

# Sistema de remodelación semi-plástica adaptativa para interfaces de usuario colaborativas

## Introducción

Al interactuar con aplicaciones colaborativas los miembros del grupo pueden encontrarse físicamente separados, lo que, sin un soporte adecuado, impide informar a cada usuario las acciones que se realizan en el espacio compartido. Esta limitación sugiere desarrollar Interfaces de Usuario (IU's) que informen sobre la actividad de cada colaborador, el cual puede interactuar desde cualquier dispositivo de cómputo. Evidente, la IU de una aplicación colaborativa, no puede ser la misma sobre una pantalla grande y una pequeña de un dispositivo. En este artículo, esta problemática es abordada mediante el desarrollo de un sistema que detecta cambios en la plataforma y que remodela la IU de una aplicación colaborativa.

Los dispositivos de cómputo móviles que actualmente han aparecido en el mercado, cada vez cuentan con mayores capacidades de comunicación, procesamiento, memoria y almacenamiento, las cuales son suficientes para soportar algunas aplicaciones colaborativas. Estos avances tecnológicos permiten imaginar que los usuarios de una aplicación colaborativa pueden interactuar desde cualquier dispositivo de cómputo (fijo o móvil). Sin embargo, es evidente que la interfaz de usuario de una aplicación colaborativa no puede ser la misma sobre pantallas grandes y pequeñas.

Dadas las diferentes capacidades de los dispositivos de cómputo, se requiere que las interfaces de usuario de las aplicaciones se adapten a las características (e.g., tamaño de pantalla) de cada dispositivo.

Con el fin de satisfacer este requerimiento, la propiedad de plasticidad permite que la interfaz de usuario de una aplicación se adapte a cambios en el contexto de uso, preservando la usabilidad y continuidad de interacción de la aplicación. Sin embargo, dicha propiedad ha sido introducida a través de prototipos de laboratorio que revelan pocos detalles de los mecanismos que permiten llevar a cabo el proceso de adaptación plástica. Esta limitante impide tener soluciones que puedan reutilizarse en otros ámbitos. Además, cabe señalar que en el dominio de las aplicaciones colaborativas la propiedad de plasticidad ha sido pobremente abordada, a pesar de la necesidad inminente de dotar a dichas aplicaciones de la capacidad de adaptación a cambios contextuales.

Nuestra solución consiste en diseñar e implementar un sistema de remodelación semi-plástica (i.e., sensible a cambios en la plataforma) y adaptativa (i.e., el proceso de adaptación es realizado por el sistema sin intervención del usuario) para interfaces de usuario de aplicaciones colaborativas. Particularmente, se pretende diseñar un sistema que: (a) detecte las variaciones en el contexto de uso (principalmente en la plataforma), (b) identifique las soluciones candidatas para adaptar la interfaz de usuario, (c) seleccione una solución particular y (d) ejecute la nueva solución. En dicho diseño también se debe incluir, un soporte de comunicación que permitirá a los usuarios interactuar simultáneamente con la aplicación colaborativa desde dispositivos heterogéneos.

Este artículo está organizado de la siguiente manera. Después de dar una síntesis de los trabajos relacionados en la sección 2, se describe SISRESAIUC (Sistema de Remodelación Semi-plástica Adaptativa para Interfaces de Usuario Colaborativas) en la sección 3. En particular, se describe sus casos de uso, así como su funcionamiento general, y las arquitecturas de comunicación y de distribución de dicho sistema. Finalmente, en la sección 4 se presentan las conclusiones de este trabajo de investigación.

## Trabajos relacionados

En la presente sección se describen y analizan los trabajos relacionados con el sistema de adaptación propuesto en este artículo.

### Descripción de los trabajos relacionados

En el presente artículo se describen cinco aplicaciones representativas del estado del arte. Algunas de dichas aplicaciones presentan la propiedad de plasticidad, otras incluyen características de los sistemas colaborativos distribuidos o una combinación de ambos.

La aplicación de “Control de Calefacción” (Coutaz y Calvary, 2008) propuesta por FEC (*French Electricity Company*) se utiliza para consultar y modificar la temperatura de una habitación en particular. Dicha aplicación puede ser accedida por medio de diversos dispositivos de cómputo para poder controlar la calefacción: a) desde la casa mediante un dispositivo montado en la pared o una PDA conectada a una red inalámbrica, b) desde la oficina, a través de la Web, empleando una PC o c) desde cualquier lugar haciendo uso de un teléfono móvil o un reloj.

La aplicación *Flexclock* (Grolaux, Roy y Vanderdonckt, 2002) es un reemplazo del reloj del sistema UNIX, cuyo objetivo es mostrar la hora y la fecha en diferentes formatos de acuerdo al tamaño de la ventana. Estos formatos van desde una simple cadena *hh:mm:ss* hasta una vista más compleja compuesta de un reloj analógico, un reloj digital y un calendario mensual. *Flexclock* ofrece 17 posibles vistas, las cuales existen de manera simultánea i.e., desde la perspectiva de la aplicación no hay diferencia entre la vista visible y las vistas ocultas al usuario.

La aplicación *CICERO* (Ciavarella y Paternò, 2004), ofrece a los visitantes de un museo una visita guiada interactiva y un conjunto de juegos colaborativos que estimulan tanto la cooperación, como el aprendizaje de los visitantes. El escenario típico de uso consiste en que los usuarios se muevan libremente por el museo e interactúen a través de dispositivos móviles. Adicionalmente, *Cicero* (Bell, et al., 2006) (Danesh, et al., 2001) utiliza despliegues públicos para mostrar la localización del visitante y las diferentes maneras de explotar juegos individuales y colectivos.

*UbiDraw* (Grolaux, Roy y Vanderdonckt, 2001), proporciona cuatro componentes de dibujo: “Archivos” (*Fichiers*), “Dibujo” (*Dessin*), “Opciones” (*Options*) y “Retocar” (*Retoucher*) agrupados en una barra de menú. Cada componente ofrece una barra de herramientas que puede ser mostrada en diferentes lugares de la ventana principal de *UbiDraw*, dependiendo del tamaño de dicha ventana.

### Análisis comparativo

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. Las cinco aplicaciones analizadas se adaptan a la plataforma. Particularmente, *Flexclock* y *UbiDraw* pueden ser ejecutadas en diversos sistemas operativos (e.g., *Windows*, *Linux* y *Mac*), mientras que las aplicaciones de “Control de Calefacción” y *CICERO* son accesibles desde diversos dispositivos de cómputo (i.e., *PDAs*, despliegues públicos, *PCs*, *Pocket PCs* y teléfonos celulares).

Por otra parte, *Flexclock* y *UbiDraw* también se adaptan al usuario porque la remodelación de su interfaz gráfica depende de la tarea realizada por el usuario (i.e., la interfaz gráfica cambia de presentación cuando el usuario reajusta el tamaño de la ventana). En la documentación referente a las aplicaciones de “Control de Calefacción” y *CICERO* no se especifica qué o quién desempeña el proceso de adaptación plástica. En *Flexclock* y *UbiDraw* la adaptación es desempeñada por el sistema.

Todas las aplicaciones analizadas implementan la remodelación de tipo intra-modal, i.e., preserva la modalidad de gráfico a gráfico.

Las interfaces de usuario de las aplicaciones de “Control de Calefacción” y *CICERO* están prefabricadas para los diversos dispositivos de cómputo (i.e.,

no son generadas en tiempo de ejecución). *FlexClock* incorpora un conjunto de diecisiete interfaces de usuario previamente diseñadas, pero además genera interfaces de usuario en tiempo de ejecución, ya que aún cuando *Flexclock* elige una interfaz de usuario de las diecisiete posibles, esta aplicación puede modificar el tamaño y la alineación de sus componentes gráficos (e.g., agrandar el reloj analógico o centrar el reloj digital). Por su parte, *UbiDraw* genera sus interfaces de usuario en tiempo de ejecución, dependiendo del reajuste de la ventana principal.

## SISRESAIUC (Sistema de Remodelación Semi-plástica Adaptativa para Interfaces de Usuario Colaborativas)

En la presente sección se detalla un diseño orientado a objetos de SISRESAIUC (Sistema de Remodelación Semi-plástica Adaptativa para Interfaces de Usuario Colaborativas). Primeramente, se describen los casos de uso de dicho sistema (cf. Sub-sección 3.1). Después, se explica el funcionamiento general de

SISRESAIUC mediante su vista dinámica (cf. Sub-sección 3.2) así como sus principales componentes (cf. Sub-sección 3.3). Posteriormente, se presenta la arquitectura de comunicación de SISRESAIUC, la cual establece la forma de llevar a cabo el intercambio de mensajes cuando sus objetos interactúan en una red (cf. sección 3.3). Finalmente, se detalla la arquitectura de distribución de SISRESAIUC (cf. sección 3.4).

### Casos de uso de SISRESAIUC

Como se mencionó en la sección 1, el objetivo principal de este trabajo de investigación consiste en diseñar e implementar un sistema semi-plástico adaptativo, llamado SISRESAIUC (Sistema de Remodelación Semi-plástica Adaptativa para Interfaces de Usuario Colaborativas), el cual adapta la interfaz de usuario de una aplicación colaborativa a las características físicas de diversas plataformas, sin intervención humana. Para lograr dicho objetivo se identificaron los casos de uso que se describen a continuación (c.f., Figura 1).

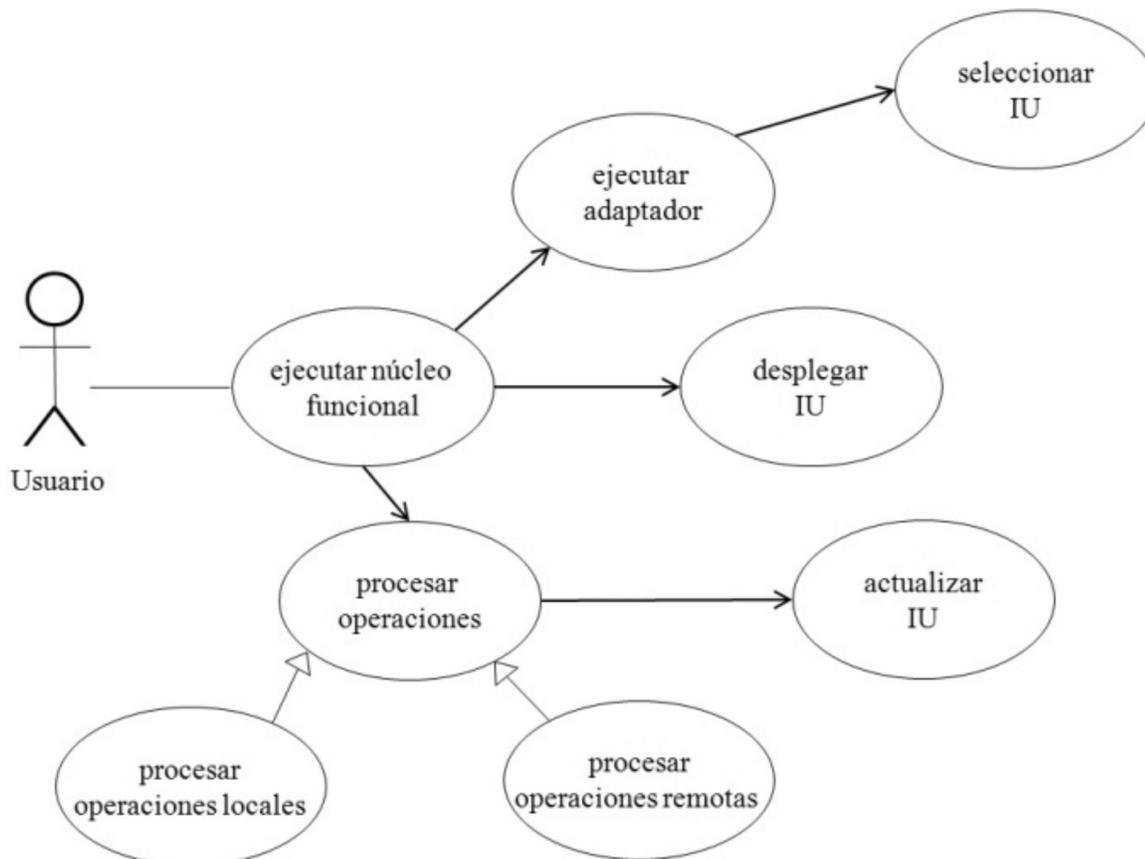


Figura 1: Casos de uso de SISRESAIUC

Se tiene un conjunto de usuarios, i.e., personas que, provistas de un dispositivo de cómputo (e.g., PC, PDA o laptop) desean emplear una aplicación colaborativa en un momento determinado y en espacios geográficos distintos. Como se mencionó antes, los usuarios emplean dispositivos de cómputo heterogéneos, por lo tanto surge la necesidad de adaptar el tamaño de la interfaz de usuario a cada uno de los dispositivos, con el propósito de que cada usuario interactúe adecuadamente con la aplicación colaborativa. Para lograr lo anterior, el usuario solicita **ejecutar el núcleo funcional** de la aplicación, el cual a su vez solicita **ejecutar adaptador** para emplear la opción de **seleccionar una interfaz de usuario**. Con base en esta selección, el núcleo funcional se da a la tarea de **desplegar una interfaz de usuario** adaptada al tamaño de pantalla del dispositivo.

Este caso de uso lo realizan todos y cada uno de los usuarios que desean interactuar con una aplicación colaborativa. De manera que, luego de que se despliega la interfaz de usuario adecuada, es necesario **actualizarla** cada que se produzca un evento relevante en el espacio de trabajo compartido, con la finalidad de que los usuarios estén informados de lo que acontece. El núcleo funcional es el encargado de **procesar operaciones**, las cuales se refieren a las acciones realizadas por el usuario **local** (aquel que ejecuta un núcleo funcional a partir de su propio dispositivo) y por los usuarios **remotos** (aquellos que interactúan a distancia con el usuario local mediante su propio dispositivo).

## Vista dinámica de SISRESAIUC

Con el propósito de identificar a los principales componentes de SISRESAIUC, en la presente sección se muestra la vista dinámica de SISRESAIUC por medio de un diagrama de colaboración.

### Vista dinámica

Para ilustrar la vista dinámica del SISRESAIUC se presenta la Figura 2, la cual muestra la colaboración entre cuatro usuarios que utilizan diferentes dispositivos de cómputo, e.g., *laptop*, *smartphone*, PC. Suponga que el usuario que emplea la *laptop* es el primero que ingresa su nombre en la interfaz de usuario de una aplicación colaborativa. En respuesta, el núcleo funcional envía el id de dicho usuario al intermediario, el cual solicita establecer una conexión (con el vigía).

Seguidamente, el vigía crea un hilo coordinador C1 que se encarga de atender a dicho intermediario.

Por cada usuario que ingresa a la aplicación colaborativa, e.g., *smartphone*, PC se lleva a cabo el proceso mencionado en el párrafo anterior. De esta forma, el vigía asigna los hilos coordinador C2, C3 y C4 para atender respectivamente a dichos usuarios.

Es necesario precisar que, aunque varios usuarios ingresen su nombre simultáneamente en la interfaz de usuario de una aplicación colaborativa, todos los intermediarios tendrán asignado un coordinador, ya que las solicitudes de conexión llegan a una cola de espera reservada por el vigía. Dicha cola es de tipo *FIFO*, i.e., el primero en entrar, es el primero en salir.

## Componentes de SISRESAIUC

Los componentes principales que integran a SISRESAIUC siguen una arquitectura cliente/servidor. La Figura 3 permite mostrar la distribución de los componentes según el modelo cliente-servidor:

1. El *software* del cliente (cf. Figura 3 #1) contiene las clases Núcleo funcional (desarrollado por los programadores de una aplicación colaborativa) e Intermediario (contiene los métodos que como parte de SISRESAIUC, se le proporciona a los programadores de una aplicación colaborativa), la cual forma parte del componente de soporte de colaboración; así mismo contiene el componente de adaptación plástica, i.e., la clase Adaptador (contiene los métodos que como parte de SISRESAIUC, se le proporciona a los programadores de una aplicación colaborativa) y un conjunto de interfaces de usuario prefabricadas (desarrolladas por los programadores de la aplicación colaborativa) para diversos dispositivos de cómputo de una aplicación colaborativa ; este *software* se ejecuta en el dispositivo de cómputo de cada usuario que participa en la aplicación colaborativa en cuestión;
  2. El *software* del servidor (cf. Figura 3 #2) contiene las clases Vigía y Coordinador; este *software* está hospedado en una PC que funge como servidor.
- A continuación se describen los componentes principales de SISRESAIUC: a) la adaptación plástica (cf. letra A de la Figura 3) y b) el soporte de colaboración (cf. Letra B de la Figura 3). En esta descripción se justifican las principales características de cada uno de estos componentes.

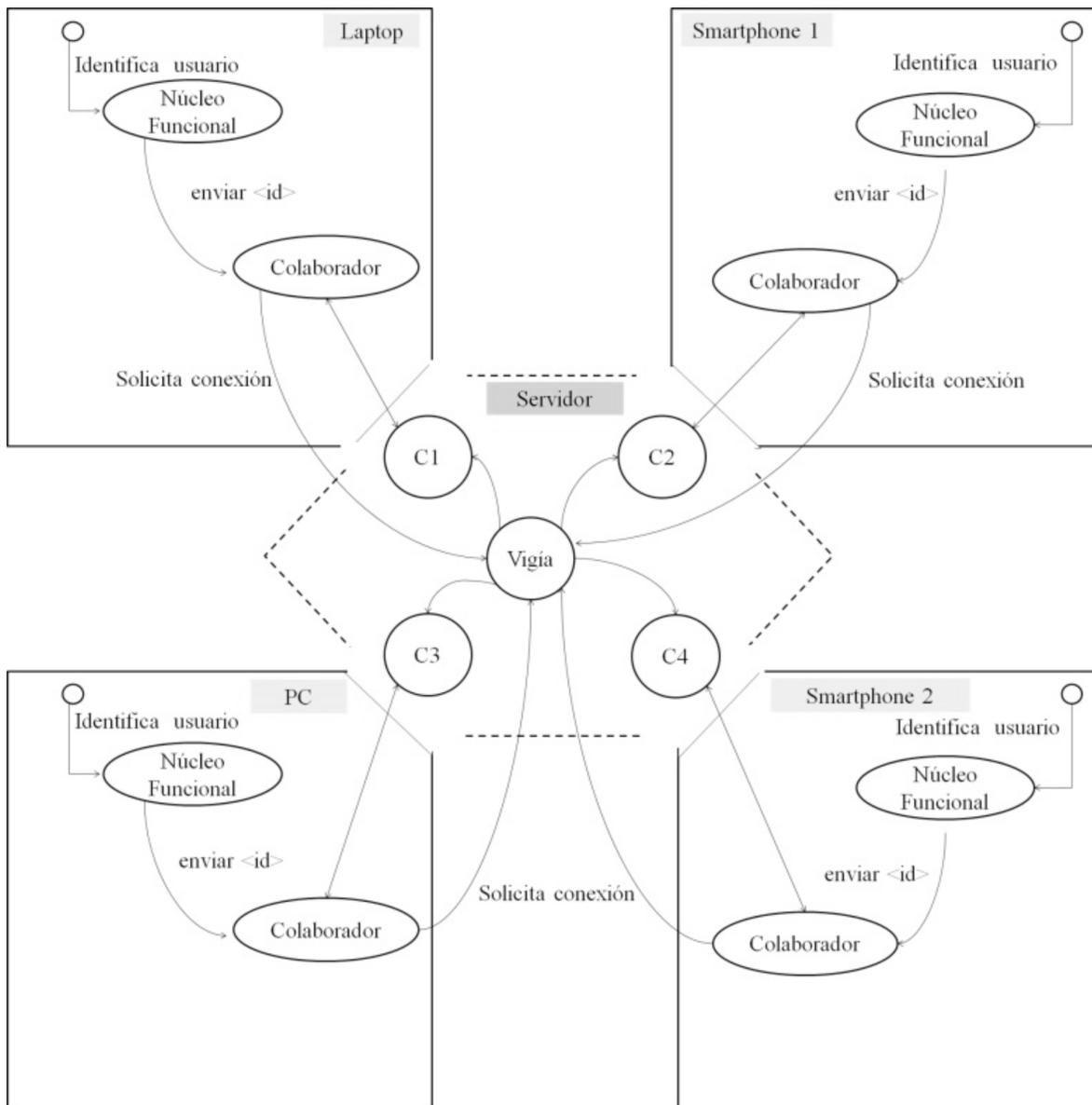


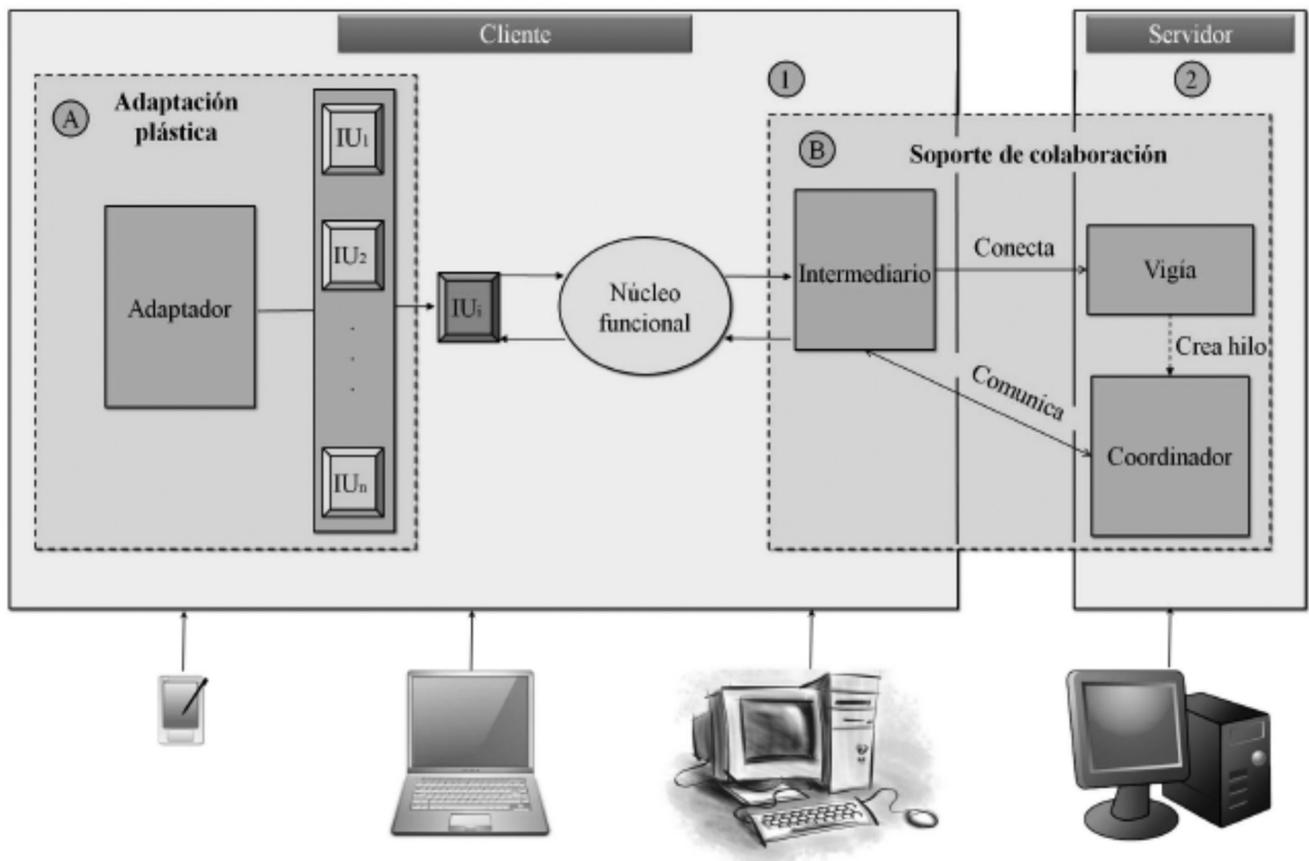
Figura 2: Diagrama de colaboración de SISRESAIUC

### Adaptación plástica

En el presente trabajo únicamente consideramos la adaptación de la interfaz de usuario de una aplicación colaborativa a diversas plataformas. Sin embargo, es necesario precisar que al ejecutarse dicha aplicación, la interfaz de usuario no varía su presentación durante una sesión colaborativa. Por lo tanto, la adaptación plástica se realiza mediante un algoritmo que consta de los siguientes cuatro pasos:

1. Detección del contexto de uso: obtiene las dimensiones (ancho y alto en píxeles) de la pantalla del dispositivo de cómputo en el que se ejecutará una aplicación colaborativa.

2. Identificación de las soluciones candidatas: identifica las interfaces de usuario de una aplicación colaborativa que son apropiadas al nuevo contexto de uso. Es evidente que las interfaces de usuario cuyas dimensiones sean mayores que las de la pantalla de un dispositivo de cómputo resultan inadecuadas (e.g.,  $IU_1, IU_2, \dots, IU_j$  de la Figura 4). En consecuencia, se consideran soluciones candidatas aquellas interfaces de usuario que tienen un ancho y un alto menores o iguales que los de la pantalla del dispositivo de cómputo. Es necesario verificar, mediante un algoritmo de búsqueda, que exista al menos una interfaz de usuario que tenga dichas características. En caso de que no exista di-



IU = Interfaz de Usuario

Figura 3: Diagrama de componentes de SISRESAIUC

cha interfaz de usuario, se desplegará un mensaje de error en la pantalla del dispositivo de cómputo.

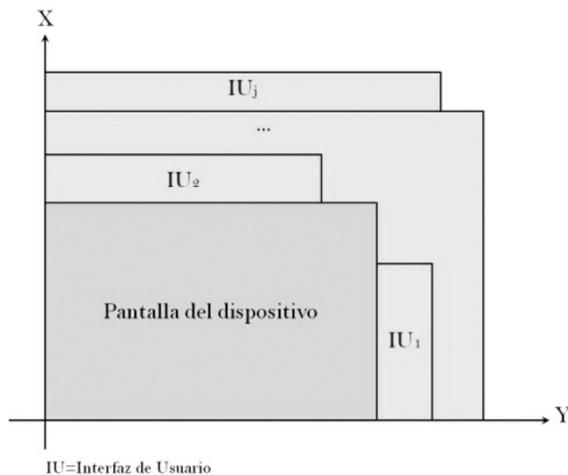


Figura 4: Interfaces de usuario inadecuadas para la pantalla de un dispositivo de cómputo

3. Selección de una solución particular: como varias soluciones candidatas pueden ser identificadas

(e.g.,  $IU_1, IU_2, \dots, IU_i$  de la Figura 5), esta etapa selecciona la interfaz de usuario más adecuada a desplegar. De todas las interfaces de usuario candidatas, se elige la que minimiza la distancia  $d$  en Norma-2 (La Norma-2 o norma euclidiana es una norma para medir distancias, la cual considera que la distancia más corta entre dos puntos es la recta que los une.) entre el ancho  $x$  y el alto  $y$  de la pantalla del dispositivo de cómputo y el ancho  $x_i$  y el alto  $y_i$  de la interfaz de usuario:

$$d = \min_{x,y} \sqrt{(x - x_i)^2 + (y - y_i)^2} \quad (1)$$

4. Ejecución de la nueva solución: despliega la interfaz de usuario más adecuada al nuevo contexto de uso. El algoritmo de adaptación plástica retorna la identificación que corresponde a la interfaz de usuario seleccionada. Posteriormente, dicha identificación permite que un objeto encargado de la gestión de gráficos muestre la interfaz de usuario correspondiente.

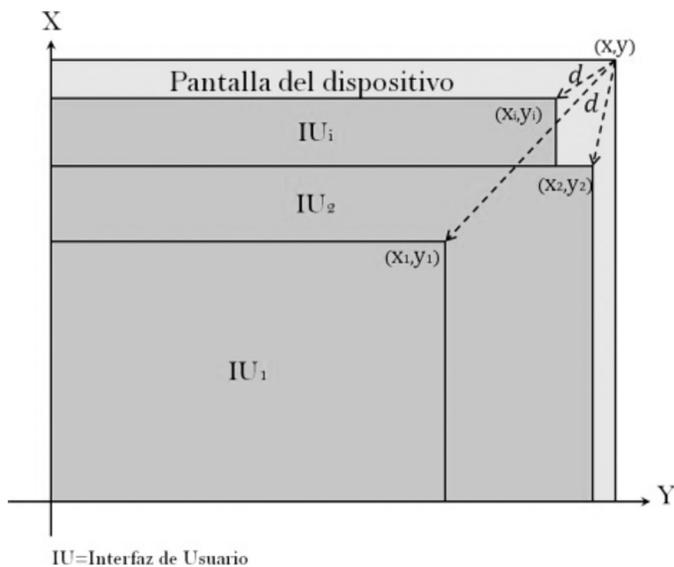


Figura 5: Interfaces de usuario candidatas para la pantalla de un dispositivo de cómputo

### Arquitectura de distribución de SISRESAIUC

Los componentes de un sistema distribuido (Ardaiz, et al., 2004) no sólo están separados lógicamente sino también físicamente, por lo que requieren mecanismos de comunicación para poder interactuar. Una arquitectura de distribución define la repartición de los objetos de un sistema entre los sitios implicados en un proceso de interacción. Las arquitecturas de distribución difieren principalmente en tres aspectos: 1) la representación de cada objeto compartido en los sitios participantes; 2) el número de instancias de cada clase que están presentes en el sistema y 3) la posible movilidad de cada objeto entre dichos sitios.

Como se mencionó en la sección 3.2, SISRESAIUC consiste de un conjunto de clases, las cuales pueden ser reutilizadas para el desarrollo de aplicaciones con un nivel básico de interacción. Una parte de dicho sistema está localizado en una PC, la cual desempeña el papel de servidor. Sin embargo, la mayor parte del sistema se encuentra en un directorio, el cual está hospedado en los diversos dispositivos de cómputo que utilizan los usuarios para interactuar en una sesión colaborativa. Dicho directorio contiene las siguientes clases:

1. Clase **Adaptador**: los métodos de esta clase tienen la finalidad de permitir el despliegue de la interfaz de usuario adecuada a un determinado dispositivo de cómputo.

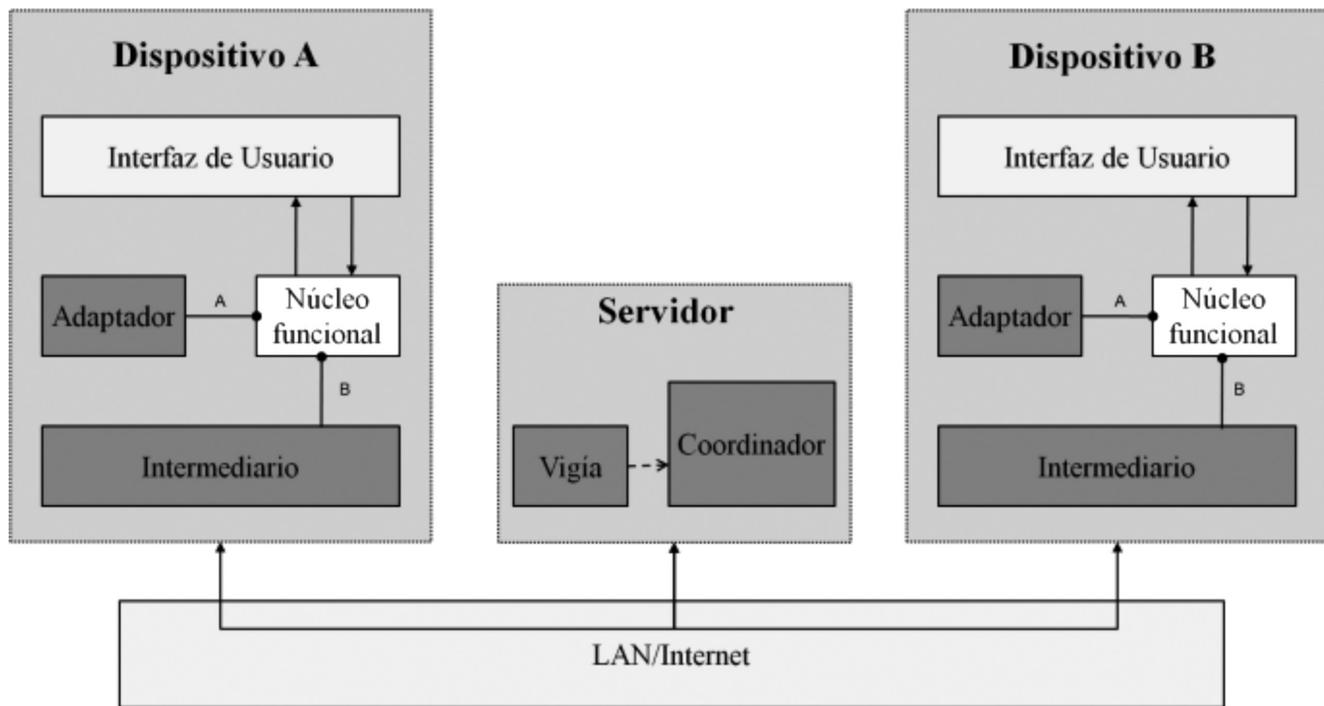
2. Clase **Intermediario**: los métodos de esta clase tienen el objetivo de proyectar la participación de un usuario en una aplicación colaborativa.

Métodos de la clase **Adaptador** (cf. letra A de la Figura 6):

- Método **detecta\_dispositivo()**: obtiene las dimensiones, i.e., ancho y alto en píxeles, de la pantalla de un dispositivo de cómputo.
- Método **selecciona\_interfaz(entero vector2D, entero x, entero y)**: devuelve el identificador (i.e., un número entero) de la interfaz de usuario que mejor se adapta a la pantalla de un dispositivo de cómputo. A este Método es necesario pasarle como parámetros: 1) un arreglo de dos dimensiones en donde se incluyan los tamaños (i.e., ancho y alto en píxeles) de todas las interfaces de usuario disponibles y 2) las dimensiones de la pantalla del dispositivo de cómputo.

Métodos de la clase **Intermediario** (cf. letra B de la Figura 6):

- Método **conectar (cadena nodo)**: permite la conexión con el servidor; el parámetro nodo indica la dirección IP del sitio en el que se ejecuta dicho servidor.
- Método **pasar\_idUsuario(cadena)**: se utiliza para que el núcleo funcional pase el identificador de un usuario a un objeto de la clase **Intermediario**. La estructura de dicho identificador se deja al criterio del programador de aplicaciones.
- Método **obtenerListaUsuarios()**: devuelve los identificadores de todos los usuarios presentes en una aplicación colaborativa.
- Método **obtenerDatos()**: obtiene los datos que envía un objeto de la clase **Coordinador**. Es necesario que el programador de aplicaciones verifique si estos datos corresponden a una acción realizada por un usuario remoto o bien a una indicación de turno de participación.
- Método **pasarAccion(cadena)**: permite pasar a un objeto de la clase **Intermediario** una cadena que contiene información sobre una acción realizada en la interfaz de usuario por el usuario local. La estructura de dicha cadena se deja al criterio del programador de aplicaciones.



#### Simbología

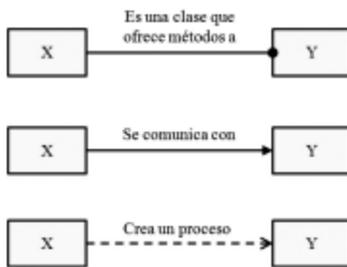


Figura 6: Arquitectura de distribución de SISRESAIUC

Las clases **Adaptador** e **Intermediario** son las API's que como parte de SISRESAIUC se les proporciona a los desarrolladores del núcleo funcional de una aplicación colaborativa; con el fin de: facilitar, transparentar y agilizar el desarrollo de dicha aplicación. De igual forma, se deja a los desarrolladores del núcleo funcional la tarea de diseñar e implementar múltiples versiones de la interfaz de usuario (para diversos dispositivos de cómputo) de la aplicación colaborativa.

## Conclusiones y trabajo futuro

El trabajo realizado en el presente artículo consistió en la adaptación de la interfaz de usuario de un sistema colaborativo a diferentes contextos de uso (específicamente a la plataforma). Para desempeñar esta adaptación, se desarrolló SISRESAIUC (Sistema de Remodelación Semi-plástica Adaptativa para Interfaces

de Usuario Colaborativas), el cual proporciona a una aplicación colaborativa las siguientes características:

1. soporta un grupo de usuarios y
2. notifica a cada usuario su turno de participación en el momento que le corresponda.

Además, SISRESAIUC dota a la interfaz de usuario de una aplicación colaborativa de las siguientes propiedades de plasticidad:

1. adaptabilidad a diversos dispositivos de cómputo (e.g., PC, laptop, smartphone y tablet); y
2. portabilidad entre diferentes sistemas operativos (e.g., Windows, Linux, Mac y Windows Mobile).

A partir de los resultados obtenidos, se concluye que SISRESAIUC podría ser utilizado en, e.g., herramientas de análisis de problemas o soluciones y paneles de discusión virtual que cuenten con un moderador.

El proceso de adaptación plástica mencionado en el presente documento sólo se lleva a cabo una vez. i.e., detecta el dispositivo en que se ejecuta la aplicación colaborativa y, en respuesta, despliega la interfaz de usuario correspondiente. Sin embargo, existen dispositivos de cómputo (e.g., Tablets, PDA, iPod, Smartphone, etc) que al ser rotados giran la vista de sus aplicaciones. En tales dispositivos, es necesario que el proceso de adaptación plástica no sólo se aplique una vez ya que, en función de dichos giros, es necesario desplegar una interfaz de usuario adecuada. Con base en tal necesidad se propone a futuro añadir al módulo de adaptación plástica un proceso de censado que verifique la posición del dispositivo de cómputo (i.e., vertical/horizontal), con la finalidad de mostrar la mejor interfaz de usuario en todo momento 

## Referencias

1. Ardaiz, O. de Cerio, L. D. Freitag, F. Gallardo, A. Marqués, J. M. Messeguer, R. Navarro, L. and Sanjeevan, K. (2004). *Sistemas Distribuidos y CSCL*. Revista Iberoamericana de Inteligencia Artificial, 8(24), pp. 13-20.
2. Bell, M. Chalmers, M. Barkhuus, L. Hall, M. Sherwood, S. Tennent, P. Brown, B. Rowland, D. Benford, S. Capra, M. and Hampshire, A. (2006). *Interweaving Mobile Games with Everyday Life*. Proceedings of the Conference on Human Factors in Computing Systems, ACM Press, pp. 417-426, Montreal, Canada.
3. Ciavarella, C. and Paternò, F. (May 2004). *The Design of a Handheld, Location-Aware Guide for Indoor Environments*. Personal and Ubiquitous Computing, 8(2), Springer Verlag, pp. 82-81.
4. Coutaz, J. and Calvary, G. (2008). *HCI and Software Engineering: Designing for User Interface Plasticity*. The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, pp. 1107-1125.
5. Danesh, A. Inkpen, K. Lau, F. Shu, K. and Booth, K. (2001). *Geney: Designing a Collaborative Activity for the Palm Handheld Computer*. Proceedings of the ACM CHI 2001 Human Factors in Computing Systems Conference, ACM Press, pp. 388-395, Seattle, USA.
6. Grolaux, D. Roy, P. V. and Vanderdonckt, J. (2001). *QTK: A Mixed Model-Based Approach to Designing Executable User Interfaces*. Lecture Notes in Computer Science, 2254, Springer-Heidelberg, pp. 109-110.
7. Grolaux, D. Roy, P. V. and Vanderdonckt, J. (2002). *FlexClock, a Plastic Clock Written in Oz with the QTK toolkit*. Proceedings of the First International Workshop on Task Models and Diagrams for User Interface Design, Costin Pribeanu and Jean Vanderdonckt (Eds), INFO-REC Publishing House Bucharest, pp. 135-142.

**Roberto Enrique Alberto Lira<sup>1</sup>,**

**Víctor Alberto Gómez Pérez<sup>2</sup>**

<sup>1</sup>Universidad del Istmo campus Ixtepec,

<sup>2</sup>Universidad de la Sierra Sur