tareas no son importantes.
- Emplea una vista tabular

Descripción de la solución: proporciona una vista general del conjunto de eventos, tareas, avisos que tiene el usuario en la semana actual distribuyendo los días en el eje horizontal y colocando bajo dichos días las instancias mediante un icono representativo.

Beneficios: la visualización es clara y no está sobrecargada con multitud de etiquetas.

Inconvenientes: la utilización de iconos puede no ser suficiente en determinados contextos para identificar adecuadamente una tarea.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Vista semanal detallada”

Problema que resuelve:

La representación de información de todos los eventos o tareas focalizada sobre la semana actual.

Características:

- Cubre un reducido rango de fechas.
- La hora asociada a cada instancia cobra importancia.
- Emplea una vista horario.

Descripción de la solución: La visualización proporciona una vista general del conjunto de eventos, tareas, avisos que tiene el usuario en la semana actual, distribuyendo los días en el eje horizontal y las horas en el eje vertical y representando las instancias sobre dicha tabla en formato horario.

Beneficios: Esta visualización proporciona una vista adecuada para aquellos contextos o escenarios en los que es necesaria una vista global de un conjunto de instancias ubicadas en una semana.

Inconvenientes: El reducido tamaño de las casillas proporcionadas en el diseño limita la interacción del usuario. Asimismo, el solape de instancias en una misma franja horario dentro de un mismo día introduce un conflicto en la representación de información.

Representación gráfica abstracta y/o ejemplo concreto:



Patrón “Vista diaria detallada”

Problema que resuelve:

La representación de información de todos los eventos o tareas focalizada sobre el día actual.

Características:

- Cubre únicamente un único día.
- La hora asociada a cada instancia cobra importancia.

Descripción de la solución:

La representación proporciona una vista de instancias centrada en el día actual. A través de ella, dichas instancias se clasifican a lo largo de las diferentes horas del día.

Beneficios:

Esta representación proporciona una vista clara de un conjunto de instancias asociadas a un día concreto. Por este motivo, la selección de una instancia particular no se ve perjudicada por el tamaño del componente gráfico que lo representa.

Inconvenientes:

El solape de instancias en la misma franja horaria puede ser conflictivo en la representación de información.

Representación gráfica abstracta y/o ejemplo concreto:



4.3 Conclusiones

En este capítulo se introducen mecanismos para simplificar diseños de interfaz de aplicaciones móviles. En concreto se introduce un conjunto de guías que sirve para detectar complejidades en los diseños y sirve para la creación de diseños de interfaz sencillos. Asimismo, para aplicar este conjunto de guías sobre las interfaces, en este capítulo se ha introducido un rol que tiene la misión de aplicar este conjunto de guías para detectar complejidades en los diseños y proponer soluciones a las mismas.

Uno de los problemas de las guías provistas para proveer simplicidad es que tienen que ser interpretadas para poder producir una solución a un diseño concreto. Así, ante esta limitación, en este capítulo se han introducido un conjunto de patrones de diseño de aplicaciones móviles que indica los comportamientos frecuentes de interacción encontrados en las aplicaciones de este dominio. Con esto, lo que perseguimos es que las guías definidas para proveer simplicidad aporten soluciones concretas a un dominio concreto.

Finalmente, la utilización de patrones de diseño de interfaces de usuario aporta beneficios en la creación de aplicaciones móviles. Esto es debido a que estas aplicaciones tienen a tener un ciclo de desarrollo rápido por la rápida evolución del mercado de aplicaciones rápidamente. Pese a la detección de los patrones de diseño típicos del tipo de aplicaciones a la que se acota la propuesta, algunas visualizaciones requieren mejoras que permitan mostrar colecciones de puntos más grandes, un hecho que choca directamente con la necesidad de poder seleccionar y obtener información de un punto particular de una forma sencilla.

Capítulo 5

Caso de Estudio

Con la finalidad de poner en práctica nuestra propuesta, hemos aplicado la simplificación de diseños de interfaz a un caso de estudio concreto. En particular, la propuesta ha sido aplicada a una aplicación móvil sensible al contexto para la gestión de tareas personales. Este capítulo describe la experiencia obtenida a lo largo de las diversas intervenciones del rol Simplificador, en las que los diseños de interfaz de usuario se han modificado y simplificado mediante la aplicación de las guías de Simplicidad provistas y el conjunto de patrones de diseño definido anteriormente.

El resto del capítulo se estructura de la siguiente forma: el apartado 5.1 describe la aplicación utilizada como caso de estudio. La sección 5.2 describe los diseños iniciales realizados por el diseñador. La sección 5.3 indica las complejidades encontradas por el Simplificador, indicando para cada una de ellas los recursos utilizados para su detección. Es decir, las guías de simplicidad utilizadas y los patrones de diseño aplicados. Igualmente indica las modificaciones propuestas a dichas complejidades. Finalmente, la sección 5.4 concluye el capítulo.

5.1 Descripción General de la Aplicación

La aplicación móvil que hemos utilizado para este caso de estudio es una aplicación móvil sensible al contexto que permite gestionar una lista de tareas. A través de esta aplicación, los usuarios pueden organizar y completar sus tareas desde cualquier localización mediante su dispositivo móvil.

Para facilitar la organización y visualización de tareas, la aplicación utiliza tres visualizaciones diferentes que muestran el conjunto de tareas del usuario en base a la

localización del usuario, al tiempo, y a la prioridad de las tareas. De esta forma, en función de las necesidades del usuario o del contexto en el que éste se ubique, la aplicación permite utilizar la vista que mejor se adecue a la situación. A continuación se sintetizan brevemente las tres visualizaciones propuestas:

- *Cloud View* representa las tareas maximizando la utilización de la prioridad.
- *Timeline View* muestra las tareas a través de la línea del tiempo.
- *Map View* representa dichas tareas de forma localizada sobre un mapa.

Aparte, más allá de las visualizaciones utilizadas para representar las tareas, otro requisito a cubrir por la aplicación es la introducción de las operaciones: “*completar tarea*”, “*editar tarea*”, “*eliminar tarea*”, “*enviar tarea a un contacto*” y “*ver localización*”.

5.2 Diseños Iniciales Propuestos

Con el fin de cubrir los requisitos establecidos, los diseños iniciales propuestos por el diseñador fueron:

- **Cloud View: vista que representa las tareas en función de la prioridad**

Esta visualización proporciona una vista general de las tareas pendientes, las cuales son diferenciadas gráficamente en base a su prioridad. Para ello, la visualización muestra el conjunto de tareas como una nube de etiquetas tal como podemos ver en la Figura 23. De esta forma, utilizando esta metáfora, cada etiqueta se corresponde con el nombre de una tarea, modificando tanto el tamaño de fuente empleado como su color en función de la prioridad establecida. Así, las tareas con prioridades más altas destacan sobre las tareas de menor prioridad y el usuario puede identificar claramente aquellas tareas más prioritarias.

Respecto a la interacción con una tarea, el diseñador propone que tras la selección de la misma, se muestre un componente gráfico que incluya tanto información propia de cada tarea como el conjunto de operaciones que se permite realizar sobre una tarea. De esta forma, este componente, el cual podemos ver en la Figura 24, incluye: un icono representativo de la tarea, el nombre de la tarea, un botón para cada operación soportada, un conjunto de etiquetas para modificar la prioridad de la tarea y un botón para aplicar los cambios realizados.

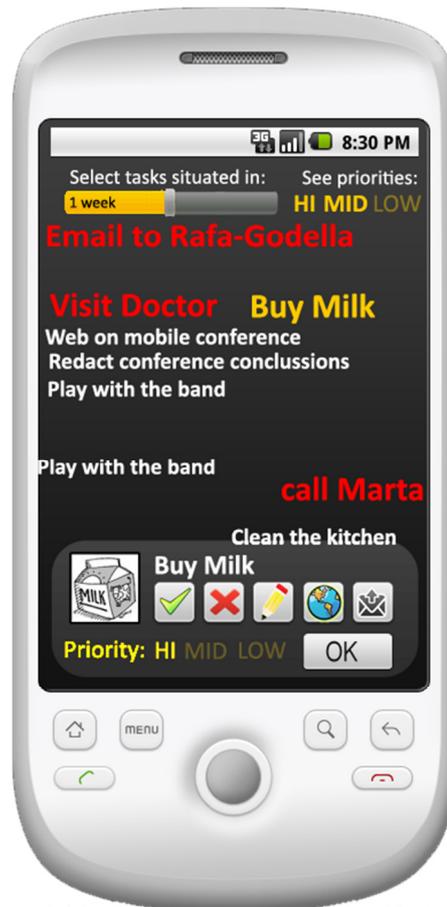


Figura 23. Diseño inicial de la vista Timeline View



Figura 24. Detalle del componente gráfico mostrado tras la selección de una tarea

- Timeline View:** vista que representa las tareas sobre la línea del tiempo
 Esta vista proporciona una visión global, y ordenada sobre la línea del tiempo, de las tareas a realizar. Para cubrir esta necesidad, la propuesta del diseñador representa el orden temporal de las tareas en base a un conjunto de flechas que indica el orden y la precedencia entre tareas. Asimismo, el usuario puede recorrer las tareas gracias a una barra de desplazamiento horizontal.

Igualmente, como vemos en la Figura 25, el diseño inicial de esta vista agrupa en su parte superior una colección de botones que da soporte a las operaciones posibles. Además, inmediatamente debajo, se representa el conjunto de tareas a través de columnas, donde cada columna representa una tarea que se

identifica por su icono representativo, el cual es rodeado por un borde cuyo color indica la prioridad de la tarea. Aparte, cada tarea muestra el medio de transporte a utilizar para llegar a la localización debe realizarse la tarea junto a precio y tiempo estimado para llegar hasta él. Finalmente, también se muestra el precio de la tarea y el tiempo estimado que cuesta llevarla a cabo.

Respecto a la interacción con una tarea, el diseñador propuso que para poder realizar una operación sobre una tarea, el usuario debe seleccionar previamente la tarea antes de pulsar el botón correspondiente a la acción que éste pretenda llevar a cabo.



Figura 25. Diseño inicial de la vista Timeline View

- **Map View: vista que representa las tareas de forma localizada**

El objetivo de esta visualización es mostrar de forma localizada el conjunto de tareas pendientes. Con este objetivo, la visualización utiliza el icono representativo de cada tarea para ubicar dichos iconos en el mapa sobre las localizaciones correspondientes en las que dichas tareas deben ser realizadas. Asimismo, y como podemos ver en la Figura 26, el mapa incluye controles para acercar o alejar la vista, permitiendo que el usuario modifique el nivel de abstracción con el que desea ver el mapa.

Respecto a la interacción con una tarea, el diseñador propuso la misma solución que en la vista **Cloud View**, la cual mostraba una capa superpuesta en la que se incluye información de la tarea y el conjunto de operaciones que puede realizarse sobre la tarea.

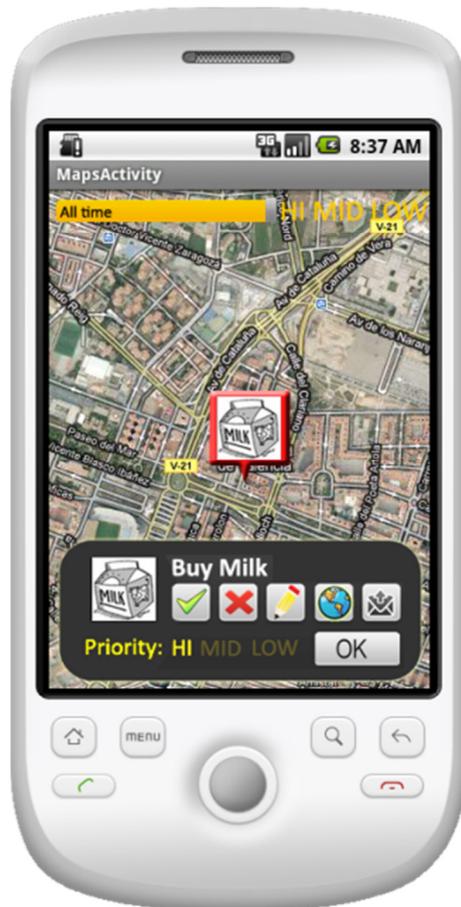


Figura 26. Diseño inicial de la vista Map View

Asimismo, aparte de las tres visualizaciones existen varios aspectos de interacción que comunes a varias vistas. Estos aspectos son: la introducción de componentes de interacción que permiten filtrar las tareas a visualizar y mecanismos para cambiar de una visualización a otra.

Mecanismos para filtrar el conjunto de tareas a visualizar

Los mecanismos para filtrar tareas permiten que el usuario conserve en la visualización únicamente aquellas tareas que tienen una determinada prioridad o que se ubican en un determinado rango de fecha. En particular, para proporcionar esta funcionalidad se proveen los filtros que podemos ver en la Figura 27. El primero de ellos es un filtro temporal que filtra las tareas a desarrollar en los siguientes plazos de tiempo: “*hoy*”, “*próximo 3 días*”, “*1 semana*”, “*1 mes*”, “*más de 1 mes*” y “*en algún momento*”. En cambio, el segundo de ellos es un filtro de prioridad que determina si las tareas de prioridad “*alta*”, “*media*” y “*baja*” deben de ser mostradas. De esta forma, las posibles combinaciones de

estos dos filtros cubren filtros más completos que pueden permitir visualizar únicamente las tareas de prioridad alta de los próximos 3 días.



Figura 27. Filtro temporal y de prioridad.

Mecanismos para cambiar de visualización

Acorde a la necesidad del usuario de poder visualizar sus tareas en función de la situación en la que se encuentre, el diseñador propone la introducción de un menú que permita cambiar de una visualización a otra. Asimismo, otro mecanismo para cambiar a la visualización “*Map View*” desde cualquiera de las otras dos visualizaciones, es la utilización de la operación “*View location*”, la cual está accesible desde un botón como una operación más. Igualmente, con la finalidad de no confundir al usuario en los cambios de visualización, se propone conservar por una parte (1) el estado de los filtros antes del cambio de la visualización, y por otra parte, (2) se propone mantener la selección de una tarea de una visualización origen a la visualización destino.

5.3 Detección de Complejidades

Tras recibir los diseños propuestos por el diseñador, el rol Simplificador detectó un conjunto de complejidades a través de las guías de simplicidad y del conjunto de patrones de diseño de aplicaciones móviles sensibles al contexto. A continuación se describen una a una las complejidades detectadas. Asimismo, para cada una de ellas se realiza una descripción del problema de complejidad. Además, se especifican también los recursos utilizados por el Simplificador para detectar dichos problemas, y las propuestas de cambios a realizar en los diseños.

5.3.1 Componente de Interacción Sobrecargado

Problema:

El componente gráfico mostrado en las vistas “*Map View*” y “*Cloud View*”, el cual puede verse en la Figura 28, sufre la sobrecarga de elementos gráficos. Como podemos ver, este

componente agrupa diferentes elementos: imagen identificativa de la tarea, nombre, un botón para cada operación soportada, etiquetas que permiten modificar la prioridad de la tarea y finalmente un botón de confirmación. En definitiva, este componente agrupa un elevado número de elementos que compiten entre sí por un pequeño espacio dificultando la interacción.

Recursos con los que el problema es detectado:

- Guía nº1: quitar información redundante o irrelevante.
- Guía nº2: seleccionar lo esencial.
- Guía nº4: organizar lo secundario.

Modificaciones propuestas:

- Eliminar la operación “*ver localización*” por encontrarse accesible desde una opción del menú (Guía nº 1).
- Eliminar las etiquetas de edición de la prioridad, ya que es una funcionalidad que se entiende debe estar incluida dentro de “*editar tarea*” (Guía nº 1).
- Dar importancia a la operación principal de la aplicación: “*completar tarea*” (Guía nº 2). En los diseños iniciales, esta operación parece tener la misma importancia que cualquier otra ya que se agrupa junto al resto de operaciones en una colección de botones que sobrecarga la interfaz. En particular, la propuesta concreta que realiza el Simplificador a este problema consiste en (1) conservar únicamente la operación “*completar tarea*” en el componente mostrado al seleccionar una tarea, y (2) facilitar la realización de dicha operación a través de una casilla de verificación situada junto al conjunto de elementos relevantes: icono, el nombre de la tarea y botón que enlaza con una interfaz secundaria donde se organiza y estructura (Guía nº4) el resto de operaciones menos importantes (“*editar tarea*”, “*eliminar tarea*” y “*enviar tarea a contacto*”).

Resultado:



Figura 28. Simplificaciones realizadas en un componente gráfico

5.3.2 Refinamiento del Componente de Interacción Simplificado

Problema:

El componente gráfico mostrado en las vistas “*Cloud View*” y “*Map View*” todavía incluye un elemento innecesario.

La funcionalidad del botón que sirve de enlace a otra interfaz de usuario donde se muestra una vista detallada de una tarea puede ser cubierta por uno de los elementos ya existentes: el nombre de la tarea. De esta forma, se evita la introducción de un elemento innecesario.

Recursos con los que el problema es detectado:

- Guía nº1: quitar información redundante o irrelevante.
- Guía nº5: mejorar la intuición.

Modificaciones propuestas:

- Eliminar el botón de enlace utilizado para navegar a una interfaz secundaria (Guía nº1).
- Introducir en su lugar un hipervínculo en el nombre de la tarea que lo sustituya aprovechando el conocimiento popular del funcionamiento de los hipervínculos (Guía nº5).

Resultado:



Figura 29. Simplificación para proveer una navegación más intuitiva

5.3.3 Información no Relevante en la Vista Timeline View

Problema:

La vista “*Timeline View*” incluye información no relevante en la vista (Ver Figura 30).

Dicha visualización muestra el coste de realizar una actividad y el coste del desplazamiento hasta la localización indicada, expresando ambos costes tanto en dinero como en tiempo. En muchos casos, la información mostrada en estos componentes gráficos es irrelevante o redundante. Por ejemplo, es redundante que realizar tareas como bañar el perro, limpiar la casa o escribir una carta no cueste dinero. De la misma forma, también la colocación de las etiquetas “see priorities” es prescindible al igual que la visualización de las etiquetas “costs” y “transport”, situadas en la izquierda de la interfaz. Esto se debe a que el usuario intuye que un coste situado al lado de un icono de un medio de transporte se asocia a dicho transporte.



Figura 30. Información no relevante o redundante de la vista Timeline View

Recursos con los que el problema es detectado:

- Guía nº1: quitar información redundante o irrelevante.

Modificaciones propuestas:

- Eliminar información no relevante. En concreto, los costes de aquellas que no implican un coste al usuario (Guía nº1).
- Eliminar las etiquetas “costs”, “transport”, “see priorities” (Guía nº1).

- Eliminar, al igual que se propuso en las vistas “*Cloud View*” y “*Map View*”, la operación “*ver localización*”, ya que es accesible desde una opción del menú (Guía nº1).

Resultado:



Figura 31. Vista Timeline View tras eliminar información no relevante o redundante

5.3.4 Sobrecarga de Componentes en la Visualización Timeline View

Problema:

Sobrecarga de componentes, de información y operaciones en la visualización “*Timeline View*”. Pese a que en este caso no existe ningún componente que aparece al interactuar con una tarea en concreto, la misma colección de botones referenciada anteriormente aparece en esta ocasión en la parte superior izquierda de esta visualización. Asimismo, a la operación “*completar tarea*” se le otorga la misma importancia que al resto de operaciones menos relevantes.

Recursos con los que el problema es detectado:

- Guía nº2: seleccionar lo esencial.
- Guía nº4: organizar lo secundario.
- Guía nº5: mejorar la intuición.

Modificaciones propuestas:

- Conservar la operación “*completar tarea*” en la interfaz principal de “*Timeline View*” (Guía nº2).

- Dar soporte a su realización a través de un gesto sobre el icono representativo de la imagen tal como vemos en la Figura 32. Así, el Simplificador propone que la realización de un movimiento en dirección superior sobre el icono representativo de una tarea sirva para que ésta pase a completarse. Además, como continuación de la metáfora, propone que un movimiento en dirección inferior sirva para eliminar una tarea (Guía nº5).
- Por otra parte, como precaución propone la introducción de un mecanismo que permita corregir un error del usuario, el cual puede completar o eliminar una tarea sin desearlo. De esta forma, se propone que a la hora de completar una tarea aparezca un dialogo que puede ser cancelado durante 5 segundos, el cual informa de que la tarea va a ser completada. Asimismo, ante la operación contraria, propone el lanzamiento de un dialogo que requiere confirmación del usuario.
- Además, para dar soporte al resto de operaciones secundarias, es decir, aquellas cuyos botones han sido eliminados de la parte superior de la interfaz de usuario, el Simplificador propone su agrupamiento y reubicación en una interfaz secundaria a la cual se proporciona acceso mediante la interacción con la imagen representativa de la tarea (Guía nº4).

Resultado:



Figura 32. Introducción de gestos en la interacción de la vista Timeline View

5.3.5 Falta de Intuición en la Visualización Timeline View

Problema:

Los mecanismos con los que se transmite al usuario la ordenación de tareas a lo largo del tiempo (ej. flechas conectoras y barra de desplazamiento horizontal) no son intuitivos.

Recursos con los que el problema es detectado:

- Guía nº1: quitar información redundante o irrelevante.
- Guía nº5: mejorar la intuición.
- Patrón “lista ordenada de instancias”.

Modificaciones propuestas:

- Eliminar las flechas que conectan los iconos representativos de cada imagen y la barra de desplazamiento horizontal que permite al usuario recorrer todas las tareas (Guía nº1).
- Siguiendo la metáfora del patrón “*lista ordenada de instancias*”, aprovechar una interacción táctil que libere la interfaz de componentes innecesarios como la barra de desplazamiento o las flechas conectoras entre tareas. De esta forma, la ordenación de instancias se consigue a través de la propia relación de orden existente en las tareas mostradas, las cuales pueden recorrerse a través de gestos hacia la izquierda o derecha, que instintivamente permiten que el usuario recorrer hacia adelante o hacia atrás la línea del tiempo (Guía nº5, patrón de diseño “*lista ordenada de instancias*”).

Resultado:



Figura 33. Eliminación de flechas conectoras y barra de desplazamiento horizontal

5.3.6 Falta de Contexto en la Visualización Timeline View

Problema:

La visualización Timeline View, la cual da especial importancia al atributo de contexto del tiempo para organizar y visualizar la información, no aprovecha al máximo el atributo de contexto temporal para informar al usuario. En particular, la información recibida por el usuario final se limita a una lista ordenada de tareas que no refleja ni la fecha ni el instante de tiempo asociado a una tarea. Así, este atributo de contexto no es aprovechado suficientemente.

Recursos con los que el problema es detectado:

- Guía nº3: complementar y contextualizar lo esencial.
- Patrón “Lista ordenada de ítems”.

Modificaciones propuestas:

- Incluir el filtro de tiempo en la parte superior izquierda de la interfaz, proporcionando así mejor consistencia con las otras vistas y mejorando la utilización del tiempo dentro de esta vista (Guía nº3).
- Introducir el instante de tiempo o fecha asociada a cada tarea en la interfaz. Tal como indica el patrón de diseño “*listado ordenado de instancias*” (referenciar), es necesario que cada instancia muestre su instante de tiempo asociado. Así, la modificación propuesta es introducción de dicho atributo. Además, y profundizando en cómo mostrar este atributo se propone adecuar el atributo a mostrar acorde a su lejanía en el tiempo. De esta forma, se profundiza en la preferencia del usuario de visualizar la hora de aquellas tareas planificadas para el día de hoy, y la de visualizar la fecha de aquellas tareas que deben realizarse en días sucesivos (Guía nº3, patrón de diseño “*listado ordenado de instancias*”).
- Centrar la vista en la siguiente tarea planificada en el tiempo. (Guía nº3, patrón de diseño “*listado ordenado de instancias*”).

Resultado:



Figura 34. Cambios realizados en Timeline View para aprovechar mejor el uso del tiempo

5.3.7 Falta de Contexto en la Visualización Map View

Problema:

La vista Map View no se apoya suficientemente en el atributo de contexto que trata de potenciar: la localización. Teniendo en cuenta que la misión de esta visualización es mostrar de forma localizada y sobre un mapa las tareas del usuario, un valor añadido para el usuario es poder ver de forma clara donde se encuentra ubicado en el instante actual respecto a las localizaciones que se muestran sobre el mapa. Así lo indican todos los patrones de diseño que representan puntos sobre un mapa.

Así, dentro del caso específico de esta aplicación, esta información permitirá decidir al usuario la ruta más óptima a sus intereses para completar las tareas pendientes.

Recursos con los que el problema es detectado:

- Patrones “*mapa de puntos de un único tipo de instancia*”, “*mapa de puntos de un único tipo de instancia con diferenciación de ítems*”, “*mapa de puntos de un único tipo de instancia con personalización de ítems*”, “*mapa de puntos de varios tipos de instancias*”, y “*mapa de puntos con agrupación de ítems*”.
- Guía nº3: complementar y contextualizar lo esencial.

Modificaciones propuestas:

- Incluir la localización actual del usuario sobre el mapa mediante un icono o marcador que lo diferencie de las tareas (Guía nº3, conjunto de patrones de diseño que representan puntos sobre un mapa).

Resultado:



Figura 35. Introducción de localización del usuario en la vista Map View

5.3.8 Solapamiento de Puntos en la Visualización Map View

Problema:

Solapamiento de iconos de tareas pendientes que tienen asociada una misma localización o localizaciones próximas entre sí. A determinados niveles de “zoom” el solapamiento de iconos asociados a localizaciones cercanas dificulta la interacción con una tarea concreta haciendo más difícil su diferenciación respecto al resto de puntos.

Recursos con los que el problema es detectado:

- Patrón “*mapa de puntos con agrupación de ítems*”.
- Guía nº3: complementar y contextualizar lo esencial.
- Guía nº5: mejorar la intuición.

Modificaciones propuestas:

- Representar el conjunto de puntos a visualizar siguiendo el patrón “mapa de puntos con agrupación de ítems”. De esta forma, se propone utilizar el nivel de zoom que el usuario tiene seleccionado para mostrar el conjunto de tareas agrupado de una u otra forma. En particular, si un usuario final tiene que realizar cuatro tareas en la universidad, otra en un museo de su ciudad, y dos más en su casa, el usuario tendrá en total 7 tareas en su ciudad. Así, si dicho usuario enfoca su vista sobre su ciudad, la vista proporcionaría únicamente 3 puntos, uno para cada localización en la que se agrupan todas las tareas a desarrollar en esa localización. En cambio, si el usuario centra su vista en un nivel de zoom más general, la vista podría agrupar los 3 marcadores de la ciudad del usuario en uno solo que represente el número total de tareas a desarrollar en la ciudad. Así, podemos contextualizar la vista en función del nivel de zoom tal como vemos en la Figura 36 mejorando la visualización de tareas y la interacción del usuario con ellas (Guía nº3, patrón “*mapa de puntos con agrupación de ítems*”).

Resultado:



Figura 36. Ejemplo de agrupamiento de iconos sobre un mapa

5.3.9 Reducción de Interacciones Innecesarias

Problema:

Espera elevada para completar una tarea e interacción innecesaria para eliminar una tarea. Con la finalidad de ahorrar esperas e interacciones innecesarias, el Simplificador se percata de que los gestos empleados para completar y eliminar una tarea en la vista “*Timeline View*” pueden ser aún más intuitivos y proporcionando una interacción más simple.

Recursos con los que el problema es detectado:

- Guía nº5: mejora de la intuición.
- Guía nº6: ahorrar tiempo.

Modificaciones propuestas:

- Eliminar el dialogo de confirmación lanzado cuando el usuario desea eliminar una tarea en la vista “*Timeline View*” (Guías nº5 y nº6).
- Eliminar el dialogo de espera a la hora de completar una tarea. (Guías nº5 y nº6).
- Introducción de nuevos gestos. En concreto, el Simplificador propone que al realizar un gesto en dirección superior, el icono representativo de la tarea sea modificado por una etiqueta o componente grafico que indique que la tarea pasa a un estado “completado”. Sin embargo, para poder correr un posible error del usuario, durante 5 segundos, el usuario puede realizar el gesto contrario para que la tarea vuelva a estar en estado pendiente. De la misma forma, si el usuario hace un movimiento hacia abajo sobre el icono de una tarea, este icono se modifica por un componente gráfico que indica que la tarea va a ser eliminada, permitiendo al usuario deshacer dicha operación mediante un gesto hacia arriba sobre este componente.

Resultado:



Figura 37. Mecanismo de interacción para completar una tarea en Timeline View

5.3.10 Falta de Contexto en el Cambio de Visualización

Problema:

El cambio de visualización que permite al usuario adecuar la vista a utilizar en base a sus necesidades no está automatizado.

Recursos con los que el problema es detectado:

- Guía nº3: complementar y contextualizar lo esencial.
- Guía nº7: generar emociones.

Modificaciones propuestas:

- Proporcionar un cambio automático de visualización que permanezca activo solo cuando el usuario no esté interactuando con la aplicación. Este mecanismo automático se lanzará cuando el usuario se aproxime a una localización en la que el usuario tiene tareas pendientes y cambiará la visualización de forma automática a la visualización “*Map View*”. De la misma forma, otro cambio tendrá lugar cuando se aproxime el tiempo en el que una tarea está planificada cambiando la visualización de forma automática y lanzando la visualización “*Timeline View*” (Guías nº3 y 7).
- Además, en caso de que el usuario esté interactuando con la aplicación, este mecanismo de cambio automático se propone que lance un aviso al usuario para preguntarle si desea cambiar de visualización (Guías nº3 y 7).

5.4 Conclusiones

El caso de estudio que se ha presentado en este capítulo muestra un ejemplo concreto en el que los diseños de interfaz de una aplicación móvil son modificados y refinados. La comparación entre los diseños iniciales y los diseños finales muestra cómo las intervenciones del Simplificador, así como la aplicación de las guías de simplicidad y el uso de los patrones de diseño definidos han aportado beneficios.

Una de las conclusiones detectadas en este caso de estudio es que el conjunto patrones de diseño utilizado ha sido de utilidad, cubriendo necesidades específicas que se asocian al tipo de aplicación de este caso de estudio. Dos ejemplos son los cambios realizados en las vistas “*Map View*” y “*Timeline View*”, vistas que tras las modificaciones realizadas aprovechan mejor los atributos de contexto en los que se basan: la localización y el tiempo.

Asimismo, otro de los descubrimientos realizados en este caso de estudio es que a medida que avanzan las iteraciones de simplificación de interfaces, el diseñador entiende mejor los objetivos del usuario y centra cada vez más sus soluciones en ellos. Además, pese a que inicialmente su actitud hacia los cambios fue negativa, a medida que se producían modificaciones en los diseños, se produjo un cambio de actitud en el diseñador, el cual dio lugar a propuestas de mejora realizadas por él mismo en las últimas iteraciones.

Otro de los puntos sobre los que es necesario reflexionar es el coste de la propuesta. Por una parte, los diseños proporcionados al Simplificador han requerido un máximo de 3 intervenciones. Además, respecto a la inversión de tiempo realizada, el Simplificador ha dedicado aproximadamente 15-20 minutos por iteración, un tiempo menor al que podría requerir una evaluación junto al usuario final de la aplicación. Sin embargo, pensamos que la aplicación de la propuesta en equipos de trabajo reales puede no ser tan positiva. Esto se debe a posibles enfrentamientos con diseñadores que son especialmente reacios a los cambios.

Para concluir, pese a las mejoras realizadas creemos conveniente la evaluación posterior de un prototipo funcional de la aplicación junto al usuario. Esto se debe a que hemos descubierto que el Simplificador reflexiona en varias ocasiones sobre cómo el usuario hará uso de la aplicación, basando algunas de sus propuestas en hipótesis que deben comprobarse.

Capítulo 6

Conclusiones

El trabajo que se presenta en esta tesis de master supone el principio de una línea de investigación cuyo objetivo general es facilitar la creación de diseños de interfaz de aplicaciones móviles. En este trabajo se ha realizado una aproximación a la definición de diversos factores que se relacionan con el concepto de la Simplicidad y se ha introducido un rol cuya misión es hacer cumplir estas guías evitando las parcialidades de los diseñadores en la validación de diseños. Además, y para acotar la propuesta a un ámbito concreto se ha definido un conjunto de patrones de diseño de aplicaciones móviles sensibles al contexto. Dado la reciente creación de esta línea de investigación y los comentarios recibido por la comunidad, este trabajo tiene la posibilidad de ser extendido y ampliado en futuros trabajos.

El resto del capítulo se organiza de la siguiente manera. La sección 6.1 sintetiza las contribuciones de este trabajo. La sección 6.2 cita las publicaciones a las que este trabajo ha dado lugar. Finalmente, la sección 6.3 indica varios posibles trabajos con los que esta línea de investigación puede continuar.

6.1 Contribuciones

Las contribuciones de este trabajo tratan de facilitar la obtención de diseños de interfaces de usuario de aplicaciones móviles sensibles al contexto. Para ello, en este trabajo se realizan las siguientes aportaciones:

- La aportación principal de este trabajo es la definición de un conjunto de guías que proveen simplicidad en diseños de interfaz de usuario de aplicaciones móviles.
- La segunda aportación realizada es un conjunto de patrones de diseño de aplicaciones móviles sensibles al contexto.
- La tercera y última aportación de este trabajo consiste en la introducción de un rol Simplificador encargado de detectar complejidades en los diseños a partir de las guías de Simplicidad presentadas y el conjunto de patrones definido.

6.2 Publicaciones

El trabajo que se ha presentado en esta tesis de master ha dado lugar a dos publicaciones que han sido publicadas por la Editorial Springer (LNCS). En particular, estas publicaciones son:

- Refining interaction designs through simplicity.
Pablo Muñoz, Pau Giner, Vicente Pelechano
 First International Joint Conference on Ambient Intelligence.
 LNCS, 2010, Volume 6439/2010, 31-40, DOI: 10.1007/978-3-642-16917-5_4
- Designing context-aware interactions for task-based applications.
Pablo Muñoz, Pau Giner, Miriam Gil
 International Workshop on Web-enabled Objects.
 LNCS, 2010, Volume 6385, DOI: 10.1007/978-3-642-16985-4

Inicialmente, una primera aproximación del trabajo se publicó en un Workshop ubicado dentro de una conferencia de especial relevancia como es el ICWE.

Posteriormente, la segunda publicación conseguida tuvo lugar dentro de la Conferencia AmI, una conferencia internacional de especial relevancia dentro del área de investigación en la que este trabajo se ubica.

6.3 Trabajos Futuros

La propuesta que ha presentado en este trabajo inicia una línea de investigación que da pie a varios posibles futuros trabajos. Estos trabajos son:

1. **La evaluación de los diseños junto al usuario final de la aplicación.** El objetivo de este futuro trabajo es el de validar que efectivamente los cambios realizados en los diseños de nuestro caso de estudio han mejorado y son más fáciles de utilizar por el usuario final. Como tareas definidas para este trabajo encontramos:
 - La selección de usuarios representativos con los que realizar la evaluación a través de cuestionarios.
 - La selección de un número reducido de usuarios acorde a las conclusiones de varios estudios que dicen que con 4 o 5 usuarios se detectan aproximadamente el 80% de los problemas de usabilidad.
 - Selección de los métodos de evaluación a utilizar (ej. uso de cámaras, cuestionarios al finalizar las pruebas, entrevistas,...) y elección del procedimiento a seguir.

2. **La evaluación de la aplicación finalizada a partir del método de evaluación *data logging*.** Los objetivos de este futuro trabajo son (1) detectar problemas de usabilidad en interfaces y componentes gráficos concretos y (2) detectar patrones de uso mediante la monitorización automática del uso de la aplicación.

3. **La definición de patrones de simplificación.** El objetivo de este futuro trabajo es la detección y definición de patrones de simplificación de interfaces. En concreto, estos patrones de simplificación tienen el objetivo de representar las modificaciones típicas que se realizan en los diseños de interfaz (ej. cambio de un componente de interfaz por otro, la colocación de una nueva opción en el menú, cambiar una funcionalidad determinada a otra interfaz secundaria, etc.). De esta forma, los diseñadores podrán disponer de estos patrones para modificar de una manera más productiva los diseños realizados.

Anexo I

Listado de Aplicaciones Analizadas

1. Goole latitude <http://itunes.apple.com/es/app/google-latitude/id306586497?mt=8>
2. Neer <http://itunes.apple.com/us/app/neer/id408503185?mt=8>
3. Plancast <http://itunes.apple.com/es/app/plancast/id360854454?mt=8>
4. Glympse <http://itunes.apple.com/app/glympse-location-sharing-made/id330316698?mt=8>
5. Groupme <http://itunes.apple.com/us/app/groupme-for-iphone/id392796698?mt=8>
6. LocalMind <http://itunes.apple.com/us/app/localmind/id422776889?mt=8>
7. Geoloqi <http://itunes.apple.com/us/app/geoloqi/id415603875>
8. LiquidSpace <http://itunes.apple.com/us/app/liquidspace/id420584007?mt=8&ls=1>
9. OnTheFly <http://itunes.apple.com/us/app/onthefly/id382818039?mt=8>
10. FourSquare <http://itunes.apple.com/es/app/foursquare/id306934924?mt=8>
11. Shell <http://itunes.apple.com/us/app/shell-station-locator/id410842823?mt=8>
12. Week Calendar <http://itunes.apple.com/us/app/week-calendar/id381059732?mt=8>
13. Gowalla <http://itunes.apple.com/es/app/gowalla/id304510106?mt=8>
14. Urbanspoon <http://itunes.apple.com/es/app/urbanspoon/id284708449?mt=8>
15. Facebook (Places) <http://itunes.apple.com/us/app/facebook/id284882215?mt=8>
16. Home Away <http://itunes.apple.com/mx/app/homeaway/id416091677?mt=8>
17. HotelPal <http://itunes.apple.com/us/app/id324473924?mt=8>
18. Buscar mi iPhone <http://www.apple.com/es/iphone/features/find-my-iphone.html>
19. Life 360 <http://itunes.apple.com/us/app/life360-family-tracker/id384830320?mt=8>
20. GasAll <http://itunes.apple.com/es/app/gasall/id296739784?mt=8>
21. All bikes now <http://itunes.apple.com/es/app/allbikesnow/id333176106?mt=8>
22. iPlanner <http://itunes.apple.com/es/app/iplanner-espana/id327100690?mt=8>
23. Trenes <http://itunes.apple.com/es/app/trenes/id312684710?mt=8>
24. CrunchMap <http://itunes.apple.com/my/app/crunch-map/id360089166?mt=8>
25. Google Calendar Client <http://itunes.apple.com/us/app/calendars-google-calendar/id371434886?mt=8>
26. Hoteles Via Michelin <http://itunes.apple.com/es/app/hoteles-viamichelin/id389732950?mt=8>
27. Páginas Amarillas <http://itunes.apple.com/es/app/paginasamarillas-es-cerca/id303686830?mt=8>
28. Wifi finder <http://itunes.apple.com/es/app/wi-fi-finder/id300708497?mt=8>
29. Zillow Real State <http://itunes.apple.com/es/app/zillow-real-estate-search/id310738695?mt=8>
30. Booking <http://itunes.apple.com/us/app/booking-com-hotel-reservations/id367003839?mt=8>
31. Trip Journal <http://itunes.apple.com/us/app/trip-journal/id341585937?mt=8>
32. Nike+ GPS <http://itunes.apple.com/es/app/nike-gps/id387771637?mt=8>
33. MotionX GPS <http://itunes.apple.com/es/app/motionx-gps/id299949744?mt=8>
34. MyTracks <http://itunes.apple.com/gb/app/mytracks-the-gps-logger/id358697908?mt=8>
35. Birthday Remember <http://itunes.apple.com/es/app/birthday-reminder/id312570207?mt=8>
36. Yelp <http://itunes.apple.com/es/app/yelp/id284910350?mt=8>
37. Around Me <http://itunes.apple.com/es/app/aroundme/id290051590?mt=8>
38. Google Places <http://itunes.apple.com/es/app/google-places/id406513617?mt=8>
39. Loopt <http://itunes.apple.com/us/app/loopt/id281952554?mt=8>
40. SCVNGR <http://itunes.apple.com/us/app/scvngr/id323248984?mt=8>

Análisis del Uso que Realizan de la Localización y Tiempo

		Código identificador de la aplicación																				
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
Características	Captura del instante de tiempo actual.	SI	SI	SI	SI	NO	NO	SI	SI	NO	SI	SI	NO	SI	SI	NO	SI	NO	NO	SI	SI	NO
	Captura de la localización actual.	SI	SI	NO	SI	SI	NO	SI	SI	NO	SI	SI	NO	SI	SI	NO	SI	SI	SI	SI	SI	SI
	Representación de información teniendo en cuenta el instante actual.	NO	SI	NO	SI	NO	NO	SI	SI	NO	SI	SI	NO	SI	SI	NO	SI	NO	NO	SI	NO	NO
	Representación de información teniendo en cuenta la localización actual.	SI	NO	NO	SI	NO	SI	NO	SI	SI	SI	SI	SI									
	Activación automática en un instante de tiempo determinado.	NO	NO	SI	NO																	
	Activación automática en una localización determinada.	NO	SI	NO	NO	NO	SI	NO														
	Seguimiento del tiempo dentro de la aplicación.	NO	NO	NO	SI	NO	NO	SI	NO													
	Seguimiento de la localización dentro de la aplicación.	SI	SI	NO	SI	NO	NO	SI	NO	SI	SI	NO	SI	SI	NO	SI	NO	NO	NO	NO	SI	SI
	Etiquetación de un elemento con el tiempo actual.	SI	SI	NO	SI	NO	NO	SI	NO													
	Etiquetación de un elemento con la localización actual.	NO	NO	NO	NO	NO	NO	NO	NO	SI	NO											
Publicación de la localización actual.	SI	SI	NO	SI	SI	NO	SI	NO	SI	NO	NO	SI	NO	NO	SI	NO	SI	NO	SI	NO	SI	
Publicación del instante de tiempo actual.	SI	SI	NO	SI	SI	NO	NO	NO	SI	NO	NO	SI	NO	NO	SI	NO	SI	NO	SI	NO	SI	
Necesidad de introducción manual de un instante de tiempo.	NO	NO	SI	NO	NO	SI	NO	SI	NO	NO	SI	NO	NO	SI	NO	NO	NO	NO	SI	NO	NO	
Necesidad de introducción manual de una localización.	NO	NO	SI	NO	NO	SI	NO	SI	NO													

		Código identificador de la aplicación																			
		21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
Características	Captura del instante de tiempo actual.	NO	NO	SI	NO	SI	NO	NO	NO	NO	SI	SI	NO	SI	SI	NO	SI	NO	NO	SI	NO
	Captura de la localización actual.	SI	NO	NO	NO	NO	SI	NO	SI	SI	NO	SI									
	Representación de información teniendo en cuenta el instante actual.	NO	SI	NO	SI	NO	SI	NO	SI	SI	NO	NO	SI								
	Representación de información teniendo en cuenta la localización actual.	SI	NO	NO	SI	NO	SI	NO	SI	SI	SI	SI									
	Activación automática en un instante de tiempo determinado.	NO	SI	NO																	
	Activación automática en una localización determinada.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
	Seguimiento del tiempo dentro de la aplicación.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI	SI	SI	NO	NO	NO	NO	NO	NO
	Seguimiento de la localización dentro de la aplicación.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI								
	Etiquetación de un elemento con el tiempo actual.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
	Etiquetación de un elemento con la localización actual.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO
Publicación de la localización actual.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI									
Publicación del instante de tiempo actual.	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	NO	
Necesidad de introducción manual de un instante de tiempo.	NO	SI	NO	NO	SI	NO	SI	NO	NO	NO	NO	NO	NO								
Necesidad de introducción manual de una localización.	NO	NO	NO	NO	SI	NO															

Referencias

- [1] Alsumait, A., & Al-Osaimi, A. (2009). Usability heuristics evaluation for child e-learning applications. (págs. 425-430). ACM.
- [2] Ayob, N., Hussin, A., & Dahlan, H. (2009). Three Layers Design Guideline for Mobile Application., (págs. 427-431).
- [3] Balagtas-Fernandez, F., & Hussmann, H. (2009). A Methodology and Framework to Simplify Usability Analysis of Mobile Applications. (págs. 520-524). IEEE Computer Society.
- [4] Balagtas-Fernandez, F., Forrai, J., & Hussmann, H. (2009). Evaluation of User Interface Design and Input Methods for Applications on Mobile Touch Screen Devices. En T. Gross, J. Gulliksen, P. Kotz, L. Oestreicher, P. Palanque, R. Prates, y otros (Edits.), *Human-Computer Interaction INTERACT 2009* (Vol. 5726, págs. 243-246). Springer Berlin / Heidelberg.
- [5] Ballagas, R., Borchers, J., Rohs, M., & Sheridan, J. (2006). The smart phone: a ubiquitous input device. *Pervasive Computing, IEEE*, 5(1), 70-77.
- [6] Ballard, B. (2007). *Designing the Mobile User Experience*. John Wiley & Sons.
- [7] Bell, G., & Dourish, P. (2007). Yesterday's tomorrows: notes on ubiquitous computing's dominant vision. *Personal Ubiquitous Comput.*, 11, 133-143.
- [8] Bellotti, V., & Edwards, K. (2001). Intelligibility and accountability: human considerations in context-aware systems. *Hum.-Comput. Interact.*, 16, 193-212.
- [9] Cartman, J., & Ting, R. (2008). *Strategic Mobile Design: Creating Engaging Experiences*. New Riders Publishing.
- [10] Cohen, M. H., Giangola, J. P., & Balogh, J. (2004). *Voice User Interface Design*. Addison Wesley Longman Publishing Co., Inc.
- [11] Cooper, A. (2004). *The Inmates Are Running the Asylum: Why High Tech Products Drive Us Crazy and How to Restore the Sanity* (2nd Edition). Pearson Higher Education.

- [12] Dey, A. K., & Abowd, G. D. (2000). Towards a better understanding of context and context-awareness.
- [13] Dotan, A., Maiden, N., Lichtner, V., & Germanovich, L. (2009). Designing with Only Four People in Mind? --- A Case Study of Using Personas to Redesign a Work-Integrated Learning Support System. (págs. 497-509). Springer-Verlag.
- [14] Duh, H. B.-L., B., G. C., & Chen, V. H.-h. (2006). Usability evaluation for mobile device: a comparison of laboratory and field tests. (págs. 181-186). ACM.
- [15] Dunlop, M., & Brewster, S. (2002). The Challenge of Mobile Devices for Human Computer Interaction. *Personal Ubiquitous Comput.*, 6, 235-236.
- [16] Eagle, N., & Pentland, A. (2005). Social Serendipity: Mobilizing Social Software. *IEEE Pervasive Computing*, 4, 28-34.
- [17] Fahrmaier, M., Sitou, W., & Spanfelner, B. (2006). Unwanted Behavior and its Impact on Adaptive Systems in Ubiquitous Computing.
- [18] Fernandes, G., & Holmes, C. (2002). Applying HCI to music-related hardware. (págs. 870-871). ACM.
- [19] Fiotakis, G., Raptis, D., & Avouris, N. (2009). Considering Cost in Usability Evaluation of Mobile Applications: Who, Where and When. En T. Gross, J. Gulliksen, P. Kotz, L. Oestreicher, P. Palanque, R. Prates, y otros (Edits.), *Human-Computer Interaction " INTERACT 2009* (Vol. 5726, págs. 231-234). Springer Berlin / Heidelberg.
- [20] Glissmann, S., Smolnik, S., Schierholz, R., Kolbe, L., & Brenner, W. (2005). Proposition of an M-Business Procedure Model for the Development of Mobile User Interfaces. (págs. 308-314). IEEE Computer Society.
- [21] Gong, J., & Tarasewich, P. (2004). Guidelines for handheld mobile device interface design.
- [22] Goodman, J., Langdon, P. M., & Clarkson, P. J. (2006). Providing Strategic User Information for Designers: Methods and Initial Findings. En J. Clarkson, P. Langdon, & P. Robinson (Edits.), *Designing Accessible Technology* (págs. 41-51). Springer London.

- [23] Jambon, F., Golanski, C., & Pommier, P.-J. (2007). Meta-Evaluation of a Context-Aware Mobile Device Usability., (págs. 21-26).
- [24] Jensen, K. L., & Larsen, L. B. (2007). Evaluating the usefulness of mobile services based on captured usage data from longitudinal field trials. (págs. 675-682). ACM.
- [25] Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K., & Buchanan, G. (1999). Improving Web interaction on small displays. *Comput. Netw.*, 31, 1129-1137.
- [26] Kallio, T., & Kaikkonen, A. (2005). {Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing}. *Journal of Usability Studies*, 1, 4-16.
- [27] Kanstrup, A. M., & Stage, J. (2009). From Paper to PDA: Design and Evaluation of a Clinical Ward Instruction on a Mobile Device. (págs. 670-683). Springer-Verlag.
- [28] Kim, H., Kim, J., Lee, Y., Chae, M., & Choi, Y. (2002). An Empirical Study of the Use Contexts and Usability Problems in Mobile Internet. (págs. 132--). IEEE Computer Society.
- [29] Kim, L., & Albers, M. J. (2001). Web design issues when searching for information in a small screen display. (págs. 193-200). ACM.
- [30] Kjeldskov, J., Skov, M. B., Als, B. S., & HÃegh, R. T. (2004). Is It Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the Field. En S. Brewster, & M. Dunlop (Edits.), *Mobile Human-Computer Interaction* â€“ MobileHCI 2004 (Vol. 3160, págs. 529-535). Springer Berlin / Heidelberg.
- [31] Krug, S. (2006). *Don't make me think! Web Usability - Das intuitive Web* (2 ed.).
- [32] Lee, K. B., & Grice, R. (2004). Developing a new usability testing method for mobile devices., (págs. 115-127).
- [33] Lee, Y. E., & Benbasat, I. (2003). Interface design for mobile commerce. *Commun. ACM*, 46, 48-52.

- [34] LeRouge, C., Ma, J., Sneha, S., & Tolle, K. (2011). User profiles and personas in the design and development of consumer health technologies. *International Journal of Medical Informatics*, In Press, Corrected Proof, - .
- [35] Ling, C., Salvendy, G. (2005). Extension of heuristic evaluation method: a review and reappraisal. *Ergonomia An International Journal of Ergonomics and Human Factors IJEHF*(2005)
- [36] Liu, N., Liu, Y., & Wang, X. (2010). Data logging plus e-diary: towards an online evaluation approach of mobile service field trial. (págs. 287-290). ACM.
- [37] Longoria, R. (2001). *Designing Mobile Applications: Challenges, Methodologies, and Lessons Learned*. (págs. 91-95). Lawrence Erlbaum Associates.
- [38] Maeda, J. (2006). *The Laws of Simplicity (Simplicity: Design, Technology, Business, Life)*. The MIT Press.
- [39] Mao, J.-Y., Vredenburg, K., Smith, P. W., & Carey, T. (2001). User-centered design methods in practice: a survey of the state of the art. (págs. 12--). IBM Press.
- [40] Nielsen, C. M., Overgaard, M., Pedersen, M. B., Stage, J., & Stenild, S. (2006). It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field. (págs. 272-280). ACM.
- [41] Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann Publishers.
- [42] Nielsen, J. (1994). Usability inspection methods. (págs. 413-414). ACM.
- [43] Nieters, J. E., Ivaturi, S., & Ahmed, I. (2007). Making personas memorable. (págs. 1817-1824). ACM.
- [44] Norman, D. A. (2005). *Human-Centered Design Considered Harmful*. {interactions}, {XII}(4).
- [45] Oulasvirta, A., Tamminen, S., Roto, V., & Kuorelahti, J. (2005). Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. (págs. 919-928). ACM.
- [46] Pascoe, J., Ryan, N., & Morse, D. (2000). Using while moving: HCI issues in fieldwork environments. *ACM Trans. Comput.-Hum. Interact.*, 7, 417-437.

- [47] Paternò, F., Russino, A., & Santoro, C. (2007). Remote evaluation of mobile applications. (págs. 155-169). Springer-Verlag.
- [48] Po, S., Howard, S., Vetere, F., & Skov, M. B. (2004). Heuristic Evaluation and Mobile Usability: Bridging the Realism Gap. En S. Brewster, & M. Dunlop (Edits.), *Mobile Human-Computer Interaction MobileHCI 2004* (Vol. 3160, págs. 591-592). Springer Berlin / Heidelberg.
- [49] Pruitt, J., & Adlin, T. (2005). *The Persona Lifecycle: Keeping People in Mind Throughout Product Design* (The Morgan Kaufmann Series in Interactive Technologies). Morgan Kaufmann Publishers Inc.
- [50] Pruitt, J., & Grudin, J. (2003). *Personas: Practice and theory*.
- [51] Ryan, C., & Gonsalves, A. (2005). The effect of context and application type on mobile usability: an empirical study. (págs. 115-124). Australian Computer Society, Inc.
- [52] Sá, M. d., & Carriço, L. (2009). A mobile tool for in-situ prototyping. (págs. 20:1--20:4). ACM.
- [53] Salman, Y.B. et al, 2009, *Proceedings of the 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, Participatory design and evaluation of e-learning system for Korean language training*. pp. 312-9.
- [55] Savio, N., & Braiterman, J. (2007). *Design Sketch: The Context of Mobile Interaction.*, (págs. 284-286).
- [56] Schusteritsch, R., Wei, C. Y., & LaRosa, M. (2007). Towards the perfect infrastructure for usability testing on mobile devices. (págs. 1839-1844). ACM.
- [57] Shneiderman, B., & Plaisant, C. (2004). *Designing the User Interface: Strategies for Effective Human-Computer Interaction* (4th Edition). Pearson Addison Wesley.
- [58] Sim, G., Read, J., & Cockton, G. (2009). Evidence Based Design of Heuristics for Computer Assisted Assessment. En T. Gross, J. Gulliksen, P. Kotz, L. Oestreicher, P. Palanque, R. Prates, y otros (Edits.), *Human-Computer Interaction INTERACT 2009* (Vol. 5726, págs. 204-216). Springer Berlin / Heidelberg.

- [59] Thompson, A.-J., & Kemp, E. A. (2009). Web 2.0: extending the framework for heuristic evaluation. (págs. 29-36). ACM.
- [60] Unger, R., & Chandler, C. (2009). A Project Guide to UX Design: For user experience designers in the field or in the making (1st ed.). New Riders Publishing.
- [61] Vastenburg, M. H., Keyson, D. V., & Ridder, H. (2008). Considerate home notification systems: a field study of acceptability of notifications in the home. *Personal Ubiquitous Comput.*, 12, 555-566.
- [62] Villamor, C., Willis, D., Wroblewski, L., & Rhim, J. (2010). Touch Gesture Reference Guide. Touch Gesture Reference Guide.
- [63] Weiser, M. (1999). The computer for the 21st century. *SIGMOBILE Mob. Comput. Commun. Rev.*, 3, 3-11.
- [64] Wiethoff, A., Conradi, B., & Butz, A. (2010). Artifacts in the design process of pervasive computing. *Artifacts in the design process of pervasive computing*.
- [65] Wright, T., Yoong, P., Noble, J., Cliffe, R., Hoda, R., Gordon, D., y otros. (2005). Usability methods and mobile devices: an evaluation of MoFax. (págs. 26-33). ACM.
- [66] Zhang, D., & Adipat, B. (2005). Challenges, Methodologies, and Issues in the Usability Testing of Mobile Applications. *International Journal of Human-Computer Interaction*, 18(3), 293-308.



UNIVERSIDAD
POLITECNICA
DE VALENCIA



Centro de Investigación en Métodos
de Producción de Software