



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

Redistribución semi-plástica mixta:
el caso de estudio de un pizarrón compartido

Tesis que presenta

Gabriela Sánchez Morales

para obtener el Grado de

Maestra en Ciencias

en Computación

Director de la Tesis

Dra. Sonia Guadalupe Mendoza Chapa

México, D.F.

Enero 2009

Resumen

Las soluciones y los prototipos propuestos en este trabajo de tesis se sitúan a la intersección de dos campos de investigación: el primero es el dominio del Trabajo Cooperativo Asistido por Computadora (TCAC o CSCW en inglés), el cual permite a los colaboradores de participar cooperativamente en la producción de entidades compartidas e intercambiar mensajes para coordinarse, aún cuando ellos no puedan reunirse físicamente en el mismo lugar y/o al mismo tiempo, e.g., editores cooperativos de documentos compartidos y distribuidos en Internet.

El otro dominio de investigación abordado es la Interacción Hombre-Máquina (IHM), la cual tiene como objetivo estudiar y proponer soluciones más ergonómicas y poderosas para las interfaces de interacción del usuario con su ambiente de trabajo y con sus aplicaciones.

El diseño y el desarrollo de interfaces de usuario plásticas constituyen una dirección de investigación prometedora: la plasticidad de una interfaz de usuario le permite adaptarse dinámicamente a las características del usuario, del entorno actual, de la plataforma y/o de los dispositivos de trabajo.

A pesar de que algunos sistemas colaborativos ya empezaron a aplicar la plasticidad de interfaces de usuario, este tema queda casi inexplorado. La motivación de este trabajo se centró en diseñar y prototipar un sistema colaborativo, el cual se adapte al usuario y a la plataforma. A nuestro saber ningún sistema colaborativo se adapta al usuario. Así, el sistema propuesto se adapta a la plataforma, ya que puede ser ejecutado en una PDA y en una PC, y también se adapta al usuario cuando detecta a un mismo colaborador en estos diferentes dispositivos.

Cuando el usuario usa a la vez una PDA y una PC, el sistema propuesto permite dividir la interfaz de usuario: una parte en la PDA y su complemento en la PC, las cuales funcionan de manera transparente como si fueran colocadas en la misma máquina. Además, en todo momento el sistema propuesto permite conocer los colaboradores en la sesión actual.

El sistema cooperativo propuesto es un pizarrón compartido, el cual permite escribir texto y dibujar figuras pre-diseñadas o a mano alzada en el área de dibujo del pizarrón, con la finalidad de soportar una sesión cooperativa de lluvia de ideas.

Este trabajo abre el campo de investigación de la plasticidad de interfaces de usuario en un ambiente de trabajo cooperativo. A partir de los resultados de este trabajo, se puede imaginar lógicamente la definición de conceptos y mecanismos de plasticidad genéricos que podrían adaptarse a cualquier aplicación cooperativa.

Palabras clave: plasticidad de interfaces de usuario, sistemas cooperativos, contexto de uso, distribución transparente de interfaz de usuario.

Abstract

The solutions and the prototypes that have been proposed in this thesis work are at the intersection of two research domains: first, applications of the Computer Supported Cooperative Work (CSCW) that allow collaborators to cooperatively produce shared entities and, to exchange messages to coordinate themselves even if they are not located in the same space and/or they are not working at the same time, e.g., cooperative editors of shared documents that are distributed on the Internet.

Computer Human Interaction (CHI) is the other addressed research domain in this work. Its objective is to study and to propose more ergonomic and efficient solutions for the user interaction interfaces of both his/her working environment and his/her applications.

The design and the implementation of plastic user interfaces constitute a promising research direction: the plasticity property of a user interface allows it to be dynamically adapted to the characteristics of the user, of his/her current environment, of the platform and/or of his/her working devices.

In spite of some collaborative systems already apply some plasticity elements within their user interface definition, this topic remains mainly unexplored. The motivation of this work is centered on designing and prototyping a collaborative system that adapts itself to both the user and the platform. As we can know, such kind of system does not exist. In this way, the proposed system adapts itself to the platform as it can be executed either on a PDA (handheld device) or on a personal computer. In a same way, it can adapt itself to the user when it detects him working on several devices.

When a user uses at the same time both a PDA and a PC, the proposed system allows to split the user interface: a part on the PDA and the complement on the PC. The distributed interface parts work in a transparent way as if they were colocalized on the same computer. Moreover, at any time, the proposed system maintains the knowledge about the session users.

The proposed cooperative system is a shared whiteboard that allows to add writing and figures (predefined or dynamically defined within the whiteboard space) in order to efficiently support a cooperative brainstorming session.

This study opens the research field of user plastic interfaces for the cooperative work environments. From the results obtained from this research effort, we can logically imagine the possibility to define plastic generic concepts and mechanisms that could be adapted to any kinds of cooperative applications.

Keywords: Plasticity of user interfaces, cooperative systems, context of use, transparent distribution of the user interface.

Índice general

Índice de figuras	viii
Índice de tablas	xi
1 Introducción	1
1.1 Contexto de investigación	1
1.2 Planteamiento del problema	3
1.3 Objetivos del proyecto	6
1.4 Organización de la tesis	7
2 Plasticidad de interfaces gráficas de usuario	9
2.1 Tipos de metamorfosis de las interfaces de usuario	9
2.2 Definición de plasticidad	12
2.2.1 Proceso de adaptación plástica	13
2.2.2 Tipos de percutores	15
2.2.3 Parámetros de la adaptación	16
2.3 Trabajos relacionados	17
2.3.1 Sitio Web Sedan-Bouillon	17
2.3.2 Sistema de control de calefacción para casas	18
2.3.3 Teclado numérico virtual	21
2.3.4 Reloj y calendario digital (<i>FlexClock</i>)	21
2.3.5 Mesas de trabajo (<i>ConnecTables</i>)	23
2.3.6 Entorno de arquitectura real colaborativa (<i>Roomware</i>)	24
2.4 Estudio comparativo	26
3 Análisis y diseño	35
3.1 Escenario	35
3.2 Arquitectura del pizarrón compartido	37
3.2.1 Definición del grano de adaptación	37
3.2.2 Estructura de una aplicación mono-usuario	38
3.2.3 Estructura de una aplicación multi-usuario	40
3.3 Nodo principal del pizarrón compartido	42
3.3.1 Almacenamiento de información	43
3.3.2 Módulo de comunicación	43
3.3.3 Redistribución de la interfaz de usuario	46

3.4	Interactores del pizarrón compartido	49
3.4.1	Barra de herramientas	50
3.4.2	área de dibujo	51
3.4.3	Barra de colaboradores	52
3.4.4	Meta-interfaz de redistribución	54
3.4.5	Ventana de autenticación de usuario	56
4	Implementación	59
4.1	Selección del lenguaje de programación	59
4.1.1	PDA con Linux	59
4.1.2	PDA con Windows Mobil 5.0	61
4.1.3	Selección de la herramienta para la PC	62
4.2	Interfaz de usuario	62
4.2.1	Interfaz de usuario para la PC	63
4.2.2	Interfaz de usuario para la PDA	66
4.3	Implementación de los módulos más relevantes	74
4.3.1	Interactores en el área de despliegue de la PDA	74
4.3.2	área de dibujo	77
4.3.3	Comunicación entre la PDA y la PC	83
4.4	Pruebas y resultados	84
5	Conclusiones y trabajo a futuro	91
5.1	Resumen de la problemática	91
5.2	Conclusiones	92
5.3	Trabajo a futuro	93
A	Instalaciones y configuraciones	95
A.1	OpenEmbedded	95
A.1.1	Primera instalación	95
A.1.2	Segunda instalación	96
A.2	Instalación de Opie	97
A.3	Instalación de Kaffe	97
A.4	Instalación de JWT	98
	Bibliografía	99

Índice de figuras

1.1	Contexto de investigación de la presente tesis	3
1.2	Información de presencia por medio de fotos	5
1.3	Información de localización mediante barras coloreadas	5
1.4	Espacio de trabajo producción (superior) y vista radar (inferior)	6
1.5	Organización del documento	7
2.1	La metáfora del pintor <i>Pick and Drop</i> , la PDA funciona como paleta de herramientas y el pizarrón como lienzo	10
2.2	El sistema IO/Brush tiene un pincel mágico que obtiene color, textura y movimiento del mundo físico	10
2.3	Superficies aumentadas (mesas y sillas) que se adaptan a la inclinación	11
2.4	Espacio de clasificación para la plasticidad de interfaces de usuario	14
2.5	Sistemas representativos del estado del arte	17
2.6	Versión ligera plástica del sitio Sedan-Bouillon: a) interfaz gráfica de usuario para la PC, b) Meta-UI en la PDA, c) título y menú de navegación en la PDA y d) menú de navegación y contenido en la PDA	19
2.7	Control de calefacción para casas: a) la interfaz de usuario en la PC muestra al mismo tiempo la temperatura de todos los cuartos, mientras que b) la interfaz de usuario en la PDA muestra por separado la temperatura de cada cuarto	20
2.8	Control de calefacción para casas desde un teléfono celular: a) el usuario selecciona la opción sala, b) el sistema indica la temperatura actual de dicho cuarto y c) el usuario modifica la temperatura	21
2.9	Interfaz de usuario del teclado numérico virtual cuando el teclado físico: a) no está disponible y b) está disponible	22
2.10	Interfaces de usuario de <i>FlexClock</i> para: a) una PC, b) un PSP, c) una PDA, d) un teléfono celular y e) un termómetro digital	23
2.11	Juan y Abel unen sus <i>ConnecTables</i> para trabajar colaborativamente	24
2.12	a) Juan transfiere a Abel una copia de la imagen, arrastrándola hacia la otra pantalla y posteriormente b) Abel señala una parte de la imagen, encerrándola con un semicírculo	25
2.13	Componentes de <i>Roomware</i> : <i>ConnecTables</i> , <i>DynaWall</i> , <i>CommChair</i> e <i>InteracTable</i>	26
2.14	Interacción por medio de <i>Roomware</i> : a) Diana trabaja desde su <i>CommChair</i> y b) su colega Alicia usa <i>DynaWall</i>	26

3.1	Escenario de trabajo cooperativo usando el pizarrón compartido plástico	36
3.2	Asociación entre el árbol MVC y el pizarrón compartido	40
3.3	Arquitectura del pizarrón compartido antes de la redistribución	40
3.4	Arquitectura del pizarrón compartido después de la redistribución	41
3.5	Componentes del proceso de adaptación	42
3.6	Comandos de figuras: a) creación, b) modificación y c) solicitud	44
3.7	Comandos de colaboradores: a) entrada a una sesión, b) salida de una sesión y c) solicitud de información sobre los colaboradores	45
3.8	Comando usado para: a) redistribuir la barra de herramientas y b) la redistribuir colaboradores	46
3.9	Proceso de adaptación plástica	48
3.10	Caso de uso de la barra de herramientas	50
3.11	Actualización del área de dibujo en los sitios A y B	52
3.12	Actualización de la barra de colaboradores	54
3.13	Redistribución de la barra de herramientas entre el sitio A y B	56
3.14	Autenticación de un colaborador	57
4.1	Interactores del pizarrón compartido	62
4.2	Interfaces de usuario propuestas para el pizarrón compartido en la PC	64
4.3	Menú del tercer prototipo de la interfaz de usuario	65
4.4	Primer prototipo de la interfaz de usuario para la PDA, a) la ventana principal muestra el área de dibujo, mientras que b) la barra de herramientas así como la meta-interfaz de redistribución y la barra de colaboradores se despliegan en una ventana secundaria	67
4.5	Segundo prototipo de la interfaz de usuario para la PDA, la cual se compone de: a) una ventana principal, b) una ventana secundaria, c) una primera variante de la ventana secundaria y c) una segunda variante de la ventana secundaria	68
4.6	Tercer prototipo de la interfaz de usuario para la PDA, la cual muestra cinco ventanas de siete en total	69
4.7	Cuarto prototipo de la interfaz de usuario para la PDA, la cual muestra tres ventanas de cuatro	70
4.8	El quinto prototipo muestra en dos posiciones diferentes la barra de herramientas: a) horizontal y b) vertical	71
4.9	Análisis del tamaño de la PDA 6945	75
4.10	Distribución de interactores para teléfono inteligente	76
4.11	Análisis de división del área de despliegue	79
4.12	División del área de dibujo en el teléfono inteligente	79
4.13	área de despliegue que utiliza la PC y la PDA	80
4.14	La autenticación en la PC se compone de: a) la ventana de autenticación y b) la ventana de usuario validado	84
4.15	Ventana principal del pizarrón compartido en la PC	85
4.16	El pizarrón compartido en la PDA informa que estableció conexión con el servidor	86
4.17	Autenticación del usuario en el teléfono inteligente	86

ÍNDICE DE FIGURAS

xi

4.18	Ventana principal del pizarrón compartido en el teléfono inteligente	87
4.19	La barra de colaboradores muestra la llegada de la colaboradora Gaby en la PC	88
4.20	Ventana de la meta-interfaz de redistribución en la PC	88
4.21	La PDA despliega la barra de herramientas y la barra de colaboradores, mientras que la PC muestra el área de dibujo	89

Lista de Tablas

2.1	Parámetros del contexto de uso	27
2.2	Sistemas mono-usuario vs multi-usuario	28
2.3	Remodelación	28
2.4	Redistribución	29
2.5	Tipo de adaptación plástica	29
2.6	Grano de adaptación	30
2.7	Espacios tecnológicos	31
2.8	Implantación de la adaptación	32
2.9	Grano de reanudación	32
2.10	Producción de las interfaces de usuario	33
2.11	Usabilidad	33
4.1	Máquinas virtuales para ejecutar Java en una PDA	61
4.2	Opciones elegidas para pintar sobre el área de dibujo con el color azul	72

Capítulo 1

Introducción

1.1 Contexto de investigación

La teoría de actividad [Baecker et al., 1994] toma en cuenta la situación de una acción en la etapa inicial del proceso de diseño. Sin embargo, la información dependiente de situación se pierde en el proceso de desarrollo debido a la falta tanto de notaciones apropiadas de diseño como de herramientas de desarrollo. En consecuencia, las herramientas actuales implícitamente asumen que la Interacción Hombre-Máquina (IHM) se confina a una sola plataforma, lo cual implica que el usuario en la mayoría de los casos: 1) se vuelva sedentario cuando está interactuando con su computadora personal (PC), la cual está localizada en un lugar específico; e 2) interactúe con su PC mediante dispositivos de entrada/salida tradicionales, e.g., monitor, teclado y ratón [Calvary et al., 2001].

Sin embargo, la creciente proliferación de dispositivos de cómputo, así como el progreso de las redes de comunicación y de 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]. Esta diversidad de dispositivos fijos y móviles no sólo pone en evidencia la necesidad de acceso ubicuo al procesamiento de la información, sino también impone nuevos requerimientos a las aplicaciones, tales como la capacidad de adaptación de una interfaz de usuario a diferentes contextos de uso.

El concepto de “contexto de uso” [Calvary et al., 2002] se define como el estado actual de un conjunto de variables que caracterizan, tanto los dispositivos de cómputo utilizados para interactuar con una aplicación (e.g., el tamaño de la pantalla, el ancho de banda, la duración de la batería y las capacidades de almacenamiento, procesamiento y comunicación), como los entornos físico y social donde se lleva a cabo la interacción. Algunos ejemplos de variables de entorno son los objetos, las personas y los eventos que son periféricos a la tarea actual, pero que pueden tener un impacto en el comportamiento del: 1) sistema, e.g., un entorno ruidoso puede eliminar la realimentación por audio y/o 2) usuario, e.g., la localización en el espacio provee un contexto para la relevancia de la información; así la tarea de escribir un artículo, que es central en una oficina, puede

volverse secundaria en un tren.

Evidentemente, la interfaz gráfica de usuario no puede ser la misma sobre una pantalla grande y una pantalla pequeña. La solución más obvia al problema del limitado espacio de despliegue consiste en reducir el tamaño de los componentes gráficos cuando estos necesitan aparecer en la misma vista. La reducción del tamaño de los componentes gráficos puede alterar su presentación, e.g., cambiando un grupo de botones de radio (*radio button*) por un lista de opciones (*combo box*). Si la reducción de los componentes no es posible, otra opción es dividir la interfaz gráfica de usuario en varias subsecciones (e.g., espacios de trabajo). El concepto de “plasticidad” propone estudiar y proveer soluciones atractivas para hacer frente a este problema [Crease, 2001].

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. Una 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. A guisa de ilustración, GSM (*Global System for Mobile communications*) no funciona en todas partes, situación que presenta una discontinuidad potencial a nivel del entorno. De manera análoga, un sistema que funciona en Palm y en teléfono celular, pero no en Psion, expone una discontinuidad latente a nivel de la plataforma [Calvary et al., 2001].

Esta tesis de maestría se inscribe en el dominio de investigación denominado Trabajo Cooperativo Asistido por Computadora (TCAC). Se trata de un dominio multidisciplinario que estudia tanto los aspectos sociales de las actividades individuales y colectivas, como los aspectos tecnológicos de la información y de las comunicaciones, con el fin de asistir la colaboración entre personas. En relación con el carácter multidisciplinario del TCAC, este trabajo de investigación toma principalmente la perspectiva tecnológica para diseñar e implementar interfaces plásticas para sistemas colaborativos (cf. Figura 1.1). Esta clase de sistemas provee entornos de trabajo computarizados que permiten a los colaboradores intercambiar mensajes, compartir información y colaborar, aún cuando ellos no puedan reunirse físicamente en el mismo lugar y/o al mismo tiempo.

Desde el punto de vista conceptual, los servicios de un sistema colaborativo se agrupan en tres espacios funcionales [Ellis and Wainer, 1994]: 1) el **espacio de producción** designa los objetos, e.g., diagramas, que resultan de una actividad de grupo; 2) el **espacio de coordinación** define actores (e.g., individuos, grupos y roles), identifica actividades y tareas así como sus relaciones temporales y designa los actores responsables de dichas actividades y tareas; y finalmente 3) el **espacio de comunicación** ofrece a los actores la posibilidad de intercambiar información. Según el dominio de aplicación de un sistema co-

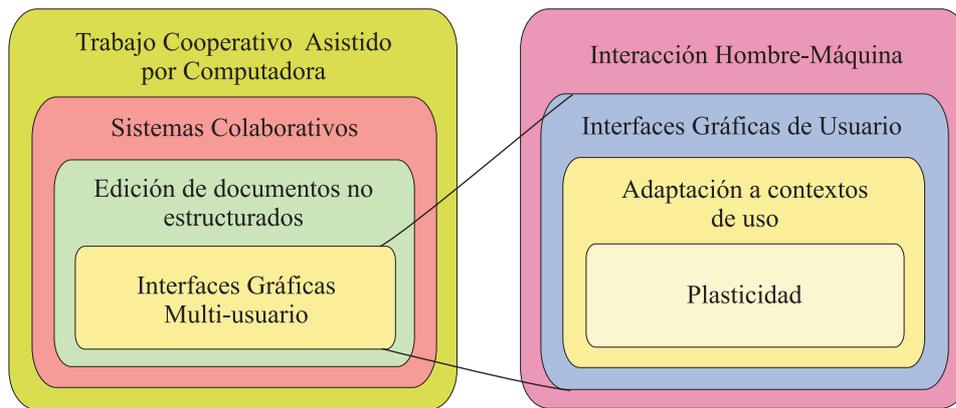


Figura 1.1: Contexto de investigación de la presente tesis

laborativo, los tres espacios funcionales no tienen la misma importancia, e.g., los sistemas de edición privilegian la producción, los sistemas de flujo de trabajo (*workflow*) hacen hincapié en la coordinación y los espacios de medios (*mediaspaces*) enfatizan la comunicación.

En particular, este trabajo de investigación se inscribe en el dominio de aplicación de la edición de documentos no estructurados en pizarrones compartidos, los cuales están diseñados para soportar grupos pequeños de colaboradores (menos de 15) que interactúan de manera síncrona (i.e., al mismo tiempo). Se eligió este dominio de aplicación por dos razones principales. La primera se refiere a las características del tipo de aplicación, ya que el núcleo funcional de un pizarrón compartido es relativamente sencillo, mientras que su interfaz gráfica multi-usuario puede volverse relativamente compleja. Puesto que el interés principal del presente trabajo de investigación es el estudio de la plasticidad de interfaces gráficas multi-usuario, las características del pizarrón compartido permiten concentrarnos en el logro de nuestro objetivo, sin tener que hacer frente a problemas de orden secundario.

La segunda razón se refiere a la madurez del campo de estudio. Los desarrolladores de interfaces gráficas multi-usuario han logrado avances significativos en el diseño de componentes gráficos para sistemas que privilegian el espacio de producción. Así, la exploración de un nuevo concepto (en este caso, el estudio de la plasticidad de interfaces gráficas multi-usuario) requiere un campo de prueba relativamente bien conocido para poder entender sus implicaciones. Puesto que el pizarrón compartido privilegia el espacio de producción, el concepto de plasticidad se aplicará a aquellos componentes gráficos cuya relevancia para el usuario ha sido puesta en evidencia en diversos estudios de usabilidad, e.g., [Gutwin et al., 1996], [Gutwin and Roseman, 1996] y [Gutwin and Greenberg, 1999].

1.2 Planteamiento del problema

Las interfaces gráficas de usuario normalmente se diseñan en base a métodos centrados en el usuario, los cuales se complementan con criterios ergonómicos [Draper et al., 1986]. A guisa de ejemplo, según el referencial propuesto por Bastien y Scapin, el criterio

“Guía”/subcriterio “Regreso de Información”, el cual preconiza el despliegue de información en el lugar donde se espera que el usuario mirará, se interpreta como: “desplegar la información en la zona más visible de la pantalla”. En este mismo referencial, el criterio “Carga de Trabajo”/subcriterio “Acciones Mínimas”, el cual incita a la trayectoria más corta posible, se entiende como: “reducir el número de acciones físicas necesarias (e.g., desplazamientos de ratón y entrada de datos por teclado) para el logro de un objetivo” [[Bastien and Scapin, 1993](#)].

La complejidad asociada al desarrollo de interfaces gráficas de usuario ha conducido al fracaso de los generadores automáticos [[Myers et al., 2000](#)]. En este ámbito, el desconfinamiento de las interfaces gráficas de usuario y la incorporación del soporte para la interacción entre usuarios añaden complejidad al problema.

A diferencia de una interfaz mono-usuario, que provee componentes gráficos para permitir al usuario hacer uso de las funciones de la aplicación, una interfaz multi-usuario ofrece componentes gráficos adicionales para reemplazar la información (de coordinación y comunicación) que se pierde cuando sus múltiples usuarios no interactúan cara a cara. Puesto que la distancia física disminuye drásticamente la comunicación entre los colaboradores, es necesario proveer mecanismos que informen sobre la actividad de los demás, con el fin de crear una conciencia grupal a todos los individuos que forman parte de un grupo. Algunos ejemplos de este tipo de información se refieren a la presencia de los miembros del grupo, a su localización en el espacio de trabajo y a las acciones que efectúan sobre las entidades compartidas.

En cuanto a los componentes gráficos que proveen información de conciencia de grupo, se han realizado importantes esfuerzos tanto a nivel de diseño como a nivel de implementación. Un ejemplo representativo de estos esfuerzos es la caja de herramientas Groupkit [[Roseman and Greenberg, 1996](#)] que ofrece a los desarrolladores de aplicaciones un conjunto de componentes gráficos para facilitar la construcción de interfaces multi-usuario. Otros ejemplos representativos de este tipo de componentes gráficos fueron originalmente propuestos por las aplicaciones de edición colaborativa GROVE [[Ellis et al., 1991](#)], Calliope [[Mitchell, 1996](#)] y SASSE [[Baecker et al., 1994](#)]:

1. GROVE muestra la información de presencia mediante un arreglo de fotos posicionado en la parte inferior de la pantalla (cf. Figura 1.2);
2. Calliope representa la localización de los colaboradores en el espacio de trabajo (i.e., documento) por medio de barras de colores (cada colaborador está asociado a un color), las cuales están posicionadas en el lado izquierdo de la barra de desplazamiento (cf. Figura 1.3);
3. SASSE divide la pantalla en dos partes: en la parte superior muestra una vista parcial del espacio de trabajo, mientras que en la parte inferior despliega una vista miniatura del espacio de trabajo completo. Esta última presenta la vista y la localización específica de cada colaborador mediante recuadros coloreados y cursores miniatura, respectivamente (cf. Figura 1.4).

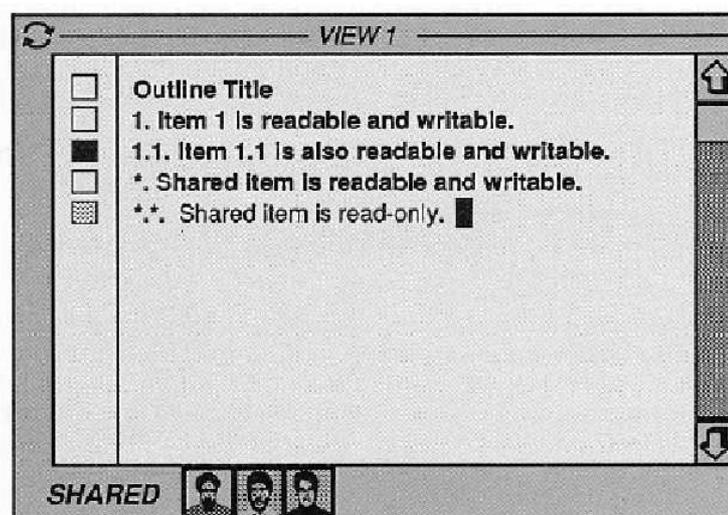


Figura 1.2: Información de presencia por medio de fotos

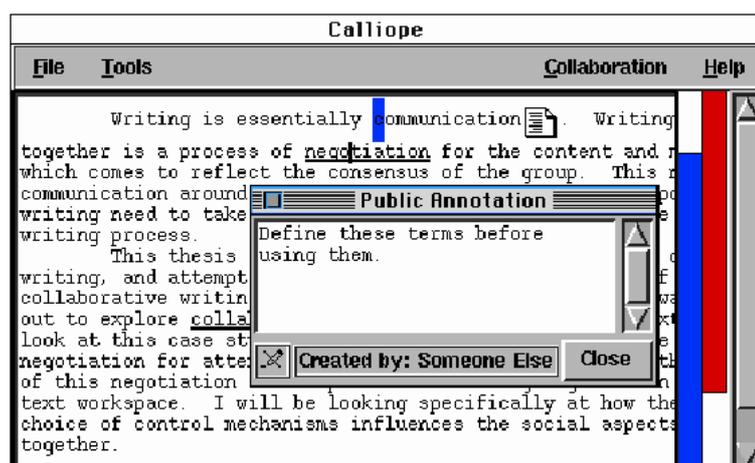


Figura 1.3: Información de localización mediante barras coloreadas

Así, la interfaz gráfica de un sistema colaborativo es un superconjunto de la interfaz gráfica de un sistema mono-usuario, si ambos sistemas soportan el mismo dominio de aplicación. De hecho, la administración del espacio de despliegue constituye una actividad necesaria en el diseño de interfaces gráficas de sistemas mono-usuario, a causa de la limitada superficie que el espacio de despliegue ofrece para posicionar todos los componentes necesarios para interactuar con la aplicación. Sin embargo, esta actividad adquiere aún mayor importancia en el diseño de interfaces gráficas de sistemas colaborativos, debido a la necesidad de desplegar componentes adicionales para soportar la colaboración entre usuarios distribuidos.

Aunque los campos de investigación la IHM y del TCAC se han fortalecido mutuamente desde el nacimiento del segundo, la plasticidad de interfaces gráficas es prácticamente un tema de investigación inexplorado por la comunidad científica del TCAC, a pesar de la

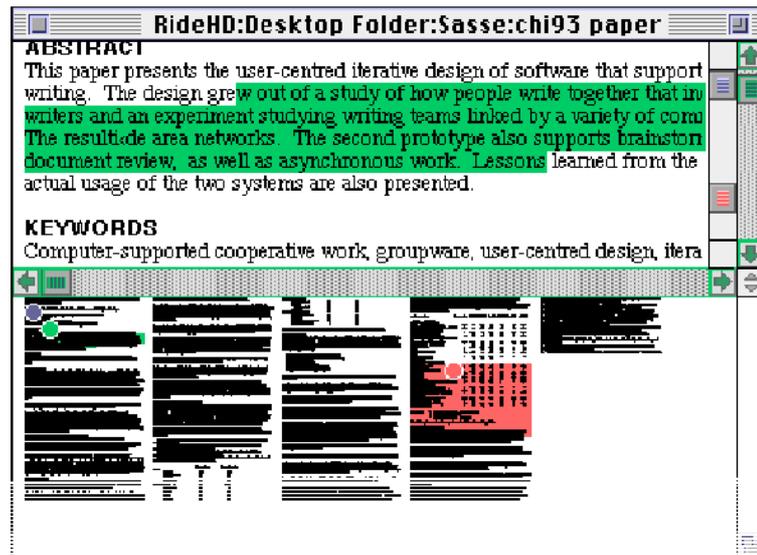


Figura 1.4: Espacio de trabajo producción (superior) y vista radar (inferior)

necesidad inminente de dotar a las interfaces gráficas multi-usuario de la propiedad de adaptación a cambios contextuales.

1.3 Objetivos del proyecto

Objetivo general

Incorporar la propiedad de plasticidad en una interfaz gráfica multi-usuario para dotarla de un carácter adaptativo a cambios en el contexto de uso. El campo de estudio y experimentación de este trabajo de investigación es una aplicación de pizarrón compartido.

Objetivos particulares

1. Identificar, diseñar e implementar los componentes gráficos tanto del dominio de aplicación como de conciencia de grupo que conformarán la interfaz plástica del pizarrón compartido. Esta aplicación colaborativa será accesible desde una PC (*Personal Computer*) y desde una PDA (*Personal Digital Assistant*).
2. Diseñar e implementar un mecanismo de plasticidad que permita que la interfaz gráfica del pizarrón compartido no sólo se adapte a cambios en el contexto de uso, sino también responda a las acciones concurrentes de los colaboradores.

1.4 Organización de la tesis

La presente tesis está estructurada en cinco capítulos (cf. Figura 1.5).

El capítulo 2 comprende el estado del arte, en este se describe los conceptos relacionados con la plasticidad en el dominio de las interfaces gráficas de usuario, así como las principales características plásticas de los sistemas relacionados con este trabajo de investigación. A partir de estas características se destacan, mediante un estudio comparativo, las principales ventajas y desventajas de los sistemas analizados.

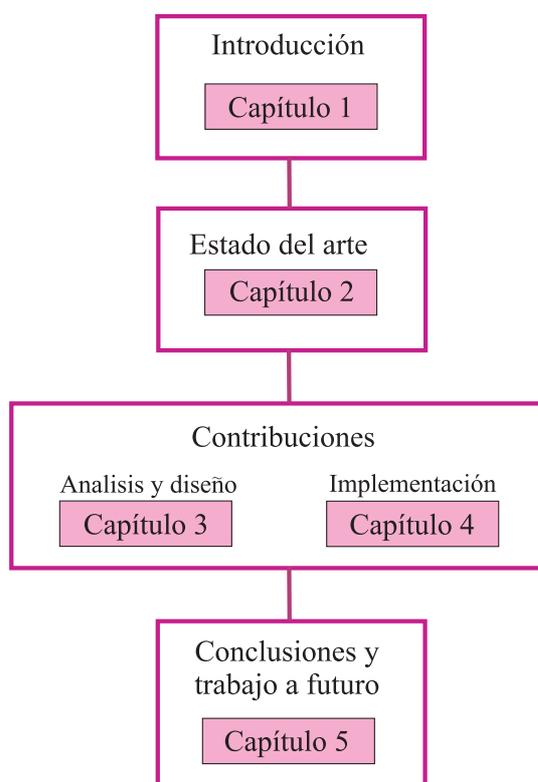


Figura 1.5: Organización del documento

Las contribuciones del presente trabajo se reflejan en los capítulos 3 y 4. En el capítulo 3 se detalla el análisis y el diseño del sistema. En particular, se describen los principales interactores de trabajo que conforman la interfaz gráfica del pizarrón compartido así como las funciones disponibles en cada interactores. Después se detallan los componentes gráficos asociados a las funciones del pizarrón compartido y el lugar que ocupan en las pantallas de la PDA y de la PC, antes y después de la redistribución. Posteriormente se describe el protocolo de comunicación utilizado para realizar la redistribución y la actualización de los componentes gráficos de la interfaz de usuario. Finalmente se detalla el diseño del pizarrón compartido mediante diagramas UML (*Unified Modeling Language*). Los diagramas más relevantes destacan los siguientes aspectos: la comunicación entre los dispositivos, la actualización de la interfaz gráfica en cada dispositivo, el ingreso y la sal-

ida de un colaborador y la redistribución del pizarrón compartido.

En el capítulo 4 se describe la implementación del sistema, así como los problemas encontrados durante esta fase a causa de las características técnicas de las PDAs utilizadas. Asimismo se detallan las pruebas que muestran la redistribución del pizarrón compartido entre una PDA y una PC.

Finalmente en el capítulo 5 se presentan las conclusiones de este trabajo de investigación, así como algunas ideas de trabajo a futuro que podrían mejorar el pizarrón compartido propuesto en cuanto a su capacidad plástica, su funcionalidad básica y su soporte de colaboración.

Capítulo 2

Plasticidad de interfaces gráficas de usuario

El presente capítulo está estructurado en cuatro partes. La primera parte explica los tipos de transformaciones que han sido identificadas en las interfaces de usuario, e.g., la transición de una interfaz rígida a una plástica (sección 2.1). La segunda parte del capítulo proporciona los conceptos básicos de la plasticidad en el dominio de las interfaces gráficas de usuario (sección 2.2). Este marco conceptual está estructurado de la siguiente forma: primero se describe una propuesta de proceso para llevar a cabo la adaptación plástica de un sistema interactivo; posteriormente se explican los tipos de remodelación y redistribución que han sido identificados en las interfaces de usuario; por último se detallan algunos parámetros de la adaptación que servirán de guía para realizar el análisis comparativo de los sistemas más relevantes del estado del arte. La tercera parte del capítulo (sección 2.3) describe precisamente las principales características plásticas de estos sistemas. Finalmente en la cuarta parte del capítulo se detalla el estudio comparativo de los sistemas analizados, destacando sus principales ventajas y desventajas (sección 2.4).

2.1 Tipos de metamorfosis de las interfaces de usuario

Se pueden distinguir seis tipos de metamorfosis [Calvary et al., 2006] de las interfaces de usuario :

1. **De lo centralizado a lo distribuido:** se refiere a la redistribución de la interfaz de usuario en diversas plataformas. La interfaz de usuario no se concentra en un solo monitor, sino que se reparte en un conjunto de plataformas, aprovechando sus características intrínsecas e extrínsecas. A guisa de ejemplo, considere la metáfora del pintor que ha sido explorada mediante la técnica de manipulación directa *Pick-and-Drop* [Rekimoto, 1997]. Según esta metáfora, la PDA es percibida por su carácter móvil (intrínseco) como una paleta de herramientas desde la cual se pueden tomar colores, dibujos y videos, en tanto que el pizarrón interactivo sirve como lienzo (cf. Figura 2.1).

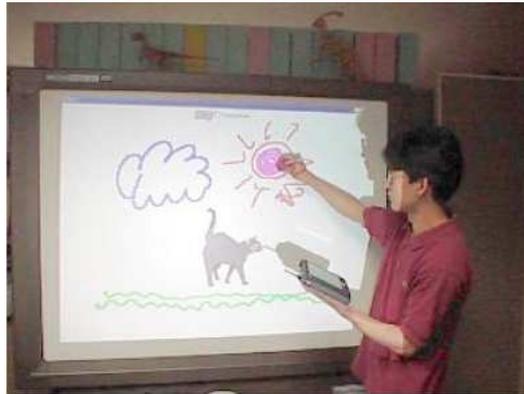


Figura 2.1: La metáfora del pintor *Pick and Drop*, la PDA funciona como paleta de herramientas y el pizarrón como lienzo

2. **De lo clásico a lo exótico:** se refiere al desconfinamiento de la interfaz de usuario en beneficio del mundo físico. Así, todo objeto del entorno, desde el momento en que es percibido y/o manipulable por el sistema, se vuelve un dispositivo potencial de E/S. En consecuencia, tales objetos se vuelven parte de la plataforma de la misma manera que los monitores, teclados y ratones. Mientras que las computadoras personales de oficina ostentan incansablemente estos dispositivos de E/S, los prototipos de laboratorios rivalizan en imaginación tanto en dispositivos de entrada como de salida. Un ejemplo representativo es el sistema IO Brush [Ryokai et al., 2005], el cual ofrece un pincel mágico que extrae motivos (e.g., color, textura, movimiento) del mundo físico y después los reproduce y manipula en mundo virtual (cf. Figura 2.2).

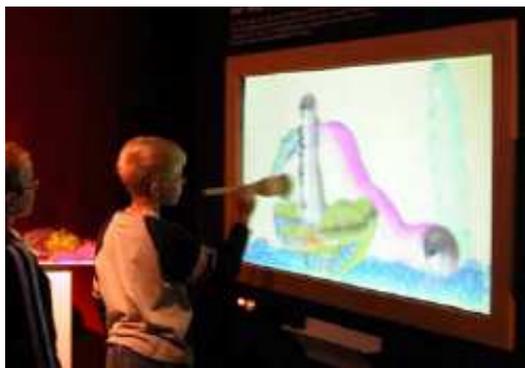


Figura 2.2: El sistema IO/Brush tiene un pincel mágico que obtiene color, textura y movimiento del mundo físico

3. **De lo monomodal a lo multimodal:** se refiere a una remodelación de la interfaz de usuario, en función de los sentidos humanos utilizados para ingresar la entrada y percibir la salida de datos. El dominio de la Interacción Hombre-Máquina clásicamente ha privilegiado los sentidos de la vista, del tacto y del oído. Una excepción representativa de esta tendencia es el proyecto *Exhalia* de *France Telecom*

que explora el sentido del olfato. Cuando el usuario navega en la Web, las imágenes están acompañadas de olores. Aún si las investigaciones actualmente son prospectivas, se podría imaginar, en términos de seguridad, la expresión de un peligro o del reforzamiento de la señal por medio de la difusión de un olor. En consecuencia, el usuario no necesita estar pegado al monitor, ya que el olor es difundido en el lugar donde se encuentra para satisfacer la exigencia de retorno de información, tal como lo expresa el referencial propuesto por Bastien y Scapin [Bastien and Scapin, 1993].

4. **De lo explícito a lo implícito:** se refiere a la redistribución de las tareas entre el usuario y el sistema. Hasta hace poco, las acciones físicas explícitas del usuario sobre los dispositivos de entrada dirigían la interacción hombre-máquina, sin embargo tales acciones explícitas comienzan a perder su monopolio: la oficina del usuario ahora es capaz de escucharlo, de comprender el propósito del discurso y de completar el contenido con información desplegada en el muro (interacción implícita).
5. **Del sedentarismo al nomadismo:** se refiere a la redistribución dinámica de la interfaz de usuario. Normalmente adherida a su soporte de ejecución (i.e., una computadora), la interfaz de usuario comienza a desplazarse, a voluntad del usuario, en su espacio de interacción en función de la llegada o la partida de recursos. La interfaz de usuario migra parcial o totalmente cambiando así su estado de distribución y adaptándose, en caso necesario, a las capacidades de la plataforma desmetronidazoltino. A guisa de ilustración, las superficies aumentadas de Rekimoto [Rekimoto and Saitoh, 1999] (cf. Figura 2.3) permiten que la presentación de los objetos (e.g., mesas y sillas) se adapte a la inclinación horizontal (vista desde arriba en 2D) o vertical (perspectiva 3D) de la superficie de despliegue (i.e., mesa aumentada o pantalla, respectivamente).

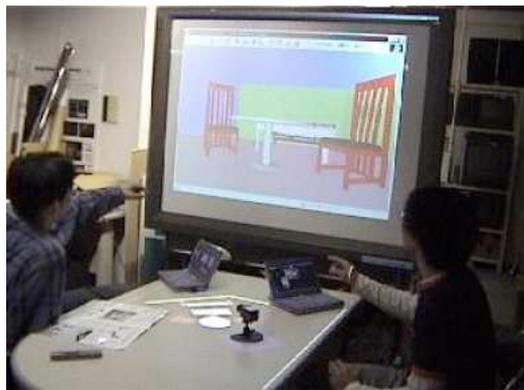


Figura 2.3: Superficies aumentadas (mesas y sillas) que se adaptan a la inclinación

6. **De la rigidez a la plasticidad:** se refiere a una remodelación de la interfaz de usuario en función del contexto de uso. Mientras que hasta hace poco tiempo la interfaz de usuario respondía únicamente a las acciones del usuario, de ahora en adelante éstas comienzan a adaptarse a un contexto de uso en constante evolución. Se habla de plasticidad cuando la adaptación respeta la usabilidad. Este tipo de

metamorfosis de la interfaz de usuario es el objeto de estudio del presente trabajo de investigación.

2.2 Definición de plasticidad

En la actualidad las interfaces gráficas de usuario ya no se limitan a una única computadora de escritorio, puesto que pueden ser distribuidas o migradas hacia un conjunto dinámico de plataformas de interacción. Si el conjunto está formado de dispositivos heterogéneos (e.g., PC y PDA), la redistribución podría requerir la remodelación de la interfaz gráfica de usuario para poder ser adaptada a la plataforma deseada [Sottet et al., 2006].

Retomando la definición del Capítulo 1 se dice que: la plasticidad denota la capacidad de una interfaz gráfica de usuario para soportar variaciones en el contexto de uso, preservando la usabilidad [Calvary et al., 2006].

Contexto de uso

La movilidad de las personas junto con 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. Por **contexto de uso** se entiende la terna «usuario, plataforma, entorno» donde [Calvary et al., 2004]:

1. el **usuario** denota el arquetipo humano que tiene por objeto utilizar el sistema interactivo,
2. la **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];
3. el **entorno** describe las condiciones físicas y sociales donde tiene lugar la interacción [Calvary et al., 2004], e.g., los objetos, las personas y los eventos que son periféricos a la tarea que se está desarrollando, en un momento dado, pero que pueden tener un impacto en el comportamiento: a) del sistema, e.g., un entorno ruidoso puede eliminar la realimentación por audio, y/o b) del usuario, e.g., su localización provee un contexto para la relevancia de la información [Calvary et al., 2001].

Continuidad y usabilidad

Una primera definición de usabilidad dice que usabilidad es el esfuerzo requerido para aprender, operar, preparar la entrada e interpretar la salida de un sistema [McCall, 1977].

Otra definición dice que una interfaz de usuario plástica preserva su “usabilidad” si las propiedades seleccionadas, durante el diseño, se mantienen dentro un rango de valores cuando ocurre la adaptación a cambios contextuales [Calvary et al., 2001]. Por las definiciones anteriores se puede decir que la usabilidad caracteriza la capacidad de un sistema para permitir al usuario lograr sus objetivos (obtener un resultado con una calidad dada) con eficacia, en confort y seguridad (la del usuario y la de otros).

Otra característica ineludible de la plasticidad es la “garantía de continuidad” cuando se produce la migración del sistema de un contexto de uso a otro. A guisa de ilustración, GSM (*Global System for Mobile communications*) no funciona en cualquier parte del mundo, situación que presenta una discontinuidad potencial a nivel del entorno [Calvary et al., 2001]. De manera análoga, un sistema que funciona en Palm y en teléfono móvil pero no en Psion expone una discontinuidad latente a nivel de la plataforma [Calvary et al., 2001].

2.2.1 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 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 de usuario.
3. **Selección de una solución particular:** usa alguna estrategia de resolución de problemas para elegir la interfaz de usuario más adecuada.
4. **Transición entre estados:** realiza la transición de la interfaz 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 de usuario mientras no ocurran las siguientes condiciones de adaptación. Dos técnicas de ejecución han sido identificadas, en donde la interfaz de usuario puede: a) ser ejecutada desde el principio o b) preservar el estado del sistema.

Cada uno de los cinco pasos implicados en el proceso de adaptación puede ser ejecutado por el sistema, por el usuario o por ambos. A este respecto, se pueden distinguir tres tipos de plasticidad:

- **Adaptativa:** el sistema es capaz de manejar los cinco pasos sin intervención humana;

- **Adaptable:** el usuario ejecuta manualmente los cinco pasos;
- **Mixta:** cubre una combinación de intervenciones tanto del usuario como del sistema.

La Figura 2.4 muestra un espacio de clasificación para la plasticidad de interfaces de usuario. El eje vertical de la gráfica indica el autor de la adaptación (i.e., el sistema, el usuario o ambos) y el eje horizontal señala los parámetros del contexto de uso (i.e., la plataforma, el entorno o ambos) a los que se adaptará la interfaz de usuario. Respecto a este último aspecto, se puede distinguir dos tipos de adaptación:

- **Semi-plasticidad:** sólo se realiza la adaptación al entorno o a la plataforma;
- **Plasticidad completa:** se lleva a cabo la adaptación tanto al entorno como a la plataforma.

De acuerdo con esta clasificación de plasticidad, el presente trabajo de investigación está ubicado en el cuadro “semi-plasticidad mixta” (de donde se desprende el título de la tesis) porque el pizarrón compartido propuesto: 1) se adapta a la plataforma (i.e., el sistema puede ejecutarse en una PC y en una PDA) y 2) permite que el usuario y el sistema realicen conjuntamente la adaptación plástica. Cabe mencionar que esta clasificación de plasticidad no contempla la adaptación al usuario, la cual si ha sido tomada en cuenta en el desarrollo del sistema propuesto en esta tesis.

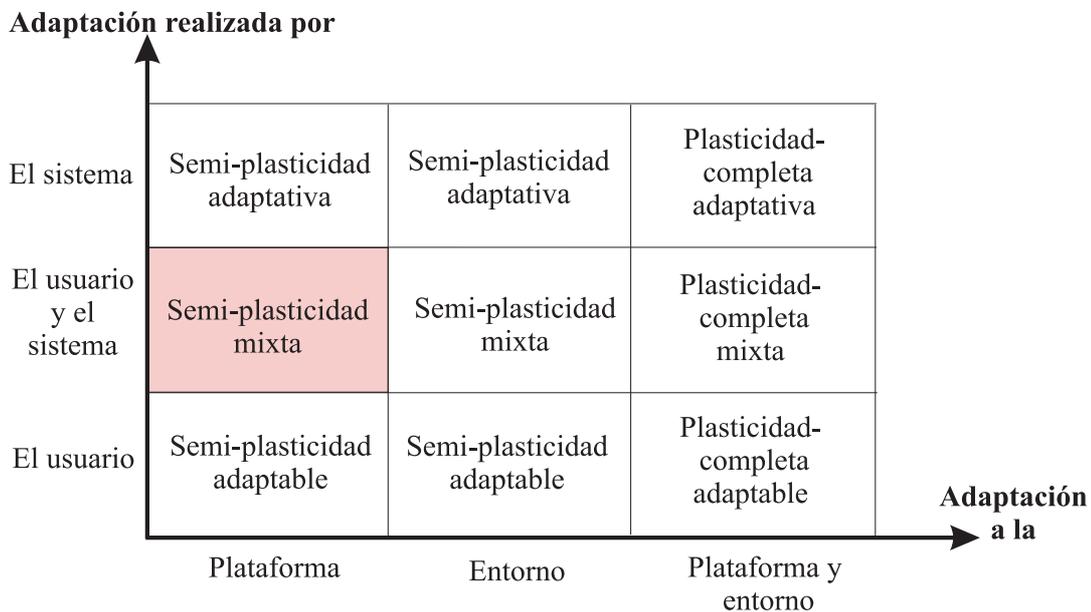


Figura 2.4: Espacio de clasificación para la plasticidad de interfaces de usuario

2.2.2 Tipos de percutores

Se han identificado dos percutores principales de la plasticidad [Calvary et al., 2006]:

1. **Remodelación:** actúa sobre la presentación de los conceptos del dominio y de las tareas del usuario.

Los investigadores de la plasticidad de interfaces de usuario comenzaron precisamente por la remodelación. Uno de los primeros resultados fue la identificación de los niveles de abstracción en los que la adaptación puede tener lugar [Thevenin, 2001]. Se aprovechó el modelo de la arquitectura ARCH para distinguir cinco niveles de abstracción: 1) el núcleo funcional (NF), 2) el adaptador del núcleo funcional (ANF), el controlador de diálogo (CD) y las presentaciones lógica y física (PL y PF, respectivamente) [Phillips, 1999]. En aquellos tiempos, los investigadores consideraban un solo cuadro gráfico (i.e. la pantalla de la PC). Desde entonces, se pueden considerar otras modalidades, e.g., la vocal [Berti y Paternò, 2005]. En consecuencia, se vuelve pertinente precisar si las modalidades humanas serán preservadas o no durante la remodelación. A este respecto, se distinguen tres tipos de remodelación:

- a) **intra-modal:** la modalidad se preserva, e.g., de gráfico a gráfico,
 - b) **inter-modal o transmodal:** la modalidad cambia, e.g. de gráfico a vocal, y
 - c) **multimodal:** las modalidades se combinan, e.g., gráfica y vocal.
2. **Redistribución:** se refiere a la dispersión de la interfaz de usuario en diferentes plataformas. Se identifican cuatro tipos de redistribución:
 - a) **de centralizada a centralizada:** conserva el estado centralizado de una interfaz de usuario, e.g., migración total de una PC a un PDA;
 - b) **de centralizada a distribuida:** desmiembra una interfaz de usuario haciéndola pasar de un estado centralizado a un estado distribuido, e.g., repartición entre una PC y un PDA;
 - c) **de distribuida a centralizada:** reconcentra una interfaz de usuario en una plataforma, haciéndola pasar de un estado distribuido a un estado centralizado;
 - d) **de distribuida a distribuida:** cambia el estado de distribución.

Puesto que la interfaz de usuario puede estar distribuida, cada plataforma juega un rol específico:

- a) **plataformas complementarias:** cada una está encargada de un subconjunto de tareas del usuario, y
- b) **plataformas totalmente equivalentes:** permiten al usuario realizar su tarea tanto en una PC como en una PDA, según su conveniencia.

2.2.3 Parámetros de la adaptación

El cálculo de la reacción engendrada por cambios en el contexto de uso puede llevarse a cabo considerando los siguientes seis parámetros [Calvary et al., 2006]:

1. **Grano de la adaptación:** indica el tamaño de grano que se toma como base para llevar a cabo la adaptación:
 - a) **interactor:** es una entidad capaz de mostrar (parte de) su estado, en algún medio de presentación [Duke and Harrison, 1993], e.g., compactación de un juego de botones de radio en un menú desplegable;
 - b) **espacio de trabajo:** es un conjunto de tareas lógicamente conectadas, e.g. modificación de una ventana, o
 - c) **interfaz de usuario completa**
2. **Localización interna y/o externa de la adaptación:** se trata de decidir quién embarca los mecanismos de adaptación:
 - a) una **parte la interfaz de usuario**, i.e. interactor, espacio de trabajo, o
 - b) un **tercero**, e.g., una infraestructura de adaptación.
3. **Espacios tecnológicos:** se refiere a los espacios tecnológicos en el sentido de la Ingeniería Dirigida por Modelos, e.g., Java y HTML. De esta manera, la adaptación puede ser:
 - a) **intra-espacio**, e.g., de Java a Java,
 - b) **inter-espacios**, e.g., de HTML a Java,
 - c) **multi-espacios**, e.g., combinación de Java y HTML antes y después de la transformación de la interfaz de usuario.
4. **Implantación de la adaptación:** se trata de decidir lo que el usuario puede o debe hacer durante la adaptación. Dos formas son posibles:
 - a) **implantación estática:** obliga al usuario a salir de su sesión de trabajo durante el tiempo que tome la adaptación, o
 - b) **implantación dinámica:** permite al usuario continuar su tarea, ya que la adaptación se hace durante el funcionamiento y el uso del sistema.
5. **Grano de reanudación:** permite medir en términos de acciones físicas el costo de la adaptación para el usuario. Se pueden identificar tres granos:
 - a) **acción física:** significa que el usuario no pierde ninguna acción durante la adaptación;
 - b) **tarea:** se refiere a que sólo las tareas terminadas son restauradas, mientras que las acciones físicas que contribuyen a la realización de una nueva tarea se pierden, y

- c) **sesión**: significa que el usuario comienza desde principio perdiendo el beneficio de todas sus acciones.
6. **Producción de las interfaces de usuario**: se refiere a la forma de generar la interfaz de usuario para cada contexto de uso:
- estáticas**: están prefabricadas,
 - dinámicas**: son generadas en tiempo de ejecución.

2.3 Trabajos relacionados

A continuación se describen los trabajos más representativos de la plasticidad de interfaces de usuario (cf. Figura 2.5). La mayoría de los sistemas analizados permite al usuario trabajar de forma individual, aunque algunos sistemas ya ofrecen un soporte básico para el trabajo colaborativo. Algunos sistemas tienen la característica de redistribuir automáticamente la interfaz de usuario, mientras otros involucran al usuario para tomar conjuntamente la decisión de la redistribución, etc. Después de proporcionar una descripción de cada uno de los sistemas analizados se hace un estudio comparativo.

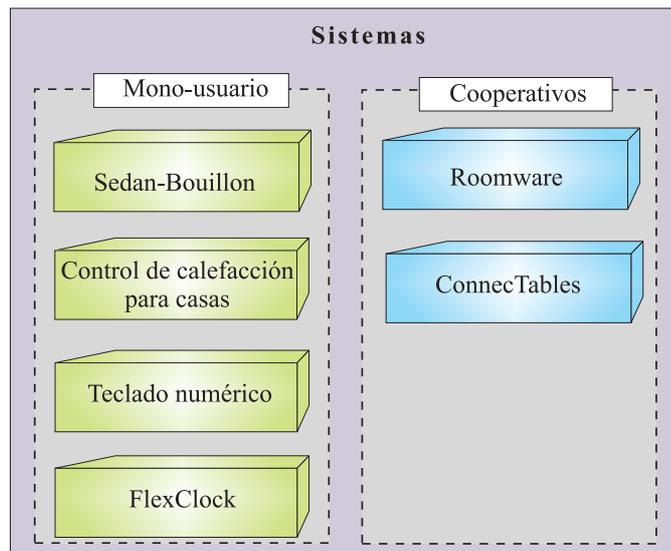


Figura 2.5: Sistemas representativos del estado del arte

2.3.1 Sitio Web Sedan-Bouillon

Sedan-Bouillon [Balme et al.,2005] es un sitio Web que tiene como objetivo hacer promoción turística de las ciudades de Sedan en Francia y de Bouillon en Bélgica. Originalmente los usuarios disponían de la siguiente información en alemán, francés y holandés: una breve explicación de Sedan y Bouillon, los lugares turísticos de interés (e.g., monumentos históricos) y la logística de la estancia (e.g., hoteles y restaurantes). Sedan-Bouillon

fue diseñado por la compañía *Intelligent Systems, Solutions & Services* (IS3) en 2003.

Inicialmente el sitio Web sólo era accesible desde PCs. Posteriormente, el proyecto CAMELEON creó una versión ligera plástica llamada **LSB** (*Light Sedan Bouillon*) con el fin de explorar las implicaciones de redistribuir el sitio Web entre una PC y una PDA. La versión LSB únicamente ofrece información de hoteles en francés.

El sitio Web LSB proporciona tres componentes gráficos (*widgets*) en la interfaz de usuario: el **título**, el **menú de navegación** y el **contenido** (cf. Figura 2.6.a). El menú de navegación ofrece una opción llamada Meta-UI (*Meta-User Interface*), que permite al usuario controlar la redistribución de la interfaz de usuario a nivel de componente gráfico. Así, el usuario puede elegir el tipo de pantalla (PC o PDA) sobre la cual desearía desplegar cada componente gráfico. Todas las combinaciones de redistribución son posibles: desde la duplicación de todos los componentes gráficos en ambos dispositivos hasta la repartición entre la PC y la PDA de algunos componentes y la omisión de los restantes.

Para ilustrar esta forma de plasticidad, considere que el usuario Lionel ingresa al sitio Web LSB desde su PC, previa validación de su nombre y contraseña. Para poder navegar confortablemente desde el sofá, Lionel enciende su PDA e ingresa nuevamente al sitio Web LSB utilizando el mismo identificador. En respuesta, la Meta-UI de la PDA le informa que está utilizando dos navegadores (`log_Lionel_0` y `log_Lionel_1`) para acceder al sitio Web LSB y que puede especificar qué componente gráfico desearía ejecutar en qué plataforma. Lionel solicita (cf. Figura 2.6.b):

1. el título (*title*) sea replicado tanto en la PC (`log_Lionel_0`) como en la PDA (`log_Lionel_1`)
2. el contenido (*contenu*) sea únicamente desplegado en la PC y
3. el menú de navegación (*navigation*) sólo sea visualizado en la PDA (cf. Figura 2.6.c).

De esta manera, cuando el usuario selecciona un hotel desde la PDA, la información del hotel se despliega en la PC.

Después, Lionel selecciona la Meta-UI de la PDA para redistribuir nuevamente el sitio Web. Esta vez, Lionel elige replicar el menú de navegación y el contenido en ambos dispositivos. Así, el título no se muestra en la PDA ni en la PC (cf. Figura 2.6.d).

2.3.2 Sistema de control de calefacción para casas

El sistema de control de calefacción [Calvary et al., 2001] propuesto por EDF (*French Electricity Company*) puede ser utilizado en diferentes contextos de uso. De esta manera la calefacción puede ser controlada:

- desde la casa por medio de un dispositivo montado en la pared o desde una PDA conectada a la red inalámbrica de dicha casa;
- desde la oficina utilizando una PC con conexión a un servidor Web,



Figura 2.6: Versión ligera plástica del sitio Sedan-Bouillon: a) interfaz gráfica de usuario para la PC, b) Meta-UI en la PDA, c) título y menú de navegación en la PDA y d) menú de navegación y contenido en la PDA

- desde cualquier lugar mediante un teléfono celular equipado con WAP (*Wireless Application Protocol*).

El sistema fue creado con la herramienta ARTStudio (*Adaptation by Reification and Translation*), la cual está implementada en Java y en CLIPS (un lenguaje basado en reglas para generar interfaces de usuario abstractas y concretas).

Una tarea típica de un usuario del sistema de calefacción consiste en consultar y modificar la temperatura de algún cuarto. Los cuartos en observación son la recámara, el baño y la sala. La interfaz de usuario, accesible desde una PC, permite mostrar al mismo tiempo la temperatura de todos los cuartos (cf. Figura 2.7.a). En cambio la interfaz de

usuario, desplegable en la pantalla de una PDA, sólo permite visualizar la temperatura de un cuarto a la vez mediante la selección de la pestaña correspondiente (cf. Figura 2.7.b).

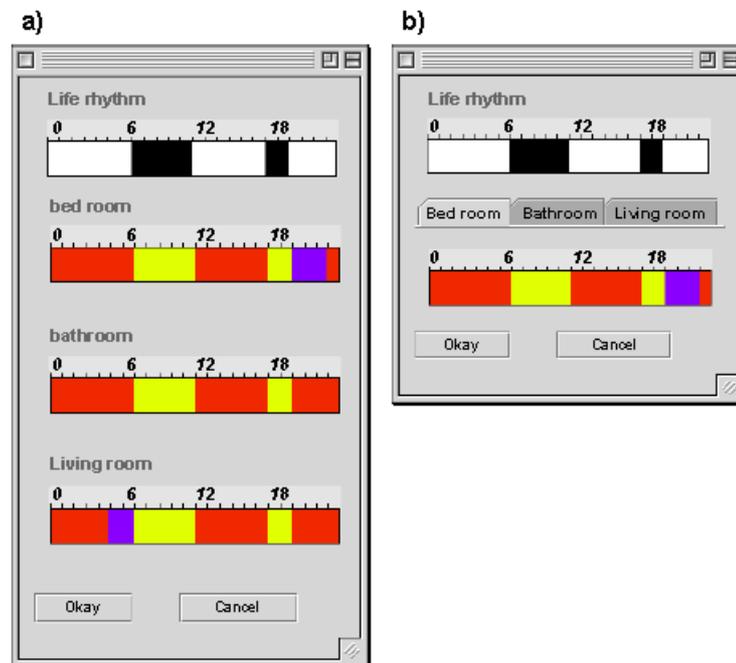


Figura 2.7: Control de calefacción para casas: a) la interfaz de usuario en la PC muestra al mismo tiempo la temperatura de todos los cuartos, mientras que b) la interfaz de usuario en la PDA muestra por separado la temperatura de cada cuarto

Sin embargo, la interfaz de usuario accesible desde un teléfono celular (cf. Figura 2.8) ha sido remodelada significativamente, con el fin de adaptarla al reducido espacio de la pantalla. Suponga que un usuario necesita modificar la temperatura de su sala mediante un teléfono celular. Cuando el usuario ingresa al sistema, observa una lista de opciones en el siguiente orden (cf. Figura 2.8.a): **recámara** (*chambre*), **sala** (*salon*) y **baño** (*salle D'eau*). Entonces, el usuario selecciona la opción **sala**. En respuesta a esta selección, el sistema despliega los siguientes componentes gráficos (cf. Figura 2.8.b):

- el título **Calefacción/sala** (*Chauffage/salon*), el cual confirma al usuario su elección;
- la opción **Ajustar** (*Donner ordre*), que permite al usuario modificar la temperatura del cuarto;
- la temperatura (*Temperature*) actual de la habitación (18 °C),
- la opción **Calentador del salón** (*Convecteurssalon*), la cual permite controlar (aumentar o disminuir) el termostato del calentador. En este caso la opción elegida es **media** (*moyen*).

En base a esta lista de opciones el usuario selecciona la opción **Ajustar** (*Donner ordre*). En consecuencia el sistema presenta una vista que contiene los siguientes componentes

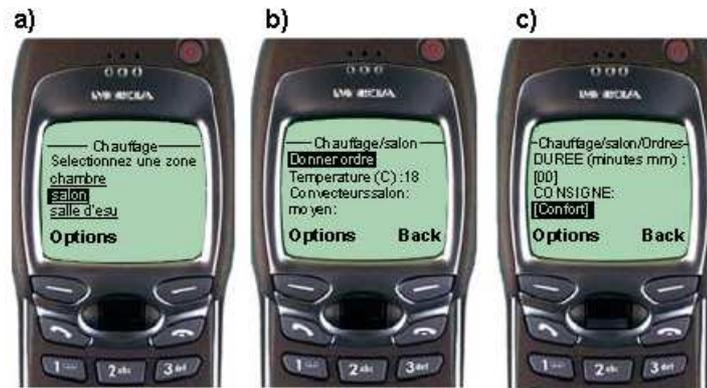


Figura 2.8: Control de calefacción para casas desde un teléfono celular: a) el usuario selecciona la opción *sala*, b) el sistema indica la temperatura actual de dicho cuarto y c) el usuario modifica la temperatura

gráficos (cf. Figura 2.8.c): el título Calefacción/salón/Ajustar (*Chauff/salon/Ordres*), la opción Duración (*Duree*) en minutos y la opción Parámetro de ajuste (*Consigne*). De esta forma, el usuario puede cambiar la temperatura del cuarto elegido.

2.3.3 Teclado numérico virtual

El teclado numérico virtual [Calvary et al., 2002, Grolaux et al., 2002] puede sustituir el teclado físico, cuando este último no está disponible. La entrada de números al sistema puede cambiar automáticamente del teclado físico al teclado virtual y viceversa, aún durante la ejecución del sistema.

La interfaz de usuario del teclado virtual (cf. Figura 2.9.a) está compuesta por los siguientes componentes gráficos: una etiqueta para mostrar el número seleccionado y trece botones que representan los números del 0 al 9, así como las opciones borrar (*Del*), seleccionar teclado (*Keyboard*) y actualizar (*Update*). Esta interfaz de usuario muestra los números del 0 al 9 con la finalidad de formar la cantidad numérica deseada. Por otra parte la interfaz de usuario de la Figura 2.9.b funciona cuando está disponible el teclado físico. Esta interfaz tiene los siguientes componentes gráficos: dos íconos (un triángulo hacia arriba y otro hacia abajo) para seleccionar el número requerido y dos botones para seleccionar el teclado y actualizar.

2.3.4 Reloj y calendario digital (*FlexClock*)

El sistema *FlexClock* está inspirado en el reloj del sistema UNIX, el cual muestra la fecha y la hora. A diferencia del reloj de UNIX, *FlexClock* es capaz de ajustar sus componentes gráficos (fecha y hora) al tamaño de la ventana [Grolaux et al., 2002]. Para llevar a cabo la adaptación, *FlexClock* provee diferentes versiones de sus componentes gráficos, e.g., la hora se muestra mediante un reloj análogo o digital, mientras que la fecha se presenta

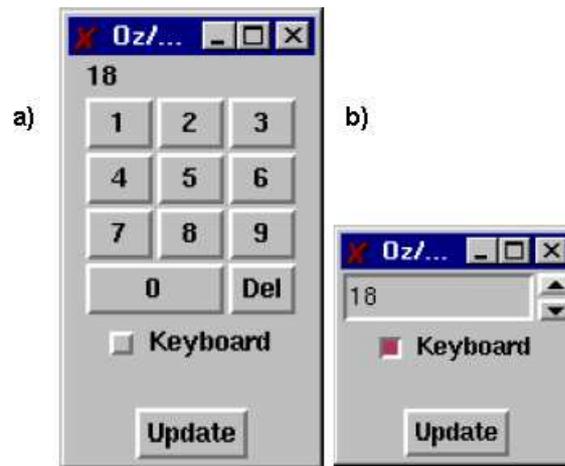


Figura 2.9: Interfaz de usuario del teclado numérico virtual cuando el teclado físico: a) no está disponible y b) está disponible

por medio de un calendario del mes actual o de un formato abreviado (día, mes y año) numérico o alfanumérico. Las diferentes versiones de los componentes dan lugar a diecisiete interfaces de usuario, las más representativas se muestran en la Figura 2.10.

La interfaz de usuario de la Figura 2.10.a puede ser adecuada para una pantalla de PC, puesto que ocupa mayor espacio que las demás. Esta versión de *FlexClock* despliega la hora usando al mismo tiempo un reloj análogo y un reloj digital; en el caso de la fecha, muestra el calendario y el formato abreviado alfanumérico. Debido al tamaño del área de despliegue, varias versiones de los componente gráficos están disponibles al mismo tiempo.

La interfaz de usuario de la Figura 2.10.b puede ser desplegada en la pantalla de un PSP (*Play Station Portable*) por su forma rectangular. Gracias al ancho del área de despliegue, el reloj análogo aún es visible junto con el reloj digital, en tanto que la fecha es presentada mediante el formato abreviado alfanumérico.

Por su forma cuadrangular, la interfaz de usuario de la Figura 2.10.c puede ser adecuada para una PDA. Esta interfaz de usuario muestra los mismos componentes que la interfaz de usuario de la Figura 2.10.b, sólo que su posición es diferente, i.e., en la parte superior se ubica el reloj análogo, en medio se localiza el reloj digital y por último, en la parte inferior, se encuentra la fecha en formato abreviado alfanumérico.

La interfaz de usuario de la Figura 2.10.d es muy pequeña, así que puede ser desplegada en un teléfono celular. Esta interfaz de usuario utiliza componentes muy compactos, como el reloj digital y la fecha en formato abreviado numérico.

Por último la interfaz de usuario de la Figura 2.10.e sólo muestra el reloj digital, así que puede ser utilizada en un termómetro digital.

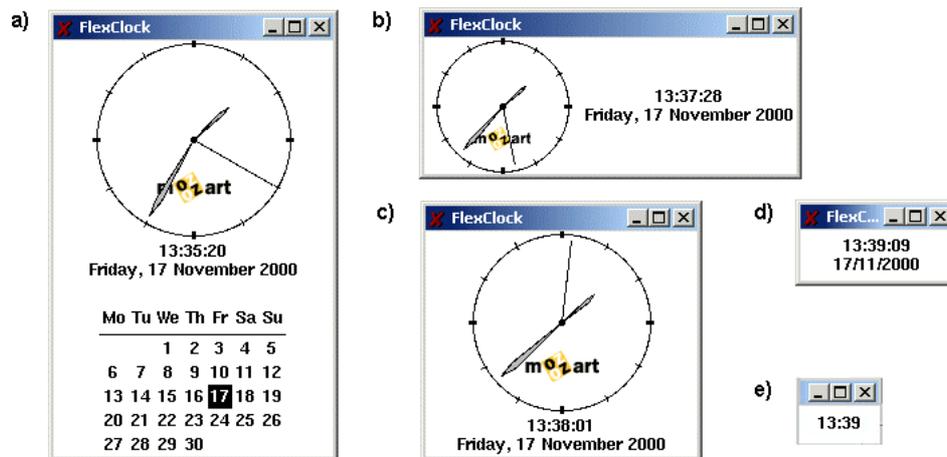


Figura 2.10: Interfaces de usuario de *FlexClock* para: a) una PC, b) un PSP, c) una PDA, d) un teléfono celular y e) un termómetro digital

La selección de la interfaz de usuario se lleva a cabo cuando cambia el tamaño del área de despliegue. En este caso, se selecciona la interfaz de usuario que mejor se adapte a las nuevas dimensiones de la ventana. Para llevar a cabo esta selección, *FlexClock* propone varias interfaces de usuario, en donde cada una requiere de un ancho y un alto mínimo en píxeles para poder ser desplegada. La regla que determina la mejor opción de interfaz de usuario consiste en obtener el mínimo de:

$$(wh - mh_i)^2 + (ww - mw_i)^2$$

en donde wh es el alto y ww es el ancho del área de despliegue, mientras que mh_i es el alto y mw_i es el ancho mínimo de la interfaz de usuario i .

Cada interfaz de usuario define un tamaño mínimo, pero no un máximo para poder adaptarse. En consecuencia, los componentes gráficos se autoconfiguran de acuerdo al gestor de geometría utilizado, e.g., el reloj digital puede centrarse en el espacio destinado a él, mientras que el reloj analógico puede desplegarse lo más grande posible.

2.3.5 Mesas de trabajo (*ConnecTables*)

ConnecTables es un dispositivo móvil de información consciente de contexto que se conecta a la red inalámbrica [Tandler et al., 2001]. El ámbito de investigación de este dispositivo se enfoca en el conjunto de situaciones generadas por el trabajo co-localizado, cuyo objetivo principal es facilitar la interacción entre las personas y el uso compartido de la información.

Por medio de este dispositivo, los usuarios pueden trabajar individual o cooperativamente, así como llevar a cabo la transición entre estas dos formas de trabajo. Este dispositivo también les permite agrandar dinámicamente el área de interacción mediante el acoplamiento de dos *ConnecTables*. De esta manera, se supera la restricción del tamaño de dicho dispositivo. La unión de dos *ConnecTables* permite a los usuarios trabajar en

paralelo, así como intercambiar información arrastrando los objetos de una pantalla a otra.

El mecanismo utilizado para la transición entre el trabajo cooperativo e individual es gradual, fluido e intuitivo, ya que está basado en movimientos explícitos en el espacio real que tienen efectos en el espacio virtual, e.g., el acoplamiento de dos pantallas individuales permite formar una pantalla compartida más larga.

El dispositivo tiene una pantalla táctil, un lápiz y una base que puede ajustarse a la altura deseada e inclinarse en diferentes ángulos. Las interacciones permitidas mediante *ConnecTables* son:

- **explícita**, la cual utiliza un lápiz,
- **implícita**, la cual emplea sensores que interpretan acciones físicas del mundo real.

Suponga que Juan y Abel quieren trabajar juntos, cada quien mediante su *ConnecTable*, así que unen sus dispositivos como se observa en el Figura 2.11. En respuesta a esta acción, el sistema crea un nuevo espacio de trabajo compartido que abarca el contenido que tenía cada dispositivo, antes de realizar el acoplamiento. Una vez que el sistema realiza la fusión, Juan transfiere una copia de su imagen hacia la pantalla de Abel, con el simple hecho de arrastrar dicha imagen a la otra pantalla (cf. Figura 2.12.a). Una vez que Abel tiene la imagen, la señala encerrándola en un semicírculo (cf. Figura 2.12.b), el cual aparece en ambas pantallas. Al terminar el trabajo en equipo, Juan y Abel separan sus dispositivos, así que el sistema desintegra el espacio de trabajo compartido.



Figura 2.11: Juan y Abel unen sus *ConnecTables* para trabajar colaborativamente

2.3.6 Entorno de arquitectura real colaborativa (*Roomware*)

El sistema *Roomware*, desarrollado por la institución GMD (Centro Nacional de Investigación Alemán para la Tecnología de la Información), explora principalmente los problemas relacionados con la “realidad aumentada” [Prante et al., 2004]. La motivación principal de este tema de investigación es añadir un carácter sintético-virtual a los objetos reales, con el fin de enriquecer con información relevante el entorno de un individuo o de un grupo. A diferencia de la “realidad virtual”, que aísla al individuo del mundo real,

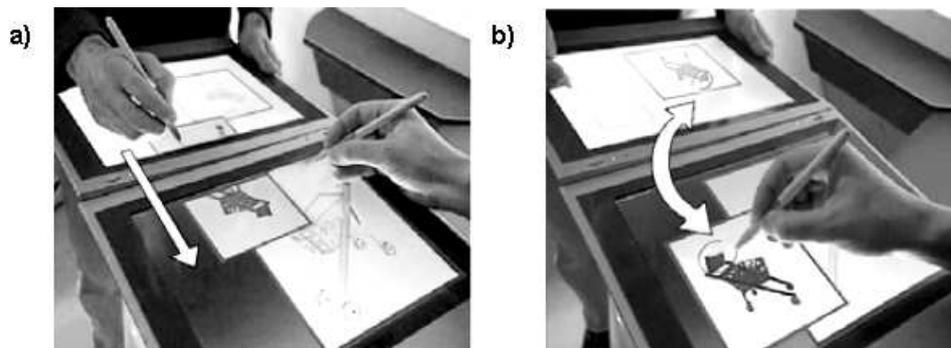


Figura 2.12: a) Juan transfiere a Abel una copia de la imagen, arrastrándola hacia la otra pantalla y posteriormente b) Abel señala una parte de la imagen, encerrándola con un semicírculo

la realidad aumentada incorpora capacidades de cómputo (información y tecnologías de comunicación) a las paredes y mesas de una sala de reunión, con el fin de facilitar nuevas formas de interacción entre los miembros de un grupo. De esta manera, se puede llevar a cabo la colaboración entre personas utilizando objetos tradicionales, e.g., en lugar de pegar hojas grandes en la pared o colocarlas sobre una mesa para realizar una lluvia de ideas, se emplean pantallas sobre las cuales los colaboradores pueden desplegar, almacenar y manipular la información capturada.

Roomware soporta el trabajo en grupo, cuyos miembros se encuentran reunidos en el mismo cuarto. Los componentes de *Roomware* son (cf. Figura 2.13):

- *DynaWall*: es la unión de tres pantallas táctiles idénticas para formar una sola pantalla;
- *InteracTable*: es una pantalla de plasma táctil colocada en una mesa,
- *ConnecTable*: es un dispositivo de información consciente de contexto (cf. sección 2.3.5);
- *CommChair*: es una *ConnecTable* integrada a un sillón.

Para lograr la interacción entre los diversos componentes, los desarrolladores de *Roomware* crearon una plantilla (*framework*) colaborativa síncrona llamada Beach (*Basic Environment for Active Collaboration with Hypermedia*).

Suponga que cinco colaboradores están trabajando en *Roomware* (cf. Figura 2.13). Una pareja (localizada en la parte izquierda del cuarto) está interactuando cara a cara mediante *ConnecTables*. Otra pareja de colaboradores (localizada al fondo del cuarto) está interactuando mediante dispositivos heterogéneos, uno con *DynaWall* y el otro con *CommChair*. Por último, el colaborador restante está trabajando por medio de *InteracTable*. En este caso, todos los colaboradores pueden ver lo mismo, pero a diferentes escalas. Además cada uno puede hacer modificaciones en el espacio compartido, las cuales



Figura 2.13: Componentes de *Roomware*: *ConnectTables*, *DynaWall*, *CommChair* e *InteracTable*

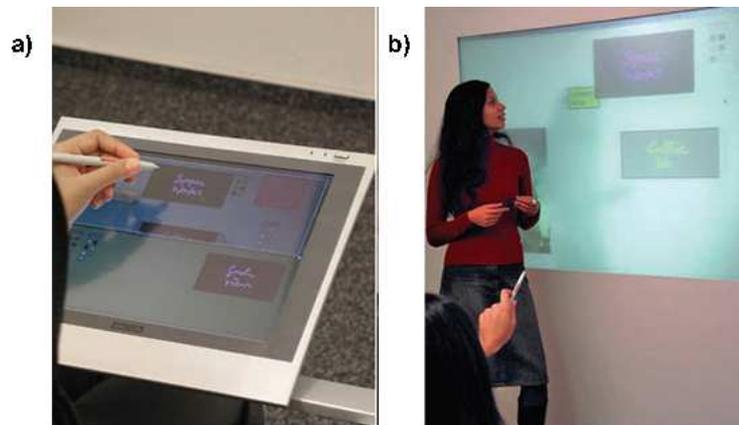


Figura 2.14: Interacción por medio de *Roomware*: a) Diana trabaja desde su *CommChair* y b) su colega Alicia usa *DynaWall*

también serán visualizadas por todos los colaboradores.

Ahora suponga que Alicia y Diana están colaborando. Alicia trabaja con *DynaWall* y Diana con *CommChair*. Diana puede ver toda la información desplegada en *DynaWall* desde su sillón. Cuando Diana ingresa una figura en *CommChair* (cf. Figura 2.14.a) Alicia puede verla en *DynaWall*, pero a una escala mayor (cf. Figura 2.14.b) .

2.4 Estudio comparativo

Después de haber presentado los conceptos básicos de plasticidad y de haber descrito cada uno de los sistemas relacionados con este trabajo de investigación, se presenta un estudio comparativo que se ilustra mediante tablas. Cada una de estas relaciona todos los sistemas sujetos a análisis con cada uno de los conceptos de plasticidad definidos en la primera parte de este capítulo.

Tabla 2.1: Parámetros del contexto de uso

Sistema interactivo	Usuario	Entorno	Plataforma	No especifica
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ConnecTables	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Roomware	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

En la tabla 2.1 se analizan los parámetros del contexto de uso contemplados por los sistemas bajo estudio. El contexto de uso se refiere a la adaptación de la interfaz de usuario a tres aspectos: a) el **usuario**, b) la **plataforma** y c) el **entorno** (sección 2.2). Sedan-Bouillon se adapta a dos aspectos del contexto de uso: a) al usuario, porque es capaz de reconocer cuando un usuario ingresa al sistema desde dos dispositivos diferentes y b) a la plataforma, ya que el sitio Web puede ser consultado desde una PDA y una PC. Por su parte, el sistema de calefacción únicamente se adapta a la plataforma porque el usuario puede consultar la temperatura de los cuartos desde una PC, una PDA o un teléfono celular. Otro sistema que sólo se adapta a la plataforma es *Roomware*, ya que puede ejecutarse en una serie de pizarrones interactivos (*DynaWall*), en una mesa (*InteracTable*) y en una silla (*CommChair*). Existen otras formas de adaptación a la plataforma como: a) el teclado numérico, que reconoce cuando el teclado físico está conectado a la PC y b) *ConnecTables*, que permite unir dos tabletas para formar una sola pantalla. En conclusión, la mayoría de los sistemas analizados se adapta a la plataforma, aunque Sedan-Bouillon también toma en cuenta al usuario. Sin embargo, ningún sistema considera el entorno.

En la tabla 2.2 se examina si el sistema es **mono-usuario** o **multi-usuario**. Sedan-Bouillon es mono-usuario porque aún cuando puede haber varios usuarios consultando el sitio, ninguno sabe que la información está siendo accedida por otros. El sistema de calefacción también es mono-usuario, puesto que un usuario no percibe si alguien más está consultando o modificando la temperatura de alguna habitación. Otros sistemas mono-usuario son el teclado numérico y *FlexClock*. En el caso de *ConnecTables*, se vuelve multi-usuario cuando los colaboradores unen sus pantallas. Por último, *Roomware* también es multi-usuario porque permite que los colaboradores interactúen por medio de algunos objetos del mundo real, e.g., mesas, sillas y muros. En conclusión, la mayoría de los sistemas estudiados se encuentran en el ámbito de los sistemas mono-usuario, sólo *ConnecTables* y *Roomware* son multi-usuario.

En la sección 2.2.1, se describieron dos percutores de la plasticidad, la remodelación y la

Tabla 2.2: Sistemas mono-usuario vs multi-usuario

Sistema interactivo	Mono-usuario	Multi-usuario
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ConnecTables	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Roomware	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Tabla 2.3: Remodelación

Sistema interactivo	Intra-modal	Inter-modal	Multimodal
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ConnecTables	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Roomware	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

redistribución. En la tabla 2.3, se analiza la remodelación, la cual puede ser **intra-modal**, **inter-modal** o **multimodal**. Todos los sistemas analizados implementan la remodelación de tipo intra-modal, i.e., preservan la modalidad de gráfico a gráfico. Ninguno ha explorado otras formas de representar las tareas del usuario o los conceptos del dominio.

En la tabla 2.4, se analiza la redistribución, la cual se divide en cuatro tipos: a) de **centralizada a centralizada**, b) de **centralizada a distribuida**, c) de **distribuida a centralizada** y d) de **distribuida a distribuida**. Sedan-Bouillon soporta todas las combinaciones posibles de redistribución: desde la duplicación de todos los componentes gráficos en una PC y en una PDA, hasta la repartición de algunos componentes entre ambos dispositivos y la omisión de los restantes. *Roomware* implementa la redistribución tipificada de **centralizada a distribuida** cuando su interfaz de usuario se reparte entre los tres pizarrones que conforman *DynaWall*. Asimismo, su interfaz de usuario puede redistribuirse en sentido inverso (**de distribuida a centralizada**) cuando se concentra en *InteracTable* o *CommChair*, además de permanecer centralizada

Tabla 2.4: Redistribución

Sistema interactivo	Centralizada a centralizada	Centralizada a distribuida	Distribuida a centralizada	Distribuida a distribuida
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Sistema de calefacción	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ConneCTables	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Roomware	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

cuando pasa de *InteracTable* a *CommChair* o viceversa. Todos los sistemas restantes aplican el tipo centralizado a centralizado.

Tabla 2.5: Tipo de adaptación plástica

Sistema interactivo	Adaptativa	Adaptable	Mixta
Sedan-Bouillon	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Sistema de calefacción	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ConneCTables	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Roomware	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Para analizar el proceso de adaptación plástica (cf. tabla 2.5) se respondieron las cinco preguntas siguientes por cada sistema:

1. ¿Quién detecta las variaciones en el contexto de uso?
2. ¿Quién identifica las posibles soluciones (interfaces de usuario)?

3. ¿Quién selecciona una solución (interfaz de usuario) particular?
4. ¿Quién realiza la transición de la solución actual hacia la nueva solución?
5. ¿Quién ordena la ejecución de la nueva solución mientras no ocurran nuevas variaciones en el contexto de uso?

Las posibles respuestas a cada una de las preguntas anteriores pueden ser: 1) el sistema, 2) el usuario o 3) ambos (cf. sección 2.2.1). Si la respuesta a estas preguntas es el sistema, entonces se dice que este es **adaptativo**. Cuando la respuesta concierne únicamente al usuario, significa que el sistema es **adaptable**. Finalmente, si la respuesta incluye tanto al usuario como al sistema, entonces se dice que el sistema es **mixto**.

En el caso de Sedan-Bouillon, se trata de un sistema mixto porque tanto el usuario como el sistema participan en el proceso de adaptación. Este sistema despliega una Meta-UI cuando identifica al mismo usuario en dos dispositivos diferentes, en tanto que el usuario decide cómo distribuir los componentes gráficos de la interfaz de usuario. El sistema de calefacción es adaptativo ya que se ejecuta automáticamente en el dispositivo en uso (e.g., PC, PDA o teléfono celular). El teclado numérico también es adaptativo porque despliega automáticamente la interfaz de usuario correspondiente, en función de la conexión o desconexión del teclado físico. *FlexClock* es adaptable ya que el usuario se encarga de redimensionar el área de despliegue y, a su vez, el sistema se ajusta al tamaño definido por el usuario. *ConnecTables* también es adaptativo puesto que crea automáticamente un espacio compartido cuando detecta la unión de dos pantallas. Por último, el sistema *Roomware* también es adaptativo porque se ajusta automáticamente al espacio ofrecido por el dispositivo de recepción (e.g., una mesa, una silla o una serie de pizarrones interactivos). En conclusión, la mayoría de los sistemas estudiados son adaptativos, sólo Sedan-Bouillon es mixto y *FlexClock* es adaptable.

Tabla 2.6: Grano de adaptación

Sistema interactivo	Interactor	Espacio de trabajo	IU completa	No especifica
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ConnecTables	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Roomware	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

El grano de adaptación (cf. tabla 2.6) señala el grado en que la interfaz de usuario ha sido metamorfoseada después de la adaptación. Como se mencionó en la sección

2.2.3, el grano de adaptación se clasifica en: a) **interactor**, b) **espacio de trabajo** e c) **interfaz de usuario completa**. Sedan-Bouillon se remodela a nivel de interactor porque compacta el menú y el título de la página Web cuando se ejecuta en una PDA. En cambio, el sistema de calefacción modifica su interfaz de usuario completa cuando se ejecuta en un teléfono celular. El teclado numérico se metamorfosea a nivel interactor, ya que compacta los botones numéricos cuando el teclado físico está conectado. Por último, *FlexClock* se transforma a nivel de espacio de trabajo porque la hora y la fecha se consideran como tareas específicas, cuya apariencia (escala, formato y posición) depende de las dimensiones del área de despliegue.

Tabla 2.7: Espacios tecnológicos

Sistema interactivo	Intra-espacios	Inter-espacios	Multi-espacios	No especifica
Sedan-Bouillon	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Sistema de calefacción	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ConnecTables	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Roomware	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Los espacios tecnológicos (cf. tabla 2.7) se refieren a la Ingeniería Dirigida por Modelos, que puede clasificarse en: a) **intra-espacio**, b) **inter-espacios** y c) **multi-espacios**. El sistema de calefacción y el teclado numérico fueron construidos en el *framework* ART-Studio [Calvary et al., 2001], el cual proporciona el mismo espacio tecnológico (intra-espacio) después de llevar a cabo la adaptación. *FlexClock* también emplea un solo espacio tecnológico porque fue implementado únicamente con la herramienta QtK. Los sistemas restantes no ofrecen información al respecto.

La implantación de la adaptación (cf. tabla 2.8) determina si el usuario puede seguir interactuando con el sistema durante la adaptación. Dos tipos de implantación de la adaptación han sido identificados: a) **estático** y b) **dinámico**. Sedan-Bouillon realiza la implantación dinámica porque permite al usuario seguir consultando la información turística de Sedan y Bouillon, mientras redistribuye su interfaz de usuario. El teclado numérico aplica el mismo tipo de implantación que Sedan-Bouillon, puesto que el usuario puede continuar introduciendo números, mientras el teclado físico está siendo conectado o desconectado. Asimismo *FlexClock* emplea la implantación dinámica, ya que cada segundo actualiza la hora y la fecha. *ConnecTables* también emplea la implantación dinámica porque los usuarios pueden seguir interactuando mientras unen sus mesas de trabajo. Por último, *Roomware* maneja igualmente la implantación dinámica, ya que los usuarios

Tabla 2.8: Implantación de la adaptación

Sistema interactivo	Estática	Dinámica	No específica
Sedan-Bouillon	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Teclado numérico	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ConnecTables	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Roomware	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

pueden continuar colaborando desde sus dispositivos durante la adaptación plástica.

Tabla 2.9: Grano de reanudación

Sistema interactivo	Acción física	Tarea	Sesión	No específica
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Teclado numérico	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
FlexClock	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ConnecTables	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Roomware	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

El grano de reanudación (cf. tabla 2.9) se refiere al grado de preservación del trabajo realizado por el usuario en el sistema antes de la adaptación plástica. Tres tipos de granos de reanudación son posibles: a) **acción física**, b) **tarea** y c) **sesión**. Sedan-Bouillon implementa la reanudación a nivel de acción física, ya que permite preservar la información que estaba siendo consultada antes de la redistribución de su interfaz de usuario. Al igual que Sedan-Bouillon, *FlexClock* permite la reanudación a nivel de acción física, gracias a un mecanismo que actualiza el área de despliegue cada segundo, impidiendo perder la hora y la fecha actual. Asimismo, *ConnecTables* implementa dicho tipo de reanudación porque permite a los usuarios conservar sus imágenes para compartirlas eventualmente con sus colegas. Finalmente, *Roomware* también soporta la reanudación a nivel de acción física, ya que mantiene el trabajo efectuado por los usuarios en sus respectivas mesas antes de haber sido conectadas entre sí.

Tabla 2.10: Producción de las interfaces de usuario

Sistema interactivo	Prefabricadas	Generadas en tiempo de ejecución
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ConnecTables	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Roomware	<input checked="" type="checkbox"/>	<input type="checkbox"/>

La producción de las interfaces de usuario (cf. tabla 2.10) permite conocer si al realizar la adaptación plástica se utilizan interfaces de usuario: a) **prefabricadas** o b) **generadas en tiempo de ejecución**. *FlexClock* genera sus interfaces de usuario en tiempo de ejecución porque aún cuando elige una interfaz de usuario de las diecisiete posibles, este sistema puede modificar el tamaño y la alineación de sus componentes (e.g., agrandar el reloj analógico y centrar el reloj digital). Todos los demás sistemas emplean interfaces prefabricadas e incluso pre-instaladas en dispositivos específicos.

Tabla 2.11: Usabilidad

Sistema interactivo	Usabilidad	No especifica
Sedan-Bouillon	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Sistema de calefacción	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Teclado numérico	<input checked="" type="checkbox"/>	<input type="checkbox"/>
FlexClock	<input checked="" type="checkbox"/>	<input type="checkbox"/>
ConnecTables	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Roomware	<input checked="" type="checkbox"/>	<input type="checkbox"/>

La usabilidad (cf. tabla 2.11) analiza si los sistemas siguen funcionando adecuadamente después de realizar la adaptación plástica. Sedan-Bouillon si conserva la usabilidad porque después de llevar a cabo la redistribución de su interfaz de usuario, el sistema permite continuar accediendo a todas las funcionalidades ofrecidas originalmente. El sistema de calefacción también conserva esta propiedad, puesto que los usuarios pueden consultar

y modificar la temperatura de los cuartos de una casa desde una PC, una PDA o un teléfono celular. El teclado numérico también mantiene la usabilidad, ya que permite capturar números antes y después de conectar el teclado físico. *FlexClock* de igual manera preserva esta propiedad porque, aún cuando el usuario haya cambiado las dimensiones del área de despliegue, el sistema continua proporcionando la fecha y la hora actual. *ConnecTables* también cuida la usabilidad ya que, una vez que los usuarios unieron sus mesas, el sistema permite que interactúen como si ambas mesas fueran una sola. Por último, *Roomware* igualmente conserva la usabilidad porque los usuarios pueden acceder a las mismas funcionalidades desde cualquiera de los dispositivos soportados (i.e., *DynaWall*, *InteracTable* y *CommChair*).

Capítulo 3

Análisis y diseño

La capacidad plástica del pizarrón compartido se introduce por medio de un escenario, el cual involucra a un grupo de colaboradores, que se encuentra realizando una lluvia de ideas (cf. sección 3.1). En el diseño del pizarrón compartido, se utilizó el estilo arquitectural MVC que permite partir de una arquitectura mono-usuario hacia una arquitectura multi-usuario redistribuida (cf. sección 3.2). La arquitectura del pizarrón compartido ha sido dividida en dos partes, una de ellas es el nodo principal y la otra es el conjunto de nodos hijos. El nodo principal se encarga de: 1) definir comandos de comunicación entre los dispositivos participantes, e.g., la redistribución de los componentes gráficos de la interfaz de usuario (cf. sección 3.3), 2) almacenar información relevante, e.g., las figuras realizadas en el área de dibujo. Por otra parte, los nodos hijos (cf. sección 3.4) se refieren a los interactores que conforman la interfaz de usuario del pizarrón compartido (i.e., el área de dibujo, la barra de herramientas, la meta-interfaz de redistribución y la barra de colaboradores), así como las funciones disponibles en cada interactor (i.e., las figuras que el usuario puede pintar sobre el área de trabajo). Por último, se detalla el diseño del pizarrón compartido mediante diagramas UML (*Unified Modeling Language*), los cuales destacan los siguientes aspectos: la comunicación entre los dispositivos participantes, el ingreso y la salida de un colaborador, la redistribución del pizarrón compartido y la actualización del área de dibujo.

3.1 Escenario

Mediante el siguiente escenario se pretende ilustrar la capacidad plástica del pizarrón compartido, propuesto en este trabajo de investigación (cf. Figura 3.1).

Suponga que un grupo de colaboradores realiza una lluvia de ideas utilizando el pizarrón compartido. Estos colaboradores pueden trabajar en su oficina o en cualquier lugar por medio de una PC o una PDA, las cuales tienen instalado el pizarrón compartido y acceso a la red inalámbrica.

Considere que Irma, Katty y Diego están trabajando cada uno en su oficina, mientras que Andrés se encuentra de viaje. Irma y Katty utilizan su computadora de escritorio, las

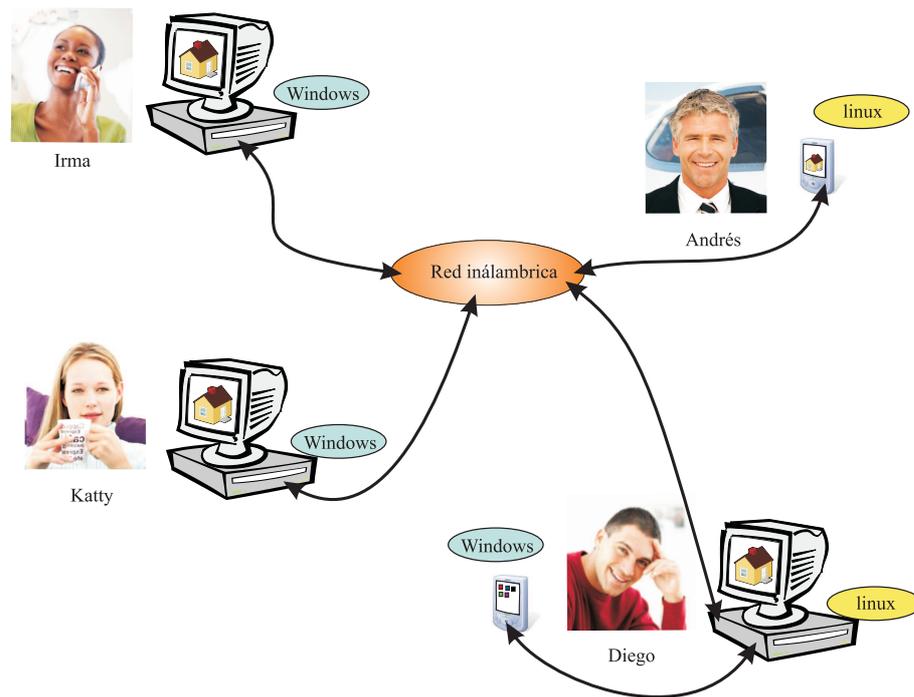


Figura 3.1: Escenario de trabajo cooperativo usando el pizarrón compartido plástico

cuales funcionan bajo el sistema operativo Windows. Diego emplea dos dispositivos, una PDA con Windows y una PC con sistema operativo Linux. Por último, Andrés utiliza únicamente una PDA con Linux.

Todos ellos pueden trabajar a distancia mediante el pizarrón compartido, el cual permite escribir texto no estructurado y hacer dibujos utilizando figuras pre-diseñadas (e.g., líneas, rectángulos y círculos) o creadas por el usuario mediante una secuencia de píxeles.

Cuando Irma ingresa al pizarrón compartido, previa validación de su nombre de usuario y contraseña, observa que se conservaron las últimas selecciones en la barra de herramientas (figuras, color y grosor) antes de finalizar la sesión precedente. Además, Irma puede recuperar la lluvia de ideas que realizó conjuntamente con sus colegas en dicha sesión. Ella también observa en la barra de colaboradores su nombre junto con su foto, lo cual indica que es la única persona presente en el pizarrón compartido.

Unos minutos más tarde, Katty ingresa al pizarrón compartido desde su computadora y nota que Irma ya se encuentra trabajando, puesto que el nombre y la foto de ambas aparece en la barra de colaboradores. Otro detalle que percibe Katty es la incorporación de nuevas propuestas en la lluvia de ideas. Por otro lado, Irma también percibe el ingreso de Katty al pizarrón compartido porque la barra de colaboradores le muestra su presencia.

Tiempo después Andrés ingresa al pizarrón compartido, lo cual implica que la barra de colaboradores será actualizada automáticamente en las instancias del sistema que se ejecutan en los dispositivos de Irma y Katty. Posteriormente Andrés empieza a unir al-

gunas ideas por medio de líneas, al mismo tiempo que Irma engloba cada una de las ideas principales mediante círculos y Katty encierra las ideas restantes con rectángulos. Los cambios realizados son observados inmediatamente por todos los colaboradores.

Por último Diego ingresa al sistema desde su PC, así que los demás colaboradores perciben su presencia por medio de la barra de colaboradores. Posteriormente, él decide usar al mismo tiempo su PC y su PDA, así que ingresa nuevamente al sistema desde la PDA. Cuando el pizarrón compartido detecta a Diego en dos dispositivos diferentes, automáticamente se despliega una meta-interfaz en la PC para permitirle redistribuir su interfaz de usuario en la PC, en particular la barra de herramientas y la de colaboradores. Diego decide colocar la barra de herramientas en la PDA y dejar la barra de colaboradores en la PC. Una vez que la interfaz de usuario ha sido redistribuida, Diego puede seleccionar la herramienta necesaria desde la PDA y pintar sobre el área de dibujo ubicada en la PC.

Cuando Andrés decide salir del pizarrón compartido, automáticamente se elimina su foto de la barra de colaboradores para notificar a los demás sobre su partida.

3.2 Arquitectura del pizarrón compartido

El diseño del pizarrón compartido se basa en el estilo arquitectural MVC. El motivo principal de esta elección reside en la facilidad de representar la estructura de un sistema antes y después de la redistribución de su interfaz de usuario. MVC también permite construir un sistema paso a paso, partiendo de una aplicación mono-usuario hasta obtener una aplicación cooperativa redistribuible. Además, MVC permite estructurar un sistema en objetos [Crease, 2001], haciendo posible la utilización de lenguajes de programación que siguen este paradigma, e.g., Java o SmallTalk.

La descripción del diseño del pizarrón compartido ha sido organizada siguiendo el orden en que Phillips [Phillips, 1999] analiza los conceptos de MVC.

3.2.1 Definición del grano de adaptación

El primer concepto se refiere a la estructura de MVC, la cual está formada por un modelo, una vista y un controlador, aunque algunos desarrolladores optan por unir la vista y el controlador [Phillips, 1999]. En el diseño del pizarrón compartido también se optó por unir la vista y el controlador, así que al aplicar el estilo arquitectural MVC al pizarrón compartido se tiene que la **vista** y el **controlador** conciernen a la interfaz de usuario, mientras que el **modelo** corresponde al núcleo funcional.

En esta etapa del diseño es importante especificar el grano de adaptación del sistema, con el fin de determinar qué tanto cambiará su interfaz de usuario de un dispositivo a otro, e.g., de una PC a una PDA. Como se mencionó en la sección 2.2.3, el grano de adaptación puede implementarse a nivel de: a) interactor, b) espacio de trabajo e c) in-

terfaz de usuario completa.

Un interactivo se define como una entidad capaz de mostrar (parte de) su estado, en algún medio de presentación [Duke and Harrison, 1993]. Aplicando este concepto al diseño del pizarrón compartido se contemplaron cinco interactivos: 1) el área de dibujo, 2) la barra de herramientas, 3) la meta-interfaz de redistribución, 4) la barra de colaboradores y 5) la ventana de autenticación. Estos interactivos se pueden compactar suficientemente para poder ser manipulados en una PDA.

Por otra parte, un espacio de trabajo es un conjunto de tareas lógicamente conectadas (cf. sección 2.2). En el diseño del pizarrón compartido, se hubiera podido contemplar este grano de adaptación considerando un espacio de trabajo que comprende tres tareas. La primera se refiere a las funciones básicas del pizarrón, e.g., el área de dibujo y las herramientas de formas y colores. La segunda tarea concierne la autenticación del usuario y la barra de colaboradores. Finalmente, la tercera tarea atañe la meta-interfaz de redistribución. Sin embargo, la incorporación de este grano de adaptación en el diseño del pizarrón compartido implica remodelar las tareas mencionadas, las cuales podrían no compactarse lo suficiente como para ser manipuladas en una PDA.

Desde nuestro punto de vista, el grano de adaptación a nivel de la interfaz de usuario completa no sería la mejor opción, ya que los usuarios experimentarían cambios bruscos debidos a la necesidad de adaptarse no simplemente al cambio en el tamaño de la pantalla (e.g., de una PC a una PDA), sino también a la metamorfosis de la interfaz de usuario. Otra desventaja adicional es que los dispositivos no usan los mismos periféricos de entrada, ya que uno de ellos utiliza un ratón y el otro emplea un lápiz. La metamorfosis de la interfaz de usuario completa añade carga cognitiva al usuario porque, además de conmutar entre dispositivos y periféricos, debe localizar los componentes gráficos equivalentes en ambos dispositivos.

El grano de adaptación más conveniente para el pizarrón compartido es el interactivo, puesto que permite cambiar el tamaño y/o la forma de un subconjunto de componentes gráficos. De esta manera, resulta más fácil para el usuario adaptarse al cambio del dispositivo y de la interfaz de usuario.

3.2.2 Estructura de una aplicación mono-usuario

El segundo concepto que define el estilo arquitectural MVC es la estructura de una aplicación mono-usuario, la cual se representa por medio de un árbol MVC [Crease, 2001].

Del lado derecho de la Figura 3.2 se muestra la interfaz de usuario del pizarrón compartido. Como se mencionó en la sección 3.2.1, esta se compone de cinco interactivos, los cuales se describen a continuación: 1) la **barra de colaboradores** informa sobre la presencia de los miembros del grupo, 2) el **área de dibujo** sirve para dibujar bosquejos y esquemas mediante figuras básicas (e.g., líneas y círculos), 3) la **barra de herramientas** contiene formas y colores, 4) la **meta-interfaz de la redistribución** se encarga de

mostrar los interactores que son susceptibles de redistribución y por último 5) la **ventana de autenticación** permite validar a los usuarios mediante un nombre de usuario y una contraseña.

Del lado izquierdo de la Figura 3.2 se encuentra el árbol MVC, el cual está compuesto de un nodo raíz o nodo principal (**R**) y cinco nodos hijos (**H1**, **H2**, **H3**, **H4** y **H5**). El nodo R se encarga de la creación del sistema completo, de la comunicación entre las instancias del sistema y del almacenamiento de información relevante (e.g., los diagramas realizados por los colaboradores y la última selección de herramientas en el entorno de cada colaborador). A su vez, el nodo R subdelega responsabilidades a sus nodos hijos:

- El nodo H1 se encarga de la barra de colaboradores. Su vista-controlador es responsable de mostrar la presencia de los colaboradores, en tanto que su modelo administra la información del ingreso y de la salida de los colaboradores. Si bien este nodo es irrelevante en una aplicación mono-usuario se ha decidido incluirlo en esta etapa del diseño.
- El nodo H2 es responsable del área de dibujo. Su vista-controlador tiene la función de desplegar en pantalla las figuras realizadas, mientras que su modelo obtiene las coordenadas provistas por el usuario y realiza los cálculos necesarios para pintar las figuras sobre el área de dibujo.
- El nodo H3 se encarga de la barra de herramientas. Su vista-controlador tiene la función de pintar los componentes gráficos en el momento necesario, e.g., después de que un usuario ha seleccionado un color, la vista-controlador debe denotar el color seleccionado como el color actual para permitir que el usuario perciba la nueva selección. Por otro lado, su modelo captura y responde a los eventos generados por la selección de sus componentes gráficos.
- El nodo H4 es responsable de la meta-interfaz de redistribución. Su vista-controlador se encarga de desplegar en pantalla los interactores susceptibles de redistribución, en tanto que su modelo pone en marcha la nueva interfaz de usuario.
- Por último, el nodo H5 es responsable de la autenticación del usuario. Su vista-controlador muestra en pantalla los componentes gráficos para introducir un nombre de usuario y una contraseña, mientras que su modelo realiza el proceso de verificación de la identidad del usuario.

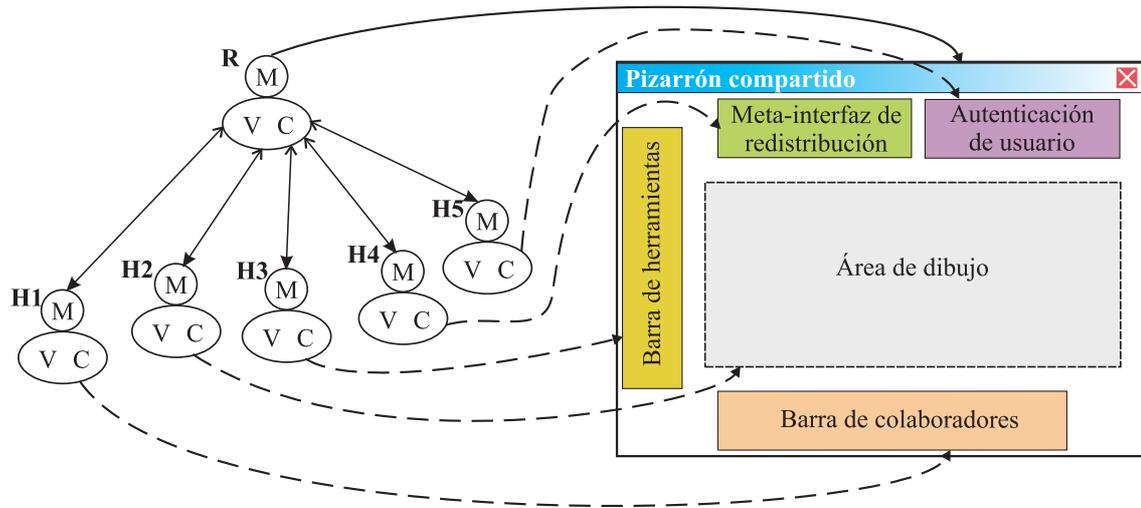


Figura 3.2: Asociación entre el árbol MVC y el pizarrón compartido

3.2.3 Estructura de una aplicación multi-usuario

El tercer concepto descrito por el estilo arquitectural MVC se refiere a la transformación de una estructura mono-usuario en una estructura multi-usuario. Como se mencionó previamente, este estilo arquitectural permite mostrar la estructura de una aplicación antes y después de la redistribución de su interfaz de usuario.

En la Figura 3.3 se muestra la arquitectura MVC multi-usuario del pizarrón compartido antes de la redistribución de su interfaz de usuario. Cada árbol MVC representa una instancia del pizarrón compartido la cual contiene los interactores de la interfaz de usuario. De esta manera, cada colaborador puede ejecutarlo localmente en su dispositivo, sin depender de un servidor. Cada cuadro punteado está asociado a una PC o una PDA, ya que el pizarrón compartido se puede ejecutar en alguno de estos dispositivos. La línea que une los árboles por medio de su nodo raíz, significa que las diferentes instancias del pizarrón compartido se pueden comunicar entre sí. De esta manera, los colaboradores pueden recibir notificación sobre eventos relevantes, e.g., el ingreso o la salida de un colega.

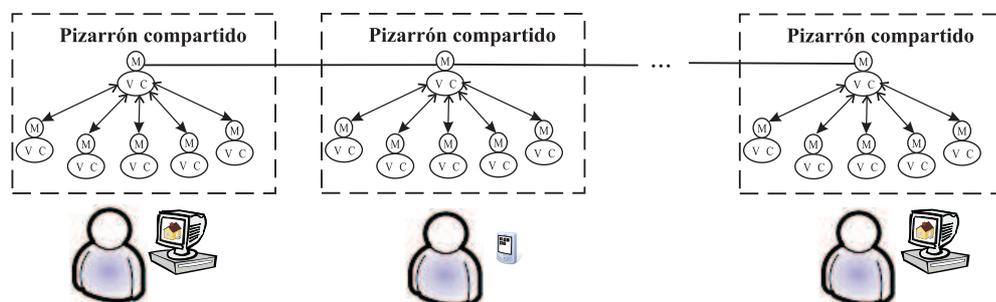


Figura 3.3: Arquitectura del pizarrón compartido antes de la redistribución

La Figura 3.4 muestra la arquitectura del pizarrón compartido después de la redis-

tribución de su interfaz de usuario. El recuadro de la izquierda representa al servidor y el de la derecha representa al cliente. El modelo está ubicado únicamente en el servidor, en tanto que el componente vista-controlador está duplicado tanto en el cliente como en el servidor. Esta arquitectura permite albergar una parte de la interfaz de usuario en la PC y la otra parte en la PDA. De esta manera, cada instancia del pizarrón compartido puede ser ejecutada en ambos dispositivos.

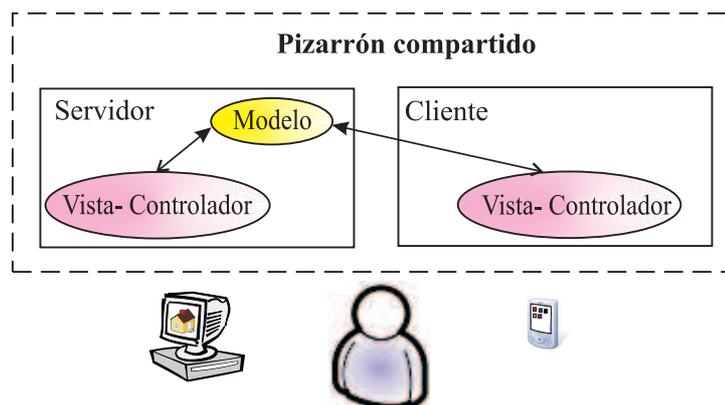


Figura 3.4: Arquitectura del pizarrón compartido después de la redistribución

Los componentes que intervienen en el proceso de redistribución de la interfaz de usuario (cf. Figura 3.5) son: la detección de cambios en el contexto, la meta-interfaz de redistribución y la evolución.

La **detección de cambios en el contexto de uso** se realiza de dos formas: 1) a nivel plataforma, ya que el pizarrón compartido permite ser ejecutado en una PC y una PDA, por medio de interfaces de usuario estáticas (pre-diseñadas) y 2) a nivel usuario, porque el pizarrón compartido permite reconocer cuando un colaborador ingresa al pizarrón compartido desde dos dispositivos diferentes. El proceso de adaptación al contexto de uso se hace solo a nivel usuario. Cuando el componente de detección de cambios observa variaciones en el contexto de uso, este componente informa a la meta-interfaz de redistribución sobre el cambio.

La **meta-interfaz de redistribución** tiene las posibles opciones de redistribución de la interfaz de usuario, las cuales son: 1) ubicar la barra de herramientas y la de colaboradores en la PDA y 2) ubicar la barra de colaboradores en la PC y la barra de herramientas en la PDA o viceversa. La meta-interfaz se comunica con la evolución por dos cuestiones: 1) proporcionar las opciones factibles para redistribuir e 2) informar a cada vista-controlador (una en la PC y otra en la PDA) la nueva posición de los interactores.

La **evolución** se encarga de establecer la comunicación entre el componente de meta-interfaz y la presentación para: 1) informar a la presentación las posibles opciones de redistribución de la interfaz de usuario, 2) avisar a la meta-interfaz la selección indicada por

el usuario para la redistribución de los interactores e 3) informar a cada vista-controlador los interactores a desplegar.

Por su parte la presentación despliega en pantalla las opciones de redistribución de la interfaz de usuario, asimismo informa a la evolución la opción elegida por el colaborador.

Como se observa en la Figura 3.5 la tripleta \ll detección de cambios en el contexto de uso, meta-interfaz de redistribución, evolución \gg es independiente para cada instancia del pizarrón compartido. La independencia de la tripleta permite que el proceso de redistribución de la interfaz de usuario sea posible desde una PDA y una PC.

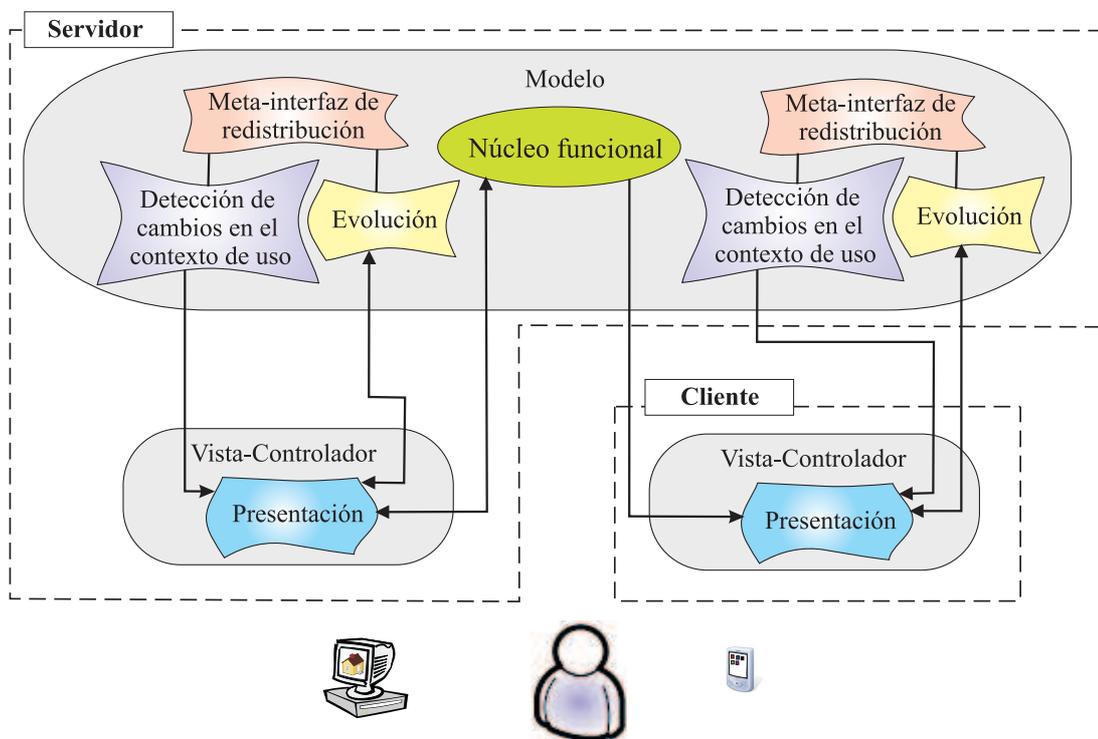


Figura 3.5: Componentes del proceso de adaptación

3.3 Nodo principal del pizarrón compartido

El nodo raíz del árbol MVC realiza tres tareas principales del pizarrón compartido. La primera tarea consiste en almacenar la información necesaria, e.g., los últimos valores seleccionados en la barra de herramientas y las figuras realizadas por cada colaborador. La segunda tarea consiste en realizar la comunicación entre las instancias del pizarrón compartido. La última tarea estriba en ejecutar los procesos necesarios para la redistribución de la interfaz de usuario. Estas tareas se detallan a continuación.

3.3.1 Almacenamiento de información

La información que se requiere almacenar de manera permanente se refiere a: 1) las figuras creadas por los colaboradores, 2) la última selección de herramientas por cada usuario y 3) el registro de los colaboradores (nombre de usuario y contraseña). De manera temporal, se guardan los nombres de los colaboradores conectados.

Las figuras creadas por los colaboradores son almacenadas en cada instancia del pizarrón compartido, e.g., suponga que el colaborador de la izquierda dibuja un círculo (cf. Figura 3.3). Su instancia del pizarrón compartido, ubicada en su PC, se comunica con las demás instancias por medio del nodo raíz. Una vez que las instancias del pizarrón compartido han recibido la notificación sobre la creación de una nueva figura, su base de datos la almacena de manera local. Cuando se realiza una operación de modificación sobre alguna figura, se lleva a cabo un proceso similar al que se acaba de describir.

Por cada usuario se almacena la última selección de las herramientas de colores y de figuras. Antes de que un colaborador abandone su sesión, el pizarrón compartido guarda su configuración final de dos formas posibles. La primera consiste en almacenar esta información de manera local en el dispositivo en uso, ya que el colaborador puede volver a emplear el mismo dispositivo. De esta forma, el sistema establecerá más rápido la configuración del colaborador, sin necesidad de recurrir a un servidor. La segunda opción consiste en almacenar la información en el servidor en caso de que el colaborador cambie de dispositivo.

La última información que se almacena en la base de datos es el registro de cada colaborador. Esta información consta de un nombre de usuario y de una contraseña. Al igual que las figuras, estos datos se almacenan de manera local en el dispositivo en uso y de manera remota en el servidor.

Los nombres de los colaboradores en línea se almacenan temporalmente en un registro. Cada vez que un usuario ingresa al pizarrón compartido, la instancia ubicada en su dispositivo avisa a las demás instancias, las cuales almacenan el nombre del colaborador de manera temporal en su dispositivo. Antes de que un colaborador salga de la sesión, su instancia del pizarrón compartido informa a las otras instancias, las cuales eliminan el nombre del colaborador de su registro.

3.3.2 Módulo de comunicación

El sistema cuenta con algunos comandos para establecer la comunicación entre las instancias del pizarrón compartido. Los comandos se dividen en categorías:

- **comandos de figuras:** permiten informar eventos relacionados con las figuras, i.e., creación, modificación y solicitud;
- **comandos de colaboradores:** permiten notificar eventos relacionados con los colaboradores, i.e., incorporar, dejar sesión y solicitar colaboradores;

- **comandos de redistribución:** notifica la di de la barra de herramientas o la barra de colaboradores.

Comandos de figuras

Cuando se crea una figura se envía el comando **crear figura** (cf. Figura 3.6.a), el cual incluye: un identificador de figura (**F**), el cual indica la realización de una operación sobre una figura, un identificador del comando de creación (**N**), el tipo de figura (**T**), el color (**C**), el grosor (**G**) y las coordenadas (**P₀** y **P₁**). Cuando el comando de creación de figura llega a una instancia del pizarrón compartido, diferente de la que emitió el comando, dicha instancia almacena y pinta la figura en base a las propiedades indicadas.

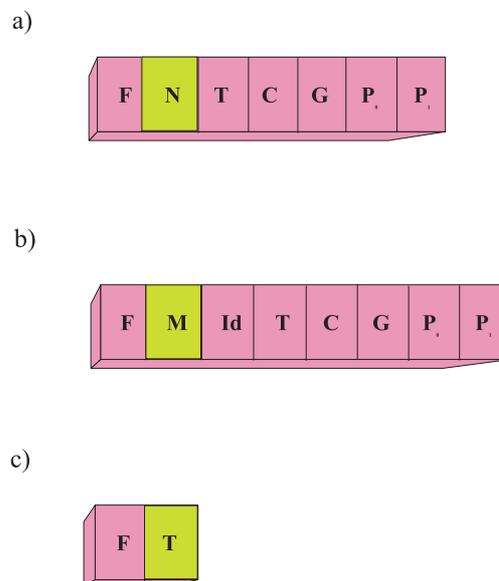


Figura 3.6: Comandos de figuras: a) creación, b) modificación y c) solicitud

Una vez que la figura se ve reflejada en el área de dibujo, los colaboradores pueden modificar su tamaño, grosor o color. Cuando un colaborador modifica alguna de las propiedades de una figura, su instancia del pizarrón compartido avisa inmediatamente a las demás instancias del pizarrón sobre las nuevas modificaciones, mediante el comando **modificar figura** (cf. Figura 3.6.b). Este comando contiene los siguientes parámetros: un identificador de figura (**F**), un identificador del comando de modificación (**M**), un número asociado a la figura (**Id**), el tipo de figura (**T**), el color (**C**), el grosor (**G**) y las coordenadas (**P₀** y **P₁**).

Cuando un colaborador ingresa al pizarrón compartido, su instancia del pizarrón solicita las figuras actuales por medio del comando **solicitar figura**. En respuesta, otra instancia del pizarrón le envíe las figuras existentes. Este comando contiene: un identificador de figura (**F**) y un identificador del comando de solicitud (**T**).

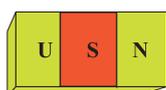
Comandos de colaboradores

El comando **incorporar colaborador** (cf. Figura 3.7.a), sirve para informar a los demás colaboradores conectados sobre el ingreso de un colega a la sesión en curso. El comando contiene: un identificador de colaboradores (**U**), el cual indica la realización de una operación sobre la información referente a un colaborador, un indicador del comando de incorporación (**I**) y un nombre asociado al colaborador (**N**). Cuando las instancias del pizarrón compartido reciben este comando, cada una debe actualizar su registro de colaboradores conectados y mostrar el cambio en la barra de colaboradores.

a)



b)



c)



Figura 3.7: Comandos de colaboradores: a) entrada a una sesión, b) salida de una sesión y c) solicitud de información sobre los colaboradores

Un comando más es **abandonar sesión** (cf. Figura 3.7.b) para indicar el nombre del colaborador que sale de la sesión en curso. El comando se compone de: un identificador de colaboradores (**U**), un identificador del comando de salida (**S**) y un nombre asociado al colaborador (**N**). Por ejemplo cuando un colaborador decide salir de la sesión en curso, la instancia del pizarrón compartido envía inmediatamente el comando de abandonar sesión para que las demás instancias del pizarrón actualicen su registro de usuarios conectados.

El comando **solicitar colaboradores** (cf. Figura 3.7.c) permite informar a un colaborador retardatario sobre los colaboradores presentes en la sesión en curso. El comando contiene: un identificador de colaboradores (**U**) y un identificador del comando de solicitud (**A**). Este comando funciona cuando un colaborador retardatario ingresa a la sesión en curso, ya que su instancia del pizarrón compartido envía automáticamente este comando a otra instancia del pizarrón, con el fin de obtener los nombres de los colaboradores presentes en dicha sesión.

Comandos de redistribución

El comando **redistribuir herramientas** (cf. Figura 3.8.a) es enviado automáticamente a la adecuada instancia del pizarrón compartido cada vez que el colaborador selecciona desde la ventana de redistribución la opción correspondiente. Este comando indica a la instancia del pizarrón elegido, mostrar la barra de herramientas actualizada. El comando contiene: un identificador de redistribución (**R**), el cual indica una operación para dividir la interfaz de usuario, un identificador del comando de redistribuir herramientas (**H**), la última opción seleccionada respecto a la figuras (**F**), al color (**C**) y al grosor (**G**).

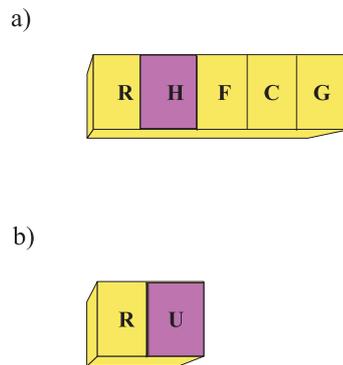


Figura 3.8: Comando usado para: a) redistribuir la barra de herramientas y b) la redistribuir colaboradores

El comando **redistribuir usuarios** (cf. Figura 3.8.b) también es enviado automáticamente a la adecuada instancia del pizarrón compartido cuando el usuario selecciona la opción correspondiente. Este comando se componen de: un identificador de redistribución (**R**) y un identificador del comando de redistribuir colaborador (**U**).

3.3.3 Redistribución de la interfaz de usuario

Como se observa en la Figura 3.5 el encargado de redistribuir la interfaz de usuario del pizarrón compartido es un servidor, el cual contiene el modelo y la vista-controlador. A su vez dentro del modelo se encuentra el componente de detección de cambios en el contexto de uso, la meta-interfaz de redistribución y el componente de evolución, el cual está encargado de realizar la evolución del sistema.

Detección de cambios en el contexto de uso

En el capítulo 2 se mencionó que el contexto de uso se define en términos del usuario, de la plataforma y del entorno. Los trabajos propuestos difieren en la cobertura del contexto de uso. Como se observó en el estudio comparativo presentado en la sección 2.4, ningún sistema multi-usuario se adapta al usuario, a excepción de Sedan-Bouillon, el cual es mono-usuario. El pizarrón compartido aborda la adaptación:

- **al usuario:** porque el pizarrón compartido puede detectar que el mismo usuario está accediendo al sistema desde dos dispositivos, asimismo ofrece la opción de redistribuir su interfaz de usuario. Además el pizarrón compartido establece en su sesión la última configuración realizada por el usuario (la selección de la figura, color y grosor), también asocia cada figura a su autor.
- **a la plataforma:** porque el pizarrón compartido puede accederse desde una PC y una PDA con diferentes sistemas operativos (Windows, Windows Mobil y Linux).

Observación y/o control de la meta-interfaz de redistribución

La meta-interfaz de redistribución representa el grado de control acordado al usuario en el proceso de adaptación plástica. Según el criterio “Guía/subcriterio: Regreso de Información”, la observación es el grado mínimo [Bastien and Scapin, 1993]. Más allá de la mera observación, puede llevarse a cabo una negociación entre el usuario y el sistema. El grado de control que tiene el usuario o el sistema sobre el proceso de adaptación plástica es un tema abierto, cuya reflexión está ligada a los criterios de control explícito y de adaptabilidad [Bastien and Scapin, 1993] que ameritan ser re-examinados bajo este ángulo.

Contrariamente a las demás funciones de la adaptación (detección de cambios en el contexto de uso y evolución) que ya han sido el objeto de algunos trabajos sobre la plasticidad de interfaces de usuario, el terreno es casi virgen en materia de meta-interfaz, ya que no existe ninguna teoría, ningún modelo sólo algunas ideas de metáforas.

Por ejemplo, la metáfora de las tijeras y la metáfora de las rasgadas propuestas por Calvary et al. [Calvary et al., 2006] permiten expresar los caracteres recortable y rompible de una interfaz de usuario: en el caso de una interfaz “recortable”, el usuario puede cortarla para eventualmente distribuirla; en el caso de una interfaz “rompible” el usuario corre riesgos, ya que puede incurrir en una pérdida de usabilidad. En Calvary et al. [Calvary et al., 2006] también se proponen la metáfora de los imanes para expresar la compatibilidad/incompatibilidad de una interfaz de usuario y la metáfora de los rompecabezas para expresar la complementariedad. A este respecto, el presente trabajo de tesis examina la semi-plasticidad mixta porque el pizarrón compartido propuesto: 1) se adapta a la plataforma (i.e., el sistema puede ejecutarse en una PC y en una PDA) y 2) permite que el usuario y el sistema realicen conjuntamente la adaptación plástica.

La meta-interfaz de redistribución se basa en el proceso de adaptación plástica (cf. capítulo 2) propuesta por Calvary et al. [Calvary et al., 2001]. En la figura 3.9 se muestra el diagrama de flujo del correspondiente.

En el caso del pizarrón compartido, el proceso de adaptación plástica se ejecuta debido al cambio en el contexto de uso a nivel usuario. Este se lleva a cabo con los siguientes pasos:

1. **Identificar soluciones candidatas:** se muestra al usuario la meta-interfaz de la redistribución, cuyas opciones se basan en un conjunto precalculado de interfaces

de usuario;

2. **Seleccionar una solución particular:** a partir de las opciones propuestas por la meta-interfaz de redistribución, el usuario selecciona aquella que satisfaga sus requerimientos;
3. **Transformar los estados:** se encarga de transferir los estados de las variables necesarias, i.e., selección en curso del color, del grosor y de la figura, hacia el nuevo dispositivo de interacción (e.g., una PDA o una PC);
4. **Establecer selección actual:** cuando se redistribuye la interfaz de usuario (entre una PC y una PDA), la instancia del pizarrón compartido en la PC funge como servidor (cf. Figura 3.4), el cual se comunica con la instancia del pizarrón en la PDA para indicarle que interactores debe desplegar. De esta forma la nueva selección de la interfaz de usuario se convierte en la selección actual;
5. **Ejecutar la nueva solución:** la instancia del pizarrón compartido en la PC se encarga de ejecutar la nueva interfaz de usuario, preservando el estado del sistema, i.e., se recuperan datos importantes (dibujos existentes, últimas selecciones en la barra de herramientas y colaboradores presentes) para que todos los datos queden reestablecidos al ejecutar la nueva interfaz de usuario.

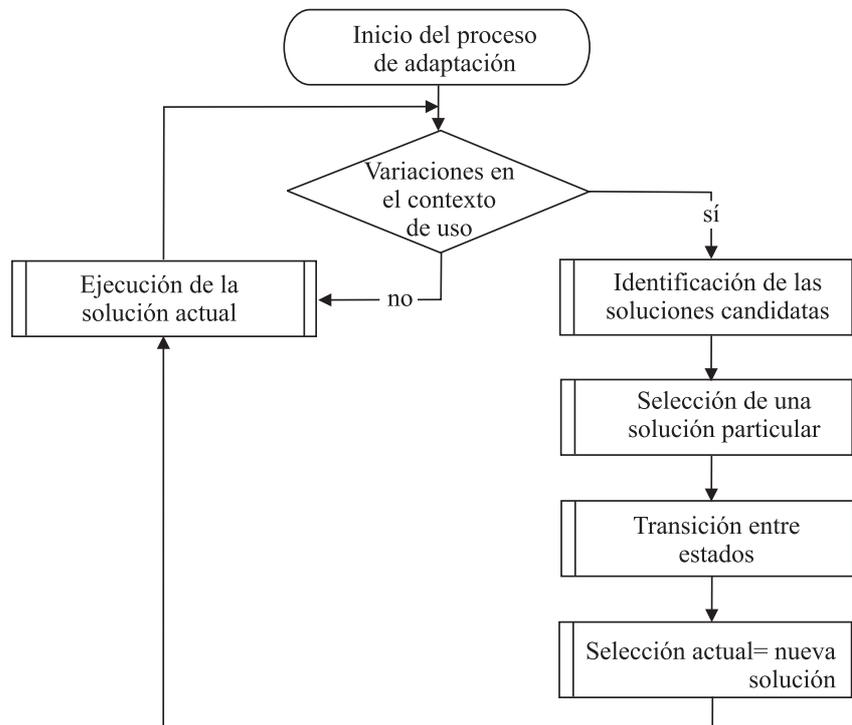


Figura 3.9: Proceso de adaptación plástica

Después de ejecutar la nueva interfaz de usuario, el pizarrón compartido puede detectar eventualmente un nuevo cambio en el contexto de uso.

Cálculo de la evolución del sistema

La evolución se encarga de calcular e implementar la reacción. El sistema propuesto aborda los dos percutores de la plasticidad: 1) la remodelación, que actúa sobre la presentación de los conceptos del dominio y de las tareas del usuario (cf. sección 2.2.2) y 2) la redistribución, que se refiere a la dispersión de la interfaz de usuario en diferentes plataformas.

En el capítulo 2 se mencionó que existen cuatro tipos de redistribución, las cuales son: 1) de centralizada a centralizada, 2) de centralizada a distribuida, 3) de distribuida a centralizada y 4) de distribuida a distribuida. El pizarrón compartido soporta dos tipos de redistribución, las cuales son: 1) de centralizada a centralizada, ya que todos los interactores pueden migrar de una PC a una PDA y 2) de centralizada a distribuida, debido a que algunos interactores en la PDA y los restantes pueden migrar a la PC.

3.4 Interactores del pizarrón compartido

A continuación se detallan las funcionalidades provistas por los cinco interactores mencionados en la sección 3.2:

- Núcleo funcional:
 - la **barra de herramientas** contiene las formas, colores y grosores de línea,
 - el **área de dibujo** muestra las figuras realizadas por el colaborador
- Conciencia de grupo por medio de:
 - el **área de dibujo**, porque permite visualizar el trabajo realizado por todos los colaboradores;
 - la **barra de los colaboradores**, ya que muestra la presencia de los colaboradores presentes en la sesión en curso;
- Control de acceso:
 - la **ventana de autenticación**, permite identificar al usuario en uno o más dispositivos,
- Meta-interfaz de redistribución:
 - la **ventana de redistribución**, porque permite redistribuir la interfaz de usuario en una PC y en una PDA.

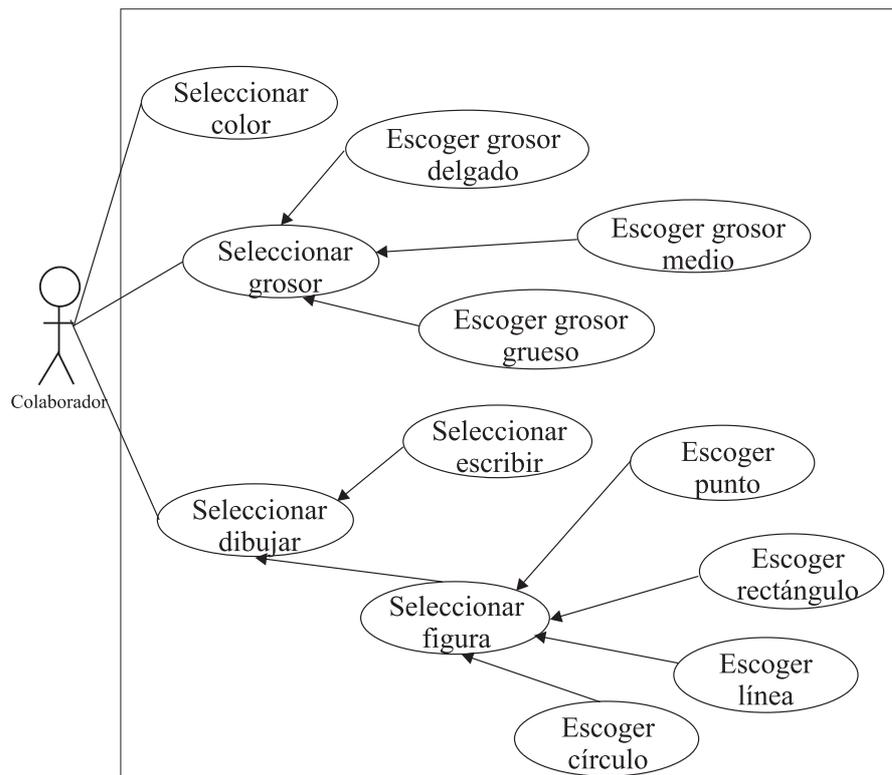


Figura 3.10: Caso de uso de la barra de herramientas

3.4.1 Barra de herramientas

El caso de uso de la barra de herramientas (cf. Figura 3.10) detalla las opciones que puede elegir un colaborador, las opciones disponibles son color, grosor y figura.

Como se ha mencionado a lo largo del diseño, la barra de herramientas proporciona la opción de seleccionar el color deseado, los colores a seleccionar son: amarillo, rojo, negro, azul, café, guinda, azul cielo, rosa, verde, violeta y verde lima.

En cuanto al grosor de las figuras, se eligieron tres tipos: delgado, medio y grueso. Respecto a las figuras que el usuario puede dibujar, se permiten dos tipos: pre-establecidas (línea, círculo y rectángulo) o diseñadas por el usuario a mano alzada.

El mayor problema en el diseño del pizarrón compartido es el reducido tamaño de la pantalla de la PDA. Por tal motivo es posible ocultar la barra de herramientas para liberar espacio que será aprovechado por el área de dibujo. Asimismo, la barra de herramientas puede ser redistribuida, i.e., puede ser visualizada en la PDA o en la PC cuando el usuario esté usando ambos dispositivos. La barra de herramientas migra completamente de un dispositivo a otro, no por partes.

3.4.2 área de dibujo

Usualmente el tamaño de la pantalla de la PDA es mucho menor que el de la PC, así que el primer problema a resolver consiste en definir el área de dibujo. Las tres opciones contempladas se describen a continuación:

1. La primera opción establece un tamaño fijo, el cual podría ser el que proporciona la PDA, sin embargo en la PC restaría demasiado espacio sin aprovechar. Si se usa como tamaño fijo el de la PC, la PDA tendría que manipular un espacio de memoria muy grande para contemplar toda el área de dibujo de la PC.
2. La segunda opción no modifica el tamaño del área de dibujo, sino que escalaría las figuras en ambos dispositivos, i.e., se reducirían las figuras provenientes de la PC para poder ser desplegadas en la PDA, y se agrandaría el tamaño de las figuras provenientes de la PDA para poder ser mostradas en la PC. Sin embargo, esta solución tiene dos inconvenientes: el primero concierne la posibilidad de que las figuras se vuelvan tan pequeñas en la PDA, que resulte imposible distinguirlas; el segundo se refiere a la dificultad de implementar en la PDA las proporciones adecuadas de una figura para poder ser visualizada en la PC.
3. La tercera opción establece un tamaño fijo, que sea más grande que el tamaño de la pantalla de la PDA, pero que no sea tan pequeño para la PC. Esta opción permite respetar el tamaño de las figuras sin importar en donde sean creadas.

Después de analizar las tres opciones, la más viable es la última porque no se satura la memoria de la PDA ni se presentan los problemas de distorsión de las figuras al escalarlas.

En la Figura 3.11 se muestra el diagrama de secuencias para actualizar el área de dibujo. Se hace la distinción entre dos sitios (A y B), los cuales corresponden a dos dispositivos diferentes como una PDA y una PC. También se observan los siguientes módulos:

- **área de dibujo:** es responsable de capturar las coordenadas señaladas por el colaborador y de pintar las figuras realizadas por este respetando las características dadas, e.g., color, tamaño y grosor;
- **base de datos:** almacena los datos necesarios para actualizar el pizarrón colaborativo, e.g., la selección actual del color, de la figura y del grosor, así como los colaboradores conectados;
- **el módulo de comunicación:** permite informar a las demás instancias del pizarrón colaborativo sobre la creación o modificación de figuras.

En el sitio A se encuentra un colaborador que está dibujando una figura. Cuando el colaborador pinta sobre el **área de dibujo**, esta envía un mensaje a la **base de datos** local para solicitarle que la almacene. Después el **área de dibujo** informa al **módulo de comunicación** de la creación de una figura. Posteriormente, el **área de dibujo** solicita a la **base de datos** local la lista de figuras existentes, con la finalidad de mostrar el nuevo

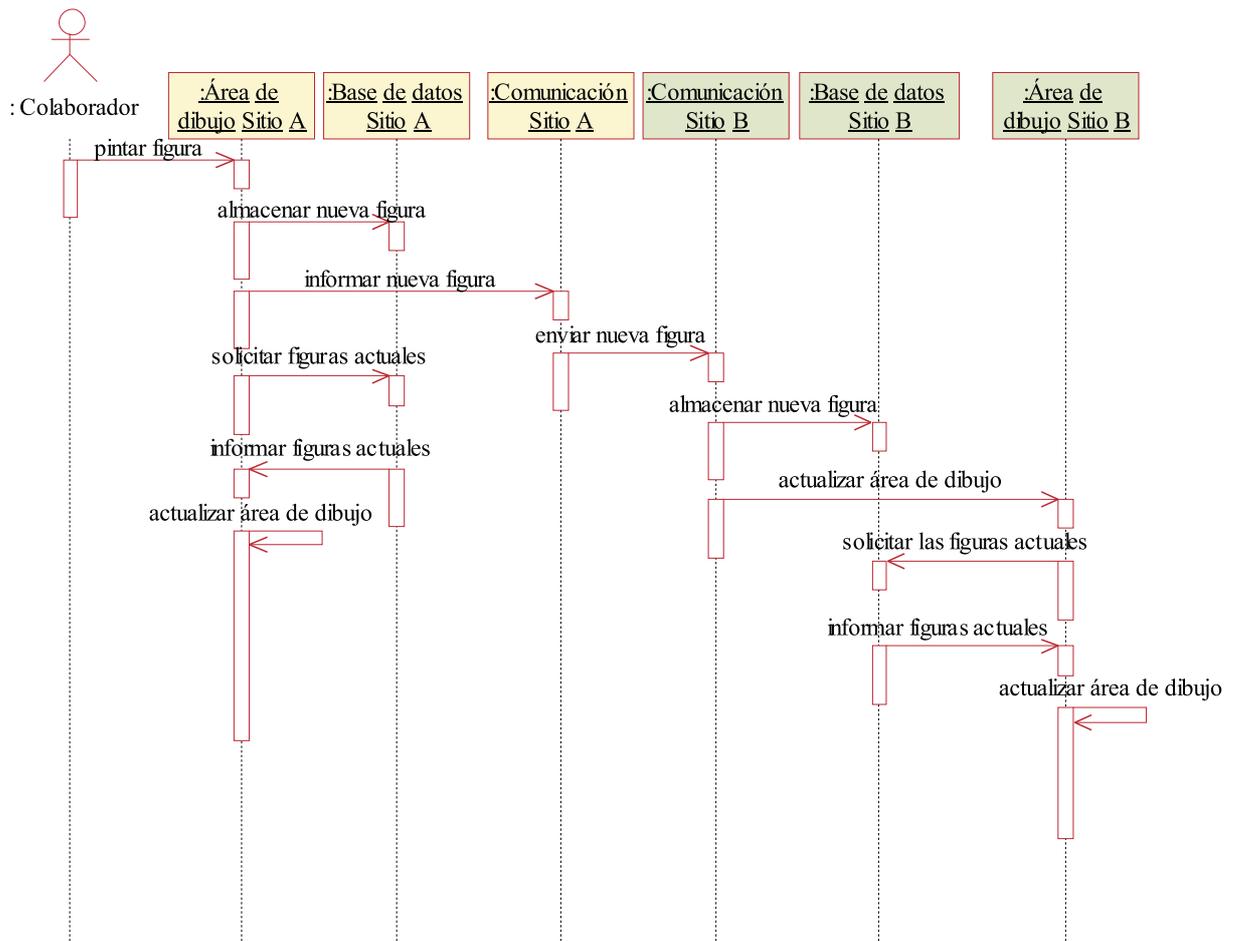


Figura 3.11: Actualización del área de dibujo en los sitios A y B

cambio (sitio A).

Cuando el módulo de comunicación del sitio A recibe el mensaje referente a la creación de una nueva figura, este avisa a su par localizado en el sitio B. Este aviso incluye las características de la figura. Al recibir este mensaje, el módulo de comunicación del sitio B solicita a la base de datos local que almacena la nueva figura. Posteriormente, el módulo de comunicación avisa al área de dibujo, la cual obtiene la lista de figuras desde la base de datos del sitio B. Por último, el área de dibujo se actualiza usando la lista de figuras.

3.4.3 Barra de colaboradores

La barra de colaboradores indica a los usuarios quién está presente en la sesión en curso. Los componentes gráficos que representan la presencia de los colaboradores tienen dos posibles formas: el nombre o la foto de a cada colaborador.

Como el objetivo de la barra colaboradores es “mostrar su presencia en todo momento”, entonces se desplegará la foto de cada colaborador junto con su nombre si existe espacio disponible e.g., en una PC . Cuando el espacio sea reducido, e.g., en una PDA, se mostrará sólo el nombre.

En la Figura 3.12 se muestra el diagrama de secuencias para actualizar la barra de colaboradores. El diagrama se compone básicamente de cuatro módulos, los cuatro de la izquierda están ubicados en el sitio A y los restantes están albergados en el sitio B. Estos módulos son:

- **módulo de autenticación:** reconoce si el nombre de usuario y la contraseña de un colaborador están registrados. Si el colaborador no está registrado, este módulo queda en espera de un nombre de usuario válido o de la salida del sistema. Por el contrario, si el colaborador está registrado, permite continuar la ejecución normal;
- **barra colaborador:** se encarga de informar al colaborador la presencia de sus colegas;
- **base de datos:** almacena los nombres de los colaboradores en el orden en que ingresaron a la sesión actual. También elimina los nombres de los usuarios que han salido de la sesión;
- **módulo de comunicación:** permite la comunicación entre las instancias del pizarrón compartido, para informar sobre el ingreso y la salida de los colaboradores. El mensaje que envía este módulo adjunta el nombre del colaborador que ingresa o sale de la sesión.

El proceso de actualización de la barra de colaboradores (cf. Figura 3.12) empieza cuando el colaborador ingresa su nombre de usuario y contraseña al pizarrón compartido. El **módulo de autenticación** recibe esta información. Después de verificar que los datos son correctos, este módulo solicita a la **base de datos** local que actualice la lista de colaboradores presentes en la sesión. Junto con el aviso envía el nombre del colaborador que se acaba de unir a la sesión. Posteriormente, el **módulo de autenticación** anuncia a la barra de colaboradores sobre el ingreso del colaborador.

La **barra de colaboradores** notifica al **módulo de comunicación** para que este informe a su para en el sitio B. Por otra parte, la **barra de colaboradores** se comunica con la **base de datos** local para solicitar el registro de colaboradores. Cuando la **barra de colaboradores** recibe el registro, esta muestra en pantalla el ingreso del colaborador a la sesión por medio de su foto y/o nombre.

Cuando el **módulo de comunicación** en el sitio B recibe la información, este se comunica con la **base de datos** local para proporcionarle la nueva información. Posteriormente, el **módulo de comunicación** informa a la **barra de colaboradores**, sobre el ingreso de un colaborador. A continuación, la **barra de colaboradores** se comunica con la **base de datos** para solicitar el registro de colaboradores existentes. Por último la **barra de colaboradores** presenta la foto y/o nombre del colaborador.

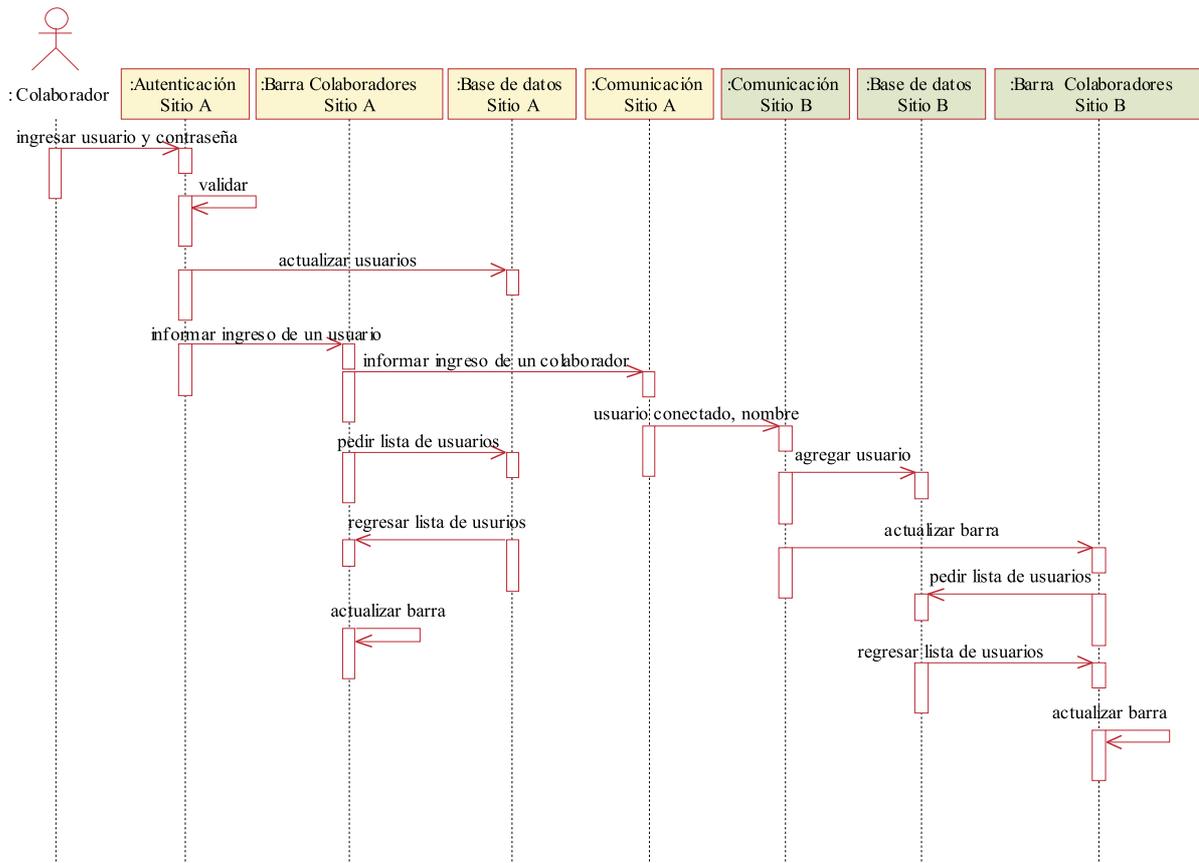


Figura 3.12: Actualización de la barra de colaboradores

3.4.4 Meta-interfaz de redistribución

La meta-interfaz de redistribución se activa automáticamente cuando el pizarrón compartido detecta que un mismo usuario está trabajando en la PDA y la PC. Por medio de esta meta-interfaz, el colaborador puede seleccionar los componentes gráficos que desea redistribuir.

La redistribución es un aspecto fundamental de esta tesis, así que es importante establecer qué partes de la interfaz de usuario serán redistribuidas. Las opciones posibles se describen a continuación:

- redistribuir a nivel del componente gráfico de cada interactor, e.g., un color o una figura. Sin embargo, esta opción presenta varias desventajas: a) el colaborador puede olvidar asignar un componente gráfico a una instancia destino del pizarrón compartido, b) el usuario podría saturar el espacio de una pantalla pequeña al incorporar demasiados componentes gráficos y c) el usuario podría cambiar las posiciones de los componentes gráficos de tal forma que no le sea fácil ubicarlos.
- redistribuir a nivel de interactor, e.g., la barra de herramientas o de colaboradores. Esta opción tiene la ventaja de que todos los componentes de un interactor migren

a un mismo dispositivo, lo cual permitiría al colaborador localizar fácilmente la posición de cada uno.

La mejor opción para redistribuir la interfaz de usuario es a nivel de interactor. Así, los interactores susceptibles de redistribución son la barra de herramientas y la de colaboradores.

En el diagrama de secuencias de la Figura 3.13 se detalla la migración de la barra de herramientas al sitio B. El diagrama se compone de dos sitios (A y B) y de cuatro módulos. Los módulos son:

- **módulo de redistribución:** se encarga de desplegar las opciones de redistribución, las cuales permiten trasladar las barras de herramientas y de colaboradores hacia una PDA, además de llevar a cabo la redistribución;
- **base de datos:** almacena y regresa la última selección de los interactores;
- **módulo de comunicación:** se encarga de avisar a otro interactores los interactores a redistribuir;
- **módulo de herramientas:** muestra los interactores y responde a los eventos generados por el colaborador.

Suponga que un colaborador está trabajando en el pizarrón compartido desde el sitio A. Posteriormente ingresa al pizarrón compartido desde el sitio B. El sistema detecta al mismo usuario en ambos dispositivos, así que despliega la meta-interfaz de redistribución. El colaborador selecciona migrar la **barra de herramientas** al sitio B (cf. Figura 3.13). En respuesta, el **módulo de redistribución** pide a la **base de datos** la selección actual (color y figuras) de la barra de herramientas. Una vez que la base de datos proporcionó la selección actual, el **módulo de redistribución** envía al **módulo de comunicación** la orden de redistribuir la **barra de herramientas** junto con la selección actual. En seguida la **comunicación** del sitio A envía a su par en el sitio B la orden de redistribuir la **barra de herramientas** y la selección actual. Por último, el **módulo de redistribución** en el sitio A informa a la **barra herramientas** que su presentación no debe mostrarse porque será desplegada en el sitio B.

El **módulo de comunicación**, al recibir la orden de redistribución, pasa los valores seleccionados de la **barra de herramientas** hacia la **base de datos**, la cual almacena estos valores. Posteriormente, el **módulo de comunicación** informa a la **barra de herramientas** que su presentación debe ser visualizada en ese sitio. En respuesta, la **barra de herramientas** obtiene los valores actuales de la **base de datos**. Por último, la **barra de herramientas** en el sitio B despliega sus componentes gráficos usando los valores seleccionados por el colaborador.

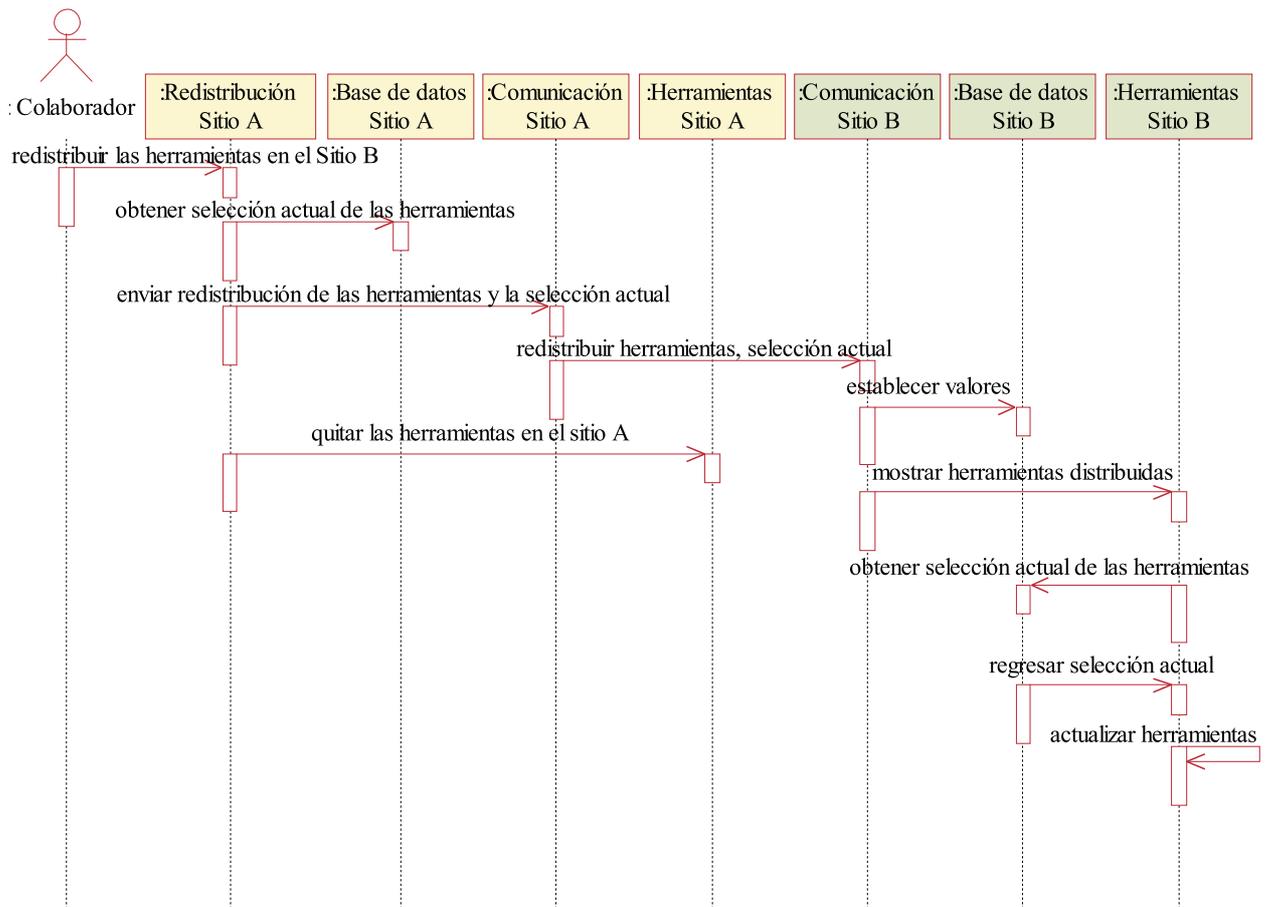


Figura 3.13: Redistribución de la barra de herramientas entre el sitio A y B

3.4.5 Ventana de autenticación de usuario

En la figura 3.14 se detalla la autenticación del usuario. El diagrama de secuencias tiene los siguientes módulos:

- **módulo de autenticación:** verifica que el nombre de usuario y la contraseña sean validos;
- **módulo de comunicación:** informa a las demás instancias del pizarrón compartido del ingreso de un nuevo colaborador;
- **base de datos del sitio A:** almacena el nombre de usuario y la contraseña de todos los colaboradores;
- **base de datos del servidor:** almacena la última configuración de todos los colaboradores y las figuras realizadas.

Suponga que un colaborador desea ingresar al pizarrón compartido desde el sitio A (cf. Figura 3.14). Al momento de ingresar al pizarrón compartido, este le solicita su

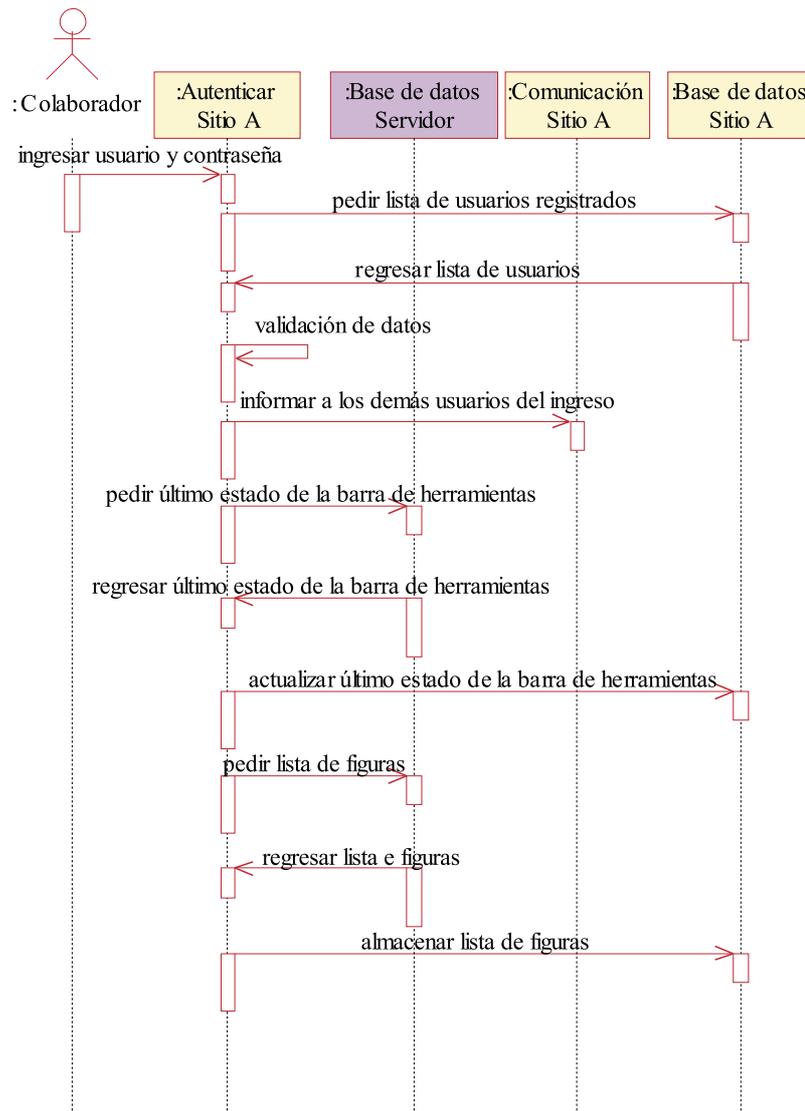


Figura 3.14: Autenticación de un colaborador

nombre de usuario y su contraseña. Una vez que el colaborador ha ingresado la información requerida, el módulo de autenticación recurre a la base de datos local para corroborar la información. Si esta corresponde a la información registrada en la base de datos, el módulo de autenticación informa al módulo de comunicación para que este informe sobre el ingreso a sus pares en los sitios remotos. Posteriormente el módulo de autenticación se comunica con la base de datos del servidor para solicitar el último estado del colaborador que acaba de ingresar, e.g., la última selección de la barra de herramientas. Cuando el módulo de autenticación recibe los datos solicitados, estos son enviados a la base de datos local para que esta los almacene.

Después, el módulo de autenticación pide a la base de datos del servidor el registro actual de figuras. La información que recibe también es enviada a la base de datos para que esta actualice su registro de figuras existentes. Finalmente, el módulo

de autenticación pide a la base de datos del servidor el registro de colaboradores conectados. Al recibir esta información el módulo de autenticación la envía a su base de datos.

Después de que el módulo de autenticación verifica la identidad del colaborador, las barras de herramientas y de colaboradores pueden ser inicializadas con la información actualizada de su propia base de datos.

Capítulo 4

Implementación

Una de las tareas fundamentales en la implementación de un sistema es definir el lenguaje de programación más conveniente para desarrollarlo. ya que el ambiente de programación de una PC difiere del reducido ambiente de programación de una PDA, más si esta última tiene sistema operativo Linux (cf. sección 4.1). Después de haber seleccionado el lenguaje de programación se proponen tres interfaces de usuario para la PC y cinco interfaces de usuario para la PDA (cf. sección 4.2). Del conjunto de interfaces de usuario para la PC y para la PDA se seleccionaron las que presentan una mejor organización de sus interactores, ya que podrían ser las más amigables para el usuario. Posteriormente se detallan los algoritmos más relevantes en la implementación del pizarrón compartido para la PDA y de la PC , e.g., la barra de desplazamiento (cf. sección 4.3). Finalmente, se presentan las pruebas realizadas en el pizarrón compartido (cf. sección 4.4).

4.1 Selección del lenguaje de programación

El primer problema a resolver para implementar el pizarrón compartido es definir el lenguaje de programación es más conveniente, al igual que el entorno de desarrollo (IDE). En la fase de implementación del pizarrón compartido se disponía de dos PDAs. Una de las PDAs es la HP iPAQ HW6945 con sistema operativo Windows Mobil 5.0 y la otra PDA es la HP 3900 con Linux Familiar.

4.1.1 PDA con Linux

La PDA HP 3900 puede utilizar los lenguajes de programación JAVA y C, debido a que esta PDA tiene instalado el sistema operativo Linux Familiar. Las posibles herramientas que se pueden utilizar son: a) OpenEmbedded junto con Open *Palmtop Integrated Environment* (Opie) y b) Kaffe.

OpenEmbedded (OE)

OpenEmbedded es un entorno de desarrollo que sirve para una gran variedad de dispositivos, y que soporta múltiples compilaciones. Este entorno de desarrollo ofrece un sistema de compilación para generar paquetes (compilador cruzado) de software, en específico para tarjetas embebidas. OpenEmbedded requiere de Bitbake (ejecutor de tareas [[URL: Bitbake](#)]), al igual que de Monotone (sistema de control de versiones) para su funcionamiento [[URL: OpenEmbedded](#)].

Los pasos a seguir para obtener una aplicación ejecutable en Linux Familiar usando OpenEmbedded son los siguientes: 1) instalar el compilador OpenEmbedded en la PC para familiarizarse con el compilador, 2) aprender a crear una aplicación ejecutable en Linux familiar usando OpenEmbedded, 3) crear el ejecutable del pizarrón compartido para el Linux Familiar y finalmente 4) descargar el pizarrón compartido en la PDA con Linux.

La instalación del compilador OpenEmbedded se realizó de dos formas diferentes, ambas instalaciones se realizaron en una computadora portátil con distribución de Linux Fedora. La primera instalación de OpenEmbedded se realizó siguiendo las instrucciones de la página Web de Linux Familiar [[URL: Linux Familiar](#)]. El proceso de instalación de OpenEmbedded se describe en el Apéndice A.

La otra instalación se realizó siguiendo las instrucciones de la página Web de OpenEmbedded. Al finalizar la instalación se decidió descartarla ya que la versión más reciente que contemplaba OpenEmbedded era Linux-familiar.0.8.3, mientras que la PDA tiene instalada la versión 0.8.4. Posterior a la instalación se ejecutaron algunos programas sencillos, los cuales fueron compilados en la PC y posteriormente fueron ejecutados en la PDA sin problema.

Es importante señalar que la versión de Linux Familiar instalada en la PDA utiliza el entorno gráfico Opie (*Open Palmtop Integrated Environment*).

Open Palmtop Integrated Environment (Opie)

Opie es un entorno gráfico de código abierto. Este entorno está diseñado para ser ejecutado en dispositivos pequeños que utilicen sistema operativo Linux. Opie es una bifurcación de Qtopia (una plataforma de aplicación y una interfaz de usuario para Linux [[URL: Qtopia](#)]), debido a que Opie extiende y mejora Qtopia [[URL: Opie](#)].

El motivo de usar Opie es que este permite crear aplicaciones con interfaz gráfica de usuario. Por otro lado OpenEmbedded permite crear aplicaciones ejecutables para la PDA. Así que se necesita Opie para crear la interfaz gráfica del pizarrón compartido, posteriormente con OpenEmbedded se crea la versión ejecutable del pizarrón compartido, la cual será ejecutada en la PDA con Linux Familiar.

Sin embargo, la instalación de Opie no se concluyó por algunos problemas de configuración. El proceso de instalación de Opie se describe en el Apéndice A.

Kaffe

Otra opción para implementar el pizarrón compartido era utilizar una máquina virtual Java en la PDA con Linux. En la tabla 4.1 se listan las máquinas virtuales encontradas. Existen dos opciones libres para Linux Familiar. Ambas opciones son compatibles con la PDA con sistema operativo Linux. Una de ellas es Blackdown, la cual fue descartada porque ya no es mantenible. La otra opción es Kaffe.

Tabla 4.1: Máquinas virtuales para ejecutar Java en una PDA

Máquina Virtual	Blackdown J2RE (ARM port)	Kaffe
Sistema operativo	Linux Familiar	Linux Familiar
Compatibilidad JMV	Java 2 (1.3.1)	Kaffe
Hardware soportado	iPAQ H36xx	PocketPC/ARM
Costo	gratuito	gratuito
Mantenible	no	si

Kaffe es una máquina virtual que permite la ejecución de código Java [[URL: Kaffe](#)]. Tiene un subconjunto de las clases J2ME y las herramientas necesarias para ejecutar Java en tiempo de ejecución. Kaffe es una distribución gratuita, ejecutable sobre Linux (excepto sobre Linux 1.2.13) que requiere poco espacio para instalarse. Por tales motivos Kaffe resulta atractivo a los desarrolladores de sistemas embebidos [[URL: Kaffe](#)].

Sin embargo, se descartó Kaffe porque presentaba problemas de compatibilidad con un archivo en la PDA al momento de la instalación. El proceso de instalación de Kaffe se describe en el Apéndice A.

4.1.2 PDA con Windows Mobil 5.0

La única opción disponible para la PDA HP iPAQ HW6945 es la plataforma JWT (*Java Wireless Toolkit*). JWT anteriormente era conocida como J2ME (*Java 2 Micro Edition*). J2ME es una plataforma que permite desarrollar aplicaciones para dispositivos móviles o embebidos, e.g., teléfonos móviles, PDA e impresoras [[Universidad de Malaga, 2003](#)]. Las interfaces de usuario del pizarrón compartido para la PDA se realizaron con J2ME usando el IDE NetBeans 6.1.

4.1.3 Selección de la herramienta para la PC

El diseño del pizarrón compartido (cf. sección 3.2) se realizó por medio del estilo arquitectural MVC. En el capítulo precedente se mencionó que esta arquitectura admite estructurar la aplicación en objetos, lo cual posibilita el uso lenguajes de programación orientados a objetos, e.g., Java o SmallTalk. J2SE (Java 2 Platform, Standard Edition) contiene la librería de componentes Swing, la cual está basada en el estilo arquitectural MVC [Eckstein et al., 1998]. Por lo anterior, se puede decir que es una buena opción elegir Java como lenguaje de programación. Otro factor importante para esta elección se debe a que la PDA con Windows sólo permite programar con J2ME.

Otra ventaja adicional de emplear J2SE reside en su portabilidad, ya que un programa creado en Java puede ser ejecutado en el sistema operativo Windows y Linux, siempre y cuando se tenga instalada la máquina de virtual de Java.

Actualmente existen varios entornos de desarrollo (IDE) para programar con el lenguaje Java. Para la implementación del pizarrón compartido se optó por el IDE JBuilder 2005.

4.2 Interfaz de usuario

Una tarea ineludible en el desarrollo de este tema de investigación es elegir la mejor interfaz de usuario, debido a las diferentes características físicas de los dispositivos, e.g., tamaño de la ventana. La interfaz de usuario para una PC no es tan complicada de crear como en una PDA, ya que una PC tiene mayor espacio en ventana para desplegar todos los componentes gráficos sin problema alguno.

En el capítulo de diseño se mencionó que el pizarrón está formado por cinco interactores (cf. Figura 4.1).

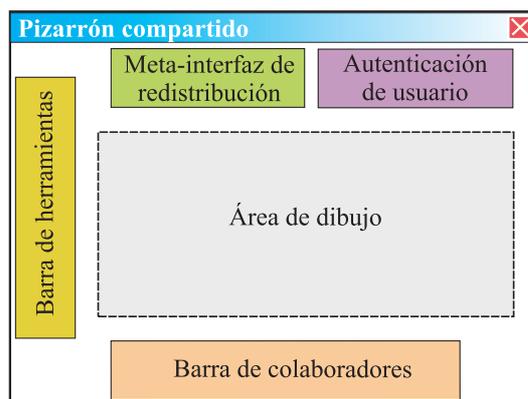


Figura 4.1: Interactores del pizarrón compartido

El interactor de la barra de herramientas incluye los siguientes componentes gráficos: colores (i.e., amarillo, azul, café, negro y rosa), figuras (i.e., círculo, cuadrado y punto)

y grosores de línea (i.e., delgado, medio y grueso). Los componentes gráficos de la barra de herramientas están agrupados por funcionalidad, i.e., los componentes de color se encuentran separados de los componentes de figura y de los componentes de grosor de línea.

El área de dibujo es el interactor que ocupa mayor espacio de ventana, ya que muestra los dibujos realizados por los colaboradores.

El interactor de la barra de colaboradores tiene como componentes gráfico el nombre y/o la foto de cada colaborador que está presente en la sesión en curso.

El interactor de la meta-interfaz de redistribución posee las opciones de los interactores en la PC susceptibles de redistribución. Los interactores que pueden ser redistribuidos son la barra de herramientas y la barra de colaboradores. La meta-interfaz de redistribución ofrece las siguientes opciones al usuario: 1) ambos interactores pueden ser visualizados en los PDA y 2) uno de los interactores mencionados se puede distribuir a la PDA y el otro a la PC. Otro componente gráfico que incluye la meta-interfaz es el botón **aceptar**, el cual permite ejecutar las opciones seleccionadas por el usuario en la meta-interfaz.

El interactor de autenticación del usuario contiene los siguientes componentes gráficos: dos etiquetas (una para el nombre de usuario y otra para la contraseña) y el botón de **aceptar**.

4.2.1 Interfaz de usuario para la PC

Las interfaces gráficas de usuario del pizarrón compartido que fueron contempladas para la PC se muestran en la Figura 4.2. Se puede apreciar en estos prototipos que la barra de colaboradores se forma por la foto y el nombre de cada colaborador, ya que existe espacio disponible.

Primer prototipo

El primer prototipo para la interfaz de usuario en la PC se observa en la Figura 4.2.a. En la parte superior de la ventana principal se encuentran dos interactores. El primero es la barra de herramientas formada por sus componentes gráficos (formas, colores y grosores de línea). Estos componentes gráficos fueron divididos en tres grupos: a) colores, b) formas y c) grosores de línea.

El segundo interactor del primer prototipo es la meta-interfaz de redistribución, la cual está constituida por un único componente gráfico (el botón **redistribuir**) que aparece en la vista principal. Los demás componentes gráficos aparecen en la ventana de la meta-interfaz.

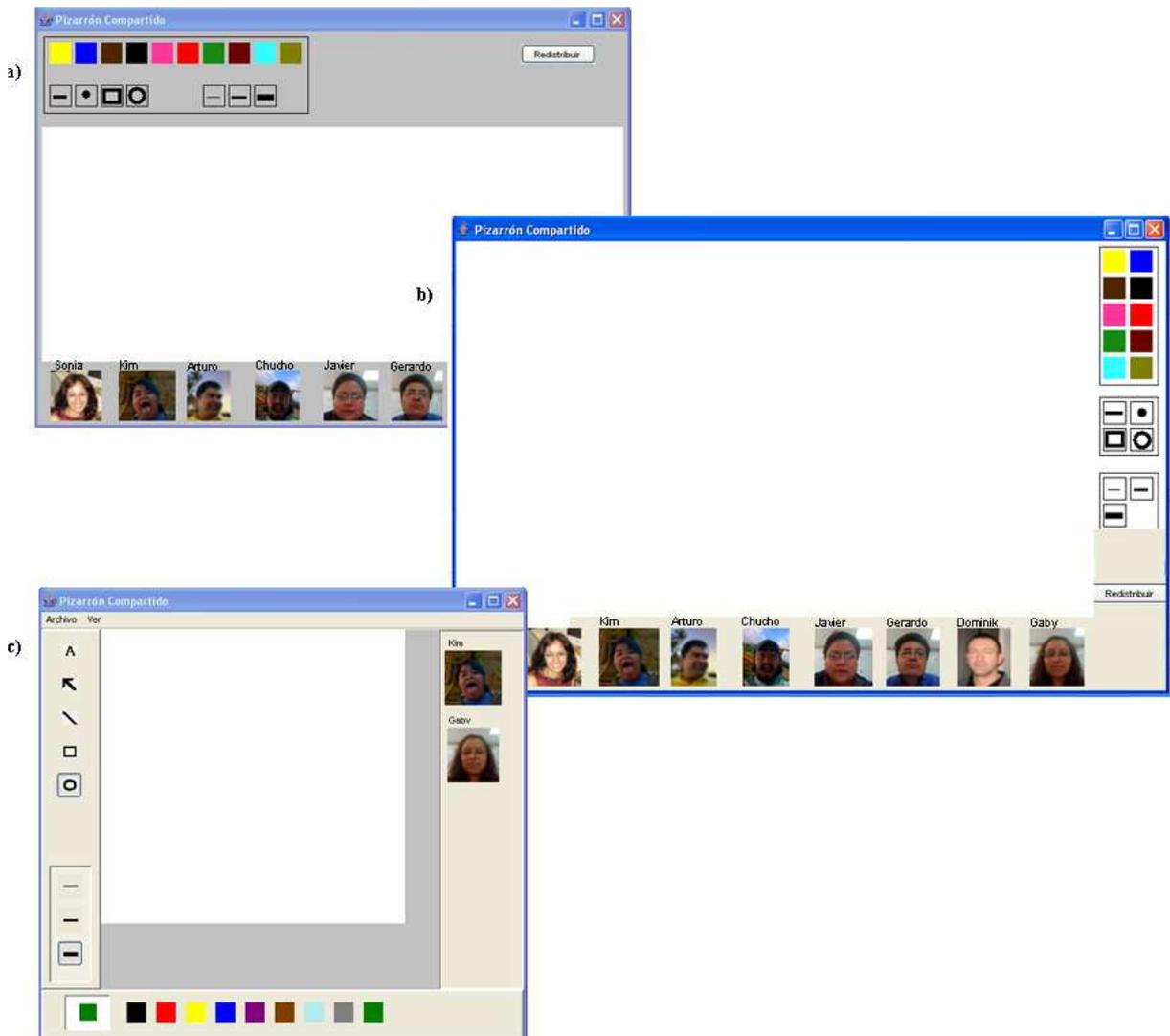


Figura 4.2: Interfaces de usuario propuestas para el pizarrón compartido en la PC

En la parte media de la ventana principal (cf. Figura 4.2.a.), se encuentra el área de dibujo y finalmente, en la parte inferior se ubica la barra de colaboradores en forma horizontal.

Segundo prototipo

El segundo prototipo de la interfaz gráfica de usuario se muestra en la Figura 4.2.b. Del lado derecho de la ventana principal se muestran dos interactores. En la parte superior derecha se ubica la barra de herramientas con sus correspondientes componentes gráficos. Del lado inferior derecho también se encuentra el botón de redistribución que da acceso a la meta-interfaz de redistribución con .

El área de dibujo ocupa la parte izquierda y el centro de la ventana. Por último, en

la parte inferior de la ventana, se localiza la barra de colaboradores en forma horizontal, cubriendo todo el ancho de la ventana.

Tercer prototipo

El último prototipo (cf. Figura 4.2.c) está inspirado en la aplicación *Paint* de Microsoft porque la mayoría de los usuarios alguna vez han usado esta aplicación, así podrían estar familiarizados con la ubicación de los componentes gráficos.

En la parte superior de la ventana (cf. Figura 4.2.c) se encuentra el menú con las opciones **archivo** y **ver**. La opción **archivo** (cf. Figura 4.3.a) contiene a su vez las opciones: **redistribuir** y **salir**. Por último, la opción **ver** (cf. Figura 4.3.b) muestra las opciones: **figuras**, **colores** y **colaboradores**. La opción **ver** permite ocultar o mostrar los interactores en la instancia del pizarrón compartido local.

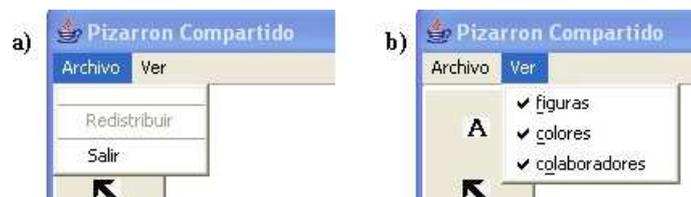


Figura 4.3: Menú del tercer prototipo de la interfaz de usuario

La interfaz de usuario de la Figura 4.2.c divide la barra de herramientas en dos partes. La primera parte, localizada en la parte izquierda de la ventana principal, está formada por los componentes gráficos de las formas y del grosor de línea. La segunda parte, localizada en la parte inferior de dicha ventana, se integra por los componentes gráficos de los colores. Por último, la barra de colaboradores se presenta del lado derecho en forma vertical.

Selección de la interfaz gráfica para la PC

Los dos primeros prototipos (cf. Figura 4.2 incisos a y b) no son del todo satisfactorios. El primero tiene el inconveniente de reducir el área de dibujo, aunque las herramientas aparecen organizadas. El segundo prototipo proporciona más espacio en el área de dibujo, pero los componentes gráficos de la barra de herramientas se aprecian demasiado juntos. El último prototipo (cf. Figura 4.2.c) muestra una mejor organización en los componentes gráficos de la barra de herramientas.

Al contrario del segundo prototipo, el primero y el tercero (cf. Figura 4.2 a y c) permiten ubicar fácilmente los componentes de la barra de herramientas (cf. Figura 4.2.b).

El prototipo más conveniente es el tercero (cf. Figura 4.2.c) porque su organización permite ubicar rápidamente los componentes gráficos, además de que el usuario final podría

estar más familiarizado con dicha interfaz de usuario.

4.2.2 Interfaz de usuario para la PDA

J2ME cuenta con dos clases para manipular la interfaz de usuario: *Screen* y *Canvas*. La clase *Screen* permite crear la interfaz de usuario de manera rápida, gracias a que incluye algunos componentes gráficos (e.g., botones y menús).

Los primeros cuatro prototipos fueron creados mediante los componentes gráficos que incluye la clase *Screen* de J2ME. La ubicación de estos componentes no puede ser manipulada por el programador porque la plataforma los coloca en el orden en que se van creando. La imposibilidad de asignar una posición específica dichos componentes se convierte en una limitante para obtener una buena interfaz de usuario. Por tal motivo se creó el último prototipo.

La segunda clase ofrecida por J2ME para manipular la interfaz de usuario es la clase *Canvas*, la cual permite manipular la ventana a nivel gráfico. Por tal motivo, esta clase fue seleccionada para manipular el área de dibujo. Posteriormente, a causa de los problemas ocasionados por la falta de control en el posicionamiento de los componentes gráficos en la ventana, se utilizó esta clase para la interfaz de usuario del quinto prototipo. En específico se usó la clase *GameCanvas*, la cual hereda de *Canvas*, ya que *GameCanvas* ofrece la posibilidad de crear animaciones rápidas y libres de parpadeo, gracias a que cuenta con doble *buffer* para repintar la ventana.

Primer prototipo

El primer prototipo (cf. Figura 4.4) se compone de dos ventanas, en una muestra el área de dibujo (cf. Figura 4.4.a), mientras que en otra se despliegan todos los componentes gráficos de la barra de herramientas y la de colaboradores (cf. Figura 4.4 b y c).

En la parte superior de la Figura 4.4.a se muestra el título de la aplicación **Pizarrón compartido**. Debajo del título se observa el área de dibujo representado por un espacio en blanco. Finalmente en la parte inferior izquierda se muestra la opción **salir**, la cual permite salir de la aplicación. En la parte inferior derecha de la ventana se presenta la opción **menú**, la cual posee una sola opción llamada **otras opciones**. Esta opción conduce a otra ventana (cf. Figura 4.4 b y c) que contiene los interactores: la barra de herramientas, la meta-interfaz de redistribución y la barra de colaboradores.

La segunda ventana de la Figura 4.4 requiere que el usuario se desplace horizontalmente, ya que los componentes gráficos de ambos interactores se muestran al mismo tiempo. El primer conjunto de componentes gráficos pertenece a la de barra de herramientas. Esta se implementó por medio de radio botones (*RadioButton*), los cuales están compuestos por el nombre del componente gráfico junto con una imagen que lo representa. El orden de



Figura 4.4: Primer prototipo de la interfaz de usuario para la PDA, a) la ventana principal muestra el área de dibujo, mientras que b) la barra de herramientas así como la meta-interfaz de redistribución y la barra de colaboradores se despliegan en una ventana secundaria

los componentes gráficos de las herramientas es el siguiente: en primer lugar se ubican los colores (e.g., rosa, rojo y negro), posteriormente se encuentran las formas (e.g., círculos y líneas) y finalmente, se muestran los grosores de línea (e.g., delgada y media). En seguida se observa la barra de colaboradores utilizando sus nombres y sus fotos (cf. Figura 4.4.b). Por último aparece la meta-interfaz de redistribución, la cual muestra las opciones para redistribuir (herramientas y colaboradores) en la PDA y en la PC.

Al final de la ventana se observa la opción **aceptar** para regresar a la ventana principal, la cual contiene el menú.

Segundo prototipo

El segundo prototipo dos de la interfaz de usuario se compone de dos ventanas (cf. Figura 4.5). La primera es similar a la ventana ilustrada en la Figura 4.4.a. Esta ventana muestra el título, el área de dibujo, la opción **salir** y el **menú** (cf. Figura 4.5.a) .

Por medio de la única opción que ofrece el menú, se accede a la segunda ventana (cf. Figura 4.5.b), la cual muestra los componentes gráficos de los interactores (barra de herramientas, meta-interfaz de redistribución y barra de colaboradores) mediante menús desplegados. Los componentes gráficos de la barra de herramientas utilizan una imagen seguida de un nombre, e.g., el color verde muestra un cuadro de color verde, además de la leyenda “verde”. En el caso de la barra de colaboradores, sus componentes gráficos sólo se componen del nombre de cada colaborador presente en la sesión en curso. Por último, se encuentra el menú desplegable **Redistribuir** correspondiente a la meta-interfaz.

Al final de la ventana se observa la opción **aceptar** que permite regresar a la ventana principal, la cual contiene el área de dibujo.

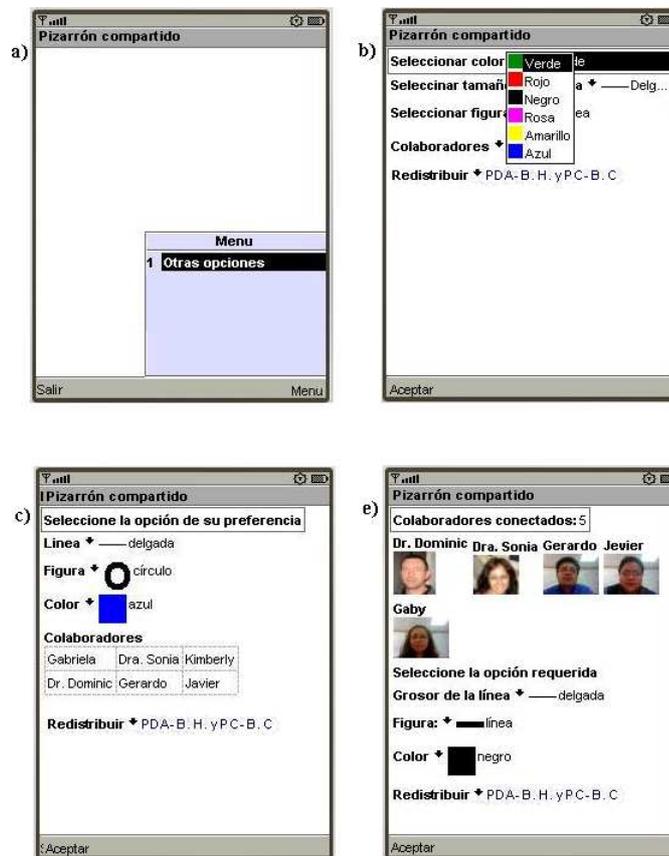


Figura 4.5: Segundo prototipo de la interfaz de usuario para la PDA, la cual se compone de: a) una ventana principal, b) una ventana secundaria, c) una primera variante de la ventana secundaria y e) una segunda variante de la ventana secundaria

Puesto que los interactores (barra de herramienta, meta-interfaz de redistribución y barra de colaboradores) se presentan de forma compacta, se requirió utilizar una sola ventana.

Respecto a las variantes del segundo prototipo, el cambio radica principalmente en los componentes gráficos de la barra de colaboradores.

En la primera (cf. Figura 4.5.c) los nombres de los colaboradores se muestran de manera extendida, i.e., el nombre de los colaboradores son exhibidos todos a la vez para que puedan ser apreciados sin necesidad de desplegar el menú de la barra de colaboradores.

En la segunda variante (cf. Figura 4.5.d) la barra de colaboradores cambia de lugar (en la Figura 4.5.c antes se mostraba en penúltimo lugar ahora es el primer interactor que aparece en la ventana). Otro cambio en la barra de colaboradores es que se presentan los componentes gráficos de manera extendida, usando el nombre y la foto de cada colaborador conectado. Aún cuando los componentes gráficos de la barra de colaboradores se ha agregado de forma extendida es posible visualizar a los demás interactores.

Tercer prototipo

En los prototipos anteriores, se muestran todos los interactores en una sola ventana, excepto por el área de dibujo. En algunos casos los componentes gráficos de estos interactores se despliegan de manera extendida (cf. Figura 4.4), lo cual implica usar una ventana larga. En otros casos, los componentes gráficos se presentan de manera simplificada (mediante menús desplegables), los cuales no generan una ventana larga (cf. Figura 4.5.b).

La desventaja de unir estos interactores (barra de herramientas, meta-interfaz de redistribución y barra de colaboradores) reside en la imposibilidad de visualizar la separación entre estos interactores. Por tal motivo, en este tercer prototipo se separaron los interactores.

Este prototipo ha sido implementado en siete ventanas (cf. Figura 4.6). La primera muestra el título, el área de dibujo, la opción salir y el menú.

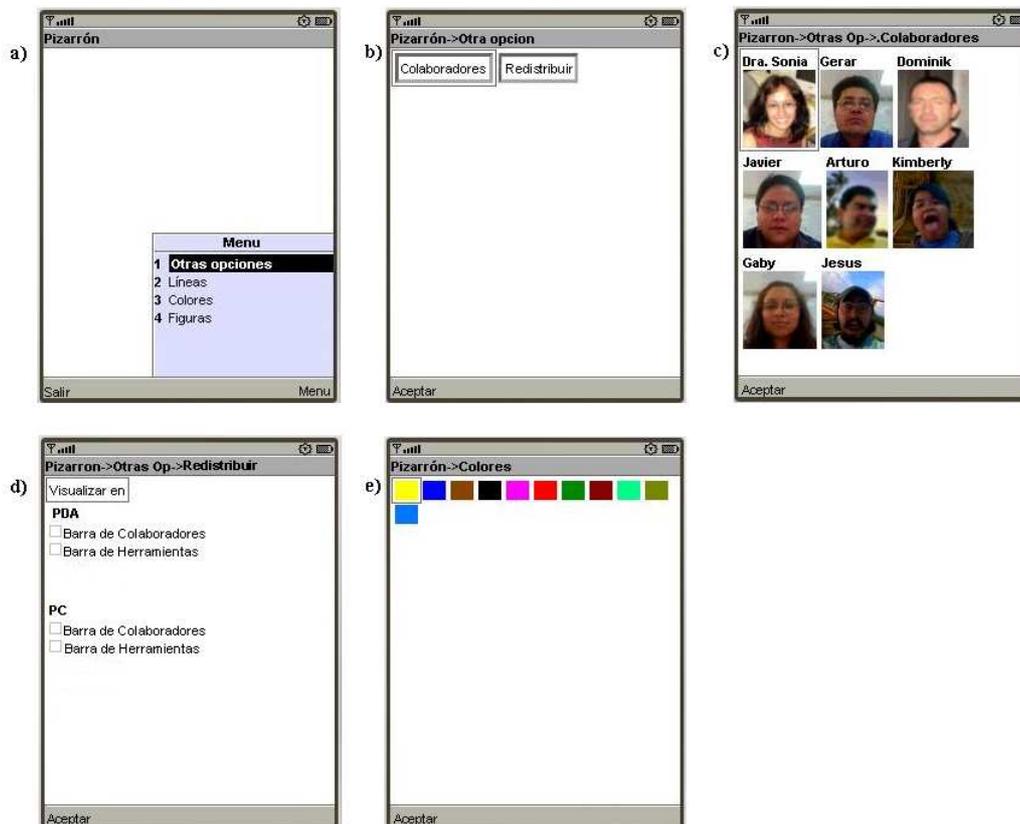


Figura 4.6: Tercer prototipo de la interfaz de usuario para la PDA, la cual muestra cinco ventanas de siete en total

El menú posee la opción **otras opciones**. Al seleccionar esta opción aparece la ventana de la Figura 4.6.b. La ventana **otras opciones** da acceso a la barra de colaboradores (cf. Figura 4.6.c), la cual contiene sus fotos en mayor tamaño respecto a las fotos mostradas en las Figura 4.4 y 4.5.e. También, la ventana **otras opciones** muestra la opción **Redistribuir**, la cual presenta la ventana de la meta-interfaz de redistribución (cf. Figura 4.6.d). Las opciones a redistribuir son la barra de colaboradores y la barra de herramientas.

Otra opción del menú es la opción **Líneas**. La opción **Líneas** despliega una ventana que contiene el grosor de línea (delgada, media y gruesa). Esta ventana no aparece en la figura mencionada. La otra opción del menú es **Colores**, cuya ventana sólo muestra las imágenes relacionadas con los colores (cf. Figura 4.6.e). La última opción del menú es la opción **Figuras**, la cual contiene en su ventana las formas (e.g., cuadro y círculo).

Cuarto prototipo

El cuarto prototipo de la interfaz de usuario para la PDA comprende cuatro ventanas (cf. Figura 4.7). La primera ventana contiene el título, el área de dibujo, la opción **salir** y el menú (cf. Figura 4.7.a). El menú está formado por tres opciones: 1) **Redistribuir**, 2) **Opciones** y 3) **Colaboradores** (cf. Figura 4.7.a), estas opciones se presentan por medio de radio botones (RadioButton).

La opción **Redistribuir** permite migrar en la PDA o en la PC la barra de colaboradores y la barra de herramientas.

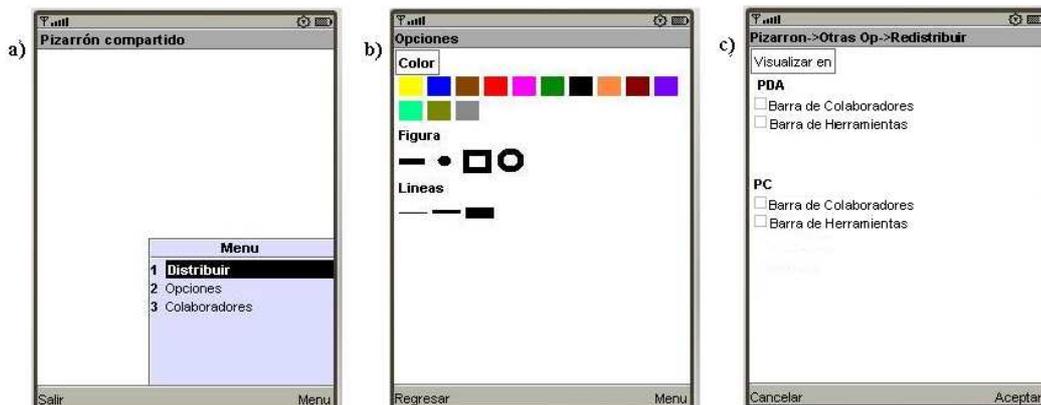


Figura 4.7: Cuarto prototipo de la interfaz de usuario para la PDA, la cual muestra tres ventanas de cuatro

La segunda opción del menú es **Opciones**, la cual contiene los componentes gráficos de la barra de herramientas (cf. Figura 4.7.b). Estos componentes sólo se presentan por medio de las imágenes que los representan, e.g., aparecen recuadros que contienen los

colores permitidos y las imágenes de las formas.

La última opción corresponde a la barra de colaboradores, la cual presenta los nombres de los colaboradores y sus respectivas fotos, como se observa en la Figura 4.6.c.

Quinto prototipo

Como se mencionó al inicio de esta sección, los componentes gráficos de este prototipo fueron creados usando la clase *GameCanvas*. Este prototipo utiliza dos ventanas: una ventana principal y una ventana secundaria.

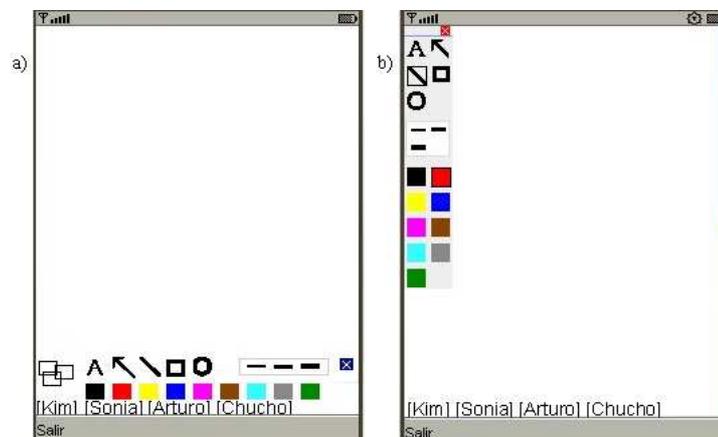


Figura 4.8: El quinto prototipo muestra en dos posiciones diferentes la barra de herramientas: a) horizontal y b) vertical

En la ventana principal se colocan la barra de herramientas, el área de dibujo y la barra de colaboradores (cf. Figura 4.8.a).

La barra de herramientas se encuentra localizada en la parte inferior a lo largo de la ventana, la cual contiene un icono (un recuadro rojo que tiene incrustado una “x”) ubicado en el extremo derecho de la barra que permite ocultarla. En este prototipo se agregaron tres recuadros encimados en el extremo izquierdo, los cuales indican la selección de los componentes gráficos. El recuadro del fondo sirve para indicar el color seleccionado, el recuadro que se encuentra en medio representa el grosor de línea seleccionado y por último el recuadro de encima indica la figura seleccionada.

La barra de colaboradores se aprecia en la parte inferior de la ventana, la cual contiene únicamente los nombres de los colaboradores de forma lineal, con el fin de ahorrar espacio. Finalmente el área de dibujo utiliza el espacio restante que deja la barra de herramientas y la barra de colaboradores.

En la Figura 4.8.b se presenta una variante de la ventana principal, el cambio radica principalmente en la ubicación de la barra de colaboradores. La barra de herramientas

se ubica del lado izquierdo de la ventana en forma vertical. En la parte superior de la barra de herramientas se observa el icono que permite ocultarla. Cuando la barra de herramientas está oculta aparece un triángulo del lado izquierdo superior, el cual permite mostrarla. La presencia o la ausencia de las herramientas en la ventana no interfiere con el área de dibujo, i.e., el usuario continua dibujando sin importar si las herramientas están presentes o no.

En la ventana secundaria se muestra la meta-interfaz de redistribución, la cual contiene los interactores a migrar (barra de herramientas y barra de colaboradores) entre una PDA y una PC, esta ventana es similar a la mostrada en la Figura 4.7.c.

Selección de la interfaz de usuario para la PDA

La selección de la interfaz de usuario para la PDA no puede sólo basarse la adecuada organización y apariencia de sus componentes gráficos, sino también en su funcionalidad, ya que el espacio de la pantalla de estos dispositivos es muy reducido en comparación con el de la PC. La interfaz de usuario se seleccionó midiendo el número clicks que se necesitan para lograr un objetivo.

Suponga que un usuario se encuentra en la ventana que muestra el área de dibujo y que necesita pintar en tono azul, ¿cuántas veces se da clic para seleccionar el color azul y pintar sobre el área de dibujo?.

Tabla 4.2: Opciones elegidas para pintar sobre el área de dibujo con el color azul

Pasos realizados	Primer prototipo	Segundo prototipo	Tercer prototipo	Cuarto prototipo	Quinto Prototipo
1	Menú	Menú	Menú	Menú	icono de herramientas
2	Otras opciones	Otras opciones	Opcion colores	Opciones	color azul
3	color azul	menú color	color azul	color azul	
4	aceptar	color azul	aceptar	aceptar	
5		aceptar			

Un usuario que utiliza el primer prototipo necesita abrir el menú (primer clic). Una vez que está abierto él selecciona **otras opciones** (segundo clic) para que aparezca la ventana con los interactores restantes (barra de herramientas, meta-interfaz de redistribución y barra de colaboradores). A continuación, el usuario debe elegir el color azul (tercer clic).

Finalmente, él opta por la opción **aceptar** (cuarto clic). Cuando el usuario selecciona el botón **aceptar** regresa a la ventana principal, estableciendo como valores actuales los valores seleccionados. En total el usuario dio cuatro clicks para seleccionar el color azul y regresar al área de dibujo. Si el usuario también decidiera elegir otra figura antes de regresar al área de dibujo, el total de clic aumentaría en uno.

En el caso del segundo prototipo, el usuario empieza por abrir el menú (primer clic) y posteriormente elige **otras opciones** (segundo clic), la cual muestra todos los componentes gráficos en forma de menú desplegable. Por lo tanto, el usuario despliega el menú **colores** (tercer clic). Una vez que él tiene a la vista los colores, selecciona el color azul (cuarto clic) y finalmente selecciona la opción **aceptar** (quinto clic) para regresar al área de dibujo. En este prototipo se requieren cinco clicks para cumplir con el objetivo marcado. En este caso no importa que este prototipo tenga variantes, puesto que la variación no se aplicó a la barra de herramientas sino a la barra de colaboradores. Si el usuario también eligiera seleccionar otra figura antes de regresar a la ventana principal, el total de clicks se vería incrementado en dos.

Un usuario emplea el tercer prototipo, así que él abre el menú (primer clic) y selecciona la opción **colores** (segundo clic). Como resultado aparecen pequeños recuadros que representan los colores. De este modo, él selecciona el color azul (tercer clic) y en seguida elige la opción **aceptar** (cuarto clic), tanto para establecer el color actual como para regresar al área de dibujo. Si el usuario eligiera además del color la forma, él necesitaría hacer todo el proceso descrito en el análisis de este prototipo, i.e., el usuario ocuparía ocho clicks para seleccionar el color y la forma.

En el caso del cuarto prototipo, el usuario selecciona el menú (primer clic), en este aparece la opción de **Distribuir, opciones y colaboradores**. él selecciona **opciones** (segundo clic), en seguida aparece la barra de herramientas completa. De esta manera él elige el color azul (tercer clic), finalmente el usuario regresa al área de dibujo a través de la opción **aceptar** (cuarto clic). Contando el número de clic se obtiene cuatro. Si el usuario eligiera además del color la forma, el total de clic se vería incrementado en uno.

Un usuario que utiliza el quinto prototipo despliega la barra de herramientas por medio del icono de herramientas (primer clic). Una vez que se muestran los componentes de la barra de herramientas, él selecciona el color azul (segundo clic). En este caso, no es necesario cerrar la barra de herramientas para empezar a dibujar. Así, que el total de clicks requeridos para pintar con el color azul son dos. En el caso que el usuario necesitara seleccionar no sólo el color, sino también la forma, el total de clic aumentaría en uno.

Se puede concluir que el quinto prototipo es funcional, ya que requiere menos cantidad de clicks para seleccionar un componente de la barra de herramientas. Además este prototipo presenta en todo momento los colaboradores conectados. Otra característica a favor es que el usuario no tiene que cambiar de ventana para saber qué componentes gráficos de la barra de herramientas han sido seleccionados. Mediante este análisis se justifica porque el quinto prototipo es el prototipo seleccionado.

4.3 Implementación de los módulos más relevantes

El paquete J2SE proporciona facilidades para el desarrollo de interfaces de usuario. Así que la implementación del pizarrón compartido para la PC no es tan relevante en comparación con la implementación del pizarrón compartido para la PDA.

4.3.1 Interactores en el área de despliegue de la PDA

En la sección 2 se presentaron varias interfaces de usuario, las cuales fueron creadas en el emulador proporcionado por J2ME. De las interfaces de usuario mostradas se seleccionó la más adecuada, ahora resta por describir la implementación de esta interfaz en la PDA o en el teléfono inteligente.

Ubicación de los interactores en el despliegue de la PDA

Como se comentó al inicio de este capítulo la primera PDA con Windows al alcance fue el modelo HP IPAQ 6945. El tamaño del área de despliegue (i.e., área permitida en la programación de un sistema) que permite manipular este dispositivo es de 240 píxeles de ancho por 188 píxeles de largo. Sin embargo, J2ME permite agrandar un poco más el área de despliegue, ya que elimina temporalmente la barra inferior de la PDA, la cual muestra el icono del teclado, la opción `menú` y en ocasiones la opción `salir`.

Por medio de opción de J2ME, el área de despliegue se agranda a 240 píxeles de ancho por 214 píxeles de largo. Aunque las proporciones son adecuadas para la interfaz de usuario del pizarrón compartido, cuando se intentó dibujar sobre estas nuevas medidas, se observó que algunos de estos píxeles no son táctiles, i.e., J2ME no puede leer sobre esas coordenadas, pero si puede escribir sobre ellas. Finalmente, que el área táctil queda de 240 píxeles de ancho por 195 píxeles de largo (i.e., $214-19=195$). Comparando estas dimensiones con las dimensiones proporcionadas por omisión (240 píxeles de ancho por 188 píxeles de largo) es conveniente optar por las primeras.

En la Figura 4.9, se observa la ubicación de los interactores utilizando las dimensiones de 240 píxeles de ancho por 214 píxeles de largo. El área no táctil se puede aprovechar colocando la barra de colaboradores, puesto que su objetivo únicamente es mostrar la presencia de los colaboradores en vez de interactuar con ellos.

Una vez que se encontró la mejor dimensión del área táctil, se prosiguió a organizar los interactores restantes (i.e., la barra de herramientas y el área de dibujo). La barra de herramientas puede ubicarse de forma vertical, puesto que existe mayor espacio en el ancho del área disponible. La dimensión de cada componente gráfico de la barra de herramientas es de 15 píxeles de ancho por 15 píxeles de largo, ya que estas dimensiones



Figura 4.9: Análisis del tamaño de la PDA 6945

son suficientes para permitir al usuario la selección de un componente gráfico.

En base al tamaño de los componentes gráficos se puede calcular el tamaño de dos opciones disponibles para la barra de herramientas:

1. La utilización de una sola columna dejando 5 píxeles entre cada fila, implica que la barra de herramientas sea de 15 píxeles de ancho por 360 píxeles de largo. Esta opción no es adecuada puesto que el tamaño de la barra es más grande que las dimensiones físicas de la PDA. Además se requiere una barra de navegación para que el usuario pueda desplazarse a lo largo de la barra de herramientas.
2. La colocación de los componentes en dos columnas, dejando 4 píxeles entre las columnas y 5 píxeles entre las filas, requiere una dimensión de 34 píxeles de ancho por 180 píxeles de largo para la barra de herramientas. Esta opción es conveniente puesto que el usuario tendría a la mano la barra completa sin tener que recorrerla para elegir algún componente gráfico.

La barra de herramientas tiene 34 píxeles de ancho y necesita, al menos, 180 píxeles de largo. El largo disponible de despliegue es de 195 píxeles, así que las dimensiones establecidas para la barra de desplazamiento son de 34 píxeles de ancho por 195 píxeles de largo.

Por último, el área de dibujo utiliza el espacio restante, el cual es de 200 píxeles de ancho por 192 píxeles de largo. El cálculo de este interactor se obtuvo de la siguiente forma:

$$\text{Ancho}_{(\text{areaDeDibujo})} = \text{ancho}_{(\text{areaDespliegue})} - \text{ancho}_{(\text{barraHerramientas})} = 240 - 34 = 206 \text{ píxeles}$$

$$\text{Largo}_{(\text{areaDeDibujo})} = \text{largo}_{(\text{areaDespliegue})} - \text{largo}_{(\text{barraColaboradores})} = 214 - 19 = 195 \text{ píxeles}$$

En el caso de que barra de herramientas esté presente, el área de dibujo tiene disponible 206 píxeles de ancho por 195 píxeles de largo. Cuando la barra de herramientas está oculta, el área de dibujo puede usar 240 píxeles de ancho por 195 píxeles de largo.

Ubicación de los interactores en el teléfono inteligente

Después de haber estado trabajando y haciendo pruebas con la PDA HP IPAQ 6945, se obtuvo el teléfono inteligente IPaq *Business Navigator* 610. El tamaño del área de despliegue que proporciona este teléfono es de 240 píxeles de ancho por 269 píxeles de largo. Si se suprime el menú, el área de despliegue queda de 240 píxeles de ancho por 360 píxeles de largo. Estas últimas dimensiones son más grandes, pero al momento de tratar de usar toda el área de despliegue se observaron algunos problemas respecto al largo, ya que a partir del píxel 270 no puede ser manipulado por el programador. Así que las dimensiones disponibles son de 240 píxeles de ancho por 269 píxeles de largo.

En la figura 4.10, se observan las medidas de la barra de herramientas, el área de dibujo y la barra de colaboradores.

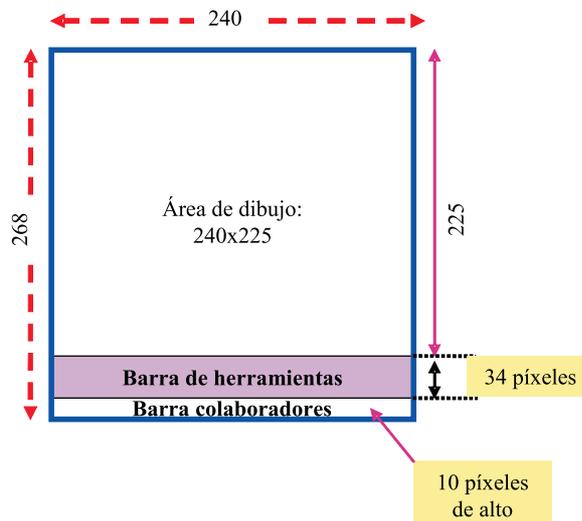


Figura 4.10: Distribución de interactores para teléfono inteligente

La barra de herramientas es colocada en la parte inferior del área de despliegue, ya que esta tiene más cantidad de píxeles a lo largo que a lo ancho. La barra de herramientas tiene las mismas dimensiones que la PDA (HP IPAQ 6945) sólo que invertidas, i.e., en vez de 34 píxeles de ancho por 195 píxeles de largo, ahora la barra de herramientas dispone de todo el ancho del área de despliegue (240 píxeles) por de alto 34 píxeles de largo.

La barra de colaboradores está localizada debajo de la barra de herramientas. Las dimensiones de la barra de herramientas son 240 píxeles de ancho por 10 píxeles de largo.

El área de dibujo ocupa el lugar no utilizado por los interactores anteriores. Cuando el menú está presente, las dimensiones del área son de 240 píxeles de ancho por 224 píxeles de largo, ya que el ancho del área de despliegue no fue modificado. En cambio, el alto del área de dibujo se calculó con una simple resta:

$$\text{Largo}_{(\text{areaDeDibujo})} = \text{largo}_{(\text{areaDespliegue})} - \text{largo}_{(\text{barraColaboradores})} - \text{largo}_{\text{barraHerramientas}} =$$

269 - 10 - 34= 225 píxeles

Cuando la barra de herramientas está ausente, el área de dibujo es de 240 píxeles de ancho por 250 de largo. El largo de esta área de dibujo se dejó en 250 píxeles para poder implementar la barra de desplazamiento de dicha área (cf. sección 4.3.2).

4.3.2 área de dibujo

En el diseño del pizarrón compartido, se había acordado en agrandar el tamaño del área de dibujo de la PDA, con la finalidad de manipular las mismas dimensiones del área de dibujo en la PDA y en la PC. En consecuencia, que el área de dibujo requiere una barra de desplazamiento para recorrer toda el área de despliegue, además de un *buffer* (memoria intermedia) que representará el área de dibujo aumentada.

Podría pensarse que la implementación de las barras de desplazamiento es una tarea fácil que simplemente requiere la inclusión del correspondiente componente gráfico creado por alguna clase predefinida de J2ME. Sin embargo, este paquete no ofrece primitivas para manipular barras de desplazamiento.

Las dimensiones del área de dibujo fueron calculadas para la PDA y para el teléfono inteligente. Las dimensiones del área de dibujo en la PDA son: ancho 206 píxeles y largo 195 píxeles. En el caso del teléfono inteligente, las dimensiones son: 240 píxeles de ancho por 250 de largo.

Barra de desplazamiento implementada en la PDA

Dado el tamaño reducido del área de dibujo se optó por implementar cuatro barras de desplazamiento, las cuales están representadas por una región específica en cada extremo del área de dibujo. Esta propuesta de la barra de desplazamiento deja atrás la idea tradicional de usar dos barras de desplazamiento (una barra ubicada a lo largo de la ventana del lado derecho y otra barra colocada a lo ancho de la ventana en la parte inferior, ambas barras tiene un par de flechas a los extremos que permitan recorrer el área).

La finalidad de usar las barras de desplazamiento propuestas, es permitir al usuario deslice el lápiz por alguno de los cuatro extremos del área de dibujo, el área de dibujo se recorra cierta cantidad de píxeles. Además, cuando se requiera, esta área puede ser utilizada para pintar sobre ella. Suponga que un usuario se encuentre dibujando una línea, mientras la dibuja desliza el lápiz por un extremo del área de dibujo, lo cual indica al pizarrón compartido dos acciones: 1) dibujar y 2)recorrer el área de dibujo.

Para poder implementar la barra de herramientas, primero se verificó que la PDA usada soportara el método encargado de leer las coordenadas cuando se desliza el lápiz. Después de realizar las pruebas correspondientes, se comprobó que la PDA no soporta el método encargado de leer las coordenadas cuando se desliza el lápiz, así que el área de la barra de desplazamiento funciona cuando se coloca el lápiz exactamente en la región reservada

para dicha barra.

La barra de desplazamiento del lado izquierdo abarca cuatro píxeles de ancho, con el objetivo de dejar un espacio considerable al área de dibujo y a la barra de herramientas. Por otra parte, la barra de desplazamiento de la derecha utiliza dos píxeles de ancho. Sin embargo, ambas barras emplean todo el largo del área de dibujo. El espacio reservado para estas dos barras no puede ser utilizado para pintar, esto implica que se reduce el ancho del área de dibujo a 200 píxeles, como se indica a continuación:

$$\text{Ancho}_{(\text{areaDeDibujo})} = \text{ancho}_{(\text{barraDesplazamientoIzquierda})} - \text{ancho}_{(\text{barraDesplazamientoDerecha})} = 206 - 4 - 2 = 200 \text{ píxeles}$$

La barra de desplazamiento superior abarca el ancho del área de dibujo y de largo un píxel. La barra de desplazamiento inferior también usa el ancho del área de dibujo y emplea dos píxeles de largo. El espacio reservado para ambas barras de desplazamiento tampoco puede ser usado para dibujar, por tal motivo el largo del área de dibujo se reduce a 192 píxeles de ancho, la resta usada en la siguiente:

$$\text{Largo}_{(\text{areaDeDibujo})} = \text{largo}_{(\text{barraDesplazamientoSuperior})} - \text{largo}_{(\text{barraDesplazamientoInferior})} = 195 - 1 - 2 = 192 \text{ píxeles}$$

Cuando la barra de herramientas está presente las dimensiones del área de dibujo son de 200 píxeles de ancho (ancho mínimo) por 192 píxeles de largo.

Cuando la barra de herramientas no está presente, las dimensiones del área de dibujo son de 240 píxeles de ancho (ancho máximo) y 192 píxeles de largo.

El ancho del área de dibujo varía cuando la barra de herramientas esta presente o ausente. Así que desde este momento se necesita las barras de desplazamiento para recorrer el ancho máximo del área de dibujo (240 píxeles), cuando esté presente la barra de herramientas (ancho del área de dibujo 200 píxeles) debe visualizarle el espacio disponible cuando no se encuentran (240 píxeles).

El área de dibujo se divide en segmentos como se observa en la figura 4.11. El recuadro formado con la línea continua representa el área de dibujo cuando está ausente la barra de herramientas. El ancho del área de dibujo se divide en seis columnas, donde cada columna es de 40 píxeles. El largo del área de dibujo se divide en seis filas de 32 píxeles cada una. Así, cuando la barra de herramientas está presente, sólo se necesita hacer un salto de 40 píxeles de ancho para visualizar el resto del área de dibujo.

La división de área de dibujo en segmentos fue la última tarea de implementación que se realizó con esta PDA, debido a que se adquirió el teléfono inteligente mencionado.

Barra de desplazamiento implementada en el teléfono inteligente

La primera tarea que se llevó a cabo en este nuevo dispositivo fue verificar que soportara el método encargado de leer las coordenadas cuando se desliza el lápiz. Después de realizar

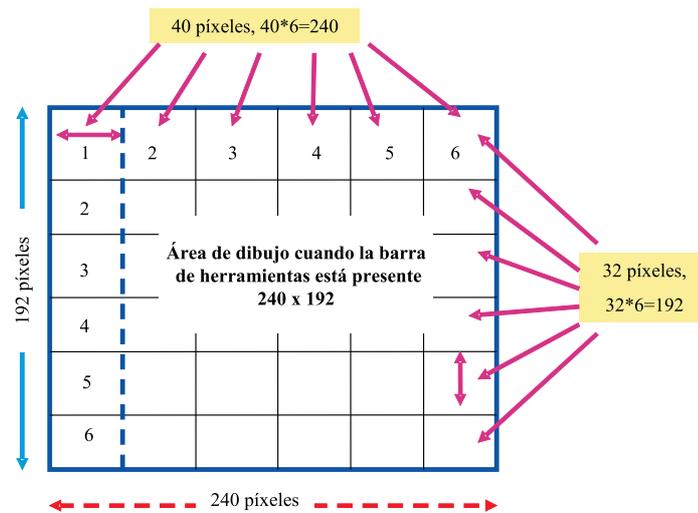


Figura 4.11: Análisis de división del área de despliegue

la prueba, se comprobó que sí soporta el método mencionado. En consecuencia las barras de desplazamiento pueden funcionar cuando se deslice el lápiz sobre ellas, sin necesidad de reducir el área de dibujo.

El ancho del área de dibujo no varía aún cuando la barra de herramientas está ausente o presente, sin embargo el largo si cambia. El largo del área de dibujo es de 225 píxeles cuando la barra de herramientas está presente, pero cuando está ausente es de 250 píxeles.

En la figura 4.12 se observa que el área de dibujo se dividió en seis columnas de 40 píxeles cada una y en ocho filas de 25 píxeles cada una. Cuando la barra de herramientas está presente estos segmentos permiten que sólo se requiera recorrer una vez la ventana de visualización para ver los restantes 25 píxeles ocultos por la barra de herramientas. Cuando la barra de herramientas está ausente, las filas de la Figura 4.12 cambian de 8 a 4 filas de 50 píxeles cada una.

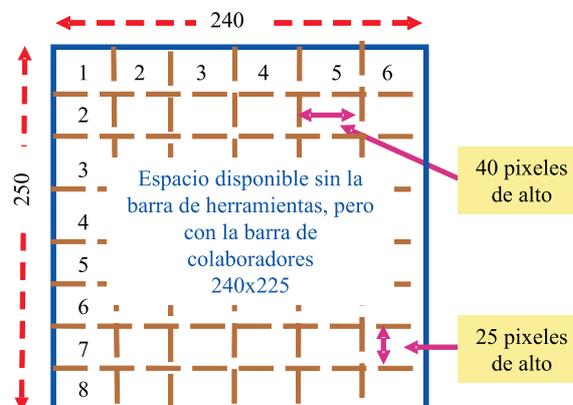


Figura 4.12: División del área de dibujo en el teléfono inteligente

El teléfono inteligente y la PC utilizan 360 píxeles de ancho para el área de dibujo no visible (se llamará de esta forma al área de dibujo, la cual es más grande que el área de despliegue). El ancho del área de dibujo no visible se formó al agregar tres columnas de 40 píxeles (120 píxeles) a la Figura 4.12. El alto del área de dibujo imaginario es de 350 píxeles, esta medida se obtuvo al incrementar en la parte inferior de la Figura 4.12 cuatro filas de 25 píxeles cada una.

En la Figura 4.13 se muestra el área de dibujo no visible. El área gris representa las filas y las columnas agregadas. El tamaño de esta área de dibujo no visible es de 350 píxeles de ancho por 360 píxeles de largo.

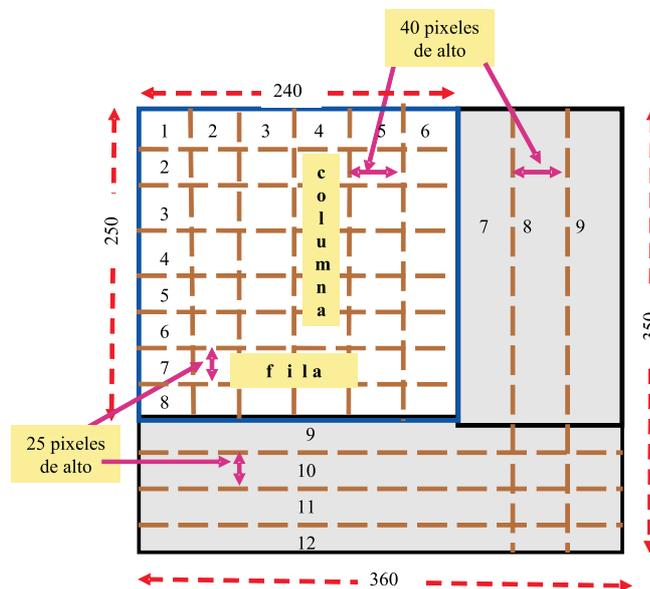


Figura 4.13: área de despliegue que utiliza la PC y la PDA

Si la barra de herramientas está presente, se recorre el ancho del área de dibujo no visible al pasar tres veces el lápiz sobre en la parte derecha de la ventana, mientras que para recorrer el largo del área de dibujo no visible se necesita que el usuario pase cinco veces el lápiz cerca de la barra de herramientas.

En el caso de que la barra de herramientas esté ausente, cambia el número de veces que un usuario debe pasar el lápiz por encima de la barra de colaboradores para visualizar la parte inferior del área de dibujo no visible, ya que las filas son de 50 píxeles. Un usuario necesita sólo dos veces deslizar el lápiz por encima del área de colaboradores.

Suponga que la barra de herramientas está presente y que un usuario visualiza, en el área de despliegue el recuadro blanco de la Figura 4.13 excepto la fila 8 (i.e., el colaborador observa de la columna 1 a la 6 y de la fila 1 a la 7). Si el colaborador quiere ver la columna 8, sólo necesita hacer dos saltos hacia la derecha, i.e., deslizar dos veces el lápiz sobre el área destinada para la barra de desplazamiento.

Para manipular el área de dibujo no visible se necesita un *buffer*, el cual se implementó usando un vector de: 126,000 píxeles (360 pÃxeles de ancho * 350 píxeles de ancho=126,000). El *buffer* fue creado usando un arreglo de enteros.

El procedimiento 1 muestra el funcionamiento de la barra de herramientas, dicho procedimiento se agrega al método que lee las coordenadas al deslizar el lápiz sobre la ventana. El procedimiento recibe como entrada: 1) la coordenada $p(x, y)$ por donde se desliza el lápiz, 2) el número de veces que se ha recorrido horizontalmente el área de dibujo no visible (*saltoHorizontal*), 3) el número de veces que se ha recorrido verticalmente el área de dibujo no visible (*saltoVertical*) y 4) un indicador para conocer si la barra de herramientas está presente o ausente (*BarraHerramientas*).

Procedimiento 1 Desplazar área de dibujo

Entrada: $p(x, y)$, *saltoHorizontal*, *saltoVertical* y *BarraHerramientas*

Salida: *saltoHorizontal* *saltoVertical*

- 1: **si** $x = 4$ **y** *saltoHorizontal* > 0 **entonces**
 - 2: *saltoHorizontal* \leftarrow *saltoHorizontal* $- 1$
 - 3: **si** $x = 223$ **y** *saltoHorizontal* < 3 **entonces**
 - 4: *saltoHorizontal* \leftarrow *saltoHorizontal* $+ 1$
 - 5: **si** $y = 4$ **y** *saltoVertical* > 0 **entonces**
 - 6: *saltoVertical* \leftarrow *saltoVertical* $- 1$
 - 7: **si** *BarraHerramientas* está presente **entonces**
 - 8: *saltoVertical* \leftarrow *saltoVertical* $- 1$
 - 9: **si** *BarraHerramientas* está presente **y** $y = 222$ **y** *saltoVertical* < 5 **entonces**
 - 10: *saltoVertical* \leftarrow *saltoVertical* $+ 1$
 - 11: **si** *BarraHerramientas* está ausente **y** $y = 247$ **y** *saltoVertical* < 4 **entonces**
 - 12: *saltoVertical* \leftarrow *saltoVertical* $+ 2$
-

En la línea 1 indica que el colaborador requiere recorrer el área de dibujo no visible a la izquierda, por lo cual se debe verificar que previamente el colaborador haya hecho un recorrido a la derecha. En caso afirmativo, se decrementa en uno el contador *saltoHorizontal*.

La línea 3 significa que el colaborador quiere recorrer el área de dibujo hacia la derecha, así que se requiere que no se haya agotado el número de veces permitidos. Si ocurre lo anterior se incrementa en uno el contador *saltoHorizontal*.

La línea 6 indica el usuario quiere recorrer hacia arriba el área de dibujo, así que se debe verificar que haya recorrido el área de dibujo hacia abajo. En caso de ocurrir lo anterior, se debe decrementar en uno *saltoVertical*. Además, si la barra de herramientas está presente, se decrementa nuevamente en uno *saltoVertical*.

Las líneas 9 y 11 sirven para recorrer hacia abajo el área de dibujo dependiendo la presencia o ausencia de la barra de herramientas. *SaltoVertical* se incrementa en uno cuando

está presente o en dos si está ausente.

Figuras

Un inconveniente que surgió al crear el *buffer* para manipular un área de despliegue mayor al que proporciona el dispositivo, fue la necesidad de crear un método para cada figura y de implementar el grosor de línea. También fue necesario implementar un método por cada letra del alfabeto y cada número del 0 al 9, para brindarle al colaborador la opción de escribir en la PDA o teléfono inteligente.

Considerando que se debía implementar los métodos para las figuras, letras y números, se optó por implementar únicamente las figuras. Así, cuando el pizarrón compartido está completo en el teléfono inteligente, no se encuentra habilitada la opción de escribir (A); pero cuando se redistribuye el pizarrón compartido la opción está activa, ya que el teléfono inteligente sólo contiene la barra de herramienta y/o la barra de colaboradores.

El procedimiento 2 permite dibujar una figura con su correspondiente color y grosor. Los parámetros de entrada son: el punto p_A , el punto p_B , el *color* de la figura, el *grosor* de la *figura* y el tipo de *figura*.

Procedimiento 2 Pintar figura

Entrada: $p_A(xA, yA), p_B(xB, yB), color, figura$ y *grosor*

Salida:

```
1:  $i \leftarrow 0$ 
2: establecer color
3: while grosor  $\neq i$  do
4:   si la figura es una línea entonces
5:     si  $x_A = x_B$  entonces
6:        $x_A \leftarrow x_A + i$  y  $x_B \leftarrow x_B + i$ 
7:     else
8:        $y_A \leftarrow y_A + i$  y  $y_B \leftarrow y_B + i$ 
9:     dibujar línea usando  $p_A$  y  $p_B$ 
10:  si figura es un círculo entonces
11:    dibujar círculo usando  $x_A, y_A, x_B, y_B$ 
12:    dibujar círculo usando  $x_A + i, y_A + i, x_B, y_B$ 
13:    dibujar círculo usando  $x_A, y_A, x_B + i, y_B + i$ 
14:    dibujar círculo usando  $x_A + i, y_A + i, x_B + i, y_B + i$ 
15:  else
16:    dibujar rectángulo usando  $x_A + i, y_A + i, x_B - i, y_B - i$ 
17:   $i \leftarrow i + 1$ 
```

En la línea 3 se establece la condición de pintar la figura el número de veces que indique el *grosor*.

En la línea 4 se verifica si la figura a dibujar es una línea. En caso de cumplir este requisito se incrementa x_A y x_B cuando la línea esté completamente vertical. El incremento en el eje x sirve para dibujar líneas consecutivas a la misma altura. Se incrementa y_A y y_B cuando la línea está inclinada. El incremento sirve para dibujar una línea debajo de otra. Por último, se llama al método *dibujar línea* con la coordenada p_A y p_B .

En la línea 10 se verifica si la figura corresponde a un círculo. En caso de ser así se dibujan cuatro círculos. Las dimensiones varían por un píxel en p_A y en p_B , lo cual se debe a que se observó existían regiones con píxeles muy separados cuando se dibujaba un círculo dentro de otro.

Cuando la figura no es una línea ni un círculo entonces se deduce que es un rectángulo. En la línea 16 se dibuja un rectángulo, incrementando en uno p_A y decrementando en uno p_B , con el fin de crear cuadrados dentro de otros para representar el grosor de línea.

4.3.3 Comunicación entre la PDA y la PC

La comunicación entre el teléfono inteligente y la PC fue por medio de *sockets*. Un *socket* para Java es un punto de comunicación por el cual un proceso puede emitir o recibir información [Hughes et al., 1999].

En el procedimiento 3 se observa los pasos necesarios para ejecutar un servidor, el cual ha sido implementado mediante un hilo (*thread*).

Procedimiento 3 Pasos para ejecutar un servidor en la PC

- 1: Iniciar servidor
 - 2: Esperar conexiones
 - 3: Enviar datos
 - 4: Recibir datos
 - 5: Cerrar conexión
-

En la línea 1 se inicializa el servidor especificando el puerto a usar (puerto 5000).

En la línea 3 el servidor espera conexiones de los clientes, además de crear e inicializar dos canales, uno para el envío y el otro para la recepción de datos.

La línea 4 es un método que sirve para enviar datos, los cuales son colocados en el canal de salida. La línea 5 es un método para recibir datos.

Finalmente en la línea 6 se cierra la conexión.

En el procedimiento 4 se observan los pasos a seguir para ejecutar un cliente en el teléfono inteligente. En la línea 1 se abre un *socket* para establecer la comunicación con el servidor especificando el puerto del servidor. También se crean los canales de comunicación, un canal de entrada y otro de salida. Finalmente, en la línea 2 se ejecuta un hilo

que se encargará de enviar datos al servidor.

Procedimiento 4 Pasos para ejecutar un cliente en un PDA

- 1: Iniciar cliente
 - 2: Enviar datos
 - 3: Recibir datos
 - 4: Cerrar conexión
-

En la línea 3 se ejecuta un hilo que será responsable de recibir los datos enviados por el servidor.

En la línea 4 se detienen los hilos encargados de recibir y enviar datos.

4.4 Pruebas y resultados

Las pruebas referentes a la plasticidad del pizarrón compartido se realizaron dentro de la red privada *groupware2* con una dirección IP fija en cada dispositivo (PDA y PC).

La colaboradora Kim ejecutó el pizarrón compartido en la PC. La primer ventana que despliega el pizarrón compartido es la de autenticación (cf. Figura 4.14.a). Kim ingresa su nombre de usuario y contraseña. En respuesta, el pizarrón compartido despliega otra ventana para indicar si la validación fue exitosa (cf. Figura 4.14.b) o no.

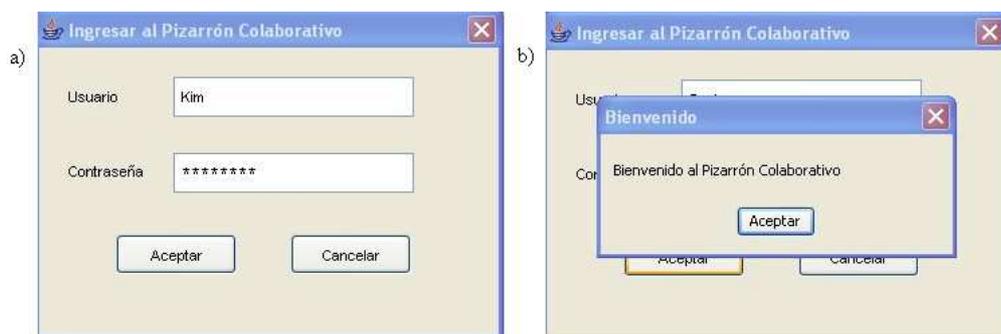


Figura 4.14: La autenticación en la PC se compone de: a) la ventana de autenticación y b) la ventana de usuario validado

Después de una validación exitosa, el pizarrón compartido despliega la ventana mostrada en la Figura 4.15, la cual esta dividida de la siguiente manera: una parte de la barra de herramientas está localizada en la parte izquierda y la otra esta posicionada en la parte interior de la ventana.

Se puede observar que la última vez Kim había seleccionado el círculo, la línea gruesa y el color verde. La barra de colaboradores indica que ella es la única colaboradora en la

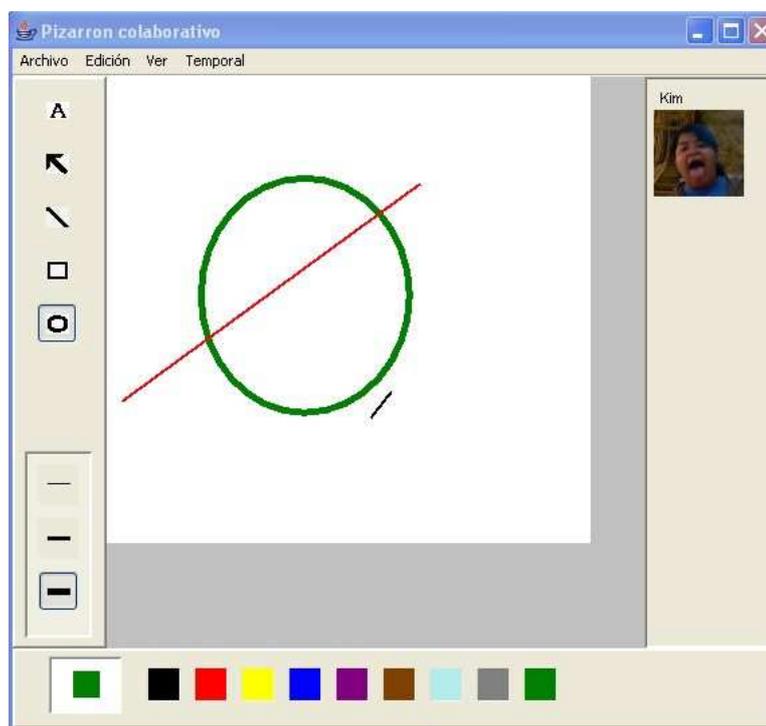


Figura 4.15: Ventana principal del pizarrón compartido en la PC

sesión.

Posteriormente ingresa la colaboradora Gaby desde el teléfono inteligente. La primera ventana que aparece se muestra en la Figura 4.16, la cual indica que la conexión con el servidor se realizó sin problemas. La siguiente ventana es la de autenticación (cf. Figura 4.17). Después de una validación exitosa del nombre de usuario y de la contraseña se despliega la ventana que contiene tres interactores (cf. Figura 4.18).

En la parte inferior de la ventana de la Figura 4.18.a se observan dos nombres, los cuales son mostrados en el orden que los colaboradores ingresaron a la sesión. El primero nombre corresponde a la colaboradora Kim y el otro se refiere a la colaboradora Gaby. Arriba de la barra de colaboradores se encuentra la barra de herramientas. En la parte izquierda de la barra de herramientas se observa que Gaby ha seleccionado la figura línea, color rojo y la línea gruesa.

La Figura 4.18.b corresponde al pizarrón compartido ejecutado en la PC, en donde se muestra la presencia de la colaboradora Gaby por medio de su foto y nombre.

Cuando Kim dibuja una línea desde la instancia del pizarrón compartido en su PC, Gaby puede percibir los cambios de manera automática. Después de realizar un bosquejo entre ambas colaboradoras, Gaby decide salir de la sesión, por tal motivo Kim no detecta la foto de Gaby. Tiempo después Kim sale de la sesión en curso.

Finalmente el colaborador G ingresa al pizarrón compartido, primero lo hace desde la

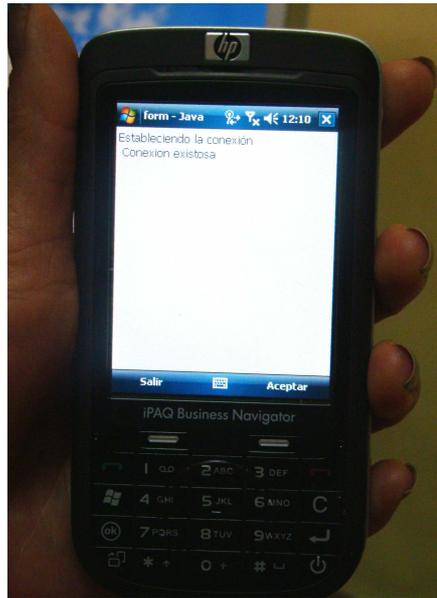


Figura 4.16: El pizarrón compartido en la PDA informa que estableció conexión con el servidor

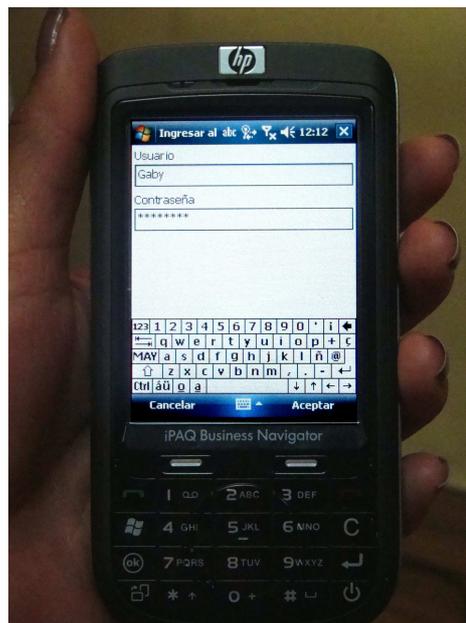


Figura 4.17: Autenticación del usuario en el teléfono inteligente

PC y posteriormente desde la PDA. Cuando el pizarrón compartido de la PC detecta al colaborador en dispositivos diferentes, despliega la ventana de la meta-interfaz de redistribución (cf. Figura 4.20). En dicha ventana el colaborador selecciona la barra de herramientas y la barra de colaboradores para redistribuirla hacia el dispositivo móvil (teléfono inteligente).

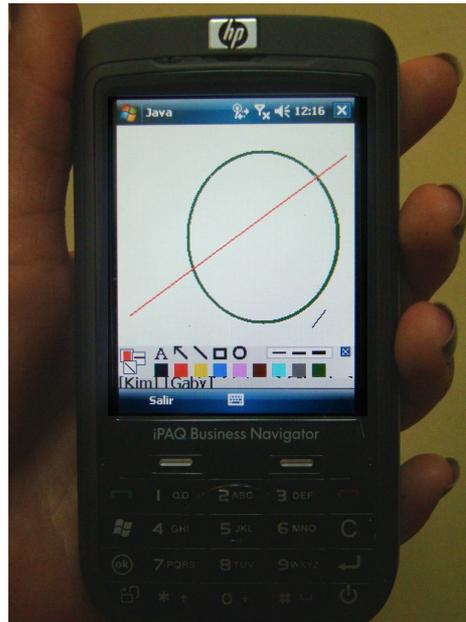


Figura 4.18: Ventana principal del pizarrón compartido en el teléfono inteligente

En la figura 4.21 se observa el área de dibujo en la PC, en tanto que en la PDA se observa la barra de herramientas. En la parte superior de la ventana del teléfono inteligente se observan los componentes gráficos de la barra de herramientas más grandes. Debajo de la barra de herramientas, se encuentra la barra de colaboradores con el nombre y la foto del único colaborador conectado.

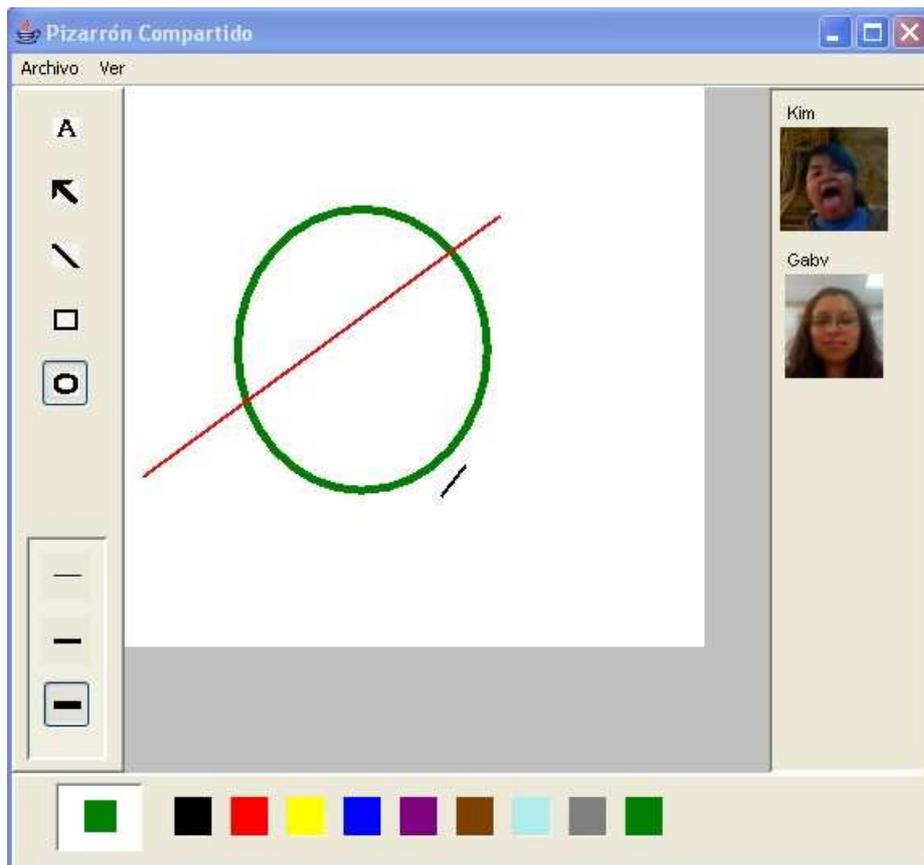


Figura 4.19: La barra de colaboradores muestra la llegada de la colaboradora Gaby en la PC

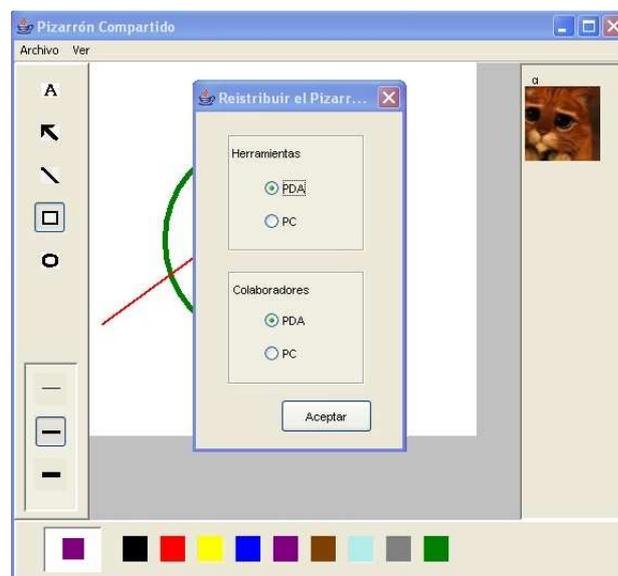


Figura 4.20: Ventana de la meta-interfaz de redistribución en la PC

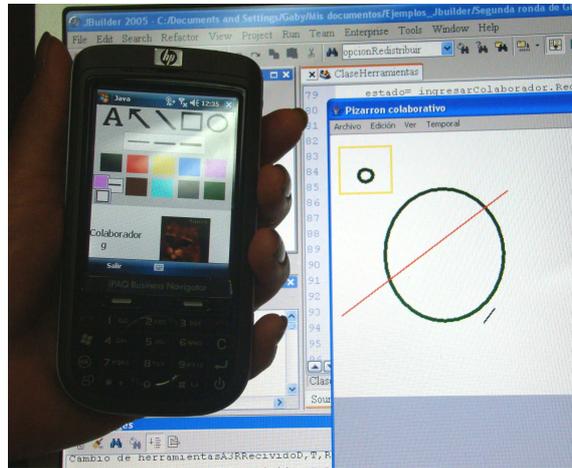


Figura 4.21: La PDA despliega la barra de herramientas y la barra de colaboradores, mientras que la PC muestra el área de dibujo

Capítulo 5

Conclusiones y trabajo a futuro

Los usuarios de algunos sistemas mono-usuario han comenzado a beneficiarse con la propiedad de plasticidad de interfaces de usuario, la cual les permite accederlos desde diversos dispositivos y cambiar los dispositivos de entrada tradicionales (e.g., ratón) por pinceles o incluso sus propios dedos, en otras ocasiones, los sistemas permiten al usuario elegir los componentes a visualizar en dos dispositivos. La plasticidad de interfaces de usuario ha sido explorada en los sistemas mono-usuario, en cambio en los sistemas colaborativos se tiene poca investigación (cf. sección 5.1). La poca exploración de la plasticidad de interfaces de usuario ha generado el tema de investigación, por lo cual se creó un pizarrón compartido (cf. sección 5.2) para una PDA y una PC. El pizarrón compartido tiene varias limitaciones, las cuales pueden contemplarse como trabajo a futuro (cf. sección 5.3).

5.1 Resumen de la problemática

En la actualidad existe una variedad de dispositivos que tienen diferentes características en hardware como en software, e.g., una PDA, una PC o un pizarrón interactivo. La heterogeneidad de estos dispositivos ha obligado a los desarrolladores de software a adaptar sus sistemas a esta gama de dispositivos.

La plasticidad de interfaces gráficas es un tema que se ha comenzado a explorar en los sistemas mono-usuario, e.g., el sitio Web Sedan-Bouillon que permite la redistribución de su interfaz gráfica al reconocer a un usuario en dos dispositivos diferentes o el sistema FlexClock cambia de interfaz gráfica al detectar la conexión o desconexión del teclado físico.

En el caso de los sistemas multi-usuario la plasticidad de interfaces gráficas no ha sido prácticamente explorada, e.g., el sistema ConnecTable permite compartir las imágenes al momento de unir las mesas de trabajo y Roomware permite visualizar la misma interfaz gráfica usando una mesa de trabajo, un sillón o en un pizarrón formado por tres pantallas. Así que la plasticidad de interfaces gráficas en sistemas multi-usuario deja un campo abierto a la investigación, lo cual ha dado origen a explorar esta combinación.

El tema de investigación de esta tesis es la adaptación de la interfaz de usuario a un sistema colaborativo, tomando en cuenta el contexto de uso (en específico la plataforma y el usuario). Para llevar a cabo este tema de investigación, se creó un pizarrón compartido que es ejecutado en una PC y en una PDA.

El pizarrón compartido tiene las funcionalidades básicas de una aplicación mono-usuario de dibujo, e.g., figuras pre-diseñadas (e.g., líneas y rectángulos) y a mano alzada, las cuales han sido suficientemente exploradas. Por otra parte el pizarrón compartido es dotado de características colaborativas, e.g., manipulación del área de dibujo por parte de los colaboradores y la provisión de información de conciencia de grupo.

5.2 Conclusiones

El pizarrón compartido aborda algunas características de la plasticidad de interfaces gráficas, algunas de ellas son: a) permitir la ejecución del sistema en una PC y una PDA (adaptación a la plataforma) a partir de interfaces gráficas pre-diseñadas (producción de las interfaces), b) permitir la división de la interfaz gráfica en dos dispositivos (redistribución de centralizada a distribuida) al detectar al mismo usuario en ambos dispositivos (adaptación al usuario), c) permitir la participación del usuario y del sistema en el proceso de adaptación plástica (localización a la adaptación) y finalmente d) conservar el estado del sistema antes y después de adaptación plástica (grano de reanudación).

Se utilizó la PDA con Windows, la cual sólo puede ejecutar programas realizados en Java por el paquete J2ME. Por otra parte se decidió utilizar Java para la PC por varias razones, e.g., Swing una clase del paquete J2SE, la cual implementa el modelo arquitectural usado en el diseño del pizarrón compartido.

Se analizaron varias interfaces de usuario, posteriormente seleccionó la más adecuada. En el caso de la PC no existe gran problema gracias a las características físicas del dispositivo y a las bondades que ofrece el paquete de J2SE. Sin embargo se analizaron tres propuestas de interfaces de usuario. En el caso de la PDA es más evidente la necesidad de crear varios prototipos de la interfaz de usuario, ya que el espacio de la pantalla es muy limitado. La mayoría de los prototipos desarrollados para la PDA usan los componentes gráficos que contiene la clase Screen, perteneciente al paquete J2ME, tal clase impide manipular la posición de los componentes gráficos. En consecuencia se tuvieron que crear componentes gráficos propios.

La interfaz gráfica elegida para la PDA no utiliza los componentes gráficos del paquete J2ME, así que sus componentes gráficos fueron creados. La primera necesidad surgida por la creación de componentes fue elegir las dimensiones de los interactores (los interactores contienen a los componentes gráficos). Se calculó el tamaño de los interactores que utilizan menos espacio (barra de herramientas y la barra de colaboradores), posteriormente el espacio restante fue asignado al interactor del área de dibujo.

El área de dibujo usada en la PDA y la PC es del mismo tamaño, para que las figuras que sean dibujadas en la PDA conserven sus dimensiones al momento de ser visualizadas en la PC y viceversa. i.e., las figuras que sean dibujadas en la PC conserven su tamaño original al momento de ser visualizadas en la PDA. Se creó un buffer para manipular el área de dibujo que presentan ambos dispositivos.

Al momento de crear el área de dibujo más grande que el área de despliegue de la PDA, surgió la necesidad de crear una barra de desplazamiento para recorrer el área de dibujo. Otra necesidad fue implementar los métodos encargados de pintar un círculo, una línea o un rectángulo, además del método encargado de hacer el grosor de línea.

La comunicación entre la PC y la PDA se realizó por medio de sockets, en donde la PC desempeña la función de servidor y la PDA tiene la función de cliente.

El pizarrón compartido de la PC permite guardar en archivos la selección de la barra de herramientas por cada usuario, asimismo almacena las figuras creadas en la sesión. En cambio la PDA no puede manipular archivos, así que este dispositivo requiere de un servidor para guardar la información mencionada.

5.3 Trabajo a futuro

En este apartado se presentan algunas mejoras para el pizarrón compartido:

- Crear un método por cada letra del alfabeto para permitir escribir texto sobre la PDA.
- Implementar un módulo para identificar el orden en que fueron realizadas las figuras.
- Hacer un módulo de concurrencia para identificar el orden en que ingresan los colaboradores.
- Implementar la comunicación peer to peer para que la comunicación sea directa entre los dispositivos.
- Implementar el uso de voz para que un colaborador indique al pizarrón compartido la creación de un círculo por medio del lenguaje común.
- Crear un módulo que permita la redistribución de centralizada a distribuida o de distribuida a distribuida, para que un colaborador pueda unir la interfaz de usuario, también para que él pueda dividir la interfaz en varias ocasiones.
- Implementar la opción borrar, cambiar color o grosor, hacer o rehacer una acción.

Apéndice A

Instalaciones y configuraciones

A.1 OpenEmbedded

OpenEmbedded es un entorno de desarrollo de compilación para generar paquetes (compilador cruzado) de Software, en específico para tarjetas embebidas.

La instalación de OpenEmbedded se ejecuto de dos formas diferentes, una desde la página Web de Linux Familiar y la otra desde la página Web de OpenEmbedded.

A.1.1 Primera instalación

La primera instalación y configuración que se presenta es la que muestra la pagina Web de Linux Familiar [[URL: Familiar Build Setup](#)]. Los pasos son los siguientes:

1. Antes de la instalación de OpenEmbedded se hacen otras instalaciones, las cuales se mencionan a continuación:
 - **git** es un sistema de control de versiones;
 - **zenity** es una herramienta que permite mostrar cajas de dialogo GTK (herramienta para crear interfaces de usuario [[URL: GTK](#)]) [[URL: Zenity](#)];
 - **arm-Linux-gcc** el cual es un compilador cruzado.
2. Se descargo el directorio de familiar-build con la siguiente instrucción:
git clone http://familiar.handhelds.org/git/familiar-build.git.
3. Se ingreso al directorio Familiar-build y se ejecuto la instrucción:
sh setup/build-env.sh
Con esta instrucción aparece una secuencia de ventanas para realizar la configuración, en la ventana se especificar el PDA usado, el entorno grafico, entro otras cosas.
4. Se establecieron las siguientes variables de entorno en el archivo *.shrc.*:

- a la variable `path` se le agregó la siguiente línea:
`/home/gabriela/familiar-build/bitbake/bin`
- se creó la variable de entorno de la siguiente forma:
`BBPATH "/home/gabriela/familiar-build/build-h3900-opie:
/home/gabriela/familiar-build/org.handhelds.familiar"`.

Los parámetros establecidos en las variables anteriores se extrajeron del archivo `env.sh` generado en la instalación.

A.1.2 Segunda instalación

La segunda instalación y configuración que se presenta es la que muestra la página Web de OpenEmbedded [[URL: OpenEmbedded](#)].

Esta instalación es un poco más extensa y se resume a continuación la instalación. Al final de la instalación se decidió descartarla puesto que no estamos seguros que funcione sobre Linux-familiar.0.8.4. Los pasos que se siguieron son los siguientes:

1. se descargó `bitbake-1.8.10.tar.gz` y se descomprimió;
2. se descargó la base de datos de OpenEmbedded con la instrucción:
`wget http://openembedded.org/snapshots/OE.mtn.bz2`
y se descomprimió con: `bunzip OE.mtn.bz2`,
3. se obtiene la rama necesaria para Linux-Familiar de la siguiente forma: `mtn -db=OE.mtn checkout -branch=org.openembedded.oz354fam083`,
4. se configura el archivo "local-conf" y se establece las variables de entorno. Para configurar el archivo se hace lo siguiente:

- desde el directorio en donde se ha estado trabajando realizar la siguiente instrucción:

```
cp org.openembedded.dev/conf/local.conf.sample build/conf/local.conf
```

Posteriormente, se ingresa al archivo "local-conf" y se establecen los parámetros necesarios para poder trabajar con la PDA, e.g., se especifica la PDA y la versión de Linux.

5. las variables de entorno modificadas son las siguientes:
 - `PATH "/stuff/bitbake/bin:"PATH BBPATH "/stuff/build:/stuff/org.openembedded.oz354fam083:" BBPATH [11, 12]`
 - `BBFILES :="/home/gabriela/openembedded/openembedded/org.openembedded/org.openembedded.0z354fam083/packages/*/*.bb"`

- MACHINE = 3900
- DISTRO = “familiar-0.8.4”
- KERNEL_VERSION = “2.4.19”

A.2 Instalación de Opie

Opie es un entorno gráfico de código abierto. Este entorno está diseñado para ser ejecutado en dispositivos pequeños que utilicen sistema operativo Linux [[URL: Opie](#)].

La instalación de Opie no se completo con éxito, para su instalación se baso en la página Web de Opie [[URL: Opie](#)]. Los pasos son los siguientes:

1. se instaló Qt-embedded, posteriormente se descomprime,
2. se crea la variable de entorno QTDIR. En esta variable de entorno se le especifica la ruta en donde se descomprimió QT-embedded.
3. se ejecuta el comando:
`./configure -q config qpe,`
posteriormente se ejecuta `make`. Después de el `make` salen dos errores, uno de ellos marca la ausencia de dos archivos. Para resolver el problema, se modificó la ruta del archivo;
4. por ultimo se ejecuta el comando `make clane`, el proceso que se ejecuta después de ejecutar el comando `make clane` pide configurar varias cosas.

A.3 Instalación de Kaffe

Kaffe es una máquina virtual que permite la ejecución de código Java [[URL: Kaffe](#)]. Las instalaciones realizadas se hicieron siguiendo las instrucciones sugeridas por la página Web de Kaffe [[URL: Kaffe](#)].

Instalación de Kaffe en una PC Para su instalación se usó `kaffe-1.1.8.tar.gz` puesto que no presentó problemas de incompatibilidad al momento de instalar. Después de desempaquetarlo, se ingresa al directorio creado al momento de desempaquetar kaffe, posteriormente se ejecuto el siguiente comando: `./configure`. Finalmente, se ejecuta el comando `make install`.

Instalación de kaffe en una PDA Se descarga la misma versión de kaffe en la PDA pero en la memoria externa, puesto que la interna no tiene el suficiente espacio para descomprimirla, se descomprime y se ejecuta el comando `./configure`. Después de ejecutar dicho comando se despliega el siguiente error: `Configure: error: install -sh c.`

A.4 Instalación de JWT

J2ME es una plataforma que permite desarrollar aplicaciones para dispositivos móviles o embebidos. La instalación se realizó siguiendo los pasos mencionados en la página Web de Sun [[URL: Sun JWT](#)].

Esta instalación fue de las más fáciles hasta ahora, los pasos realizados se muestran a continuación:

Se descargó y ejecutó el siguiente archivo:
sun_java_wireless_toolkit-2.5.2-windows.exe. En esta instalación se aceptó la licencia y prácticamente fue dar clic en la opción siguiente, de las ventanas presentadas.

Referencias

- [Baecker et al., 1994] R. M. Baecker, G. Glass and A. Mitchell and I. Posner, *SASSE: the collaborative editor*, CHI '94: Conference companion on Human factors in computing systems, ACM, pp. 459-462, Boston, Massachusetts (USA), 1994.
- [Balme et al.,2005] L. Balme, A. Demeure, G. Calvary y J. Coutaz, *Sedan-Bouillon: A Plastic Web Site*, Plastic Services for Mobile Devices (PSMD), Workshop held in conjunction with Interact'05, pp. 1-3, Rome, 2005 .
- [Bastien and Scapin, 1993] J. M. C. Bastien y D. L. Scapin, *Preliminary findings on the effectiveness of ergonomic criteria for the evaluation of human-computer interfaces*, CHI 1993: INTERACT 1993 and CHI 1993 conference companion on Human factors in computing systems, ACM, pp. 187-188, New York, NY (USA), 1993.
- [Berti y Paternò, 2005] S. Berti y F. Paternò, *Migratory MultiModal interfaces in MultiDevice environments*, ICMI, pp. 92-99, 2005.
- [Burridge] R. Burridge, *Java Shared Data Toolkit User Guide*, Sun Microsystems, Java-Soft Division, Available from <http://java.sun.com>.
- [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
- [Calvary et al., 2001] G. Calvary, J. Coutaz and D. Thevenin, *A Unifying Reference Framework for the Development of Plastic User Interfaces*, Lecture Notes in Computer Science, Vol. 2254, pp. 173-194, 2001.
- [Crease, 2001] M. Crease, *A Toolkit of Resource-Sensitive, Multimodal Widgets*, PhD Thesis, Department of Computing Science, University of Glasgow, 2001.

- [Dieterich et al, 1993] H. Dieterich and U. Malinowski and T. Kühme and M. Schneider-Hufschmidt, *State of the art in adaptive user interfaces*, In Schneider-Hufschmidt M., Kuehme T., Malinowski U. (eds), Adaptive User Interfaces, Elsevier, pp. 13-48, 1993.
- [Draper et al., 1986] S. W. Draper, D. A. Norman and C. Lewis, *User Centered System Design: New Perspectives on Human-Computer Interaction*, In D. A. Norman and S. W. Draper, Erlbaum, pp. 1-6, Hillsdale, NJ, 1986.
- [Duke and Harrison, 1993] D. J. Duke and M. D. Harrison, *Abstract interaction objects*, Computer Graphics Forum, Vol 12, No. 3, pp.25-36, 1993.
- [Eckstein et al., 1998] R. Eckstein, M. Loy, and D. Wood. *Java Swing. O'Reilly and Associates*, 1998.
- [Ellis et al., 1991] C. A. Ellis, S. J. Gibbs and G. Rein, *Groupware: some issues and experiences*, ACM, pp. 39-58, New York, NY (USA), 1991.
- [Ellis and Wainer, 1994] C. A. Ellis and J. Wainer, *A Conceptual Model of Groupware*, In CSCW, pp. 79-88, 1994.
- [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.
- [Gutwin et al., 1996] C. Gutwin, M. Roseman, and S. Greenberg, *A Usability Study of Awareness Widgets in a Shared Workspace Groupware System*, In Proceedings of ACM Conference on Supported Cooperative Work- ACM CSCW'96., ACM Press, pp. 258-267, Boston, Mass., 1996.
- [Gutwin and Greenberg, 1999] C. Gutwin and S. Greenberg *The Effects of Workspace Awareness Support on the Usability of Real-Time Distributed Groupware*, ACM Transactions on Computer-Human Interaction (TOCHI), Vol. 6, No. 3, pp. 243-281, 1999.
- [Gutwin and Roseman, 1996] C. Gutwin and R. Roseman, *A Usability Study of Workspace Awareness Widgets*, In ACM Conference on Human Factors in Computing System, Companion Proceedings - ACM CHI'06, pp. 214-215, 1996.
- [Hughes et al., 1999] M. Hughes, M. Shoffner, D. Hamner, M. Winslow *Java Network Programming*, 2da (Ed.), pp. 256-261, Editorial Manning
- [Mitchell, 1996] A. Mitchell, *Communication and Shared Understanding in Collaborative Writing*, Science master thesis, Computer Science Department, University of Toronto, 1996.
- [Myers et al., 2000] B. Myers, S. E. Hudson and R. Pausch, *Past, present, and future of user interface software tools*, ACM Trans. Comput.-Hum. Interact., Vol. 7, No. 1, pp. 3-28, ACM Press, 2000.
- [McCall, 1977] J. McCall, *Factors in Software Quality*, General Electric Eds., 1977.

-
- [Phillips, 1999] W. G. Phillips, *Architecture for Synchronous Groupware*, Technical Report, Department of Computing and Information Science, pp. 11-24, Queen's University Kingston, Ontario, 1999.
- [Prante et al., 2004] T. Prante, N. A. Streitz and P. Tandler, *Roomware: computers disappear and interaction evolves*, Computer, Vol. 37, No. 12, pp. 47-54, 2004.
- [Rekimoto and Saitoh, 1999] J. Rekimoto and M. Saitoh, *Augmented Surfaces: A Spatially Continuous Workspace for Hybrid Computing Environments*, Proceedings of CHI'99, ACM Press, pp. 378-385, 1999.
- [Rekimoto, 1997] J. Rekimoto, *Pick-and-Drop: A Direct Manipulation Technique for Multiple Computer Environments*, ACM Symposium on User Interface Software and Technology, pp. 31-39, 1997.
- [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.
- [Roseman and Greenberg, 1996] M. Roseman and S. Greenberg, *Building real-time groupware with GroupKit, a groupware toolkit*, ACM Transactions on Computer-Human Interaction, Vol. 3, No. 1, pp. 66-106, 1996.
- [Ryokai et al., 2005] K. Ryokai, S. Marti and H. Ishii, *Designing the world as your palette*, CHI 2005: CHI 2005 extended abstracts on Human factors in computing systems, ACM, pp. 1037-1049, New York, NY (USA), 2005.
- [Sendín and Collazos, 2006] M. Sendín and C. A. Collazos, *Implicit Plasticity Framework: A Client-Side Generic Framework for Collaborative Activities.*, Y. A. Dimitriadis, I. Zigurs and E. Gómez-Sánchez (Eds), CRIWG, pp. 219-227, 2006.
- [Sendín and Lorés, 2004] M. Sendín and J. Lorés, *Plasticity in mobile devices: a dichotomic and semantic view*, Workshop Engineering Adaptive Web, pp. 58-67, 2004.
- [Sottet et al., 2006] , J.-S Sottet, G. Calvary, J.-M. Favre, J. Coutaz and A. Demeure, *Towards Mapping and Model Transformation for Consistency of Plastic User Interfaces*, Workshop on The Many Faces of Consistency in Cross-platform Design, ACM conf. on Computer Human Interaction, CHI 2006, Montréal, pp. 1-5, 2006.
- [Tandler et al., 2001] P. Tandler, T. Prante and C. Müller-Tomfelde, N. Streitz and R. Steinmetz, *Connectables: dynamic coupling of displays for the flexible creation of shared workspaces*, UIST 2001: Proceedings of the 14th annual ACM symposium on User interface software and technology, ACM, pp. 11-20, New York, NY (USA), 2001.
- [Thevenin, 2001] D. Thevenin, *L'adaptation en Interction Homme-Machine: le cas de la plasticité*, PhD thesis. Thèse de doctorat Informatique préparée au Laboratoire de Communication Langagière et Interaction Personne-Système (IMAG), Université Joseph Fourier, 2001.

[Thevenin and Coutaz, 1999] D. Thevenin and J. Coutaz, *Plasticity of User Interfaces: Framework and Research Agenda*, In Proceedings of Interact'99, vol. 1, Edinburgh: IFIP, IOS Press, pp. 110-117, 1999.

[Universidad de Malaga, 2003] Universidad de Malaga, *Java a Tope J2ME (Java 2 Micro Edition)*, pp 10-20, 2003.

Páginas Web consultadas

[URL: Bitbake] *Bitbake*, Berlios Developer, <http://developer.berlios.de/projects/bitbake>.

[URL: Familiar Build Setup] *Familiar Build Setup*, <http://familiar.handhelds.org/build-setup.html>.

[URL: GTK] The GTK + project, <http://www.gtk.org/>.

[URL: Kaffe] *Kaffe*, <http://www.kaffe.org/doc/kaffe/README>.

[URL: Linux Familiar] *Linux Familiar* <http://familiar.handhelds.org/build-setup.html>.

[URL: OpenEmbedded] Kooi K., *OpenEmbedded*, <http://www.openembedded.org/>.

[URL: Opie] *Opie*, <http://opie.handhelds.org/cgi-bin/moin.cgi/WhatIsOpie>.

[URL: Qtopia] *Qtopia*, Trolltech, <http://trolltech.com/products/qtopia/qtopia-overview>

[URL: Sun JWT] *Download Sun Java Wireless Toolkit for CLDC 2.5.2*
<https://sdlc2d.sun.com/ECom/EComActionServlet/DownloadPage:com.sun.sunit.sdlc.content.DownloadPageInfo;jsessionid=E92EDEB564315D3123779D921331B0B2;jsessionid=E92EDEB564315D3123779D921331B0B2>

[URL: Zenity] *Zenity*, <http://freshmeat.net/projects/zenity>