



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS  
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

**Departamento de Computación**

# Mecanismo móvil para adaptación plástica de interfaces gráficas multi-plataforma

Tesis que presenta

**José Eduardo Girón Camacho**

Para Obtener el Grado de

**Maestro en Ciencias**

**en Computación**

Directores de la Tesis

**Dra. Sonia Guadalupe Mendoza Chapa**

**Dr. César Torres Huitzil**

México, Distrito Federal.

Febrero 2012



---

## RESUMEN

---

Constantemente, nuevas plataformas y dispositivos móviles están saliendo al mercado sin embargo, con el afán de atraer al público, los diversos fabricantes incorporan en sus modelos características particulares, propiciando que no exista una uniformidad entre ellos. En consecuencia, dichos dispositivos se presentan en diversos tamaños y con diferentes capacidades de procesamiento, memoria, almacenamiento, despliegue y comunicación.

Esta diversidad de dispositivos y plataformas móviles no sólo pone en evidencia la necesidad de acceso ubicuo al procesamiento de la información, sino también impone nuevos requerimientos en el desarrollo de aplicaciones, tales como la adaptación de las interfaces de usuario, el desarrollo multi-plataforma, una mayor interactividad con el usuario, la adecuación de las aplicaciones a los contextos de uso del usuario, entre otros.

En este trabajo de tesis se toma, como caso de estudio, a la adaptación de las interfaces gráficas de usuario (IGU) a los despliegues de los dispositivos móviles y, como técnica de adaptación, a la plasticidad de interfaces gráficas de usuario, la cual se define como la capacidad de un sistema interactivo de adaptarse a cambios en el contexto de uso, preservando su usabilidad y continuidad de interacción.

Mediante el diseño de un mecanismo que relaciona las técnicas de plasticidad con el cómputo móvil, se definió un modelo para llevar a cabo una adecuada adaptación e implementación de las interfaces gráficas de usuario en un ambiente móvil sujeto a cambios en el contexto de uso.

El mecanismo de adaptación plástica surge como una respuesta a la gran diversidad de dispositivos móviles disponibles en el mercado y como una opción para homogeneizar el proceso de desarrollo de aplicaciones, al implementar y combinar un desarrollo multi-plataforma con métodos de la plasticidad de interfaces gráficas de usuario. Esta propuesta permite un ciclo de desarrollo rápido y eficiente.

**Palabras clave:** plasticidad de interfaces de usuario, sistemas móviles, contexto de uso, cómputo móvil y cómputo ubicuo.



---

## ABSTRACT

---

New platforms and mobile devices are constantly coming on to the market but, in order to attract the public, the several manufacturers include particular characteristics in their models, causing that any uniformity does not exist among them. Consequently, such devices present several sizes and different processing, memory, storage displaying and communication capacities.

This diversity of platforms and mobile devices not only highlights the need of ubiquitous access to information processing, but also imposes new requirements in the development of applications, such as the adaptability of graphical user interfaces, multi-platform development, interactivity with the user, adequacy of applications to the user's context of use, among others.

In this thesis work, we take as study case the adaptation of the graphical user interfaces to the displays of mobile devices and as adaptation technique, the plasticity of graphical user interfaces, which is defined as the capability of an interactive system to adapt itself to changes in the context of use, while preserving the usability and interaction continuity properties.

By designing a mechanism that relates plasticity techniques with mobile computing, a model has been defined to carry out a suitable adaptability and implementation of graphical user interfaces within a mobile environment, which is subject to changes in the context of use.

The plastic adaptation mechanism appears as a response to the large diversity of mobile devices available in the market and as an option to homogenize the process of application development by implementing and combining a multi-platform development using graphical user interface plasticity methods. This proposal allows an efficient development cycle.

**Keywords:** Plasticity of user interfaces, mobile systems, context of use, mobile and ubiquitous computing.



## *Dedicatoria*

*Este trabajo está dedicado a*

*San Pedro Mártir de Verona, Italia,*

*por ser mi guía y mi temple.*



## *Agradecimientos*

**A CONACYT y CINVESTAV-IPN:** *muchas gracias por brindarme las facilidades para llevar a cabo esta tesis.*

**A la Dra. Sonia G. Mendoza y al Dr. César Torres Huitzil:** *les agradezco mucho todo su tiempo y esfuerzo para la realización de esta tesis. Gracias por su apoyo y guía a través de esta etapa.*

**A mis profesores:** *gracias por todas sus enseñanzas.*

**A mi familia:** *gracias por su cariño y apoyo en esta etapa de mi vida.*

**A Sofi:** *muchas gracias por tu amistad y apoyo en estos años. Siempre apoyandonos en todo !! :D*

**A mis amigos y compañeros de generación:** *gracias por su amistad y compañía en cada momento que compartimos a lo largo de estos años.*



---

## ÍNDICE GENERAL

---

RESUMEN	III
ABSTRACT	V
Índice de figuras	XII
Índice de tablas	XIV
1 INTRODUCCIÓN	1
1.1 Motivación	1
Contexto de investigación	2
1.2 Planteamiento del problema	4
1.3 Hipótesis	5
1.4 Objetivos del proyecto	5
1.5 Organización de la tesis	6
2 FUNDAMENTOS Y ESTADO DEL ARTE	9
2.1 Definición Adaptación	9
2.2 Plasticidad	12
2.2.1 Dimensiones de la Plasticidad	12
2.2.2 Transformaciones Plásticas	14
2.2.3 Proceso de adaptación plástica	16
2.3 Contexto de uso	17
2.4 Sistemas móviles	18
2.4.1 Dimensiones de la movilidad	19
2.5 Ingeniería orientada a modelos (MDE)	21
Modelos	22
2.6 Trabajos relacionados	23
2.6.1 Modelo de asignación: un primer paso para asegurar usabilidad y sustentabilidad de plasticidad de interfaces gráficas de usuario	24
2.6.2 Un enfoque MDE-SOA para soportar interfaces de usuario plásticas en espacios de ambiente	25
2.6.3 Un proceso de desarrollo para interfaces de usuario plásticas	26
2.6.4 Un framework de MDE para interfaces de usuario plásticas	28
3 ENTORNO MÓVIL	31
3.1 Hardware gráfico móvil	31
3.1.1 Procesadores en dispositivos móviles	32
3.1.2 Pantallas de dispositivos móviles	34

## Índice general

3.2	Interfaces gráficas en dispositivos móviles	39	
3.3	Tipos de desarrollo de software para dispositivos móviles	40	40
3.3.1	Aplicaciones nativas ( <i>Uni-Platform</i> )	40	
3.3.2	Desarrollo multi-plataforma ( <i>cross-platform</i> )	42	
4	MECANISMO MÓVIL PARA ADAPTACIÓN PLÁSTICA	47	
4.1	Fundamentos del mecanismo	47	
4.1.1	Movilidad vs plasticidad	48	
4.1.2	Espacio de diseño para interfaces gráficas de usuario en dispositivos móviles	51	
4.1.3	Ciclo de adaptación de interfaces gráficas de usuario	56	56
4.2	Descripción del mecanismo de adaptación	58	
4.2.1	Arquitectura	59	
4.2.2	Interfaz abstracta	60	
4.2.3	XML y aplicaciones móviles	60	
4.2.4	Capas presentes en el mecanismo	61	
5	IMPLEMENTACIÓN DEL MECANISMO PROPUESTO	65	
5.1	Niveles tecnológicos abarcados en casos de implementación	65	65
5.2	Esquema de interacción entre tecnologías de desarrollo de software para el despliegue de una interfaz gráfica de usuario	68	
5.3	Contexto de uso en casos prácticos	69	
5.4	Ciclo de adaptación de las interfaces gráficas	70	70
5.5	Casos de uso implementados	72	
5.5.1	Aplicación Mobile Clock	72	
5.5.2	Aplicación Airport Reservation	76	76
5.5.3	Aplicación Demo Banco	81	
6	CONCLUSIONES Y TRABAJO FUTURO	93	
	Publicaciones del autor	97	
	Bibliografía	99	

---

## ÍNDICE DE FIGURAS

---

Figura 1	Áreas involucradas en la presente tesis.	3
Figura 2	Organización del documento.	6
Figura 3	Espacio de diseño	10
Figura 4	Espacio del problema de plasticidad en interfaces de usuario [Vanderdonckt et al., 2008].	13
Figura 5	Plasticidad de interfaces gráficas	14
Figura 6	Dimensiones móviles	20
Figura 7	Modelo de asignación [Vanderdonckt et al., 2006].	24
Figura 8	Enfoque MDE-SOA para el soporte de interfaces de usuario plásticas en espacios de ambiente [Coutaz et al., 2007].	26
Figura 9	Un framework basado en modelo para plasticidad de interfaces de usuario [Thevenin et al., 2002]	27
Figura 10	MDE Framework para interfaces gráficas de usuario plásticas [Coutaz and Calvary, 2008]	28
Figura 11	Arquitectura <i>hardware</i> en dispositivos móviles de alto rendimiento [Woo et al., 2010].	32
Figura 12	Configuración de los procesadores	33
Figura 13	Principales resoluciones de pantalla	35
Figura 14	Enfoques de desarrollo de software para dispositivos móviles	41
Figura 15	Correlación entre las dimensiones móviles y plásticas	49
Figura 16	Espacio de diseño para interfaces gráficas de usuario en dispositivos móviles	52
Figura 17	Técnicas plásticas para la adaptación de interfaces gráficas de usuario	55
Figura 18	Ciclo de adaptación	57
Figura 19	Mecanismo para el despliegue dinámico de una interfaz gráfica de usuario	58
Figura 20	Arquitectura cliente-servidor	59
Figura 21	Estructura de navegación	61
Figura 22	Capa de control	62
Figura 23	Capa de adaptación	63
Figura 24	Niveles tecnológicos abarcados	66
Figura 25	Esquema de interacción entre tecnologías	68

## Índice de figuras

Figura 26	Contexto de uso	70	
Figura 27	Ciclo de adaptación de interfaces gráficas móviles		71
Figura 28	Aplicación Mobile Clock	73	
Figura 29	Tecnologías Web para el caso de uso <i>Mobile Clock</i>		74
Figura 30	Tecnologías Web para el caso de uso <b>Reloj analógico</b>		74
Figura 31	Canvas en un dispositivo móvil	75	
Figura 32	Vistas de la aplicación <b>Mobile Clock</b>	75	
Figura 33	Tecnologías Web para el caso de uso <b>Airport Reservation</b>		76
Figura 34	Vista de la aplicación <b>Airport Reservation</b> desde un navegador Web en PC de escritorio	77	
Figura 35	Vista de la aplicación <b>Airport Reservation</b> desde un iPad		78
Figura 36	Vista de la aplicación <b>Airport Reservation</b> desde un iPhone		79
Figura 37	Vista de la aplicación <b>Airport Reservation</b> desde un Android		80
Figura 38	Pantalla de bienvenida en la aplicación <b>Demo Banco</b> en iPad y iPhone	83	
Figura 39	Vistas de la aplicación <b>Demo Banco</b> desde un iPhone: a) principal, b) inicio de sesión y c) alta	84	
Figura 40	Despliegue de datos de <b>cuenta bancaria</b> en iPad y iPhone		85
Figura 41	Opción <b>Consulta</b> vista desde un iPad	86	
Figura 42	Opción <b>Consulta</b> vista desde un iPhone	87	
Figura 43	Opción <b>Operaciones</b> vista en iPad y iPhone	88	
Figura 44	Opción <b>Transferencia</b> vista desde un iPad	89	
Figura 45	Opción <b>Transferencia</b> vista desde un iPhone	90	
Figura 46	Opción <b>Pago a tarjeta</b> vista en iPad y iPhone	91	
Figura 47	Opción <b>Contacto</b> vista en iPad y iPhone	92	

---

## ÍNDICE DE CUADROS

---

Cuadro 1	Principales resoluciones de pantalla en dispositivos móviles comerciales	36
Cuadro 2	Principales plataformas para dispositivos móviles	42
Cuadro 3	Tabla comparativa de los enfoques de desarrollo de software para dispositivos móviles	45



---

## INTRODUCCIÓN

---

El presente capítulo hace una breve descripción de la importancia que el cómputo móvil y el cómputo ubicuo están cobrando en la actualidad. La sección 1.1, se centra tanto en los retos que generan estas áreas en el desarrollo de software, como en el contexto de investigación que se está abordando. Posteriormente en la sección 1.2, se exponen los motivos que llevaron al desarrollo del presente trabajo, así como los objetivos esperados. Finalmente, en la sección 1.5 se describe la estructura de la tesis.

### 1.1 MOTIVACIÓN

En la actualidad el cómputo móvil y el cómputo ubicuo han tomado una gran importancia, debido a que cada día estamos incorporando y asimilando nuevos sistemas de cómputo en nuestras actividades cotidianas, así como a nuestra necesidad creciente de tener información relevante en cualquier momento y lugar.

El cómputo ubicuo ha facilitado a los usuarios el acceso a servicios y aplicaciones desde cualquier lugar, en cualquier momento y utilizando cualquier dispositivo [Poslad, 2009].

El área del cómputo ubicuo pretende incorporar capacidades de cómputo y de comunicación a los objetos de la vida cotidiana para crear un nuevo modelo de la realidad, en el que estos objetos interoperan para facilitar a las personas la realización de tareas, tales como intercambio de archivos, control de actividades personales, consulta y envío de correos electrónicos, búsqueda de información, manejo de contactos, trámites electrónicos, etc.

Por su parte, el cómputo móvil tiene por finalidad el procesamiento automático de información, por medio de dispositivos computacionales con capacidad de movilidad y con acceso digital a fuentes de información vía una infraestructura de comunicación [B'Far, 2005].

## INTRODUCCIÓN

Los sistemas de cómputo móvil son sistemas computacionales que se pueden mover físicamente con cierta facilidad y cuyas capacidades de cómputo se pueden usar mientras se están moviendo, i.e., laptops, PDAs (Personal Digital Assistants), teléfonos inteligentes (Smartphones), tablets, etc. [B'Far, 2005].

La creciente proliferación de dispositivos de cómputo móvil, así como el progreso de las redes de comunicación y la miniaturización de dichos dispositivos, han permitido imaginar al usuario como: “un ente que evoluciona en un entorno variado y que utiliza, de manera oportunista, plataformas de interacción diversas con el fin de satisfacer sus múltiples necesidades en cualquier lugar donde se encuentre” [Calvary et al., 2001].

Los ambientes de cómputo móvil se caracterizan por restricciones importantes de recursos y cambios frecuentes en las condiciones de operación, lo cual impone desafíos que involucran diversas áreas de la ciencias computacionales y de las ingenierías computacional, eléctrica y de telecomunicaciones.

### *Contexto de investigación*

Constantemente, nuevas plataformas y dispositivos móviles están saliendo al mercado sin embargo, con el afán de atraer al público, los diversos fabricantes incorporan en sus modelos características particulares, propiciando que no exista una uniformidad entre ellos, i. e., se presentan en diversos tamaños y con diferentes capacidades de procesamiento, memoria, almacenamiento y comunicación.

Esta diversidad de dispositivos y plataformas móviles no sólo pone en evidencia la necesidad de acceso ubicuo al procesamiento de la información, sino también impone nuevos requerimientos en el desarrollo de aplicaciones, i.e., adaptación de las interfaces de usuario, desarrollo multi-plataforma, mayor interactividad con el usuario, adecuación a los contextos del usuario y su medio, etc.

Como caso de estudio, seleccionamos la adaptación de interfaces de usuario en dispositivos móviles a cambios en el contexto de uso, i.e., al conjunto de variables observables que caracterizan las condiciones en que un determinado sistema está funcionando [Sottet et al., 2008].

Por lo cual es necesario tomar en cuenta ciertas áreas, identificadas en la fig. 1, que nos permitan establecer los requerimientos y necesidades en este tipo de interfaces de usuario.

De la fig. 1, podemos resumir que el diseño, la adaptación, la implementación y la validación de las interfaces gráficas móviles, es una combinación de varias áreas. Entre ellas las mencionadas previamente 1.1: cómputo móvil y cómputo ubicuo.

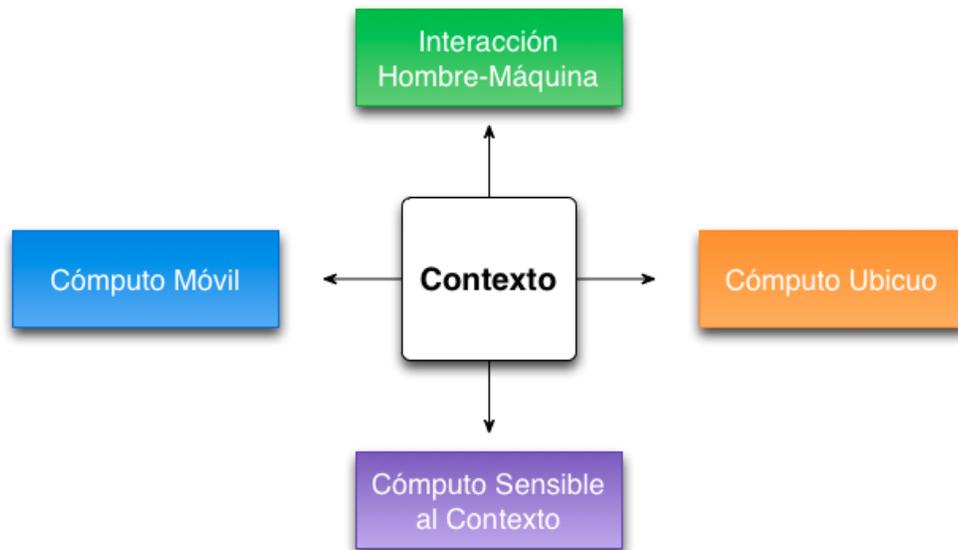


Figura 1: Áreas involucradas en la presente tesis.

Para llevar a cabo una correcta interacción con el usuario una interfaz gráfica debe cumplir con ciertos requisitos. El área de la Interacción Hombre-Maquina, establece el proceso de interacción, ya que se enfoca al estudio de cómo las personas interactúan con las computadoras y en qué grado las computadoras están desarrolladas para una interacción exitosa con los seres humanos [Zaphiris and Siang, 2009].

Hasta el momento las diversas interfaces de usuario propuestas en cómputo móvil, se han programado de manera estática, i. e., se diseñan y desarrollan para un equipo con hardware y plataforma particulares, por lo que no ofrecen al usuario su uso en otro dispositivo que no cumpla esas características.

El cómputo sensible al contexto establece la capacidad de los dispositivos de cómputo para detectar, sentir, interpretar y responder a los cambios de contexto del entorno local del usuario y de los propios dispositivos [Dey and Abowd, 2000].

## INTRODUCCIÓN

Por las circunstancias anteriores, se genera la necesidad de crear aplicaciones que implementen el despliegue dinámico de su interfaz gráfica de usuario, lo cual permitiría a la aplicación adaptarse a cualquier dispositivo y plataforma móvil, manteniendo su usabilidad.

Por lo tanto, es necesario añadir al software la capacidad de responder dinámicamente a los cambios de contextos que se presenten, i.e., adaptarse de acuerdo a las propiedades hardware y software que se puedan presentar en los dispositivos móviles.

### 1.2 PLANTEAMIENTO DEL PROBLEMA

En años recientes ha surgido una gran cantidad de dispositivos computacionales con la capacidad de movilidad y acceso digital a fuentes de información, i.e., teléfonos inteligentes (*smartphones*), tablets, PDA, etc., cada uno de ellos con características únicas y específicas tanto a nivel de hardware y como de software.

Más específicamente, existe una heterogeneidad entre estos dispositivos. A nivel de hardware podemos observar que cada uno de ellos cuenta con diversos componentes, i.e., batería, procesador, tarjeta de gráficos, memoria, GPS, tarjeta de red inalámbrica, pantalla, teclados *qwerty* o táctil, etc., sin embargo estos componentes se presentan en diversos tamaños y con diferentes capacidades.

Adicionalmente al hardware, los dispositivos móviles cuentan con una variedad de plataformas de software, i.e., Android, Bada, BlackBerry, iOS, Symbian, Windows Mobile, etc. Cada plataforma cuenta con sus propios *frameworks* para el desarrollo de software compatible con ellas y sus respectivos lenguajes de programación.

Ambos aspectos implican que el desarrollo de software para dispositivos móviles no sea homogéneo. Es decir, el desarrollo de software se tiene que adecuar al dispositivo para el cual se desea desarrollar una aplicación. Por lo tanto, nos enfrentamos a diversos retos cuando deseamos extender una aplicación de software hacia múltiples dispositivos en diversas plataformas.

En este trabajo de tesis tomamos como caso de estudio la adaptación de las interfaces gráficas de usuario (IGU) sobre dispositivos móviles y como técnica de adaptación la plasticidad de interfaces gráficas de usuario, la cual es la capacidad de un sistema de adaptarse a cambios en el contexto de uso, preservando la “usabilidad” [Calvary et al., 2006].

Por lo tanto, planteamos la siguiente **pregunta de investigación**:

¿Como podemos implementar y validar una determinada interfaz de usuario para  $d$  diferentes dispositivos móviles, sin escribir  $p$  programas, entrenar en  $l$  lenguajes y  $f$  frameworks y mantener  $l^*p$  modelos arquitecturales para describir la misma interfaz gráfica de usuario?

### 1.3 HIPÓTESIS

Mediante el diseño de un mecanismo que relacione e incorpore las técnicas de plasticidad y cómputo móvil establecer un procedimiento para llevar a cabo una adecuada adaptación e implementación de las interfaces gráficas de usuario en un ambiente móvil con múltiples contextos de uso.

### 1.4 OBJETIVOS DEL PROYECTO

#### **General**

Desarrollar un mecanismo que permita el despliegue dinámico de sistemas plásticos en multiples plataformas móviles.

#### **Particulares**

- Analizar e implementar la adaptación de interfaces gráficas de usuario mediante la técnica de plasticidad en dispositivos móviles.
- Diseñar e implementar un mecanismo de despliegue dinámico de interfaces gráficas de usuario.
- Desarrollar un conjunto de aplicaciones (redistribuibles en dispositivos heterogéneos) que validen la funcionalidad del mecanismo propuesto.
- Ejemplificar los diferentes tipos de transformaciones plásticas en dispositivos móviles.

## INTRODUCCIÓN

### 1.5 ORGANIZACIÓN DE LA TESIS

La presente tesis está estructurada en seis capítulos, como se puede apreciar en la fig. 2.

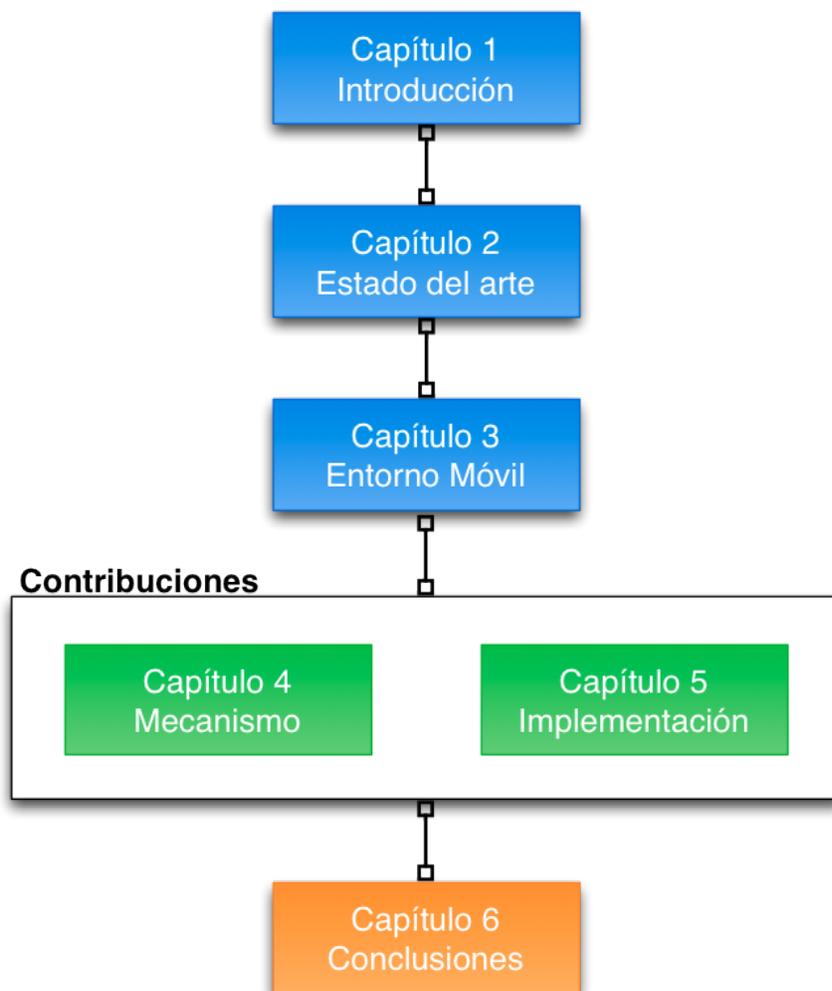


Figura 2: Organización del documento.

El capítulo 2 comprende el estado del arte. En éste se describe los conceptos relacionados con las interfaces gráficas de usuario, i.e., propiedades, áreas relacionadas, tipos de adaptaciones. Además se introduce el tema de plasticidad así como trabajos relacionados con este tema de investigación.

En el capítulo 3 se aborda, de manera teórica, las propiedades gráficas que presentan los dispositivos móviles, así como las características que debe cubrir una interfaz en un entorno móvil. Finalmente, se detalla los tipos de desarrollo de software existentes, para llegar al desarrollo multi-plataforma, el cual se usó en esta tesis.

Las contribuciones del presente trabajo se reflejan en los capítulos 4 y 5. En el capítulo 4 se detalla el análisis y el diseño del sistema de despliegue dinámico. En particular, se describen los principales componentes del sistema y la forma en la que interactúan. Se da a conocer el esquema general del mecanismo para desarrollo multi-plataforma y el mecanismo de coherencia de interfaces gráficas de usuario y de servicios.

En el capítulo 5 se describe la implementación del mecanismo propuesto. En él se da a conocer el software desarrollado que permite ejemplificar el modelo propuesto en diferentes casos de uso.

Finalmente, en el capítulo 6, se presentan las conclusiones de este trabajo de investigación, así como algunas ideas de trabajo a futuro que podrían mejorar el despliegue dinámico de sistemas plásticos.



---

## FUNDAMENTOS Y ESTADO DEL ARTE

---

El presente capítulo consta de 6 secciones. En la sección 2.1 se introduce el concepto de adaptación en un sistema y la adaptación como caso de estudio en interfaces gráficas de usuario. Posteriormente en la sección 2.2 se describe la plasticidad como una propiedad de adaptación en los sistemas interactivos y sus principales características. En la sección 2.3 se da a conocer la definición de contexto de uso y los elementos que lo conforman para el presente trabajo. Después, para la sección 2.4 se describen los sistemas móviles y sus propiedades, los cuales son nuestro objeto de estudio. En la sección 2.5 se presenta la ingeniería orientada a modelos y conceptos relacionados para el modelado de la solución. Finalmente en la sección 2.6 se dan a conocer los principales trabajos relacionados.

### 2.1 DEFINICIÓN ADAPTACIÓN

En HCI (Interacción Hombre-Máquina) la **adaptación** es modelada con base en dos propiedades complementarias del sistema: la capacidad de adaptación y la adaptabilidad.

La capacidad de **adaptación** es la característica de un sistema que permite a los usuarios configurar sus propiedades de acuerdo a un conjunto de parámetros predefinidos, mientras que la **adaptabilidad** es la capacidad del sistema de llevar a cabo una adaptación automáticamente, sin la necesidad de intervención de un usuario mediante una acción deliberada.

De acuerdo a Thevenin y Coutaz [Thevenin and Coutaz, 1999] para llevar a cabo la adaptación de un determinado sistema se define un espacio de diseño, cuyos ejes son: objetivo de la adaptación, medios de adaptación, tiempo de adaptación y finalmente el actor, como se puede apreciar en la fig. 3.

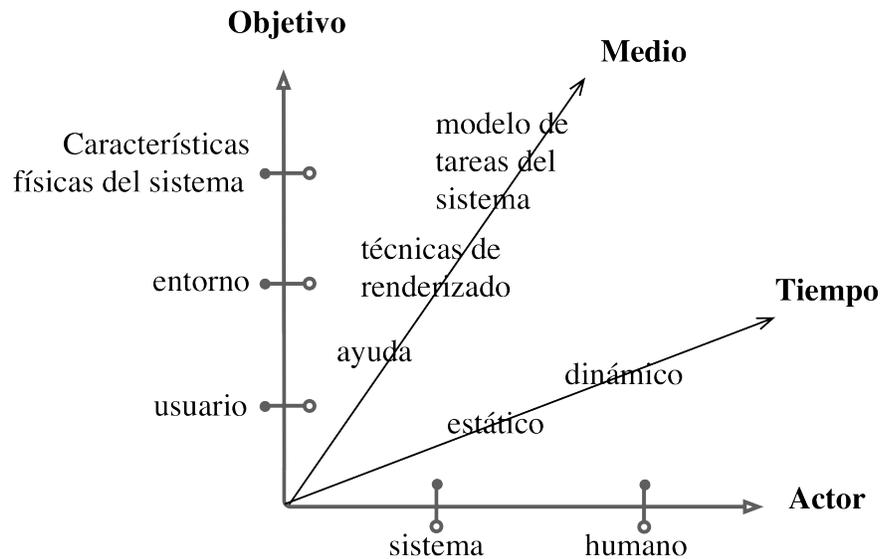


Figura 3: Espacio de diseño de la adaptación de un sistema [Thevenin and Coutaz, 1999].

- **Objetivo de la adaptación:** este eje representa a las entidades para las que se pretende llevar a cabo la adaptación: la adaptación a los usuarios, la adaptación al entorno y la adaptación a las características físicas del sistema. Particularmente las características físicas de un sistema pueden ser refinadas en términos de dispositivos de interacción (e.g., ratón, teclado, pantalla, cámaras de vídeo), hardware (e.g., memoria y capacidad de procesamiento) y medio de comunicación (e.g., ancho de banda de los canales de comunicación).
- **Medios de adaptación:** este eje denota los componentes del sistema que están envueltos en el proceso de adaptación: típicamente, el modelo de tareas del sistema, las técnicas de renderizado y los subsistema de ayuda, los cuales pueden ser modificados para adaptarse a las entidades objetivo. El modelo de tareas del sistema es la implementación de un modelo para una tarea especificada por el usuario (factor humano). La técnica de renderizado denota la presentación y el comportamiento observable del sistema. Finalmente, los subsistemas de ayuda incluyen soporte sobre el sistema mismo y el dominio de la tarea.
- **Dimensión temporal de la adaptación (tiempo):** determina si la adaptación puede ser **estática** (i.e., efectiva entre sesiones) y/o **dinámica** (i.e., que se produce en tiempo de ejecución).

- **Actor:** este eje finalmente nos indica quién es el encargado de mandar a llamar el proceso de adaptación, i.e., el usuario o el propio sistema.

### *Adaptación de Interfaces Gráficas de Usuario*

En el presente trabajo nos enfocamos en la adaptación de las interfaces gráficas de usuario que componen un sistema software.

En el área de la **Interacción Hombre-Máquina**, la interfaz gráfica de usuario, es el canal mediante el cual un sistema interactivo posibilita, a través del uso y la representación del lenguaje visual, una interacción amigable con un sistema informático.

La interfaz gráfica de usuario, es un software que actúa como un medio de comunicación entre el usuario y un determinado dispositivo hardware, utilizando un conjunto de imágenes y objetos gráficos para representar la información y las acciones disponibles en dicha interfaz. Su principal uso consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo del dispositivo con el cual se encuentre interactuando un usuario.

La interfaz gráfica de usuario debe proveer específicamente los siguientes componentes de diseño en una forma funcional y estética (también conocida como forma de rendimiento y preferencias de orientación) [Helander et al., 1997]:

- **Metáforas:** conceptos esenciales para comunicar a través de términos e imágenes.
- **Modelo mental:** organización de datos, funciones, tareas y roles.
- **Navegación del modelo mental:** menús, iconos, cuadros de diálogo y ventanas.
- **Apariencia:** visual, auditiva, verbal, así como propiedades de controles y de fondo ornamental.
- **Interacción:** comportamiento de los controles interactivos de la pantalla, así como la entrada y salida de datos en los dispositivos.

Entre los procedimientos para llevar a cabo la adaptación de interfaces gráficas de usuario, nos enfocaremos en la adaptación e implementación de la técnica denominada **plasticidad**, sobre dispositivos móviles, la cual se describe a continuación.

## 2.2 PLASTICIDAD

La plasticidad de interfaces gráficas de usuario [Thevenin and Coutaz, 1999] es una propiedad de los sistemas interactivos, introducida en respuesta a la diversidad de dispositivos de cómputo.

Este concepto se inspira en la propiedad de los materiales que les permite expandirse y contraerse bajo restricciones naturales, sin romperse. Aplicada a interfaces gráficas de usuario, la plasticidad es la capacidad de adaptarse a cambios en el contexto de uso, preservando la “usabilidad” [Calvary et al., 2006]. Una interfaz de usuario plástica preserva la usabilidad si sus propiedades, seleccionadas durante el diseño, se mantienen dentro un rango de valores cuando ocurre la adaptación a cambios contextuales. Otra propiedad ineludible de la plasticidad es la “garantía de continuidad” cuando se produce la migración de la aplicación de un contexto de uso a otro.

### 2.2.1 Dimensiones de la Plasticidad

A continuación se describen los principales problemas en los que se centra la plasticidad como área de estudio [Vanderdonckt et al., 2008]:

1. **Métodos de adaptación:** consiste en los métodos que aporta la plasticidad que permiten llevar a cabo la adaptación de una interfaz gráfica de usuario. Los métodos de adaptación existentes son: redistribución y remodelación.
2. **Granularidad de los componente de la interfaz de usuario:** es la unidad de elementos gráficos pertenecientes a una interfaz de usuario, que puede ser afectada por un método de adaptación. Existen 3 niveles principalmente: 1) interactor, un único elemento gráfico, 2) diálogo, un conjunto de elementos gráficos y 3) total, abarca a todos los elementos gráficos de la interfaz de usuario.
3. **Granularidad del estado de recuperación:** permite medir en términos de acciones físicas el costo de la adaptación para el usuario.
4. **Despliegue de la interfaz de usuario:** se refiere a la forma de generar la interfaz de usuario para cada contexto de uso, i.e., determina si la interfaz gráfica de usuario se encuentra definida o se genera en tiempo de ejecución.
5. **Contexto de uso:** es el conjunto de variables observables que caracterizan las condiciones en las que un sistema en particular se encuentra ejecutándose. Se definen tres principalmente: el usuario, la plataforma y el ambiente.

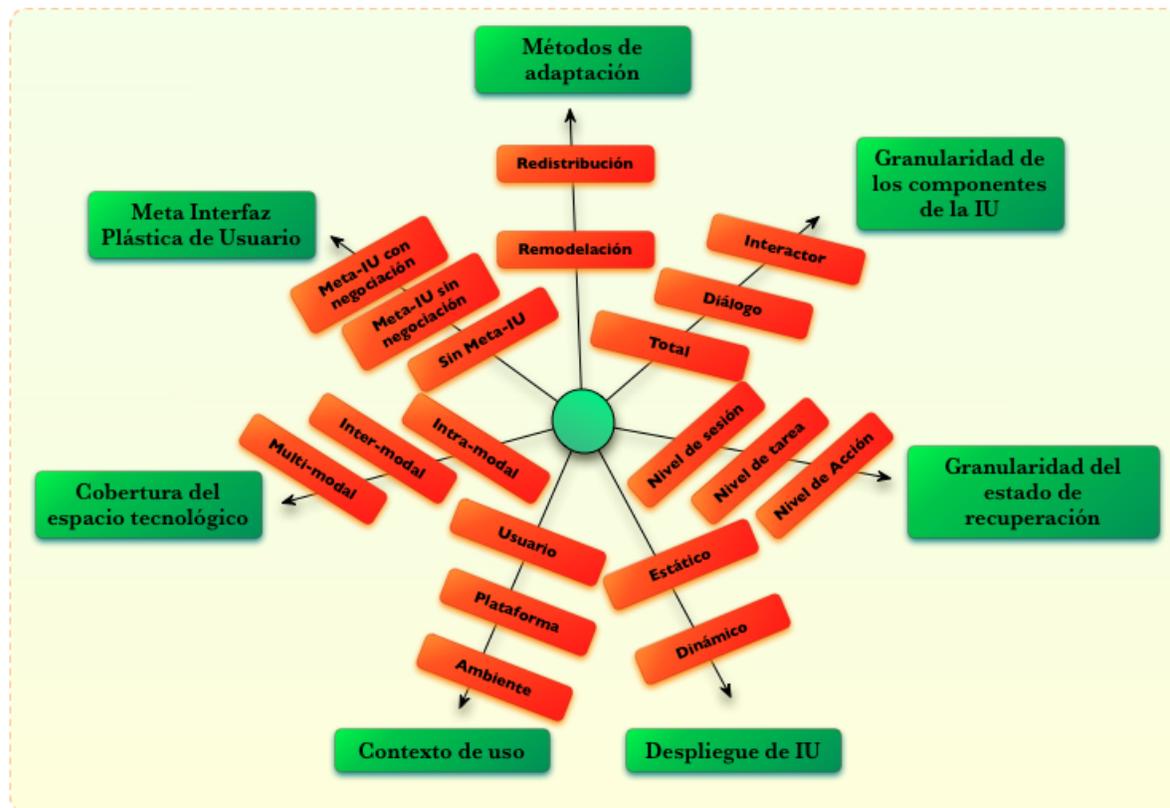


Figura 4: Espacio del problema de plasticidad en interfaces de usuario [Vanderdonckt et al., 2008].

6. **Cobertura del espacio tecnológico:** un espacio tecnológico es un marco de trabajo con un conjunto de conceptos asociados, conocimientos, herramientas y habilidades necesarias. Ejemplo de espacio tecnológico es son el lenguaje de programación para el desarrollo de las interfaces gráficas de usuario, la adaptación se puede llevar a cabo de la siguiente manera:

- a) **intra-modal**, e.g., de Java a Java,
- b) **inter-modal**, e.g., de HTML a Java,
- c) **multi-modal**, e.g., combinación de Java y HTML antes y después de la transformación de la interfaz de usuario.

7. **Meta-interfaz plástica de usuario:** es un entorno de interactivo para el usuario final. Una meta interfaz permite a los usuarios finales el configurar y controlar sus espacios activos, para depurarlos (evaluar) y para mantener y reutilizar programas.

### 2.2.2 Transformaciones Plásticas

Existen múltiples formas en las que un sistema puede llevar a cabo el proceso de adaptación en respuesta a un cierto contexto de uso. En este trabajo limitaremos la adaptación de un sistema a los cambios efectuados sobre la interfaz gráfica de usuario en dispositivos móviles.

De acuerdo a Coutaz y Calvary [Coutaz and Calvary, 2008] el proceso de adaptación en interfaces gráficas de usuario se puede llevar a cabo usando una o la combinación de las siguientes técnicas plásticas: remodelación, distribución y migración de la interfaz gráfica de usuario, como se puede apreciar en la fig. 5.

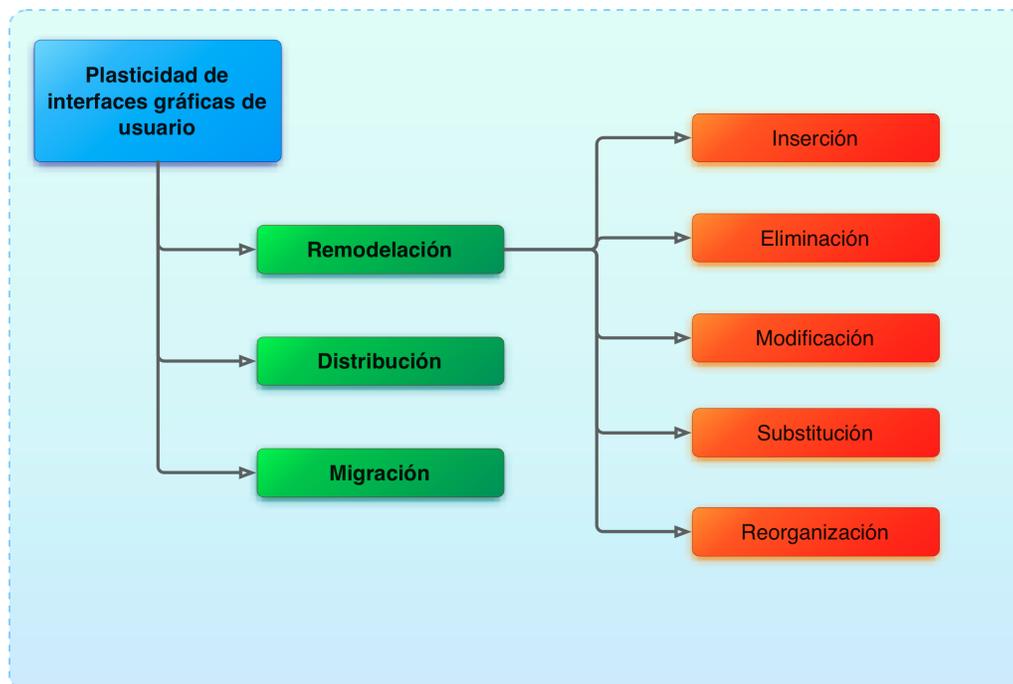


Figura 5: Plasticidad de interfaces gráficas de usuario [Coutaz and Calvary, 2008].

En el presente trabajo centraremos nuestra atención en la técnica de remodelación, por ser ésta la que denota la reconfiguración de las interfaces de usuario, de la cual daremos a conocer los métodos implicados en esta técnica y finalmente cómo implementarlos en interfaces gráficas de usuario móviles.

Brevemente damos a conocer las características de las otras dos técnicas:

La técnica de **Migración** corresponde a la transferencia de todos o parte de los componentes de la interfaz de usuario a diferentes fuentes de interacción, ya sea que estos recursos pertenezcan a la plataforma actual o a otra.

Mientras que la técnica de **distribución** se implementa cuando los recursos de interacción se encuentran distribuidos a lo largo de un cluster.

#### *Remodelación de las interfaces gráficas de usuario*

Esta técnica es la que tiene mayor importancia en este trabajo. La remodelación de interfaz gráfica de usuario es la encargada de reconfigurar la interfaz gráfica de usuario que resulta de la aplicación de una o de diversas transformaciones, de forma parcial o total sobre la interfaz en la cual se está trabajando [Coutaz and Calvary, 2008].

Existen cuatro diferentes técnicas de adaptación de interfaces gráficas de usuario que son: eliminación, adición, sustitución y reorganización de la interfaz gráfica de usuario. Estas transformaciones se pueden aplicar a múltiples niveles de abstracción.

De acuerdo a Vanderdonckt et al. [Vanderdonckt et al., 2008] la **remodelación de interfaces gráficas de usuario**, denota cualquier reconfiguración que sea perceptible por el usuario y que resulte de la aplicación de transformaciones en los componentes gráficos de dicha interfaz de usuario, dando como resultado una degradación o mejora en la interfaz. Las técnicas de transformación para interfaces gráficas de usuario son:

- **Inserción** de nuevos componentes de interfaz de usuario: facilita el acceso a nuevos servicios relevantes en un nuevo contexto de uso. Por ejemplo, si hay más espacio de pantalla disponible, más información se puede mostrar.
- **Eliminación** de componentes de interfaz de usuario que se vuelven irrelevantes en el nuevo contexto de uso. Por ejemplo, la eliminación de componentes innecesarios de la interfaz gráfica debido a las limitaciones de la pantalla de un PDA es una técnica frecuente.

- **Reorganización** de los componentes de la interfaz de usuario, mediante una re-evaluación de su disposición espacial y/o su dependencia temporal. La reorganización puede ser consecuencia de la inserción y/o eliminación de componentes de interfaz de usuario.

La reorganización puede afectar a diferentes partes de interfaz gráfica de usuario en función de su tipo. Por ejemplo, la reorganización de la interfaz de usuario de un sitio Web puede afectar el estilo, el diseño, los contenidos, la estructura, la navegación, así como las capacidades de interacción del usuario. Todo esto en una posible forma independiente.

Por otro lado, el cambio de orientación vertical a horizontal o viceversa requiere una reorganización espacial única.

- **Substitución o transformación** de componentes de la interfaz gráfica de usuario. Es un caso particular de el reorganización, en la cual se lleva a cabo la sustitución de una interfaz gráfica de usuario por otra o de un componente gráfico por otro, i.e., un cuadro de lista completa se puede reducir a una lista desplegable si el usuario cambia de un PC a un PDA. Esta situación se presenta cuando los componentes de la interfaz gráfica de usuario son reemplazados por otros nuevos. Así mismo esta técnica se puede entender como como una combinación de las técnicas anteriores de **Eliminación** y de **Adición**.

Una degradación elegante consiste en una reorganización de la interfaz gráfica de usuario después de un cambio de plataforma de cómputo: de una plataforma menos limitada a una más limitada , e. g., de una PC a un PDA. Una mejora elegante el proceso inverso: se trata de la mejora de la interfaz de usuario cuando el usuario cambia de una plataforma restringida a una menos restringida.

### 2.2.3 *Proceso de adaptación plástica*

Clavary et al. [Calvary et al., 2001] proponen que la adaptación plástica sea estructurada de acuerdo con el siguiente proceso de cinco pasos:

1. **Detección de variaciones en el contexto de uso:** capta el entorno físico y modela las plataformas que serán soportadas.

2. **Identificación de las soluciones candidatas:** identifica las interfaces gráficas de usuario candidatas que sean adecuadas al nuevo contexto de uso. Dos técnicas de identificación son susceptibles de implementación: a) cálculo en tiempo de ejecución (*on the fly*); y b) selección de un conjunto precalculado de interfaces gráficas de usuario.
3. **Selección de una solución particular:** usa alguna estrategia de resolución de problemas para elegir la interfaz gráfica de usuario más adecuada.
4. **Transición entre estados:** realiza la transición de la interfaz gráfica de usuario actual hacia la nueva solución seleccionada. Aunque la transición entre estados ha sido analizada desde los primeros desarrollos en interfaces hombre-máquina (e.g. uso de animación gráfica para transferir la carga cognitiva a nivel perceptivo), este tema de investigación está abierto [Robertson et al., 1991].
5. **Ejecución de la nueva solución:** ejecuta la nueva interfaz gráfica de usuario mientras no ocurran las siguientes condiciones de adaptación. Dos técnicas de ejecución han sido identificadas, en donde la interfaz gráfica de usuario puede: a) ser ejecutada desde el principio o b) preservar el estado del sistema.

### 2.3 CONTEXTO DE USO

Un contexto se define como las condiciones generales o circunstancias en las cuales toma lugar una actividad, evento, acción, etc. [Porzel, 2011]. Por lo tanto, el contexto de uso en un sistema se refiere al conjunto de variables observables que caracterizan las condiciones en las que un sistema en particular se encuentra ejecutándose. [Sottet et al., 2008].

El desarrollo de una amplia variedad de dispositivos de acceso han engendrado nuevos requerimientos en el campo de investigación de la Interacción Hombre-Máquina, tales como la capacidad de los sistemas interactivos para ejecutarse en diferentes contextos de uso.

Al tratarse de un desarrollo móvil, el número de contextos de uso en los cuales puede estar inmerso un sistema se incrementa. Para el presente trabajo el contexto de uso va a estar delimitado por las características de *hardware* y *software* del dispositivo móvil en la cual se encuentre nuestro sistema ejecutándose.

Una **plataforma** se refiere a los dispositivos de *hardware* y *software* disponibles para sostener la interacción del usuario con el sistema [Calvary et al., 2004].

La plataforma puede ser modelada en términos de los recursos computacionales que determinan la forma en que se procesa, se transmite y se presenta la información, así como la manera en que el usuario manipula la información. Comúnmente el tamaño de la memoria, el ancho de banda de la red y la plataforma de interacción motivan la selección de un conjunto de modalidades de E/S y la cantidad de información disponible [Calvary et al., 2001].

## 2.4 SISTEMAS MÓVILES

La movilidad es la parte primordial de un sistema móvil y nos va a permitir tener acceso a nuestra información en cualquier momento y en cualquier lugar.

Un sistema móvil se va a caracterizar por la **condición de movilidad**, la cual es el conjunto de características que distinguen al usuario móvil del usuario de un sistema de cómputo estacionario típico.

Existen cuatro componentes fundamentales en el problema de movilidad: **el usuario, el dispositivo, la aplicación y la red móvil.**

### ■ Usuario Móvil:

El usuario móvil se caracteriza por:

1. Estar en movimiento, al menos ocasionalmente, entre ubicaciones conocidas o desconocidas.
2. No se focalizar en la tarea de cómputo.
3. Requerir alto grado de interactividad y tiempo de respuesta corto de un sistema.
4. Cambiar de tareas frecuente o abruptamente.
5. Requerir de acceso a la información en cualquier momento y lugar.

### ■ Dispositivos móviles inteligentes:

Los dispositivos inteligentes se caracterizan por la habilidad de ejecutar múltiples aplicaciones, en algunos casos de manera concurrente, i.e., soportar diferentes grados de movilidad y de configuración, ejecutar servicios de acceso remoto a datos de manera intermitente y operar de acuerdo con sus limitados recursos locales.

Estos dispositivos se encuentran bajo el control de usuarios individuales, i.e., su propiedad, operación, configuración y control son exclusivos del usuario. Existen una gran cantidad de dispositivos inteligentes, e.g. computadoras personales, teléfonos, cámaras, consolas de juego y otros dispositivos periféricos como discos duros externos, impresoras, etc. [Poslad, 2009].

El diseño de servicios de movilidad se basa en el desarrollo de dispositivos inteligentes y servicios, pero con una variante más especializada con el fin de simplificar el acceso de las aplicaciones y usuarios y la movilidad, particularmente, permite servicios tales como: localización (GPS), enrutamiento de datos para receptores móviles, asignación de dirección IP, los cuales deben estar diseñados para ser más transparentes para las aplicaciones y los usuarios [Fling, 2009].

- **Aplicación:**

Consiste en el software mediante el cual el usuario interactúa con el dispositivo móvil para llevar a cabo una tarea específica. Existen dos tipos de aplicaciones: **1) verticales**, orientadas a un dominio de aplicación específico y **2) horizontales** las cuales se caracterizan por ser amplias e independientes del dominio que sirven a mercado en masa.

- **Red móvil:**

Los dispositivos móviles pueden utilizar una amplia gama de protocolos de comunicaciones inalámbricas heterogéneas, i.e., celular (GSM, TDMA y CDMA), satelital, Bluetooth, Wi-Fi, etc.

Gracias a los dispositivos móviles inteligentes, los usuarios pueden tener acceso directo a su información, de acuerdo a sus preferencias personales. La ventaja de los dispositivos móviles inteligentes es que son conscientes de su contexto de ubicación y tienen la capacidad de adaptarse a él.

#### 2.4.1 Dimensiones de la movilidad

Es el conjunto de propiedades que distinguen al sistema de cómputo móvil del sistema de cómputo estacionario. En la fig. 6 se pueden observar las dimensiones que la movilidad agrega a un sistema de cómputo [B'Far, 2005].

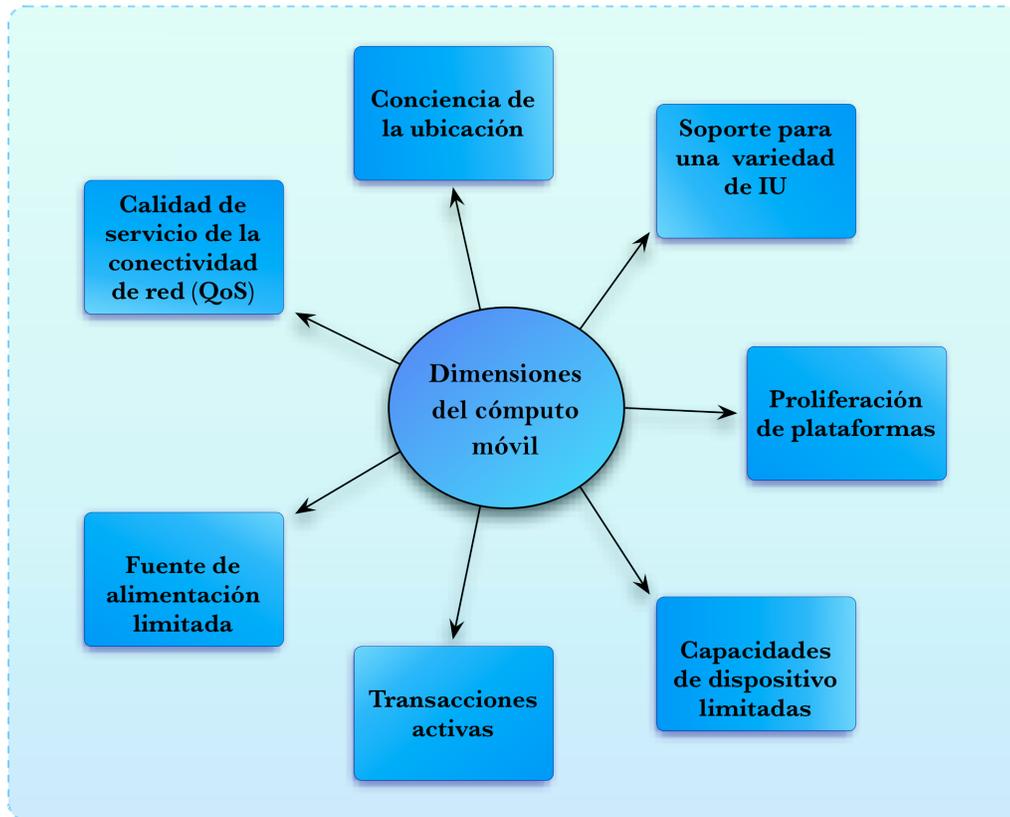


Figura 6: Dimensiones móviles

A continuación se describe cada una de las dimensiones que abarcan en el cómputo móvil [B'Far, 2005]:

- **Conciencia de la ubicación:** la movilidad conlleva a un cambio constante de localización, por lo cual un dispositivo móvil debe tener la capacidad de determinar su posición en cualquier momento y lugar.
- **Calidad de servicio de la conectividad de red (QoS):** el medio de intercambio de información en los dispositivos móviles se basa en redes inalámbricas, por tal motivo el dispositivo móvil experimenta cambios constantes su red de comunicación, i.e., al cambiar de posición es necesario ajustarse a los cambios presentes en la red como constantes conexiones y desconexiones. Pero siempre tratar de garantizar al usuario una calidad de servicio en su conectividad al tratar de disminuir las eventualidades en la red.

- **Capacidades limitadas de dispositivos:** debido al tamaño reducido de los dispositivos móviles, sus componentes de *hardware* del dispositivo se ven reducidos en capacidad, e.g., capacidad limitada de procesamiento y de memoria (complejidad y tamaño reducido de programas).
- **Fuente de alimentación limitada:** existe un consumo de energía restringido, debido al límite de energía que puede aportar una fuente de alimentación pequeña.
- **Soporte para una amplia variedad de interfaces de usuario:** debido a que un dispositivo móvil puede tener diferentes medios de entrada y salida de información, se debe brindar al usuario un soporte para todas las tecnologías que puedan existir en el dispositivo, así mismo de las propiedades de sus componentes.
- **Proliferación de plataformas :** en la actualidad no se cuenta con un solo sistema operativo móvil en el mercado, por lo cual cada fabricante incorpora el sistema más adecuado a su producto.
- **Transacciones activas:** el dispositivo móvil incorpora la capacidad de poder ejecutar tareas sin la intervención del usuario.

Este trabajo solo tendrá presente tres de estas dimensiones: 1) **soporte para una amplia variedad de interfaces de usuario**, en la cual nos centraremos en proponer un modelo para adaptar interfaces gráficas de usuario, 2) **capacidades de dispositivo limitadas**, esta dimensión permite determinar las propiedades *hardware* del dispositivo móvil, i.e., el contexto de uso y 3) **proliferación de plataformas**, la cual permite extender la adaptación a más de una plataforma móvil.

## 2.5 INGENIERÍA ORIENTADA A MODELOS (MDE)

Después de dar a conocer las áreas que se involucran en este trabajo, es necesario establecer un procedimiento que permita involucrar cada una de estas áreas, con el objetivo de plantear una solución. Para ello nos auxiliamos de la ingeniería orientada a modelos conocida como MDE (del inglés (Model-driven engineering)).

La ingeniería orientada a modelos es un paradigma de desarrollo de software que se centra en la creación y explotación de modelos de dominio, i.e., en representaciones abstractas de los conocimientos y actividades que rigen un dominio de aplicación particular.

La ingeniería orientada a modelos proporciona un conjunto de herramientas para unificar las tecnologías de modelado y para expresar los conceptos necesarios para su uso y comprensión. Constituye un enfoque de desarrollo que aprovecha los aspectos descriptivos y productivos de los modelos [Gauffre and Dubois, 2011].

La motivación para la ingeniería orientada a modelos es la integración de los conocimientos y técnicas desarrolladas en la ingeniería de software, utilizando las nociones de modelo y de transformación del modelo y de mapeo como los conceptos clave [Coutaz and Calvary, 2008].

### *Modelos*

Un modelo es una representación de un objeto (i.e., un sistema), con un propósito específico, i.e., "ser capaz responder a preguntas concretas en lugar del objeto de estudio actual" [Bézivin, 2001]. Por lo tanto, un modelo construido para hacer frente a un aspecto específico de un problema, es por definición una simplificación del objeto que se encuentra bajo estudio.

A continuación, se da a conocer los usos actuales de dos aspectos del MDE en el diseño de interfaces de usuario [Gauffre and Dubois, 2011]:

- **Soporte descriptivo:** los modelos tienen la capacidad de describir aspectos específicos de un sistema. Rothenberg menciona que "un modelo representa una realidad para un determinado propósito; el modelo es una abstracción de la realidad en el sentido que no se pueden representar todos los aspectos de la realidad. Esto nos permite tratar con el mundo en una manera simplificada, evitando la complejidad, el peligro y la irreversibilidad de la realidad" [Rothenberg, 1989]. Debido a la cantidad de conceptos que se pueden llegar a abordar, surge la necesidad de su representación comprensiva para entenderlos y manipularlos. La ingeniería orientada a modelos presenta tres beneficios potenciales para ello:
  - **Caracterización:** un modelo contribuye a identificar y caracterizar claramente los distintos elementos que constituyen el sistema.
  - **Documentación:** el modelo apoya la generación de la documentación del diseño.

- **Evolución:** las herramientas de la ingeniería orientada a modelos soportan la evolución de un modelo, i.e., facilita nuevas mejoras y evoluciones de un modelo, al permitir su rápida edición gráfica y una descomposición clara de sus elementos constitutivos.
- **Soporte productivo:** diferentes modelos que expresan diferentes consideraciones y/o que están relacionados a diferentes niveles de abstracción, son requeridos en el desarrollo de sistemas, especialmente aquellos que necesitan tener en cuenta las propiedades de dispositivos, las *APIs* disponibles, las características de los objetos físicos, la estructura del software, la definición de componentes, etc. La ingeniería orientada a modelos presenta dos ventajas importantes para apoyar a estas consideraciones:
  - **Articulación:** la ingeniería orientada a modelos ofrece herramientas de soporte para las transformaciones de modelos contribuyendo a la articulación de diferentes facetas de diseño.
  - **Proceso:** la ingeniería orientada a modelos también ofrece soporte para llevar a cabo transformaciones para vincular modelos espesados en diferentes niveles de abstracción del desarrollo. Esta segunda forma de transformación admite la progresión a lo largo de las diferentes etapas de desarrollo.

## 2.6 TRABAJOS RELACIONADOS

Hasta el momento se han propuesto diferentes enfoques para la adaptación dinámica de interfaces gráficas de usuario con base en modelos, tales como roles y tareas, flujos de datos de entrada y salida, así como anotaciones o la composición de componentes visuales.

A continuación, describimos algunos de los principales trabajos que involucran MDE y plasticidad para describir sistemas de adaptación de interfaces de usuario:

2.6.1 *Modelo de asignación: un primer paso para asegurar usabilidad y sustentabilidad de plasticidad de interfaces gráficas de usuario*

En tiempo de ejecución, un sistema interactivo es un conjunto de gráficas de modelos que expresan diferentes aspectos del sistema a múltiples niveles de abstracción [Coutaz, 2010].

En la fig. 7 se representan un sistema interactivo, i.e., un gráfica de modelos, en los cuales se puede incluir un modelo de trabajo, un modelo de concepto, un modelo abstracto de la interfaz gráfica de usuario (expresado en términos de espacios de trabajo) y un modelo concreto de interfaz de usuario (expresado en términos de interactores), todas ellas unidas por asignaciones [Coutaz et al., 2007].

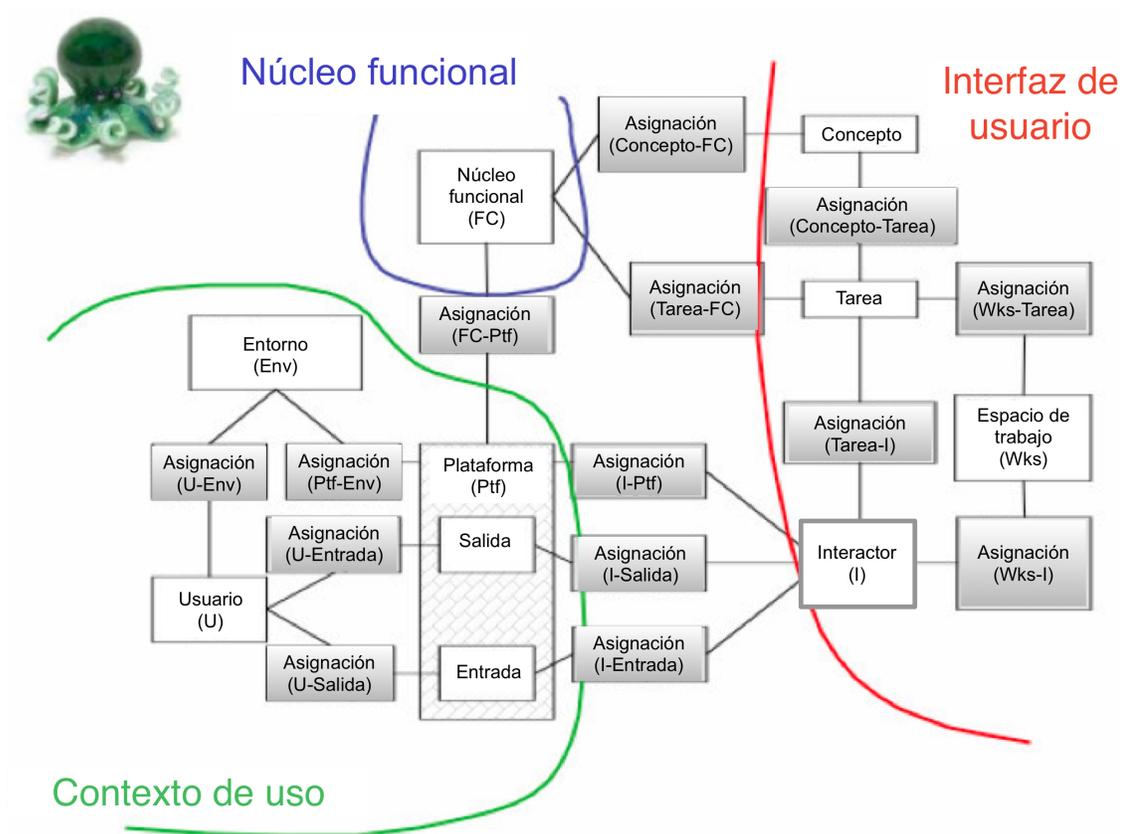


Figura 7: Modelo de asignación [Vanderdonckt et al., 2006].

Tanto tareas como conceptos se asignan a las entidades del núcleo funcional del sistema interactivo, mientras que los interactores de la interfaz gráfica de usuario concreta, se asignan a entradas y salidas en los dispositivos. Las asignaciones entre los interactores y dispositivos de e/s permiten la expresión explícita de interfaces de usuario centralizadas y distribuidas.

En el enfoque tradicional basado en modelos para la generación de la interfaz gráfica de usuario, las reglas de transformación se diluyen dentro de la herramienta, i.e., las transformaciones y las asignaciones se vuelven parte del modelo. En consecuencia, la conexión entre la especificación y el resultado final puede ser muy difícil de controlar y de comprender [Coutaz et al., 2007].

### 2.6.2 *Un enfoque MDE-SOA para soportar interfaces de usuario plásticas en espacios de ambiente*

Como se muestra en la fig. 8, la descomposición funcional del *middleware* que soporta la adaptación incluye [Coutaz, 2010]:

- Una infraestructura de contexto que crea y mantiene un contexto de uso.
- Un sintetizador de la situación que determina la situación e informa de un posible proceso de evolución en el momento en que ocurra una nueva situación.
- Un productor de la adaptación que implementa el plan de adaptación producido por el motor de evolución.

Como el enfoque de descomposición es usado comúnmente para el desarrollo automático de sistemas, en este modelo para adaptar la descomposición de interfaces de usuario se propuso:

- El usuario final se mantiene en el ciclo: la reacción a una situación nueva puede ser una mezcla de las especificaciones proporcionadas por los desarrolladores o aprendidas por el motor de evolución.
- Los componentes mencionados en el plan de acción no necesariamente existen como código ejecutable.

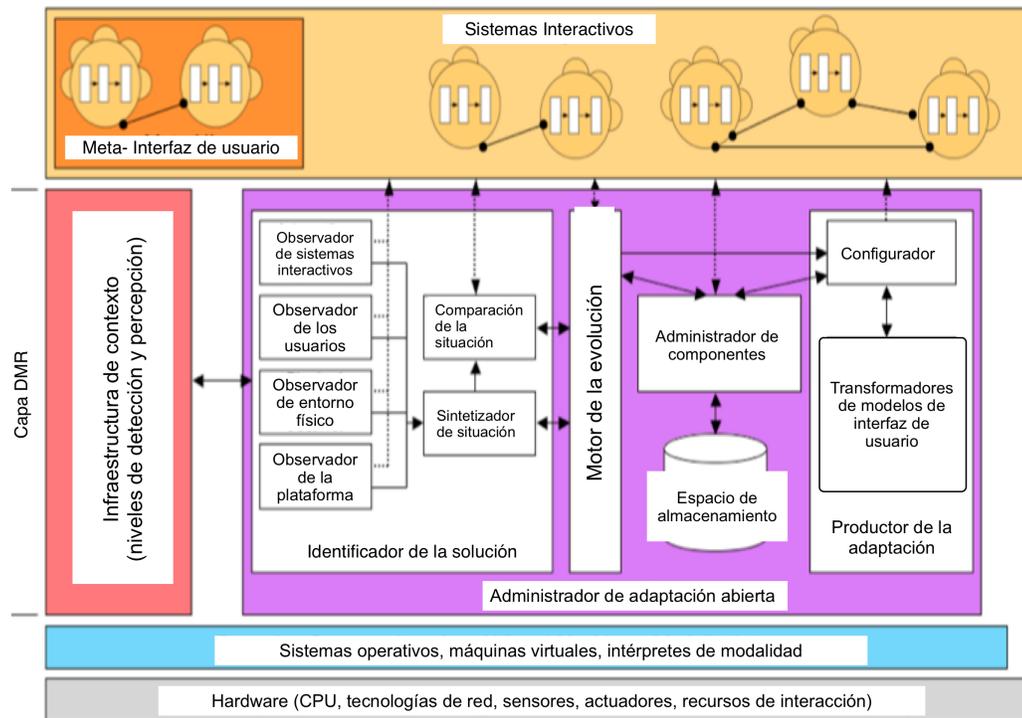


Figura 8: Enfoque MDE-SOA para el soporte de interfaces de usuario plásticas en espacios de ambiente [Coutaz et al., 2007].

### 2.6.3 Un proceso de desarrollo para interfaces de usuario plásticas

Como se muestra en la fig. 9, el *framework* [Thevenin et al., 2002]:

- Se basa en los modelos más conocidos, como el modelo de dominio y el modelo de tareas, pero mejora al adaptarse a la variación del contexto de uso.
- Explícitamente, presenta nuevos modelos y métodos heurísticos que han sido pasados por alto o ignorados hasta ahora para expresar los diferentes contextos de uso: la plataforma, interactores, el medio ambiente y los modelos de evolución.

En la fig. 9 se muestra el proceso de adaptación de la interfaz de usuario cuando se aplica a dos contextos distintos: contexto 1 y contexto 2.

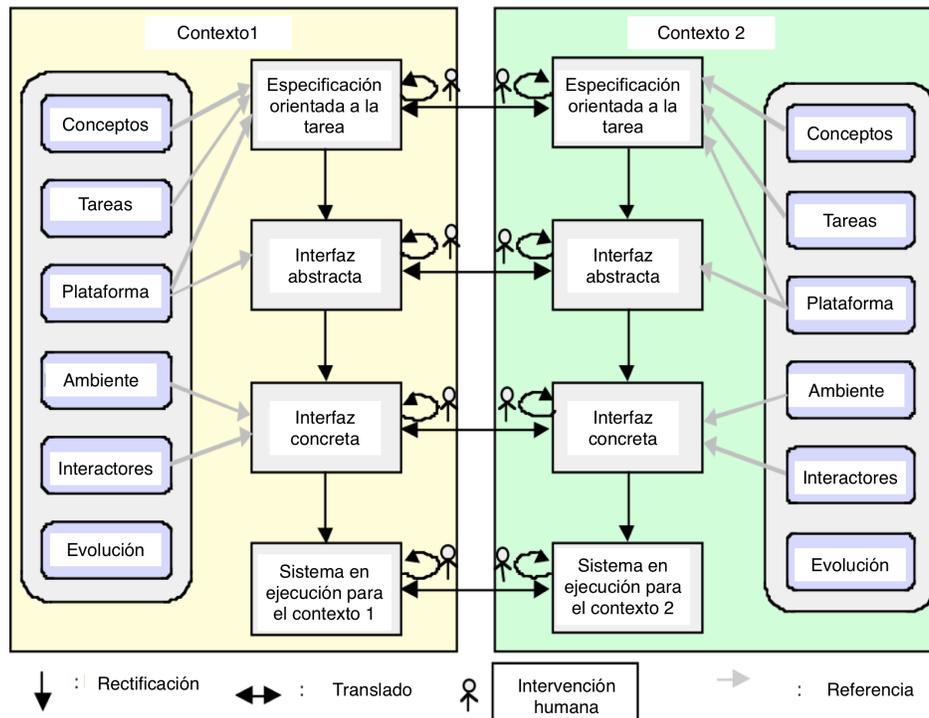


Figura 9: Un framework basado en modelo para plasticidad de interfaces de usuario [Thevenin et al., 2002]

El modelo de evolución especifica el cambio de estado dentro de un contexto, así como las condiciones para entrar y salir de un contexto particular. El modelo de interactores describe los *widgets* de recursos multimodales disponibles para la producción de la interfaz de usuario concreta.

El proceso es una combinación de una rectificación vertical y una translación horizontal. La rectificación vertical cubre el proceso de derivación, i.e., descripciones del contenido, correspondiente a la translación entre modelos al mismo nivel de rectificación.

La rectificación y translación pueden ser llevadas a cabo automáticamente a partir de las especificaciones o manualmente por personas expertas. La rectificación y translación se pueden combinar en multiples formas dando lugar a múltiples instancias del *framework* de referencia.

2.6.4 Un framework de MDE para interfaces de usuario plásticas

El *framework* presentado en la fig. 10 tiene las siguientes características:

- Reúne a los modelos productivos de trabajo con los diagramas de clases UML para cubrir los servicios dependientes del dominio y conceptos.
- Pasa de la descripción contemplativa del contexto de uso a términos del espacio ontológico.

El nivel M2 está compuesto por un conjunto de meta-modelos (y relaciones), que especifican la estructura de los conceptos importantes en el espacio del problema. Mientras que el nivel M1 es llenado con múltiples modelos donde cada uno proporciona una importante perspectiva sobre un determinado sistema interactivo.

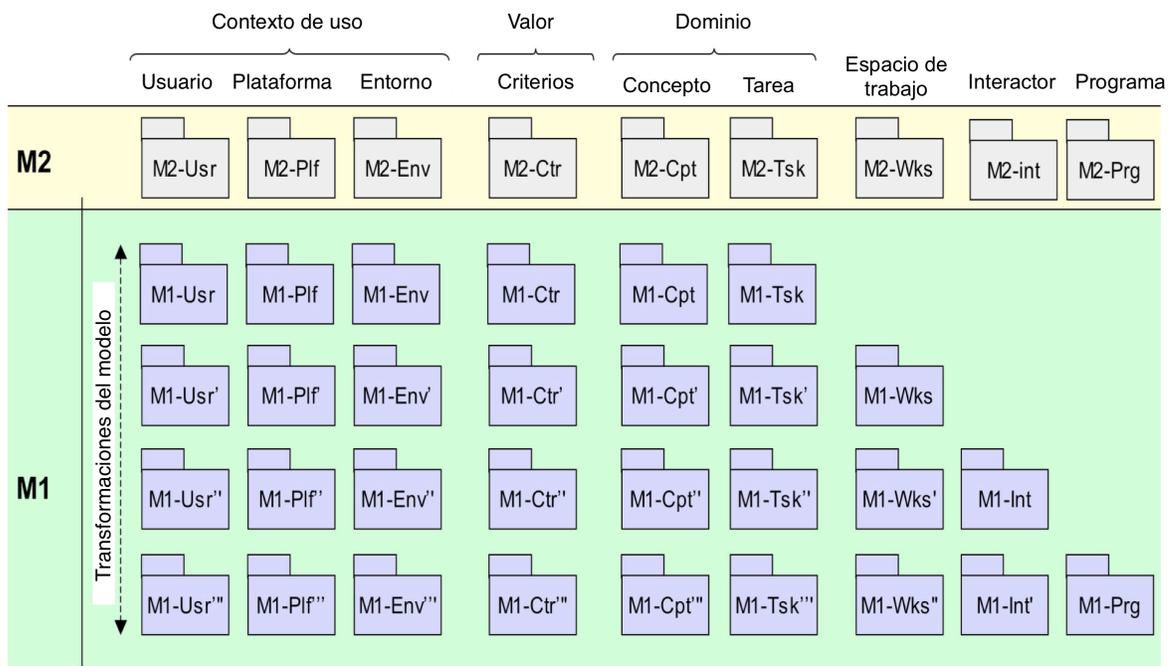


Figura 10: MDE Framework para interfaces gráficas de usuario plásticas [Coutaz and Calvary, 2008]

El conjunto de columnas representa diferentes perspectivas sobre una interfaz gráfica de usuario en particular. En una columna todos los modelos conforman el mismo meta-model. Una línea corresponde al resultado de un paso en la transformación [Coutaz and Calvary, 2008].

El marco hace explícita la necesidad de revisar los modelos durante el proceso de ingeniería para adaptarse a las nuevas restricciones.

Por ejemplo, un modelo de tarea inicial ( $M_1\text{-Tsk}$ ) puede ser ajustado a otra versión de la tarea ( $M_1\text{-Tsk}'$ ), donde se sustituye una tarea, intercalando el operador con un operador de secuencia para hacer frente a una pantalla de tamaño pequeño.

El enfoque convencional en la generación de la interfaz gráfica de usuario es empezar desde la producción de un modelo de tarea  $M_1$  y un modelo de concepto  $M_1$ . A continuación, se debe aplicar transformaciones verticales sucesivas hasta la generación de un programa  $M_1$ .

En este enfoque, todos los modelos de nivel  $M_1$  deben mantener una asignación explícita entre ellos. Ya que se encuentran vivos en tiempo de ejecución, cualquier modelo puede ser utilizado sobre la marcha para informar sobre el proceso de adaptación. Por lo tanto, este enfoque hace que sea posible combinar las transformaciones en tiempo de ejecución, mezclando transformaciones de abajo hacia arriba, para alcanzar el nivel adecuado de abstracción.



---

## ENTORNO MÓVIL

---

El presente capítulo se divide en tres secciones. En la sección 3.1.2 se aborda las propiedades gráficas del *hardware* móvil, i.e., tamaño, resolución y densidad de pantallas de dispositivos móviles. En la sección 3.2 se da a conocer una panorámica de las interfaces gráficas en dispositivos móviles; en particular se describe algunos principios fundamentales de este enfoque y pautas para su correcto diseño e implementación. Finalmente en la sección 3.3 se explica los tipos de desarrollo software para dispositivos móviles.

### 3.1 HARDWARE GRÁFICO MÓVIL

Debido a que este trabajo de tesis se centra en el estudio de interfaces gráficas de usuario, describiremos exclusivamente el funcionamiento del hardware gráfico en los dispositivos móviles.

Los dispositivos móviles han evolucionado a un ritmo acelerado en los últimos años. Para satisfacer las diversas necesidades de los usuarios se han creado una gran gama de estos dispositivos, i.e., el asistente digital personal (*PDA*), el navegador móvil, el reproductor multimedia personal (*PMP*), el teléfono celular, la *tablet*, el teléfono inteligente (*smartphones*), etc.

Existen varias formas de clasificar a los dispositivos móviles. Para los fines de este trabajo los clasificaremos en dos grupos: los de gama baja y los de alto rendimiento. Los dispositivos de gama baja solo cuentan con un único procesador para llevar a cabo todas las tareas del dispositivo, mientras que dispositivos de alto rendimiento cuentan con al menos dos procesadores, uno responsable de las operaciones de tiempo real y los otro(s) dedicado(s) a operaciones multimedia.

En la fig. 11 se puede apreciar la arquitectura *hardware* para un dispositivo móvil de alto rendimiento, e.g., un *smartphone* [Woo et al., 2010]. De acuerdo a esta arquitectura, un dispositivo móvil se compone de: un procesador central, un sistema de memorias (*DRAM* y memoria Flash), un procesador de aplicación para el procesamiento multimedia y un controlador de entrada/salida.

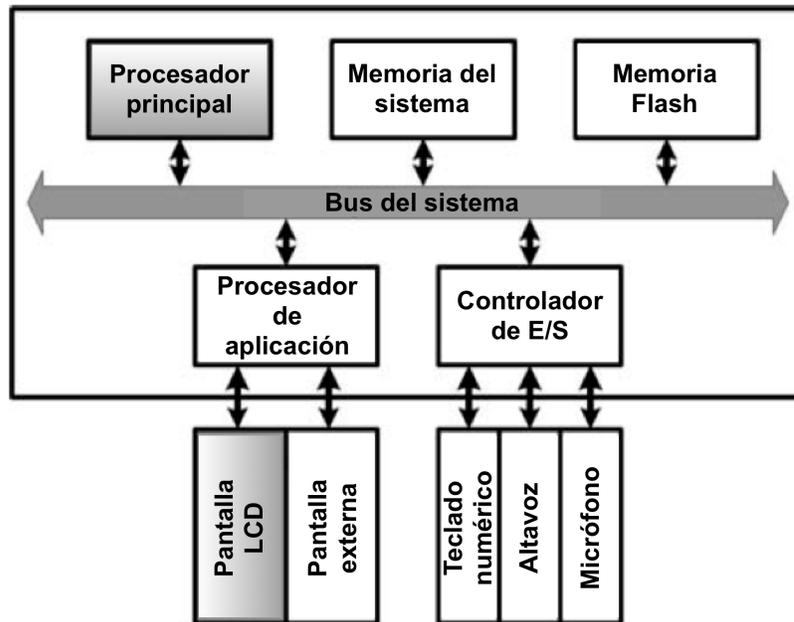


Figura 11: Arquitectura *hardware* en dispositivos móviles de alto rendimiento [Woo et al., 2010].

### 3.1.1 Procesadores en dispositivos móviles

Como se puede observar en la fig. 12, los dispositivos móviles pueden tener uno o más procesadores. Cada procesador incorpora una unidad central de procesamiento <sup>1</sup>, un procesador digital de señales <sup>2</sup> y tal vez un hardware dedicado para audio, imágenes, gráficos y otras tareas.

<sup>1</sup> CPU del inglés *Central Processing Unit*

<sup>2</sup> DSP del inglés *Digital Signal Processor*

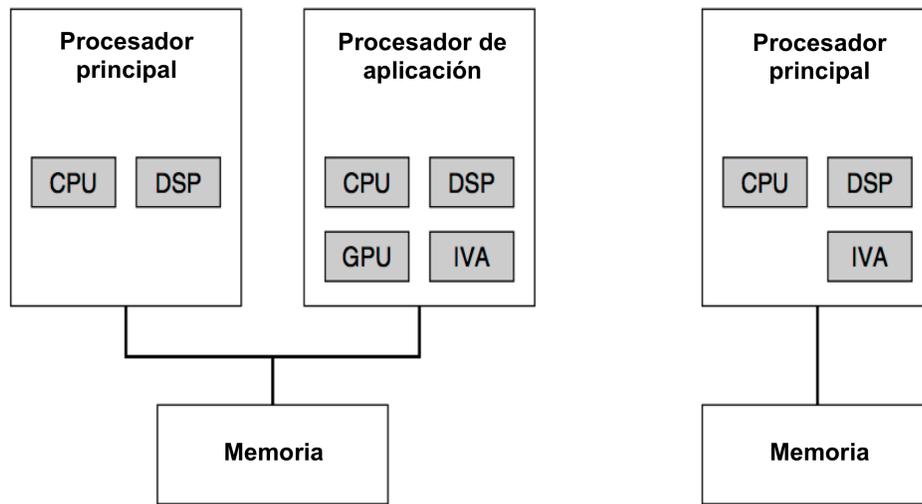


Figura 12: Configuración de los procesadores en dispositivos móviles [Pulli et al., 2008].

El procesador principal se encarga de llevar a cabo operaciones fundamentales tales como operaciones del sistema en tiempo real, i.e., procesamiento de las señales de voz y radio.

En dispositivos de gama baja, el procesador va a encargarse de ejecutar el sistema operativo, las aplicaciones y la interfaz de usuario [Pulli et al., 2008], vease fig. 12, diagrama derecho.

Un dispositivo de alto rendimiento tiene un o más procesadores de aplicación para fines específicos, e.g., video, audio, etc. En la fig. 12 en el diagrama izquierdo, se puede observar la arquitectura de un dispositivo móvil de alto rendimiento con un procesador principal y un procesador de aplicaciones.

El procesador de aplicaciones está a cargo de las operaciones multimedia de alto rendimiento. Para aumentar la potencia de cálculo, los procesadores más nuevos emplean arquitectura multi-núcleo.

Como se observa en el diagrama izquierdo de la fig. 12, el procesador de aplicaciones puede contar adicionalmente con una unidad de procesamiento gráfico<sup>3</sup> y/o acelerador de audio y video<sup>4</sup>, las cuales tienen como función encargarse del procesamiento gráfico necesario para desplegar la interfaz gráfica de usuario en la pantalla del dispositivo.

### 3.1.2 Pantallas de dispositivos móviles

Las pantallas de los dispositivos móviles juegan un papel muy importante en este trabajo, ya que son el medio de entrada y salida mediante el cual las interfaces gráficas interactúan con el usuario, además de ser parte del contexto de uso requerido para llevar a cabo una correcta adaptación de estas interfaces.

Las pantallas de los dispositivos móviles pueden variar de acuerdo a las siguientes propiedades: tamaño, resolución y densidad de píxel [Hooper and Berkman, 2012].

A continuación, se describen las principales propiedades en las pantallas de los dispositivos móviles:

- **Tamaño de pantalla:** el tamaño de la pantalla varía de acuerdo al tipo de dispositivo móvil. Se determina al medir la diagonal entre las dos esquinas de la pantalla.
- **Resolución de pantalla:** esta propiedad es determinada por el número total de píxeles que pueden ser mostrados en la pantalla.
- **Densidad de píxel:** densidad de píxeles (ppi) se basa en la resolución de la pantalla. Es igual al número de píxeles en un área.

En la fig. 13, se puede apreciar gráficamente la diferencia entre los tamaños de las principales resoluciones de pantalla en *smartphones*.

Cuando se habla de resoluciones de pantalla, no es tan simple como pensar en el número de píxeles que la pantalla ofrece.

---

<sup>3</sup> GPU del inglés *Graphics Processing Unit*

<sup>4</sup> IVA del inglés *Imaging and Video Accelerator*

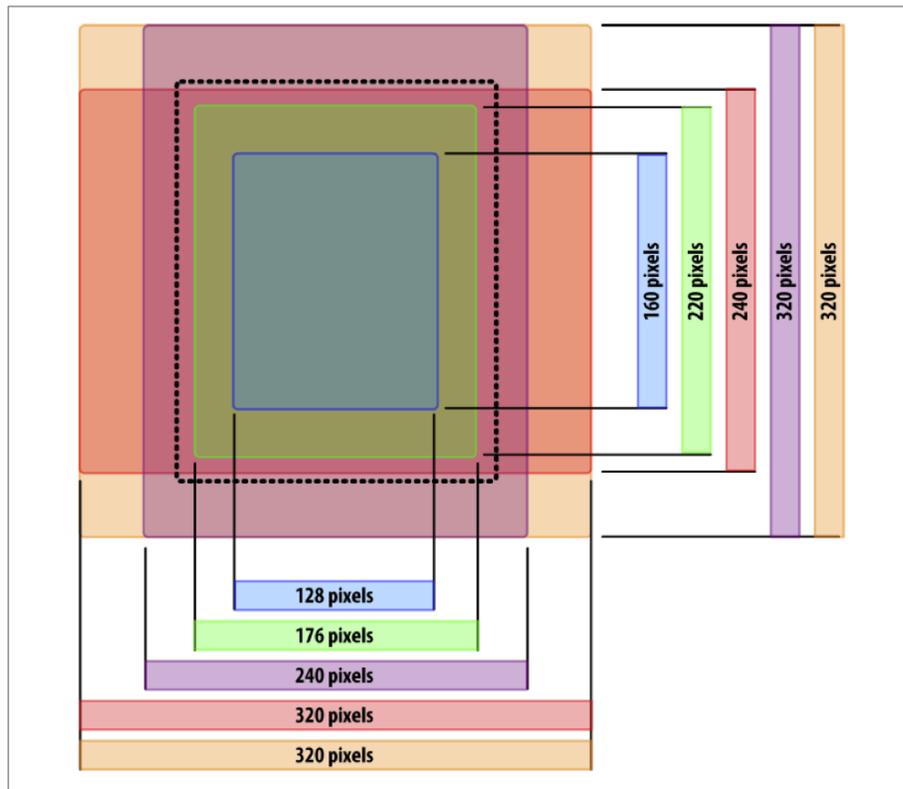


Figura 13: Principales resoluciones de pantalla en dispositivos móviles [Fling, 2009].

Hay que pensar también en las dimensiones físicas de la pantalla, i.e., se puede tener dos dispositivos que tienen una resolución de  $480 \times 800$  px, pero un dispositivo podría tener un tamaño de pantalla de  $3 \times 5.7$  otro puede tener  $2 \times 3.3$  de tamaño. Ambos dispositivos tienen el mismo número de píxeles, sin embargo el tamaño de pantalla del primer dispositivo es 1.5 veces el tamaño de pantalla del segundo.

Las principales resoluciones de pantalla (en píxeles) en dispositivos móviles son las siguientes:

- Pequeño: 128, 176, 208, 220.
- Mediano: 240, 230.
- Largo: 320, 360, 480.
- Tablet: 600, 800, 768, 1024.

En la tabla 1, se pueden apreciar las principales resoluciones de pantalla en dispositivos móviles en el mercado. En ella se clasifican algunos dispositivos móviles de acuerdo al tipo de resolución de sus pantallas y de acuerdo a su tipo de plataforma.

Resolución	Dispositivos
320 x 240	<b>Dispositivos Blackberry:</b> Curve 8530, Pearl Flip. <b>Dispositivos Android:</b> Motorola Charm. <b>Sony Ericsson:</b> Xperia X10 Mini. <b>Dispositivos Symbian OS:</b> Nokia E63.
320 x 480	<b>Dispositivos Apple OS:</b> iPhone, iPod. <b>Dispositivos Android:</b> HTC Dream, HTC Hero, Droid Pro, i7500 Galaxy, Samsung Moment.
360 x 640	<b>Dispositivos Symbian OS:</b> Nokia N8, Nokia C6-01.
480 x 360	<b>Dispositivos Blackberry:</b> Torch, Storm, Bold
480 x 800	<b>Dispositivos Android:</b> Liquid A1, HTC Desire, Nexus One, i9000, Galaxy S. <b>Dispositivos Maemo (Linux):</b> Nokia 900. <b>Dispositivos Windows Mobile 6:</b> Sharp So1SH. <b>Dispositivos Windows 7 Phone:</b> Venue Pro, Samsung Omnia 7, HTC 7 Pro.
768 x 1024	<b>Dispositivos Android:</b> iPad
640 x 960	<b>Dispositivos Android:</b> iPhone 4
1280 x 800	<b>Dispositivos Android:</b> Motorola Xoom, Samsung Galaxy Tab 10.1 <b>Dispositivos Windows OS:</b> Asus Eee Pad EP121

Cuadro 1: Principales resoluciones de pantalla en dispositivos móviles comerciales

En el diseño de las pantallas de dispositivos móviles, es importante tener en cuenta la densidad de píxeles del dispositivo. Las imágenes y los componentes que están diseñados para determinadas altura y anchura en píxeles aparecen más grandes en una pantalla de baja densidad y más pequeños en una pantalla de alta densidad. Los objetos que se vuelven demasiado pequeños afectarán la claridad, legibilidad y la detección de los detalles.

*Tecnologías en pantallas de dispositivos móviles*

Los dispositivos móviles utilizan una diversa gama de tecnologías de pantalla. Algunos dispositivos pueden estar integrados por varios tipos de hardware. Cada tecnología puede servir a un propósito único, e.g., pantalla principal, funcionalidad de retroiluminación o indicadores parpadeantes [Hooper and Berkman, 2012].

Es importante entender que cada una de estas tecnologías tiene sus limitaciones en términos de duración de la batería del dispositivo, la vida útil del dispositivo o la restricción en los modos de orientación. Por lo tanto, tenemos que crear interfaces de usuario que pueden maximizar la experiencia del usuario en torno a estas limitaciones. A continuación se describe los principales tipos tecnologías usados en pantallas de dispositivos móviles [Hooper and Berkman, 2012]:

- **LED:** los diodos emisores de luz son una fuente de luz superconductor. Cuando el diodo está encendido, los electrones se mueven en el dispositivo y se recombinan con los huecos de electrones, provocando una liberación de fotones.

A continuación se describen sus ventajas y deventajas:

- Se utiliza en la iluminación de indicadores.
  - Sus beneficios incluyen: 1) bajo consumo de energía, 2) un tamaño pequeño y 3) una operación rápida y eficiente.
  - Sus limitaciones comprenden la capacidad de comunicar sólo una pieza de información a la vez, i.e., un solo canal, un bit de información.
- **OLED:** diodo orgánico de emisión de luz <sup>5</sup>, el cual contiene una película de semiconductor orgánico que emite luz en respuesta a una corriente eléctrica.

A continuación se describen sus ventajas y deventajas:

- Sus beneficios abarcan: 1) la capacidad de funcionar sin una luz de fondo, ya que se trata de una capa ultradelgada y ultrabrillante, 2) la posibilidad de lograr una elevada relación de contraste y 3) cuando está inactivo no produce luz o el uso de energía.

---

<sup>5</sup> OLED (acrónimo inglés de *Organic Light-Emitting Diode*)

- Sus limitaciones engloban: 1) la utilización de una gran cantidad de energía de la batería para mostrar una imagen (como el texto en negro) con un fondo blanco y 2) la eventual susceptibilidad de poder provocar quemaduras en su superficie con el tiempo.
- **AMOLED:** OLED de matriz activa <sup>6</sup>, consiste en un conjunto de píxeles OLED que se depositan o integran en una serie de transistores de película fina (*TFT*) para formar una matriz de píxeles.

Los píxeles OLED se iluminan cuando han sido activados eléctricamente, i.e., son controlados por los interruptores que regulan el flujo de corriente que se dirige a cada uno de los ellos. El *TFT* continuamente regula la corriente que fluye por cada uno de los píxeles para así caracterizar el píxel con el nivel de brillo que mostrará.

A continuación se describen sus ventajas y deventajas:

- Los beneficios incluyen: 1) la capacidad de funcionar sin una luz de fondo y 2) la posibilidad de ser utilizados en pantallas de gran tamaño.
  - Las limitaciones comprenden: 1) la visibilidad en resplandor y 2) luz directa del sol.
- **ePaper:** no emite luz, generalmente se basa en luz reflejada mediante la suspensión de partículas en un líquido. Su funcionamiento consiste en emitir una carga que permite que las partículas oscuras aumenten y se hagan visibles, i.e., denotar las zonas oscuras sobre una superficie más clara, al igual que tinta sobre papel.

A continuación se describen sus ventajas y deventajas:

- Los beneficios abarcan: 1) bajo consumo de energía, 2) reducción de brillo y 3) una elevada relación de contraste.
- Las limitaciones engloban: 1) una lenta tasa de actualización y 2) el hecho de que las tecnologías de color son apenas emergentes.

---

6 *Active Matrix OLED*

#### 3.2 INTERFACES GRÁFICAS EN DISPOSITIVOS MÓVILES

Las interfaces de usuario son difíciles de diseñar e implementar por las siguientes razones [Meyers, 1993]:

1. Se dificulta el aprendizaje en las tareas del usuario, i.e., las tareas y los dominios se vuelven complejos.
2. Es necesario establecer un balance entre los aspectos de diseño, i.e., estándares, diseño gráfico, escritura técnica, internacionalización, rendimiento, niveles de detalle, factores sociales y tiempo de implementación.
3. Las teorías y las directrices existentes no son suficientes.
4. El diseño interactivo se complica a medida que crece el número de interfaces gráficas de usuario.
5. Existen requisitos en tiempo real para el manejo de eventos de entrada. Es difícil probar el software de interfaz de usuario.

#### *Retos en el diseño móvil*

Aunque la funcionalidad de los dispositivos móviles ha mejorado, aun existen muchos desafíos en el modelado y la adecuación de sus componentes. En resumen, hay tres grandes retos [Woo et al., 2010]:

- **Dimensión física:** la principal limitación de los dispositivos móviles es sin duda su tamaño físico. Para mantener la portabilidad de un dispositivo éste debe limitar su dimensión física, i.e., mantener un rango alrededor de 5 pulgadas (12,5 cm), sin embargo la última gama alta de teléfonos inteligentes no excede de las 4 pulgadas (10.16 cm), lo cual significa que existe un espacio limitado en la placa base, por lo tanto los componentes deben ser diseñados con dimensiones reducidas.
- **Consumo de energía:** debido a que un dispositivo móvil funciona con una batería, el consumo de energía determina el tiempo disponible de funcionamiento. A medida que aumenta el rendimiento se consume más energía, debido a una frecuencia de reloj más rápida o a los bloques de *hardware* por los que está compuesto. Por lo tanto, el aumento en el tiempo de operación y la reducción el consumo de energía constituyen una tarea muy importante.

- Recursos del sistema:** los dispositivos móviles no cuentan con una gran cantidad de recursos en su sistema, debido a la dimensión física y al consumo de energía, i.e., no pueden utilizar un bus de sistema de gran anchura y ni memoria de alto rendimiento, tales como *DDR2* o *DDR3*. A pesar de estas limitaciones, los dispositivos móviles ofrecen un alto rendimiento para satisfacer las necesidades de los usuarios.

### 3.3 TIPOS DE DESARROLLO DE SOFTWARE PARA DISPOSITIVOS MÓVILES

El desarrollo de aplicaciones móviles implica una gran cantidad de retos. En el desarrollo de software para dispositivos móviles se necesita utilizar herramientas y librerías específicas, así como escribir en diferentes lenguajes para las diferentes plataformas. Cada plataforma cuenta con diferentes procesos y requerimientos para formar parte de su programa de desarrollo, una variedad de herramientas y librerías, así como con documentación seccionada en diferentes partes del proceso de desarrollo [Allen et al., 2010].

Cuando se requiere desarrollar software para un dispositivo móvil en específico, se debe tener en cuenta los siguientes elementos: dispositivo, plataforma y *frameworks*. Debido a la heterogeneidad de estos elementos, se tiene que uno de los retos tanto en cómputo móvil como ubicuo es poder desarrollar software para dispositivos con diferentes propiedades y plataformas, i.e., *cross-platform*, *multi-platform*. En la figura 14, podemos observar los enfoques de desarrollo de software para dispositivos móviles.

A continuación, se presenta un panorama general del desarrollo de software para dispositivos móviles, así como las características de cada enfoque.

#### 3.3.1 Aplicaciones nativas (*Uni-Platform*)

Este enfoque se fundamenta en desarrollar aplicaciones exclusivamente con las librerías y herramientas que ofrece cada uno de los sistemas operativos <sup>7</sup> móviles en el mercado, i.e., los dispositivos de Apple como son el iPhone y el iPad, que se basan en el sistema operativo iOS, se limitan al uso del SDK propietario y de su editor Xcode para el desarrollo de aplicaciones.

Cada aplicación debe ser codificada en el lenguaje *Objective-C* y su interfaz de usuario es definida de manera manual por el desarrollador. Ninguna aplicación desarrollada en esta plataforma podrá correr en algún otro dispositivo que contenga otro SO.

---

<sup>7</sup> OS acrónimo inglés de *Operating System*

### 3.3 TIPOS DE DESARROLLO DE SOFTWARE PARA DISPOSITIVOS MÓVILES

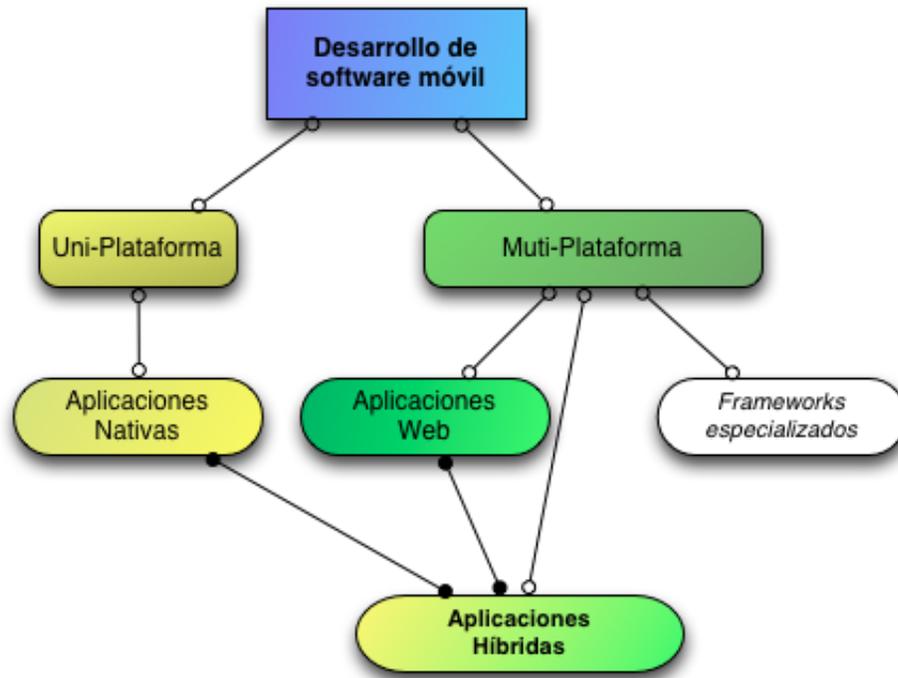


Figura 14: Enfoques de desarrollo de software para dispositivos móviles

Las aplicaciones nativas son instaladas en el dispositivo móvil en uso y acceden directamente a las características de hardware del dispositivo móvil, e.g., altavoces, acelerómetro, cámara, etc., lo cual permite aprovechar los recursos del dispositivo y mejorar el rendimiento de la aplicación software. Lo que las define es que éstas pueden ser adquiridas en el mercado de aplicaciones del sistema operativo en la que se desarrollaron [Stark, 2010].

Existen varias plataformas para dispositivos móviles en el mercado. En la tabla 2 se puede observar las principales plataformas, los lenguajes nativos que acepta cada una de ellas y algunos comentarios asociados [Virkus, 2011].

Cuadro 2: Principales plataformas para dispositivos móviles

OS	Lenguaje(s)	Comentarios
Android	Java, C	Sistema operativo de código abierto (Basado en Linux) <a href="http://developer.android.com">http://developer.android.com</a>
Bada	C, C++	Plataforma móvil de Samsung sobre Linux <a href="http://developer.bada.com">http://developer.bada.com</a>
Blackberry	Java	J2ME compatible <a href="http://us.blackberry.com/developers/">http://us.blackberry.com/developers/</a>
MeeGo	Qt, Web Apps, C++, Others	Sistema operativo de código abierto creado por Intel y Nokia <a href="http://meego.com/developers">http://meego.com/developers</a>
iOS	Objective-C, C	Requiere cuenta de desarrollo de Apple <a href="http://developer.apple.com/iphone">http://developer.apple.com/iphone</a>
Symbian	C, C++, Qt, WebApp, others	Sistema operativo de código abierto <a href="http://www.symbian.org">http://www.symbian.org</a>
Windows 7 Mobile	C#, C, XNA, Silverlight	Sistema operativo móvil basado en Windows 7 <a href="http://create.msdn.com/en-US/">http://create.msdn.com/en-US/</a>

### 3.3.2 Desarrollo multi-plataforma (cross-platform)

El desarrollo multi-plataforma tiene como objetivo desarrollar aplicaciones de software que puedan ser ejecutadas de manera transparente en cualquier plataforma.

El desarrollo multi-plataforma está influenciado por técnicas de desarrollo de software rápido, las cuales provienen del desarrollo de páginas Web. Las tres principales técnicas empleadas son:

- Diseño con lenguajes de marca (HTML/CSS).
- Uso de direcciones URL para identificar diseños de pantalla y estados visuales.
- Incorporación de lenguajes dinámicos, como Javascript y Ruby.

Diseñadores y desarrolladores de interfaces de usuario están influenciados por HTML y CSS para el diseño y la construcción de elementos visuales [Allen et al., 2010].

Al aprovechar estas habilidades se permite a un navegador Web integrado en el sistema operativo del dispositivo móvil, ser el mecanismo para la visualización de la interfaz de usuario de la aplicación.

#### *Frameworks para el desarrollo de aplicaciones*

Un *framework* para el desarrollo de aplicaciones consiste en un conjunto de librerías usadas por desarrolladores software para implementar la estructura de una aplicación en un entorno de desarrollo específico. Mediante una máquina virtual y/o alguna aplicación nativa, la aplicación desarrollada en el *framework* es traducida al lenguaje nativo y ejecutada.

**J2ME/JAVA ME** Las aplicaciones escritas en Java ME pueden ser ejecutadas en una gran cantidad de dispositivos basados en Java o que contengan la máquina virtual de esta plataforma. El hecho de que esta plataforma nos brinde un estandar en el desarrollo es una de sus fortalezas, sin embargo al mismo tiempo es una fuente de problemas de fragmentación. Actualmente, esta plataforma propietaria de Oracle, es el entorno de desarrollo más difundido [Allen et al., 2010].

**FLASH** Usa ActionScript, un lenguaje script propietario similar a JavaScript, para crear archivos ejecutables (SWF), que serán ejecutados sobre una variedad de dispositivos que soporten la plataforma Flash Lite o Flash Player 10. Es una plataforma propietaria de Adobe<sup>8</sup> [Fling, 2009].

#### *Websites/Aplicaciones Web*

Una aplicación Web móvil es una aplicación que está construida principalmente con tecnologías Web como HTML, CSS y JavaScript. Su diseño está dirigido específicamente a dispositivos móviles [Oehlman and Blanc, 2011].

---

<sup>8</sup> <http://www.adobe.com/products/flashlite/>

Se basa en desarrollar aplicaciones que se ejecuten directamente en el navegador Web de cada sistema operativo, i.e., los usuarios tienen que conectarse con un servidor Web, a través de una conexión de datos, para su uso y manipulación.

Las características principales de una aplicación Web son: la interfaz de usuario (GUI) está construida con tecnologías Web estándar. Esta se encuentra disponible en una dirección Web y es optimizada de acuerdo a las propiedades del dispositivo móvil [Stark, 2010].

### *Aplicaciones híbridas/Widgets*

Se trata de un enfoque que toma lo mejor de las aplicaciones nativas y aplicaciones Web. Para empezar se dice que es híbrido debido a que, por un lado, la aplicación se codifica en un lenguaje no nativo, pero mediante un software nativo se ejecuta directamente en el dispositivo móvil. Por otro lado puede permitir el acceso a aplicaciones Web sin conexión a internet, una vez descargadas, los usuarios pueden acceder a través de un icono en su escritorio, ya sea que encuentren en línea o sin conexión.

El principal objetivo de una aplicación híbrida es solventar las limitaciones que tiene una aplicación Web, e.i., acceder directamente al hardware del equipo móvil.

Para los fines del presente trabajo, analizamos las herramientas del *framework* PhoneGap. PhoneGap permite crear aplicaciones nativas con tecnologías Web y obtener acceso a las *APIs* nativas y a tiendas de aplicaciones. Este *framework* permite el uso de los principales lenguajes Web como HTML5, JavaScript y CSS para el desarrollo de aplicaciones multi-plataforma. Su código es libre, abierto y soporta los seis principales sistemas operativos móviles del mercado actual.

### *Aplicaciones nativas vs aplicaciones Web vs aplicaciones híbridas*

En la tabla 3, se muestran las principales ventajas y desventajas de los enfoques de programación para móviles. Como resultado de esta comparación, se seleccionó como solución viable el enfoque híbrido para el desarrollo *cross-plataform*.

### 3.3 TIPOS DE DESARROLLO DE SOFTWARE PARA DISPOSITIVOS MÓVILES

Cuadro 3: Tabla comparativa de los enfoques de desarrollo de software para dispositivos móviles

Enfoque	Ventajas	Desventajas
Aplicaciones nativas	Distribución de la aplicación a través de la tienda de la plataforma	Es necesario pagar por una cuenta para desarrollar en la plataforma
	Acceso a las propiedades físicas del equipo	La aplicación solo se ejecutará en equipos que tengan instalada la plataforma para la cual fue desarrollada
		Solo se puede desarrollar con las herramientas proporcionadas por el fabricante, e.g., lenguajes de programación y frameworks.
		El ciclo de desarrollo es lento
Aplicaciones Web	Los desarrolladores Web pueden utilizar sus herramientas de desarrollo actuales	No se puede acceder a todas las propiedades físicas del equipo
	Posibilidad de usar plantillas y diseños Web	No existe un método para distribución eficiente
	La aplicación se ejecutará en cualquier dispositivo con un navegador Web	Es difícil obtener efectos sofisticados en la interfaz gráfica de usuario
	Se puede corregir errores en tiempo real	
	El ciclo de desarrollo es rápido	
Aplicaciones híbridas	Todas las anteriores	Usar un <i>framework</i> de terceros.
		No hay documentación amplia



---

## MECANISMO MÓVIL PARA ADAPTACIÓN PLÁSTICA DE INTERFACES GRÁFICAS MULTI-PLATAFORMA

---

En este capítulo se describe la propuesta de un mecanismo plástico para el despliegue dinámico de interfaces gráficas multi-plataforma. En la sección 4.1 se abordan los fundamentos del mecanismo. En la sección 4.2 se da a conocer las principales propiedades y componentes de dicho mecanismo.

### 4.1 FUNDAMENTOS DEL MECANISMO

Como se ha dado a conocer en los capítulos previos, la detección, la adaptación y la provisión de un soporte para las diversas interfaces gráficas de múltiples dispositivos han sido algunos de los principales problemas en el diseño y desarrollo de software para dispositivos móviles.

Por cada dispositivo móvil se tiene la necesidad de: 1) encontrar las mejores interfaces gráficas de usuario para un sistema de software, 2) definir la arquitectura del sistema para dar cabida a las interfaces de usuario adecuadas, 3) aplicarlas y 4) tener en cuenta que una nueva interfaz gráfica de usuario puede ser requerida en cualquier momento.

Un mecanismo es un proceso cuya principal característica es regular la producción de cierto comportamiento. Por lo tanto, mediante el diseño del mecanismo que se describe en este capítulo, se pretende regular el desarrollo de software orientado a dispositivos móviles, con la particularidad de adaptación de las interfaces gráficas de usuario.

A continuación, se da a conocer los principales resultados teóricos que dieron origen al mecanismo de adaptación.

#### 4.1.1 *Movilidad vs plasticidad*

En los dominios de investigación del cómputo móvil y de la plasticidad de interfaces gráficas de usuario existen ciertas propiedades, denominadas dimensiones, que permiten diferenciar a cada área de estudio. Las dimensiones de la plasticidad se encuentran detalladas en la sección 2.2.1 y las de la movilidad en la sección 2.4.1. De estas dimensiones, seleccionamos las que permiten abordar el problema de la adaptación de interfaces gráficas de usuario en dispositivos móviles.

En primer lugar queremos introducir, como hemos establecido, una correlación entre las dimensiones plásticas y móviles. Como se puede apreciar en la fig. 15 esta correlación es el punto de partida para el desarrollo de nuestra propuesta.

En la fig. 15 se puede observar las siete dimensiones que conforman cada una de las áreas de estudio, sin embargo no todas ellas tienen una relación con la generación de la interfaz gráfica de usuario.

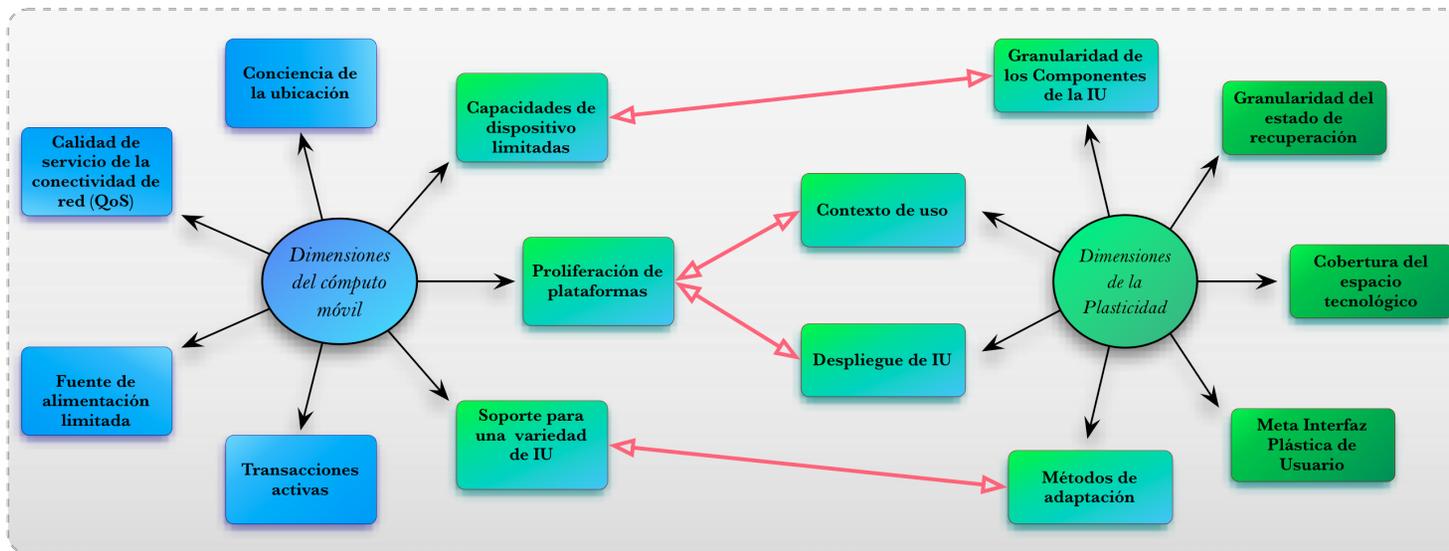


Figura 15: Correlación entre las dimensiones móviles y plásticas

Por lo tanto, en primer instancia se hace una selección de las dimensiones que involucran la correcta generación y despliegue de la interfaz gráfica de usuario. Respecto a las dimensiones móviles, existen tres de ellas que afectan directamente la generación de la interfaz gráfica de usuario en dispositivos móviles: **capacidades de dispositivos limitadas, proliferación de plataformas y soporte para una variedad de interfaces de usuario.** Estas dimensiones están asociadas al hardware del dispositivo móvil.

Respecto a las dimensiones de la plasticidad, estas se asocian a los requerimientos del software para llevar a cabo un correcto despliegue de la interfaz gráfica de usuario. Las dimensiones que hemos implicado en la presente investigación son: **granularidad de los componentes de la interfaz gráfica de usuario, contexto de uso, despliegue de la interfaz gráfica de usuario** y finalmente **métodos de adaptación.**

Ambas áreas de estudio permiten optimizar el correcto tratamiento de las interfaces gráficas de usuario, de acuerdo a las propiedades de hardware y software que implican la movilidad.

A continuación, se da a conocer los motivos y la forma en la cual se combinó y abordó cada una de las dimensiones relacionadas en la fig. 15:

- **Caso 1: capacidades limitadas de los dispositivos - granularidad de los componentes de la interfaz gráfica de usuario**

Como hemos mencionado anteriormente los dispositivos móviles tienen **capacidades limitadas** respecto a su *hardware*, i.e., las aplicaciones de software para dispositivos móviles tienen que ser diseñadas para optimizar el uso del almacenamiento de datos y el poder de procesamiento del dispositivo en términos del tipo de aplicación utilizada por el usuario. Principalmente se debe tener en cuenta los recursos necesarios de acuerdo con el procesamiento requerido para adaptar cada una de las interfaces gráficas de usuario.

Por lo tanto, esta dimensión está directamente relacionada con la **granularidad de los componentes de la interfaz gráfica de usuario**, la cual permite determinar el número de componentes de la interfaz de usuario que resulta más apropiado para llevar a cabo el proceso de adaptación.

Esta correlación nos permite establecer el apropiado uso de los recursos de hardware del dispositivo móvil en términos de la cantidad de componentes gráficos necesarios en las interfaces gráficas de usuario. Adicionalmente, esta correlación permite determinar si el dispositivo cuenta con los recursos de hardware necesarios para llevar a cabo el mencionado proceso de adaptación, sin afectar en su propio rendimiento.

- **Caso 2: proliferación de plataformas - contexto de uso - despliegue de interfaz gráfica de usuario**

La **proliferación de plataformas** refuerza la importancia de diseñar y desarrollar aplicaciones de *software* independientes de la plataforma de los dispositivos. Al definir las características de cada dispositivo como su perfil único, nos permite definir el tipo de adaptación más adecuado con base en esa información, i.e., se vincula directamente con la dimensión de la plasticidad denominada **contexto de uso**.

Su relación permite contextualizar un determinado dispositivo en función de sus características y llevar a cabo una correcta adaptación de las interfaces gráficas de usuario.

La **proliferación de plataformas** limita el desarrollo de software al sistema operativo del dispositivo. Por lo que, de acuerdo al enfoque de desarrollo que se tome, e.g., nativo, Web o híbrido, se llevará a cabo el despliegue de la interfaz gráfica de usuario. Esta propiedad la enlazamos con la dimensión llamada **despliegue de interfaz gráfica de usuario**, donde podemos seleccionar un tipo de despliegue de acuerdo al tipo de enfoque de desarrollo que se elija o de acuerdo a los tipos de dispositivos a los que va dirigido el desarrollo del *software*, e.g., estático o dinámico.

- **Caso 3: soporte para una amplia variedad de interfaces gráficas de usuario - métodos de adaptación**

Una interfaz gráfica de usuario es diseñada comunmente con ciertas propiedades para un dispositivo móvil específico. Sin embargo, la dimensión móvil nombrada **soporte para una amplia variedad de interfaces gráficas de usuario**, presenta el reto de extender una interfaz gráfica a diferentes dispositivos y con diferentes tecnologías de desarrollo, para llevar a cabo la adaptación de la interfaz gráfica. En este punto entra la vinculación con la dimensión plástica denominada **métodos de adaptación**, la cual permite dar la pauta para definir un conjunto de métodos y procedimientos para modificar las propiedades de la interfaz gráfica original con el fin de generar una interfaz óptima para el tipo de dispositivo que se encuentre en uso.

#### 4.1.2 *Espacio de diseño para interfaces gráficas de usuario en dispositivos móviles*

De acuerdo a la información que se analizó en capítulos previos, se ha modelado un espacio de diseño para la adaptación de interfaces gráficas de usuario en dispositivos móviles, en el cual se describen los principales aspectos requeridos para diseñar e implementar una interfaz gráfica de usuario. Este espacio de diseño se muestra en la fig. 16 y está compuesto de los siguientes elementos:

- **Contexto de uso:** se refiere a los parámetros de hardware y software que una aplicación toma en cuenta para llevar a cabo la adaptación de la interfaz gráfica de usuario.

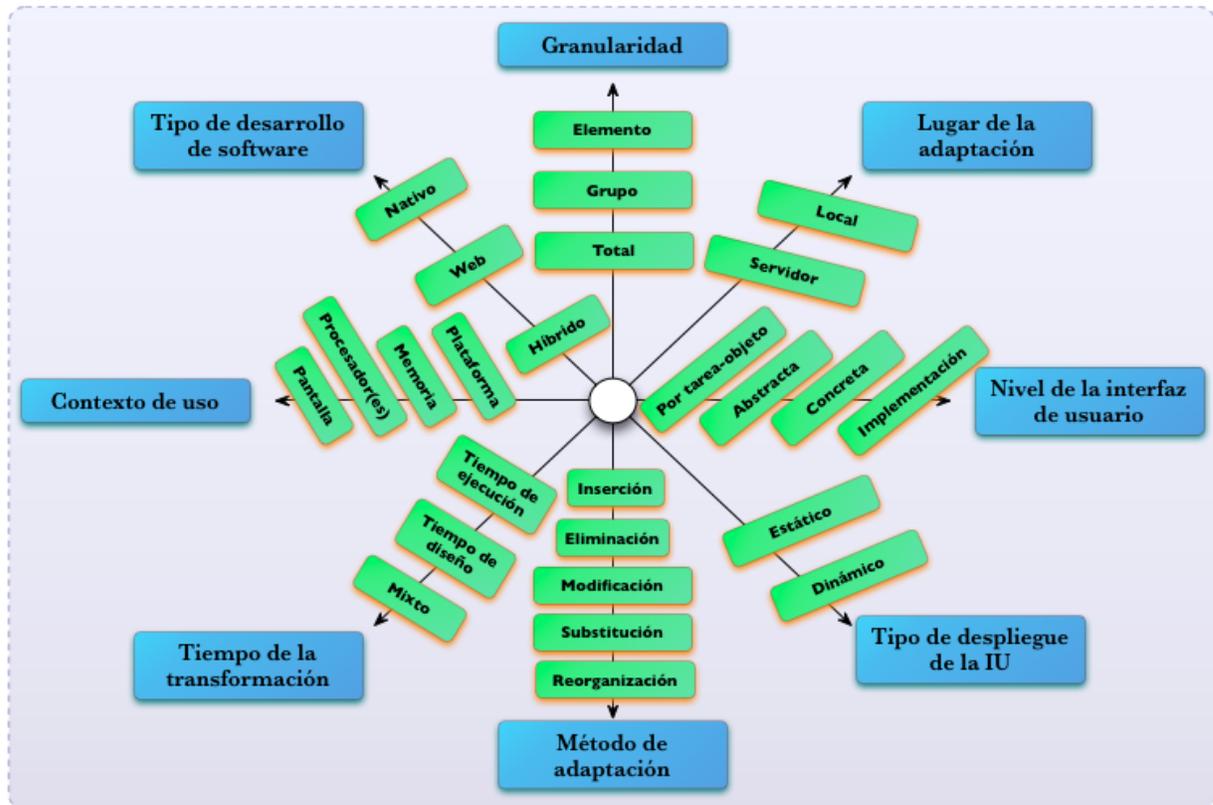


Figura 16: Espacio de diseño para interfaces gráficas de usuario en dispositivos móviles

- **Tipo de desarrollo de software:** establece el tipo de desarrollo que un programador selecciona de acuerdo a la cantidad y a los tipos de dispositivos que abarque su implementación de software.
- **Granularidad:** cantidad de elementos gráficos pertenecientes a la interfaz gráfica de usuario.
- **Lugar de la adaptación:** determina en qué lugar se lleva a cabo el proceso de adaptación de la interfaz gráfica de usuario.
- **Nivel de la interfaz de usuario:** describe el nivel de abstracción de la interfaz gráfica de usuario.
- **Tipo de despliegue de la interfaz de usuario:** determina si la interfaz gráfica de usuario se encuentra definida o se genera en tiempo de ejecución.

- **Método de adaptación:** selecciona una o más técnicas plásticas para llevar a cabo la adaptación de una interfaz gráfica de usuario. Cada una de estas técnicas se describe en la sección 2.2.2.
- **Tiempo de la transformación:** establece el momento en que se lleva a cabo la composición de la interfaz gráfica de usuario.

### *Granularidad*

La granularidad se refiere al tamaño de los elementos que componen a las interfaces de usuario. De hecho, nosotros podemos componer elementos de la interfaz gráfica de usuario en grupos de objetos y podemos unirlos en secciones, a fin de obtener la interfaz gráfica de usuario para una aplicación entera. A continuación, se da a conocer los diversos valores en esta dimensión:

- **Elemento:** unidad mínima de una interfaz gráfica de usuario, un solo gráfico, e.g., un botón, una etiqueta, etc.
- **Grupo:** composición de elementos gráficos para formar un grupo con una determinada relación.
- **Sección:** composición de un conjunto de elementos gráficos que conforman una interfaz gráfica en particular. Una sección es un conjunto de elementos de la interfaz gráfica de usuario que son perceptibles en un momento determinado, e.g., para un sitio Web, la página que el usuario está visitando actualmente es un ejemplo de una sección.

### *Lugar de la adaptación*

Como se ha comentado desde el comienzo de esta tesis, los dispositivos móviles se encuentran limitados en recursos de *hardware*. En consecuencia, es necesario que aplicaciones o procesos que demanden un alto consumo de recursos sean ejecutadas en otro entorno para maximizar el uso del dispositivo. A continuación, se dan a conocer los casos previstos en esta dimensión:

- **Local:** el proceso de adaptación se lleva a cabo automáticamente en el dispositivo móvil, ya que cuenta con los suficientes recursos de *hardware* para que el rendimiento de la aplicación no se vea comprometido.

- **Servidor:** la adaptación de la interfaz gráfica de usuario requiere de una cantidad de procesamiento que sobrepasa los recursos del móvil. Los casos previstos son: a) que la interfaz de usuario se componga de múltiples elementos gráficos y b) que el dispositivo se encuentre muy limitado en recursos, por lo que la aplicación se podría ejecutar lentamente.

#### *Nivel de la interfaz gráfica de usuario*

Una interfaz gráfica de usuario se puede describirse en diversos niveles de abstracción, e.g., abstracta, concreta, implementación, por tareas o por objetos, a partir de los niveles más altos de abstracción hasta los más concretos:

- **Actividad:** este nivel se refiere al modelado de las actividades que llevará a cabo el usuario en la interfaz gráfica de usuario para alcanzar sus objetivos. En este nivel se define la estructura de navegación y la interactividad de dicha interfaz.
- **Abstracta:** en este nivel se describen sintácticamente los componentes de la interfaz gráfica independientemente de una plataforma específica, e.g. se definen los componentes gráficos mediante etiquetas en un archivo de lenguaje XML. Utilizando una interfaz abstracta, definimos la estructura de nuestra aplicación móvil donde establecemos los componentes que van a conformar las distintas interfaces gráficas de usuario. El diseño de interfaz abstracta propuesto sigue algunos principios de diseño orientado a objetos y se centra en definir los componentes gráficos tal y como se deben mostrar en las vistas de navegación y aumentados con más elementos de interacción, tales como botones y enlaces.
- **Concreta:** una interfaz concreta se define como la interfaz gráfica que se visualiza en la pantalla del dispositivo, con el propósito de interactuar con el usuario.
- **Implementación:** descripción de una interfaz gráfica de usuario en un lenguaje o *framework* específico.

#### *Tipo de despliegue de la interfaz gráfica de usuario*

Una interfaz gráfica de usuario se puede mostrar en la pantalla de un dispositivo de dos formas posibles:

- **Estática:** la interfaz gráfica de usuario es definida desde el comienzo del desarrollo de la aplicación y no lleva a cabo ningún cambio en tiempo de ejecución.

- **Dinámica:** la interfaz gráfica de usuario es definida parcialmente o de manera abstracta, para posteriormente desplegar en tiempo de ejecución una interfaz concreta en el dispositivo de acuerdo a sus propiedades.

### *Método de adaptación*

Los métodos de adaptación considerados en este trabajo se encuentran representados en la fig. 17. Como se puede observar, la técnica de adaptación plástica para este trabajo es la **remodelación**, la cual nos permite manipular directamente una interfaz gráfica de usuario mediante el uso de los siguientes métodos: inserción, eliminación, modificación, sustitución y reorganización de los elementos gráficos.

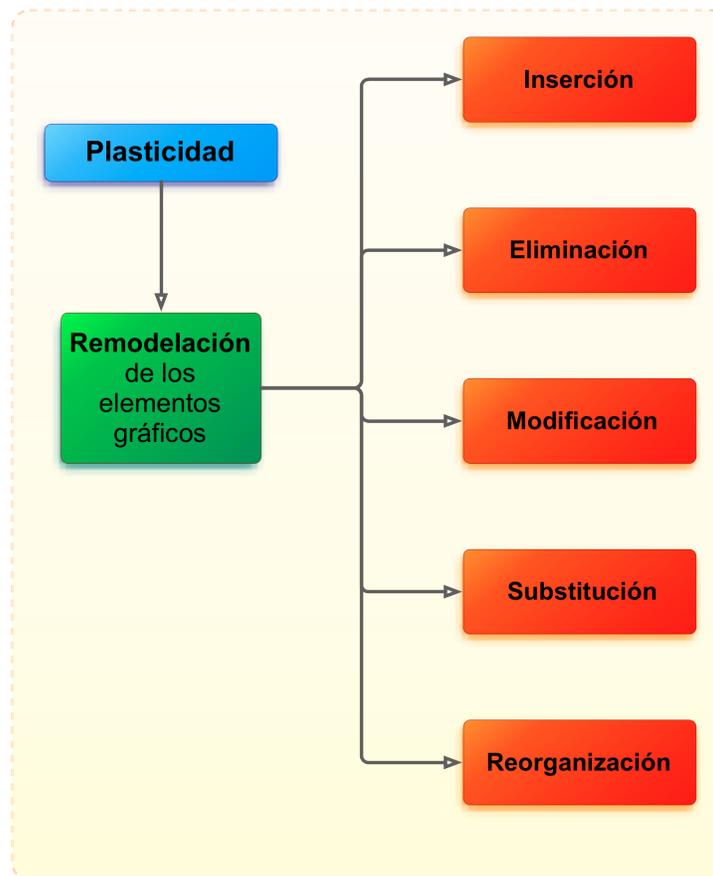


Figura 17: Técnicas plásticas para la adaptación de interfaces gráficas de usuario

### *Tiempo de la transformación*

Esta dimensión especifica en qué momento se produce la composición de la interfaz gráfica de usuario. Puede ser una composición estática que se produce en tiempo de diseño, en cuyo caso los elementos que vamos a componer son conocidos de antemano. También puede ser una composición dinámica, la cual se produce durante la ejecución de la aplicación. La composición dinámica es especialmente importante en aplicaciones ubicuas, ya que en esos entornos el contexto de uso puede variar significativamente y puede requerir diferentes servicios en función de dichas variaciones.

- **Tiempo de ejecución:** la transformación se lleva a cabo dinámicamente durante la ejecución de la aplicación.
- **Tiempo de diseño:** se trata de una composición estática, i.e., se define durante el diseño de la aplicación.
- **Mixto:** la composición se produce parcialmente en tiempo de diseño y parcialmente en tiempo de ejecución.

#### 4.1.3 *Ciclo de adaptación de interfaces gráficas de usuario*

Nuestra meta se centra en el desarrollo de la capa de adaptación, la cual tiene como objetivo presentar una interfaz gráfica al usuario con la cual pueda interactuar de manera amigable y transparente, de acuerdo a las características del dispositivo que el cliente esté usando en ese momento.

Como se puede observar en la fig. 18 se define una serie de pasos consecutivos que permiten adaptar la interfaz gráfica de usuario de acuerdo al perfil del dispositivo. Este ciclo consta de cuatro pasos: petición, detección, adaptación y respuesta.

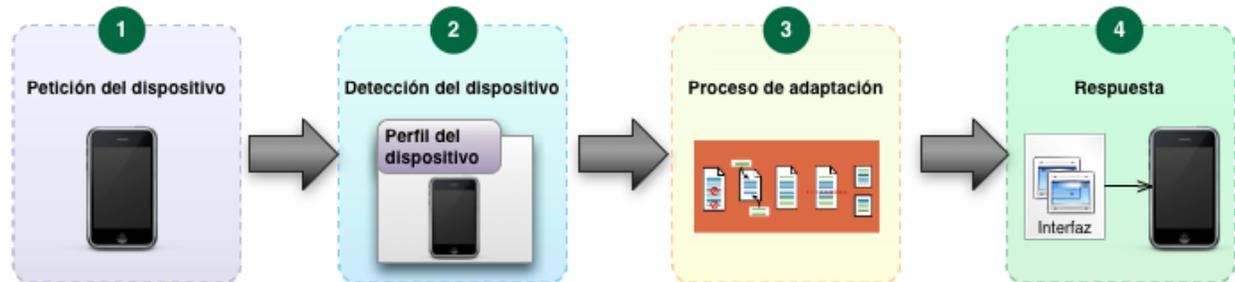


Figura 18: Ciclo de adaptación

A continuación, se da a conocer el contenido de cada etapa en el ciclo de adaptación para interfaces gráficas de usuario:

- **Petición:** el *software* instalado en el dispositivo móvil envía una solicitud para obtener una interfaz gráfica de usuario.
- **Detección:** la aplicación de *software* se encarga de obtener el perfil del dispositivo móvil, i.e., recaba información del *hardware* del propio dispositivo.
- **Adaptación:** consiste en ajustar, modificar o transformar los elementos de la interfaz gráfica de usuario de acuerdo a las características del *hardware* del dispositivo móvil. Algunos casos que se deben tener en cuenta son: ajuste de los componentes gráficos al tamaño de la pantalla, fotos e imágenes de un tamaño adecuado, tipos de medios compatibles y soporte de estándares Web.
- **Respuesta:** finalmente se debe desplegar la interfaz gráfica de usuario previamente ajustada a las características del *hardware* del dispositivo móvil, e.g., al tamaño de la pantalla. Se recomienda llevar a cabo una serie de pruebas para cada dispositivo o clase de dispositivo para verificar que la interfaz gráfica de usuario se despliega correctamente.

4.2 DESCRIPCIÓN DEL MECANISMO DE ADAPTACIÓN

En esta sección se describen las principales propiedades y métodos que utiliza nuestra propuesta de mecanismo plástico para la visualización dinámica de una interfaz gráfica de usuario. Este mecanismo se puede observar en la fig. 19, el modelo va a permitir apreciar sus diferentes componentes y la forma en la que interactúan cada uno de ellos para llevar a cabo el proceso de adaptación dinámica.

En la fig. 19 se puede observar que el modelo consta de dos elementos principales, un cliente y un servidor. Dentro de cada uno de estos elementos se encuentran un conjunto de capas, las cuales tienen como propósito tratar una parte del ciclo de adaptación de la interfaz gráfica de usuario, así como las propiedades del dispositivo móvil para optimizar dicho proceso.

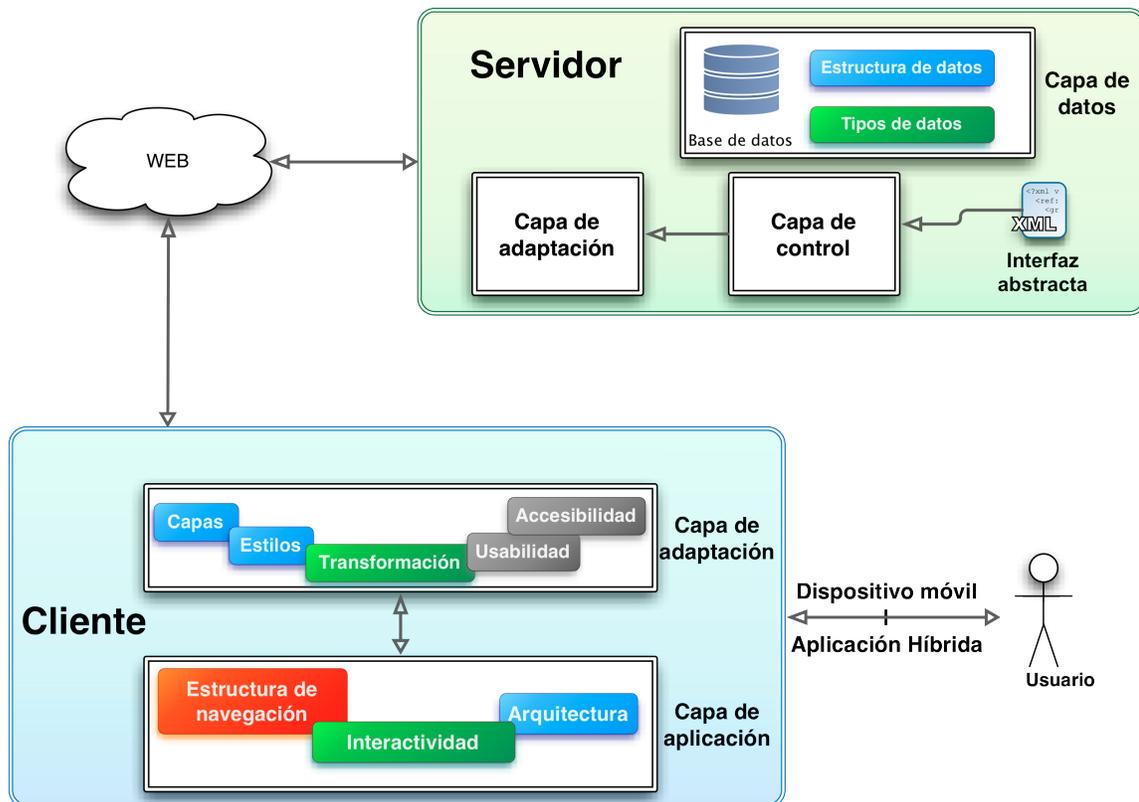


Figura 19: Mecanismo para el despliegue dinámico de una interfaz gráfica de usuario

En la fig. 19 se denotan un total de cinco capas, de las cuales las capas de control y de adaptación (en cliente y servidor), son las encargadas de la adaptación y optimización de la interfaz gráfica de usuario. Mientras que la capa de datos se encuentra en el servidor, en caso de que la aplicación de *software* requiera de una conectividad a la base de datos.

Finalmente, la capa de aplicación se encarga de estructurar la forma en la que las diferentes interfaces gráficas de usuario se comunican entre sí para interactuar con el usuario de la aplicación de *software* desarrollada. Más adelante se describen con mayor detalle cada una de las capas que intervienen en nuestro modelo.

En un proceso de ingeniería, los modelos se utilizan para la especificación, documentación y comunicación. Ellos mismos son objetos de transformación y procesamiento y constituyen una base para la toma de decisiones, análisis, validación, verificación y pruebas.

#### 4.2.1 Arquitectura

Para el presente trabajo se ha decidido implementar una arquitectura cliente-servidor, ilustrada en la figura 20, debido a la necesidad de proveer al dispositivo móvil de un lugar para realizar las transacciones con la base de datos de la aplicación y de proveer el archivo XML, donde se guarda la representación abstracta o definida de las interfaces gráficas para que el usuario interactúe con el sistema.

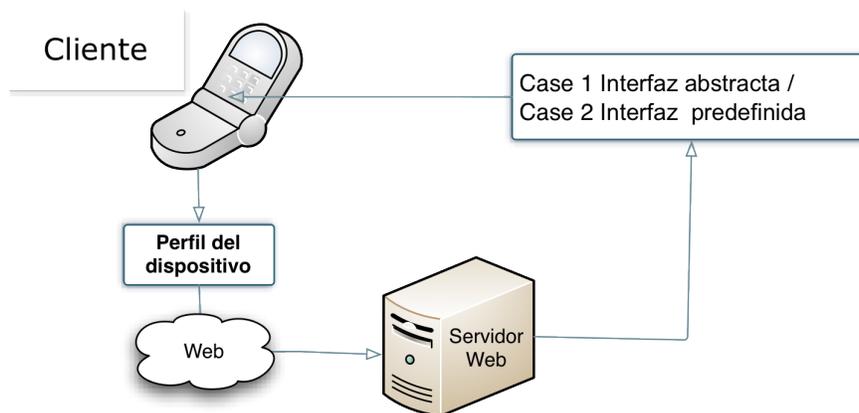


Figura 20: Arquitectura cliente-servidor

#### 4.2.2 *Interfaz abstracta*

Utilizando una interfaz abstracta, definimos la estructura de nuestra aplicación móvil donde establecemos los componentes que van a contener las distintas interfaces. El diseño propuesto de interfaz abstracta sigue algunos principios del diseño orientado a objetos y se centra en definir los componentes gráficos de la interfaz de usuario y la forma en la que van a interactuar con el usuario. Con este modelo recomendamos utilizar un archivo XML para definir los elementos gráficos de cada interfaz de usuario.

Un conjunto de vistas definidas mediante datos abstractos pueden ser usados para especificar el comportamiento de aparición de los objetos. Estas pueden ser vistas como abstracciones fachada, las cuales representan diferentes apariencias de los nodos de navegación para diferentes usuarios en diferentes contextos. Ofrecen también un aspecto de comportamiento interactivo y dinámico.

#### 4.2.3 *XML y aplicaciones móviles*

De acuerdo a B'Far [B'Far, 2005] las aplicaciones móviles se encuentran relacionadas con XML de las siguientes dos maneras:

- Las aplicaciones móviles deben interpretar y ser capaces de manipular el contenido XML. Contenidos de Internet y de otras redes incluyen un formato XML por lo cual, es necesario que las aplicaciones móviles puedan manejarlo. La tarea de analizar e interpretar los datos XML se puede hacer en el propio dispositivo móvil o en algún servidor de aplicaciones que procese todo el contenido para el dispositivo.
- Las aplicaciones móviles usan XML para facilitar sus implementaciones, i.e., los documentos XML pueden ser usados por las aplicaciones móviles para el intercambio de datos, ya que es posible encapsular la configuración de un dispositivo o de un servidor en un documento XML.

En caso de que la aplicación móvil maneje contenido XML o lo use internamente, esta debe ser capaz de tomar acciones respecto a dicho contenido.

## 4.2.4 Capas presentes en el mecanismo

De acuerdo a nuestro mecanismo, ilustrado en la figura 19, se establecen cinco capas, las cuales se encargan de llevar a cabo un conjunto de procedimientos para el correcto despliegue dinámico de la interfaz gráfica de usuario en dispositivos móviles. A continuación se da a conocer en qué consiste cada capa así como su rol en el mecanismo para llevar a cabo el despliegue de la interfaz gráfica de usuario en multiplataformas.

- **Capa de aplicación:** en esta capa se especifican los elementos que permiten definir el comportamiento y la interacción de la aplicación con el usuario, i.e., la estructura de navegación, interactividad, arquitectura y protocolos.

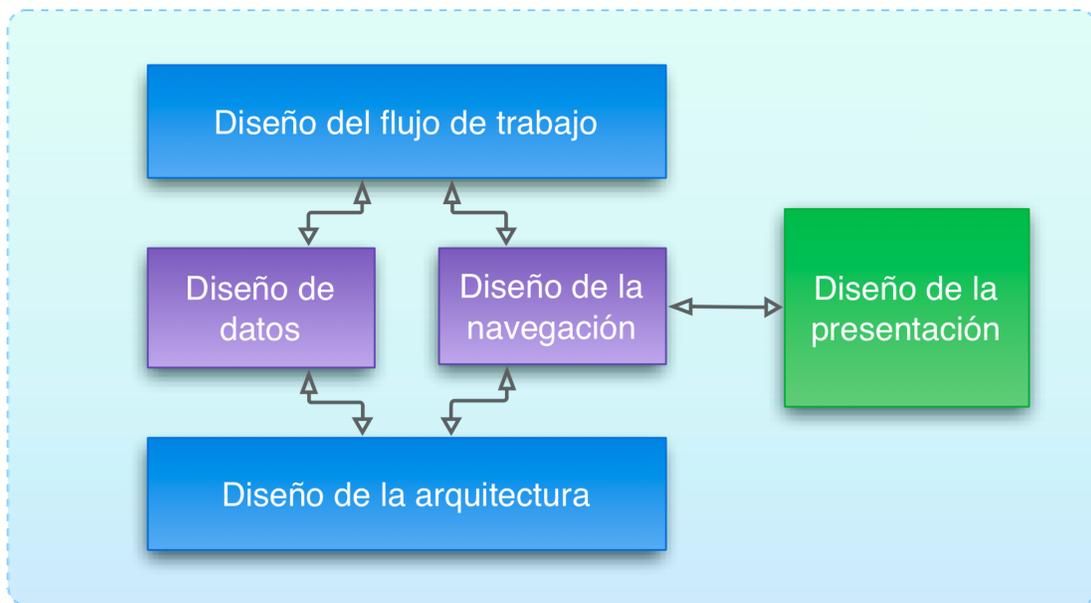


Figura 21: Estructura de navegación

Como se puede observar en la fig. 21, el flujo de trabajo de una aplicación de *software* se compone de cuatro elementos: diseño de datos, diseño de navegación, diseño de la arquitectura y el diseño de la presentación.

En primer instancia es necesario definir la arquitectura de nuestra aplicación, en la cual establecemos los datos que va a manejar durante su ejecución; posteriormente se requiere definir la forma en la que se van a ir presentando las diferentes interfaces gráficas al usuario, lo cual nos lleva al diseño de navegación.

Finalmente, es necesario establecer la forma y el momento en el que se va a ir desplegando en la pantalla cada una de las interfaces gráficas de usuario generadas por las demás capas del proceso de adaptación.

- Capa de control:** esta capa de encarga de determinar el grano de los elementos gráficos definidos en la interfaz abstracta. Si el grano es fino, la adaptación se llevará a cabo en el cliente mediante su capa de adaptación. Sin embargo, si el grano es grueso, la adaptación se realizará en el servidor, debido a las restricciones de los recursos del dispositivo móvil.

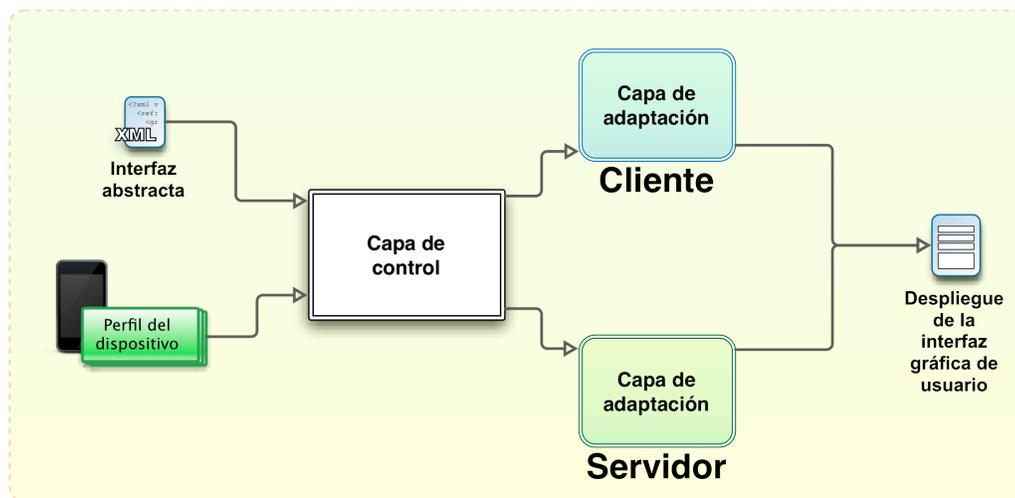


Figura 22: Capa de control

En la fig. 22 se puede observar que la capa de control tiene dos elementos de entrada: la interfaz abstracta y el perfil del dispositivo. Estos dos elementos sirven propiamente para establecer una evaluación del grado de dificultad para generar la interfaz gráfica de usuario en el dispositivo móvil. Finalmente, se lleva a cabo una selección entre la capa de adaptación que se encuentra en el cliente o la capa de adaptación en el servidor para realizar el siguiente paso en el ciclo de adaptación.

- Capa de adaptación en el cliente:** esta capa se encarga de llevar a cabo la adaptación en tiempo de ejecución directamente en el dispositivo móvil, siempre y cuando el grano de los elementos sea fino.

Para ello, esta capa se compone de las opciones para cada uno de los posibles casos definidos en la figura 23, i.e., cambio de estilos, transformación, cambio de capas, etc., siempre preservando la usabilidad y la accesibilidad de la aplicación al usuario. Si el grano es grueso, esta capa simplemente se encargará de desplegar la interfaz predefinida por el servidor.

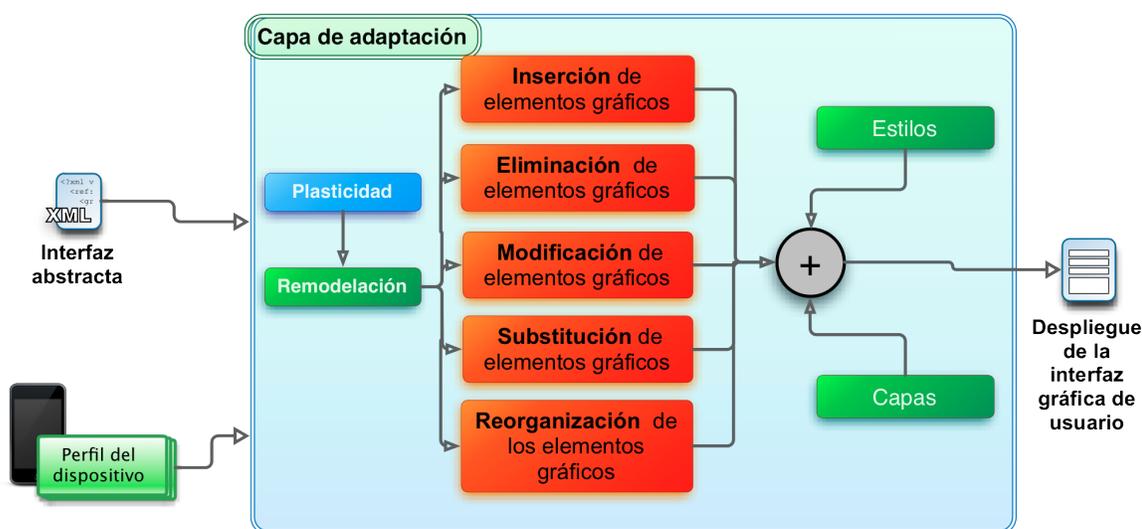


Figura 23: Capa de adaptación

Como se puede observar en la fig. 23 los elementos de entrada son: la interfaz abstracta y el perfil del dispositivo. Internamente la capa de adaptación se encarga de seleccionar la técnica o conjunto de técnicas que proporciona el método de adaptación por remodelación para generar, adecuar y optimizar una interfaz gráfica de usuario para el dispositivo móvil en el cual se ejecute la aplicación.

Finalmente, se adiciona a la interfaz gráfica de usuario los estilos y las capas de acuerdo al equipo en el cual se va a desplegar.

- **Capa de adaptación en el servidor:** su objetivo es llevar a cabo la transformación de la interfaz abstracta en una interfaz concreta cuando el grano de los elementos gráficos sea grueso, i.e., una vez que se ha evaluado y dictaminado que el perfil del dispositivo no cumple con los requerimientos mínimos para adaptar la interfaz gráfica de usuario sin comprometer su rendimiento, el servidor adquiere la responsabilidad de llevar a cabo el proceso de adecuación de la interfaz, mediante la activación de su capa de adaptación y posterior envío de la interfaz concreta al dispositivo móvil.
- **Capa de datos:** es la capa que se encarga del intercambio de información con el usuario, i.e., almacenamiento, actualización y modificación de datos, en una base de datos localizada en el servidor.

Un gran número de aplicaciones requieren comunicarse con un servicio de base de datos, con el fin de llevar a cabo una determinada actividad. Por lo cual esta capa permite adicionar la capacidad de intercambio de información entre el dispositivo móvil y un servidor de la base de datos.

---

## IMPLEMENTACIÓN DEL MECANISMO PROPUESTO

---

En este capítulo se da a conocer el software desarrollado que permite ejemplificar el modelo propuesto en sus diferentes casos de uso. En la sección 5.1 se describe las herramientas y la estructura del software en la cual se implementaron los casos de uso. Posteriormente en la sección 5.2 se da a conocer el esquema de interacción entre las tecnologías implementadas. Después, en la sección 5.3 se define el contexto de uso para los casos prácticos. En la sección 5.4 se muestra el ciclo de adaptación de las interfaces gráficas complementado con el modelo de adaptación. Finalmente en la sección 5.5 se da a conocer las implementaciones y sus principales características.

### 5.1 NIVELES TECNOLÓGICOS ABARCADOS EN CASOS DE IMPLEMENTACIÓN

En la fig. 24, se puede observar las áreas tecnológicas que se han cubierto en la implementación del modelo propuesto.

A modo de jerarquía, los casos de implementación han cubiertos los siguientes niveles:

- **Dispositivos móviles:** en los casos de implementación del modelo ya no se presenta la limitante de la proliferación de dispositivos móviles, ya que al utilizar implicar la contextualización del dispositivo y un desarrollo multi-plataforma. Se ha podido extender un desarrollo software a varias plataformas a la vez. El hardware usado en los casos de uso son:
  - **Apple iPad**
  - **Apple iPhone 3g**
  - **Motorola MB300**

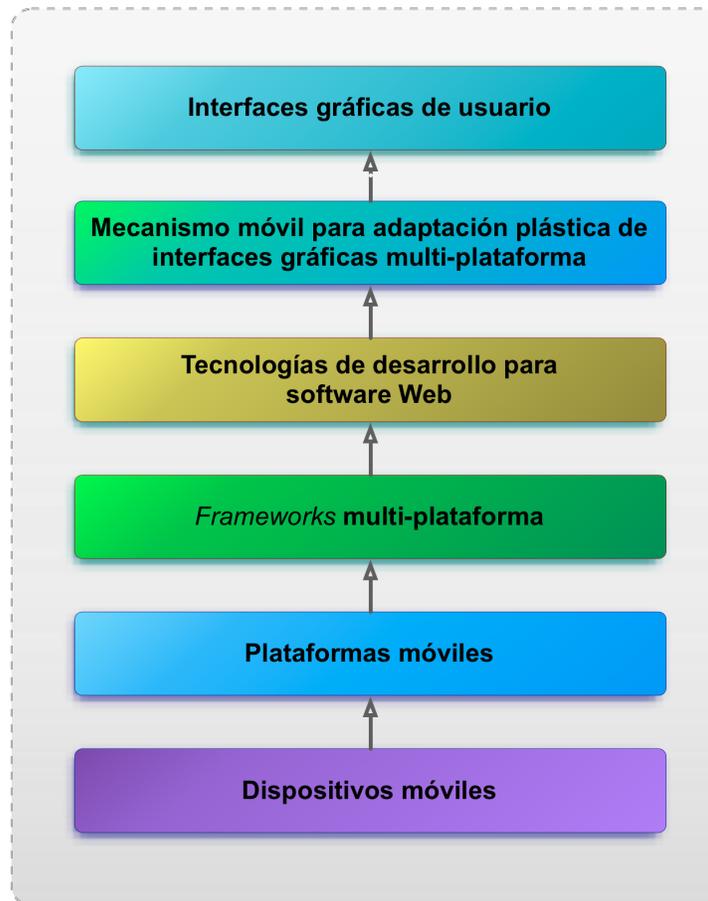


Figura 24: Niveles tecnológicos abarcados

- **Plataformas móviles:** En los casos de estudio, la plataforma móvil se toma como contexto de uso, lo que permite abarcar más de una sola plataforma para un desarrollo de software a la vez. En los casos de estudio implementados se ha trabajado sobre dos plataformas móviles:
  - Apple iOS
  - Google Android

## 5.1 NIVELES TECNOLÓGICOS ABARCADOS EN CASOS DE IMPLEMENTACIÓN

- **Framework(s) multiplataforma:** para el desarrollo de las aplicaciones de software mostradas en los casos de uso se han utilizado dos *frameworks* móviles para la programación en múltiples plataformas:
  - PhoneGap
  - Rhodes
  
- **Tecnologías de desarrollo de software para software Web:** al incorporar tecnologías para desarrollo Web en nuestro enfoque para adaptación de interfaces gráficas, es posible disminuir el tiempo de desarrollo y abarcar una gran cantidad de herramientas gráficas para mejorar la presentación de la interfaz gráfica de usuario con la cual el usuario interactúa al final del proceso de adaptación. A continuación, se enlistan algunas de las tecnologías que se han usado en la presente implementación:
  - HTML5
  - CSS3
  - jQueryMobile
  - Librerías gráficas
  
- **Mecanismo propuesto:** este nivel corresponde a la implementación del *mecanismo móvil para la adaptación plástica de interfaces gráficas multi-plataforma*, la cual permite adaptar las interfaces a los diferentes contextos de uso presentes en las capas anteriores y generar una interfaz gráfica adecuada a las características del dispositivo en cuestión.
  
- **Interfaces gráficas de usuario:** este nivel corresponde a la fase final de la implementación del software, en la cual se despliega la interfaz gráfica de usuario en el dispositivo móvil al finalizar el proceso de adaptación.

5.2 ESQUEMA DE INTERACCIÓN ENTRE TECNOLOGÍAS DE DESARROLLO DE SOFTWARE PARA EL DESPLIEGUE DE UNA INTERFAZ GRÁFICA DE USUARIO

En la fig. 25 se puede observar la manera en la cual interactúan los diversas tecnologías que se han requerido para llevar a cabo la implementación del mecanismo móvil en los casos de uso que se presentan en este capítulo.

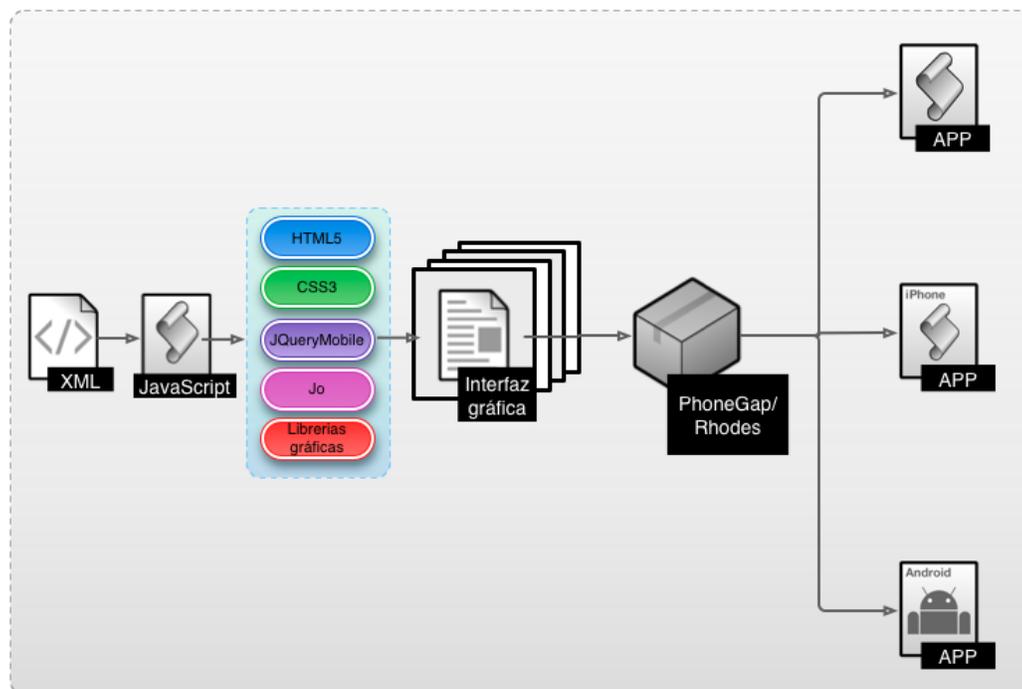


Figura 25: Esquema de interacción entre tecnologías Web y multi-plataforma

- **XML:** esta herramienta nos permite definir de manera abstracta los elementos que contiene nuestra interfaz de usuario. En un archivo se colocan los elementos que conforman cada sección de la interfaz, los cuales se interpretarán para generar la interfaz concreta de manera gráfica.
- **JavaScript:** permite llevar el control de cada una de las etapas de implementación del modelo, ya que en este lenguaje está desarrollada cada una de las funciones para llevar el proceso de adaptación de la interfaz gráfica.

- **Frameworks para desarrollo Web:** permite desplegar de manera gráfica los elementos que conforman la interfaz de usuario, de acuerdo a dicho proceso de adaptación.
- **Frameworks para desarrollo de software multi-plataforma:** permite ser el medio para llegar a cada una de las plataformas móviles en el mercado. En la aplicación que se genere de este *framework* estarán empaquetados los procedimientos para la comunicación con el servidor y el de la adaptación de la interfaz de usuario.

### 5.3 CONTEXTO DE USO EN CASOS PRÁCTICOS

Los elementos que se han seleccionado para ser el contexto de uso para la adaptación de interfaces gráficas en dispositivos móviles son:

- **Plataforma:** al obtener los datos de la plataforma nos permite dar a la interfaz gráfica ciertas propiedades correspondientes al tipo de interfaz nativa presente en la plataforma en uso.
- **Pantalla:** la pantalla nos permite obtener las capacidades gráficas del dispositivo, las cuales abarcan: el tamaño de pantalla, resolución y profundidad de pixel. Estos son fundamentales para establecer el máximo número de elementos gráficos presentes por cada sección de la interfaz gráfica y determinar el tipo de transformación más adecuada.
- **Procesador(es):** esta variable contextual permite establecer si el dispositivo cuenta con el suficiente poder de procesamiento para llevar a cabo la adaptación en el dispositivo sin que se vea afectado el rendimiento de la aplicación de software o sea necesario trasladar este proceso a una entidad externa.

En la fig. 26 se da a conocer los elementos que conforman el perfil del dispositivo, el cual sirve como variable contextual para llevar a cabo la correcta adaptación de una interfaz gráfica en el dispositivo móvil.

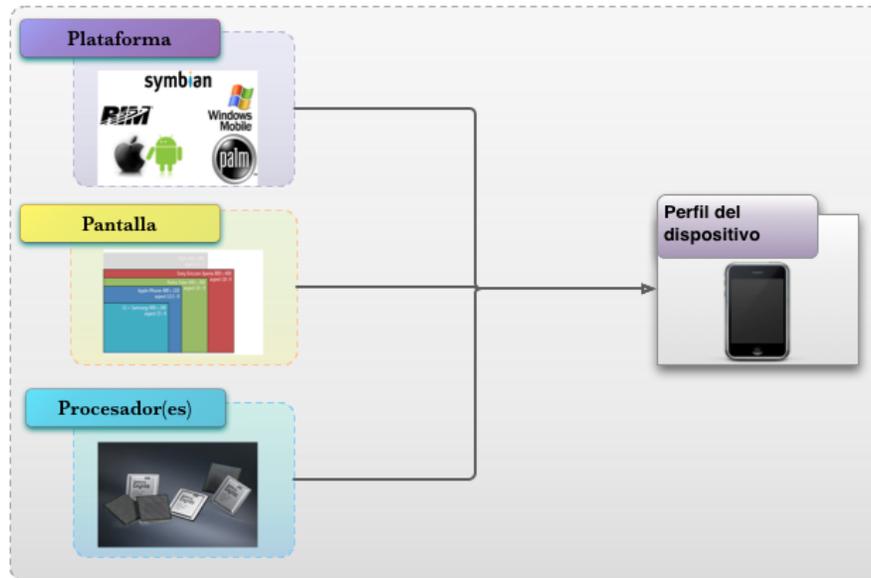


Figura 26: Contexto de uso

#### 5.4 CICLO DE ADAPTACIÓN DE LAS INTERFACES GRÁFICAS

En la fig. 27 se puede observar las etapas por las cuales debe pasar la interfaz de usuario antes de ser desplegada en la pantalla del dispositivo móvil. A continuación se detalla cada una de estas etapas:

- **Definición de la interfaz abstracta:** en esta fase, la interfaz es definida de manera abstracta en un archivo XML, en el cual se detallan los elementos gráficos que conformarán la interfaz final.
- **Análisis:** en esta etapa se obtiene los datos de la interfaz abstracta y el perfil (*profile*) del dispositivo móvil, con el fin de determinar dónde y cómo se llevará a cabo el proceso de adaptación.
- **Adaptación:** esta fase se encarga de incorporar los elementos gráficos a la interfaz de acuerdo a los datos recabados en el análisis y de agregar los parametros de navegación a la interfaz concreta.

#### 5.4 CICLO DE ADAPTACIÓN DE LAS INTERFACES GRÁFICAS

- **Creación de la interfaz concreta:** es la interfaz gráfica que se ha adecuado a los elementos contextuales y está lista para ser desplegada de manera dinámica en la pantalla del dispositivo móvil.
- **Despliegue:** en esta fase la interfaz gráfica se despliega en el dispositivo móvil y el usuario inicia su interacción con la misma.

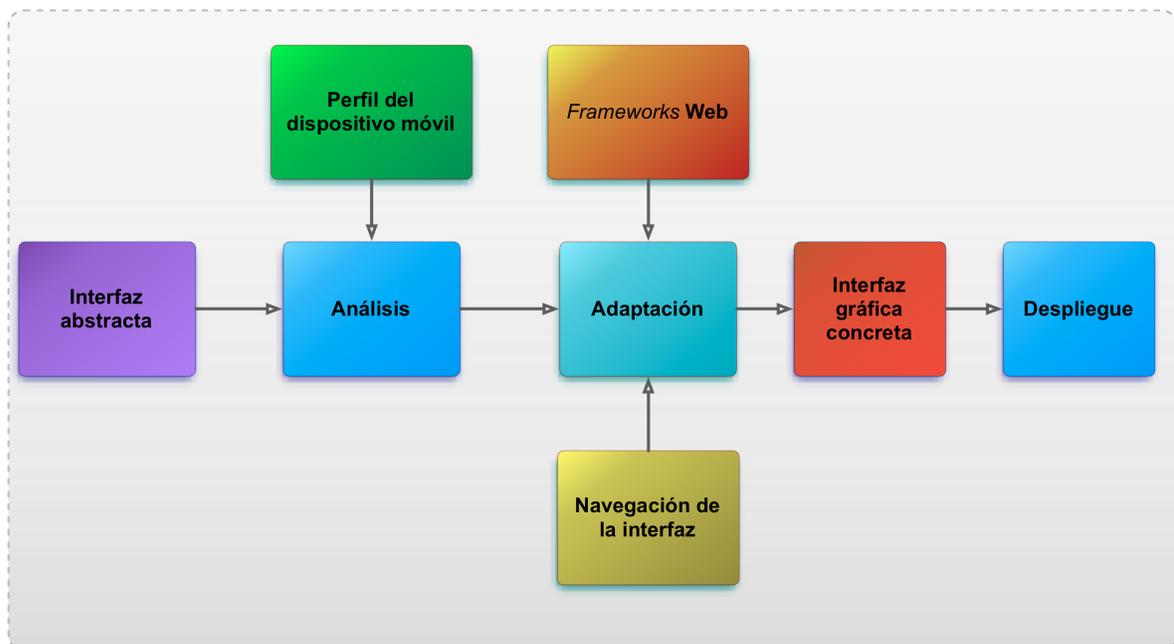


Figura 27: Ciclo de adaptación de interfaces gráficas móviles

### 5.5 CASOS DE USO IMPLEMENTADOS

En esta sección se dan a conocer los casos de uso en los cuales se implementó el mecanismo plástico propuesto.

#### 5.5.1 *Aplicación Mobile Clock*

La aplicación **Mobile Clock** consiste en el despliegue de varios elementos gráficos relacionados con la hora y el tiempo actual. Este caso de uso demuestra las tres primeras transformaciones plásticas aplicadas a un cierto contexto.

Los elementos que conforman la interfaz abstracta son:

- **Reloj analógico**
- **Reloj digital**
- **Fecha actual**

Este caso de uso pretende demostrar los siguientes casos de transformación plástica:

- **Inserción:** consiste en determinar cuál o cuáles son los objetos más importantes de acuerdo al contexto presente en el dispositivo.
- **Eliminación:** pretender retirar los elementos innecesarios de acuerdo al contexto en el que se encuentre el dispositivo.
- **Modificación:** consiste en modificaciones de las propiedades de un elemento gráfico con el fin de ser adaptado al contexto que se presente. Para este caso, el reloj analógico toma las medidas de la pantalla para llevar a cabo dicha adaptación.

En la fig. 28 se puede observar el despliegue de la interfaz gráfica de usuario de la aplicación de *software* **Mobile Clock**, para los casos de: a) iPad, b) iPhone y c) Android.

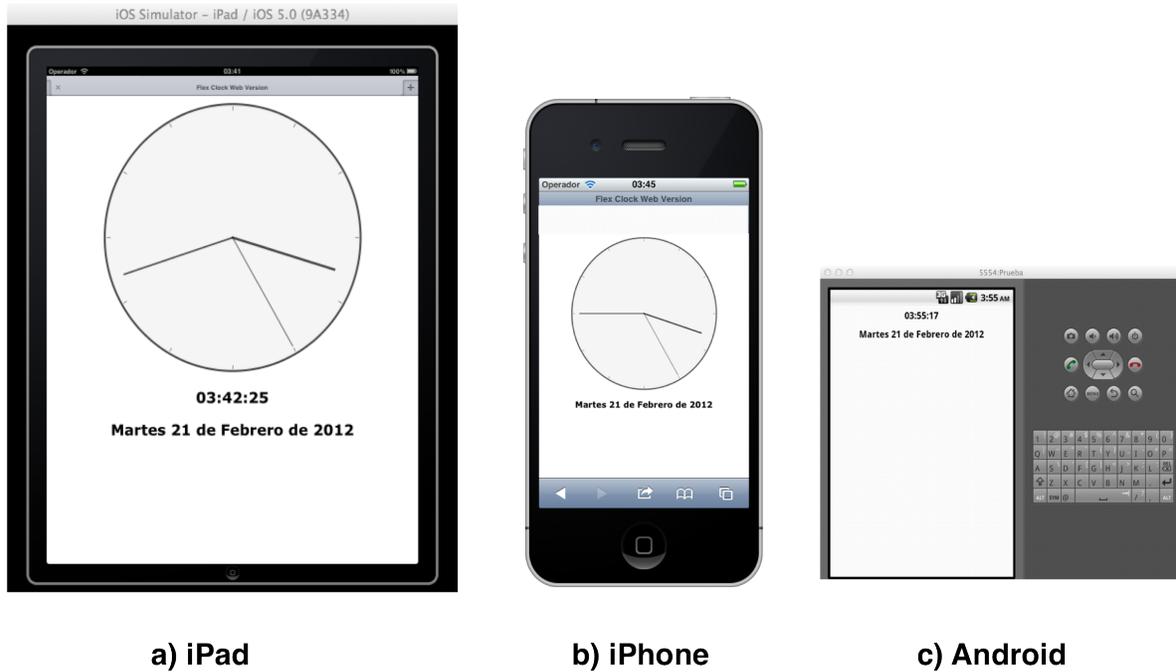


Figura 28: Aplicación Mobile Clock

En el caso a) iPad se pueden observar todos los elementos gráficos que puede contener la aplicación *software Mobile Clock*. Para el caso b) iPhone, el dispositivo cuenta con un tamaño inferior de pantalla, por lo cual es retirado el elemento de la hora digital. Finalmente para el caso c) Android no es posible desplegar el reloj analógico, por lo cual solo se muestran el reloj digital y la fecha del dispositivo.

Las tecnologías implicadas en este caso de uso se encuentran ilustradas en la fig. 29, que a continuación se muestra:



Figura 29: Tecnologías Web para el caso de uso *Mobile Clock*

### *Reloj analógico*

En esta subsección se a conocer cómo un elemento gráfico puede varias sus propiedades, con el objetivo de adaptarse al área que tiene disponible en un objeto canvas. En la fig. 30, se muestra las tecnologías presentes en esta sección de la adaptación.



Figura 30: Tecnologías Web para el caso de uso **Reloj analógico**

En la fig. 31 se da conocer la forma en la cual se toman las medidas y la localización del canvas para llevar a cabo la graficación del reloj analógico.

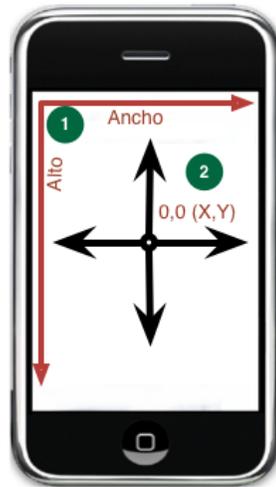


Figura 31: Canvas en un dispositivo móvil

Los resultados de la transformación por modificación se pueden observar en la fig. 32.

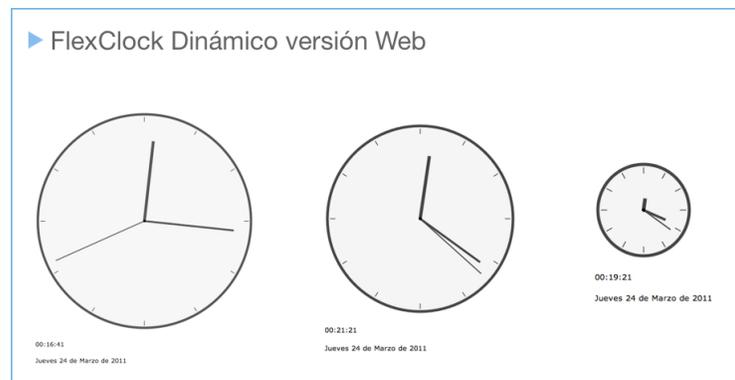


Figura 32: Vistas de la aplicación **Mobile Clock**

### 5.5.2 Aplicación *Airport Reservation*

La aplicación *Airport Reservation* consiste en un sistema que permite llevar a cabo reservas de vuelo de una compañía aérea. Esta aplicación ofrece las siguientes opciones:

- **Fecha de despegue**
- **Aeropuerto origen**
- **Fecha de retorno**
- **Aeropuerto destino**

Este caso de uso pretende demostrar los siguientes casos de transformación plástica:

- **Substitución:** consiste en determinar cuál es la forma gráfica más adecuada para mostrar en la pantalla del dispositivo móvil de acuerdo al contexto de uso presente.
- **Reorganización:** consiste en reordenar los elementos que se encuentran en la pantalla gráfica, con el fin de adecuar los objetos al contexto de uso y brindar al usuario una correcta usabilidad de la aplicación.

Las tecnologías implicadas en este caso de uso se encuentran ilustradas en la fig. 33, que a continuación se muestra:



Figura 33: Tecnologías Web para el caso de uso **Airport Reservation**

En la fig. 34 se muestra el despliegue de la interfaz gráfica de usuario para la reserva-ción de un vuelo desde un navegador de escritorio, donde todos los elementos gráficos se muestran desplegados en su totalidad.

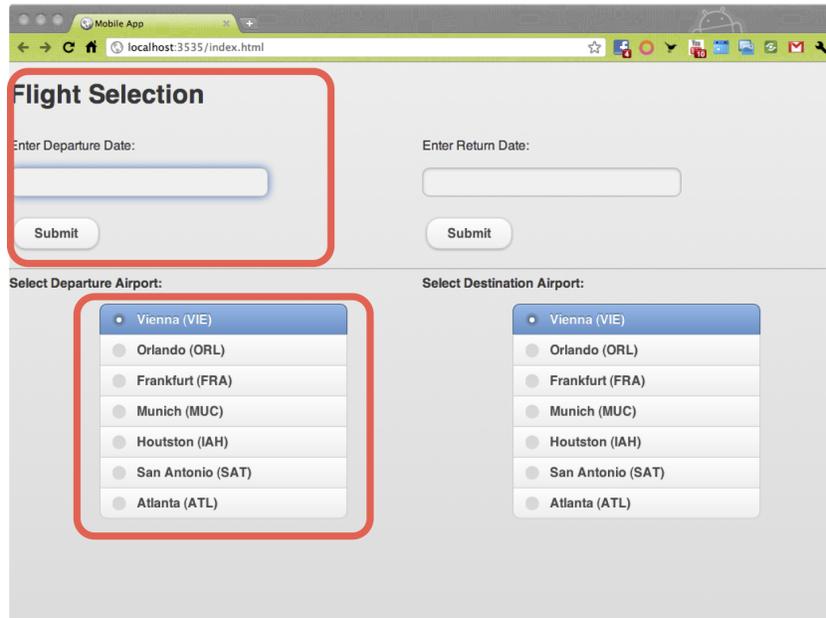


Figura 34: Vista de la aplicación **Airport Reservation** desde un navegador Web en PC de escritorio

En la fig. 35, se ilustra el despliegue de la interfaz de usuario para la reservación de un vuelo desde un dispositivo iPad. A diferencia del caso anterior, las listas desplegadas se han sustituido por listas desplegables y el orden de los elementos se ha modificado.

En la fig. 36, el despliegue de la interfaz de usuario para la reservación de un vuelo desde un dispositivo iPhone. Para este caso, el tamaño de la pantalla se ve considerablemente reducido por lo que es necesario reorganizar la interfaz gráfica en dos pestañas que permitan mostrar de manera adecuada el contenido de dicha interfaz. En segundo punto, las lista desplegables se adecúan al estilo del dispositivo.

## IMPLEMENTACIÓN DEL MECANISMO PROPUESTO

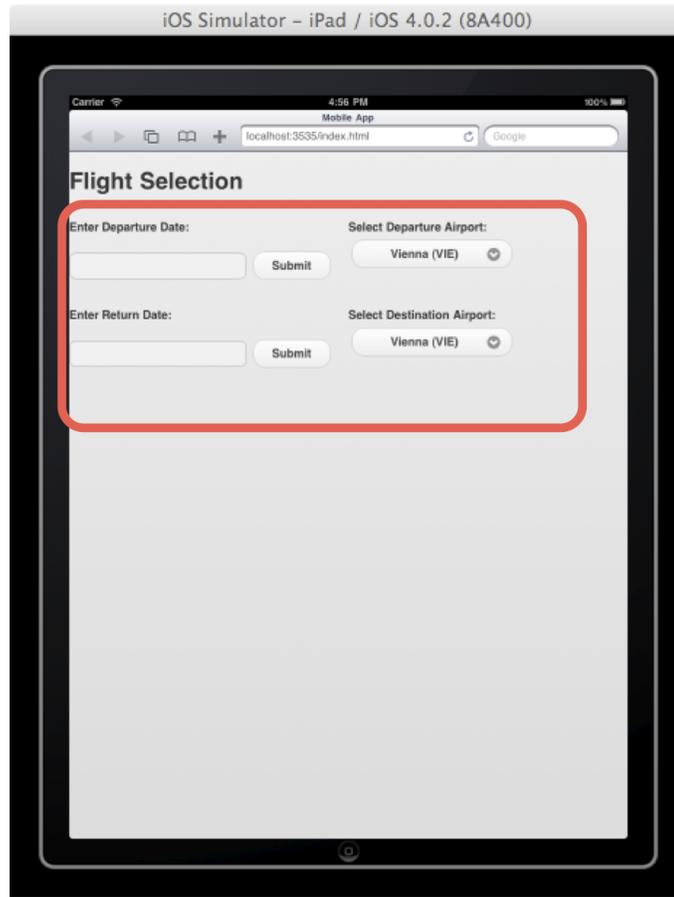


Figura 35: Vista de la aplicación **Airport Reservation** desde un iPad

En la fig. 37, se ilustra el despliegue de la interfaz de usuario para la reservación de un vuelo desde un dispositivo Android con tamaño de resolución *HVGA*. Para este caso, el tamaño de la pantalla se ve muy reducido por lo que es necesario reorganizar la interfaz gráfica en cuatro pestañas que permitan mostrar de manera adecuada el contenido de esta interfaz. Se muestra una sola opción por cada una de las pestañas en la interfaz gráfica.

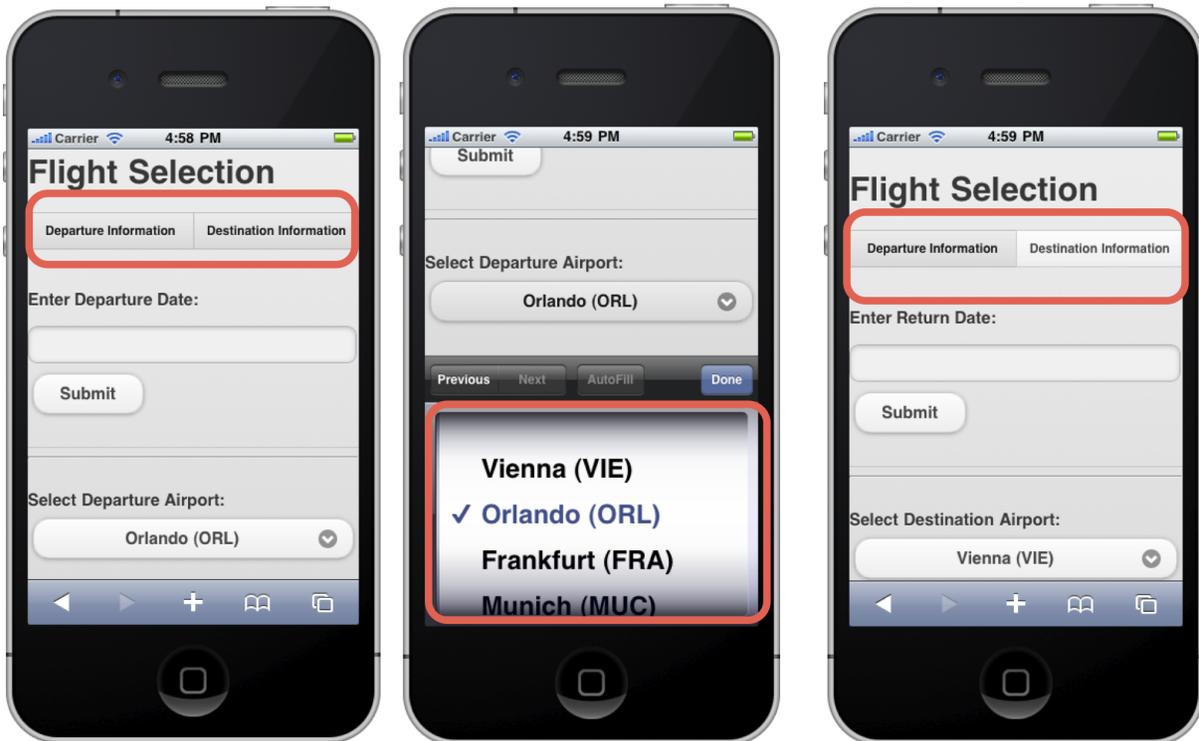


Figura 36: Vista de la aplicación **Airport Reservation** desde un iPhone

## IMPLEMENTACIÓN DEL MECANISMO PROPUESTO

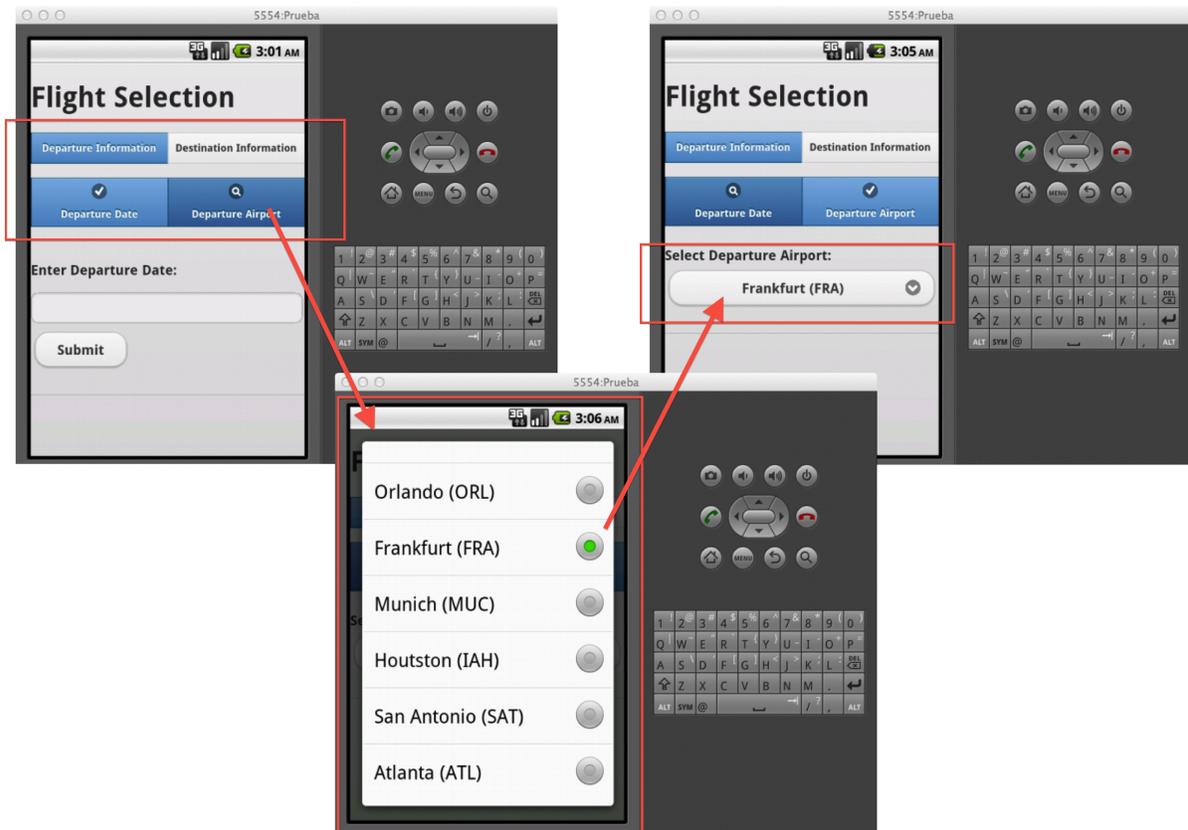


Figura 37: Vista de la aplicación **Airport Reservation** desde un Android

### 5.5.3 Aplicación Demo Banco

La aplicación **Demo Banco** es un sistema de software que está codificado en *jQuery-Mobile* y compilado con *PhoneGap*. Esta aplicación tiene como objetivos: a) proveer a sus usuarios una forma fácil y eficiente de llevar a cabo operaciones bancarias en dispositivos móviles y b) ser un caso de uso en la implementación del mecanismo de adaptación plástica.

Este software fue implementado en los dispositivos móviles de Apple iPhone 3G y Ipad (1ra generación).

La adaptación de las interfaces gráficas de usuario de la aplicación de software **Demo Banco** es llevada a cabo directamente en los dispositivos móviles, i.e, de acuerdo al mecanismo de adaptación plástica, los dispositivos en los que se implementó este desarrollo software cuentan con el *hardware* necesario para ejecutar el proceso de adaptación plástica, sin afectar el rendimiento del equipo y la interactividad de la aplicación con el usuario.

El método de adaptación plástica implementado es el de remodelación de la interfaz gráfica de usuario. La técnica de remodelación que se observa en el despliegue de la interfaz gráfica de usuario en la aplicación *software Demo Banco* es:

- **Reorganización:** la reorganización consiste en cambiar la forma en la cual se muestran los componentes de la interfaz de usuario, mediante una re-evaluación de su disposición espacial y/o su dependencia temporal, i.e., la reorganización puede consistir desde reubicar, cambiar o modificar un solo componente gráfico hasta el cambio de la toda la interfaz gráfica de usuario.

La reorganización puede afectar a diferentes partes de interfaz gráfica de usuario en función de su tipo. Por ejemplo, la reorganización de la interfaz de usuario puede afectar el estilo, el diseño, los contenidos, la estructura, la navegación, así como las capacidades de interacción del usuario. Todo esto en una posible forma independiente.

Las interfaces gráficas de usuario por las que está conformada la aplicación **Demo Banco** son:

## IMPLEMENTACIÓN DEL MECANISMO PROPUESTO

- **Pantalla de bienvenida:** consiste en la primera interfaz gráfica con la que interactúa el usuario, en ella se muestra el mensaje de bienvenida y las opciones de inicio de sesión y alta de nuevo usuario.
  - **Inicio de sesión:** esta interfaz gráfica permite al usuario entrar al sistema después de haber acreditado sus respectivas credenciales.
  - **Alta:** interfaz gráfica que permite registrar a un nuevo usuario al sistema de banca móvil.
  
- **Cuenta bancaria:** despliega al usuario del sistema de banca móvil los datos de su cuenta: 1) cuenta de debito y 2) de credito si tuviese.
  
- **Consulta:** consiste en el despliegue de las últimas operaciones que ha llevado a cabo el usuario.
  
- **Operaciones:** esta interfaz gráfica permite al usuario seleccionar cuál es el tipo de operación bancaria que desea llevar a cabo.
  - **Transferencia:** permite enviar una cantidad monetaria de la cuenta del usuario a la cuenta de otra persona.
  - **Pago a tarjeta de credito:** permite llevar a cabo el pago de las tarjetas de crédito del usuario.
  
- **Contacto:** permite al usuario contactar con su entidad bancaria a través de sus diferentes medios.

A continuación, se dan a conocer las interfaces gráficas de usuario de la aplicación **Demo Banco**, así como las versiones que surgieron entre la versión para iPad y iPhone respectivamente:

En fig. 38 se puede observar la pantalla de bienvenida que se presenta para los dos casos iPad y iPhone.

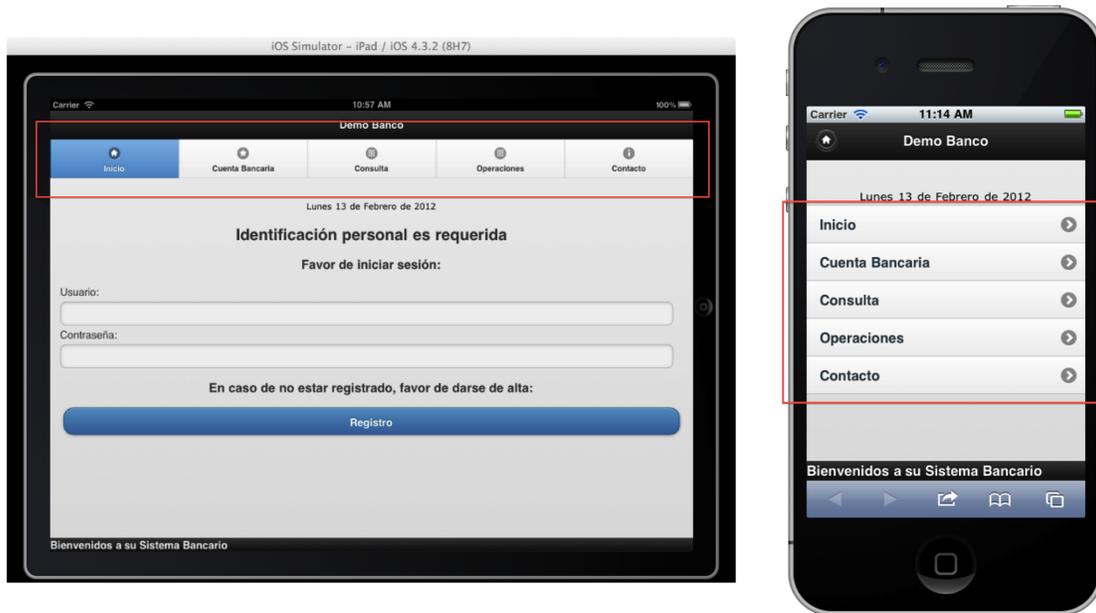


Figura 38: Pantalla de bienvenida en la aplicación **Demo Banco** en iPad y iPhone

En primer lugar podemos observar el primer caso de la técnica de remodelación:

- **Diseño del menu:** a simple vista podemos observar en la fig. 38 que el menu que nos permite acceder a los diferentes opciones de la aplicación se presenta de diferente manera, i.e., mientras que en el iPad el menu se encuentra siempre visible en la parte superior, en el iPhone solo se puede acceder desde la pantalla inicial.
- **Estructura de navegación:** en el caso de iPad el inicio de sesión se muestra directamente en la pantalla inicial, mientras que en el caso del iPhone, el usuario se ve obligado a interactuar de manera distinta. Como se puede observar en la fig. 39, la interfaz de usuario tiene que ser dividida en varias secciones debido al tamaño limitado de la pantalla.

## IMPLEMENTACIÓN DEL MECANISMO PROPUESTO



Figura 39: Vistas de la aplicación **Demo Banco** desde un iPhone: a) principal, b) inicio de sesión y c) alta

En la fig. 40 se puede observar la interfaz gráfica de usuario correspondiente a los datos de la cuenta bancaria del usuario. En esta interfaz se despliega los datos del usuario, su cuenta personal y el estado de las tarjetas de crédito que tenga disponibles.

En ambos casos, el número de interfaces que se pueden llegar a generar va acorde a la cantidad de información que el usuario tenga disponible.

En el caso particular de la fig. 40, se está considerando que el usuario solo tiene una cuenta de débito principal y hasta dos tarjetas de crédito.

La interfaz que sigue en la secuencia es la correspondiente a la **consulta**, la cual se muestra en las fig. 41 y 42. Esta interfaz nos muestra los registros de operaciones que el usuario ha llevado a cabo en el transcurso del mes.

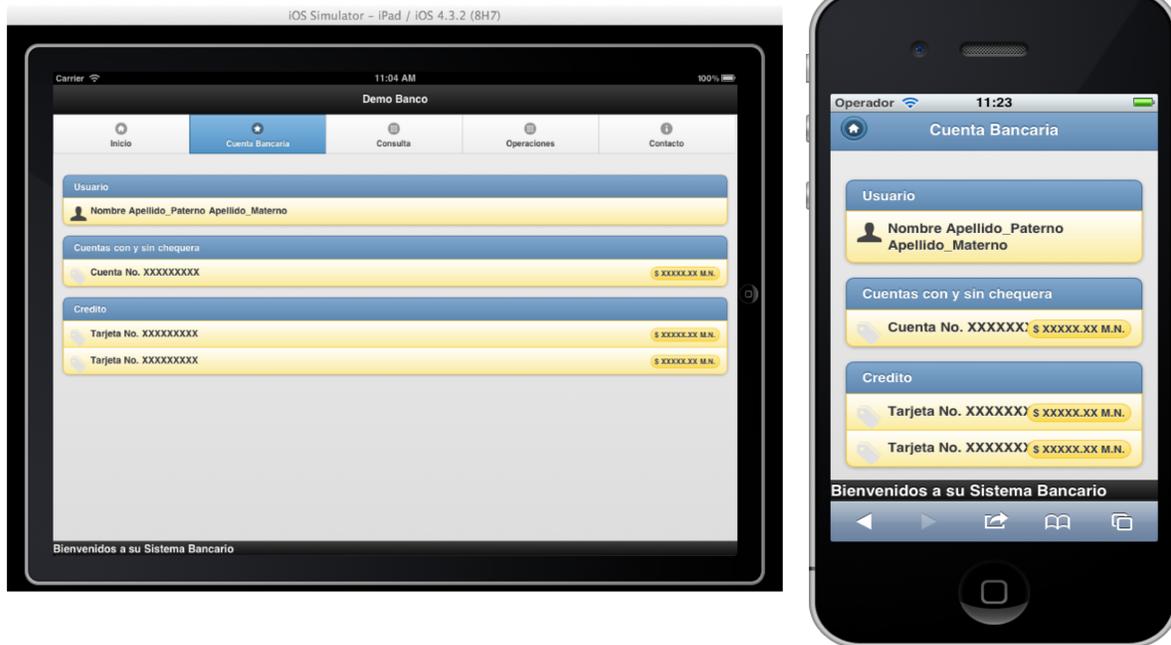


Figura 40: Despliegue de datos de **cuenta bancaria** en iPad y iPhone

Por ende la cantidad de registros que se puede presentar en esta interfaz pueden variar considerablemente, i.e., puede tener desde ningún registro hasta una gran cantidad de ellos.

Para este ejemplo ilustrativo se han colocado una cantidad de 6 registros. Debido al tamaño del iPad, le es permitido desplegar los 6 registros a la vez, como se puede observar en la fig. 41, por lo que no es necesaria la intervención de alguna técnica de remodelación.

Sin embargo, en el caso del iPhone, el asunto es totalmente diferente, ya que como se ilustra en la fig. 42, es necesario dividir el contenido en diversas interfaces de usuario, para llevar a cabo la misma operación, por lo cual los elementos que se ven afectados son:

- Distribución de los componentes gráficos
- Estructura de navegación

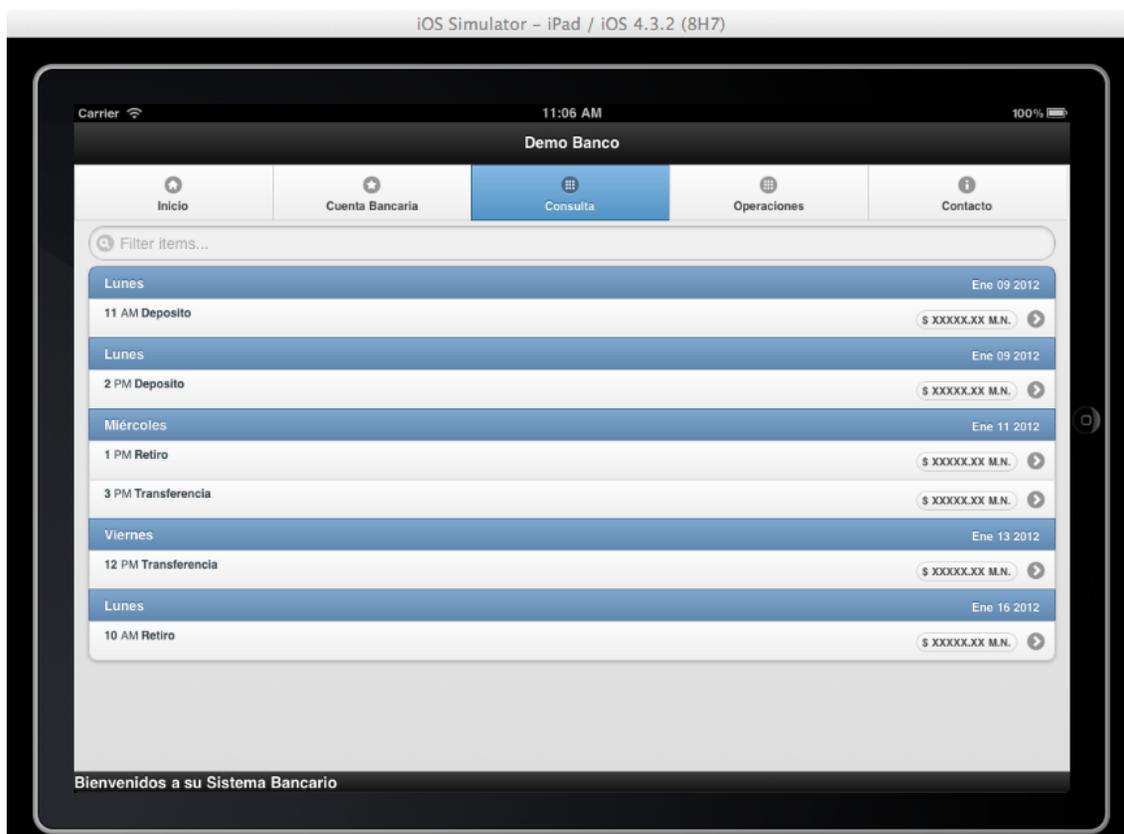


Figura 41: Opción **Consulta** vista desde un iPad

En la fig. 43 se muestra la interfaz gráfica de usuario que nos permite acceder a las operaciones bancarias que se pueden llevar a cabo en el sistema de banca móvil. Las operaciones presentes en este trabajo son: transferencia y pago a tarjeta de crédito.

En las fig. 44 y fig. 45 se puede observar otro caso de aplicación de la técnica de remodelación por reorganización, la cual consiste en redistribuir los componente de entrada de información en diversas pantallas para mejorar la interacción entre el usuario y el dispositivo.

En la fig. 44 se puede observar toda la información que el usuario debe introducir para llevar a cabo la transferencia monetaria.



Figura 42: Opción **Consulta** vista desde un iPhone

Sin embargo en el iPhone, no es posible desplegar todos esos componentes gráficos sin la necesidad de ocultar algunos de ellos.

Por lo tanto en la fig.45, se observa cómo la técnica de reorganización se encarga de determinar el número de interfaces gráficas que son necesarias para que el usuario introduzca la información correspondiente. Particularmente para el iPhone son dos pantallas, modificando la estructura de navegación de la aplicación.

En el caso de la segunda operación bancaria para el **pago a tarjeta de crédito**, esta se muestra en la fig. 46. En ambos casos, la información que contiene la interfaz gráfica se despliega en una sola ocasión, sin necesidad de aplicar alguna técnica de adaptación plástica.

## IMPLEMENTACIÓN DEL MECANISMO PROPUESTO

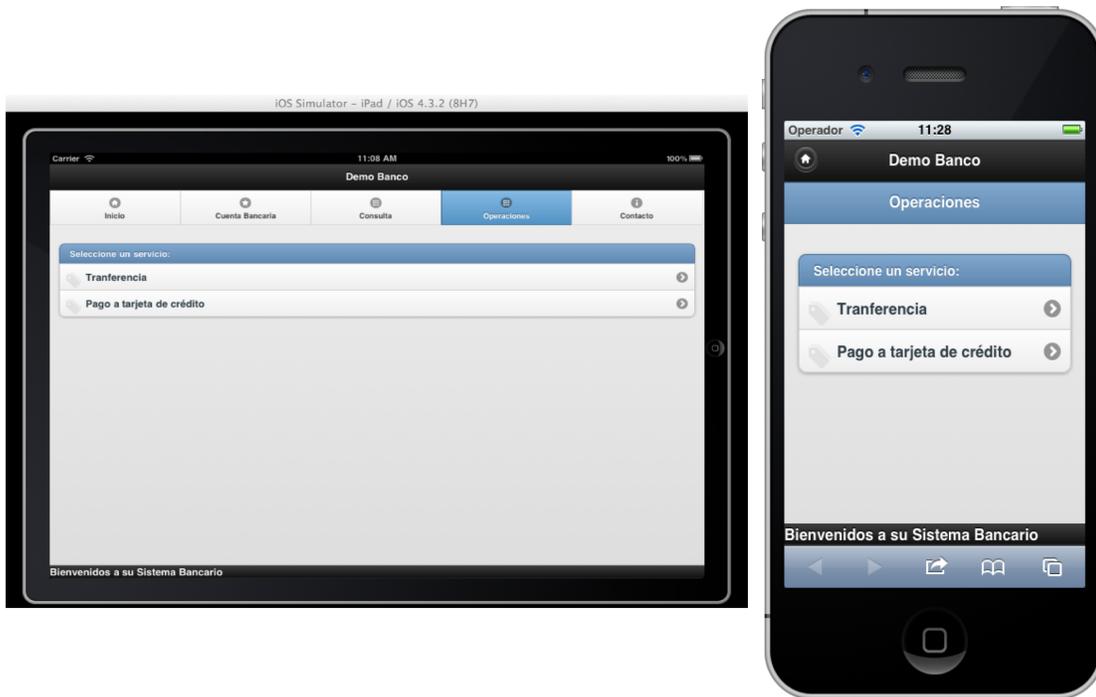


Figura 43: Opción **Operaciones** vista en iPad y iPhone

Finalmente en la fig. 47, se puede observar la última interfaz gráfica de usuario que se diseñó para la aplicación de *software* "Demo Banco". Esta interfaz muestra la información de contacto con el respectivo banco del usuario. Sin embargo al caracer de una gran cantidad de componentes gráficos, esta interfaz no conlleva a ninguna acción por parte del mecanismo de adaptación. Solo su despliegue es necesario.

## 5.5 CASOS DE USO IMPLEMENTADOS

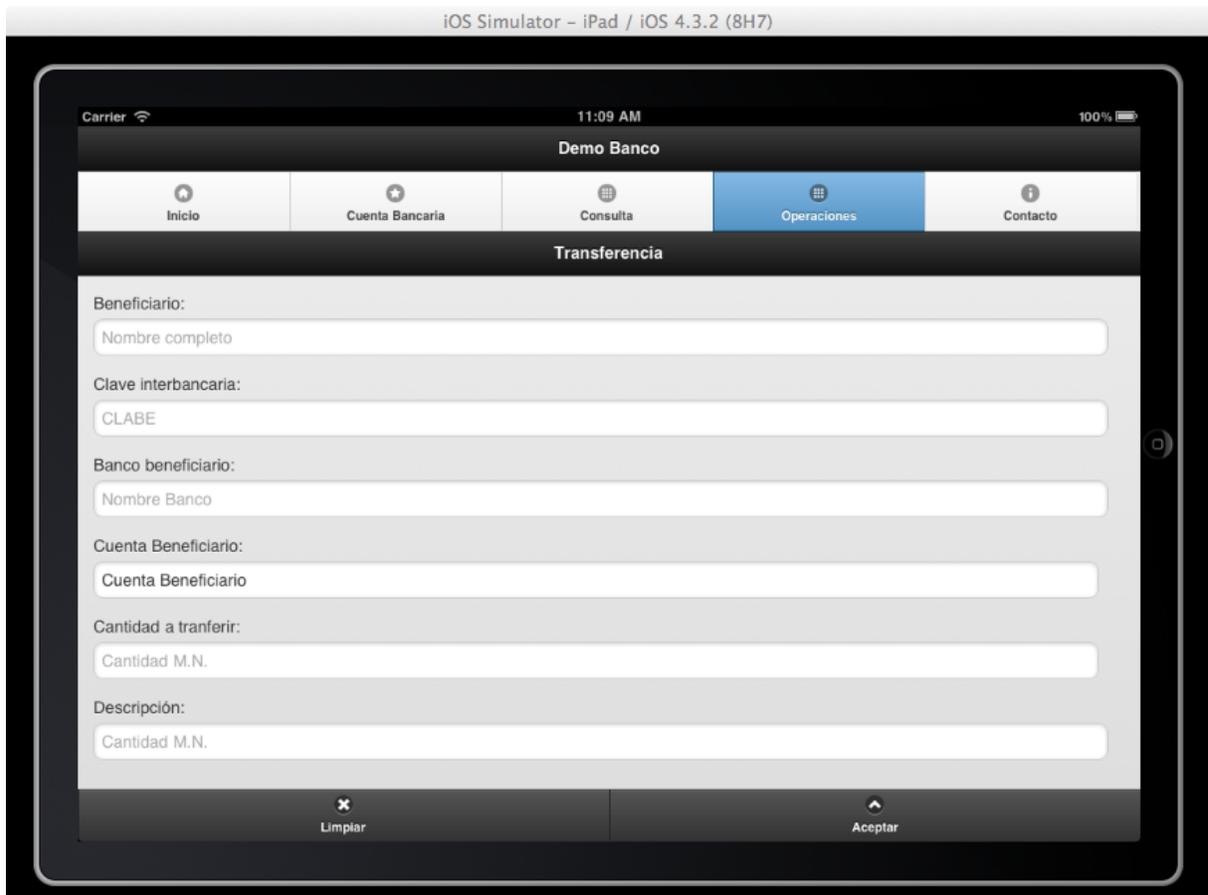


Figura 44: Opción **Transferencia** vista desde un iPad



Figura 45: Opción **Transferencia** vista desde un iPhone

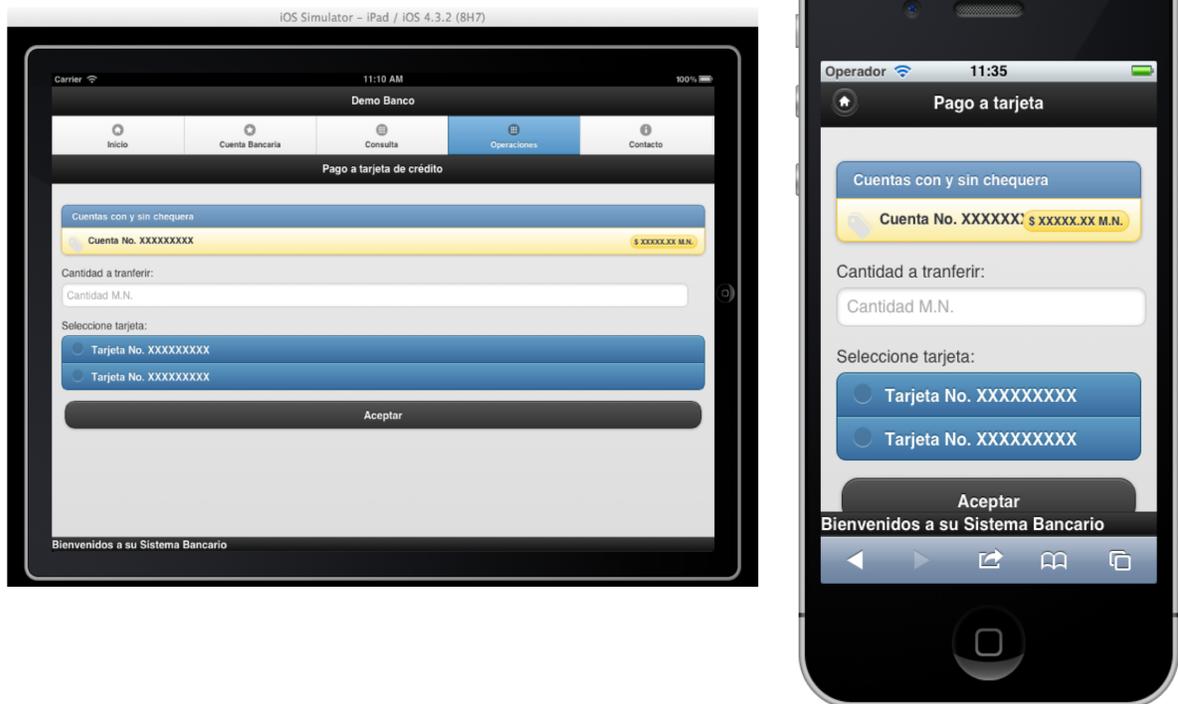


Figura 46: Opción **Pago a tarjeta** vista en iPad y iPhone

## IMPLEMENTACIÓN DEL MECANISMO PROPUESTO

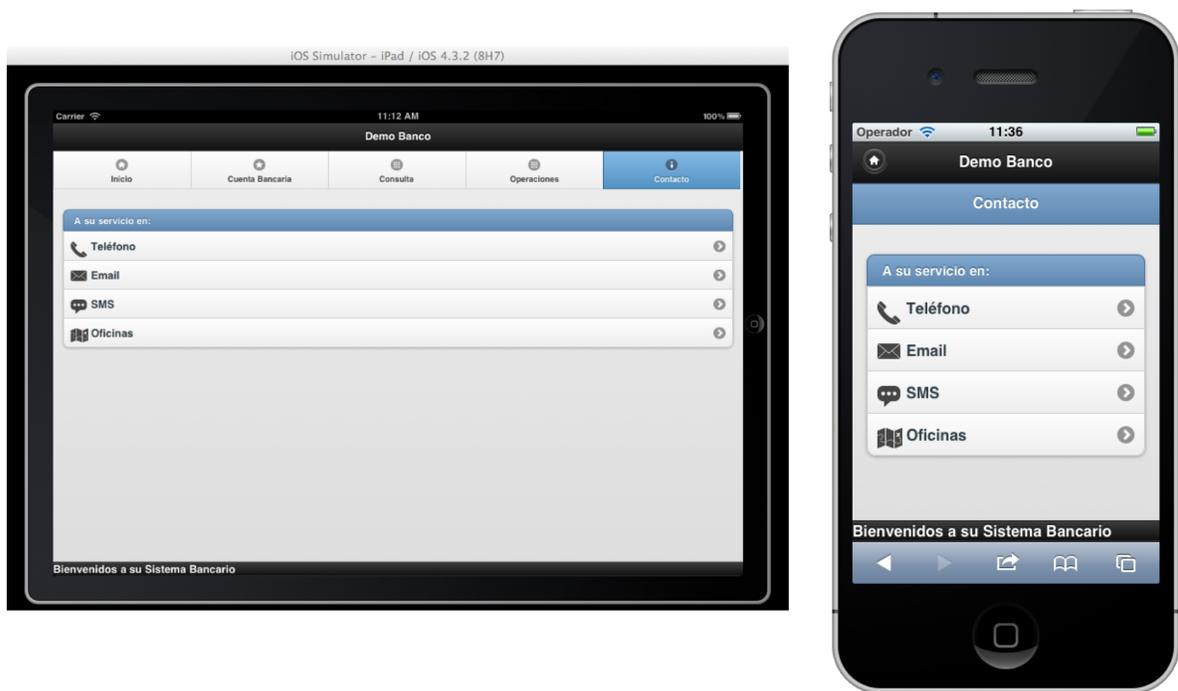


Figura 47: Opción **Contacto** vista en iPad y iPhone

---

## CONCLUSIONES Y TRABAJO FUTURO

---

Constantemente, nuevas plataformas y dispositivos móviles están saliendo al mercado sin embargo, con el afán de atraer al público, los diversos fabricantes incorporan en sus modelos características particulares, propiciando que no exista una uniformidad entre ellos, i. e., se presentan en diversos tamaños y con diferentes capacidades de procesamiento, memoria, almacenamiento y comunicación.

Esta diversidad de dispositivos y plataformas móviles no sólo pone en evidencia la necesidad de acceso ubicuo al procesamiento de la información, sino también impone nuevos requerimientos en el desarrollo de aplicaciones, i.e., adaptación de las interfaces de usuario, desarrollo multi-plataforma, mayor interactividad con el usuario, adecuación a los contextos del usuario y su medio, etc.

Como resultados de la presente tesis se generó:

- Un estudio detallado de las características *hardware* y *software* de los dispositivos móviles, que afectan directamente en el despliegue de las interfaces gráficas de usuario.
- Un espacio de diseño para interfaces gráficas en dispositivos móviles, para el cual se consideran las dimensiones de la plasticidad y la movilidad.
- Un mecanismo de adaptación plástica para dispositivos móviles, el cual se detalla a través de un modelo gráfico.
- Aplicaciones *software*, que permitieron implementar el mecanismo propuesto y las técnicas para la mejora en el desarrollo de *software* sobre dispositivos móviles.

## CONCLUSIONES Y TRABAJO FUTURO

El mecanismo de adaptación plástica surge como una respuesta a la gran diversidad de dispositivos móviles en el mercado y como una opción para homogeneizar el proceso de desarrollo de aplicaciones, al implementar y combinar un desarrollo multi-plataforma y métodos de la plasticidad de interfaces gráficas de usuario. Esto permite un ciclo de desarrollo rápido y eficiente.

Un punto muy importante a destacar es que anteriormente en el diseño e implementación de la plasticidad en dispositivos móviles, no se ha tomado en cuenta sus características particulares y por ende las dimensiones de la movilidad, las cuales permitieron mejorar el rendimiento de la aplicación de software, respecto al hardware del mismo.

El uso de tecnologías Web implicadas en el mecanismo nos permitió un ciclo de desarrollo rápido y con la capacidad de ser implementado en más de una plataforma en el mercado.

Mediante este mecanismo es posible implementar y validar la interfaz de usuario de una aplicación para diferentes dispositivos. Al mismo tiempo, por medio de un desarrollo multiplataforma se puede utilizar el mismo código para múltiples plataformas.

Y finalmente se tiene un modelo arquitectural único para todas las versiones de nuestro software. Como hemos demostrado en la sección implementación, este modelo es adecuado para utilizarse en el desarrollo de *software* para dispositivos móviles con diferentes características.

Como parte del trabajo a futuro, se pretende desarrollar diferentes casos de uso que demuestran cada una de las diferentes adaptaciones que se pueden hacer mediante la aplicación de este mecanismo.

Como una extensión de las propiedades del mecanismo de adaptación plástica es necesario establecer un modelo del grado de dependencia existente entre los elementos gráficos de una interfaz de usuario, con el fin de determinar el correcto comportamiento de un elemento gráfico cuando se lleva a cabo una adaptación plástica, sin afectar la interacción con el usuario.

Otra posible extensión consiste en integrar los métodos de migración y distribución plástica en el mecanismo de adaptación plástica y en desarrollar una aplicación software que permita ejemplificar todos los casos de implementación que se pueden llevar a cabo.

Finalmente, se podría ampliar cada uno de los casos de estudios para contemplar las diferentes plataformas móviles disponibles en el actual mercado, mediante el desarrollo de una librería de software que constituya un intermediario entre cada uno de los métodos de adaptación plástica y los diferentes *frameworks* de desarrollo para aplicaciones móviles.



---

## PUBLICACIONES DEL AUTOR

---

GIRÓN, J. E., MENDOZA, S. and TORRES-HUITZIL, C., *Mechanism for Dynamic Deployment of Plastic Mobile Cross-Platform User Interfaces*, The 2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE' 2011), Mérida, Yucatán, México. 2011. IEEE Computer Society Press, 26-28 October (2011): 847-851, ISBN 978-1-4577-1013-1.



---

## BIBLIOGRAFÍA

---

- [Allen et al., 2010] ALLEN, S., GRAUPERA, V. and LUNDRIGAN, L., *Pro Smartphone Cross-Platform Development: iPhone, Blackberry, Windows Mobile and Android Development and Distribution*, Apress, 2010.
- [Bézivin, 2001] BÉZIVIN, J., *In Search of a Basic Principle for Model Driven Engineering*, Novatica Journal, Special Issue, March-April, 2004.
- [B'Far, 2005] B'FAR, R., *Mobile Computing Principles*, Cambridge University Press, 2005.
- [Casteleyn et al., 2009] CASTELEYN, S., DANIEL, F., DOLOG, P. and MATERA, M., *Engineering Web Applications*, Springer-Verlag, Berlin, 2009.
- [Calvary, 2008] CALVARY, G. and PINNA, A., *Lessons of Experience in Model-Driven Engineering of Interactive Systems: Grand challenges for MDE?*, First International Workshop on Challenges in Model-Driven Software Engineering (ChaMDE), MODELS'08, Toulouse, 2008.
- [Calvary et al., 2006] G. CALVARY, J. COUTAZ, O. DAASSI, V. GANNEAU, L. Balme, A. Demeure and J.-S. SOTTET, *Métamorphose des IHM et Plasticité: Article de synthèse*, Ergo IA 2006, pp. 11-18, 2006.
- [Calvary et al., 2004] G. Calvary, J. Coutaz, O. Dâssi, L. Balme and A. Demeure, *Towards a New Generation of Widgets for Supporting Software Plasticity: The Comet*, In R. Bastide, P. A. Palanque and J. Roth (Eds.), EHCI/DS-VIS, pp. 306-324, 2004.
- [Calvary et al., 2002] G. Calvary, J. Coutaz, D. Thevenin, Q. Limbourg, N. Souchon, L. Bouillon, M. Florins and J. Vanderdonckt, *Plasticity of User Interfaces: A Revised Reference Framework*, TAMODIA 2002: Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design, pp. 127-134, 2002

## Bibliografía

- [Calvary et al., 2001] CALVARY, G., COUTAZ, J. and THEVENIN, D., *A Unifying Reference Framework for the Development of Plastic User Interfaces*, Lecture Notes in Computer Science, Vol. 2254, pp. 173-194, 2001.
- [Coutaz, 2010] COUTAZ, J., *User Interface Plasticity: Model Driven Engineering to the Limit!*, In ACM, Engineering Interactive Computing Systems (EICS 2010), International Conference, pages 1-8. 2010.
- [Coutaz and Calvary, 2008] COUTAZ, J and CALVARY, G., *HCI and Software Engineering: the Case for User Interface Plasticity*, In The Human-Computer Interaction Handbook, Fundamentals, Evolving Technologies and Emerging Applications, Taylor & Francis CRC Press, Chapter 56, pages 1107-1125, 2008.
- [Coutaz et al., 2007] COUTAZ, J., BALME, L., CALVARY, G., DEMEURE, A. and SOTTET, J., *An MDE-SOA Approach to Support Plastic User Interfaces in Ambient Spaces*, In Proc. HCI International 2007, Beijing, pages 152-171, 2007.
- [Dey and Abowd, 2000] DEY, A and ABOWD, D., *Towards a Better Understanding of Context and Context-Awareness*, In the Workshop on The What, Who, Where, When, and How of Context-Awareness, as part of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), The Hague, The Netherlands, 2000.
- [Fling, 2009] FLING, B., *Mobile Design and Development*, O'Reilly Media, 2009.
- [Firtman, 2010] FIRTMAN, M., *Programming the Mobile Web*, O'Reilly Media, 2010.
- [Gauffre and Dubois, 2011] GAUFFRE, G and DUBOIS, E., *Taking Advantage of Model-Driven Engineering Foundations for Mixed Interaction Design*, In Model-Driven Development of Advanced User Interfaces, Springer-Verlag Berlin, pages 219-240, 2011.
- [García et al., 2010] GARCÍA, A., CALVARY, G. and DUPUY-CHESSA, S., *Self-Explanatory User Interfaces by Model-Driven Engineering*, Proceedings of the CHI'10 Workshop on Model Driven Development of Advanced User Interfaces (MDDAUI'10), pages 1-4, 2010.

- [Grolaux et al., 2002] D. Grolaux, P. V. Roy and J. Vanderdonckt, *FlexClock, a Plastic Clock Written in Oz with the QtK toolkit.*, Costin Pribeanu and Jean Vanderdonckt (Eds), TAMODIA, pp. 135-142, 2002.
- [Helander et al., 1997] HELANDER, M., LANDAUER, T. and PRABHU, P., *Handbook of Human-Computer Interaction*, ELSEVIER, pp. 424-426, 1997.
- [Hooper and Berkman, 2012] HOOBER, S. and BERKMAN, E., *Designing Mobile Interfaces*, O'Reilly Media, Canada, 2012.
- [Meyers, 1993] MEYERS, A. *Why Are Human-Computer Interfaces Difficult to Design and Implement?*, Carnegie Mellon University, July 1993.
- [Oehlman and Blanc, 2011] OEHLMAN, D. and BLANC, S., *Pro Android Web Apps*, Apress, 2011.
- [Pérez-Medina et al., 2007] PÉREZ-MEDINA, J., DUPUY-CHESSA, S. and FRONT, A., *A Survey of Model Driven Tools for User Interface Design*, In 6th International Workshop on TAsk Models and DIAgrams (TAMODIA'2007), France, pages 84-97, 2007.
- [Porzel, 2011] PORZEL, R., *Contextual Computing, Models and Applications*, Springer-Verlag, Berlin Heidelberg, 2011
- [Poslad, 2009] POSLAD, S., *Ubiquitous Computing: Smart Devices, Environments and Interactions*, Queen Mary, University of London, WILEY, UK, 2009.
- [Pulli et al., 2008] PULLI, K., AARNIO, T., MIETTINEN, V., ROIMELA, K. and VAARALA, J., *Mobile 3D Graphics with OpenGL ES and M3G*, Elsevier Inc., USA, 2008.
- [Robertson et al., 1991] G. G. Robertson, J. D. Mackinlay and S. K. Card, *Cone Trees: animated 3D visualizations of hierarchical information*, CHI 1991: Proceedings of the SIGCHI conference on Human factors in computing systems, ACM, pp. 189-194, New York, NY , 1991.

## Bibliografia

- [Rothenberg, 1989] ROTHENBERG, J, *The Nature of Modeling*, In Artificial Intelligence, Simulation and Modeling. John Wiley & Sons, Inc., Chichester, 1989.
- [Sottet et al., 2008] SOTTET , J., CALVARY, G., COUTAZ, J. and FAVRE, J. *A Model-Driven Engineering Approach for the Usability of User Interfaces*, Engineering Interactive Systems, Lecture Notes in Computer Science, 2008, Volume 4940/2008, Springer, 2008.
- [Stark, 2010] STARK, J., *Building iPhone Apps with HTML, CSS, and JavaScript* , O'Reilly Media, USA, 2010.
- [Thevenin and Coutaz, 1999] THEVENIN, D. and COUTAZ, J., *Adaptation and Plasticity of User Interfaces*, In Workshop on Adaptive Design of Interactive Multimedia, I3 Spring Days, Barcelone, 1999.
- [Thevenin and Coutaz, 1999] THEVENIN, D. and COUTAZ, J., *Plasticity of User Interfaces: Framework and Research Agenda*, In Proceedings of Interact'99, vol. 1, Edinburgh: IFIP, IOS Press, pp. 110-117, 1999.
- [Thevenin et al., 2002] THEVENIN, D., CALVARY, G. and COUTAZ, J., *A Development Process for Plastic User Interfaces*, In Companion UIST'2002 proceedings. 2002.
- [Vanderdonckt et al., 2008] VANDERDONCKT, J., CALVARY, G., COUTAZ, J. and STANCIULESCU A., *Multimodality for Plastic User Interfaces: Models, Methods, and Principles*, In Multimodal User Interfaces. Signals and Communication Technology, Springer, 2008.
- [Vanderdonckt et al., 2006] VANDERDONCKT, J., CALVARY, G. and FAVRE, J., *Mapping Model: A First Step to Ensure Usability for sustaining User Interface Plasticity*, In Model Driven Development of Advanced User Interfaces ( MDDAUI 2006), 2006.
- [Virkus, 2011] VIRKUS, R., *Dont Panic Mobile Developers Guide to the Galaxy*, Enough Software, 7 ed., 2010. <http://www.enough.de/>

[Woo et al., 2010] Woo, J., SOHN, J., NAM, B. and Yoo, H., *MOBILE 3D GRAPHICS SoC, From Algorithm to Chip*, John Wiley & Sons, Singapore, 2010.

[Zaphiris and Siang, 2009] ZAPHIRIS, P. and SIANG, C., *Human Computer Interaction: Concepts, Methodologies, Tools, and Applications*, Information Science reference, UK, 2009.