



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL
UNIDAD ZACATENCO
DEPARTAMENTO DE COMPUTACIÓN

Heurísticas de coherencia para el diseño de meta-interfaces de usuario

Tesis que presenta

Luis Martín Sánchez Adame

Para obtener el grado de

Maestro en Ciencias en Computación

Directora de la Tesis

Dra. Sonia Guadalupe Mendoza Chapa

Ciudad de México, México

Febrero 2016

Resumen

El frenético desarrollo de ambientes interactivos ha supuesto la creación de muchos tipos de herramientas de usuario final para el control y la configuración de los mismos. Actualmente, se están desarrollando modelos y mecanismos para la remodelación y redistribución de interfaces de usuario en los ámbitos de Plasticidad, Adaptabilidad y Sensibilidad al Contexto. Sin embargo, los desarrolladores de sistemas interactivos deben ser cuidadosos al considerar el nivel de control dejado a los usuarios finales, ya que no todas las aplicaciones consideran hasta qué punto podrían volverse inutilizables. En el presente, el mantenimiento de la coherencia de la interfaz de usuario de un sistema en un ambiente interactivo es un problema abierto. Las meta-interfaces de usuario ofrecen el contexto necesario para desarrollar mecanismos que solucionen este problema. La usabilidad y experiencia de uso del sistema son aspectos clave, que han sido soslayados en diversos desarrollos. Ya sea que el sistema o el usuario decida la configuración de los componentes de interacción, es de vital importancia que en todo momento se mantenga la coherencia de la interfaz de usuario. La coherencia no sólo proporciona al usuario un marco estable en contextos similares, sino también es un elemento esencial de aprendizaje y una de las principales palancas para asegurar el uso eficiente de la interfaz de usuario. Por esta razón, el presente trabajo de tesis propone heurísticas de coherencia que ofrece a los desarrolladores, una serie de escenarios y preguntas a considerar para el diseño de meta-interfaces de usuario. Las heurísticas de coherencia propuestas son libres de dependencias de dominio específicas, esto significa que a diferencia de las interfaces de usuario especiales para trabajos y servicios, las meta-interfaces de usuario permiten a los usuarios finales explorar su entorno, identificar las tareas y servicios compatibles con el ambiente, y asignar los recursos disponibles para ejecutar tareas e interactuar con servicios. Aplicando las heurísticas de coherencia a una aplicación prototipo, se permitió a los usuarios finales combinar recursos independientes de una manera significativa y coherente.

Abstract

The frenetic development of interactive environments has led to the creation of many types of end-user tools for controlling and configuring them. Currently, there are models and mechanisms developed for remodeling and redistribution of user interfaces in the areas of plasticity, adaptability and sensitivity to context. However, developers of interactive systems should be careful when considering the level of control left to end users, since not all applications considered to what extent may become unusable. At present, maintaining the coherence of the user interface of a system in an interactive environment is an open issue. The meta-user interfaces provide the necessary context to develop mechanisms to solve this problem. Usability and user experience are key aspects of the system which have been bypassed in various developments. Whether the system or the user decides the configuration of the components of interaction, it is vital that at all times the coherence of the user interface is maintained. Coherence not only provides users with a stable framework in similar contexts, but is also an essential element of learning and one of the main levers to ensure the efficient use of the user interface. For this reason, this thesis proposes a set of heuristics for coherence that gives developers a series of scenarios and questions to consider for the design of meta-user interfaces. The proposed set of heuristics is free of domain-specific dependencies, this means that unlike the special user interfaces for work and services, the meta-user interfaces allow end users to explore their environment, identify the tasks and services that support the environment, and allocate resources to implement tasks and interact with services. Applying the set of heuristics to a prototype application, end users are allowed to combine independent resources in a meaningful and coherent manner.

Índice general

| | | |
|----------|---|-----------|
| 1 | Introducción | 1 |
| 1.1 | Motivación | 1 |
| 1.2 | Contexto de investigación | 4 |
| 1.3 | Planteamiento del problema | 5 |
| 1.4 | Objetivos | 6 |
| 1.5 | Organización del documento | 7 |
| 2 | Estado del arte | 9 |
| 2.1 | Marco teórico | 9 |
| 2.1.1 | Ergonomía | 9 |
| 2.1.2 | Usabilidad | 10 |
| 2.1.3 | Heurísticas de usabilidad para el diseño de interfaces de usuario | 11 |
| 2.1.4 | Coherencia | 17 |
| 2.1.5 | Meta-interfaz de usuario | 20 |
| 2.1.6 | Espacio de dimensiones para meta-interfaces de usuario | 22 |
| 2.2 | Trabajos relacionados | 27 |
| 2.2.1 | CamNote | 28 |
| 2.2.2 | Sitio Web Sedan-Bouillon | 29 |
| 2.2.3 | Photo-Browser | 30 |
| 3 | Desarrollo de las heurísticas de coherencia | 33 |
| 3.1 | Estructura de las heurísticas de coherencia | 33 |
| 3.2 | Honestidad de meta-interfaces | 34 |
| 3.2.1 | Descripción de escenario | 34 |
| 3.2.2 | Dimensiones, elementos y preguntas | 35 |
| 3.3 | Núcleos funcionales | 37 |

| | | |
|----------|---|-----------|
| 3.3.1 | Descripción de escenario | 37 |
| 3.3.2 | Dimensiones, elementos y preguntas | 37 |
| 3.4 | Multimodalidad | 38 |
| 3.4.1 | Descripción de escenario | 39 |
| 3.4.2 | Dimensiones, elementos y preguntas | 39 |
| 3.5 | Limitaciones de usabilidad | 40 |
| 3.5.1 | Descripción de escenario | 40 |
| 3.5.2 | Dimensiones, elementos y preguntas | 41 |
| 3.6 | Trazabilidad | 42 |
| 3.6.1 | Descripción de escenario | 42 |
| 3.6.2 | Dimensiones, elementos y preguntas | 42 |
| 3.7 | Análisis de trabajos relacionados mediante las heurísticas propuestas | 43 |
| 3.7.1 | CamNote | 43 |
| 3.7.2 | Sitio Web Sedan-Bouillon | 46 |
| 3.7.3 | Photo-Browser | 49 |
| 4 | Análisis y diseño de la aplicación de prueba | 53 |
| 4.1 | Análisis | 53 |
| 4.2 | Diseño | 55 |
| 4.2.1 | Vista de escritorio | 55 |
| 4.2.2 | Layout | 56 |
| 4.2.3 | Sincronización entre dispositivos | 59 |
| 4.2.4 | Interfaces en dispositivos móviles | 59 |
| 4.2.5 | Presencia | 61 |
| 4.2.6 | Aplicación de las heurísticas de coherencia | 62 |
| 5 | Implementación de la aplicación de prueba | 67 |
| 5.1 | Tecnologías usadas | 67 |
| 5.1.1 | HTML5 y CSS3 | 67 |
| 5.1.2 | JavaScript | 69 |
| 5.1.3 | Firebase | 69 |
| 5.2 | Arquitectura de la aplicación | 69 |
| 5.2.1 | Vista de escritorio | 70 |
| 5.2.2 | Meta-menú | 75 |
| 5.2.3 | Herramientas móviles | 79 |

| | | |
|----------|--|-----------|
| 5.3 | Pruebas con usuarios finales | 83 |
| 6 | Conclusiones y trabajo a futuro | 87 |
| 6.1 | Recapitulación del problema | 87 |
| 6.2 | Conclusiones | 87 |
| 6.3 | Limitaciones | 90 |
| 6.4 | Trabajo a futuro | 91 |
| 7 | Bibliografía | 93 |

Capítulo 1

Introducción

1.1 Motivación

La capacidad de los usuarios para controlar y evaluar el estado de un sistema es fundamental para la Interacción Humano-Computadora (HCI - *Human Computer-Interaction*) [Norman y Draper, 1986]. Este principio, promovido en la década de 1980 por psicólogos cognitivos y especialistas en factores humanos, se había aplicado veinte años antes por científicos de la Computación, quienes introdujeron el concepto de lenguaje de control de trabajos (JCL - *Job Control Language*). JCL fue utilizado para el procesamiento por lotes, que con el paso del tiempo fue reemplazado por el *shell* de Unix y en la actualidad por escritorios gráficos en los sistemas operativos. Con la aparición de la Computación Ubicua, los usuarios no se limitan a un solo sistema operativo ni a las aplicaciones de una sola computadora. En su lugar, la Computación Ubicua abarca un modelo en el que los usuarios, servicios y recursos descubren otros usuarios, servicios y recursos para integrarlos en un ambiente interactivo [Coutaz et al., 2005].

Un ambiente interactivo [Garlan et al., 2002] [Coutaz, 2006] es un conjunto dinámico de entidades físicas, computacionales y de comunicación para apoyar las actividades humanas. Un ambiente interactivo puede ser tan simple como una estación de trabajo o un PDA conectado a Internet, o tan complejo como un ecosistema computacional que evoluciona para adaptarse al contexto de uso.

El concepto de contexto de uso [Sottet et al., 2009] [Douglas y Liu, 2011] se define como las condiciones reales en las que se utiliza o se utilizará un sistema determinado, bajo condiciones de trabajo cotidiano.

Según el enfoque de la Computación Ubicua, el contexto de uso varía desde aplicaciones confinadas a una sola estación de trabajo hasta un espacio dinámico e interactivo, donde los límites entre el mundo físico y el mundo digital desaparecen progresivamente [Fishkin, 2004]. Como resultado, las soluciones bien conocidas, proporcionadas por *shells* y escritorios, no son suficientes [Bellotti et al., 2002]. Muchas técnicas de interacción se están desarrollando para la Computación Ubicua, aunque en una base de caso por caso. Este enfoque *ad-hoc* es adecuado para la exploración local, pero podría no proporcionar información suficiente al problema global.

Un cambio de contexto puede definirse como la modificación de un valor de cualquier elemento en los espacios de información contextual de un sistema interactivo [Crowley et al., 2005] [Coutaz y Calvary, 2012]. De acuerdo a lo que se puede observar, los espacios de información se clasifican de tres maneras:

- Modelo de **usuario**: denota los atributos y funciones que describen la persona arquetípica que se pretende pueda utilizar, o en realidad está utilizando, el sistema interactivo. Esta información incluye desde las preferencias básicas del usuario de lo dispuesto por la mayoría de los sistemas interactivos, hasta descripciones más sofisticadas, tales como idiosincrasias y actividades, perfiles inferidos por el uso repetitivo de servicios, comandos y secuencias de tareas actuales.
- Modelo de **ambiente**: incluye atributos y funciones que caracterizan a los momentos y lugares físicos donde la interacción se llevará a cabo o de hecho ocurre, por ejemplo: lugares georreferenciados (e.g., coordenadas GPS) y/o lugares físicos (e.g., en casa, en un espacio público, en movimiento por la calle, en un tren o en un coche), características temporales (e.g., el 4 de enero) y simbólicas (e.g., otoño o invierno), normas y actividades sociales, así como condiciones físicas perceptibles por humanos tales como la luz, el calor y las condiciones de sonido.
- Modelo de **plataforma**: describe los recursos de computación y de interacción que unen el medio físico con el mundo digital. En el paradigma de interfaz gráfica de usuario convencional, la plataforma se limita a un único dispositivo de computación, típicamente una estación de trabajo o un teléfono inteligente, conectado a una red y equipado con un conjunto fijo de recursos de interacción tales como una pantalla, un teclado o un *stylus*. Los avances tecnológicos están dando lugar a que los individuos tengan la capacidad de ensamblar y moldear sus propios espacios interactivos, a partir de puntos de acceso públicos y dispositivos privados, para acceder a los servicios dentro de la red de la

computación mundial. Los espacios interactivos pronto tomarán la forma de “islas” autónomas de computación o ecosistemas, cuyo horizonte evolucionará, se dividirá y se combinará bajo el control humano. Los recursos se acoplarán de manera oportuna para amplificar las actividades humanas en las que cualquier objeto del mundo real tiene el potencial de desempeñar el papel de un recurso de interacción. Como resultado, la plataforma debe ser modelada como un grupo dinámico de recursos heterogéneos, en lugar de como un dispositivo estático convencional mono-computadora.

Cuando un cambio de contexto ocurre, es necesario tomar las medidas pertinentes para mantener la usabilidad del sistema [Nielsen, 2010]. Se tiene que pensar en nuevas formas de interacción para los ambientes compuestos de diversos dispositivos heterogéneos, sin olvidar los avances de investigación sobre sistemas monolíticos convencionales. Precisamente, las Meta-Interfaces de Usuario (Meta-UI por sus siglas en inglés) se proponen para definir el nivel de integración de diversas plataformas en un ambiente interactivo.

Una meta-interfaz de usuario [Coutaz, 2006] es un sistema interactivo, cuyo conjunto de funciones es necesario y suficiente para controlar y evaluar el estado de un ambiente interactivo. Este conjunto es meta, ya que sirve como un umbral más allá de los servicios dependientes de dominio que apoyan las actividades humanas en este espacio. Es una interfaz orientada a los usuarios, ya que su papel es el de permitir a los mismos controlar, moldear, entender y evaluar el estado del ambiente interactivo.

En la actualidad, se están desarrollando modelos y mecanismos para la remodelación y redistribución de interfaces de usuario en los ámbitos de Plasticidad, Adaptabilidad y Sensibilidad al Contexto [Reed y Monk, 2010]. Las meta-interfaces ofrecen una solución holística para la interacción entre sistema y usuario, priorizando la usabilidad y la continuidad de interacción. Sin embargo, se debe ser muy cuidadoso al considerar el nivel de control dejado a los usuarios finales. Cuando las interfaces de usuario se generan de forma dinámica, hay que prestar especial atención a que el sistema se mantenga coherente y no demerite la experiencia de uso.

1.2 Contexto de investigación

A continuación se describen brevemente los temas en los que se inscribe el presente trabajo de tesis (ver Figura 1.1):

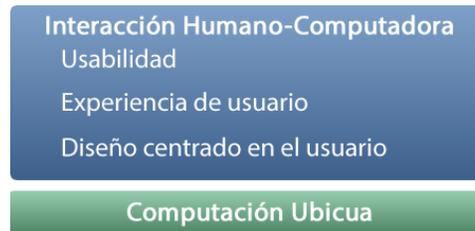


Figura 1.1 Contexto de investigación del presente trabajo de tesis

- **Interacción Humano-Computadora:** es un área de investigación y práctica que surgió en la década de 1980, inicialmente como un área de especialidad en informática que abarca la ciencia cognitiva y la ingeniería de factores humanos. La Interacción Humano-Computadora (HCI por sus siglas en inglés) se ha expandido rápida y constantemente desde hace tres décadas, atrayendo a profesionales de muchas otras disciplinas. La HCI investiga el diseño y uso de la tecnología informática, con especial atención a las interfaces entre las personas (usuarios) y las computadoras. Los investigadores en el campo de la HCI observan las formas en que los seres humanos interactúan con las computadoras y diseñan tecnologías que permiten a los humanos interactuar con las computadoras de formas novedosas [Hippe et al., 2014].
- **Usabilidad:** es un indicador de cuán eficaz, eficiente y satisfactoriamente un usuario puede interactuar con una interfaz de usuario [Usability.gov, 2015].
- **Experiencia de usuario:** es la calidad de la experiencia que una persona tiene al interactuar con un diseño específico. Originalmente el concepto era usado en referencia a la interacción persona-computadora. El término ahora se utiliza para referirse a cualquier interacción específica del diseño humano, que va desde un dispositivo digital, a un proceso de venta o a toda una conferencia [Knemeyer y Svoboda, 2015].
- **Diseño centrado en el usuario:** es un enfoque para el diseño de un producto o servicio (e.g., diseño de la interfaz de usuario), en el que el usuario final se coloca en el centro del proceso. Es un marco de procesos (no restringido a las interfaces o tecnologías) en

el que a las necesidades, deseos y limitaciones de los usuarios finales de un producto, servicio o proceso se les da una gran atención en cada etapa del proceso de diseño . El diseño centrado en el usuario se puede caracterizar como un proceso de resolución de problemas de varias etapas, que requiere de diseñadores no sólo para analizar y prever cómo es probable que los usuarios utilicen un producto, sino también para poner a prueba la validez de sus supuestos, con respecto al comportamiento de los usuarios en el mundo real [Usability.gov, 2015].

- **Computación Ubicua:** es un concepto de Ingeniería de Software e Informática, donde la Computación aparece en cualquier momento y en todas partes. En contraste con la Computación de “escritorio”, la Computación Ubicua se puede presentar utilizando cualquier dispositivo, en cualquier localización y en cualquier formato [Weiser, 1999] [Usability.gov, 2015].

1.3 Planteamiento del problema

En los últimos años se han desarrollado algunas meta-interfaces, sin embargo muy pocas toman en cuenta las condiciones en las que los elementos de la interfaz de usuario se deben remodelar o redistribuir [Coutaz, 2006]. La usabilidad y experiencia de uso del sistema son aspectos clave, pero, en diversos desarrollos, no se les presta especial atención [Demeure et al., 2005] [Coutaz, 2006] [Coutaz, 2010]. De igual forma, el nivel de control dejado al usuario final con respecto a la remodelación y redistribución de los elementos de la interfaz de usuario es una incógnita. Ya sea que el sistema o el usuario decida la configuración de los componentes de interacción, es de vital importancia que en todo momento se mantenga la coherencia de la interfaz de usuario. De lo contrario, se puede llegar a un estado en el que el sistema se vuelve completamente inútil, dando como resultado que el usuario final vea sus necesidades completamente insatisfechas. La coherencia de la interfaz de usuario se refiere al uso constante de los elementos de diseño, como la tipografía, colores, iconos, navegación, imágenes y fondos.

En la actualidad, el mantenimiento de la coherencia de la interfaz de usuario de un sistema en un ambiente interactivo es un problema abierto (ver Figura 1.2). Las meta-interfaces ofrecen el contexto necesario para desarrollar mecanismos que solucionen este problema.

La hipótesis de investigación para el proyecto de tesis es que al añadir elementos de coherencia en el diseño de las meta-interfaces, se mejora su usabilidad y se reducen las probabilidades de un escenario con una experiencia de usuario negativa.

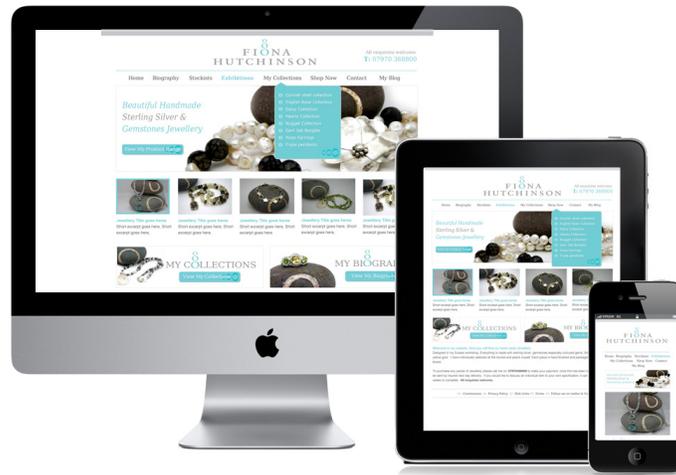


Figura 1.2 Cuando un sitio Web está disponible para múltiples dispositivos, se puede perder la coherencia cuando un usuario interactúa con un dispositivo o con otro. Se pueden estropear controles, elementos gráficos, modelos de navegación, etc. En un dispositivo, el sitio puede ser usable y en otro dispositivo puede ser completamente inutilizable

1.4 Objetivos

General

Desarrollar heurísticas de coherencia que sirvan como guía a los desarrolladores de sistemas interactivos para mantener la coherencia de las meta-interfaces de usuario.

Particulares

- Crear heurísticas de diseño de meta-interfaces orientadas a la coherencia.
- Generar escenarios donde se identifique cómo afecta la coherencia en la usabilidad de una meta-interfaz.

- Identificar elementos genéricos en el espacio de dimensiones de meta-interfaz que preserven la usabilidad y la coherencia.
- Validar las heurísticas de coherencia mediante una aplicación como prueba de concepto y probarla con usuarios finales.

1.5 Organización del documento

La presente tesis de maestría está estructurada en seis capítulos (ver Figura 1.3). Después de haber presentado la motivación del proyecto y de haber planteado el problema que se pretende resolver, así como los objetivos que se persiguen, se expone el estado del arte en el Capítulo 2. Particularmente, se describen conceptos en los que se fundamenta el trabajo de tesis como usabilidad, coherencia y el espacio de dimensiones para meta-interfaces. A partir de estas características se destacan, mediante un estudio comparativo, las principales ventajas y desventajas de trabajos relacionados. En el Capítulo 3, se describen las heurísticas de coherencia, producto principal de la tesis. Cada heurística encierra su definición, un escenario de meta-interfaz donde se refleja su aplicación y una tabla donde se despliegan los elementos pertenecientes al espacio de dimensiones para meta-interfaz relacionados con el escenario, a partir de esos elementos se formulan una serie de preguntas que guían a los desarrolladores de sistemas interactivos. Finalmente se crea un análisis, aplicando las heurísticas de coherencia a los trabajos relacionados. En el Capítulo 4 se presenta el análisis y diseño del prototipo construido. Se explica su funcionamiento general, así como sus principales componentes. Enseguida, en el Capítulo 5, se describen las tecnologías usadas en la construcción de la aplicación desarrollada como prueba de concepto; se detallan las arquitectura de comunicación y de distribución, las cuales establecen respectivamente la forma de realizar el intercambio de mensajes entre sus componentes y la distribución de los mismos. Se presentan aquí también, los resultados de las pruebas realizadas con usuarios finales del prototipo. Finalmente, el Capítulo 6 recopila conclusiones, aportes, limitaciones y trabajo a futuro.

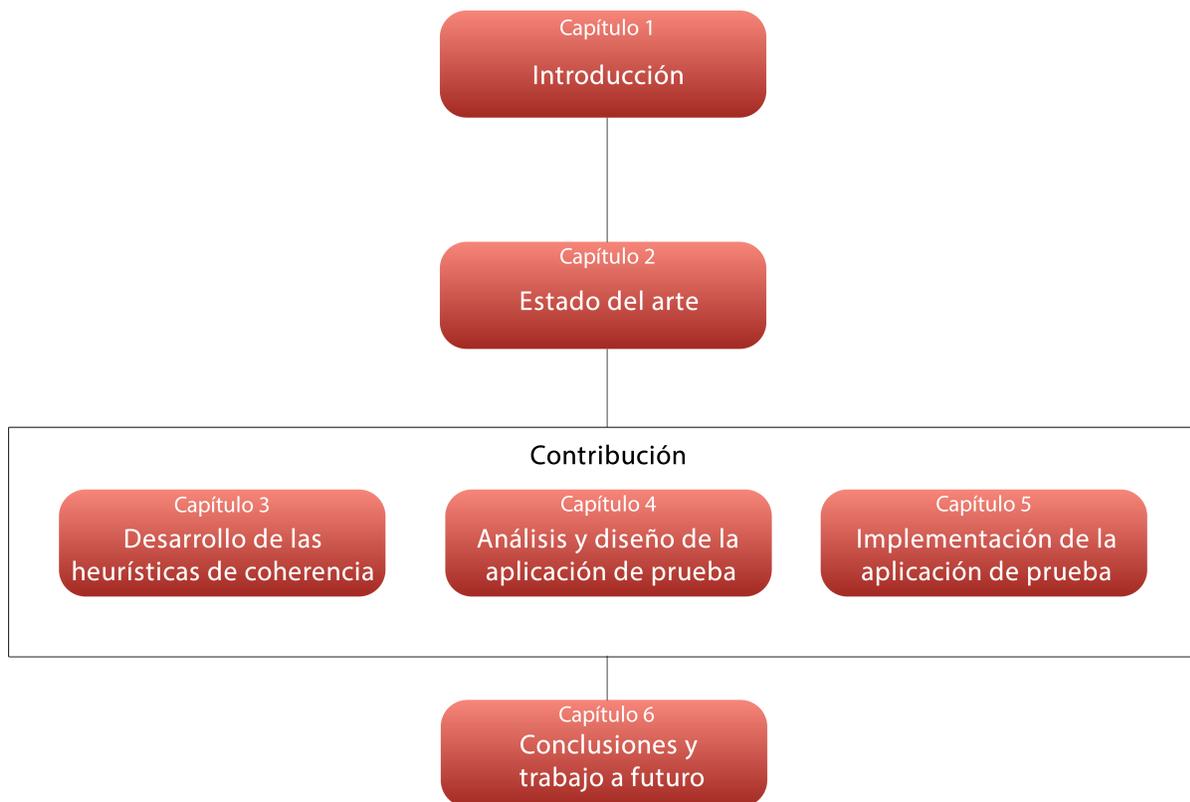


Figura 1.3 Capítulos que conforman la tesis

Capítulo 2

Estado del arte

El capítulo de estado del arte recopila el conocimiento teórico utilizado en el trabajo de investigación. La Sección 2.1 ofrece una descripción de las dimensiones exploradas, ideas y estándares fundamentales para la comprensión y construcción del proyecto. En la Sección 2.2, se presentan los trabajos más sobresalientes que se relacionan con la investigación realizada.

2.1 Marco teórico

Este apartado comprende un compendio de elementos conceptuales que son de vital importancia para el desarrollo del presente trabajo de tesis. Sustentan la formulación de las propuestas presentadas.

El marco teórico está integrado por seis elementos: Ergonomía - que estudia las interacciones entre humanos y sistemas; Usabilidad - se refiere a la calidad de uso de un producto o servicio; Heurísticas de usabilidad para el diseño de interfaces de usuario - usabilidad aplicada al diseño de interfaces de usuario; Coherencia - patrones de diseño para interfaces de usuario; Meta-Interfaz de Usuario - interacción entre usuarios y ambientes interactivos y Espacio de dimensiones para meta-interfaces - elementos genéricos de interacción en meta-interfaces de usuario.

2.1.1 Ergonomía

La ergonomía es un derivado de las palabras griegas *ergon*, o trabajo, y *nomos*, es decir leyes. Mientras que el término trabajo se ha asociado tradicionalmente con una ocupación, un sentido más amplio del término se puede aplicar a cualquier actividad no planificada que

requiere habilidad o esfuerzo. En el año 2000, la Asociación Internacional de Ergonomía (IEA) define la Ergonomía en factores humanos como: “La disciplina científica que se ocupa de la comprensión de las interacciones entre los seres humanos y otros elementos de un sistema, y la profesión que aplica teoría, principios, datos y métodos para diseñar, con el fin de optimizar el bienestar humano y el rendimiento general del sistema.” En el diseño, la ergonomía intenta lograr una relación óptima entre humanos y máquinas en un entorno particular.

Los practicantes de la ergonomía y ergonomistas contribuyen al diseño y evaluación de tareas, trabajos, productos, entornos y sistemas con el fin de hacerlos compatibles con las necesidades, capacidades y limitaciones de las personas [Li, 2009].

2.1.2 Usabilidad

La Organización Internacional para la Estandarización (ISO) ofrece dos definiciones de usabilidad en sus especificaciones [Bevan, 1999]: ISO/IEC 9126 y ISO/IEC 9241.

1. **ISO/IEC 9126:** la usabilidad se refiere a la capacidad de un software de ser comprendido, aprendido y atractivo para el usuario, en condiciones específicas de uso. Esta definición hace énfasis en los atributos internos y externos de un producto, los cuales contribuyen a su funcionalidad y eficiencia. La usabilidad depende no sólo del producto sino también del usuario. Por esta razón, un producto no es en ningún caso intrínsecamente usable, ya que sólo tendrá la capacidad de ser utilizado en un contexto específico por usuarios particulares. La usabilidad no puede ser valorada estudiando un producto de manera aislada [Bevan and Macleod, 1994].
2. **ISO/IEC 9241:** usabilidad es la eficacia, eficiencia y satisfacción con la que un producto permite que un usuario particular alcance sus objetivos en un contexto de uso específico. Esta definición se centra en el concepto de calidad de uso, i.e., se refiere a cómo el usuario realiza con efectividad tareas particulares en escenarios específicos.

A partir de la conceptualización propuesta por ISO, se puede obtener los tres principios en los que se basa la usabilidad:

1. **Facilidad de aprendizaje:** se refiere a la facilidad con la que nuevos usuarios pueden tener una interacción efectiva con la aplicación. Este principio está relacionado con la predicibilidad, la sintetización, la familiaridad, la generalización de conocimientos previos y la consistencia.

2. **Flexibilidad:** hace referencia a la variedad de posibilidades en las que el usuario y el sistema pueden intercambiar información. También abarca la posibilidad de diálogo, la multiplicidad de vías para realizar una tarea, la similitud con tareas precedentes y la optimización del intercambio de información entre el usuario y el sistema.
3. **Robustez:** es el nivel de apoyo al usuario que facilita el cumplimiento de sus objetivos. Este principio está relacionado con la capacidad de observar al usuario, de recuperar información y de ajustar la tarea al usuario.

De las definiciones anteriores se concluye que la usabilidad se refiere al grado en el que el diseño de un sistema facilita o dificulta su manejo por parte del usuario.

2.1.3 Heurísticas de usabilidad para el diseño de interfaces de usuario

Jakob Nielsen presenta las heurísticas para el diseño de interacción en interfaces de usuario [Nielsen, 1990]. Se les llama “heurísticas” porque son reglas amplias y generales mas no directrices específicas de usabilidad (ver Figura 2.1).



Figura 2.1 Heurísticas de usabilidad de Nielsen

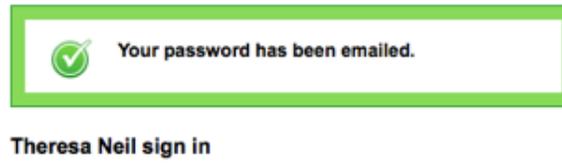


Figura 2.2 Mensaje de retroalimentación cuando se realiza una acción



Figura 2.3 Organización como una biblioteca, que contiene la biblioteca multimedia del usuario: música, películas, espectáculos, audiolibros. Debajo de la biblioteca (LIBRARY) se encuentra la tienda (STORE) donde se pueden comprar más medios para poner en la biblioteca

- **Visibilidad del estado del sistema:** el sistema siempre debe mantener a los usuarios informados acerca de lo que está pasando, a través de la retroalimentación adecuada en un tiempo razonable (ver Figura 2.2).
- **Relación entre el sistema y el mundo real:** el sistema debe hablar el idioma de los usuarios, con palabras, frases y conceptos familiares para los usuarios, en lugar de términos orientados al sistema. El diseñador del sistema debe seguir las convenciones del mundo real, haciendo que la información aparezca en un orden natural y lógico (ver Figura 2.3).
- **Control del usuario y la libertad:** los usuarios suelen elegir funciones del sistema por error y necesitarán, marcada claramente, una “salida de emergencia” para salir del estado no deseado sin tener que pasar a través de un diálogo engorroso. El soporte para deshacer y rehacer es importante (ver Figura 2.4).

| | A | B | C | D |
|---|-------|----------|--------|--------|
| 1 | Item | Quantity | Price | Total |
| 2 | Tacos | 40 | \$5.00 | =B2*C2 |
| 3 | | | | |

Figura 2.4 La edición de la celda muestra los identificadores de fila y columna, así como las celdas utilizadas en la ecuación. La ecuación se puede guardar o cancelar

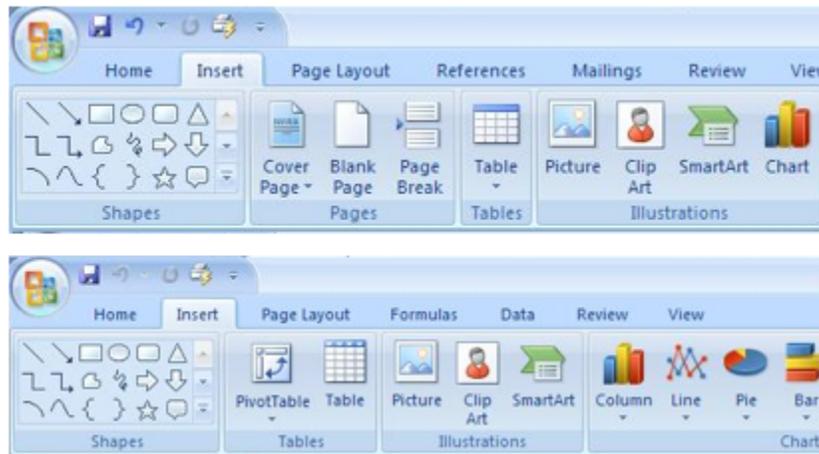


Figura 2.5 Word, Excel y PowerPoint, utilizan la misma barra de herramientas de estilo con las mismas opciones del menú principal: Inicio, Resultados, Insertar, Diseño de página, etc. La consistencia resulta en eficiencia e intuición percibida

- **Consistencia y estándares:** los usuarios no deberían tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo (ver Figura 2.5).
- **Prevención de errores:** incluso mejor que buenos mensajes de error, es realizar un diseño cuidadoso que evita que un problema se produzca en primer lugar, ya sea eliminando situaciones propensas a errores o detectándolas y presentando a los usuarios una opción de confirmación antes de realizar la acción (ver Figura 2.6).
- **Reconocer en lugar de recordar:** reducir al mínimo la carga de memoria del usuario, haciendo visibles objetos y acciones. El usuario no debería tener que recordar la información de una parte del diálogo a otro. Las instrucciones de uso del sistema deben ser visibles o fácilmente recuperables cuando sea apropiado (ver Figura 2.7).

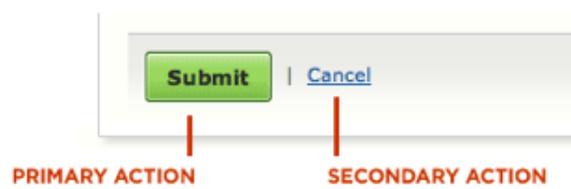


Figura 2.6 Realza la acción primaria con un área de clic más grande y prominente. “Cancelar” y acciones secundarias aparecen sólo como enlaces



Figura 2.7 Previsualización de las tipografías que se pueden seleccionar, en lugar de sólo mostrar su nombre

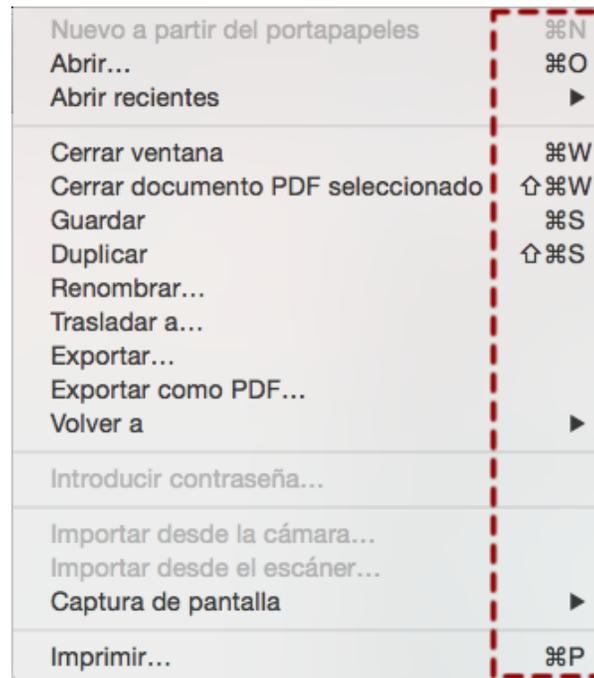


Figura 2.8 Lista de los atajos del teclado y “aceleradores”

- **Flexibilidad y eficiencia de uso:** los “aceleradores” - no vistos por el usuario principiante - a menudo pueden acelerar la interacción para el usuario experto de tal manera que el sistema puede servir tanto a los usuarios noveles como a los de mayor experiencia. El sistema debe permitir a los usuarios adaptar las acciones frecuentes (ver Figura 2.8).
- **Diseño estético y minimalista:** los diálogos no deben contener información que sea irrelevante o raramente necesaria. Cada unidad adicional de información en un diálogo compite con las unidades pertinentes de información y disminuye su visibilidad relativa (ver Figura 2.9).
- **Ayudar a los usuarios a reconocer, diagnosticar y recuperarse de errores:** los mensajes de error deben ser expresados en un lenguaje sencillo (sin códigos), indicar con precisión el problema y sugerir una solución constructiva (ver Figura 2.10).
- **Ayuda y documentación:** a pesar de que es mejor un sistema que puede ser utilizado sin documentación, puede ser necesario proporcionar ayuda y documentación. Dicha información debe ser fácil de buscar, centrarse en la tarea del usuario, listar medidas concretas para llevar a cabo y no ser demasiado grande (ver Figura 2.11).



Figura 2.9 El menú del sitio Web de Microsoft ejemplifica los cuatro principios de diseño visual: **Contraste (a)**: cuando un elemento está seleccionado cambia de color. **Repetición (b)**: todas las secciones que cuentan con subsecciones están señaladas con un pequeño carácter. **Alineación (c)**: todos los enlaces están alineados a la izquierda. **Proximidad (d)**: el submenú se despliega de forma adyacente al elemento seleccionado

Full name

Firstname Secondname ✓ Name looks great.

Email address

howtowinatecommerce@gmail.com ✓ We will email you a confirmation.

Create a password

..... ✗ Password is too obvious.

Choose your username

howtowinatecomm ✓ Username is available.
You can change it later.

Figura 2.10 Proporciona retroalimentación inmediata con instrucciones específicas

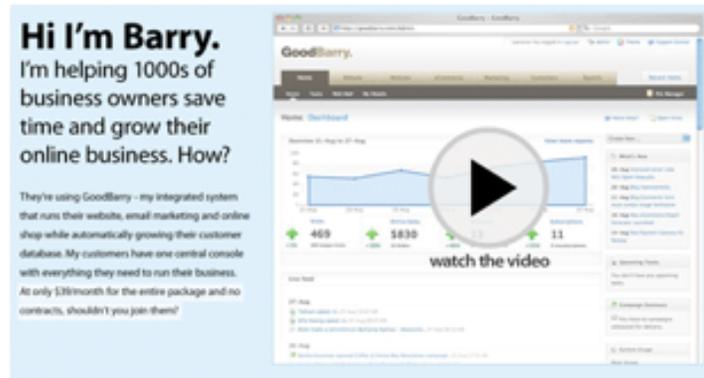


Figura 2.11 Se pueden utilizar videos incrustados para mostrar características, así como para hacer que los usuarios se familiaricen con el producto

2.1.4 Coherencia

Una de las principales formas en que las personas aprenden es mediante el descubrimiento de patrones. Nuevas situaciones se hacen más manejables cuando el conocimiento se convierte en un patrón que los usuarios pueden aplicar a la comprensión de cómo funcionan las cosas. La consistencia es la clave para ayudar a los usuarios a reconocer y aplicar patrones.

La coherencia (o consistencia) visual es el uso constante de los elementos de diseño, como tipografía, diseño, colores, iconos, navegación, imágenes y fondos. Mientras que los usuarios pueden superar ciertas inconsistencias (e.g., rellenar campos de entrada o presionar botones son tareas que los usuarios realizan manualmente), las interfaces consistentes pueden reducir errores y tiempos de finalización de tareas. De igual forma, pueden reducir las curvas de aprendizaje y aumentar la satisfacción del usuario [Grosjean, 2011] (ver Figura 2.12).

Cosas que parecen similares deben hacer cosas similares. Por ejemplo, si se aprende que superficies sobresalientes con etiquetas en ellas son botones que pueden ser presionados, entonces la próxima vez que se vea una superficie que sobresale con una etiqueta en ella, se tiende a reconocerla como un botón. De igual forma, el comportamiento y las convenciones deben ser consistentes a través de tareas y acciones similares [Erickson, 1995].

La inconsistencia causa confusión porque las cosas no funcionan de la forma en que el usuario lo esperaba. No se debe obligar a los usuarios a memorizar excepciones a las reglas, ya que esto aumenta la carga cognitiva y provoca resentimiento. La atención a la coherencia es importante para infundir confianza en el producto, ya que da la impresión de que hay un diseño racional, confiable y lógico detrás de escena.

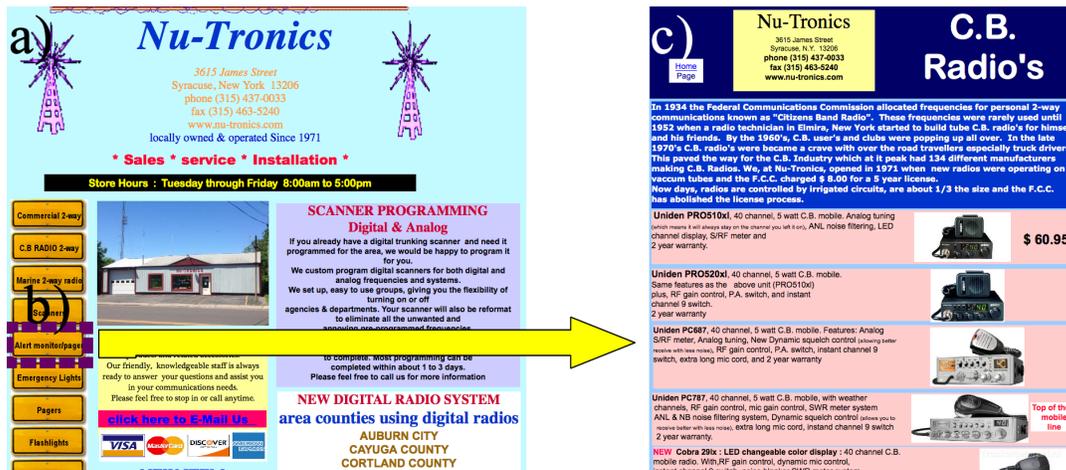


Figura 2.12 Dos páginas de un mismo sitio Web. El enlace (b) en la página (a) redirecciona a la página (c). No se conserva ninguna estructura, colores, imágenes, tipografía ni elementos de navegación. Todo esto causa confusión y aumenta la carga cognitiva del usuario

Para aplicaciones de escritorio y móviles, se debe tratar de entender y cumplir con las directrices de la interfaz de usuario del sistema operativo o plataforma (ver Figura 2.13). La consistencia con estas convenciones estándares reduce el número de nuevas cosas que un usuario necesita para aprender.

La consistencia permite que las personas transfieran sus conocimientos y habilidades de una parte de la interfaz de usuario de una aplicación a otra y de una aplicación a otra aplicación. Una aplicación coherente no es una copia servil de otras aplicaciones y no es estilísticamente estancada; más bien, se presta atención a las normas y paradigmas con los que las personas se sientan cómodas, de igual forma, les proporciona una experiencia internamente consistente [Apple, 2015].

Para determinar si una aplicación sigue el principio de coherencia, se debe de pensar en estas preguntas [Apple, 2015]:

- **¿Cumple con los estándares?** - ¿Utiliza los controles, vistas e iconos proporcionados por el sistema correctamente? ¿Incorpora las funciones del dispositivo de la manera en que los usuarios esperan? ¿La aplicación utiliza correctamente los atajos del teclado?
- **¿Es la aplicación coherente en sí misma?** - ¿Utiliza texto, terminología y estilo uniforme? ¿Los mismos iconos siempre significan lo mismo? ¿La gente puede predecir lo que sucederá cuando realizan la misma acción en diferentes lugares? ¿Los elementos de

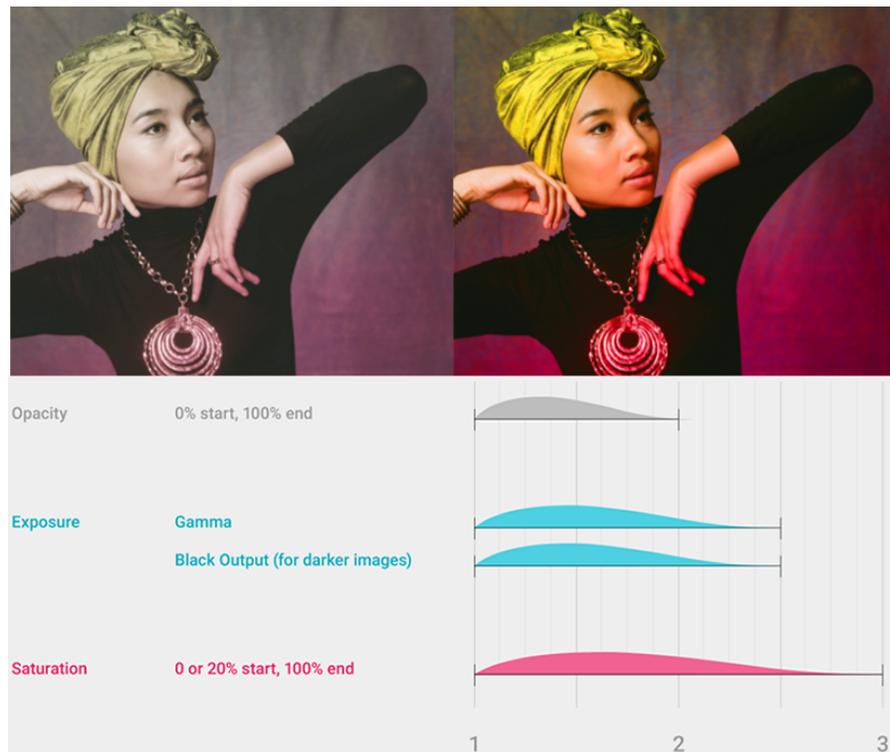


Figura 2.13 Google establece niveles recomendados de opacidad, exposición y saturación para mejorar la visualización de imágenes en plataformas móviles

interfaz de usuario personalizados se ven y se comportan de la misma manera a lo largo de la aplicación?

- **¿Cómo fueron las primeras versiones de la aplicación?** - Dentro de lo razonable, ¿es la aplicación compatible con sus versiones anteriores? ¿Los términos y significados sigue siendo los mismos? ¿Los conceptos fundamentales y la funcionalidad principal permanecen esencialmente sin cambios?
- **¿Cumple con las expectativas de los usuarios?** - ¿Cumple con las necesidades del usuario sin características adicionales? ¿Se ajusta al modelo mental del usuario?

2.1.5 Meta-interfaz de usuario

Una meta-interfaz de usuario proporciona al usuario final una vista interactiva en un entorno físico o virtual que puede ser observado y manipulado en tiempo de ejecución. La meta-interfaz de usuario cierra la brecha entre los proveedores de recursos y los usuarios finales, mediante la abstracción de características de recursos como actividades ejecutables, que pueden ser ensambladas en tiempo de ejecución para alcanzar un objetivo común [Vanderhulst, et al., 2009].

A diferencia de las aplicaciones de Computación tradicionales, un entorno ubicuo (*pervasive*) no ofrece necesariamente apoyo a los objetivos específicos de forma predeterminada. En su lugar, permite al usuario adaptar el comportamiento del entorno para apoyar sus objetivos en función de la situación actual. Es posible que el usuario final por si mismo pueda enriquecer su entorno, por ejemplo, mediante la integración de recursos como los teléfonos móviles. El verdadero poder de la computación ubicua no viene simplemente de cualquier dispositivo o servicio, sino que surge de la interacción de todos ellos.

Las personas deben ser capaces de dar forma a su entorno ubicuo de acuerdo a sus demandas y preferencias, lo que trae ciertas necesidades del usuario final:

- **Obtener una visión de los servicios digitales que el ambiente ofrece**, es decir, ¿qué recursos están a la mano y qué tareas soportan?
- **Interactuar con los recursos disponibles**, por ejemplo, navegar por la lista de contactos de un teléfono móvil a través del parabrisas de un coche o navegar por una serie de fotografías visualizadas en un televisor utilizando un dispositivo de mano.

- **Configurar el comportamiento y contexto por defecto de los recursos**, por ejemplo, apagar automáticamente las luces y bajar la calefacción al salir de la casa.

Para hacer frente a estas necesidades, un entorno ubicuo, independiente de su dominio de aplicación, debe ofrecer un punto de control para inspeccionar y manipular su configuración actual. La interfaz de usuario de dicho punto de control, se denota como meta-interfaz de usuario.

Una meta-interfaz tiene por objetivo simplificar el papel del usuario final (la interacción con el entorno) y el papel del desarrollador (el diseño y la integración de nuevos recursos en el entorno). La función específica del desarrollador depende del tipo de recurso: un desarrollador de software proporciona servicios, mientras que un diseñador gráfico proporciona interfaces de usuario, etc. Los roles del usuario final y el desarrollador se fusionan en lo que respecta a la integración de nuevos recursos y configuración del entorno. Servicios o interfaces de usuario pueden ser programados para adaptarse automáticamente al contexto de uso, pero los usuarios finales todavía tienen que afinar la configuración de un recurso. La meta-interfaz de usuario ayuda a cerrar la brecha entre los dos campos.

Desde la perspectiva de un usuario final, la meta-interfaz de usuario actúa como un instrumento para examinar y manipular el estado de su entorno.

Desde el punto de vista de un desarrollador, la meta-interfaz de usuario es un componente de software con un conjunto de características comunes: descubrir los recursos disponibles, integrarlos en una vista y hacer que la interfaz de software de los mismos, sea accesible a través de una interfaz de usuario.

Escalar un meta-interfaz de usuario a un contexto de uso, va más allá de la adaptación de su interfaz gráfica para resoluciones de pantalla de dispositivos heterogéneos y sus modalidades de interacción. La adaptación de la meta-interfaz de usuario también abarca muchos otros factores de los que destacan dos:

- **Trasfondo del usuario**: dado que las personas están interesadas en diferentes tareas, su punto de vista en un entorno similar puede diferir. La meta-interfaz de usuario tiene que anticipar esta observación, mediante la personalización de las vistas, de acuerdo con las tareas y objetivos previstos de los usuarios.
- **Explosión de recursos**: un ambiente saturado con recursos, es difícil de inspeccionar. Especialmente en entornos desconocidos, será difícil de localizar, por ejemplo, la impresora más cercana en una habitación. Para evitar una explosión de recursos, la información

espacial podría ser tomada en cuenta para mostrar sólo los servicios y tareas próximos al usuario.

2.1.6 Espacio de dimensiones para meta-interfaces de usuario

Coutaz [Coutaz, 2006] conforma un espacio de dimensiones, representado en la Figura 2.14, el cual se tomará en cuenta para el desarrollo de las heurísticas de coherencia (ver Sección 3.1).

■ Técnicas de interacción

- **Nivel de integración** [Coutaz, 2006]: teniendo en cuenta que un meta-interfaz de usuario proporciona a los usuarios los medios para regular los servicios específicos de dominio en un ambiente interactivo, ¿Cómo se relacionan los elementos de la interfaz gráfica de usuario con los servicios específicos de dominio? ¿Cuál es el nivel de integración? Se proponen dos perspectivas para responder:
 - Todos o parte de los componentes de la interfaz gráfica de usuario de la meta-interfaz están “**embebidos**” en los componentes de los servicios específicos de dominio.
 - Alternativamente, los componentes de interfaz gráfica de usuario, de los servicios de meta-interfaz de usuario, no se mezclan con los componentes de los servicios específicos de dominio. Es decir, los componentes son **externos**.
- **Representación de objetos** [Coutaz, 2006]: los objetos que pueden estar implicados en un servicio de meta-interfaz de usuario se indican con los elementos de un vocabulario. Algunos objetos pueden estar indicados por un representante que, a su vez, puede ser **numérico** o **físico**. Otros no tienen **ningún representante**. Por ejemplo, al escribir en un PDA mediante un stylus, éste se convierte en la representación física de una pluma. Por otra parte, en un sistema GPS, las calles del mundo real se representan numéricamente como líneas en mapas gráficos. Cuando se conecta un ratón a un puerto USB, el objeto físico “ratón” es parte de la frase. No está representado. La ausencia de representante se podría aplicar a los objetos digitales: al mover una ventana, se actúa en la ventana *per se*, no en un representante.
- **Objeto/función/designación** [Coutaz, 2006]: la elaboración de un enunciado, que requiere objetos y designación de funciones, puede ser **directa** al actuar sobre los elementos del vocabulario o **indirecta** por la forma de los “instrumentos”.

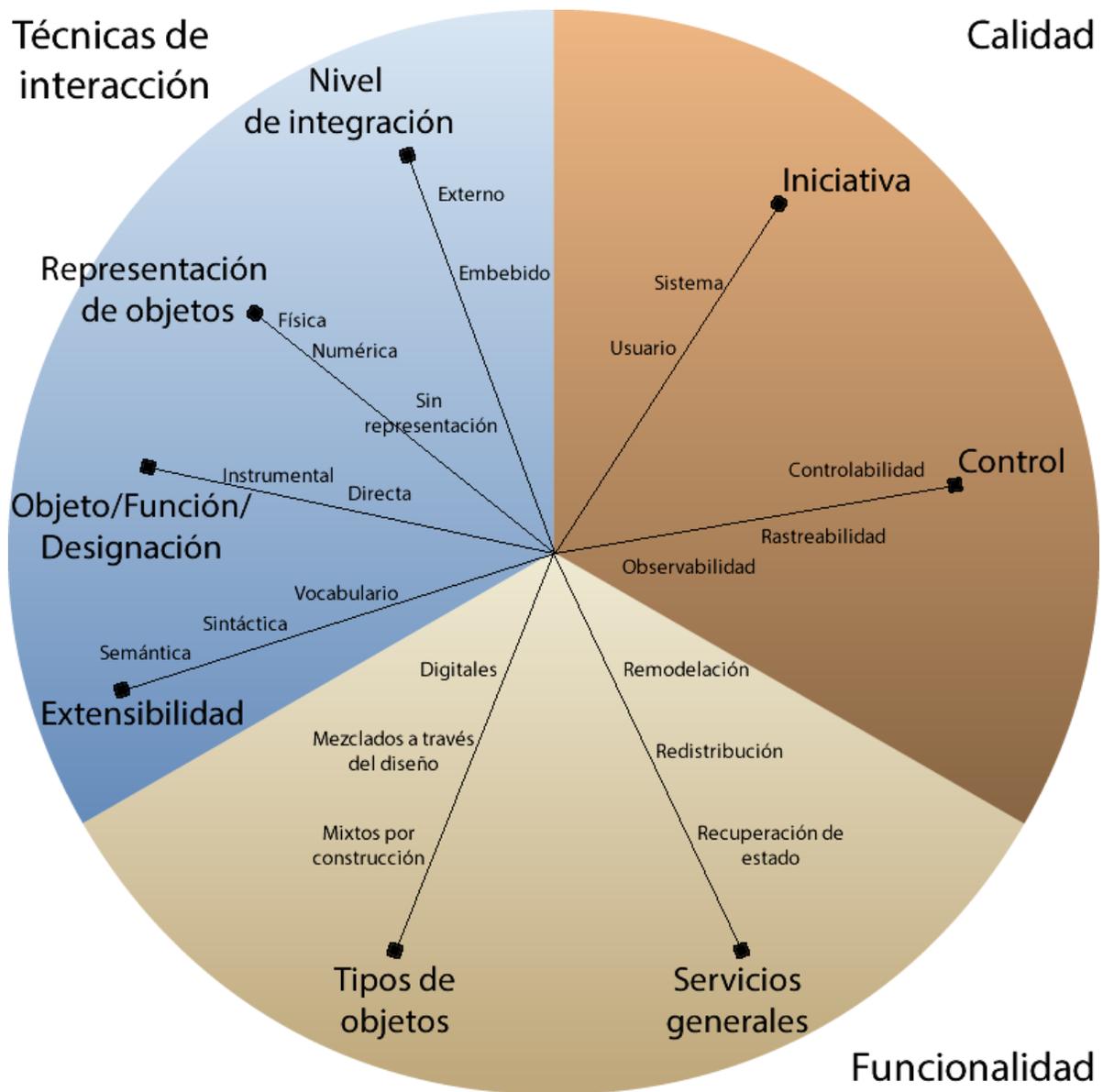


Figura 2.14 Espacio de dimensiones para meta-interfaces de usuario [Coutaz, 2006]

A su vez, los “instrumentos” pueden ser físicos o digitales con varios niveles de indirección. Mover una ventana con un dedo es una acción directa, mientras que, moverla con una pluma es instrumental (la pluma actúa como un instrumento físico). Alternativamente, mover una ventana con un ratón es instrumental con un nivel adicional de indirección: la ventana no es movida por el ratón, sino por un puntero, una representación digital del ratón.

- **Extensibilidad** [Coutaz, 2006]: cuando se razona sobre el vocabulario, la extensibilidad es un problema típico a considerar ¿Es posible que los usuarios puedan ampliar el **vocabulario** de la técnica de interacción, cambiar su **sintaxis**, y desde allí, extender su **semántica**, por tanto, crear nuevos servicios de meta-interfaz de usuario? En un dominio sin límites como la Computación Ubicua, la extensibilidad parece inevitable, pero el riesgo de introducir una complejidad adicional es alta. En equipos de escritorio convencionales, la mayoría de los usuarios pueden construir oraciones simples como “mover esta ventana aquí” o “cancelar este proceso”. De esta manera se entra en el área de desarrollo del usuario final. Aunque los usuarios finales no son programadores, quieren programar su espacio interactivo sin tener que programar.

■ **Calidad**

- **Control** [Coutaz, 2006]: como para cualquier sistema interactivo, los servicios de una meta-interfaz pueden ser ejecutados por **iniciativa del sistema** y/o la **iniciativa de usuario**. La iniciativa depende de las interacciones explícitas e implícitas que a su vez, se basan en un modelo de contexto de uso. Un usuario puede tener el control de una meta-interfaz a tres niveles:
 - **Observabilidad**: los usuarios deben ser capaces de evaluar el estado interno del servicio, desde su representación perceptible actual.
 - **Rastreabilidad**: los usuarios pueden observar la evolución del servicio a través del tiempo, pero no pueden modificar esta evolución.
 - **Controlabilidad**: los usuarios pueden observar, rastrear e intervenir en la evolución del servicio a través de una meta-interfaz.

■ **Funcionalidad**

- **Tipos de objetos** [Coutaz, 2006]: los tipos de objetos que participan en los servicios de una meta-interfaz pueden ser digitales, “**mezclados a través del diseño**”

y/o “**mixtos por construcción**”. Aplicaciones y archivos son ejemplos típicos de los objetos *digitales* manipulados a través de los servicios de una meta-interfaz. Interactores tales como ventanas, punteros, menús y formas, son otros ejemplos de objetos *digitales*.

- **Un objeto “mezclado a través del diseño”** es una entidad que resulta del acoplamiento, por el diseñador, de entidades físicas con servicios digitales. Un PDA y un teléfono móvil son objetos mezclados a través del diseño: el conjunto de componentes físicos con piezas digitales han sido realizados por los diseñadores de antemano.
- **Un objeto “mixto por construcción”** es un objeto mixto que resulta de la unión, por el usuario final, de entidades físicas con servicios digitales a fin de que esos objetos cumplan con su razón de ser. Por ejemplo, para funcionar como un dispositivo señalador, el objeto físico que se sostiene en la mano (ratón) debe acoplarse con el *driver* del sistema por parte del usuario final.
- **Servicios generales** [Coutaz, 2006]: iniciar/detener la ejecución de un servicio, mover o renombrar archivos, cortar y pegar datos, así como buscar/encontrar, son las bases de una meta-interfaz convencional. Todas las bases anteriores son conceptualmente válidas en un ambiente interactivo, pero tienen que ser ampliadas y refinadas.
 - **Remodelación** [Coutaz y Calvary, 2012]: consiste en el cambio de la “forma” de la interfaz gráfica de usuario mediante la aplicación de una o varias transformaciones en todos, o parte, de la interfaz de usuario (ver Figura 2.15). Estas transformaciones incluyen: la supresión de los componentes de interfaz de usuario que se vuelven irrelevantes en la nueva situación/contexto; la inserción de nuevos componentes de interfaz de usuario para facilitar el acceso a nuevos servicios relevantes en la nueva situación/contexto; la sustitución de componentes de interfaz de usuario cuando los mismos son reemplazados por otros nuevos (la sustitución puede ser vista como una combinación de la supresión y de inserción); la reorganización de los componentes de interfaz de usuario revisitando su disposición espacial y/o su dependencia temporal.
 - **Redistribución** [Coutaz y Calvary, 2012]: denota la reasignación de los componentes de interfaz de usuario del sistema a diferentes recursos de interacción



Figura 2.15 Remodelación. Se diseñan formas específicas de presentación para cada contexto de uso

(ver Figura 2.15). La granularidad de la redistribución de la interfaz de usuario puede variar de nivel de aplicación a nivel de píxel:

- ◇ A nivel de aplicación, la interfaz de usuario está totalmente replicada en cada dispositivo. Cuando la redistribución es dinámica, toda la interfaz de usuario del sistema interactivo migra a un nuevo dispositivo de computación, que a su vez puede desencadenar una remodelación.
 - ◇ A nivel de área de trabajo, la unidad de distribución es el espacio de trabajo, el cual es un espacio lógico que apoya la ejecución de un conjunto de tareas conectadas lógicamente.
 - ◇ La distribución de nivel interactor es un caso especial del nivel de espacio de trabajo, donde la unidad de distribución es un interactor elemental.
 - ◇ A nivel de píxel, cualquier componente de la interfaz de usuario se puede dividir en varios recursos de interacción.
- **Recuperación de estado** [Coutaz y Calvary, 2012]: la granularidad de recuperación del estado caracteriza el esfuerzo que los usuarios deben realizar para continuar con su actividad después de haberse producido un proceso de

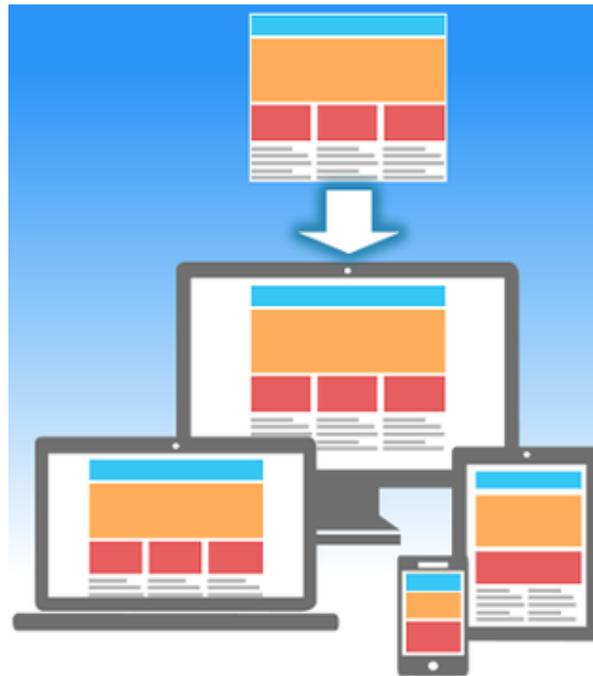


Figura 2.16 Redistribución. Una misma presentación se adapta a los cambios de contexto

adaptación. La recuperación de estado puede llevarse a cabo en los niveles de sesión, de tareas o de acción física:

- ◇ **Nivel de sesión:** el usuario tiene que reiniciar su actividad desde el estado inicial del sistema, i.e., el estado en el que el sistema inicia cuando es lanzado;
- ◇ **Nivel de tarea:** el usuario puede continuar el trabajo desde el inicio de la tarea interrumpida (a condición de que la tarea es realizable en la IU);
- ◇ **Nivel de acción física:** el usuario puede continuar la tarea actual en el punto exacto donde la interrumpió (a condición de que la tarea es realizable en la nueva versión de la IU).

2.2 Trabajos relacionados

A continuación, se describen brevemente algunos de los trabajos relacionados más relevantes en el dominio de las meta-interfaces de usuario.

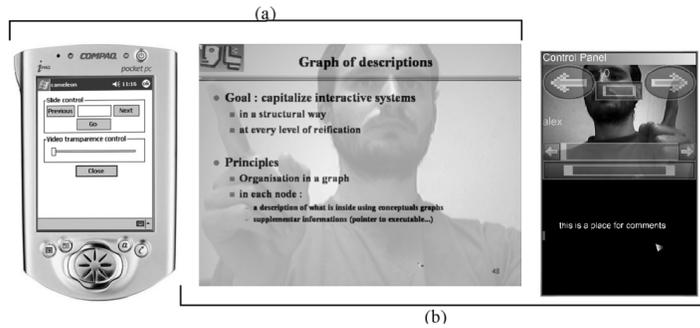


Figura 2.17 La interfaz de usuario de CamNote. (a) La interfaz de usuario de CamNote cuando se distribuye en una PC y una Pocket PC, la cual se transforma en un control remoto para la navegación de diapositivas y la nivelación de transparencia del video; (b) el panel de control cuando aparece en la pantalla de una PC que, a diferencia del dispositivo móvil, muestra también el espacio para las notas del presentador.

2.2.1 CamNote

CAMELEON Note¹ [Demeure et al., 2005] es un visor de diapositivas que se ejecuta en una plataforma heterogénea dinámica, la cual puede ir desde una sola PC hasta un grupo compuesto por una computadora y un PDA. Su interfaz de usuario se estructura en cuatro áreas de trabajo: (a) un visor de diapositivas, (b) un editor de notas para asociar comentarios a las diapositivas, (c) un visor de video, también conocido como “espejo de píxeles”, que muestra un video en directo del conferencista, y (d) un panel de control para navegar por las diapositivas y configurar el nivel de transparencia del espejo.

La Figura 2.17 muestra una configuración en la interfaz gráfica de usuario, la cual se distribuye a través de las pantallas de una computadora y de un PDA. El visor de diapositivas se muestra en un lienzo rotativo que se puede orientar apropiadamente cuando se proyectan sobre una superficie horizontal. Si el PDA desaparece, el panel de control migra automáticamente a la pantalla de la PC. Debido a los diferentes recursos que están disponibles, el panel de control incluye diferentes *widgets*, sino también una representación en miniatura de video del orador. Durante el proceso de adaptación, los usuarios pueden ver que el panel de control emerge progresivamente desde el visor de diapositivas para que puedan evaluar el desarrollo de dicho proceso de adaptación.

Las principales limitaciones de CamNote son:

- El contexto de uso está limitado a la plataforma.

¹CamNote es una contracción de CAMALEON Note

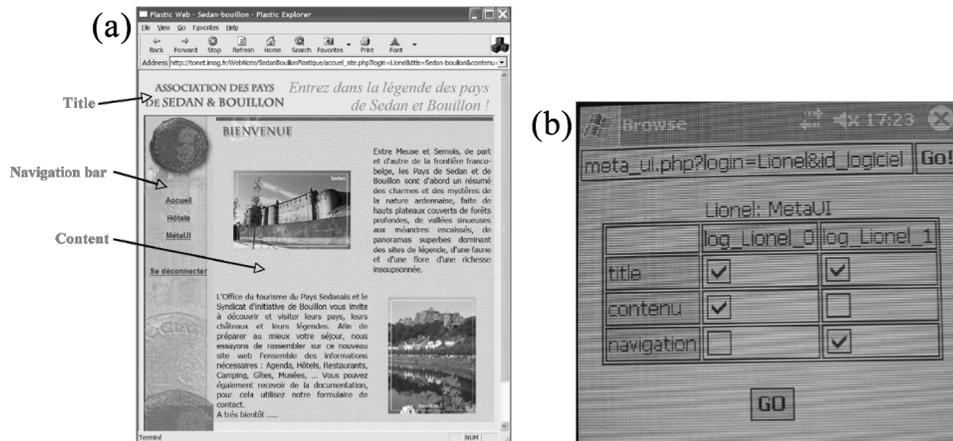


Figura 2.18 El sitio Web Sedan-Bouillon. La interfaz de usuario centralizada en la pantalla de una PC (a) El panel de control de la meta-interfaz de usuario para visualizar los espacios de trabajo de interfaz de usuario de atributos a través de los recursos de interacción disponibles. Las líneas de la matriz corresponden a los espacios de trabajo y las columnas indican los navegadores utilizados por el mismo usuario (b)

- El usuario no tiene control sobre el proceso de adaptación.

2.2.2 Sitio Web Sedan-Bouillon

Sedan-Bouillon [Coutaz, 2006] es un sitio Web que tiene como objetivo promover el turismo en las regiones de Sedan (Francia) y Bouillon (Bélgica). Ofrece a los turistas información para visitar estas regiones e incluye una selección de hoteles, campamentos y restaurantes.

Al iniciar sesión de forma dinámica a la misma página Web con un PDA y una PC, los usuarios son informados en el PDA que pueden distribuir los componentes de la interfaz de usuario del sitio a través de los recursos de interacción disponibles. En el ejemplo de la Figura 2.18, el usuario pide la siguiente configuración: el título debe aparecer en el PDA, así como en la PC (las casillas de título están marcadas para los dos navegadores disponibles), mientras que el contenido debe permanecer en la PC y la barra de navegación debe migrar al PDA. La Figura 2.19 muestra la interfaz de usuario resultante. En cualquier momento, el usuario puede pedir una reconfiguración de la interfaz de usuario, seleccionando el enlace “meta-interfaz de usuario” en la barra de navegación. La interfaz de usuario se reconfigurará en consecuencia.

Las limitantes de mayor importancia del sitio Web Sedan-Bouillon son:

- De igual forma que CamNote, el contexto de uso se encuentra limitado a la plataforma.

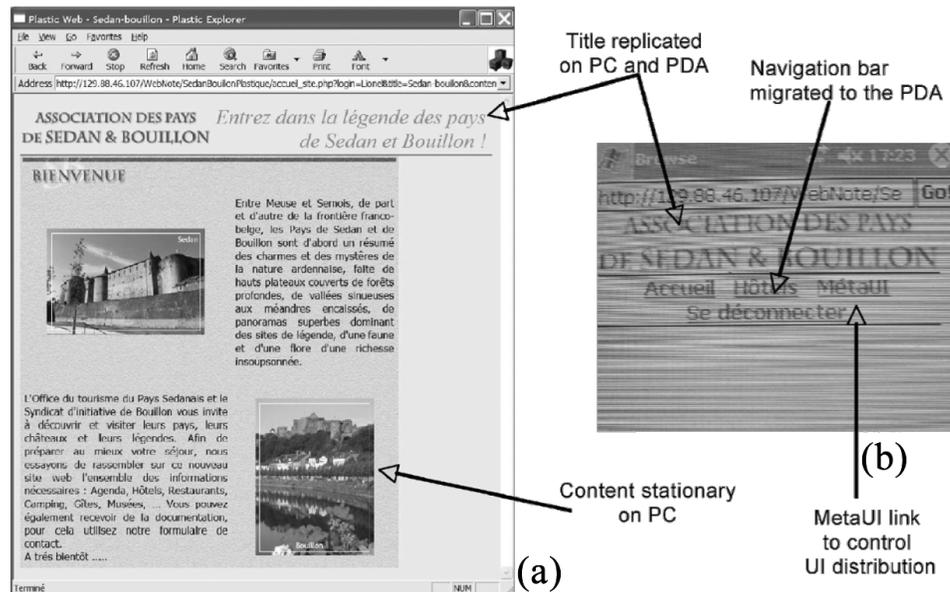


Figura 2.19 El sitio Web Sedan-Bouillon cuando se distribuye a través de los recursos de interacción disponibles. El enlace MetaUI (b) permite a los usuarios volver al panel de configuración que se muestra en la Figura 2.18-b.

- Si la adaptación se produce cuando el usuario está llenando el formulario, el contenido del mismo se pierde a causa del propio proceso de adaptación.

2.2.3 Photo-Browser

Photo-Browser [Coutaz, 2010] soporta la navegación de fotos de una manera centralizada o distribuida en función de la disponibilidad de un conjunto dinámico de dispositivos heterogéneos. Estos incluyen una mesa Diamond Touch interactiva, una pared y un teléfono inteligente (gPhone) con Windows, MacOS X y Android. La interfaz de usuario de Photo-Browser se compone de forma dinámica de los siguientes elementos:

- Un componente de Java que muestra una lista de los nombres de las imágenes (ver Figura 2.20-a).
- Un navegador basado en HTML para navegar a través de las imágenes establecidas (ver Figura 2.20-b).

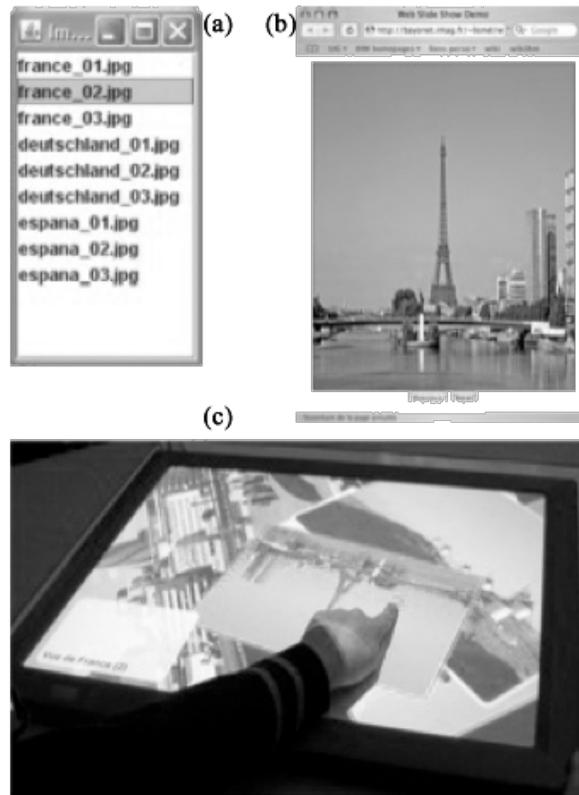


Figura 2.20 La aplicación Photo-Browser: una composición dinámica de componentes ejecutables y transformables, gestionado por un conjunto dinámico de dispositivos interconectados basados en diferentes plataformas (Windows, MacOS X y Android). (a) Lista de nombres de los archivos; (b) Navegador HTML para desplegar imágenes; (c) Mesa digital con la que los usuarios pueden interactuar mediante gestos táctiles.

- Un componente Tcl-Tk que se ejecuta en la superficie interactiva de múltiples puntos (ver Figura 2.20-c).
- Un componente de Java que se ejecuta en el dispositivo gPhone para navegar secuencialmente a través de las fotos, usando los botones “Siguiente” y “Anterior” (ver Figura 2.21).

Las limitaciones más relevantes de Photo-Browser son:

- La meta-interfaz de usuario es simulada ya que no existe cambio de apariencia.

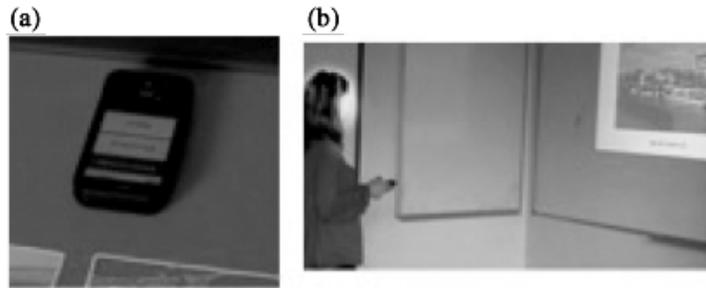


Figura 2.21 Conexión de un gPhone en el espacio interactivo colocándolo sobre la mesa interactiva (a). Utilizando gPhone como control remoto para recorrer las fotos mostradas por el navegador de interfaz de usuario HTML (Figura 2.20-c) y el video proyectado en la pared (b).

- Los métodos de interacción del usuario con los recursos de interacción disponibles son limitados; las interfaces gráficas de usuario se diseñaron para cada dispositivo en específico, esto es, no hay adaptación.

Capítulo 3

Desarrollo de las heurísticas de coherencia

El presente capítulo presenta el producto principal del trabajo de tesis, las heurísticas de coherencia para diseñar meta-interfaces de usuario. Primero, se describe la estructura general de cada heurística y la relación entre los conceptos del marco teórico (ver Sección 2.1) y la formulación de las propias heurísticas de coherencia. Después, cada heurística es presentada y se describen sus particularidades. Finalmente, se utilizan las heurísticas de coherencia para analizar los trabajos relacionados presentados en la Sección 2.2.

3.1 Estructura de las heurísticas de coherencia

Con base en las heurísticas de usabilidad de Nielsen (ver Sección 2.1.3), se formula una serie de heurísticas orientadas a la coherencia para su aplicación en meta-interfaces.

Las heurísticas se conectan con las meta-interfaces mediante el espacio de dimensiones de Coutaz (ver Sección 2.1.6). Cada heurística se expresa mediante una tabla cuya primera columna contiene las tres dimensiones que conforman el espacio. La segunda columna despliega los elementos de cada dimensión que se consideran más relevantes en consonancia con la heurística, i.e., aquellos elementos que permiten en conjunto el cumplimiento de la heurística. La tercera columna presenta una serie de preguntas para los desarrolladores de sistemas interactivos. Las preguntas tienen la intención de guiar a los desarrolladores en sus decisiones de diseño, ya sea recordándoles elementos de interacción que deben integrar, o bien, indicándoles cómo conformar la interfaz de usuario.

En cada caso, se creó un escenario donde se refleja de manera positiva el concepto de una heurística en particular.

Se proponen cinco heurísticas de coherencia (ver Figura 3.1): Honestidad de meta-interfaces (Sección 3.2), Núcleos funcionales (Sección 3.3), Multimodalidad (Sección 3.4), Limitaciones de usabilidad (Sección 3.5) y Trazabilidad (Sección 3.6).

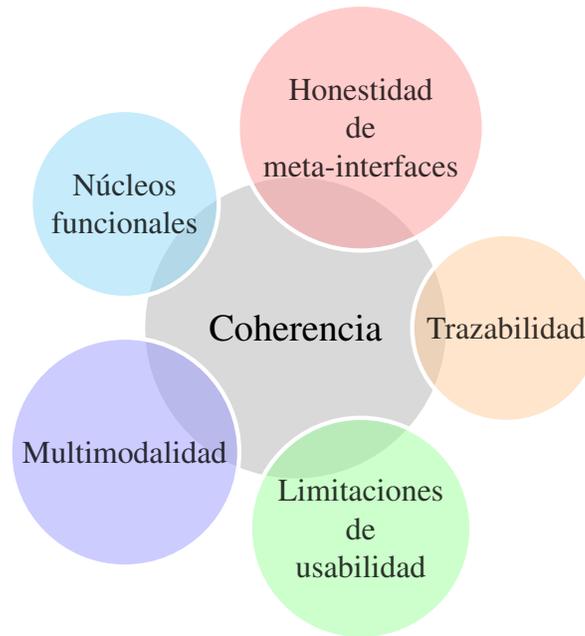


Figura 3.1 Heurísticas de coherencia para meta-interfaces

3.2 Honestidad de meta-interfaces

La honestidad de meta-interfaces se refiere a que los componentes de interacción deben hacer lo que dicen y se deben comportar de la manera esperada. Una meta-interfaz honesta tiene el cometido de reforzar la decisión del usuario de utilizar el sistema. Cuando los elementos de una meta-interfaz son confusos, engañosos o incluso sospechosos, la confianza de los usuarios comenzará a menguar.

3.2.1 Descripción de escenario

Alice está utilizando un editor de video en su PC. Está por terminar un proyecto y desea ver el resultado en pantalla completa de la PC. Para ello necesita acceso a los controles de reproducción, que le permiten a Alice ver fragmentos del clip de manera repetitiva, avanzar o

retroceder fotograma a fotograma y ver el video completo. Alice cuenta con su teléfono móvil, por lo que decide que ahí desplegará los controles.

El editor de video detecta el teléfono, por lo que Alice puede verlo disponible en una lista de dispositivos desde su PC. Alice autoriza que su teléfono pueda recibir interfaces. Lo anterior se ve reflejado inmediatamente en el editor de video, que ahora no sólo indica a Alice que su teléfono está disponible, sino que ahora puede admitir interfaces. Lo único que a Alice le resta por hacer es arrastrar el grupo de controles de reproducción desde el editor de video hasta la lista donde aparece su teléfono.

Ahora Alice es capaz de controlar la reproducción del video desde su propio teléfono móvil, mientras observa el video en la pantalla de su PC.

3.2.2 Dimensiones, elementos y preguntas

Para asegurar la honestidad de meta-interfaces se deben contestar las preguntas contenidas en el Cuadro 3.1, acordes al espacio de dimensiones para meta-interfaces (ver Sección 2.1.6).

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------------|--|
| Técnicas de interacción | Representación de objetos | ¿Cómo se representarán los elementos del espacio interactivo? ¿Cómo se representarán las interfaces? |
| | Nivel de integración | ¿Dónde se distribuirán las interfaces? ¿Cómo se retroalimentará al usuario sobre las interfaces distribuidas? |
| Calidad | Nivel de control | ¿Los elementos del espacio interactivo se comunicarán automáticamente? ¿Cómo se informará al usuario sobre el estado de los elementos del espacio interactivo? ¿Cómo se dará acceso a la distribución de interfaces? |
| Funcionalidad | Tipos de objetos | ¿Qué tipo(s) de elementos integran el espacio interactivo? |
| | Servicios generales | ¿Se agregan/eliminan constantemente elementos al espacio interactivo? ¿Cómo se agregan/eliminan elementos al espacio interactivo? ¿Las interfaces son dinámicas? |

Cuadro 3.1 Elementos a considerar para garantizar meta-interfaces honestas

La dimensión *técnicas de interacción* denota la interfaz gráfica de usuario de las meta-interfaces.

Una meta-interfaz está compuesta de objetos, que pueden estar *representados* numericamente o de forma física. Otros no tienen representación.

El editor de video representa de forma gráfica (numericamente) sus interfaces y los elementos que pueden formar parte de ella. Los controles de reproducción remotos se representan de forma física en el teléfono, mientras que el teléfono *per se* no tiene representación, ya que se está usando de forma física en el ambiente interactivo.

El nivel en que los objetos se *integran* puede ser: todos están embebidos en la interfaz de dominio, o bien pueden estar de forma externa.

Como Alice arrastra un elemento hasta una lista de dispositivos, i.e., una interfaz dedicada, por lo tanto, la integración es externa.

La dimensión de *calidad* expresa lo bien que el sistema cumple con o conforme a un diseño determinado.

Como para cualquier sistema interactivo, los servicios de un meta-interfaz de usuario pueden ser ejecutados por iniciativa del sistema y/o la iniciativa de usuario.

Cuando el editor de video detecta el teléfono móvil y lo despliega en la lista, el sistema tiene la iniciativa. Pero cuando Alice decide arrastrar un componente hacia su teléfono, ella tiene el *control* sobre esa interacción.

La *funcionalidad* es la dimensión que explora las bases, que, como cualquier sistema, las meta-interfaces deben cumplir.

Los *tipos de objetos* que participan en los servicios de una meta-interfaz pueden ser digitales, “mezclados a través del diseño” y/o “mixtos por construcción”.

Cuando Alice despliega los controles a su teléfono, éste se vuelve un objeto “mezclado a través del diseño”, ya que es un objeto físico que se mezcla con una interfaz digital. Lo anterior permitido por el diseñador de la aplicación, mediante *servicios genéricos* como acoplamiento, redistribución y readaptación.

3.3 Núcleos funcionales

Los núcleos funcionales son unidades indivisibles de interfaces gráficas. Los elementos que integran un núcleo funcional forman un campo semántico, si lo sobrepasan pierden significado.

3.3.1 Descripción de escenario

Bob está jugando un videojuego en su PC. Como todo juego de video, el de Bob posee múltiples elementos gráficos, algunos, en ocasiones, resultan más importantes que otros. Existe un componente que es igual de útil en todo momento, el mapa; permite a Bob estar consciente de su entorno, ver en dónde se encuentra su personaje, situar a sus enemigos y ver la localización de ciertos objetos o destinos que le resultarán útiles en algún momento.

Bob tiene la posibilidad de emplazar el mapa a una tableta, lo que le permite visualizarlo con más detalle, además, despeja un poco el resto de la interfaz del videojuego para que Bob no tenga que hacer una pausa cada vez que quiera consultar el mapa.

Cuando Bob muda el mapa a la tableta, también se mueven todos los elementos inherentes a éste. Botones para alejar y acercar, guías y filtros que permiten conocer el punto exacto de alguna misión, coordenadas, simbología, etc. Son piezas cuya existencia está supeditada al mapa de manera tal, que, si se les aislara, perderían todo sentido y el mapa sería, en el mejor de los casos, menos útil para Bob.

3.3.2 Dimensiones, elementos y preguntas

Para construir núcleos funcionales, se proponen las preguntas del Cuadro 3.2.

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------|---|
| Técnicas de interacción | Extensibilidad | ¿Se extenderá el vocabulario? ¿Cambiará la sintaxis? ¿Cómo cambia la semántica? |
| Calidad | Nivel de control | ¿Cuál es el nivel de control dejado al usuario? |
| Funcionalidad | Servicios generales | ¿Cuándo rediseñar? ¿Cuándo redistribuir? |

Cuadro 3.2 Elementos a considerar para garantizar núcleos funcionales

La *extensibilidad* es un problema típico a considerar. Aplicado al dominio de las meta-interfaces, es posible que los usuarios puedan ampliar el *vocabulario* de la técnica de interacción, cambiar su *sintaxis*, y desde allí, extender su *semántica*, por tanto, crear nuevos servicios de meta-interfaz de usuario.

Por ejemplo, en el videojuego de Bob, es posible que el vocabulario cambie; si la interacción original en la PC era “hacer clic en el botón X”, cuando el mapa se pasa a la tableta cambia a “presionar el botón X”. La sintaxis no cambia, pero la semántica sí, “presionar” puede significar presionar un botón físico con un dedo, o bien, presionar un botón digital con un puntero, un stylus o un dedo.

El *nivel de control* se puede reducir simplemente a que el usuario puede migrar todo un núcleo, o no migrar nada. No se puede comprometer la funcionalidad de la aplicación.

Cuando Bob muda el mapa del videojuego a la tableta, lo mueve con todos sus elementos inherentes, no puede decidir por cada elemento individual.

En cuanto a *servicios generales*, la *redistribución* denota la reasignación de los componentes de la interfaz de usuario del espacio interactivo, a diferentes recursos de interacción. El *rediseño* de objetos, es remodelar objetos sin distorsionar su papel.

Al trasladar el mapa del juego a la tableta, Bob lo está *redistribuyendo*. Los elementos del mapa, al encontrarse ahora en un dispositivo táctil, posiblemente tengan que ser *rediseñados*, cambiar su forma o tamaño para adaptarse a la experiencia de usuario de un dispositivo háptico, con el objetivo de mantener su *funcionalidad*.

Se deben *redistribuir* núcleos funcionales enteros, pero se pueden *rediseñar* sus componentes internos.

3.4 Multimodalidad

La multimodalidad es la propiedad que tienen las meta-interfaces para transformar sus medios de interacción al cambiar de contexto. Por lo general, es deseable que sin importar los medios de entrada/salida, el usuario pueda llegar a un mismo resultado.

3.4.1 Descripción de escenario

Jenny es una piloto de la fuerza aérea. Tiene a su cargo un avión de caza con tecnología de punta. La cabina del jet contiene numerosos indicadores que le señalan a Jenny el estado de su nave. También cuenta con un casco con micrófono, auriculares y un visor que le entrega información adicional.

Durante el transcurso de una misión, Jenny debe estar pendiente de las condiciones del avión y además, según los objetivos de su cometido, debe interactuar con múltiples sistemas a la vez, sin descuidar las tareas esenciales que la mantienen volando.

Cabe destacar que, para reducir peso, Jenny no tiene mucho espacio en la cabina y sus formas de interacción con los sistemas son limitadas.

Al realizar ciertas maniobras, Jenny debe enfocarse en datos precisos mientras interactúa con ciertos sistemas y pilota el avión. Para ello el sistema se adapta de manera tal que, la información se transforma, por ejemplo, algunos datos que antes se desplegaban en pantalla, ahora son menos importantes y le son transmitidos a Jenny por medio de sus auriculares; las alertas que requieran atención inmediata, encontrarán lugar en el visor de su casco; o bien, la maquinaria que antes requería presión sobre ciertos botones ahora obedece comandos de voz de Jenny.

3.4.2 Dimensiones, elementos y preguntas

Se proponen las cuestiones en el Cuadro 3.3 para desarrollar escenarios de multimodalidad.

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Extensibilidad | ¿Se añade complejidad? |
| | Objeto/ Función/ Designación | ¿Se pierde funcionalidad? |
| Calidad | Nivel de control | ¿Se puede intervenir en la evolución del sistema? |
| Funcionalidad | Tipos de objetos | ¿Cuáles son las características de los elementos en el ambiente interactivo? |
| | Servicios generales | ¿Cómo se asignan los roles? |

Cuadro 3.3 Elementos a considerar para garantizar multimodalidad

En un ambiente interactivo la naturaleza de la *extensibilidad* parece inevitable, pero el riesgo de añadir *complejidad* a la interacción con el sistema puede ser alta.

En el caso particular de Jenny, cuando su voz se convierte en un componente de entrada, se añade un elemento de interacción más, tanto para ella como para el sistema. Cuando aumenta la complejidad, hay que tener cuidado en prever todos los escenarios que ello puede implicar.

La elaboración de una sentencia, que requiere *objetos* y *designación de funciones*, puede ser directa al actuar sobre los elementos del vocabulario o indirecta mediante el uso de instrumentos.

En el cambio de interacción anterior, es posible que se perdiera funcionalidad; quizás no sea posible expresar todos los comandos mediante instrucciones de voz. No obstante, es probable que Jenny no echara de menos algunas funciones, pues en el momento no las necesitaba todas. En ocasiones, se puede justificar la pérdida de funcionalidad.

A Jenny no se le permite *intervenir en la evolución del sistema*, esto se hace para reducir complejidad y asegurar que no se encuentre en situaciones que comprometan su seguridad.

Los *roles* que se asignan a los elementos del ambiente interactivo están estrechamente relacionados con sus *características*, por ejemplo, un micrófono sólo es hardware de entrada, así que, es sencillo asignarle un rol. Cuando hay dispositivos que pueden ser tanto de entrada como de salida, estos pueden tomar distintos roles dependiendo del contexto o del tiempo.

3.5 Limitaciones de usabilidad

Cuando existen escenarios de multimodalidad, es posible que se llegue a situaciones de usabilidad limitada. Cuando un medio de interacción cambia de contexto y se transforma, puede restringir la interacción del usuario con el sistema.

3.5.1 Descripción de escenario

Garp acaba de adquirir un reloj inteligente. Uno de los usos que descubre, es que puede recibir notificaciones por cada mensaje de texto entrante y cada nuevo correo electrónico recibido.

Cuando Garp recibe un correo electrónico, él puede ver en la pantalla del reloj el remitente y asunto del mensaje; pulsa la pantalla para abrirlo y así ver su contenido. Después de leer, Garp decide que es necesario responder el comunicado; el reloj le advierte que sólo cuenta con respuestas predefinidas como: “Sí”, “No”, “En camino”, etc. Y si quiere responder de una forma más personalizada tendrá que hacerlo desde su teléfono móvil.

Cuando Garp presiona desde su reloj la opción para contestar en su teléfono, éste último abre el mensaje correspondiente de manera que todo está listo para que Garp comience a redactar su respuesta.

3.5.2 Dimensiones, elementos y preguntas

Las preguntas en el Cuadro 3.4 tratan sobre limitaciones de usabilidad.

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Objeto/ Función/ Designación | ¿Las designaciones son homogéneas? |
| Calidad | Nivel de control | ¿El sistema responde según las capacidades del dispositivo? ¿Qué medidas se toman ante las limitaciones? ¿Se ofrecen alternativas? |
| Funcionalidad | Servicios generales | ¿Existen comportamientos predefinidos? |

Cuadro 3.4 Elementos a considerar para manejar las limitantes de usabilidad

Al reloj de Garp se le *asignaron* las *funciones* de leer y responder mensajes, que hasta cierto punto, son las mismas funciones que obtiene de su teléfono, por lo tanto, se tiene una designación homogénea. Cuando se distribuyen meta-interfaces con funciones específicas a dispositivos específicos, existe una designación heterogénea.

Cuando Garp desea responder el correo electrónico, el reloj le ofrece una serie de respuestas predefinidas, esto se debe a que sería muy difícil “teclear” todo un mensaje desde el propio reloj, es decir, se toman en cuenta las capacidades del dispositivo. Las respuestas predefinidas son un atenuante ante la falta de otro medio para responder.

En el momento que las limitaciones frenan la funcionalidad o la productividad del usuario, lo ideal sería ofrecer alguna *alternativa*, para hacerle saber al usuario que aún puede completar su tarea. A Garp se le ofrece, por ejemplo, que redacte su respuesta desde su teléfono móvil.

En el instante en que Garp presiona en su reloj la opción para responder en su teléfono, éste abre el mensaje correspondiente para que se comience a escribir. La transición entre el reloj y el teléfono es un *comportamiento predefinido* que ayuda a no romper el flujo de trabajo.

3.6 Trazabilidad

La trazabilidad denota la situación en la cual los usuarios pueden observar, y en algunos casos modificar la evolución del servicio de meta-interfaz a través del tiempo.

3.6.1 Descripción de escenario

Dax es un médico. En su consultorio cuenta con un novedoso sistema de análisis. Cuando un paciente llega a consulta, Dax le coloca en su muñeca una pulsera que cuenta con diversos sensores y una pantalla.

La computadora de Dax está sincronizada con la pulsera. Cuando se inicia el proceso de pruebas, la computadora le muestra a Dax la lista completa de los análisis que realizará, en qué estado se encuentra y el tiempo restante hasta que terminen. Mientras tanto, la pantalla de la pulsera refleja una serie de datos preliminares que le permiten observar el estado del paciente mientras Dax los cuestiona o ausculta.

Al terminar el proceso, Dax cuenta con datos suficientes para dar un primer diagnóstico.

3.6.2 Dimensiones, elementos y preguntas

Se debe dar respuesta a las incógnitas del Cuadro 3.5 para verificar los escenarios de trazabilidad.

| Dimensiones | Elementos | Preguntas |
|-------------------------|----------------------|---|
| Técnicas de interacción | Nivel de integración | ¿La evolución del ambiente interactivo se puede observar internamente o externamente? |
| Calidad | Iniciativa | ¿Se puede modificar la evolución del sistema? ¿Se puede observar la evolución del sistema? |
| Funcionalidad | Servicios generales | ¿Cómo se unen los dispositivos en el ambiente interactivo? |

Cuadro 3.5 Elementos a considerar para garantizar la trazabilidad

Si la *evolución del sistema* se puede *observar*, hay que considerar si es de forma *externa* o *interna*. Cuando Dax inicia con la batería de análisis, el sistema le reporta el avance de sus

tareas, por lo que Dax puede *observar* de manera *externa* los resultados, pero *no se observa* de manera *interna* la distribución que hubo de su computadora a la pulsera.

Dax tampoco tiene la posibilidad de controlar la *evolución del sistema*, simplemente lo puede iniciar o detener.

La comunicación entre la pulsera y la computadora de Dax es bidireccional, por lo que la *unión* es *directa*. Si existiera un dispositivo entre esa comunicación, entonces la *unión* sería *indirecta*.

3.7 Análisis de trabajos relacionados mediante las heurísticas propuestas

Se describieron ya las heurísticas de coherencia para meta-interfaces (ver Sección 3.1), ahora se aplicarán a los trabajos relacionados (ver Sección 2.2), con el fin de analizar con qué elementos de cada heurística cuenta cada desarrollo.

3.7.1 CamNote

CamNote [Demeure et al., 2005] es un visor de diapositivas que se ejecuta en una plataforma heterogénea dinámica, la cual puede ir desde una sola PC hasta un grupo compuesto por una computadora y un PDA (ver Sección 2.2.1).

Honestidad de meta-interfaces

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------------|---|
| Técnicas de interacción | Representación de objetos | <p>¿Cómo se representarán los elementos del espacio interactivo? - Ni el PDA ni el PC están representadas para el conocimiento del usuario</p> <p>¿Cómo se representarán las interfaces? - No existe representación de interfaces</p> |
| | Nivel de integración | <p>¿Dónde se distribuirán las interfaces? - La distribución es automática, no se puede controlar ni desde la PC ni desde el PDA</p> <p>¿Cómo se retroalimentará al usuario sobre las interfaces distribuídas? - No existe retroalimentación</p> |

| | | |
|---------------|---------------------|---|
| Calidad | Nivel de control | <p><i>¿Los elementos del espacio interactivo se comunicarán automáticamente?</i> - Sí, el usuario no interviene</p> <p><i>¿Cómo se informará al usuario sobre el estado de los elementos del espacio interactivo?</i> - No se le informa</p> <p><i>¿Cómo se dará acceso a la distribución de interfaces?</i> - El usuario no tiene acceso a la distribución</p> |
| Funcionalidad | Tipos de objetos | <p><i>¿Qué tipo(s) de elementos integran el espacio interactivo?</i></p> <p>- La aplicación <i>per se</i> es digital, mientras que el PDA y la PC son mezclados a través del diseño, ya que la fusión de componentes físicos y digitales ha sido realizada de antemano por los diseñadores de cada dispositivo</p> |
| | Servicios generales | <p><i>¿Se agregan/eliminan constantemente elementos al espacio interactivo?</i> - No, el único elemento con el que se puede interactuar es un PDA</p> <p><i>¿Cómo se agregan/eliminan elementos al espacio interactivo?</i> - Inhabilitando el PDA</p> <p><i>¿Las interfaces son dinámicas?</i> - No, están predefinidas</p> |

Cuadro 3.6 Elementos de honestidad en CamNote

Núcleos funcionales

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------|---|
| Técnicas de interacción | Extensibilidad | <p><i>¿Se extenderá el vocabulario?</i> - Sí, cambia según el uso en una PC o PDA, e.g., en una PC se hace “clic” con un ratón y en el PDA se “presiona” con un dedo</p> <p><i>¿Cambiará la sintaxis?</i> - No</p> <p><i>¿Cómo cambia la semántica?</i> - Sí, cambia según el uso en una PC o PDA, e.g., “presionar” puede significar presionar una tecla física o un botón digital</p> |
| Calidad | Nivel de control | <p><i>¿Cuál es el nivel de control dejado al usuario?</i> - El usuario sólo decide sobre la presencia del PDA</p> |

| | | |
|---------------|---------------------|---|
| Funcionalidad | Servicios generales | <p><i>¿Cuándo rediseñar?</i> - Toda la interfaz para el PDA está rediseñada, e.g., el <i>slider</i> para ajustar la transparencia del video del presentador, se adapta al diseño del sistema operativo del PDA</p> <p><i>¿Cuándo redistribuir?</i> - Todos los elementos de interacción se redistribuyen al PDA, a excepción de las notas del presentador</p> |
|---------------|---------------------|---|

Cuadro 3.7 Núcleos funcionales en CamNote

Multimodalidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Extensibilidad | <i>¿Se añade complejidad?</i> - Sí, el uso del PDA |
| | Objeto/ Función/ Designación | <i>¿Se pierde funcionalidad?</i> - Sí, no se pueden ver las notas del presentador desde el PDA |
| Calidad | Nivel de control | <i>¿Se puede intervenir en la evolución del sistema?</i> - No |
| Funcionalidad | Tipos de objetos | <i>¿Cuáles son las características de los elementos en el ambiente interactivo?</i> - El PDA es táctil, mientras que la PC se controla con un teclado y un mouse |
| | Servicios generales | <i>¿Cómo se asignan los roles?</i> - En automático y están preestablecidos |

Cuadro 3.8 Multimodalidad en CamNote

Limitaciones de usabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Objeto/ Función/ Designación | <i>¿Las designaciones son homogéneas?</i> - Sí |
| Calidad | Nivel de control | <i>¿El sistema responde según las capacidades del dispositivo?</i> - No, todo está predefinido |

| | | |
|---------------|---------------------|---|
| | | <p><i>¿Qué medidas se toman ante las limitaciones?</i> - Se adecuaron los componentes de interacción al PDA, e.g., el <i>slider</i> para ajustar la transparencia del video del presentador, se adapta al diseño del sistema operativo del PDA</p> <p><i>¿Se ofrecen alternativas?</i> - No</p> |
| Funcionalidad | Servicios generales | <p><i>¿Existen comportamientos predefinidos?</i> - Sí, la redistribución de la interfaz de usuario y la interacción PC - PDA</p> |

Cuadro 3.9 Limitantes de usabilidad en CamNote

Trazabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|----------------------|---|
| Técnicas de interacción | Nivel de integración | <i>¿La evolución del ambiente interactivo se puede observar internamente o externamente?</i> - Externamente, todos los cambios ocurren en los dispositivos |
| Calidad | Iniciativa | <p><i>¿Se puede modificar la evolución del sistema?</i> - No</p> <p><i>¿Se puede observar la evolución del sistema?</i> - Sí, el usuario puede observar cómo aparecen los controles en el PDA</p> |
| Funcionalidad | Servicios generales | <i>¿Cómo se unen los dispositivos en el ambiente interactivo?</i> - Automáticamente |

Cuadro 3.10 Trazabilidad en CamNote

3.7.2 Sitio Web Sedan-Bouillon

Sedan-Bouillon [Coutaz, 2006] es un sitio Web que tiene como objetivo promover el turismo en las regiones de Sedan (Francia) y Bouillon (Bélgica). Ofrece a los turistas información para visitar estas regiones e incluye una selección de hoteles, campamentos y restaurantes (ver Sección 2.2.2).

Honestidad de meta-interfaces

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------------|--|
| Técnicas de interacción | Representación de objetos | <p><i>¿Cómo se representarán los elementos del espacio interactivo?</i> - Mediante columnas en una tabla</p> <p><i>¿Cómo se representarán las interfaces?</i> - Mediante filas en una tabla</p> |
| | Nivel de integración | <p><i>¿Dónde se distribuirán las interfaces?</i> - Desde un menú específico</p> <p><i>¿Cómo se retroalimentará al usuario sobre las interfaces distribuídas?</i> - Mediante casillas de verificación en una tabla</p> |
| Calidad | Nivel de control | <p><i>¿Los elementos del espacio interactivo se comunicarán automáticamente?</i> - Sí, el usuario no interviene</p> <p><i>¿Cómo se informará al usuario sobre el estado de los elementos del espacio interactivo?</i> - Mediante una tabla</p> <p><i>¿Cómo se dará acceso a la distribución de interfaces?</i> - Con un menú específico</p> |
| Funcionalidad | Tipos de objetos | <p><i>¿Qué tipo(s) de elementos integran el espacio interactivo?</i></p> <p>- La aplicación <i>per se</i> es digital, mientras que el PDA y la PC son mezclados a través del diseño, ya que la fusión de componentes físicos y digitales ha sido realizada de antemano por los diseñadores de cada dispositivo</p> |
| | Servicios generales | <p><i>¿Se agregan/eliminan constantemente elementos al espacio interactivo?</i> - No, el único elemento con el que se puede interactuar es un PDA</p> <p><i>¿Cómo se agregan/eliminan elementos al espacio interactivo?</i> - Mediante casillas de verificación en una tabla</p> <p><i>¿Las interfaces son dinámicas?</i> - No, están predefinidas</p> |

Cuadro 3.11 Elementos de honestidad en Sedan-Bouillon

Núcleos funcionales

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------|---|
| Técnicas de interacción | Extensibilidad | <p><i>¿Se extenderá el vocabulario?</i> - Sí, cambia según el uso en una PC o PDA, e.g., en una PC se hace “clic” con un ratón y en el PDA se “presiona” con un dedo</p> <p><i>¿Cambiará la sintaxis?</i> - No</p> <p><i>¿Cómo cambia la semántica?</i> - Sí, cambia según el uso en una PC o PDA, e.g., “presionar” puede significar presionar una tecla física o un botón digital</p> |
| Calidad | Nivel de control | <i>¿Cuál es el nivel de control dejado al usuario?</i> - El usuario puede decidir qué elementos de interacción se muestran en el PDA o en la PC |
| Funcionalidad | Servicios generales | <p><i>¿Cuándo rediseñar?</i> - Las interfaces están predefinidas</p> <p><i>¿Cuándo redistribuir?</i> - El usuario decide qué elementos de interacción redistribuye y cuándo los redistribuye</p> |

Cuadro 3.12 Núcleos funcionales en Sedan-Bouillon

Multimodalidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Extensibilidad | <i>¿Se añade complejidad?</i> - Sí, el uso del PDA |
| | Objeto/ Función/ Designación | <i>¿Se pierde funcionalidad?</i> - Sí, si el PDA contine interfaces de usuario y si después éste se inhabilita |
| Calidad | Nivel de control | <i>¿Se puede intervenir en la evolución del sistema?</i> - Sí |
| Funcionalidad | Tipos de objetos | <i>¿Cuáles son las características de los elementos en el ambiente interactivo?</i> - El PDA es táctil, mientras que la PC se controla con un teclado y un mouse |
| | Servicios generales | <i>¿Cómo se asignan los roles?</i> - Mediante un menú específico y están preestablecidos |

Cuadro 3.13 Multimodalidad en Sedan-Bouillon

Limitaciones de usabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|---|
| Técnicas de interacción | Objeto/ Función/ Designación | <i>¿Las designaciones son homogéneas?</i> - Sí |
| Calidad | Nivel de control | <i>¿El sistema responde según las capacidades del dispositivo?</i> - No, todo está predefinido <i>¿Qué medidas se toman ante las limitaciones?</i> - Se adecuaron los componentes de interacción al PDA <i>¿Se ofrecen alternativas?</i> - No |
| Funcionalidad | Servicios generales | <i>¿Existen comportamientos predefinidos?</i> - Sí, la redistribución de la interfaz de usuario y la interacción PC - PDA |

Cuadro 3.14 Limitantes de usabilidad en Sedan-Bouillon

Trazabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|----------------------|--|
| Técnicas de interacción | Nivel de integración | <i>¿La evolución del ambiente interactivo se puede observar internamente o externamente?</i> - Ambas, internamente por la liga de meta-interfaz en el sitio y externamente mediante la tabla de componentes de interacción |
| Calidad | Iniciativa | <i>¿Se puede modificar la evolución del sistema?</i> - Sí <i>¿Se puede observar la evolución del sistema?</i> - Sí |
| Funcionalidad | Servicios generales | <i>¿Cómo se unen los dispositivos en el ambiente interactivo?</i> - El usuario inicia una sesión en cada dispositivo |

Cuadro 3.15 Trazabilidad en Sedan-Bouillon

3.7.3 Photo-Browser

Photo-Browser [Coutaz, 2010] soporta la navegación de fotos de una manera centralizada o distribuida en función de la disponibilidad de un conjunto dinámico de dispositivos heterogéneos.

Estos incluyen una mesa Diamond Touch interactiva, una pared y un teléfono inteligente (iPhone) con Windows, MacOS X y Android (ver Sección 2.2.3).

Honestidad de meta-interfaces

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------------|---|
| Técnicas de interacción | Representación de objetos | <p><i>¿Cómo se representarán los elementos del espacio interactivo?</i> - Los objetos no tienen representación</p> <p><i>¿Cómo se representarán las interfaces?</i> - No existe representación de interfaces</p> |
| | Nivel de integración | <p><i>¿Dónde se distribuirán las interfaces?</i> - La distribución está predefinida</p> <p><i>¿Cómo se retroalimentará al usuario sobre las interfaces distribuidas?</i> - No existe retroalimentación</p> |
| Calidad | Nivel de control | <p><i>¿Los elementos del espacio interactivo se comunicarán automáticamente?</i> - Sí, el usuario no interviene</p> <p><i>¿Cómo se informará al usuario sobre el estado de los elementos del espacio interactivo?</i> - No se le informa</p> <p><i>¿Cómo se dará acceso a la distribución de interfaces?</i> - El usuario no tiene acceso a la distribución</p> |
| Funcionalidad | Tipos de objetos | <p><i>¿Qué tipo(s) de elementos integran el espacio interactivo?</i></p> <p>- La aplicación <i>per se</i> es digital, mientras que el teléfono, la mesa interactiva y la PC son mezclados a través del diseño</p> |
| | Servicios generales | <p><i>¿Se agregan/eliminan constantemente elementos al espacio interactivo?</i> - No, el único elemento que se puede agregar es un teléfono</p> <p><i>¿Cómo se agregan/eliminan elementos al espacio interactivo?</i> - Quitando y poniendo el teléfono móvil sobre la mesa ineteractiva</p> <p><i>¿Las interfaces son dinámicas?</i> - No</p> |

Cuadro 3.16 Elementos de honestidad en Photo-Browser

Núcleos funcionales

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------|--|
| Técnicas de interacción | Extensibilidad | <p><i>¿Se extenderá el vocabulario?</i> - Sí, cambia según el uso en una PC, mesa interactiva o teléfono, e.g., en una PC se hace “clic” con un ratón y en el PDA y en la mesa interactiva se “presiona” con un dedo</p> <p><i>¿Cambiará la sintaxis?</i> - No</p> <p><i>¿Cómo cambia la semántica?</i> - Sí, cambia según el uso en una PC, mesa interactiva o teléfono, e.g., “presionar” puede significar presionar una tecla física o un botón digital</p> |
| Calidad | Nivel de control | <i>¿Cuál es el nivel de control dejado al usuario?</i> - El usuario puede incluir o eliminar el teléfono como recurso de interacción |
| Funcionalidad | Servicios generales | <p><i>¿Cuándo rediseñar?</i> - Las interfaces de usuario están predefinidas</p> <p><i>¿Cuándo redistribuir?</i> - Las interfaces de usuario están predefinidas</p> |

Cuadro 3.17 Núcleos funcionales en Photo-Browser

Multimodalidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Extensibilidad | <i>¿Se añade complejidad?</i> - No, los comportamientos están predefinidos |
| | Objeto/ Función/ Designación | <i>¿Se pierde funcionalidad?</i> - No |
| Calidad | Nivel de control | <i>¿Se puede intervenir en la evolución del sistema?</i> - No |
| Funcionalidad | Tipos de objetos | <i>¿Cuáles son las características de los elementos en el ambiente interactivo?</i> - El teléfono y la mesa interactiva son táctiles, mientras que la PC se controla con un teclado y un mouse |
| | Servicios generales | <i>¿Cómo se asignan los roles?</i> - Están preestablecidos |

Cuadro 3.18 Multimodalidad en Photo-Browser

Limitaciones de usabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Objeto/ Función/ Designación | ¿Las designaciones son homogéneas? - No se puede determinar, cada interfaz de usuario está desarrollada <i>ad hoc</i> |
| Calidad | Nivel de control | ¿El sistema responde según las capacidades del dispositivo? - No, todo está predefinido ¿Qué medidas se toman ante las limitaciones? - El ambiente interactivo está controlado y delimitado ¿Se ofrecen alternativas? - No |
| Funcionalidad | Servicios generales | ¿Existen comportamientos predefinidos? - Sí, e.g., el teléfono se tiene que poner sobre la mesa interactiva para acoplarlo |

Cuadro 3.19 Limitantes de usabilidad en Photo-Browser

Trazabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|----------------------|--|
| Técnicas de interacción | Nivel de integración | ¿La evolución del ambiente interactivo se puede observar internamente o externamente? - No se puede observar |
| Calidad | Iniciativa | ¿Se puede modificar la evolución del sistema? - No ¿Se puede observar la evolución del sistema? - No |
| Funcionalidad | Servicios generales | ¿Cómo se unen los dispositivos en el ambiente interactivo? - De manera predefinida |

Cuadro 3.20 Trazabilidad en Photo-Browser

Capítulo 4

Análisis y diseño de la aplicación de prueba

Aquí se describe el objetivo y justificación de la aplicación prototipo realizada. Todos los elementos de la interfaz gráfica siguen lineamientos y recomendaciones que se detallan en ésta sección. Los esquemas más importantes para el funcionamiento están mostrados mediante diagramas de flujo.

4.1 Análisis

A lo largo del documento se describieron conceptos e ideas fundamentales para lograr el objetivo de la tesis, desarrollar heurísticas que permitan mantener la coherencia de las meta-interfaces de usuario. Con el motivo de desarrollar tales heurísticas, se construyó una aplicación prototipo que no sólo los integra, sino que tiene como propósito mostrar cómo los desarrolladores pueden integrar dichas heurísticas en sus futuros proyectos.

El primer paso fue pensar qué tipo de aplicación realizar, con qué características debía contar, tal que, cada elemento visual fuera fácilmente distinguible, de igual forma que su tratamiento, es decir, que cumpla con las heurísticas de usabilidad. Era muy importante que la aplicación fuera tan sencilla como fuera posible, de esa forma la carga cognitiva de su uso estaría en función de los mecanismos de meta-interfaz implementados.

Finalmente se decidió crear un editor de gráficos rasterizados elemental, con base en la aplicación Paint de Microsoft. Esto permitió tener varias herramientas que se pudieran distribuir en diversos dispositivos como meta-interfaces. Es también una manera clara de cómo diferentes

paquetes de software como Photoshop, Word, AutoCAD, etc. que cuentan con múltiples barras de herramientas, podrían evolucionar para soportar meta-interfaces.

La arquitectura de la aplicación prototipo se compone de cuatro elementos básicos (ver Figura 4.1):

- El editor de gráficos rasterizados.
- Un meta-menú (como control para la meta-interfaz de usuario).
- La sincronización entre dispositivos.
- Los objetos de interfaz distribuibles entre dispositivos.

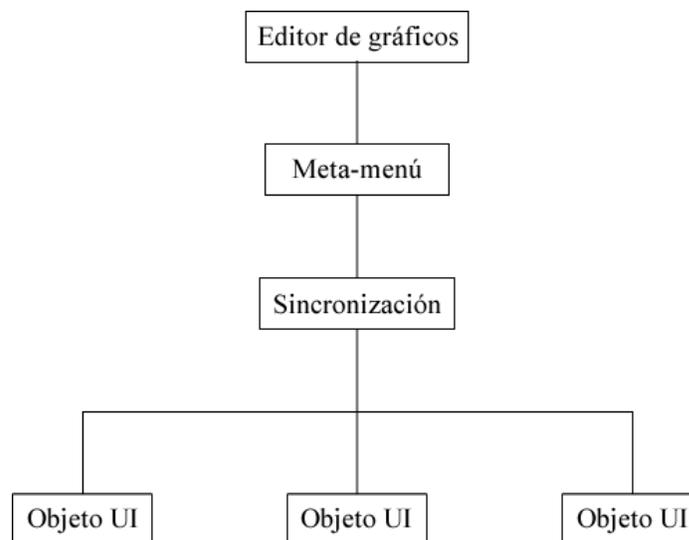


Figura 4.1 Arquitectura general del prototipo

El editor de gráficos rasterizados es la aplicación principal que se construyó, sirve de modelo para mostrar las heurísticas de cohercia desarrolladas. Contiene herramientas básicas para crear imágenes: pincel, círculo, elipse, borrador, línea y rectángulo, además se puede seleccionar el grosor y color del trazo. Todo el proyecto se realizó con HTML5, CSS3 y JavaScript, como se describe en el Capítulo 5.

Se optó por el esquema de aplicación Web por dos razones principales:

- **Aplicación como servicio:** al tratarse de una aplicación Web, lo único que necesita el usuario es contar con conexión a Internet y un navegador Web. De esta manera se eliminan requisitos previos y se ahorra cualquier tipo de instalación.

- **Diseño adaptable:** es una filosofía de diseño y desarrollo cuyo objetivo es adaptar la apariencia de las páginas Web al dispositivo que se esté utilizando para visualizarla. Por lo tanto, la aplicación no está intrínsecamente ligada a ninguna plataforma.

4.2 Diseño

A continuación se detallan las decisiones de diseño de la aplicación. Vista de escritorio, sincronización entre módulos, interfaces en dispositivos móviles, presencia y *layout*, son los elementos críticos en el prototipo, de manera que se describen el cómo y por qué de su diseño. Finalmente todo se valida aplicando las heurísticas de coherencia.

4.2.1 Vista de escritorio

La base del prototipo está pensada como una aplicación de PC convencional. Ésta cuenta con las funciones de dibujo descritas en el Cuadro 4.1.

| Función | Elemento gráfico | Valor por omisión |
|-------------------|--|-------------------|
| Pincel |  | Seleccionado |
| Círculo |  | - |
| Elipse |  | - |
| Borrador |  | - |
| Segmento de recta |  | - |
| Rectángulo |  | - |
| Color del trazo |  | Negro |
| Grosor del trazo |  | 5 |

Cuadro 4.1 Herramientas de dibujo

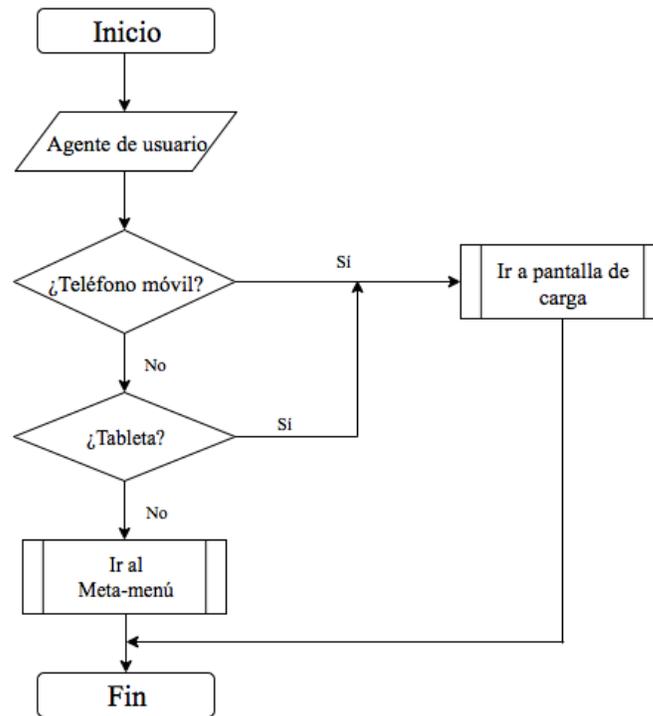


Figura 4.2 Diagrama de flujo para el botón “Go Meta!”

La interacción con todas las herramientas se realizan mediante el ratón o cualquier otro dispositivo señalador que reemplace las funciones del mismo. También se contempla la posibilidad para eventos de toque, es decir, compatibilidad con pantallas táctiles.

El único elemento gráfico implementado, es la barra de herramientas (ver Figura 4.3) que además de las opciones que ya se mencionaron, cuenta con un botón “Go Meta!”, éste tendrá una respuesta diferente al ser pulsado dependiendo si la aplicación se visualiza desde una PC o un dispositivo móvil.

El comportamiento antes señalado sigue el diagrama de flujo de la Figura 4.2.

4.2.2 Layout

El esquema de distribución de elementos, conocido principalmente como *layout*, es el tamaño, espaciado y colocación de contenidos dentro de una ventana o página. Un *layout* eficaz es crucial para ayudar a los usuarios a encontrar lo que están buscando rápidamente, así como para hacer el aspecto más atractivo a la vista. Un *layout* eficaz puede hacer la diferencia entre los

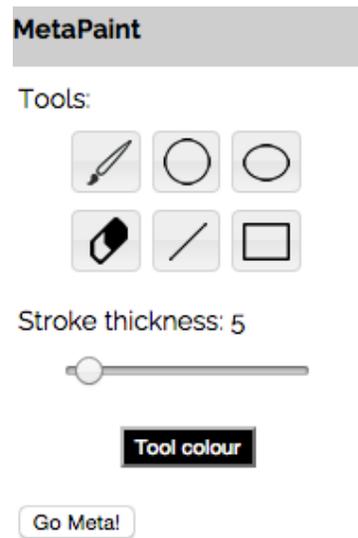


Figura 4.3 Barra de herramientas en el editor gráfico

diseños que los usuarios entienden de inmediato y los que dejan a los usuarios desconcertados y abrumados [Microsoft, 2015].

En la barra de herramientas (ver Figura 4.3) se aplicaron las heurísticas de usabilidad. La barra de herramientas aparece en la esquina superior izquierda de la pantalla y proyecta una sombra sobre el lienzo. La primera característica es por jerarquía visual y la segunda indica que la barra puede ser desplazada por el usuario en cualquier parte dentro del lienzo.

Una ventana o página tiene una jerarquía visual clara cuando su aspecto indica la relación y la prioridad de sus elementos. Sin una jerarquía visual, los usuarios tendrían que averiguar estas relaciones y prioridades por sí mismos [Microsoft, 2015].

La jerarquía visual se consigue combinando hábilmente los siguientes atributos [Microsoft, 2015]:

- **Atención:** la disposición indica a los usuarios qué necesitan mirar primero.
- **Flujo:** la vista fluye sin problemas y de forma natural por un camino claro a través de la superficie, encontrando elementos de la interfaz gráfica de usuario (GUI por sus siglas en inglés) en el orden adecuado para su uso.
- **Agrupación:** elementos de GUI relacionados lógicamente tienen una relación visual clara. Artículos relacionados se agrupan; artículos no relacionados se separan.

- **Énfasis:** elementos de la GUI se acentúan en función de su importancia relativa.
- **Alineación:** los elementos de la GUI tienen un lugar específico, por lo que son fáciles de escanear y parecerán ordenados.

Además, la disposición eficaz tiene estos atributos [Microsoft, 2015]:

- **La independencia de dispositivos:** el diseño aparece como se pretende, independientemente del tipo de letra o tamaño, puntos por pulgada (ppp), pantalla, o adaptador gráfico.
- **Fácil de escanear:** los usuarios pueden encontrar el contenido que están buscando de un vistazo.
- **Eficiencia:** los elementos de la GUI que necesitan ser grandes, son grandes y los que necesitan ser pequeños, son pequeños.
- **Redimensionar:** si es de ayuda, una ventana de tamaño variable y su distribución de contenido son efectivos no importa cuán grande o pequeña sea la superficie.
- **Balance:** el contenido aparece distribuido uniformemente a través de la superficie.
- **Simplicidad visual:** la percepción de que el diseño no es más complicado de lo que tiene que ser. Los usuarios no se sienten abrumados por el aspecto del diseño.
- **Consistencia:** ventanas o páginas similares utilizan un diseño similar, por lo que los usuarios siempre se sienten orientados.

Si bien el tamaño, espaciado y colocación son conceptos simples, el reto con la disposición es lograr la mezcla correcta de estos atributos.

Los usuarios eligen lo que leen por la apariencia del contenido y la organización. Para crear un diseño eficaz, se necesita entender lo que los usuarios tienden a leer y por qué.

Se pueden tomar decisiones de diseño usando el modelo de lectura mostrado en la Figura 4.4 [Microsoft, 2015]:

- La gente lee en un orden de izquierda a derecha, de arriba a abajo (en las culturas occidentales).

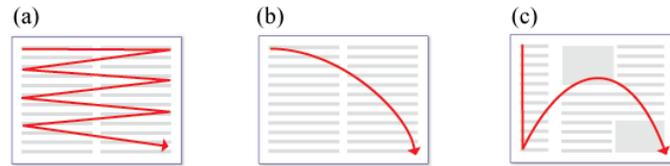


Figura 4.4 Modelo de lectura occidental. (a) Lectura inmersiva. (b) Ruta promedio de escaneo. (c) Texto en el borde izquierdo [Microsoft, 2015]

- Hay dos modos de lectura: lectura inmersiva y “escaneo”. El objetivo de la lectura inmersiva es la comprensión. Por el contrario, el objetivo del escaneo es localizar cosas. Si hay texto a lo largo del borde izquierdo de la página, los usuarios escanean el borde izquierdo primero.

4.2.3 Sincronización entre dispositivos

Las herramientas con las que un usuario puede dibujar en la aplicación, se listaron con anterioridad (ver Sección 4.2.1). Las asignaciones de cada función en un momento dado se sincronizan entre todos los dispositivos para que siempre tengan el mismo valor.

Como se aprecia en la Figura 4.5, primero hay que asignar valores por defecto a cada función (ver Sección 4.2.1), de esa forma se evitan conflictos cuando el usuario comienza a utilizar la aplicación. Los valores se guardan en una base de datos para que el resto de los dispositivos puedan actualizar sus asignaciones locales.

Todos los elementos, de todas las interfaces, en todos los dispositivos, se mantienen en sincronía y se actualizan en tiempo real gracias a una base de datos. Lo anterior permite que el sistema interactivo se mantenga consistente (ver Figura 4.6).

4.2.4 Interfaces en dispositivos móviles

El sistema interactivo puede trabajar hasta con tres dispositivos de manera simultánea: una PC, un teléfono móvil y una tableta. Las interfaces que tanto el teléfono como la tableta pueden adoptar en un momento dado son “Tools” y “Colour”, que despliegan las funciones de dibujo y la paleta de colores respectivamente. Además, existe una interfaz única para la tableta, “Both”, que contiene todos los elementos de “Tools” y “Colour”. Todas las interfaces móviles tienen

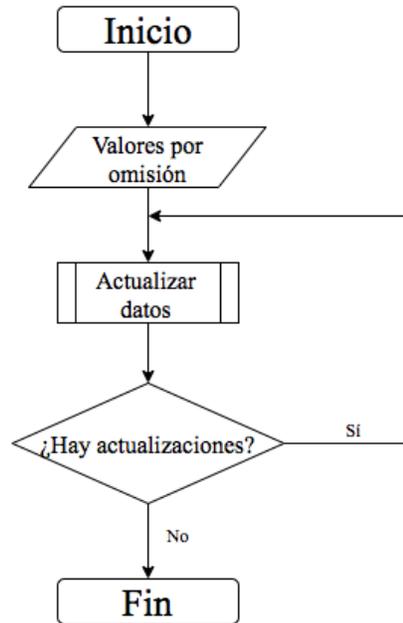


Figura 4.5 Sincronización de valores para las funciones de dibujo entre dispositivos

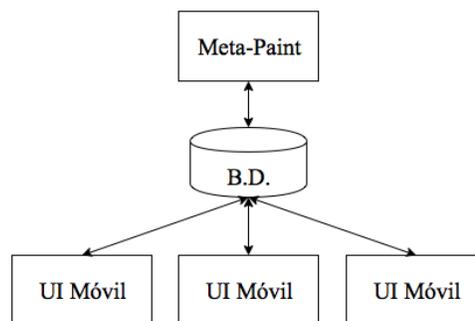


Figura 4.6 El nodo central que permite mantener la coherencia entre dispositivos es la base de datos

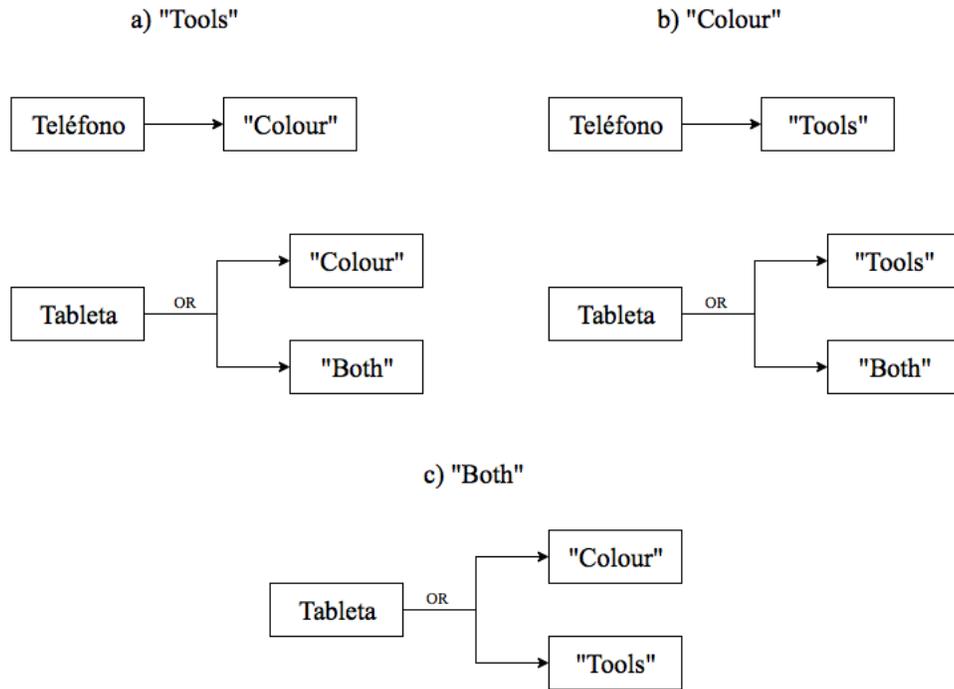


Figura 4.7 Control de interfaces móviles. Si en un momento dado el teléfono se encuentra en a) “Tools”, entonces el usuario sólo podrá desplegar “Colour”. De manera similar, si la tableta está en b) “Colour”, el usuario puede decidir cambiarla a “Tools”, o bien a “Both”

restricciones que dictan, despendiendo de dónde vienen, a dónde pueden ir, con el propósito de evitar redundancias y conflictos (ver Figura 4.7).

El meta-menú permite al usuario decidir cómo disponer las interfaces gráficas entre los dispositivos móviles. Como ya se mencionó, la asignación varía un poco en función de si se trata de un teléfono (ver Figura 4.8) o una tableta (ver Figura 4.9).

4.2.5 Presencia

Con el objetivo de crear una meta-interfaz honesta, se diseñó un sistema de presencia que informa al usuario sobre el estado de los dispositivos en el ambiente interactivo (ver Figura 4.10).

Al iniciar una nueva sesión de trabajo, el usuario encontrará a todos los dispositivos en un estado “Desconectado” (E1). Cuando se ingresa a la página de la aplicación principal, el sistema cambia el estado de los dispositivos a “Conectado” (E2), pero hasta este punto el usuario todavía

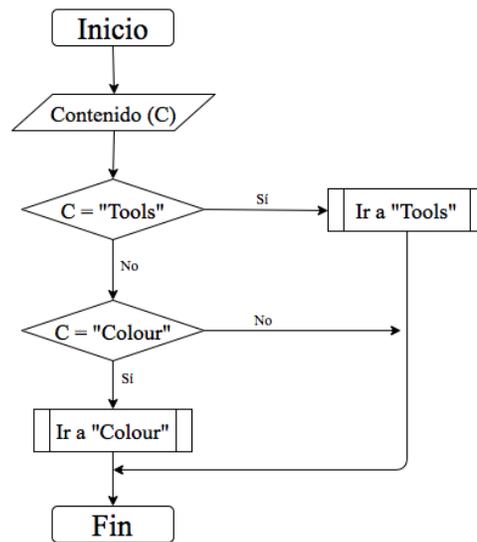


Figura 4.8 El meta-menú permite que se coloque en el teléfono las herramientas “Tools” o “Colour”

no puede distribuir interfaces. Es hasta que se presiona el botón “Go Meta!” en algún dispositivo para llevarlo a la pantalla de carga, que el sistema indica al usuario que los dispositivos están en estado “Conectado” (E3) y puede comenzar a distribuir elementos. Se permanecerá en este último estado hasta que el usuario cierre la página en el dispositivo.

4.2.6 Aplicación de las heurísticas de coherencia

Así como se analizaron los trabajos relacionados mediante las heurísticas de coherencia propuestas (ver Sección 3.7), a continuación se validan las decisiones de diseño de la aplicación prototipo desarrollada (Meta-Paint).

Honestidad de meta-interfaces

| Dimensiones | Elementos | Preguntas |
|-------------------------|---------------------------|---|
| Técnicas de interacción | Representación de objetos | ¿Cómo se representarán los elementos del espacio interactivo? - Mediante íconos ¿Cómo se representarán las interfaces? - Mediante íconos |
| | Nivel de integración | ¿Dónde se distribuirán las interfaces? - Desde un menú específico |

| | | |
|---------------|---------------------|---|
| | | <i>¿Cómo se retroalimentará al usuario sobre las interfaces distribuídas?</i> - Mediante íconos |
| Calidad | Nivel de control | <i>¿Los elementos del espacio interactivo se comunicarán automáticamente?</i> - Sí, el usuario no interviene <i>¿Cómo se informará al usuario sobre el estado de los elementos del espacio interactivo?</i> - Con un sistema de presencia <i>¿Cómo se dará acceso a la distribución de interfaces?</i> - Con un menú específico |
| Funcionalidad | Tipos de objetos | <i>¿Qué tipo(s) de elementos integran el espacio interactivo?</i> - La aplicación <i>per se</i> es digital, mientras que el teléfono, la tableta y la PC son mezclados a través del diseño |
| | Servicios generales | <i>¿Se agregan/eliminan constantemente elementos al espacio interactivo?</i> - Sí, el teléfono y la tableta son “plug and play” <i>¿Cómo se agregan/eliminan elementos al espacio interactivo?</i> - Se agregan mediante el botón “Go Meta!” y se eliminan cuando no hay conexión a Internet <i>¿Las interfaces son dinámicas?</i> - Sí |

Cuadro 4.2 Elementos de honestidad en MetaPaint

Núcleos funcionales

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------|--|
| Técnicas de interacción | Extensibilidad | <i>¿Se extenderá el vocabulario?</i> - Sí, cambia según el uso en una PC, tableta o teléfono, e.g., en una PC se hace “clic” con un ratón, mientras que en la tableta como en el teléfono se “presiona” con un dedo <i>¿Cambiará la sintaxis?</i> - No <i>¿Cómo cambia la semántica?</i> - Sí, cambia según el uso en una PC, tableta o teléfono, e.g., “presionar” puede significar presionar una tecla física o un botón digital |
| Calidad | Nivel de control | <i>¿Cuál es el nivel de control dejado al usuario?</i> - El usuario puede decidir qué elementos de interacción se muestran en el teléfono o en la tableta |

| | | |
|---------------|---------------------|--|
| Funcionalidad | Servicios generales | <p><i>¿Cuándo rediseñar?</i> - Se rediseñan los componentes de interacción para los dispositivos móviles, e.g., en una PC las herramientas de dibujo son botones con íconos, mientras que en los dispositivos móviles son botones con palabras</p> <p><i>¿Cuándo redistribuir?</i> - El usuario decide qué elementos de interacción redistribuye y cuándo los redistribuye</p> |
|---------------|---------------------|--|

Cuadro 4.3 Núcleos funcionales en MetaPaint

Multimodalidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|---|
| Técnicas de interacción | Extensibilidad | <i>¿Se añade complejidad?</i> - Sí, el uso del teléfono y la tableta |
| | Objeto/ Función/ Designación | <i>¿Se pierde funcionalidad?</i> - No |
| Calidad | Nivel de control | <i>¿Se puede intervenir en la evolución del sistema?</i> - Sí, el usuario elige los dispositivos con los cuales quiere interactuar |
| Funcionalidad | Tipos de objetos | <i>¿Cuáles son las características de los elementos en el ambiente interactivo?</i> - El teléfono y la tableta son táctiles, mientras que la PC se controla con un <i>mouse</i> |
| | Servicios generales | <i>¿Cómo se asignan los roles?</i> - Mediante un menú específico y están preestablecidos |

Cuadro 4.4 Multimodalidad en MetaPaint

Limitaciones de usabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|------------------------------------|--|
| Técnicas de interacción | Objeto/ Función/ Designación | <i>¿Las designaciones son homogéneas?</i> - Sí |

| | | |
|---------------|---------------------|---|
| Calidad | Nivel de control | <p><i>¿El sistema responde según las capacidades del dispositivo?</i> - Sí, las interfaces de usuario de los dispositivos móviles tienen programados eventos de toque</p> <p><i>¿Qué medidas se toman ante las limitaciones?</i> - Todas las interfaces de usuario siguen las heurísticas de usabilidad de Nielsen (ver Sección 2.1.3)</p> <p><i>¿Se ofrecen alternativas?</i> - Sí, diversas combinaciones entre interfaces de usuario y dispositivos móviles, e.g., el usuario puede interactuar sólo con la PC o con la PC y el teléfono</p> |
| Funcionalidad | Servicios generales | <p><i>¿Existen comportamientos predefinidos?</i> - Sí, la redistribución de las interfaces de usuario y la interacción PC - teléfono - tableta</p> |

Cuadro 4.5 Limitantes de usabilidad en MetaPaint

Trazabilidad

| Dimensiones | Elementos | Preguntas |
|-------------------------|----------------------|---|
| Técnicas de interacción | Nivel de integración | <p><i>¿La evolución del ambiente interactivo se puede observar internamente o externamente?</i> - Ambas, internamente por el botón “Go Meta!” en la aplicación en su vista de escritorio y externamente mediante el menú de meta-interfaz</p> |
| Calidad | Iniciativa | <p><i>¿Se puede modificar la evolución del sistema?</i> - Sí, mediante el “meta-menú”</p> <p><i>¿Se puede observar la evolución del sistema?</i> - Sí, mediante el “meta-menú”</p> |
| Funcionalidad | Servicios generales | <p><i>¿Cómo se unen los dispositivos en el ambiente interactivo?</i> - Mediante Internet</p> |

Cuadro 4.6 Trazabilidad en MetaPaint

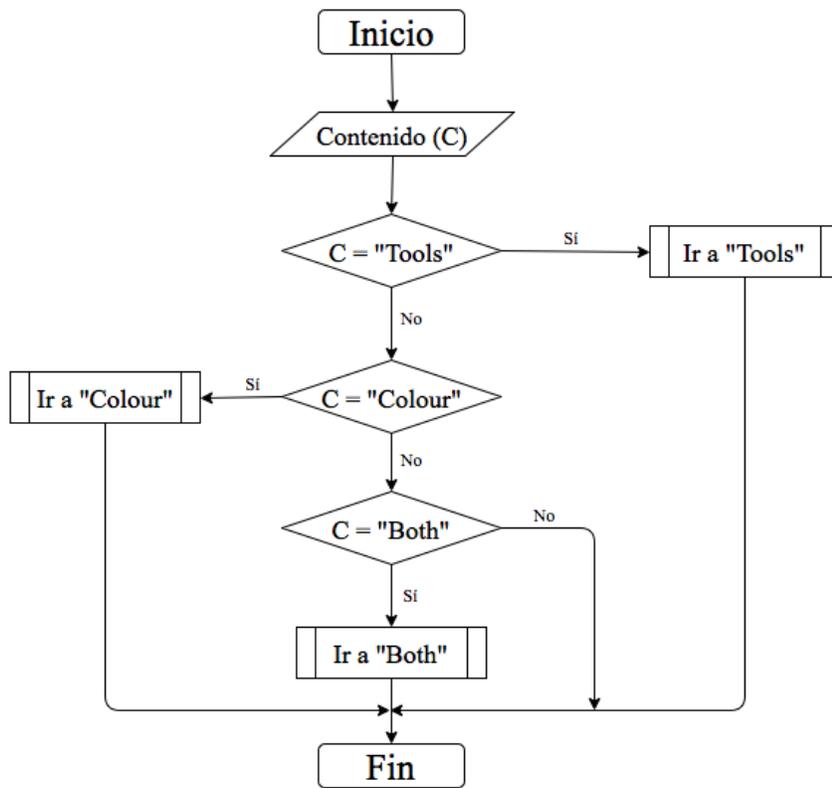


Figura 4.9 El meta-menú permite colocar en la tableta todas las herramientas

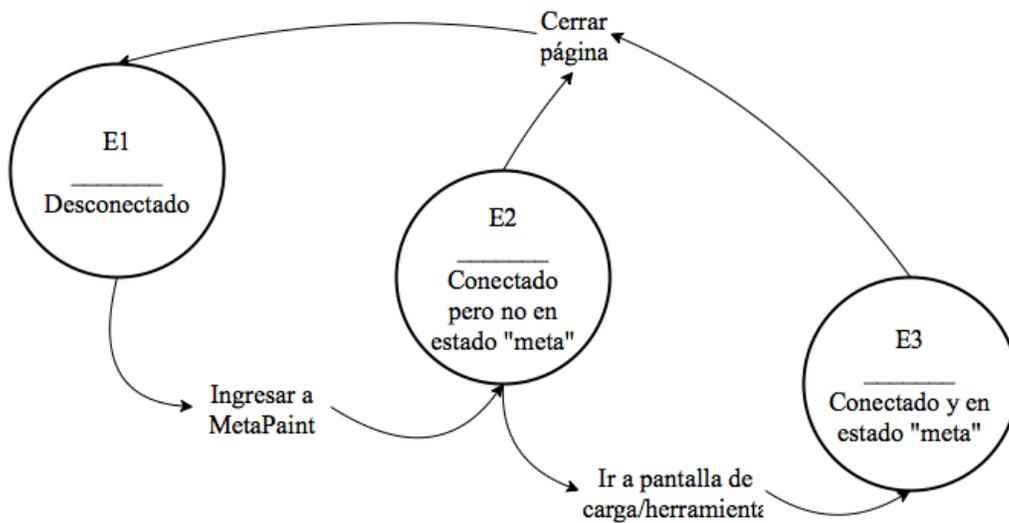


Figura 4.10 Estados en el sistema de presencia

Capítulo 5

Implementación de la aplicación de prueba

El presente capítulo tiene como preámbulo una breve descripción de las tecnologías usadas en el proyecto. La arquitectura de la aplicación prototipo se basa en tres módulos principales: la aplicación en modo escritorio, la versión móvil de las herramientas y el panel de distribución de la interfaz gráfica de usuario. En esta sección se detallará la creación de cada módulo. Al final del capítulo, se encuentran las pruebas con usuarios finales y sus resultados.

5.1 Tecnologías usadas

Se justificó con anterioridad (ver Sección 4.1) la naturaleza Web de la aplicación. HTML, CSS3 y JavaScript son tecnologías estándares para el desarrollo de contenidos Web, por lo que se seleccionaron para la construcción del prototipo. En Firebase recae la responsabilidad de la sincronización entre dispositivos.

5.1.1 HTML5 y CSS3

El lenguaje de marcado (o lenguaje de marcas) de la World Wide Web ha sido siempre HTML. Este lenguaje fue diseñado principalmente como un lenguaje para describir semánticamente documentos científicos, sin embargo, su diseño general y adaptaciones en los últimos años han permitido que se utilice para describir una serie de otros tipos de documentos. El área principal que no ha sido tratada adecuadamente por HTML es un tema denominado

Aplicaciones Web. La más reciente versión de HTML trata de corregir esta situación [MDN, 2015].

HTML5 es la quinta y más reciente especificación de HTML. El término representa dos conceptos diferentes:

- Se trata de una nueva versión del lenguaje HTML, con nuevos elementos, atributos y comportamientos.
- Contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y tener mayor alcance.

Las características principales de HTML5 que son importantes para el proyecto son:

- Fuera de línea y almacenamiento: permite a páginas Web almacenar datos, localmente, en el lado del cliente y operar fuera de línea de manera más eficiente.
- Gráficos y efectos 2D/3D: proporciona una amplia gama de nuevas características que se ocupan de los gráficos en la Web como lo son el lienzo 2D, WebGL, SVG, etc.
- Rendimiento e integración: proporciona una mayor optimización de la velocidad y un mejor uso del hardware.
- Acceso a dispositivos móviles: proporciona APIs para el uso de varios componentes internos de entrada y salida de numerosos dispositivos.
- CSS3: ofrece una nueva gran variedad de opciones para la sofisticación del diseño.

Cascading Style Sheets (CSS) es un lenguaje de hojas de estilo usado para describir el aspecto y el formato de un documento escrito en un lenguaje de marcas. Aunque más a menudo se utiliza para cambiar el estilo de las páginas Web y las interfaces de usuario escritas en HTML y XHTML, el lenguaje se puede aplicar a cualquier tipo de documento XML, incluyendo XML llano, SVG y XUL. Junto con HTML y JavaScript, CSS es una tecnología fundamental utilizada por la mayoría de los sitios Web para crear páginas Web visualmente atractivas, interfaces de usuario para aplicaciones Web, e interfaces de usuario para muchas aplicaciones móviles [MDN, 2015].

CSS3 es la última evolución del lenguaje de hojas de estilo en cascada y pretende ampliar CSS2.1. Incluye numerosas novedades, como esquinas redondeadas, sombras, gradientes, transiciones o animaciones, así como nuevos diseños como multi-columnas, cajas flexibles o diseños de cuadrícula.

5.1.2 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (*client-side*), implementado como parte de un navegador Web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas, aunque existe una forma de JavaScript del lado del servidor (*Server-side JavaScript* o *SSJS*). Su uso en aplicaciones externas a la Web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente *widjets*) es también significativo.

JavaScript se diseñó con una sintaxis similar a C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas Web. Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del *Document Object Model* (DOM) [Domínguez-Dorado, 2005].

5.1.3 Firebase

Firebase es un proveedor de servicios en la “nube” y *back-end* como servicio con sede en San Francisco, California. La compañía fabrica una serie de productos para desarrolladores de software que crean aplicaciones móviles o Web. El producto primario de Firebase es una base de datos en tiempo real que proporciona una API que permite a los desarrolladores almacenar y sincronizar los datos a través de múltiples clientes. La compañía fue adquirida por Google en octubre 2014.

Los datos en Firebase se almacena como JSON y son sincronizados en tiempo real a cada cliente conectado. Al generar aplicaciones multiplataforma (Android, iOS y JavaScript) todos los clientes comparten una base de datos Firebase y automáticamente reciben actualizaciones con los datos más recientes [Firebase, 2015].

5.2 Arquitectura de la aplicación

El prototipo que se desarrolló, en el presente trabajo de investigación, es un editor de gráficos, que cuenta con herramientas básicas para el dibujo y está basado en la aplicación de Microsoft “Paint”. Con la particularidad que se implementaron escenarios de meta-interfaz para

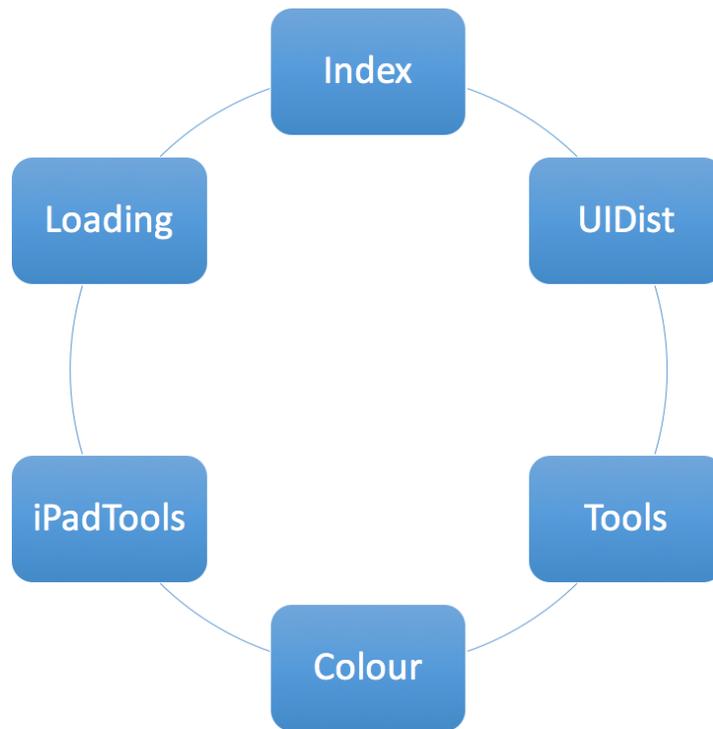


Figura 5.1 Bloques funcionales de la aplicación

que el usuario contara con la capacidad de distribuir diversos elementos de la interfaz gráfica. Tal distribución está pensada de manera tal que los elementos puedan ser visualizados en una tableta y un teléfono móvil.

La Figura 5.1 representa los archivos HTML que integran el prototipo. La versión de escritorio, que el usuario podrá utilizar en una PC, es el módulo principal y la aplicación *per se* (Index). Desde el modulo anterior el usuario puede ingresar al meta-menú (UIDist) o si se trata de un dispositivo móvil a la pantalla de carga (Loading), para después organizar la asignación de las herramientas en los dispositivos móviles (Tools, Colour, iPadTools).

5.2.1 Vista de escritorio

La base del prototipo está pensada como una aplicación de PC convencional. Se decidió que fuera una aplicación Web para tener más flexibilidad y no estar atado a una sola plataforma; en la misma línea, se eliminan procesos de instalación y configuración adicionales a los inherentes al uso de la aplicación.



Figura 5.2 Editor de gráficos en su versión de escritorio

La Figura 5.2 representa cómo un usuario puede visualizar la aplicación en un navegador Web de PC.

La interacción con todas las herramientas (ver Sección 4.2.1) se realizan mediante el ratón o cualquier otro dispositivo señalador que reemplace las funciones del mismo. También se contempla la posibilidad para eventos de toque, es decir, compatibilidad con pantallas táctiles.

La barra de herramientas (ver Figura 5.3) cuenta con un botón “Go Meta!”, el cual tendrá una respuesta diferente al ser pulsado dependiendo si la aplicación se visualiza desde una PC o un dispositivo móvil.

El comportamiento antes señalado está implementado en JavaScript y sigue el Pseudocódigo 1. Cuando una persona utiliza la aplicación, ésta última revisa su agente de usuario, que proporciona, entre otra información, el tipo de dispositivo que el usuario está empleando.

En el momento que el botón “Go Meta!” es presionado, la aplicación comprueba si se trata de un dispositivo móvil o no, cuando ocurre lo segundo, el botón guía hacia el menú de distribución de interfaces (meta-menú).

En el caso de que sí se trate de un dispositivo móvil, se restringieron los posibles escenarios a sólo dos: se pueden distribuir las interfaces a un iPhone o a un iPad.

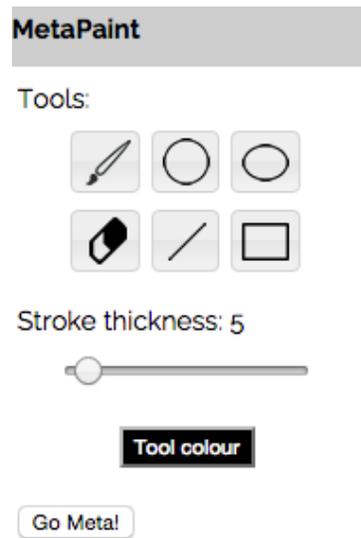


Figura 5.3 Barra de herramientas en el editor gráfico

En el caso de utilizar un iPhone existen dos alternativas: si es una nueva sesión entonces la variable contenido no tiene ninguna asignación y en el iPhone se mostraría una pantalla de carga hasta que el usuario elija una herramienta. Cuando se distribuye una interfaz, contenido guarda ese valor, así, si el usuario decidió poner la paleta de colores en el teléfono, contenido guardará "Colour", si por el contrario, las herramientas de dibujo fueron elegidas, contenido será igual a "Tools".

Cuando se está trabajando con un iPad, ocurren casos similares. Primero se comprueba si contenido encierra algún valor, de no ser así, se muestra la pantalla de carga (ver Figura 5.8). Al igual que el iPhone, la tableta puede desplegar la paleta de colores o las herramientas de dibujo; además, existe un caso adicional, que consiste en tener las dos al mismo tiempo; cuando el usuario realiza tal acción la variable contenido guarda "Both".

A la técnica de mostrar cierto contenido en función del agente de usuario, se le conoce como *browser sniffing*, sin embargo tiene ciertos inconvenientes. En ocasiones resulta en que navegadores poco conocidos reciben sólo parte del contenido original, aunque sean capaces de mostrar el contenido completo de manera correcta; en casos extremos se puede tener bloqueado el acceso al sitio.

No obstante, se decidió que éste era el comportamiento más simple y funcional. En otro caso es recomendable evitar la técnica de *browser sniffing* y decantarse por soluciones más robustas que no comprometan la experiencia de usuario. El problema es que en la actualidad

Pseudocódigo 1 *Browser sniffing* para el botón “Go Meta!”

Require: Agente de usuario (ua), contenido

- 1: **if** ua = dispositivo móvil **then**
- 2: **if** au = iPhone **then**
- 3: **if** contenido ≠ “Tools” **and** contenido ≠ “Colour” **then**
- 4: Ir a pantalla de carga
- 5: **else**
- 6: **if** contenido = “Colour” **then**
- 7: Ir a la paleta de colores
- 8: **else if** contenido = “Tools” **then**
- 9: Ir a las herramientas de dibujo
- 10: **else if** au = iPad **then**
- 11: **if** contenido ≠ “Tools” **and** contenido ≠ “Colour” **and** contenido ≠ “Both” **then**
- 12: Ir a pantalla de carga
- 13: **else**
- 14: **if** contenido = “Colour” **then**
- 15: Ir a la paleta de colores
- 16: **else if** contenido = “Tools” **then**
- 17: Ir a las herramientas de dibujo
- 18: **else if** contenido = “Both” **then**
- 19: Ir a las herramientas de iPad
- 20: **else**
- 21: Ir al Meta-Menú

es cada vez más difícil encontrar diferencias entre PC y dispositivos móviles, ya sea por el aumento de capacidades de los últimos o por la creación de híbridos que a momentos cambian por completo la experiencia de uso.

El último elemento que resalta en este apartado, es la sincronización de herramientas entre dispositivos, lo cual se logra mediante Firebase. Al iniciar la aplicación se establecen valores por omisión para inicializar las variables que controlan la herramienta de dibujo seleccionada y el color y grosor del trazo (ver Figura 5.4)

Para utilizar la API de Firebase es necesario agregar su CDN (*Content Delivery Network*) en la cabecera de cada documento HTML que haga uso de algún método:

```
<script src="https://cdn.firebase.com/js/client/2.2.7/firebase.js"></script>
```

Para leer y escribir datos desde y hacia una base de datos Firebase, se tiene que crear primero una referencia a esa base de datos. Para hacer esto el constructor Firebase recibe como

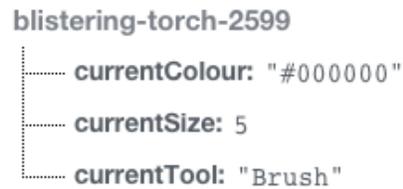


Figura 5.4 Árbol JSON en Firebase. La variable `currentColour` mantiene un valor hexadecimal para el color del trazo, `currentSize` guarda un entero entre 1 y 100 para el grosor del trazo y `currentTool` almacena una cadena que describe la herramienta de dibujo seleccionada

parámetro la URL que es proporcionada con anterioridad para cada proyecto en específico (Pseudocódigo 2).

Pseudocódigo 2 Creación de una referencia y objetos “hijo” para una base de datos en Firebase

Require: URL Firebase (uf)

- 1: `tools` ← nueva referencia `Firestore(uf)`
 - 2: `currentTool` ← crear objeto hijo de `tools`
 - 3: `currentColour` ← crear objeto hijo de `tools`
 - 4: `currentSize` ← crear objeto hijo de `tools`
-

Se crean objetos “hijo” de la referencia de la base de datos, los cuales serán los que almacenarán las variables que mantienen en sincronización la asignación de las herramientas. Y se les da un valor inicial para guardarlos en la base de datos referenciada, en sustitución de los datos existentes en ese camino. Cuando un objeto JSON se guarda en la base de datos, las propiedades de los objetos se asignan automáticamente a los lugares de los objetos “hijo” de una forma anidada (Pseudocódigo 3).

Pseudocódigo 3 Inicialización de herramientas

- 1: `currentTool` ← “Brush”
 - 2: `currentSize` ← 5
 - 3: `currentColour` ← “000000”
-

Finalmente para leer los datos se crea una función, por ejemplo, para recuperar el color del trazo (Pseudocódigo 4). Esta función será llamada en cualquier momento en que nuevos datos se añadan a la base de datos referenciada, y no es necesario escribir ningún código extra para hacer que esto ocurra.

Pseudocódigo 4 Obtención del color del trazo

- 1: actualizar valor `currentColour`
 - 2: valor ← `val.(DataSnapshot)`
-

La función de devolución de llamada recibe `DataSnapshot`, que es una instantánea de los datos. Una instantánea es una imagen de los datos a una referencia de base de datos concreta en un solo punto en el tiempo. Utilizando el método `val()` en una instantánea, devuelve la representación de objetos JavaScript de los datos. Si no existen datos en la ubicación de la referencia, el valor de las instantáneas será nulo.

Hay que tener en cuenta que se utilizó el tipo de evento `value` en el ejemplo anterior, que lee todo el contenido de una referencia de base de datos Firebase, aunque sólo un dato haya cambiado.

5.2.2 Meta-menú

Tal como se explicó en el Pseudocódigo 1, al recibir el agente de usuario, se puede conocer si el dispositivo que está usando un usuario se trata de una PC o un dispositivo móvil. Cuando ocurre el primer caso, al oprimir el botón “Go Meta!”, el usuario es direccionado al menú de distribución de la interfaz gráfica o meta-menú.

Tal como se ve en la Figura 5.5, la parte superior de la pantalla contiene tres elementos que representan la barra de herramientas dividida en sus núcleos funcionales, esto es, se dividió de forma tal que cada elemento es independiente del otro sin perder usabilidad (ver Figura 5.6)

La sección baja de la pantalla muestra dos casillas que representan a los dispositivos con los que el sistema puede interactuar. Gracias a Firebase, el usuario no requiere ninguna configuración previa ni que los tres dispositivos (PC, teléfono y tablet) estén conectados a una misma red ni se encuentren en un mismo lugar. El único requisito es que todos los aparatos electrónicos tengan acceso a Internet.

Cada núcleo funcional puede ser arrastrado a una casilla de la parte inferior, de manera tal que, el elemento arrastrado será desplegado en el dispositivo hasta el que se le arrastró; cuando esto sucede, el usuario obtiene retroalimentación mediante un ícono que aparece en el dispositivo (ver Figura 5.7)

El teléfono puede desplegar dos de los tres elementos: “Tools” o “Colours”. Mientras que la tableta puede aceptar los tres. Esto debido a cuestiones funcionales y de experiencia de usuario, además de que se agrega variedad a la distribución de la interfaz. Cabe aclarar que cada

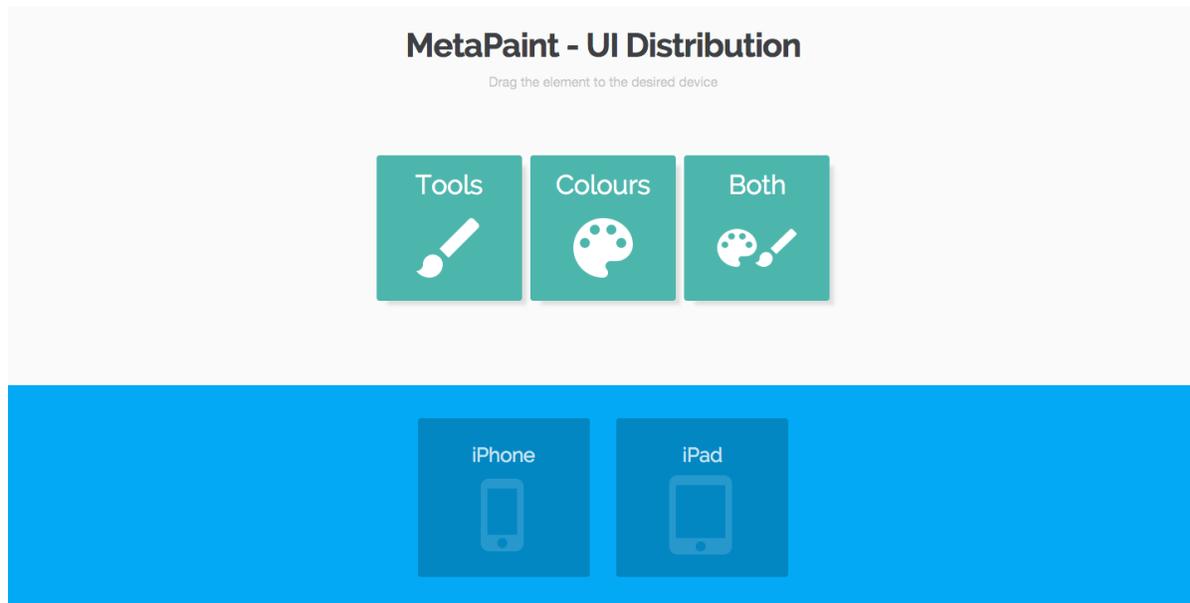


Figura 5.5 Meta-menú para la distribución de interfaces

elemento es excluyente, es decir, cada dispositivo puede desplegar un núcleo funcional por vez. Lo anterior tiene como propósito reducir el nivel de carga cognitiva de los usuarios.

De manera similar al manejo de herramientas en Firebase, se realiza la sincronización de la distribución. Primero creando una referencia a la base de datos y después creando objetos “hijo”, sólo que esta vez corresponden a los dispositivos (ver Pseudocódigo 5).

Pseudocódigo 5 Creación de una referencia y objetos “hijo” para una base de datos en Firebase

Require: URL Firebase (uf)

- 1: uiDist ← nueva referencia Firebase(uf)
 - 2: varPhone ← crear objeto hijo de uiDist
 - 3: varPad ← crear objeto hijo de uiDist
-

Para desplegar alguno de los dos núcleos funcionales en el teléfono, cuando el usuario termina de arrastrar un elemento a la casilla del iPhone, se verifica a cuál herramienta corresponde (Pseudocódigo 6).

Para mantener las selecciones del usuario actualizadas, cada herramienta comprueba en tiempo real la distribución de los núcleos funcionales. Por ejemplo, si el iPhone está actualmente mostrando “Tools”, quiere decir que el usuario sólo tiene una alternativa: ir a la paleta de colores (Pseudocódigo 7). De manera similar, si el iPad se encuentra en la paleta de colores (“Colour”),

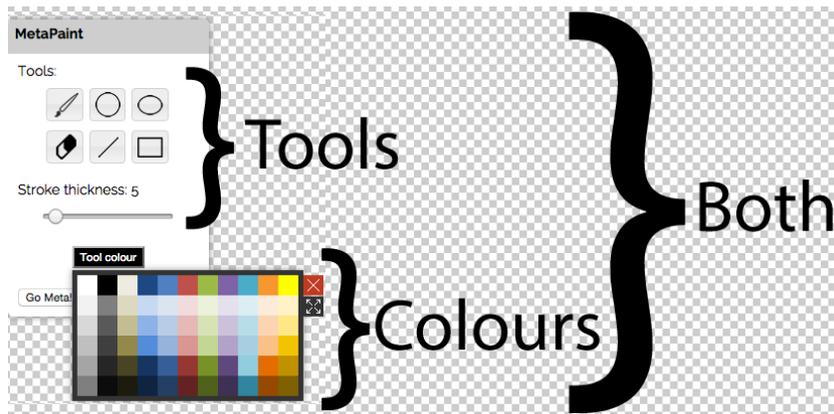


Figura 5.6 División por núcleos funcionales de la barra de herramientas

Pseudocódigo 6 Selección de núcleos funcionales para iPhone

Require: Elemento arrastrado (nf)

- 1: **if** nf = “tools” **then**
 - 2: varPhone ← Ir a “Tools”
 - 3: **else if** nf = “colours” **then**
 - 4: varPhone ← Ir a “Colour”
-

el usuario puede mostrar “Tools”, o bien, todas las herramientas (Pseudocódigo 8). Para los núcleos funcionales en el iPad, se puede mostrar “Tools” o “Colour” (Pseudocódigo 9).

Pseudocódigo 7 Navegación de núcleos funcionales en “Tools”

Require: Agente de usuario (ua), contenido

- 1: **if** ua = iPhone **then**
 - 2: **if** contenido = “Colour” **then**
 - 3: Ir a la paleta de colores
 - 4: **else if** ua = iPad **then**
 - 5: **if** contenido = “Colour” **then**
 - 6: Ir a la paleta de colores
 - 7: **else if** contenido = “Both” **then**
 - 8: Ir a las herramientas de iPad
-

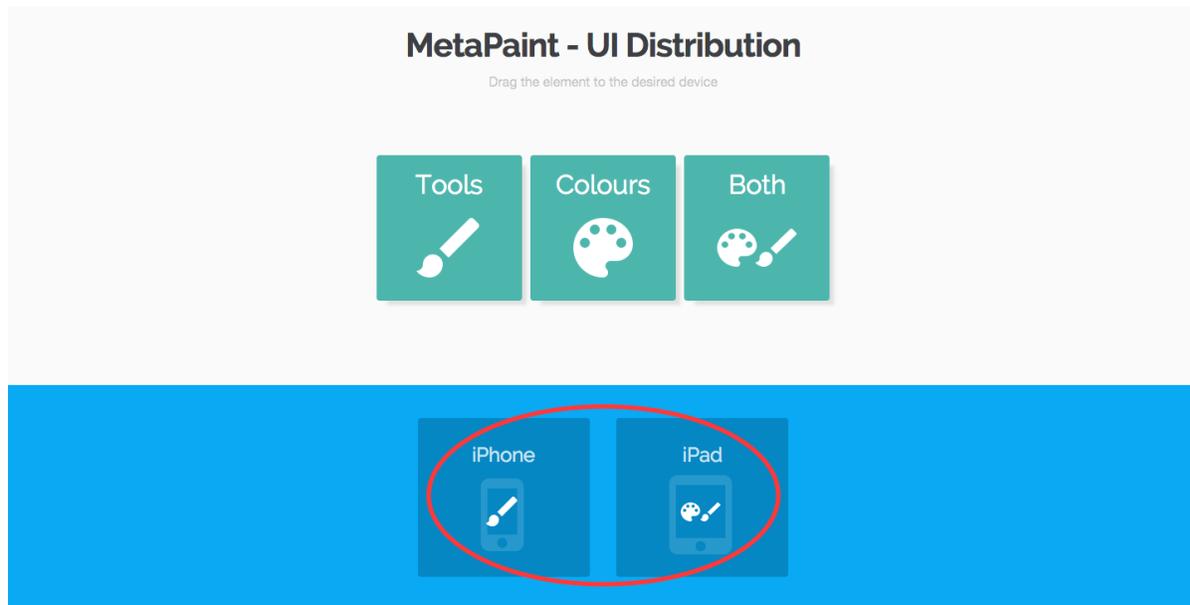


Figura 5.7 Retroalimentación gráfica. El usuario puede notar que arrastró el elemento “Tools” al teléfono, mientras que la tableta despliega “Both”

Pseudocódigo 8 Navegación de núcleos funcionales en “Colour”

Require: Agente de usuario (ua), contenido

- 1: **if** au = iPhone **then**
 - 2: **if** contenido = “Tools” **then**
 - 3: Ir a las herramientas de dibujo
 - 4: **else if** au = iPad **then**
 - 5: **if** contenido = “Tools” **then**
 - 6: Ir a las herramientas de dibujo
 - 7: **else if** contenido = “Both” **then**
 - 8: Ir a las herramientas de iPad
-

Pseudocódigo 9 Navegación de núcleos funcionales en “iPadTools”

Require: Agente de usuario (ua), contenido

- 1: **if** contenido = “Tools” **then**
 - 2: Ir a las herramientas de dibujo
 - 3: **else if** contenido = “Colour” **then**
 - 4: Ir a la paleta de colores
-

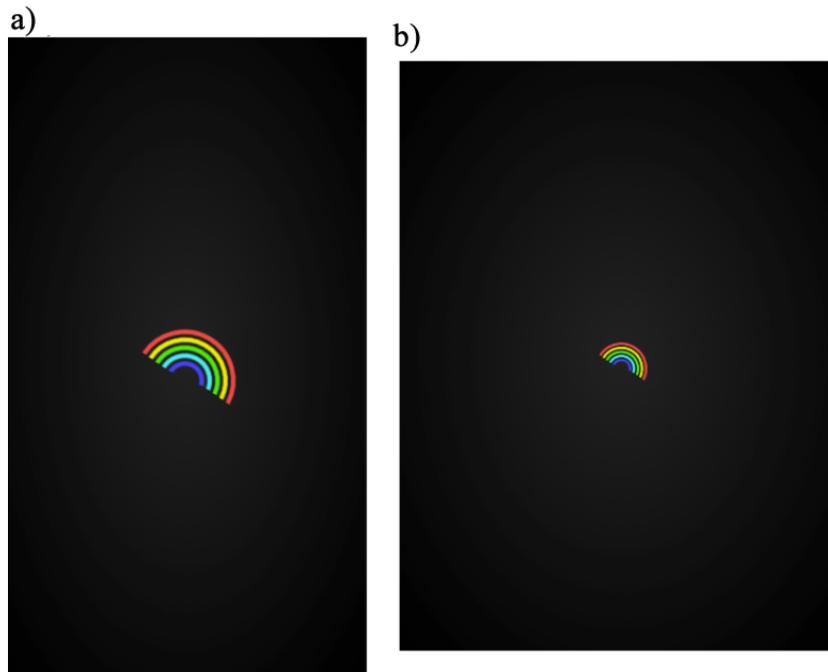


Figura 5.8 Pantallas de carga en a) Teléfono móvil y b) Tableta

5.2.3 Herramientas móviles

Cuando el usuario comienza a utilizar el prototipo por primera vez (i.e., cada nueva sesión de uso), si se oprime el botón “Go Meta!” en un dispositivo móvil, éste mostrará una pantalla de carga.

La pantalla de carga es un preámbulo para la distribución de la interfaz de usuario; es hasta este punto en el que los dispositivos están listos para recibir por vez primera alguno de los núcleos funcionales (ver Figura 5.8) Se permanecerá en este estado hasta que el usuario coloque una herramienta en un dispositivo.

Una vez que el usuario efectúe algún arrastre, se pasará de la pantalla de carga a la herramienta correspondiente. Todos los cambios ocurren en tiempo real y una vez hechos, ya no se volverá a la pantalla de carga (Pseudocódigo 10).

Todos los elementos en cualquier barra de herramientas se encontrarán siempre en sincronización (ver Figura 5.9). Para ello, el primer paso es obtener el valor desde la base de datos, así se asegura que todos los núcleos funcionales estén mostrando lo mismo en un tiempo determinado. Después, se obtiene el valor actual del núcleo funcional, ya sea mediante un clic o un toque (ver Figura 5.10). Lo único que resta es actualizar ese valor, es decir, sincronizarlo con la base de

Pseudocódigo 10 Navegación de núcleos funcionales para la pantalla de carga

Require: Agente de usuario (ua), contenido

- 1: **if** au = iPhone **then**
 - 2: **if** contenido = “Tools” **then**
 - 3: Ir a las herramientas de dibujo
 - 4: **else if** contenido = “Colour” **then**
 - 5: Ir a la paleta de colores
 - 6: **else if** au = iPad **then**
 - 7: **if** contenido = “Tools” **then**
 - 8: Ir a las herramientas de dibujo
 - 9: **else if** contenido = “Colour” **then**
 - 10: Ir a la paleta de colores
 - 11: **else if** contenido = “Both” **then**
 - 12: Ir a las herramientas de iPad
-

datos. De esa manera, cuando los núcleos funcionales obtengan valores desde la base de datos ya contarán con el valor actual. El Pseudocódigo 11 muestra un ejemplo de actualización para el color del trazo.

El diseño de las interfaces móviles es adaptable, es decir, se adecúan a las dimensiones de las pantallas del teléfono y la tableta respectivamente (ver Figura 5.11).

Pseudocódigo 11 Obtención y actualización del color del trazo

Require: Valor local (vl)

- 1: actualizar valor currentColour
 - 2: valor ← val.(DataSnapshot)
 - 3: valor ← vl
-

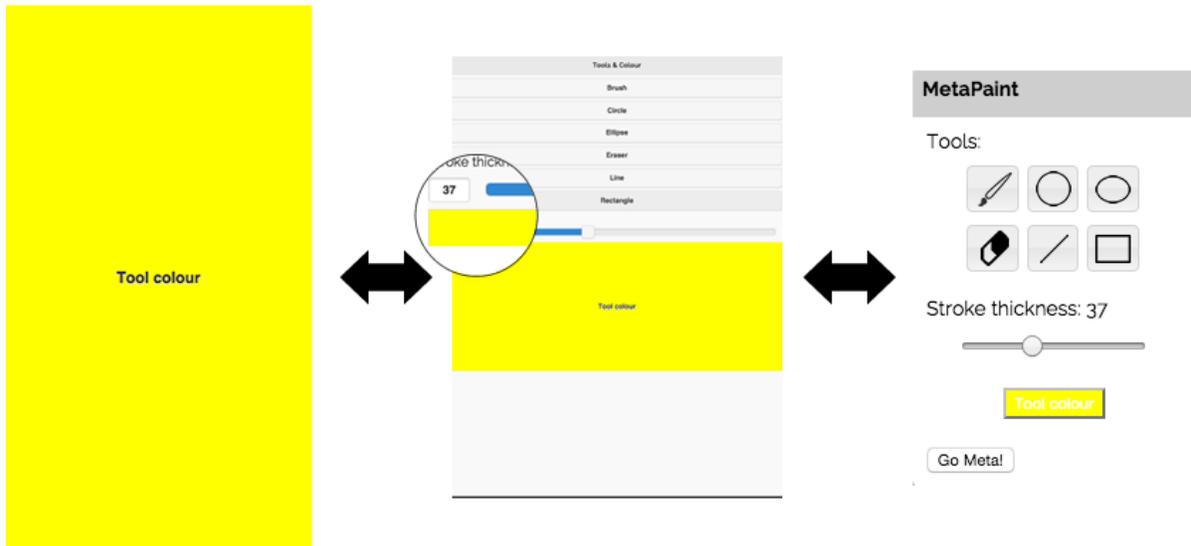


Figura 5.9 En todo momento los cambios ocurren en tiempo real y todos los núcleos funcionales están sincronizados

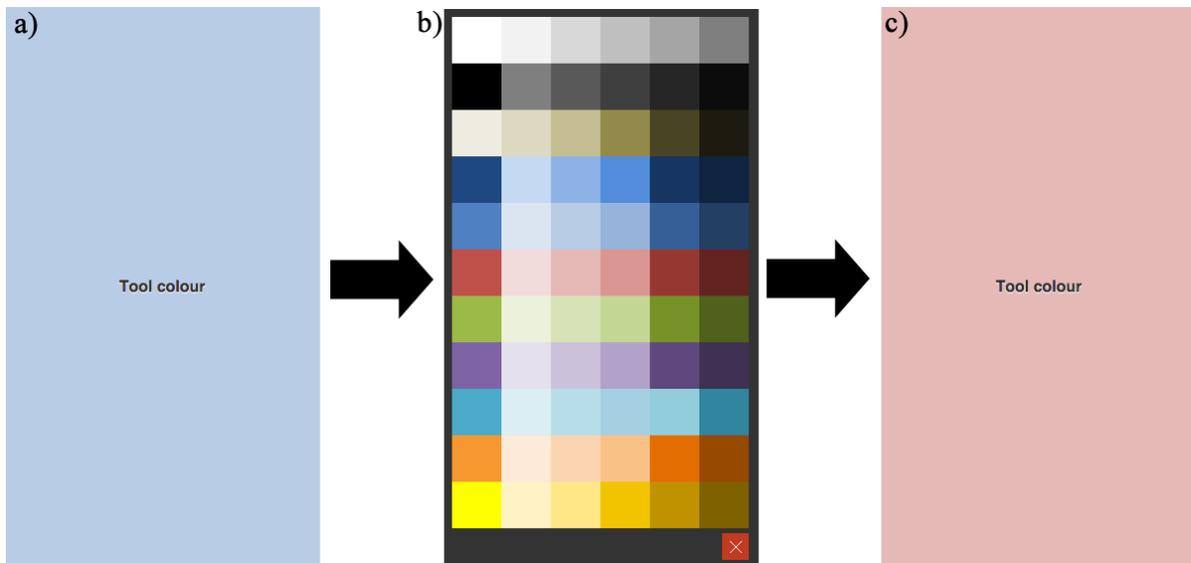


Figura 5.10 Paleta de colores en teléfono móvil (a). Toda la pantalla de navegación adquiere el color que en ese momento está en uso, cuando se da un toque en cualquier lugar se presenta la paleta de colores (b), cuando se selecciona el color deseado, la pantalla ahora adoptará ese color (c). Lo mismo ocurre cuando se despliega en la tableta

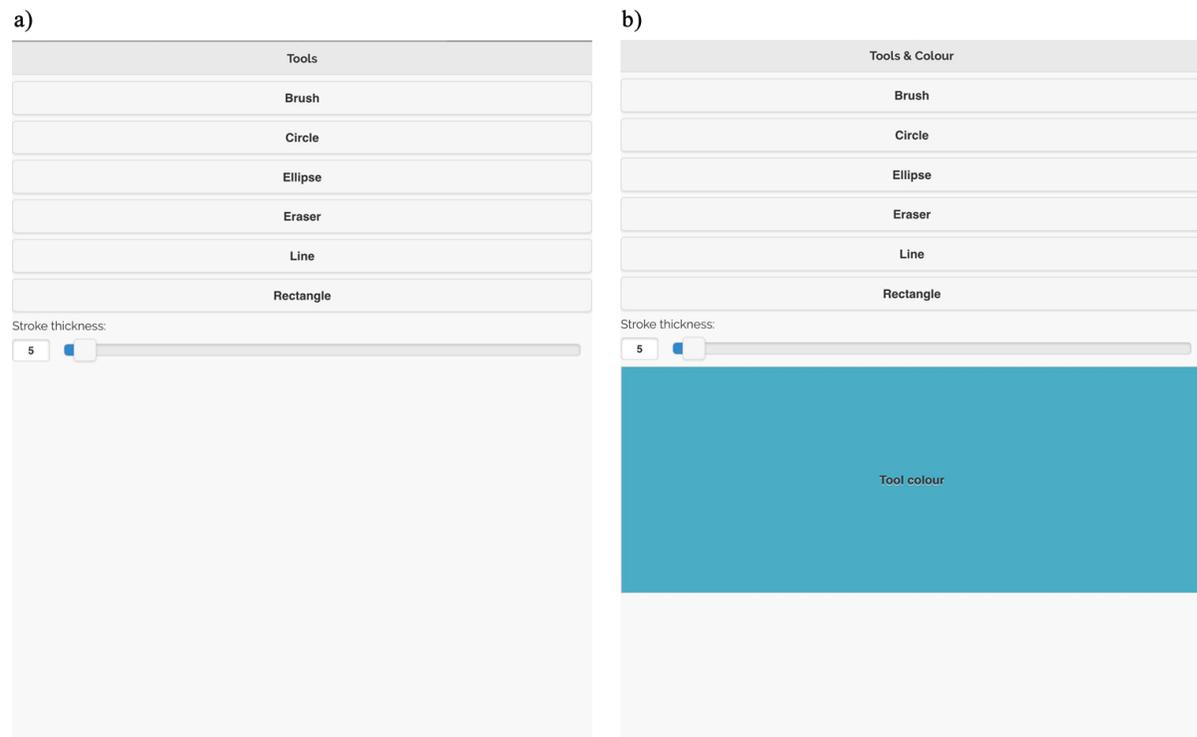


Figura 5.11 Herramientas visualizadas en una tableta a) “Tools” y b) “Both”

5.3 Pruebas con usuarios finales

La aplicación prototipo tiene como propósito ejemplificar cómo se podrían implementar las heurísticas de coherencia para el diseño de una meta-interfaz (Sección 3.1).

Para realizar las pruebas, se aplicó el cuestionario proporcionado por la NASA denominado *Task Load Index* (NASA-TLX) que tasa la carga de trabajo percibida por los usuarios finales, a fin de evaluar la eficacia de una tarea, un sistema o de un grupo u otros aspectos de rendimiento.

El cuestionario mide seis escalas:

1. **Exigencia mental:** ¿Cuánta actividad mental y perceptiva fue necesaria? (Por ejemplo: pensar, decidir, calcular, recordar, buscar, investigar, etc.) ¿Se trata de una tarea fácil o difícil, simple o compleja, pesada o ligera?
2. **Exigencia física:** ¿Cuánta actividad física fue necesaria? (Por ejemplo: empujar, tirar, girar, pulsar, accionar, etc.) ¿Se trata de una tarea fácil o difícil, lenta o rápida, relajada o cansada?
3. **Exigencia temporal:** ¿Cuánta presión de tiempo se sintió, debido al ritmo al cual se sucedían las tareas o los elementos de las tareas? ¿Era el ritmo lento y pausado o rápido y frenético?
4. **Esfuerzo:** En qué medida se ha tenido que trabajar (física o mentalmente) para alcanzar el nivel de resultados?
5. **Rendimiento:** ¿Cuál es el grado de satisfacción con el nivel de ejecución?
6. **Nivel de frustración:** Durante la tarea, en qué medida se sintió inseguridad, desaliento, irritabilidad, tensión, preocupación, o por el contrario, seguridad, agrado, relajación y satisfacción.

Se contó con una muestra de cinco usuarios, a cada uno se le explicó brevemente el objetivo de la aplicación y las tareas que debían de realizar, de igual forma, la intención del cuestionario y el significado de cada escala.

El conjunto de personas que fueron seleccionadas tienen conocimientos tanto básicos como avanzados en Computación. La primera prueba tuvo una duración promedio de 10 minutos, mientras que la segunda duró aproximadamente 15 minutos.

Por cada usuario final, se aplicaron dos cuestionarios, el primero tenía como objetivo medir la carga de trabajo al usar la vista de escritorio del prototipo, es decir, sólo se usó la versión

| Usuario | Exigencia mental | Exigencia física | Exigencia temporal | Rendimiento | Esfuerzo | Nivel de frustración |
|-----------|------------------|------------------|--------------------|-------------|-------------|----------------------|
| Usuario 1 | 5 | 5 | 5 | 100 | 5 | 0 |
| Usuario 2 | 2.5 | 0 | 5 | 90 | 2.5 | 2.5 |
| Usuario 3 | 5 | 2.5 | 5 | 100 | 5 | 0 |
| Usuario 4 | 10 | 5 | 10 | 80 | 10 | 5 |
| Usuario 5 | 5 | 0 | 2.5 | 95 | 5 | 0 |
| \bar{x} | 5.5 | 2.5 | 5.5 | 93 | 5.5 | 1.5 |
| σ | 2.738612788 | 2.5 | 2.738612788 | 8.366600265 | 2.738612788 | 2.236067977 |

Cuadro 5.2 Resultados del cuestionario para la vista de escritorio del prototipo

de PC, sin ningún dispositivo ni meta-interfaz. El segundo, mide el uso del prototipo con los mecanismos de meta-interfaz, por lo que el usuario tuvo que interactuar con la PC, el teléfono y la tableta.

La tarea del usuario es asignar a cada escala un puntaje entre 0 y 100. El Cuadro de resultados 5.2 muestra la ponderación de cada usuario, así como el promedio (\bar{x}) y la desviación estándar (σ) de cada apartado.

La Figura 5.12 representa los resultados en forma gráfica del primer cuestionario. Se puede apreciar que los niveles de exigencia mental, física y temporal así como el esfuerzo y la frustración tuvieron calificaciones muy bajas. Por el contrario, los niveles de rendimiento están por encima de 80 puntos.

| Usuario | Exigencia mental | Exigencia física | Exigencia temporal | Rendimiento | Esfuerzo | Nivel de frustración |
|-----------|------------------|------------------|--------------------|-------------|-------------|----------------------|
| Usuario 1 | 15 | 5 | 5 | 100 | 15 | 0 |
| Usuario 2 | 5 | 10 | 5 | 90 | 10 | 5 |
| Usuario 3 | 20 | 5 | 10 | 100 | 20 | 0 |
| Usuario 4 | 30 | 20 | 15 | 75 | 40 | 5 |
| Usuario 5 | 25 | 10 | 5 | 95 | 25 | 0 |
| \bar{x} | 19 | 10 | 8 | 92 | 22 | 2 |
| σ | 9.617692031 | 6.123724357 | 4.472135955 | 10.36822068 | 11.51086443 | 2.738612788 |

Cuadro 5.4 Resultados del cuestionario para el uso de MetaPaint

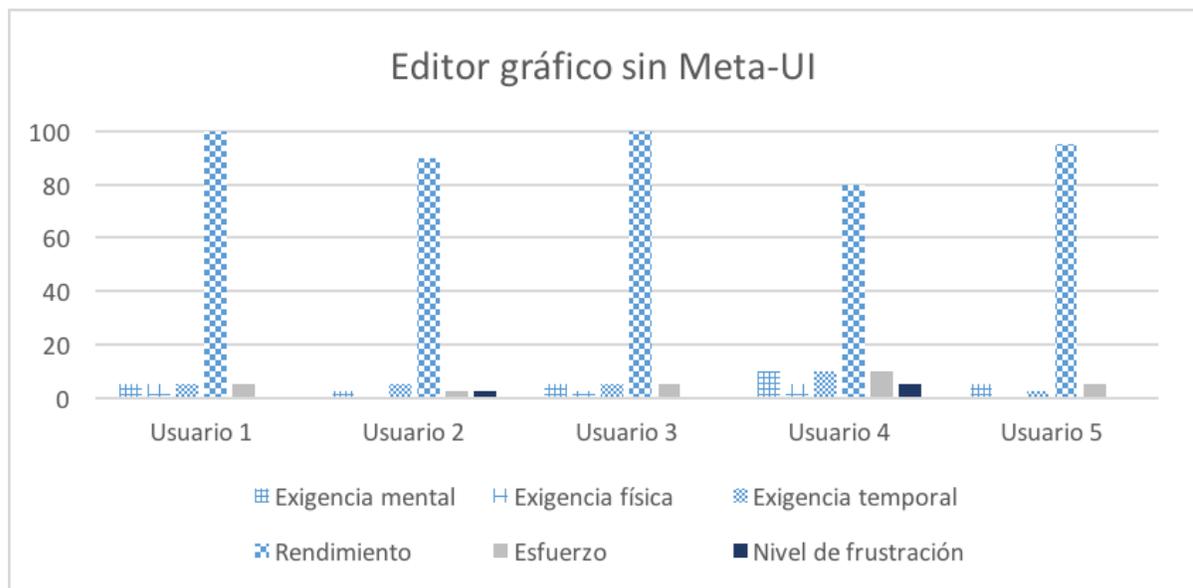


Figura 5.12 Resultados del cuestionario para la vista de escritorio del prototipo

El Cuadro 5.4 y la Figura 5.13 representan los resultados para el segundo cuestionario, el cual midió la carga de trabajo del prototipo “MetaPaint”, es decir, ahora con los mecanismos de meta-interfaz funcionando.

En esta ocasión, se muestra un aumento en los niveles de exigencia mental, física y temporal, así como en los de esfuerzo, en contraste con los de la Figura 5.12. No obstante, se observa que los valores de rendimiento mantienen un valor predominante. La frustración sigue con tasas triviales.

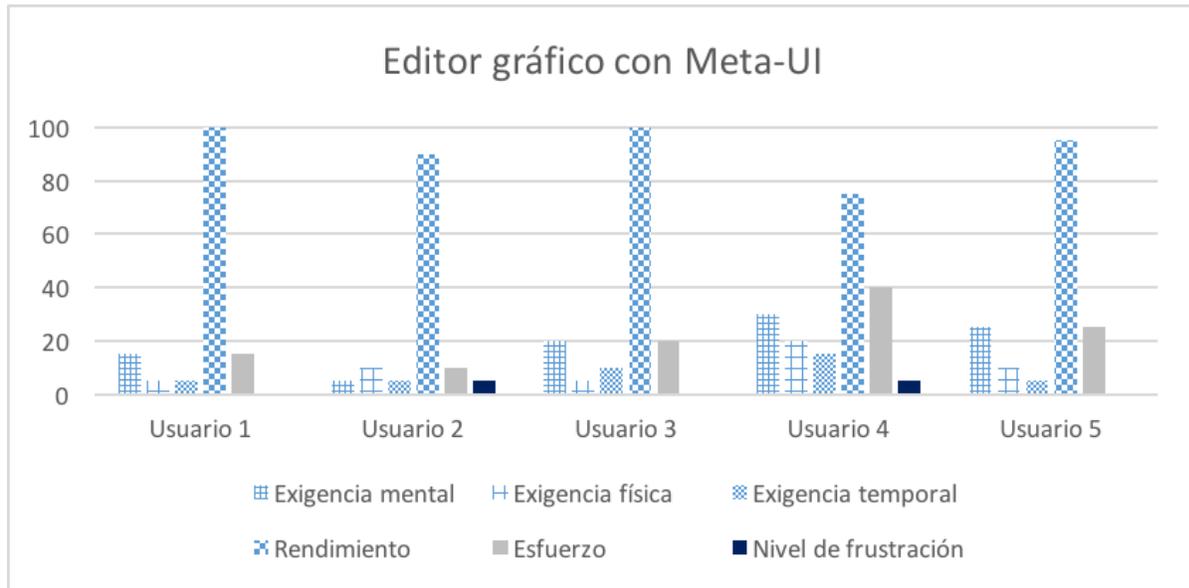


Figura 5.13 Resultados del cuestionario para el uso de MetaPaint

Por las variaciones entre los experimentos se puede deducir lo siguiente:

- La carga de trabajo en la primer prueba fue baja debido a la sencillez de la aplicación. Los usuarios tenían poco lugar para errar.
- El rendimiento fue alto porque la aplicación es coherente, responde rápidamente, entrega retroalimentación y es previsible.
- La exigencia mental, física y temporal, así como el esfuerzo, fueron de mayor ponderación en el segundo experimento, ya que el usuario ahora tiene que controlar y dividir su atención entre tres dispositivos en lugar de uno.
- Los índices anteriores se mitigaron gracias a la coherencia.
- Se redistribuyeron núcleos funcionales, y se rediseñaron sus componentes internos para adaptarlos a los dispositivos móviles. Se probó una característica de la heurística de núcleos funcionales: se deben redistribuir núcleos funcionales enteros, pero se pueden rediseñar sus componentes internos (ver Sección 3.3). Sin resultados negativos aparentes.

Capítulo 6

Conclusiones y trabajo a futuro

En este apartado se compendia el trabajo de investigación realizado. Se resaltan las aportaciones al estado del arte y los resultados que se obtuvieron a lo largo del trabajo. También se incluyen las restricciones y tareas a futuro del proyecto.

6.1 Recapitulación del problema

En los últimos años se han desarrollado algunas meta-interfaces de usuario, sin embargo muy pocas toman en cuenta la coherencia entre sus elementos de distribución.

La coherencia proporciona al usuario un marco estable en contextos similares. Es un elemento esencial de aprendizaje y una de las principales palancas para asegurar el uso eficiente de la interfaz.

En un ambiente interactivo con meta-interfaces de usuario, que las interfaces sean coherentes se vuelve una necesidad más evidente, con la diversidad de dispositivos, formas de interacción y entradas y salidas, se tiene que mantener al usuario concentrado.

Es por ello que se ideó una serie de heurísticas de coherencia para el diseño de meta-interfaces de usuario.

6.2 Conclusiones

En este trabajo, se presentan heurísticas de coherencia para diseñar meta-interfaces de usuario, basadas en los diferentes aspectos que pueden influir en el éxito de una distribución: técnicas de interacción, calidad y funcionalidad.

Cada aspecto se convierte en una dimensión en la que los principios de diseño se pueden elaborar, a fin de superar las limitaciones actuales impuestas por interfaces de usuario estereotipadas: un usuario está llevando a cabo una tarea en un dominio en un contexto de uso fijo.

Esta interfaz de usuario estereotipada ya no es aplicable en la actualidad, debido a las fronteras permeables de:

- **Usuarios:** un usuario podría participar con diferentes roles en diferentes grupos.
- **Plataformas:** una plataforma por si misma podría estar incluida en un grupo, es decir, en una infraestructura más grande.
- **Entornos:** un ambiente podría dar lugar a diferentes situaciones y diferentes ambientes podrían conectarse fácilmente unos a otros.

Las meta-interfaces permiten a los usuarios finales programar su ambiente interactivo, diseñar un flujo de trabajo acorde a sus necesidades y personalizar su interacción con los sistemas. No obstante, tal poder no se consigue fácilmente. Es necesario explorar diversos aspectos para garantizar una buena relación entre dominio y comodidad. En el presente trabajo se eligió indagar sobre la coherencia en las meta-interfaces de usuario.

La coherencia proporciona al usuario un marco estable en contextos similares. Es un elemento esencial de aprendizaje y una de las principales palancas para asegurar el uso eficiente de la interfaz. Además, es un complemento de la usabilidad, para mejorar la experiencia de uso.

El sitio Web “Nu-tronics” representado en la Figura 6.1 de la Sección 2.1.4 es un ejemplo de la mancuerna coherencia-usabilidad. Imaginemos un escenario donde tanto en la página a) como en la b) se aplicaran todas las heurísticas de usabilidad de Nielsen y quizás otras que no se consideraron en la tesis, dejando ambas páginas totalmente usables. Con el rediseño, ahora el sitio Web es usable pero sigue sin ser coherente, porque esas dos páginas son totalmente distintas.

Retomando el ejemplo de “Nu-tronics”: es coherente que un botón “Tienda” dirija al usuario a la tienda virtual del sitio, o que el botón “Radios”, muestre la selección de radiocomunicadores con los que cuenta la empresa. Otro ejemplo es que en la página a) todos los botones de la izquierda son visualmente coherentes, porque son rectángulos del mismo tamaño, del mismo color y con la misma tipografía. Lo anterior indica que el sitio cuenta con ciertos elementos de coherencia, pero no es usable. Una interfaz de usuario totalmente usable tiene ciertas características de coherencia, pero no es inherentemente coherente.

a

Nu-Tronics
3615 James Street
Syracuse, New York 13206
phone (315) 437-0033
fax (315) 463-5240
www.nu-tronics.com
locally owned & operated Since 1971

* Sales * service * Installation *

Store Hours : Tuesday through Friday 9:00am to 5:00pm

Commercial 2-way
C.B. RADIO 2-way
Marine 2-way radio
Scanner
Alert monitoring
Emergency Lights
Pagers
Flashlights

SCANNER PROGRAMMING
Digital & Analog
If you already have a digital trunking scanner and need it programmed for the area, we would be happy to program it for you.
We custom program digital scanners for both digital and analog frequencies and systems.
We set up, easy to use groups, giving you the flexibility of turning on or off agencies & departments. Your scanner will also be reformat to eliminate all the unwanted and unwanted non-programmed frequencies.

Our friendly, knowledgeable staff is always ready to answer your questions and assist you in your communications needs.
Please feel free to stop in or call anytime.

to complete, most programming can be completed within about 1 to 3 days.
Please feel free to call us for more information

NEW DIGITAL RADIO SYSTEM
area counties using digital radios
AUBURN CITY
CAYUGA COUNTY
CORTLAND COUNTY

b

c

Nu-Tronics
3615 James Street
Syracuse, N.Y. 13206
phone (315) 437-0033
fax (315) 463-5240
www.nu-tronics.com

C.B. Radio's

In 1934 the Federal Communications Commission allocated frequencies for personal 2-way communications known as "Citizens Band Radio". These frequencies were rarely used until 1952 when a radio technician in Elmira, New York started to build tube C.B. radio's for himself and his friends. By the 1960's, C.B. user's and clubs were popping up all over. In the late 1970's C.B. radio's were becoming a craze with over the road travellers especially truck drivers. This paved the way for the C.B. industry which at it peak had 134 different manufacturers making C.B. Radio's. We, at Nu-Tronics, opened in 1972 when new radios were operating on vacuum tubes and the F.C.C. changed \$ 8.00 for a 5 year license. Now days, radios are controlled by irrigated circuits, are about 1/3 the size and the F.C.C. has abolished the license process.

Uniden PRO510d, 40 channel, 5 watt C.B. mobile. Analog tuning (which means it will always stay on the channel you set it on), ANL noise filtering, LED channel display, SRF meter and 2 year warranty. **\$ 60.95**

Uniden PRO520xl, 40 channel, 5 watt C.B. mobile. Same features as the above unit (PRO510d) plus, RF gain control, P.A. switch, and instant channel 9 switch. 2 year warranty.

Uniden PC687, 40 channel, 5 watt C.B. mobile. Features: Analog SRF meter, Analog tuning, New Dynamic squelch control (allows you to receive better with less noise), RF gain control, P.A. switch, instant channel 9 switch, extra long mic cord, and 2 year warranty.

Uniden PC787, 40 channel, 5 watt C.B. mobile, with weather channels, RF gain control, mic gain control, SWR meter system ANL & NB noise filtering system. Dynamic squelch control (allows you to receive better with less noise), extra long mic cord, instant channel 9 switch 2 year warranty. **Top of the mobile line**

NEW Cobra 29s: LED changeable color display: 40 channel C.B. mobile radio. With RF gain control, dynamic mic control, instant channel 9 switch, noise blander, SWR meter, and more.

Figura 6.1 Dos páginas de un mismo sitio Web. El enlace (b) en la página (a) redirecciona a la página (c). No se conserva ninguna estructura, colores, imágenes, tipografía ni elementos de navegación. Todo esto causa confusión y aumenta la carga cognitiva del usuario

Es por ello que para una experiencia de uso satisfactoria es necesario tener una interfaz de usuario coherente y usable, ya que con una interfaz totalmente coherente no es posible asegurar la usabilidad, ni tampoco en el caso contrario.

Las meta-interfaces hacen cambiar el rol de los usuarios, su entorno e incluso la plataforma en que trabajan; la coherencia es el elemento que los mantiene en una base estable. La consistencia es clave como apoyo a la distribución de interfaces, la inducción de un efecto significativo en el usuario final, la experiencia de uso, la ejecución de tareas, o cualquier otra métrica relevante.

El capítulo de estado del arte en el presente documento, en el apartado de meta-interfaces de usuario (Sección 2.1.5), engloba la investigación realizada por científicos europeos [Vanderhulst, et al., 2009], que ayuda a validar el trabajo de tesis realizado. En ese trabajo se presenta un punto a considerar para la adaptación de meta-interfaces, el trasfondo del usuario.

Básicamente, el trasfondo del usuario requiere considerar las expectativas que tenga el usuario final sobre el sistema, es decir, la manera de llevar a cabo tareas y los objetivos que tiene que cumplir. Esto en cierta medida lo resuelve la coherencia, ya que permite cumplir con la intuición de los usuarios.

También se menciona que una meta-interfaces tiene como objetivo simplificar los roles tanto del usuario como del desarrollador. La coherencia le permite al usuario simplificar su interacción con el sistema, al tener características usables, que le permiten aprender e inferir acciones para realizar sus objetivos. Una meta-interfaces consistente, es el producto de un desarrollo consistente,

es decir, se planearon escenarios de uso, se diseñaron elementos gráficos similares, se tomaron en cuenta las heurísticas de usabilidad (Sección 2.1.3), etc.

Las heurísticas de coherencia que se proponen, proporcionan al desarrollador las herramientas que le permiten reflexionar sobre los puntos anteriores. Adicionalmente, como otro aporte significativo, las heurísticas son libres de dependencias de dominio específicas, esto significa que a diferencia de las interfaces de usuario específicas para trabajos y servicios, las meta-interfaces permiten a los usuarios finales explorar su entorno, identificar las tareas y servicios compatibles con el ambiente, y asignar los recursos disponibles para ejecutar tareas e interactuar con servicios.

Una meta-interfaz dispone de una configuración de los recursos y su papel en el medio ambiente, guiada por el usuario, permite a los usuarios finales combinar recursos independientes de una manera significativa.

Se descubrió adicionalmente, que la diferencia entre un dispositivo móvil y aquel que no lo es, no es una cuestión trivial. En la actualidad muchos diseñadores se decantan por el diseño Web adaptable (*responsive*), que ofrece el mismo diseño a todos los dispositivos, pero adaptándose dinámicamente a cada uno, no obstante, no es una solución que carezca de inconvenientes.

Gracias a la aparición constante de nuevos aparatos, es imposible diseñar interfaces específicas para cada uno. Con el paso del tiempo, el límite entre un dispositivo móvil y uno no móvil, se desvanece, pero hasta que se alcance una clasificación totalmente homogénea, el tratamiento entre dispositivos es un problema abierto.

6.3 Limitaciones

A continuación se describen las restricciones de la aplicación prototipo desarrollada:

- La comunicación entre las interfaces/dispositivos recae en Firebase, por lo que no se tiene control sobre fallas ni interrupciones de servicio.
- Todos los dispositivos deben tener en todo momento acceso a Internet.
- Al utilizar características recientes de HTML, CSS y Javascript, todos los navegadores Web tienen que estar actualizados.
- Sólo se pueden utilizar versiones de iPhone y iPad como dispositivos móviles.

Mientras que las limitantes de las heurísticas de coherencia son:

- Al ser heurísticas, su naturaleza es empirista, i.e., es necesario experimentar para probar.
- Sólo están orientadas a la coherencia.

6.4 Trabajo a futuro

Algunas ideas para el prototipo son:

- Hacerlo compatible con más dispositivos.
- Comprimir el código de Javascript.
- Mejorar el rendimiento de la comunicación.
- Compatibilidad con multisesión.

Trabajos futuros para las heurísticas:

- Desarrollar más heurísticas de meta-interfaz.
- Refinar los escenarios existentes.
- Agrupar las características de los escenarios en un framework.
- Validar las heurísticas de coherencia con otros prototipos.
- Validar las heurísticas de coherencia con desarrolladores.

Capítulo 7

Bibliografía

[Apple, 2015] Apple, *Design Principles*. Recuperado de: <https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/Principles.html>

[Bellotti et al., 2002] Victoria Bellotti, Maribeth Back, W. Keith Edwards, Rebecca E. Grinter, Austin Henderson y Cristina Lopes, *Making Sense of Sensing Systems: Five Questions for Designers and Researchers*. In Proceedings of the SIGCHI conference on Human factors in computing systems (CHI 2002), ACM Press, pp. 415-422, Nueva York, EE.UU., 2002.

[Boehm, 1986] Barry W. Boehm, *A spiral model of software development and enhancement*, In ACM SIGSOFT Software Engineering Notes, Vol. 11, No. 4, pp. 14-24, ACM, 1986.

[Coutaz, 2010] Joëlle Coutaz, *User interface plasticity: model driven engineering to the limit!*, In Proceedings of the 2nd ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '10), ACM, pp. 1-8, Nueva York, EE.UU., 2010.

[Coutaz, 2006] Joëlle Coutaz, *Meta-User Interfaces for Ambient Spaces*, Task Models and Diagrams for Users Interface Design, In Proceedings of the 5th International Workshop, Springer, pp. 1-15, Hasselt, Bélgica, 23-24 octubre 2006.

[Coutaz et al., 2005] Joëlle Coutaz, Stanislaw Borkowski, Nicolas Barralon, *Coupling Interaction Resources: an Analytical Model*, In Proceedings of the European Symposium on Ambient Intelligence (EUSAI 2005), ACM, pp. 183-188, Grenoble, Francia, 2005.

- [Coutaz y Calvary, 2012] Joëlle Coutaz y Gaëlle Calvary, *HCI and Software Engineering for user interface plasticity*, In *The Human-Computer Handbook - Fundamentals, Evolving Technologies, and Emerging Applications*, 3ra edición, CRC Press Taylor and Francis Group, cap. 52, pp. 1195-1220, 2012.
- [Crowley et al., 2005] James L. Crowley, Simon Dobson, David Garlan y Joëlle Coutaz, *Context is key*, In *Communications of the ACM*, Vol. 48, No. 3, pp. 49-53, ACM, 2005.
- [Demeure et al., 2005] Alexandre Demeure, Lionel Balme, Gaëlle Calvary y Joëlle Coutaz, *CamNote: A Plastic Slides Viewer*, In *International Workshop on Plastic Services for Mobile Devices 2005*, Roma, Italia, septiembre 2005.
- [Domínguez-Dorado, 2005] M. Domínguez-Dorado, *Bases de datos en el cliente con JavaScript DB*, In *Todo Programación*, No. 12, pp. 48-51, Editorial Iberprensa, Madrid, 2005.
- [Douglas y Liu, 2011] Ian Douglas y Zhengjie Liu, *Issues in Adapting Usability Testing for Global Usability*, In *Global Usability*, 1ra edición, Springer, cap. 2, pp. 23, 2011.
- [Erickson, 1995] Thomas Erickson, *Coherence and Portrayal in Human-Computer Interface Design*, In *Dialogue and Instruction: Modelling Interaction in Intelligent Tutoring Systems*, 1ra edición, Springer Science & Business Media, pp. 302-304, 1995.
- [Firebase, 2015] Firebase, *Features*. Recuperado de: <https://www.firebase.com/features.html>
- [Fishkin, 2004] Kenneth P. Fishkin, *A taxonomy for and analysis of tangible interfaces*, In *Personal and Ubiquitous Computing*, Vol. 8, No. 5, pp. 347-358, Springer, 2004.
- [Garlan et al., 2002] David Garlan, Daniel P. Siewiorek, Asim Smailagic y Peter Steenkiste, *Project Aura: Toward Distraction-Free Pervasive Computing*, In *IEEE Pervasive Computing*. Vol. 21, No. 2, pp. 22-31, IEEE, 2002.
- [Grosjean, 2011] Jean Claude Grosjean, *Design d'interface et critère ergonomique 9: Cohérence*. Recuperado de: <http://www.qualitystreet.fr/2011/01/23/design-dinterface-et-critere-ergonomique-9-coherence/>
- [Hippe et al., 2012] Hippe, Z., Kulikowski, J., Mroczek, T., y Wtorek, J., *Human-Computer Systems Interaction: Backgrounds and Applications 3*, In *Advances in Intelligent Systems and Computing*, 300, Springer, cap. 1, pp. 3-14, 2014.

[Jacobson, 1992] Ivar Jacobson, *Object-oriented programming*, In Object Oriented Software Engineering: A Use Case Driven Approach, 1ra edición, Addison-Wesley Professional, cap. 3, pp. 65-80, 1992.

[Knemeyer y Svoboda, 2015] Dirk Knemeyer y Eric Svoboda *User Experience - UX*, In The Glossary of Human Computer Interaction, 1ra edición, The Interaction Design Foundation, cap. 39, pp. 70-85, 2015.

[Li, 2009] Stan Z. Li, *Ergonomics*, In Encyclopedia of Biometrics, 1ra edición, Springer US, pág. 281, 2009.

[Love, 2005] Steve Love, *HCI Research Methods*, In Understanding Mobile Human-Computer Interaction, 1ra edición, Butterworth-Heinemann, cap. 4, pp. 63 - 75, 2005.

[MDN, 2015] Mozilla Developer Network, *HTML5*. Recuperado de: <https://developer.mozilla.org/es/docs/HTML/HTML5>

[MDN, 2015] Mozilla Developer Network, *CSS3*. Recuperado de: <https://developer.mozilla.org/en-US/docs/Web/CSS/CSS3>

[Microsoft, 2015] Microsoft, Layout. Recuperado de: <https://msdn.microsoft.com/es-mx/library/windows/desktop/dn742486.aspx>

[Nielsen, 2010] Jakob Nielsen, *What Is Usability?*, In User Experience Re-Mastered - Your Guide to Getting the Right Design, 1ra edición, Morgan Kaufmann, cap. 1, pp. 3-22, 2010.

[Norman y Draper, 1986] Donald A. Norman y Stephen W. Draper, *User Centered System Design: New Perspectives on Human-computer Interaction*, 1ra edición, CRC Press, 1986.

[Ole y Dybkjær, 2010] Niels Ole Bernsen y Laila Dybkjær, *Structure, Usability, Readership*, In Multimodal Usability, 1ra edición, Springer, cap. 1, pp. 7-8, 2010.

[Reed y Monk, 2010] Darren Reed y Andrew Monk, *Inclusive design: beyond capabilities towards context of use*, In Universal Access in the Information Society, Vol. 10, No. 3, pp. 295-305, Springer, 2010.

[Sottet et al., 2009] Jean-Sébastien Sottet, Gaëlle Calvary, Jean-Marie Favre y Joëlle Coutaz, *Megamodeling and Metamodel-Driven Engineering for Plastic User Interfaces: Mega-UI*, In Human-Centered Software Engineering, 1a edición, Springer, cap. 8, pp. 173-200, 2009.

[Usability.gov, 2015] *Usability Evaluation Glossary Terms*. Recuperado de: <http://www.usability.gov/what-and-why/glossary/tag/usability-evaluation/>

[Vanderdonckt, et al., 2008] Vanderdonckt, J., Calvary, G., Coutaz, J., y Stanciulescu, A., *Chapter 4: Multimodality for Plastic User Interfaces: Models, Methods, and Principles*, Multimodal User Interfaces, Signals and Communication Technology Series, pp. 61-84, Springer, Berlin-Heidelberg, Alemania, 2008.

[Vanderhulst, et al., 2009] Geert Vanderhulst, Daniel Schreiber, Kris Luyten, Max Mühlhäuser, Karin Coninx, *Edit, Inspect and Connect your Surroundings: A Reference Framework for Meta-UIs*, In Proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems (EICS '09), ACM, pp. 167-175, Pittsburgh, EE.UU., 2009.

[Weiser, 1999] Mark Weiser, *The computer for the 21st century*, In SIGMOBILE Mob. Comput. Commun. Rev., Vol. 3, No. 3, pp. 3-11, ACM, Nueva York, EE.UU., Julio 1999.