



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MCPRESENTER: HERRAMIENTA COLABORATIVA MÓVIL DE  
APOYO A DIVERSAS PRÁCTICAS PEDAGÓGICAS Y GESTIÓN DEL  
CONOCIMIENTO

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN  
COMPUTACIÓN

DUŠAN NIKOLA JURETIĆ DÍAZ

PROFESOR GUÍA:  
NELSON ANTRANIG BALOIAN TATARYAN

MIEMBROS DE LA COMISIÓN:  
SERGIO OCHOA DELORENZI  
GUSTAVO ZURITA ALARCÓN

SANTIAGO DE CHILE  
ABRIL 2009

# Resumen

Actualmente, los profesores disponen de diversos medios computacionales para presentar contenido en la sala de clases. Para apoyar la labor del profesor frecuentemente se emplea software basado en diapositivas. Sin embargo, la utilización de estos programas en la sala de clases ocasiona que la enseñanza sea entregada de forma lineal y con poco nivel de interactividad por parte de los alumnos. Eso hace más difícil la tarea del profesor de estimar el nivel de conocimiento de los alumnos de la clase.

La llegada de dispositivos computacionales a la sala de clases junto con el desarrollo de software educativo para estos dispositivos permite a los profesores realizar actividades para ver la comprensión de los tópicos de la clase por parte de los estudiantes, o realizar una exposición de manera no lineal. Una revisión bibliográfica de la literatura existente permitió establecer que hay una carencia de aplicaciones para la sala de clase que permitan al profesor medir el conocimiento de los alumnos, ofrecer un esquema no lineal de enseñanza y que puedan ser ejecutados en dispositivos móviles.

El presente trabajo propone una aplicación, llamada MCPresenter, que permite al profesor, además de gestionar el contenido entregado a los alumnos en las diapositivas, la creación y gestión de actividades destinadas a ver el nivel de conocimiento de los alumnos. Así, el profesor dispone de más herramientas para poder adaptar sus métodos de enseñanza dependiendo de las características de la clase. La aplicación permite al alumno tomar notas en clase y compartirlas con sus pares, además de participar en la clase, responder las preguntas del profesor tanto individual como grupalmente y ver cómo ha sido su desempeño respondiendo preguntas.

Utilizando la aplicación el profesor puede realizar presentaciones a los alumnos, pudiendo modificar en tiempo real el orden de la presentación y guardarlo para uso posterior. La interfaz de usuario está basada en el reconocimiento de gestos a mano alzada con el lápiz para aprovechar poco espacio disponible en los dispositivos móviles y aprovechar las ventajas de las pantallas sensibles al tacto, además del hecho de asemejarse a la escritura con lápiz y papel.

Si bien el resultado final del trabajo cumplió con la mayoría de los objetivos planteados, falta hacer una validación con usuarios reales del sistema para ver qué aspectos mejorar en términos de usabilidad de la interfaz gráfica y funcionalidad.

A mi padre Carlos por su apoyo incondicional y por ayudarme a que este momento llegara, tu recuerdo siempre estará conmigo.

A mi madre Ximena por todo el amor, ayuda y comprensión que me ha entregado.

A a mis hermanos, Nevenka y Jerko, por su constante apoyo y alegría en los buenos y malos momentos de mi vida.

# Tabla de Contenidos

<b>1. Introducción</b>	<b>1</b>
1.1. Definiciones . . . . .	3
1.2. Motivación . . . . .	5
1.3. Objetivos . . . . .	7
1.4. Metodología . . . . .	7
<b>2. Trabajos relacionados</b>	<b>8</b>
2.1. Software usado en la sala de clases . . . . .	8
2.2. Uso de interfaces con gestos y trazos a mano alzada . . . . .	11
2.3. Toma de notas en clase . . . . .	12
2.4. SmartCore . . . . .	12
<b>3. Requerimientos del sistema</b>	<b>14</b>
3.1. Ejecución en dispositivos móviles . . . . .	14
3.2. Arquitectura P2P . . . . .	15
3.3. Uso interfaz a mano alzada y compatibilidad de pantallas . . . . .	15
3.4. Gestión de contenidos . . . . .	15
3.5. Toma de notas . . . . .	16
3.6. Gestión de actividades pedagógicas para los alumnos . . . . .	16
<b>4. Solución implementada</b>	<b>17</b>
4.1. Actividades pedagógicas . . . . .	17
4.2. Comentarios generales . . . . .	17
4.3. Área del pizarrón . . . . .	18
4.3.1. Dar permiso a los alumnos para escribir . . . . .	19
4.3.2. Filtrar trazos por estudiante . . . . .	19
4.3.3. Traer alumnos a página actual . . . . .	20
4.3.4. Pedir permiso de escritura al profesor . . . . .	20
4.4. Toma de notas . . . . .	22
4.4.1. Escribir notas . . . . .	23
4.4.2. Compartir notas . . . . .	23
4.4.3. Ver notas de otros . . . . .	23
4.5. Manejo de presentaciones . . . . .	24
4.5.1. Creación de presentaciones . . . . .	24
4.5.2. Creación de diapositivas . . . . .	25
4.5.3. Mover diapositivas . . . . .	25
4.5.4. Duplicación de diapositivas . . . . .	25
4.5.5. Eliminación de diapositivas . . . . .	26
4.5.6. Scroll/zoom en la vista de diapositivas . . . . .	26
4.5.7. Importar diapositivas de PowerPoint . . . . .	27
4.6. Edición de recorridos . . . . .	28
4.6.1. Creación de recorridos . . . . .	29
4.6.2. Edición de recorridos . . . . .	30

4.6.3.	Contracción/expansión de árbol de nodos . . . . .	31
4.6.4.	Scroll/zoom . . . . .	31
4.6.5.	Ir a página de un nodo . . . . .	31
4.6.6.	Definición de recorrido activo . . . . .	32
4.6.7.	Navegación usando el recorrido actual . . . . .	33
4.7.	Definición de grupos . . . . .	34
4.7.1.	Creación de grupos . . . . .	34
4.7.2.	Asignación de estudiantes a un grupo . . . . .	34
4.7.3.	Sacar a estudiantes de un grupo . . . . .	36
4.7.4.	Eliminación de grupos . . . . .	36
4.8.	Definición de preguntas . . . . .	36
4.8.1.	Creación de una nueva pregunta . . . . .	37
4.8.2.	Envío de preguntas a los estudiantes en forma individual . . . . .	38
4.8.3.	Envío de preguntas a los estudiantes en forma grupal . . . . .	38
4.8.4.	Confirmar respuestas libres como correctas/incorrectas . . . . .	39
4.8.5.	Ver respuestas de los alumnos . . . . .	39
4.8.6.	Superposición trazos en preguntas libres . . . . .	40
4.9.	Responder preguntas . . . . .	40
4.9.1.	Responder de manera individual . . . . .	41
4.9.2.	Responder de manera grupal . . . . .	42
4.9.3.	Ver preguntas recibidas . . . . .	43
4.10.	Resultados de las preguntas . . . . .	44
4.10.1.	Vista de las respuestas de los alumnos . . . . .	44
4.10.2.	Gráficos por estudiante y por pregunta . . . . .	46
4.10.3.	Scroll/zoom . . . . .	46
<b>5.</b>	<b>Implementación</b>	<b>46</b>
5.1.	Ambiente de programación . . . . .	47
5.2.	Arquitectura de la plataforma SmartCore . . . . .	47
5.2.1.	SmartDraw . . . . .	48
5.2.2.	Postprocesables . . . . .	48
5.2.3.	Módulo de red . . . . .	48
5.2.4.	Dibujables . . . . .	49
5.2.5.	Contenedor de dibujables . . . . .	50
5.2.6.	Widgets . . . . .	50
5.2.7.	Gestos . . . . .	50
5.2.8.	Modos . . . . .	51
5.2.9.	Dibujado de la pantalla . . . . .	52
5.3.	Detalles de la implementación . . . . .	52
5.3.1.	Comentarios generales . . . . .	53
5.3.2.	Área del pizarrón . . . . .	53
5.3.2.1.	Dar permiso a los alumnos para escribir . . . . .	54
5.3.2.2.	Filtrar trazos por estudiante . . . . .	55
5.3.2.3.	Traer alumnos a la página actual . . . . .	55
5.3.2.4.	Pedir permiso de escritura al profesor . . . . .	55

5.3.3.	Toma de notas . . . . .	55
5.3.3.1.	Escribir notas . . . . .	56
5.3.3.2.	Compartir notas . . . . .	56
5.3.3.3.	Ver notas de otros . . . . .	56
5.3.4.	Manejo de presentaciones . . . . .	57
5.3.4.1.	Creación de presentaciones . . . . .	57
5.3.4.2.	Creación de diapositivas . . . . .	57
5.3.4.3.	Mover diapositivas . . . . .	57
5.3.4.4.	Duplicación de diapositivas . . . . .	58
5.3.4.5.	Eliminación de diapositivas . . . . .	58
5.3.4.6.	Scroll/zoom en la vista de diapositivas . . . . .	58
5.3.4.7.	Importar diapositivas de PowerPoint . . . . .	58
5.3.5.	Edición de recorridos . . . . .	59
5.3.5.1.	Creación de recorridos . . . . .	60
5.3.5.2.	Edición de recorridos . . . . .	61
5.3.5.3.	Contracción/expansión de árbol de nodos . . . . .	61
5.3.5.4.	Scroll/zoom . . . . .	62
5.3.5.5.	Ir a página de un nodo . . . . .	62
5.3.5.6.	Definición de recorrido activo . . . . .	62
5.3.5.7.	Navegación usando el recorrido . . . . .	62
5.3.6.	Definición de grupos . . . . .	63
5.3.6.1.	Creación de grupos . . . . .	63
5.3.6.2.	Asignación de estudiantes a un grupo . . . . .	63
5.3.6.3.	Sacar a estudiantes de un grupo . . . . .	63
5.3.6.4.	Eliminación de grupos . . . . .	63
5.3.7.	Definición de preguntas . . . . .	64
5.3.7.1.	Creación de una nueva pregunta . . . . .	66
5.3.7.2.	Envío de preguntas a los estudiantes en forma individual . . . . .	66
5.3.7.3.	Envío de preguntas a los estudiantes en forma grupal . . . . .	66
5.3.7.4.	Confirmar respuestas libres como correctas/incorrectas . . . . .	66
5.3.7.5.	Ver respuestas de los alumnos . . . . .	66
5.3.7.6.	Superposición trazos en preguntas . . . . .	67
5.3.8.	Responder preguntas . . . . .	67
5.3.8.1.	Responder de manera individual . . . . .	67
5.3.8.2.	Responder de manera grupal . . . . .	67
5.3.8.3.	Ver preguntas recibidas . . . . .	68
5.3.9.	Resultados de las preguntas . . . . .	68
5.3.9.1.	Vista de las respuestas por alumno . . . . .	68
5.3.9.2.	Gráficos por estudiante y pregunta . . . . .	69
5.3.9.3.	Scroll/zoom . . . . .	70
5.3.10.	Funciones generales . . . . .	70
5.3.10.1.	Diferenciación de trazos de participante por color . . . . .	70
5.3.10.2.	Gesto de scroll/zoom . . . . .	70
5.3.10.3.	Ocultamiento de dibujables . . . . .	71
5.3.10.4.	Gestos genéricos . . . . .	72

<b>6. Conclusiones</b>	<b>74</b>
<b>7. Trabajo futuro</b>	<b>76</b>
7.1. Rol de proyector . . . . .	76
7.2. Mejora en serialización de objetos . . . . .	76
7.3. Pruebas de usabilidad del sistema . . . . .	76
7.4. Diferenciación de trazos de personas . . . . .	76
7.5. Simplificar el diseño de los modos . . . . .	77
7.6. Utilización de clases parametrizables en SmartCore . . . . .	77
7.7. Aviso a modos de nueva sesión . . . . .	77
7.8. Mejorar la edición a mano alzada . . . . .	77
7.9. Inserción de objetos en las presentaciones . . . . .	78
7.10. Edición de contenido de diapositivas PowerPoint . . . . .	78
<b>8. Bibliografía y referencias</b>	<b>79</b>

# Índice de figuras

1.	Screenshot de Classroom Presenter . . . . .	9
2.	Diagrama de los módulos principales de la plataforma SmartCore . . . . .	13
3.	Screenshot de SmartCore . . . . .	14
4.	Modo del área del profesor (vista de profesor) . . . . .	18
5.	Expansión de la barra de lista de estudiantes . . . . .	19
6.	Utilización de barra de estudiantes para filtrar visibilidad de trazos de estudiantes	20
7.	Expansión de “barra de opciones adicionales del modo” . . . . .	21
8.	Awareness visual que recibe un estudiante cuando el profesor le da o le quita el permiso de escritura. . . . .	21
9.	Barra de participantes en modo de notas . . . . .	22
10.	Cambio de visibilidad de notas de un participante . . . . .	23
11.	Gesto de creación de una presentación . . . . .	25
12.	Gesto de creación de nuevas diapositivas . . . . .	25
13.	Cambiando diapositivas de lugar dentro de la presentación . . . . .	26
14.	Copiado y pegado de diapositivas . . . . .	27
15.	Borrado de diapositivas . . . . .	28
16.	Gesto de scroll/zoom . . . . .	28
17.	Manejo de zoom en la vista de diapositivas . . . . .	29
18.	Importación de presentación PowerPoint al área de trabajo . . . . .	29
19.	Cambio de zoom de imagen de diapositiva importada . . . . .	30
20.	Gesto de creación de recorrido . . . . .	30
21.	Agregar diapositiva al recorrido . . . . .	31
22.	Expansión/contracción del árbol de modo de edición de recorridos . . . . .	32
23.	Ir a página determinada en modo de edición de recorridos . . . . .	32
24.	Fijando un recorrido como activo . . . . .	33
25.	Gesto de cambio de diapositivas según recorrido . . . . .	33
26.	Modo de grupos . . . . .	34
27.	Creación de grupos . . . . .	35
28.	Asignación de grupos . . . . .	35
29.	Ver integrantes de un grupo . . . . .	35
30.	Sacando un estudiante de un grupo . . . . .	36
31.	Eliminación de un grupo . . . . .	36
32.	Barra de menú de tipo de pregunta . . . . .	37
33.	Vista de pregunta del profesor y alumno . . . . .	38
34.	Gestos de confirmación de respuesta del alumno como correcta/incorrecta . .	39
35.	Vista de la respuesta de un estudiante a una pregunta . . . . .	40
36.	Vista combinada de las respuestas de los estudiantes a una pregunta libre . .	40
37.	Indicador de pertenencia a un grupo . . . . .	41
38.	Respondiendo a una pregunta individual de selección simple . . . . .	42
39.	Contestar una pregunta grupal . . . . .	43
40.	Gesto de scroll/zoom en modo de “Ver preguntas” . . . . .	44
41.	Filas y columnas en modo de “Resultados de preguntas” . . . . .	45
42.	Ir a respuesta en modo de “Resultados de preguntas” . . . . .	45



43.	Gesto de scroll/zoom modificado . . . . .	46
44.	Arrastrando el stylus en forma diagonal cambia el nivel de zoom del modo. .	47
45.	Diagrama de clases de los modos de MCPresenter . . . . .	53
46.	Diagrama de estados del alumno en TeacherMode . . . . .	54
47.	División por capas de la vista del “modo de notas”. . . . .	56
48.	Diagrama de estados de una pregunta libre . . . . .	65
49.	Diagrama de estados de una pregunta no libre . . . . .	65
50.	Diagrama de clases de VGraph y clases relacionadas . . . . .	69
51.	Diagrama de clases de HideManager y sus clases derivadas . . . . .	71
52.	Diagrama de secuencia de HideManager . . . . .	72

# 1. Introducción

La gestión del conocimiento se define como la habilidad de crear, validar, distribuir, revisar, compartir y aplicar conocimiento [1]. Uno de los desafíos de la gestión del conocimiento es pasar del conocimiento tácito (conocimiento internalizado que no puede ser fácilmente expresado) a conocimiento explícito (que puede ser fácilmente comunicado), por medio de la externalización de esos conocimientos. Varias instituciones han aplicado sistemas de gestión de conocimiento para proveer un medio para la distribución del conocimiento, con el fin que las personas puedan acceder a la información y conocimiento, y lo interioricen. La utilización de sistemas de gestión del conocimiento se ha restringido a escenarios fijos, excluyendo aquellos escenarios donde existe movilidad e interacciones cara a cara entre los participantes [1].

Las instituciones educativas son fuente de conocimiento [2]. La gestión del conocimiento puede aplicarse a las salas de clases, permitiendo el intercambio de datos e información entre profesores y alumnos. Es necesario soporte computacional para el manejo de datos de información, que pueda ser convertido en conocimiento por sus usuarios. Una forma de intercambiar conocimiento es mediante una exposición que realiza un moderador a una audiencia, donde el moderador utiliza un sistema de gestión del conocimiento para traspasar conocimiento a la audiencia. En el caso de una sala de clases, el moderador representa a un profesor, la audiencia a los estudiantes, y el sistema de gestión de conocimiento al software utilizado por el profesor para exponer contenido.

Hoy en día los profesores cuentan con diversos dispositivos computacionales y variado software que les permite presentar contenido multimedial (datos e información) en la sala de clases, como por ejemplo [3]: videos, diapositivas PowerPoint y presentaciones basadas en Web. El contenido multimedial, así como los dispositivos usados ayudan a enriquecer la presentación del profesor haciéndola más interesante y entretenida, a la vez que abre la posibilidad a que el profesor y los alumnos manipulen colaborativamente este material haciendo la clase más participativa y “constructivista”. Sin embargo, en las salas de clase frecuentemente se utiliza software basado en diapositivas como herramienta de apoyo a la docencia. Este tipo de presentaciones se suele usar más bien como un esquema para guiar al presentador en vez de educar a la audiencia [4], además de promover la generación de contenido con una jerarquía compleja que conduce a la desorientación de los espectadores [4]. Normalmente se presenta el material siguiendo un orden secuencial, mostrando el contenidos sin ningún tipo de apoyo a prácticas pedagógicas que enriquezcan el proceso de aprendizaje, como por ejemplo, retroalimentación (feedback), seguimiento y evaluación (assessment), reflexión (reflection), resolución de problemas en forma individual o grupal, actividades colaborativas, etc. [5]. Por lo tanto, en estas clases los alumnos sólo pueden escuchar lo que el profesor dice y las oportunidades de participar son mínimas, es decir, las interacciones alumno-profesor y alumno-alumno se dan en un nivel muy bajo. Lo anterior significa que el profesor presenta la información a los estudiantes sin guiar su proceso educativo, afectando negativamente la atención y motivación de los estudiantes [25]. Por otro lado, el paradigma del “single-speaker”, en el cual sólo una persona (alumno o profesor) comunica a la vez, limita la participación de los alumnos en la sala de clases [6]. Además de los factores mencionados anteriormente, la falta de interacción por parte de los alumnos puede ser causada por el temor de éstos de

hablar dado el ambiente de la clase [6], o por la posibilidad de quedar en vergüenza frente a los demás compañeros [7].

Todo esto dificulta que el profesor sepa con certeza cuál es el grado de avance y comprensión de la materia que el alumno presenta, con lo cual el profesor difícilmente se podrá adaptar a la clase [25] y dar una adecuada retroalimentación.

Esta forma de enseñar está en diametral oposición a lo que los expertos en educación recomiendan. Dentro de las teorías educativas más aceptadas hoy en día tenemos el constructivismo, que entre otras cosas sostiene que el conocimiento se construye de manera colaborativa [8]. Esta es la teoría más importante de las ciencias de la educación que justifica y promueve el desarrollo de lo que se conoce como Computer-Supported Cooperative Learning (CSCL).

Con la aparición y el posterior popularización de los dispositivos computacionales en la sala de clases, tales como PCs, PDAs, Tablet-PCs, SmartPhones y agendas electrónicas, se han desarrollado aplicaciones que apoyan la docencia y el aprendizaje dentro de la sala de clases con mayor interactividad, utilizando dispositivos computacionales para implementar funcionalidades que introduzcan prácticas pedagógicas que favorezcan la interacción alumno-profesor y alumno-alumno. Entre estas funcionalidades se puede nombrar la posibilidad que al profesor pueda definir preguntas que los alumnos respondan durante la clase [5][9], permitir a los alumnos escribir sobre las diapositivas de la presentación del profesor [10], realizar actividades colaborativas formando grupos de estudiantes para resolver problemas [11], etc.

En consecuencia, es posible realizar una clase combinando la exposición con el trabajo de los alumnos interactuando entre ellos y con el profesor, dejando reflejadas en medios digitales tanto las indicaciones surgidas de la misma exposición como también los apuntes personales de cada participante.

A lo anterior podemos agregar la capacidad de implementar prácticas colaborativas que permitan que las preguntas o aportes de cada uno de los participantes pueda ser conocida por todos (eventualmente filtradas por el expositor previamente) y permitir una dinámica integrante entre los participantes de todo el grupo. Además, el profesor puede realizar preguntas dirigidas a sus alumnos y revisar las respuestas que han dado, para posteriormente mostrar a la clase las respuestas que merezcan discusión. Esto hace que la clase avance a medida que el conjunto de los participantes va demostrando que ha entendido la información entregada.

En los últimos años ha surgido otra forma de apoyar la docencia con computadores llamada aprendizaje móvil. El aprendizaje móvil (m-learning), complementa la educación presencial a través del uso de dispositivos móviles como el teléfono celular y otros dispositivos móviles con conexión a la red [12].

Sin embargo, una revisión bibliográfica permite darse cuenta que hay una carencia en lo que respecta a sistemas móviles que apoyen de presentación de material multimedial educativo y que además permitan al profesor realizar actividades colaborativas con sus alumnos utilizando el material de presentación como base. En este escenario el proceso de enseñan-

za/aprendizaje se puede beneficiar al contar el profesor con funcionalidades que le permitan exponer contenidos y realizar actividades colaborativas con los alumnos usando este material sin tener que cambiar de contexto, ya que los cambios de contexto han mostrado ser una de las principales causas de pérdida de atención y confusión de los alumnos durante clases apoyadas por tecnología computacional [13].

En esta trabajo se desarrolló una herramienta de apoyo a clases, denominada MCPresenter (*Mobile Collaborative Presenter*), la cual está concebida para ser usada en Tablet-PC, PDA y pizarra electrónica (usado por un presentador/profesor) durante una clase presencial. El sistema permite al profesor (que actúa como moderador) distribuir material educativo, el cual es recibido y modificado por los alumnos en sus PDAs. Cada participante tiene la posibilidad de realizar sus propios apuntes sobre el material entregado por el presentador. Los participantes pueden plantear sus dudas al profesor o hacer aportes a la clase a través del sistema. Las dudas son tomadas por el profesor y respondidas por el sistema, mientras que los aportes son mostrados al curso. El presentador puede definir durante la clase enunciados a problemas que estén asociados al material presentado, las cuales pueden ser respondidos de inmediato por los alumnos. El presentador puede evaluar las respuestas y así medir el nivel de avance y comprensión de la materia. Además, el profesor puede separar a la clase en grupos, asignarles un problema, y puede ver a través del sistema qué es lo que está haciendo cada uno de ellos. El sistema implementa un estilo de interacción humano-computador basado principalmente en gestos y escritura a mano alzada como modo de ingresar la información de modo de tener más espacio disponible para ingresar contenido [14], y es el modo más natural de ingreso de datos en una pantalla sensible al tacto (*touch-screen*) [14]. MCPresenter se implementa sobre una plataforma, denominada SmartCore, que se encarga de la comunicación entre dispositivos, el manejo de los gestos, la escritura a mano alzada, la toma de notas personales y el manejo de grupos. La comunicación entre los dispositivos se realiza de manera inalámbrica. MCPresenter no estará limitado a un área específica del conocimiento. En otras palabras, el sistema será genérico, lo que da la flexibilidad necesaria para apoyar a la enseñanza en cualquier área.

## 1.1. Definiciones

En esta sección se detallan términos que van a ser utilizados a lo largo de este documento junto con su definición respectiva.

- **Assessment:** Proceso iterativo que entrega información al profesor para mejorar la enseñanza y el aprendizaje, y que evalúa si los cambios realizados afectan el aprendizaje de los estudiantes [5]. Gracias al *assessment* el profesor puede ir evaluando cómo va avanzando el aprendizaje de los alumnos, lo que posibilita la adaptación de los métodos de enseñanza para ajustarse a la clase. El grado de *assessment* depende del método de enseñanza que aplique el profesor.
- **Feedback :** Proceso usado para notificar a los alumnos de la calidad y/o exactitud de sus respuestas a las preguntas [5], con el objetivo de apoyar el aprendizaje de los alumnos. Para la enseñanza convencional en la sala de clases los investigadores recomiendan

que el *feedback* sea inmediato. Algunos niveles de *feedback* incluyen: la respuesta correcta, explicación de porqué la respuesta correcta lo es, o explicación sobre qué está mal en la respuesta incorrecta [5].

- **.NET Framework:** Componente de software, desarrollado por Microsoft, que puede ser agregado al sistema operativo Windows. Provee un conjunto de soluciones predefinidas para la programación de aplicaciones, y se encarga de administrar los programas escritos para esa plataforma [15]. Los programas que han sido diseñados para correr en la plataforma deben poder ser ejecutados en cualquier sistema donde el *framework* haya sido implementado.
- **.NET Compact Framework:** Versión reducida del .NET Framework, diseñada con el fin de poder ejecutar aplicaciones en dispositivos compatibles con Windows Mobile (por ejemplo, PDAs, Pocket PCs y SmartPhones). Al igual que en el .NET Framework, las aplicaciones diseñadas para esta plataforma pueden ejecutarse en cualquier sistema donde este *framework* haya sido implementado, puesto que la plataforma permite ejecutar programas independiente del hardware del dispositivo.
- **PDA:** Sigla de *Personal Digital Assistant* (en español Asistente Digital Personal). Computador que cabe en la palma de una de mano, que posee una pantalla táctil. Estos computadores poseen varias capacidades que proveen los computadores de escritorio modernos [16]. Hoy en día esos dispositivos poseen conectividad Wi-Fi, Bluetooth, IrDa e incluso los equipos más avanzados ofrecen servicios de telefonía. En principio fue utilizada como agenda electrónica. Dado su tamaño reducido ayudan a la movilidad (pues posee una batería interna lo que permite su uso independiente por un tiempo, además de ofrecer conectividad inalámbrica) y posibilitan las interacciones *face-to-face*.
- **Pizarra electrónica:** Pantalla táctil conectada a un computador, donde es proyectado el escritorio de ese computador. Los usuarios pueden controlar ese computador por medio de un lápiz o con los dedos. Según [17], usando una pizarra electrónica el profesor puede mantener la atención de los alumnos en sus explicaciones y aprovechar el potencial de la pizarra para dibujar esquemas, gráficos, etc.
- **Problem based learning:** Práctica pedagógica en la cual el aprendizaje es conducido a través de problemas abiertos, que pueden ser resueltos por los estudiantes tanto de forma grupal o individual. El profesor va guiando el proceso para facilitar el aprendizaje, y el proceso puede ser usado para fomentar la comunicación, interacción social, habilidades colaborativas y de resolución de problemas [5].
- **Red inalámbrica ad-hoc:** Red punto a punto creada dinámicamente sobre el protocolo IEEE 802.11. Este tipo de red permite que los dispositivos se comuniquen entre sí de manera directa sin la mediación de un *router*, mejorando la velocidad [15]. Esto permite formar redes sin tener que configurar una complicada infraestructura de red. Actualmente, los computadores personales, notebooks y dispositivos móviles con tarjetas de red inalámbricas pueden crear una red ad-hoc o conectarse a una ya existente.
- **Stylus:** Vara similar a un lápiz, que es utilizada como método de entrada para algunos dispositivos con pantalla táctil (como por ejemplo, PDA).

- **Tablet-PC:** Computador que permite la escritura a través de su pantalla táctil, en vez de utilizar el teclado o el mouse, incorporando todas las funcionalidades de un notebook normal (incluyendo conectividad inalámbrica). Estos equipos tienen un sistema de reconocimiento de texto, que permite pasar a texto plano lo que el usuario ha dibujado en la pantalla táctil. Una de las ventajas de estos dispositivos es la escritura a mano alzada con el stylus, que se asemeja a la escritura tradicional con lápiz y papel. En una sala de clases, se ha observado que un profesor usando Tablet-PC mantiene contacto visual con los estudiantes mayor tiempo, dado que no tiene que darse vuelta para anotar en un pizarrón [18].

## 1.2. Motivación

El modelo de aprendizaje colaborativo está basado en los supuestos constructivistas, en los cuales se visualiza el aprendizaje como una construcción del conocimiento que realiza el alumno al interactuar con el medio [19]. Es por esto que el paradigma constructivista contempla un modelo del estudiante que describe su estado. Ese estado consiste en su conocimiento previo al tema, conocimiento actual, su interés, etc. Al modelar aprendizaje colaborativo se incluyen factores del grupo como: tareas realizadas en grupo, información del alumno del resto de los componentes, creencias del grupo, etc. Eso lleva a poner especial atención a las interacciones alumno-profesor y alumno-alumno para apoyar, estimular y evaluar el trabajo colaborativo del grupo. La aplicación de sistemas computacionales de apoyo a la enseñanza en la sala de clases ha mostrado ser beneficioso para los estudiantes, dado que, entre otras cosas, aumenta la comunicación entre alumnos con el profesor y con sus pares [20].

Según Collazos *et al.*[21]: “Los efectos del aprendizaje colaborativo son a menudo evaluados por medidas de desempeño de test individuales. Podría considerarse que una evaluación más válida sería medir el desempeño del grupo”. Como las evaluaciones se van haciendo a medida que avanza la clase, al llegar a metas previamente fijadas se puede saber, sin nuevos diagnósticos, cómo va el aprendizaje del grupo.

En varios trabajos se ha logrado mostrar que los dispositivos móviles son apropiados para apoyar el trabajo colaborativo de grupos (ver ejemplos en [22]). Algunas investigaciones han tratado de utilizar dispositivos móviles para promover la interacción o cooperación (ver ejemplos en [23]). Estos dispositivos traen como ventaja su movilidad, que permite acercarse a las personas con las cuales se requiere interactuar y formar grupos. Además, por su tamaño permite interacción cara a cara (*face-to-face*) entre los estudiantes y el profesor.

Un factor importante que puede afectar el aprendizaje de los alumnos es la atención. La atención de los estudiantes usualmente dura alrededor de 20 minutos [24]. Una manera de mantener la atención es mediante cambios de actividades durante la clase [25]. Pero hay factores que limitan a los estudiantes a participar en clase [6][7]. Los estudiantes que no se ven involucrados en las discusiones en clases, simplemente esperan que los otros estudiantes respondan las preguntas, lo que ocasiona que ellos se pierdan el proceso de razonar, concluir, equivocarse y analizar los errores propios como parte del proceso de aprendizaje [26]. Una forma de disminuir la cohibición de los estudiantes a la hora de responder preguntas del

profesor es utilizar un sistema provea anonimidad a los estudiantes [6].

Una propuesta alternativa para los sistemas de presentación existentes (ver sección 2.1) sería uno donde sea posible definir presentaciones para especificar contenido, que no limite la participación y promueva un alto grado de interacción. Adicionalmente, que se puedan crear preguntas para los alumnos en cualquier lugar (por ejemplo, en la misma presentación) para poder dar información de contexto a las preguntas, con la finalidad de mantener la atención de los alumnos y ver su nivel de conocimiento.

La propuesta de esta memoria, MCPresenter, apoyará el trabajo grupal, y promoverá la participación y colaboración entre los miembros. De esta manera los miembros de los grupos podrán aportar a la discusión de la materia, y el instructor, mediante preguntas dirigidas a los alumnos, podrá ayudar al aprendizaje, estimulando, coordinando y sugiriendo caminos. Por cuanto el paradigma constructivista describe el aprendizaje como la construcción del conocimiento en grupo, todos los participantes deberán involucrarse con la clase, ya sea contribuyendo con ideas a la discusión o respondiendo a las preguntas del profesor.

Un sistema de las características nombradas anteriormente debe permitir al instructor promover la participación de cada alumno en línea. La aplicación adicionalmente permitirá el manejo de preguntas para que el profesor, durante la clase o antes de ella, pueda enunciar problemas mediante el sistema y posteriormente ver cómo están respondiendo los alumnos ante ellos. Además el profesor podría realizar diagnósticos posteriores tomando como base la información que entrega la herramienta para estimar el grado de retención de la materia, de generalización de los conceptos, de profundización del estudio, de relación con otras materias.

Un ejemplo del escenario que se quiere apoyar con este sistema es el siguiente: en una clase de cálculo, en el método tradicional de hacer clases, cuando el profesor quiere saber si los alumnos han entendido la resolución de integrales, él pregunta de forma abierta a la clase o hace pasar al pizarrón a algunos alumnos para resolver un ejercicio, donde generalmente participan de forma voluntaria los alumnos con mayor entendimiento de la materia. Si el profesor hace participar a los alumnos que entienden menos, ellos se sienten avergonzados o cohibidos a la hora de responder. Con MCPresenter, el profesor puede plantear preguntas hacia la clase entera o a un grupo de ella, por ejemplo, dibujando una integral que se desea resolver. Lo que dibuje el profesor es recibido por los alumnos en sus dispositivos, donde ellos pueden trabajar sobre el dibujo y plasmar su respuesta (si era una pregunta abierta) o seleccionar la respuesta correcta (si la pregunta es de selección múltiple). El anonimato que provee la aplicación disminuye el grado de cohibición de los alumnos al responder. Mientras los alumnos están respondiendo, el profesor puede monitorear a través del sistema cómo están trabajando los alumnos (tanto en forma individual como grupal), y si él encuentra que un alumno o grupo no avanza lo suficiente, puede dibujarles una ayuda en su espacio de trabajo para que puedan seguir adelante con la resolución del problema. A medida que los alumnos van respondiendo el profesor puede ver los resultados para formarse una idea del nivel de conocimiento de los alumnos, y si hay una “tendencia” en las respuestas. Por otro lado, si el profesor encuentra una respuesta interesante, la puede seleccionar en la aplicación y mostrarla a toda la clase para poder hacer un análisis de ella.

### 1.3. Objetivos

El objetivo general es desarrollar una herramienta colaborativa móvil que sirva de apoyo a la enseñanza en la sala de clases, apoyando la adaptación del contenido a la clase y prácticas pedagógicas como el *assessment* y *feedback* a través de preguntas que puede hacer el profesor, permitiendo la exposición de material educativo a través de presentaciones, y que pueda ser utilizado desde Tablet-PC, PDA, pizarra electrónica o PC.

Los objetivos específicos asociados al proyecto son los siguientes:

- Identificar cuáles características son relevantes para un software que es usado en una sala de clases.
- Identificar qué cualidades deben tener las interfaces de usuario de un software de estas características.
- Evaluar el uso del sistema en un ambiente controlado para observar el impacto de su uso, obteniendo ventajas y desventajas de la configuración en uso.
- Permitir que la aplicación sea compatible en PDA, Tablet-PC y pizarra electrónica, tomando en cuenta que cada dispositivo tiene un tamaño de pantalla diferente.
- Implementar la aplicación sobre SmartCore y hacer testing correspondiente.

### 1.4. Metodología

Para el desarrollo de la aplicación, los pasos que se siguieron fueron los siguientes:

- **Revisión de referencias bibliográficas relevantes para el tema de la memoria.** Durante las semanas iniciales del trabajo de memoria se realizó una revisión acerca de sistemas de presentación que fueran utilizados en salas de clases para apoyar al profesor y/o a los alumnos para poder identificar su rol en el mejoramiento de las prácticas pedagógicas, además de buscar información acerca del trabajo a realizar, como por ejemplo, colaboración en la sala de clases, cómo afectan los dispositivos móviles a la educación, entre otros.
- **Estudio de sistemas de presentación relacionados.** Algunos sistemas de presentación estaban disponibles de manera gratuita desde la página web de sus creadores, con lo cual se procedió a la descarga y ejecución esos programas para poder identificar sus ventajas y desventajas al usarlos en una sala de clases.
- **Investigación del lenguaje, estructura y funcionamiento del software de plataforma subyacente (SmartCore).** En paralelo con la revisión bibliográfica, se hizo un trabajo de familiarización con la plataforma SmartCore, que consistía básicamente en ir creando programas pequeños de prueba (sólo con el fin de aprender a usar la plataforma) e ir agregándole más funcionalidades. Como se disponía del código fuente de desarrollos anteriores que se han realizado con SmartCore, se tomó como referencia dicho código para saber cómo estaban implementadas las funcionalidades. Lo anterior



también contribuyó a la familiarización con el funcionamiento de PDAs y Tablet-PCs, dos de las plataformas que pueden ejecutar SmartCore.

- **Desarrollo (diseño e implementación) de MCPresenter**, en etapas:
  - Diferenciación en los roles de alumno y profesor.
  - Módulo de preguntas/respuestas, por medio del cual el profesor puede enunciar preguntas a los alumnos, con la opción de mostrar una vista para el profesor donde puede ver las respuestas de los alumnos.
  - Módulo que permita al profesor autorizar/desautorizar a alumnos a escribir sobre la presentación en curso.
  - Envío de peticiones de escritura por parte de los alumnos, cuando ellos quieran escribir en el área del profesor.
  - Módulo de notas, que permita compartir notas y ver aquellas notas de los demás que han decidido compartir.
  - Módulo para poder manejar diapositivas (crear/modificar/eliminar/importar), y poder realizar presentaciones.
  - Módulo para importar diapositivas desde presentaciones PowerPoint.
  - Modo para poder definir la forma de recorrer o mostrar las diapositivas creadas (definición de recorridos).
  - Modificación de interfaz para que la aplicación pueda ser usada en PDA, Tablet-PC y pizarra electrónica.

## 2. Trabajos relacionados

En esta sección se describen otros sistemas que son utilizados en la sala de clases para de establecer una comparación de las características que ofrecen.

Adicionalmente se realizará una revisión bibliográfica acerca del uso de interfaces con gestos y trazos a mano alzada, y la toma de notas, dado que la solución a implementar tendrá que hacer uso de aquellas funcionalidades. Finalmente se da la descripción general de la plataforma sobre la cual se construirá MCPresenter, llamada SmartCore, para poder diferenciar qué funcionalidades de MCPresenter provienen de la plataforma o son adicionales a ella.

### 2.1. Software usado en la sala de clases

Classroom Presenter [9][27] (ver Figura 1) es un sistema basado en Tablet-PC que posee orientación a la enseñanza en una aula de clases, focalizado en el profesor. El profesor utiliza una Tablet-PC para poder manejar una presentación que contiene diapositivas y puede usar trazos (*sketches*) para profundizar temas o dar ejemplos encima de las diapositivas de la presentación. La diapositiva que está viendo el profesor más sus anotaciones son mostradas a través de un proyector. El sistema permite a los alumnos responder preguntas abiertas del

profesor (por medio de *sketches*) o contestar encuestas (seleccionando la alternativa que consideran correcta). Las presentaciones de PowerPoint pueden ser importadas en la aplicación, pero como imágenes estáticas, por lo cual no es posible modificar las diapositivas originales en sí. Sin embargo, el sistema no provee soporte para la interacción estudiante-estudiante, pues los estudiantes no pueden compartir notas en este sistema. Además, no existe una versión compatible con PDAs.

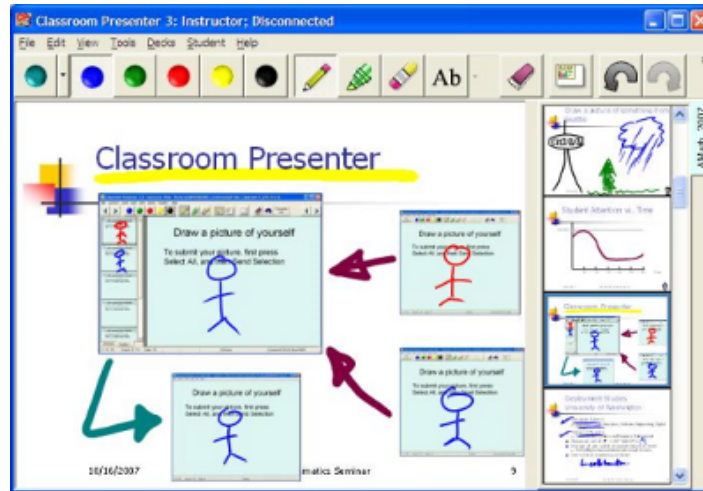


Figura 1: Screenshot de Classroom Presenter (Fuente imagen: [28])

Group Learning Partner [11] es una extensión a Classroom Presenter, heredando sus funcionalidades. Como característica adicional permite al profesor formar grupos de estudiantes a través del sistema, con lo cual los integrantes de ese grupo disponen de un espacio compartido para escribir, y soporta el envío de *sketches* entre estudiantes a través de una red inalámbrica.

Ubiquitous Presenter [29] también extiende a Classroom Presenter. La interfaz gráfica es web, lo cual posibilita a cualquier dispositivo con acceso a red a interactuar con el sistema. Esto último permite que se pueda utilizar en clases a distancia (*distance learning*). A diferencia del Classroom Presenter, el profesor puede asignar tareas (después de la clase) que tienen que pueden ser respondidas a través del sistema por los alumnos. El sistema no posibilita a los alumnos la toma de notas privadas.

Classroom Feedback System (CFS) [6] está orientado a obtener retroalimentación por parte de los alumnos de una clase realizada en Power-Point. El profesor, al igual que en el caso del Classroom Presenter, utiliza una Tablet-PC. Los alumnos, que se conectan a la herramienta a través de laptops, pueden agregar anotaciones para que el profesor sepa si tiene que profundizar en algún tópico o si los alumnos entendieron. No obstante, no hay interacción directa por medio del sistema desde el profesor a los alumnos: no permite al profesor escribir a mano alzada en la presentación para apoyar las explicaciones que esté dando, y no puede asignarles problemas a los alumnos a través del sistema.

En [30] se detalla una aplicación que permite a los alumnos tomar notas de las clases utilizando PDAs, de manera tal que las notas ingresadas pueden ser vistas por los compañeros (los estudiantes deciden quiénes tienen acceso a sus notas personales). El acceso al sistema es por medio de un servidor web. Las notas sólo son en texto plano, pudiéndose reusar las palabras que están en la presentación del profesor o en las notas de otros compañeros para tomar notas personales. Las notas posteriormente pueden ser almacenadas para ser consultadas por los alumnos después de la clase. Sin embargo, el sistema no permite el ingreso de dibujos a mano alzada (*sketching*).

LiveNotes [31] es un sistema que consta de un área compartida donde los alumnos pueden tomar notas a través de Tablet-PCs, usando la comunicación inalámbrica entre ellas. El sistema permite tomar las notas de forma individual o colaborativa. En esta última configuración los estudiantes se forman en pequeños grupos, y se ponen de acuerdo entre ellos para que algunos tomen notas en el área compartida mientras otros ponen atención al profesor. Todo lo que escribe un alumno en el área compartida lo verán los demás integrantes de su grupo. En este sistema no hay interacción alguna con el profesor de manera directa.

DyKnow Vision [10][32] posibilita la toma de notas colaborativa entre profesor y alumno, y repetición del contenido, trazo por trazo (*content replay*). El profesor puede realizar preguntas a los alumnos y ver los resultados desplegados en un gráfico. El profesor en cualquier momento puede pasar el control a determinados estudiantes para que ellos escriban en las diapositivas, para mostrar lo que hacen al resto de la clase. No obstante, no apoya la conformación de grupos de estudiantes para resolver un problema común, ni se puede ejecutar en dispositivos móviles.

En la Tabla 1 se muestra una comparación entre MCPresenter y las aplicaciones nombradas anteriormente, para tener una noción de las características que poseen estos sistemas en términos de las prácticas pedagógicas que soportan, como el *assessment*, *feedback*, *problem-based learning* y colaboración. Si bien, comparando por prácticas pedagógicas MCPresenter, DyKnow Vision y Group Learning Partner apoyan todas las prácticas nombradas en la tabla, sólo MCPresenter puede ser utilizada en dispositivos móviles y tiene soporte para gestos.

Se ha desarrollado una plataforma para dispositivos móviles (*handheld* o PDA) llamada SmartCore [15], en la cual se pueden realizar trabajos colaborativos entre los dispositivos. Este sistema está implementado en C# y desarrollado en Visual Express 2005. Esta herramienta sirve de apoyo a reuniones y usando comunicación (alámbrica o inalámbrica) entre PDAs, permite detectar a otros participantes y obtener el estado actual de la interacción. Además permite que todos los participantes vean lo que se ha escrito (con el lápiz o stylus) y participar en posteriores modificaciones. Tiene manejo de gestos sobre los *sketching* creados y es extensible, pues se pueden crear nuevos módulos sobre la plataforma base. La sección 2.4 explica la plataforma con más detalle.

Nombre sistema	Feedback	Problem-based learning	Colaboración	Assessment
Classroom Presenter [9][27]	Sí, gracias a que los alumnos pueden enviar sus respuestas usando Tablet-PC al profesor.	Sí, el profesor puede preparar preguntas en las diapositivas y hacer que los alumnos las respondan.	No	Sí, el profesor puede enunciar preguntas en vivo y ver cómo responden sus alumnos.
Group Learning Partner [11]	Sí, pues los alumnos pueden enviar lo que han escrito al profesor, y él puede monitorear el avance del grupo.	Sí. Esta característica la hereda de Classroom Presenter.	Sí. La aplicación permite la conformación de grupos para resolver problemas en forma colaborativa.	Sí, pues Classroom Presenter posibilita esta práctica.
Ubiquitous Presenter [29]	Sí, gracias a que los alumnos pueden enviar respuestas vía web al profesor.	Sí, pues Classroom Presenter posibilita esta práctica.	No	Sí, pues Classroom Presenter posibilita esta práctica.
Classroom Feedback System (CFS) [6]	No	No	No	Sí. El docente puede saber con el sistema qué temas han quedado claros para los alumnos.
Collaborative Note Taking [30]	No	No	Sí. Los alumnos ven las notas escritas por sus pares, y escribir las suyas reutilizando las palabras que han escrito otros.	No
LiveNotes [31]	No	No	Sí. Los alumnos, conformados en grupos pequeños, pueden tomar notas en forma colaborativa sobre un área común.	No
DyKnow Vision [10][32]	Sí. Los alumnos pueden enviar su trabajo al profesor, y el docente puede mostrarlo a la clase entera para hacer análisis de él.	Sí, el profesor puede enviar preguntas o encuestas a los alumnos.	Sí. Se pueden tomar notas colaborativas entre alumno y profesor.	Sí, el profesor puede hacer preguntas abiertas o encuestas y ver las respuestas de los alumnos.
MCPresenter	Sí, gracias a que los alumnos pueden enviar sus respuestas usando sus Tablet-PC/PDA al profesor y puede dar las explicaciones pertinentes usando el sistema.	Sí. El profesor puede enviar problemas a sus alumnos, y monitorear su avance.	Sí, todos los integrantes del grupo de alumnos tienen que estar de acuerdo para enviar respuesta a un problema grupal. Además, se pueden tomar notas colaborativas.	Sí, el profesor puede enunciar preguntas en vivo.

Tabla 1: Comparación entre diferentes aplicaciones por prácticas pedagógicas

## 2.2. Uso de interfaces con gestos y trazos a mano alzada

Las interfaces gráficas de algunas aplicaciones de dispositivos móviles usan componentes tomados directamente de las interfaces de computadores personales, tales como botones, menús y ventanas, ocasionando problemas de usabilidad en la interfaz [33]. Existen investigaciones que sustentan que el modo más natural de ingreso de datos en un dispositivo con pantalla táctil es a través de trazos a mano alzada (*sketches*) [14], imitando el uso del lápiz y papel [34]. Mediante *sketches* se pueden describir símbolos visuales, relaciones espaciales [35] e intercambiar opiniones en un medio rápido y eficiente para poder compartir y discutir ideas complejas [36]. A pesar que texto dibujado a mano alzada puede ocupar más espacio

que el texto ingresado por medio de un teclado, permite una combinación flexible entre dibujado y escritura [14]. Algunas aplicaciones que utilizan *sketches* son usadas para diagramar presentaciones informales [37], especificar diagramas de curso de acción en el campo militar [38] o apoyar la generación de ideas en reuniones [39].

En [40] se hizo una comparación entre la utilización por parte de los alumnos de texto plano o *sketches* a mano alzada en las actividades propuestas por el profesor. Como resultado de las actividades en clase, las respuestas de los alumnos con *sketches* son más fáciles de leer para el instructor que un párrafo de texto plano, dado que en el último caso se tienen que analizar las palabras escritas por el alumno con mayor detenimiento. Además, la escritura con el stylus en un dispositivo con pantalla táctil (como Tablet-PC) es natural, el ingreso de datos se realiza de manera fluida [34][40].

Cabe notar que para ciertas actividades el uso de *sketches* no es el medio preferido mayoritariamente por los alumnos. Por ejemplo, en [40], los alumnos tenían divididas sus preferencias (entre usar Tablet-PC o un computador normal) frente a un problema que involucraba escribir código en un lenguaje de programación.

### 2.3. Toma de notas en clase

Usando LiveNotes [31] se experimentó con la toma de notas colaborativa. Los alumnos, con sus Tablet-PC, se juntaban en grupos para tomar notas en clases, sobre un espacio compartido. Además, en el área de notas se muestran las diapositivas del profesor lo cual tiene un efecto positivo comparado con escribirlas a partir de un área vacía [31]. El proceso de “toma de notas colaborativa” tuvo una buena valoración por parte de los estudiantes, dado que los alumnos pueden completar o corregir las notas de sus pares, y se pueden turnar de manera tal que siempre haya alguien pendiente de prestar atención al profesor mientras los otros escriben apuntes.

En [30] utiliza un sistema que permite a los alumnos usar palabras de las notas ingresadas por los compañeros para escribir las notas personales. Según [30], compartir notas mejora la eficiencia del proceso de toma de notas, puesto que un alumno no reescribe notas que fueron escritas por otros. Sin embargo, algunos alumnos admiten la sobrecarga de información producida por prestar atención tanto al profesor como a las notas de los compañeros que van apareciendo en el sistema.

### 2.4. SmartCore

SmartCore es un sistema escrito íntegramente en C# que permite la interacción entre diferentes participantes, y es extensible de modo que posibilita la creación de diferentes aplicaciones usando su núcleo. Los módulos de la aplicación se pueden ver en la Figura 2 (para más detalle ver la Sección 5.2) La comunicación es manejada por el sistema, de manera que cuando uno ejecuta una aplicación basada en SmartCore y en la misma red hay otra aplicación basada en SmartCore corriendo, el sistema descubre la existencia de otro participante (a través de multicast) y realiza la conexión entre ambos, para poder compartir lo que están

haciendo. La comunicación de objetos de las aplicaciones se realiza por medio de XML, y los objetos deben ser serializables para poder ser transmitidos por red. SmartCore abstrae la comunicación de tal manera que el usuario del sistema no se ve enfrentado a ningún cuadro de diálogo para poder iniciar/configurar las conexiones con el resto de los equipos, el usuario no tiene que saber los números IP de los otros participantes para interactuar con ellos, y el desarrollador que esté usando SmartCore como plataforma base sólo necesita saber el identificador de un participante para enviarle objetos por red a esa persona.

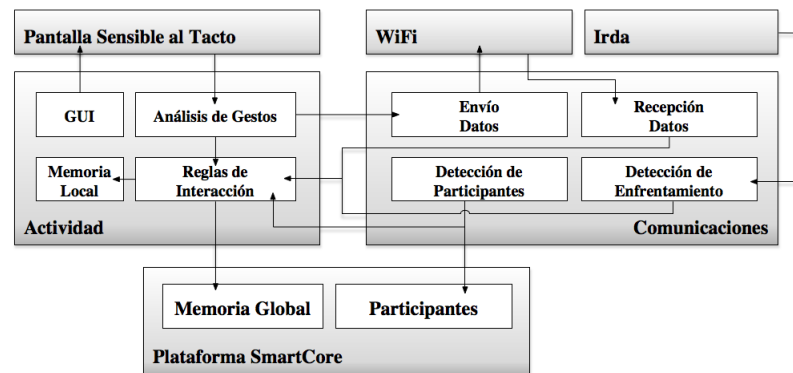


Figura 2: Diagrama de los módulos principales de la plataforma SmartCore (Fuente: [15])

SmartCore permite guardar la sesión actual (incluyendo todo el contenido que se ha creado) en un archivo con formato XML. La sesión puede ser posteriormente reconstruida cargando ese archivo en la aplicación.

El sistema tiene dos versiones: una para PC (usando el .NET Framework 2.0) y otra para dispositivos móviles (que utiliza el .NET Compact Framework 2.0). Ambas versiones comparten gran parte del código, son muy pocas secciones del código que difieren entre las dos versiones de la aplicación. Esto permite que las aplicaciones basadas en el sistema puedan ejecutarse en diferentes medios e interoperar entre sí.

SmartCore provee detección por cercanía de otros dispositivos que están ejecutando el mismo sistema por medio de IrDA (infrarrojo), lo que podría tener aplicaciones en sistemas donde la proximidad de los usuarios tenga un rol fundamental. En [41] se usa esta característica en una simulación donde los participantes toman roles de vendedores o compradores, y para poder intercambiar es necesario que los dispositivos estén frente a frente.

Las aplicaciones construidas a partir de SmartCore se basan en un sistema de modos para funcionar. Al ejecutar cualquier aplicación, en la barra inferior de la pantalla aparecen varios iconos que representan diferentes modos disponibles. Sólo puede haber un único modo activo a la vez. La separación por modos permite que en cada modo uno se puedan mostrar diferentes objetos en la aplicación, aceptar ciertos gestos como válidos, que los objetos en la pantalla tengan un comportamiento distinto, mandar determinados objetos por red e incluso actuar de manera diferente cuando se reciben ciertos objetos por parte de otros participantes.



Figura 3: Screenshot de SmartCore

SmartCore posee un sistema para poder reconocer gestos del stylus o el mouse. Cada gesto tiene asociado un evento (activado al reconocer el gesto), y condiciones para que los trazos que se hagan en SmartCore se reconozcan como un gesto en particular. Cuando se realiza un gesto en la pantalla (con el mouse o en la pantalla táctil del dispositivo según corresponda) SmartCore va almacenando los puntos hasta que se termina de realizar el gesto (cuando se levanta el stylus o el botón del mouse). Posteriormente, según la aplicación y el modo en el cual se esté trabajando, se analizan los puntos del movimiento para ver si se asemejan a las figuras de los gestos. Cuando un gesto es reconocido se llama al evento asociado a él. Los objetos básicos que se pueden crear en la aplicación a través de gestos son los nodos (que contienen páginas en su interior) y los trazos, como se ve en la Figura 3.

### 3. Requerimientos del sistema

Se debieron cumplir algunos requisitos básicos al desarrollar el proyecto. Los requisitos más relevantes se explican a continuación:

#### 3.1. Ejecución en dispositivos móviles

La aplicación debe ser poder ejecutada en dispositivos móviles, para usar sus características (como la movilidad y portabilidad) con el fin de favorecer el trabajo colaborativo, la interacción y la cooperación entre los integrantes (alumnos y profesores). Así se puede promover la interacción *face-to-face* de alumno-profesor y alumno-alumno. Estas interacciones son importantes para desarrollar habilidades colaborativas y de aprendizaje [42].

## 3.2. Arquitectura P2P

Las redes ad-hoc se caracterizan por la poca infraestructura necesaria para poder montarlas y por la facilidad para formarlas [43], además de permitir una topología descentralizada de red. Estas características son buenas para una sala de clase, puesto que el profesor evitaría gastar mucho tiempo de la clase en configurar la red para poder utilizar la aplicación. La descentralización permitiría que la aplicación sea utilizada por los alumnos en actividades donde no es necesario que el profesor esté conectado a la red ad-hoc para poder desarrollarlas.

## 3.3. Uso interfaz a mano alzada y compatibilidad de pantallas

Como el tamaño de la pantalla de los dispositivos móviles es menor que el de un computador o una Tablet-PC, la interfaz de la aplicación debe representar la información de forma tal de aprovechar el espacio disponible. El diseño de interfaz para una aplicación para PDA debe estar contenida en ella, y mostrar suficiente información para que la interfaz sea útil [44]. Tomando en cuenta lo anterior, la interfaz deberá estar basada en gestos, con la finalidad de minimizar el uso de widgets y tener más espacio disponible para ingresar contenido [14]. Si la información a mostrar no cabe en la pantalla, se debe utilizar un mecanismo basado en gestos para poder mostrar la información, aunque sea de manera parcial.

Existen investigaciones que sustentan que el modo más natural de ingreso de datos en un dispositivo con pantalla táctil es a través de *sketches* [14], imitando el uso del lápiz y papel [34]. El uso de *sketches* no limita el tipo de contenido que se puede crear a través del sistema, a diferencia de la entrada de datos por texto plano, donde se hace difícil hacer diagramas o gráficos explicativos.

Dado que la aplicación va a ser ejecutada en dispositivos con diferentes resoluciones de pantalla, la aplicación debe adaptarse a la pantalla del equipo actual. Se debe proveer un mecanismo que permite la compatibilidad entre los diferentes tamaños de pantalla de los diversos dispositivos presentes. Los diferentes tamaños de pantalla no deberían restringir la visualización del área de trabajo de la aplicación en los dispositivos con menor resolución.

## 3.4. Gestión de contenidos

El sistema debe proveer mecanismos para la gestión (creación, modificación, borrado y duplicación) de datos e información a ser presentados como contenidos para ser expuestos por el profesor en una sala de clase. Esos mecanismos pueden definir si proveen linealidad o no-linealidad a la presentación del contenido, en ese último caso se debe agregar una opción a la aplicación que permita definir qué contenidos visitar y en qué orden. Ese mecanismo debe ser simple de usar, de modo que pueda ser utilizado incluso en tiempo real. Gracias a eso, el profesor podrá adaptarse a la clase, de manera de poder agregar contenido si el nivel de la clase no es bueno, o sacar contenido si ya fue visto por los alumnos o es muy avanzado para ellos.

También los alumnos tienen mecanismos de acceso a la gestión de información de los contenidos del profesor bajo ciertas reglas. Para evitar que el contenido del profesor sea mo-



dificado en cualquier momento por alumnos, el profesor puede dar o revocar a los alumnos un permiso para poder modificar el contenido del profesor. Así el profesor puede tener ciertas ventanas de tiempo en las cuales sabe que sólo él puede manipular el contenido de la clase, aunque se debe proveer una forma de alertar al profesor cuando un alumno desea realizar un aporte o tiene una pregunta, para que la creación de contenido no sea un privilegio exclusivo del profesor.

La información se debe poder almacenar a disco en un formato estándar. Esto permite recuperar el contenido en posteriormente en otra sesión de la aplicación para poder continuar el trabajo en curso.

### **3.5. Toma de notas**

Tomar notas en clase ayuda al aprendizaje de los estudiantes, dado que ellos tienen que re-expresar lo que el profesor está explicando, con lo cual las ideas son reiteradas en la mente del estudiante, integradas a un nivel más profundo o recodificadas mentalmente en una forma más fácil de recordar [45].

En [30] se habla de un sistema que permite que los alumnos vean las notas de sus compañeros de clase, lo cual era bien valorado por ellos, además de mejorar la eficiencia del proceso de toma de notas. Adicionalmente, los alumnos encuentran positivo el poder visualizar las diapositivas del profesor cuando se están tomando notas [31]. MCPresenter debe permitir que los alumnos puedan tomar notas personales, viendo lo que ha escrito el profesor en el fondo, y compartir las anotaciones que realizan los alumnos en la clase.

### **3.6. Gestión de actividades pedagógicas para los alumnos**

En [25] se dice que es necesario tener cambios de actividades durante la clase para que los estudiantes mantengan su nivel de atención. Un cambio de actividad sería pasar de la clase expositiva a la realización de actividades pedagógicas por parte del profesor.

La aplicación debe permitir al profesor definir preguntas (ya sea en tiempo real como antes de la clase), con el fin de evaluar el aprendizaje de los estudiantes. Como el profesor tiene una idea del conocimiento de los alumnos puede adaptar de mejor manera la enseñanza de la clase. Además, los alumnos pueden recibir feedback acerca de las respuestas que van dando, ayudando a su proceso de aprendizaje.

Se debe poder mostrar información relativa al desempeño individual de los estudiantes. Así el profesor puede detectar si los alumnos están teniendo problemas, y los alumnos pueden conocer como ha sido su desempeño al responder las preguntas.

## 4. Solución implementada

A continuación se detallará cómo MCPresenter apoya a las prácticas pedagógicas, y las funcionalidades de la aplicación junto con una explicación de cómo ejecutarlas y cuál es el efecto que producen.

### 4.1. Actividades pedagógicas

Las actividades pedagógicas enriquecen el proceso de aprendizaje.

MCPresenter incluye varias de estas prácticas pedagógicas de manera de apoyar al profesor en el desarrollo de la clase. Las prácticas pedagógicas que MCPresenter apoya son:

- *Assessment*: La aplicación permite al profesor realizar preguntas a los alumnos, y el profesor puede ver en tiempo real cómo han ido respondiendo los alumnos a las actividades que él ha planteado. Esto le permite ver si hay estudiantes en la clase con dificultades para aprender o si los alumnos han aprendido.
- *Feedback*: Cuando los alumnos responden a las preguntas del profesor, reciben retroalimentación inmediata acerca de la respuesta dada en caso que el sistema pueda determinar si es correcta o no. Si no es así, el profesor es quien confirma la correctitud de la respuesta.
- *Reflection*: El sistema permite al alumno ver si las respuestas que ha dado a los problemas del profesor han sido correctas o no, para que puedan determinar sus fortalezas y debilidades.
- *Colaboración*: Los alumnos pueden formar grupos para responder preguntas de manera colaborativa, o se pueden generar instancias de toma de notas colaborativa entre alumnos. También se puede generar interacción entre profesor y alumnos, cuando los alumnos quieren complementar los contenidos entregados por el profesor o el alumno hace una pregunta al profesor usando el sistema.

### 4.2. Comentarios generales

La aplicación implementada, construida sobre SmartCore, tiene dos roles para los usuarios del sistema: profesor y alumno. Al estar basada en SmartCore, MCPresenter utiliza un sistema de modos para su funcionamiento. Existen algunos modos que sólo pueden ser utilizados por el profesor, otros de sólo pueden ser usados por los alumnos, y el resto pueden ser utilizado por cualquiera de los dos roles.


Al tomar uno de estos roles sólo se muestran en la pantalla los modos que están asociados a ese rol. Los roles anteriormente mencionados son excluyentes, no se permite ser profesor y alumno a la vez. Sólo se permite un dispositivo con el rol de profesor a la vez, mientras que no hay limitaciones en el número de dispositivos con el rol de alumno.

La plataforma SmartCore provee funcionalidad de guardar la sesión con formato XML a un archivo en disco, cargar sesión e iniciar una nueva sesión. Esas opciones están presentes

en la mayoría de los modos de la aplicación.

Los modos disponibles se muestran como botones en la parte inferior de la pantalla. Haciendo click en sobre el icono de un modo active tal modo. El modo activo se muestra con un icono de mayor tamaño que el resto. Algunos modos tienen una barra de herramientas adicional que se puede expandir al hacer click en el icono del modo activo (si es que tiene un triángulo), a esa barra de herramientas se le denominará “*barra de opciones adicionales del modo*” . La Figura 7 muestra un ejemplo de expansión de la barra.

### 4.3. Área del pizarrón

El modo del “área del pizarrón” (accesible en la barra de menú que está en el costado inferior de la pantalla, haciendo tap en el icono ) permite al docente tener un espacio donde puede escribir libremente. Tanto el profesor como los alumnos pueden entrar a este modo. Por defecto los alumnos no tienen habilitada la escritura en esa área de trabajo. Para que los alumnos puedan escribir ellos deben enviar, a través del sistema, una solicitud de escritura al profesor. El profesor es quien decide a cuál alumno se le da o quita el permiso para escribir en el área del profesor.

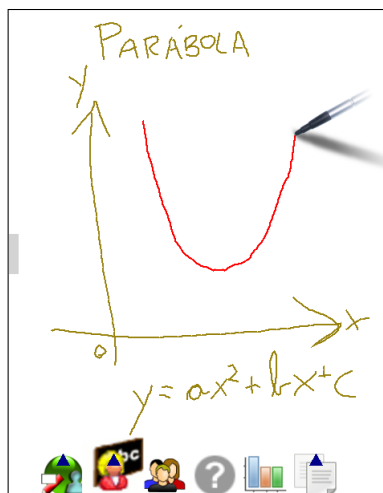




Figura 4: Modo del área del profesor (vista de profesor)

Para poder convertirse a profesor, se debe hacer un tap en el icono del modo de “área del pizarrón”, y hacer tap en el icono , con lo cual el dispositivo actual toma el rol de profesor. Los otros dispositivos asumen automáticamente el rol de estudiante, incluso si se unen a la sesión en un momento posterior.

En este modo, cuando los estudiantes tienen permiso para escribir, el sistema no permite que puedan borrar los dibujos de otras personas, únicamente pueden borrar sus propias creaciones. El profesor, en cambio, está habilitado para borrar dibujos de cualquier persona.

A continuación se describen las funcionalidades más relevantes de este modo.

#### 4.3.1. Dar permiso a los alumnos para escribir

Como se dijo anteriormente, el profesor es quien gestiona los permisos de escritura de los estudiantes. El profesor dispone de una barra al lado izquierdo de la pantalla donde se puede ver la lista de los alumnos conectados a la sesión actual, junto con iconos representando las acciones que se pueden realizar con ellos. Esa barra es denominada la *barra de los estudiantes*. Para no interferir visualmente en el área de trabajo, inicialmente esa barra está contraída. El profesor puede hacer click en esa barra cuando está en ese estado para expandirla (ver Figura 5), mostrando todos los alumnos que están conectados, junto a un icono al lado del nombre del dispositivo del alumno, que puede ser rojo (el alumno no ha pedido autorización para escribir), amarillo (el alumno ha realizado una petición pero no se le ha conferido el permiso) o verde (el estudiante sí puede escribir en el área del profesor). El profesor puede entonces seleccionar a un alumno de la lista para dar o quitarle el permiso de escritura. Para ocultarla se debe hacer click en la opción “Ocultar” o “Hide” (mostrada con el icono ) , que se muestra en la última posición en esa barra.

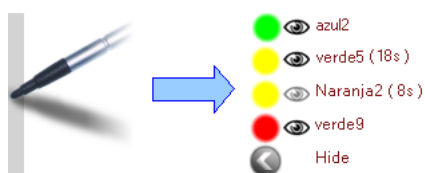


Figura 5: Expansión de la barra de lista de estudiantes

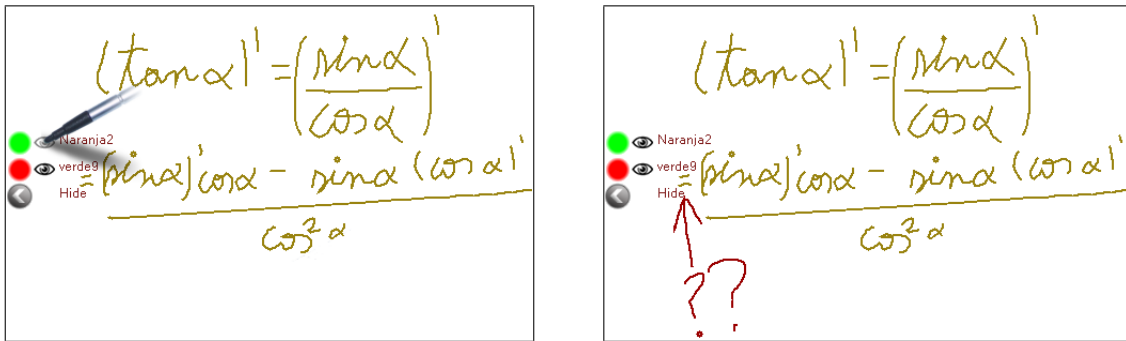
Para evitar que el profesor, frente a peticiones de varios alumnos, no entregue permiso a los alumnos que han pedido la autorización con más anticipación, al lado del nombre del dispositivo de la persona en la barra de estudiantes se muestra hace cuánto tiempo (en minutos y segundos) fue realizada la respectiva petición (como en la Figura 5, donde los dispositivos *verde5* y *Naranja2* han pedido permiso hace 18 y 8 segundos, respectivamente). Aquella lista de alumnos que han pedido permiso se ordena en forma decreciente en función de ese tiempo de espera, con el fin de mostrar la información de manera más ordenada para el profesor.

Si varios alumnos reciben permiso del profesor para escribir en el área del pizarrón, se puede generar una instancia de colaboración en un área de trabajo compartida, hasta que el profesor decida quitarles el permiso de escritura.

#### 4.3.2. Filtrar trazos por estudiante

Dado que en una misma página potencialmente muchos alumnos pueden escribir aportes y comentarios, se puede dificultar la visibilidad de ella. Para evitar ese problema, el profesor dispone de una funcionalidad que le permite ocultar o mostrar los trazos de determinados estudiantes. Cabe notar que ese cambio es local al dispositivo del profesor, la vista en los



dispositivos de los estudiantes no se ve afectada.




(a) Haciendo tap en el icono para mostrar/ocultar trazos

(b) Mostrando trazos que antes estaban ocultos

Figura 6: Utilización de barra de estudiantes para filtrar visibilidad de trazos de estudiantes

En la misma barra de estudiantes, aparece al lado del nombre el icono de un ojo (como se ve en la Figura 5, donde los iconos dan información que los trazos del dispositivo *Naranja2* están siendo ocultados). Al presionar sobre el icono  se ocultan los trazos de esa persona (ver Figura 6). Al presionar sobre el icono  se hacen visibles los dibujos de aquella persona. Cuando se ocultan o muestran los dibujos de una persona, esos dibujos parpadearán por unos segundos para poder identificar cuáles son los dibujos que están siendo afectados.

#### 4.3.3. Traer alumnos a página actual

Si el profesor desea que todos sus alumnos se encuentren en la misma página que él (por ejemplo, cuando desea realizar una explicación sobre un tópico importante y necesita que la atención de los estudiantes se centre en la página actual en la que se encuentra), él debe expandir la “barra de opciones adicionales del modo” (ver Figura 7), y posteriormente hacer click en el icono , con lo cual todos los estudiantes son cambiados a la misma página que está el profesor en ese momento, independiente de la página y el modo en que ellos se encuentren.

#### 4.3.4. Pedir permiso de escritura al profesor

Un alumno puede usar este modo para realizar un aporte a la clase (complementando lo que ha escrito el profesor o algún otro alumno) o indicar al profesor que no ha entendido alguna explicación. Si un alumno desea escribir en el área del profesor, él debe arrastrar el stylus en un área vacía de la pantalla, con lo cual el profesor recibe una “petición de escritura” por parte del alumno, a la espera que el profesor de la autorización respectiva para poder escribir.

Cuando al alumno se le da o quita permiso de escritura, se muestra un *awareness* visual (ver Figura 8) por unos segundos, independiente del modo en que esté el alumno. Posterior

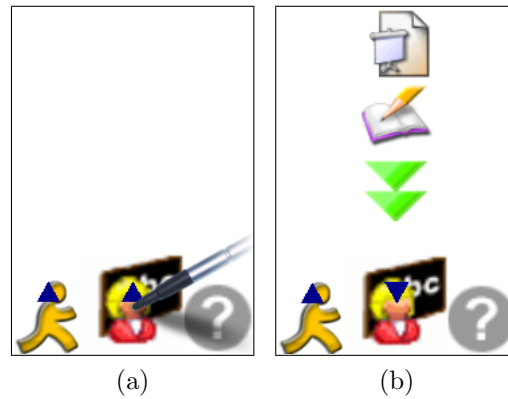
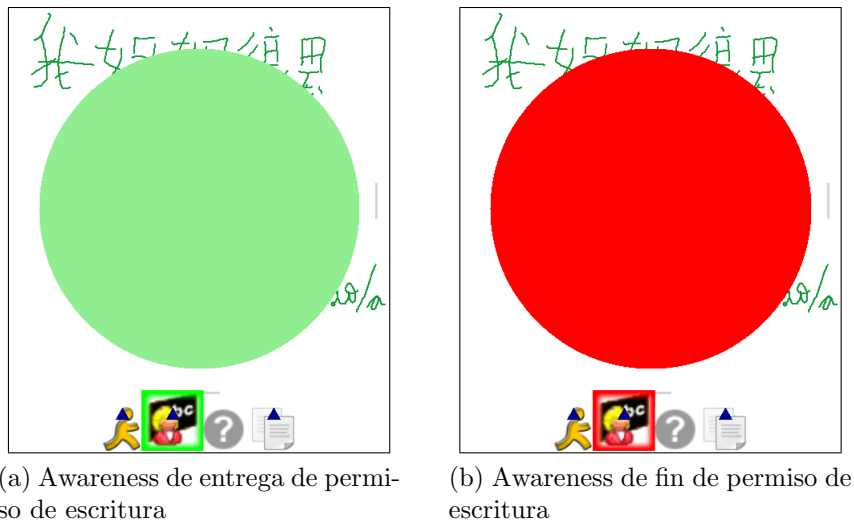


Figura 7: Al hacer un tap en el icono del modo actual (a) se muestra la “barra de opciones adicionales del modo” (b).






(a) Awareness de entrega de permiso de escritura

(b) Awareness de fin de permiso de escritura

Figura 8: Awareness visual que recibe un estudiante cuando el profesor le da o le quita el permiso de escritura.

a ese aviso, el alumno puede saber si le dan dado autorización para escribir en el área del profesor según cuál sea el icono del modo en su dispositivo:


- Si el icono contiene un borde rojo () , el profesor no le ha dado o le ha quitado el permiso (como en Figura 8b). Éste es el estado inicial de los alumnos cuando hay un profesor en la sesión actual.
- Si el icono tiene un borde verde () , el alumno sí tiene permiso para poder escribir en el área del profesor (como en Figura 8a). El único que puede promover al alumno a este estado es el profesor.
- Si el icono no tiene ningún borde alrededor () , significa que no hay profesor presente

en la sesión en ese instante. Esto se puede dar cuando nadie ha sido declarado como profesor (por ejemplo, al inicio de la sesión), cuando el profesor se sale del sistema, o después que el profesor se desconecta de la sesión.

Las acciones del alumno en este modo cuando no tiene autorización para escribir se limitan a seleccionar/ deseleccionar objetos y a navegar a través de los nodos del área de trabajo. Bajo ese estado, no se le permite al alumno borrar, mover, cambiar de tamaño o duplicar elementos.

#### 4.4. Toma de notas

Los estudiantes pueden ir al “modo de notas” para realizar anotaciones personales encima de la página actual, compartir a otros lo que han escrito y ver las notas que los demás han decidido compartir. El modo además posibilita la toma de notas colaborativa entre estudiantes.

Estando en el “modo del pizarrón”, se deben expandir la “barra de opciones adicionales del modo” (ver Figura 7b), y se debe hacer click en el icono  para poder activar el “modo de notas”. Haciendo tap sobre el icono nuevamente vuelve al “modo del pizarrón”. Para tomar notas, sólo basta dibujar en el área de trabajo. Únicamente en este modo se pueden ver, modificar y/o borrar las anotaciones. Cuando un dispositivo toma el rol de estudiante, pasa automáticamente a este modo.

El contenido sobre el cual se está tomando notas se dibuja al fondo de la página actual, para evitar que los alumnos gasten tiempo en copiar lo que ha escrito el profesor o en cambiarse de página/modo sólo para buscar lo que había escrito el profesor.

Este modo contiene una “barra de participantes” (ver Figura 9) similar a la barra del “modo del pizarrón” que ofrece funcionalidades relacionadas con las notas, que se irán explicando a lo largo de esta sección. Al igual que la otra barra, se puede contraer para no obstaculizar la vista.

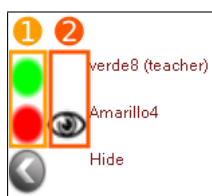


Figura 9: La barra de participantes contiene botones para compartir notas (1) y para ver notas de otros (2).

Al guardar la sesión en formato XML, las notas también son almacenadas, pudiendo ser recuperadas al cargar la sesión.

#### 4.4.1. Escribir notas

Para poder realizar anotaciones basta con hacer trazos con el stylus en la pantalla. Este modo permite adicionalmente mover, copiar y borrar. El borrado sólo se aplica a los trazos propios, no hay forma de poder eliminarlos si los dibujos son de otra persona. Por defecto, las notas no son visibles por el resto de los participantes.

#### 4.4.2. Compartir notas

Como se dijo anteriormente, este modo posibilita enviar a los demás participantes las anotaciones realizadas. Dentro del “modo de notas” se puede ir a la barra de participantes (ver Figura 9), y seleccionar con quién se van a compartir las notas. Ese mismo menú es utilizado a la hora de cesar de compartir de notas con determinados participantes.

Si se está compartiendo las notas con una persona, aparece un icono de círculo verde en la misma fila que su nombre, en caso contrario aparece el icono de un círculo de color rojo. Presionando sobre el círculo correspondiente a una persona ocasionará que se inicie o termine de compartir notas con esa persona.

#### 4.4.3. Ver notas de otros

Si alguien ha compartido notas con una determinada persona, en el dispositivo de esa persona parpadeará la barra de participantes si es que está contraída, para dar aviso al usuario de dicha situación.

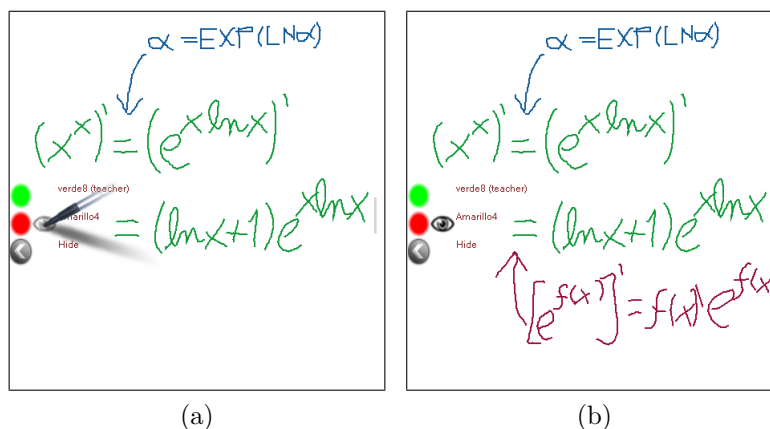




Figura 10: Haciendo click en el icono de ver notas (a) muestra (u oculta) las notas de un participante (b)

Por medio de la barra de participantes (ver Figura 10) se pueden mostrar/ocultar las notas que los demás han decidido compartir con el dispositivo actual. Tanto al seleccionar la opción de mostrar la notas de una persona o de ocultarlas, los trazos de esa persona parpadearán por unos segundos como una forma de dar *awareness* al usuario. Si un participante





desea ver las notas que le han compartido, se tiene que ir al “barra de participantes” y seleccionar las notas de quién quiere ver. En ese menú aparece los nombres dispositivos del resto, junto con un icono de un ojo  que indica esa persona que le está compartiendo notas con esa persona, en caso contrario aparecerá el icono de un ojo más tenue .

Si una persona ha decidido compartir sus notas con alguien, esa persona podrá ver todas las notas que ha escrito en todas las páginas del área de trabajo, incluso las modificaciones que haya realizado a las notas después de haberlas compartido. No es posible en la versión actual de la aplicación compartir las notas de una página y no hacerlo con las de otra página.

Si dos o más estudiantes están compartiendo sus notas entre sí, se puede generar una dinámica de toma de notas colaborativa. Dado que en esa configuración lo que escribe uno lo ve el resto de sus compañeros, los estudiantes se pueden organizar para tomar notas. Por ejemplo, un estudiante puede tomar notas mientras otro presta atención al profesor.


## 4.5. Manejo de presentaciones

El “modo Slideshow” (accesible presionando el icono , en la “barra de opciones adicionales del modo” de pizarrón) permite al profesor la creación de presentaciones, que contengan un número variable de *slides* o diapositivas. Las diapositivas pueden contener el material que el profesor desea exponer a la clase, y será enviado a los alumnos conforme los dispositivos en red se sincronicen. Esto permite al profesor preparar material expositivo antes o durante la clase (en tiempo real). Para mantener un orden y para poder individualizar de manera más fácil las diapositivas, cada una de ellas tiene un número asociado que corresponde a su posición en la presentación (la primera diapositiva lleva un 1, la segunda, un 2, etc.),

Este modo además permite ingresar al interior de las diapositivas. Para ingresar a una diapositiva, seleccionar una (debe haber sólo una seleccionada), y presionar en el icono  para poder ver lo que hay en el interior de ella, y editar su contenido por medio del “modo del pizarrón”: el profesor puede escribir libremente, mientras que los alumnos necesitan tener el permiso del profesor para hacerlo.

La totalidad de las funcionalidades nombradas en esta sección pueden ser utilizadas por los profesores. El caso de los alumnos es diferente: ellos pueden hacer scroll/zoom en la vista de diapositivas (ver Sección 4.5.6).

### 4.5.1. Creación de presentaciones

Para crear una presentación, se debe ir al “modo Slideshow”, y se debe hacer un gesto de “L” como se ve en la Figura 11. Eso creará un nodo de presentación en la página actual. El profesor puede ingresar al interior de dicho nodo, haciendo un click sobre la presentación recién creada y presionando en el icono  para poder agregar/modificar/eliminar/reordenar diapositivas.

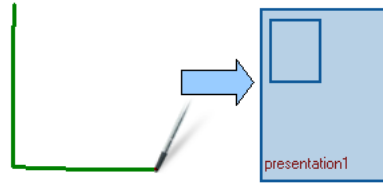


Figura 11: Haciendo el gesto de una “L” se crea una presentación en la página actual

#### 4.5.2. Creación de diapositivas

Las presentaciones creadas inicialmente contienen una sola diapositiva. Si se desean agregar más diapositivas, se debe ir al interior de la presentación, y hacer doble tap en cualquier lugar del fondo de la pantalla (cuando la presentación no tiene ninguna diapositiva) o hacer doble tap a los costados de las diapositivas existentes (ver Figura 12), para crear una diapositiva vacía en el lugar que se desee. Notar que al hacer el primer tap va a aparecer una línea negra (ver Figura 12a) para indicar que el gesto se ha reconocido y que la diapositiva nueva se va a crear en esa posición.

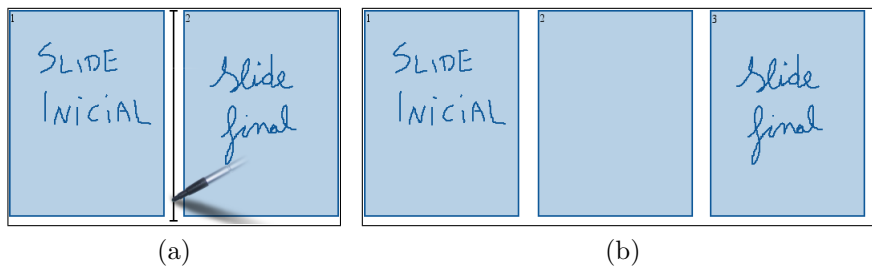


Figura 12: Haciendo doble tap entre diapositivas (a) o al costado de ellas, se crea una diapositiva vacía en el lugar deseado (b)

#### 4.5.3. Mover diapositivas

Para poder reubicar diapositivas dentro de la misma presentación, basta con seleccionar una o varias de ellas y arrastrarlas hacia el lugar deseado (ver Figura 13). Al estar arrastrando aparecerá una barra negra vertical que muestra la posición en la que se insertarán las diapositivas que se están moviendo, como en la Figura 13b. Notar que cuando las diapositivas están siendo arrastradas se dibuja sólo su borde y no su contenido, para poder ver qué es lo que hay debajo de ellas.

#### 4.5.4. Duplicación de diapositivas

Para copiar diapositivas (ver Figura 14), se debe hacer un dibujo de un punto con el stylus hasta que aparezca un círculo rojo, con lo cual posteriormente se debe tocar con el stylus los elementos que se desean copiar.

Posteriormente, para poder pegar diapositivas en la presentación actual, se debe realizar un punto denso (como en la Figura 14c). A diferencia del gesto de copiado, cuando aparezca

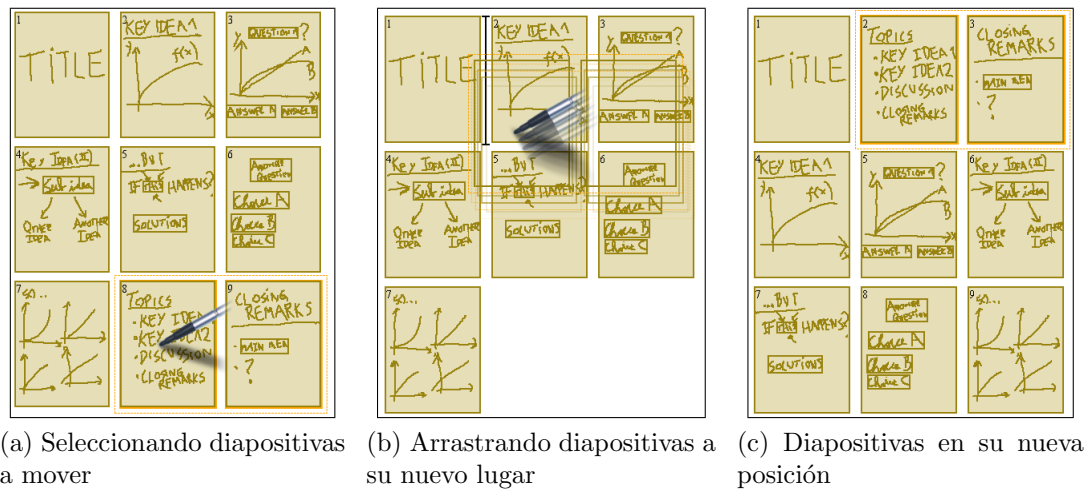


Figura 13: Cambiando diapositivas de lugar dentro de la presentación

el círculo rojo se debe soltar el stylus, ocasionando que las diapositivas anteriormente pegadas se agreguen a la presentación. Si se desea que las diapositivas a pegar queden en una posición en particular en la presentación (por ejemplo, entremedio de otras diapositivas, o adelante o atrás de una diapositiva específica), se debe hacer el gesto al costado de alguna diapositiva existente. En caso contrario, las diapositivas se anexarán al final de la presentación.

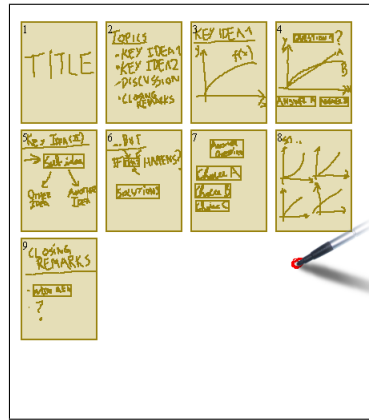
También es posible copiar de diapositivas de una presentación a otra: se pueden copiar las diapositivas de una presentación, navegar hasta otra presentación y hacer el gesto de pegado en esa página.

#### 4.5.5. Eliminación de diapositivas

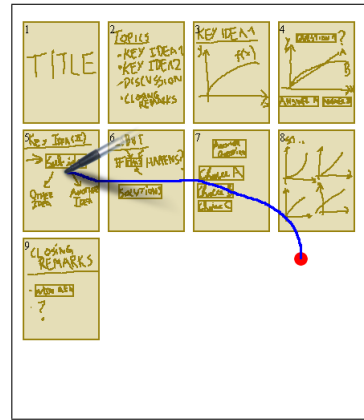
Si se desea eliminar diapositivas, se debe hacer el gesto de una cruz, iniciando el movimiento del stylus desde el interior de una diapositiva (como en la Figura 15). Al soltar el stylus, todas las diapositivas que estén en el área de la cruz serán eliminadas. Una vez que las diapositivas han sido eliminadas, no se pueden recuperar.

#### 4.5.6. Scroll/zoom en la vista de diapositivas

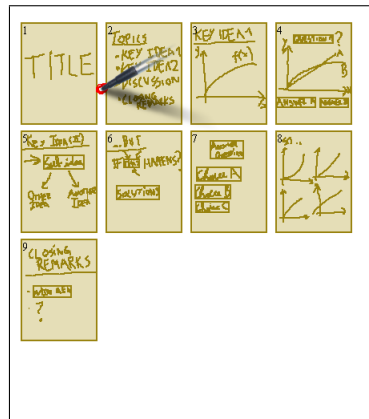
Cuando se tengan muchas diapositivas en la misma presentación, se puede tornar difícil de ver o leer el contenido de las diapositivas (especialmente en una PDA), por lo cual el modo provee la opción de realizar zoom y desplazamiento. El usuario debe mantener presionado el stylus en la pantalla en un área vacía de la página (pues si parte desde el interior de una diapositiva se puede detectar el gesto de borrado), y moverlo en dirección horizontal para aumentar/disminuir el nivel de zoom, y moverlo verticalmente para desplazarse hacia arriba/abajo, como se puede apreciar en las Figuras 16 y 17. Esta funcionalidad puede ser utilizada tanto por el profesor como los alumnos, en el “modo de pizarrón” y el “modo Slideshow”.



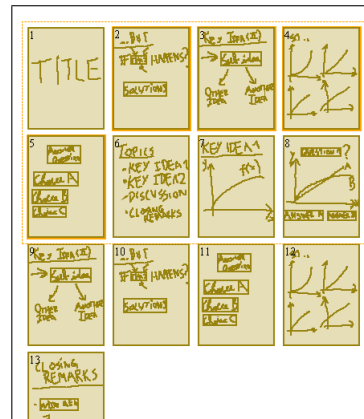
(a) Se hace un punto denso para activar el gesto de copiado/pegado.



(b) Se arrastra el stylus, tocando las diapositivas que se desean copiar



(c) Haciendo el punto denso para pegar



(d) Las copias de las diapositivas se ubican en la posición deseada

Figura 14: Copiado y pegado de diapositivas

#### 4.5.7. Importar diapositivas de PowerPoint

El profesor puede importar como presentaciones de MCPresenter las presentaciones PowerPoint que tenía creadas con anterioridad, evitando que el profesor tenga que crear desde cero su contenido. Esta funcionalidad no está disponible en la versión PDA de la aplicación, y en la versión PC funciona si en el equipo está instalado PowerPoint.

Para importar presentaciones PowerPoint (ver Figura 18), se debe hacer tap en el botón de “Importar PPT” (como se ve en la Figura 18a), presente en la “barra de Archivo”. Aparecerá un cuadro de diálogo donde se tiene que seleccionar el archivo PPT que se desea importar a la sesión actual. Ya seleccionado el archivo a importar, se crea en la página actual una presentación cuyas diapositivas representan a las diapositivas de la presentación PowerPoint, o se anexan las diapositivas importadas a la presentación de la página actual.

Las diapositivas cargadas desde un archivo PowerPoint se importan como imágenes, no

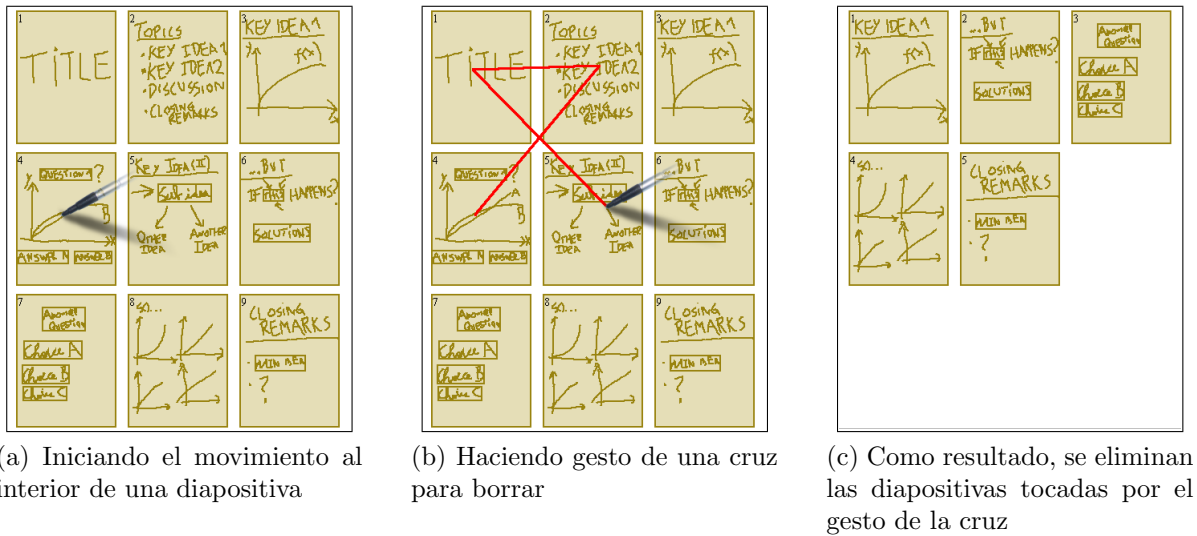


Figura 15: Borrado de diapositivas

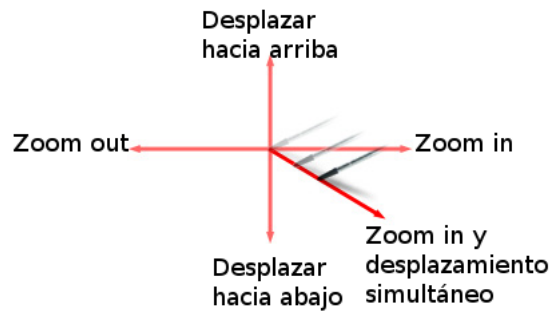
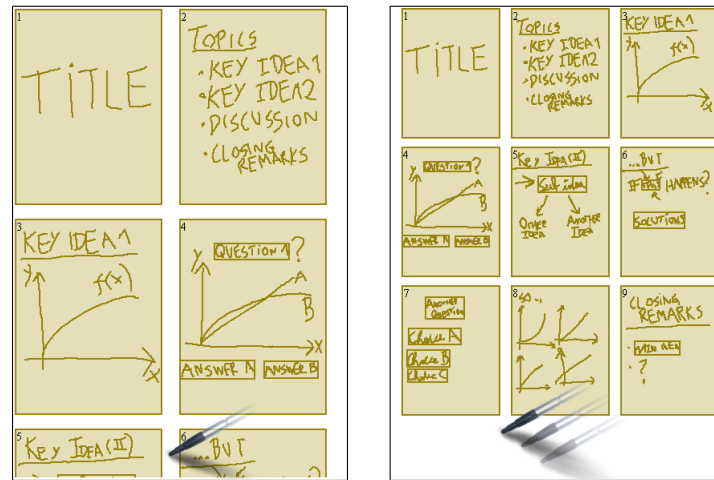


Figura 16: Arrastrando el stylus en la vista de diapositivas permite cambiar el zoom y desplazar sin necesidad de gestos adicionales. (Fuente imagen: [15])

siendo posible interactuar con los elementos individuales de las diapositivas (por ejemplo, un texto o un gráfico). Lo que sí es posible es cambiar el tamaño de aquellas imágenes en la pantalla, como se puede ver en la Figura 19. Ese cambio modifica el zoom de la página actual, por lo cual los trazos y nodos son redimensionados, permitiendo tanto a alumnos como profesores de disponer más espacio para escribir o escribir con mayor precisión.

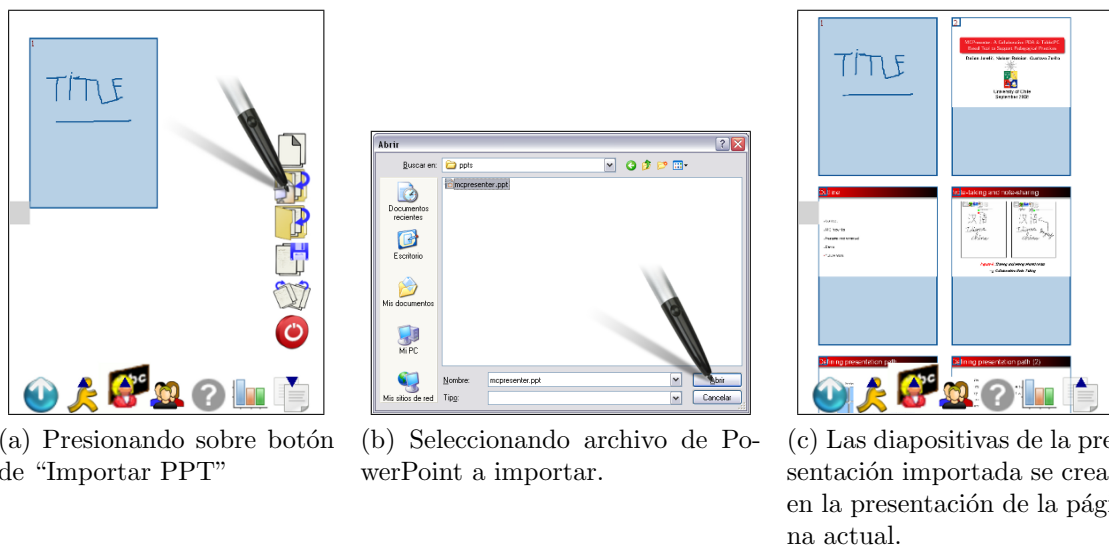
## 4.6. Edición de recorridos

El modo de “Definición de recorridos” permite al profesor seleccionar las diapositivas o nodos que desea mostrar a los alumnos, en el orden que él desee, constituyendo un *recorrido* de la presentación. Un recorrido consiste en una secuencia ordenada de nodos, y su utilidad radica en poder cambiar la forma en que se visitan las diapositivas o nodos, con lo cual profesor puede ajustar su enseñanza según la clase. En la aplicación un recorrido es representado visualmente por una especie de nodo que contiene almacenada la secuencia de nodos que componen el recorrido.



(a) Antes de cambiar el zoom (b) Arrastrando el stylus para cambiar el zoom de las diapositivas

Figura 17: Manejo de zoom en la vista de diapositivas



(a) Presionando sobre botón de "Importar PPT"

(b) Seleccionando archivo de PowerPoint a importar.

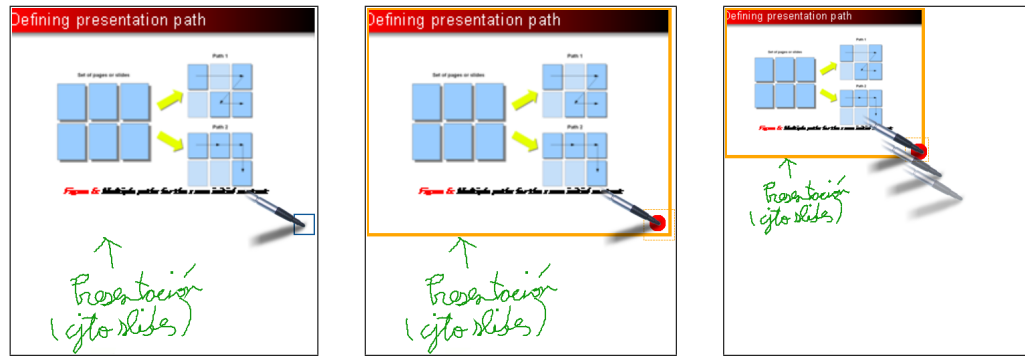
(c) Las diapositivas de la presentación importada se crean en la presentación de la página actual.

Figura 18: Importación de presentación PowerPoint al área de trabajo

A este modo no se puede entrar de manera directa. Para poder ingresar, se deben seguir las instrucciones de la sección 4.6.2.

#### 4.6.1. Creación de recorridos

Para crear un recorrido, estando en el "modo Slideshow", se debe hacer el gesto de una flecha horizontal (ver Figura 20) para crear uno en la página actual. No hay restricción en la cantidad de recorridos que se puedan crear (salvo limitaciones de memoria del dispositivo



(a) Presionando sobre el extremo inferior izquierdo de la imagen de la diapositiva importada. (b) Presionando sobre el círculo rojo que aparece. (c) Al mover el stylus se cambia el zoom de la página actual.

Figura 19: Cambio de zoom de imagen de diapositiva importada

que ejecute la aplicación). El recorrido es susceptible a ser movido, borrado, etc.

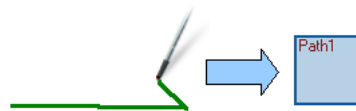



Figura 20: Haciendo de una flecha, se crea un recorrido en la página actual

#### 4.6.2. Edición de recorridos

Una vez creado el recorrido, hay que hacer un click en el recorrido para seleccionarlo y hacer click en el icono  que aparece cerca del extremo inferior derecho de él para poder editar su contenido. Esta acción ocasionará que la aplicación active el modo de “Definición de recorridos”, y muestre los nodos que componen a ese recorrido (las diapositivas son consideradas como nodos). En dicho modo la pantalla está dividida en dos mitades: la mitad izquierda muestra un árbol jerárquico de todos los nodos del área de trabajo, y el lado derecho muestra una lista ordenada de los nodos que pertenecen al recorrido que se está editando. Para poder identificar qué recorrido se está editando, en la parte superior de la mitad derecha aparece el nombre del recorrido (el nombre del recorrido lo define el sistema, no se puede modificar por el usuario).

Un recorrido recién creado inicialmente estará vacío. Para modificar un recorrido las acciones posibles a seguir son:

- Agregar un nodo al recorrido: se arrastra un nodo desde el lado izquierdo de la pantalla al lado derecho (ver Figura 21).
- Eliminar un nodo del recorrido: el nodo que se desea eliminar se arrastra desde el lado derecho hasta cualquier lugar del lado izquierdo.

- Cambiar el lugar de un nodo dentro del recorrido: se arrastra tal nodo dentro del mismo lado derecho.
- Agregar diapositivas de una presentación al recorrido: en el “modo Slideshow” se debe arrastrar una presentación hacia el recorrido y soltarla. Esto ocasionará que todas las diapositivas contenidas en la presentación se agregan al final del recorrido.

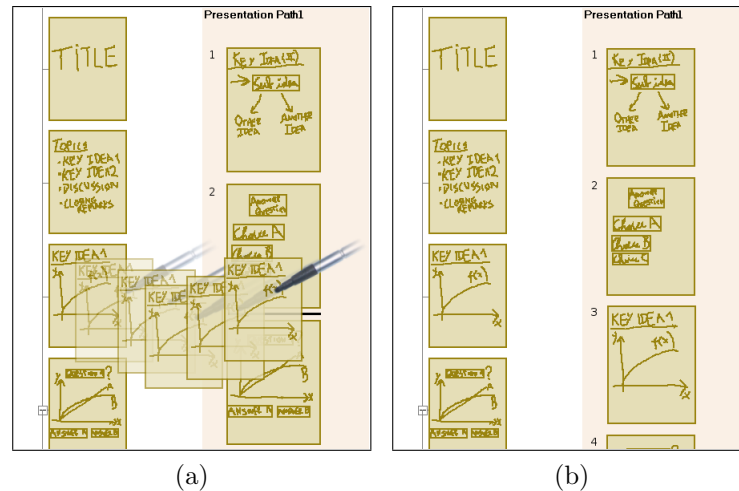


Figura 21: Arrastrando una diapositiva (a) para agregarla al recorrido (b).

Para poder identificar de manera visual en qué posición va a quedar el nodo al estar agregándolo al recorrido o cambiándolo de lugar dentro del mismo recorrido, se muestra una línea negra (como en Figura 21a) que indica la posición tentativa del nodo.

#### 4.6.3. Contracción/expansión de árbol de nodos

Si se tienen muchos nodos en el espacio de trabajo el árbol del lado izquierdo se puede tornar muy grande y complejo. Para facilitar la visualización, al lado izquierdo de los nodos que tienen nodos hijos aparece un signo más (+) o menos (-), que cuando son presionados muestran u ocultan a los nodos hijos de aquellos nodos, respectivamente (ver Figura 22).

#### 4.6.4. Scroll/zoom

Adicionalmente, en ambos lados de la pantalla se puede aplicar el gesto de zoom y desplazamiento (análogo al explicado en la Figura 16). Para usar el gesto hay que arrastrar el stylus, partiendo desde una posición vacía (es decir, que no haya un nodo en esa posición inicial, porque al mover el stylus provocará que se arrastre el nodo). Cabe notar que el factor de zoom y desplazamiento es independiente entre las dos mitades de la pantalla.

#### 4.6.5. Ir a página de un nodo

Al estar viendo los nodos en este modo puede surgir la necesidad de visitar un nodo en particular para poder verlo con más detalle o editar lo que está escrito en él. Haciendo



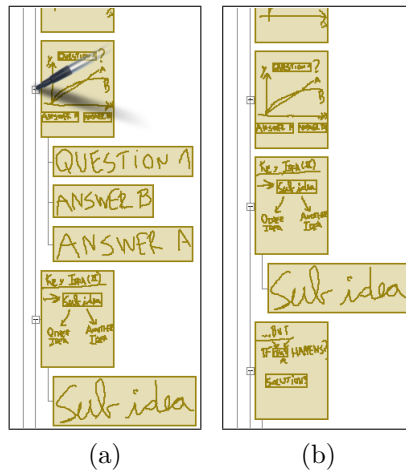


Figura 22: Al presionar sobre el recuadro al costado de un nodo (a), esa rama del árbol se contrae (b) o expande, según sea el caso.

doble tap en cualquiera de los nodos dibujados en la pantalla (tanto para los nodos del lado izquierdo como los del lado derecho) la aplicación va a la página correspondiente (ver Figura 23), y la aplicación cambia al “modo de pizarrón”.

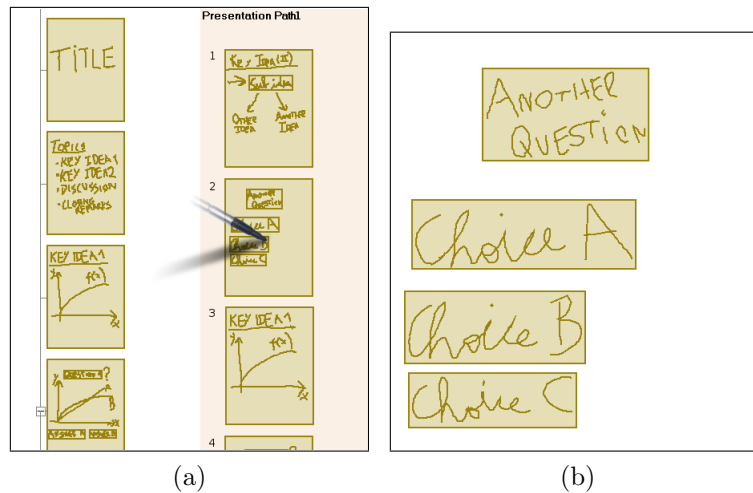



Figura 23: Haciendo doble tap sobre un nodo (a) hace que la aplicación se vaya a esa página (b)

#### 4.6.6. Definición de recorrido activo

El recorrido activo es el recorrido utilizado para navegar por los nodos de la sesión actual. Si bien se pueden tener varios recorridos en la misma sesión, sólo puede haber un recorrido activo a la vez, que se indica con un icono , como en la Figura 24. Para poder fijar un recorrido como activo, se debe seleccionar el recorrido en el “modo Slideshow”, y seleccionar

el icono ★. Cuando un recorrido pasa a ser activo, si hay algún otro recorrido que era activo, pasará a ser inactivo de manera de no tener más de un recorrido activo a la vez.

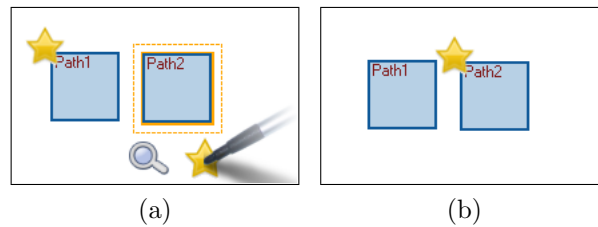


Figura 24: Fijando un recorrido como activo (a), automáticamente el antiguo recorrido activo deja de serlo (b).

#### 4.6.7. Navegación usando el recorrido actual

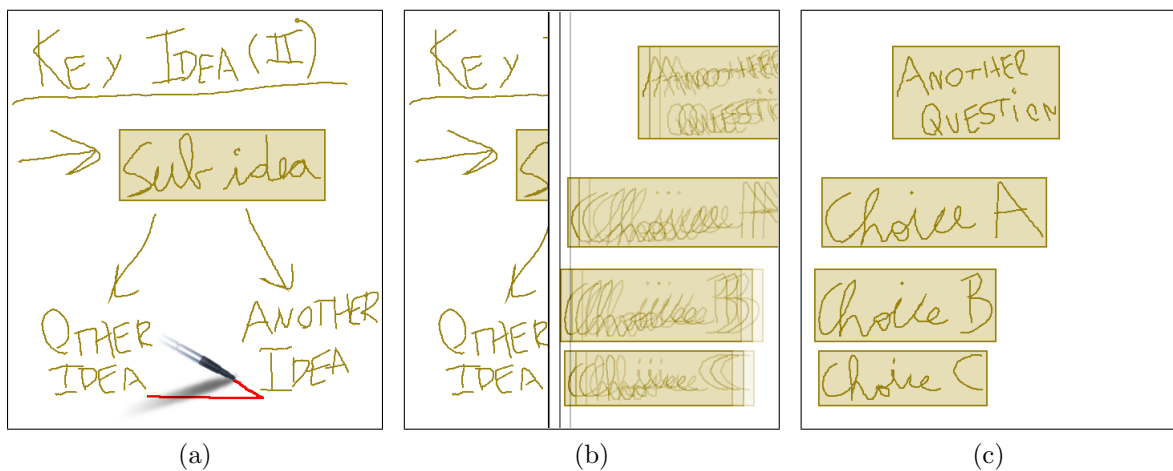


Figura 25: Haciendo el gesto de la flecha en una de los nodos del recorrido (a), se produce una transición (b) hacia el nodo anterior o siguiente en el recorrido (c), según el sentido de la flecha

Cuando ya se ha definido un recorrido, haciendo un gesto de una flecha horizontal en el “modo del pizarrón” provoca que se cambie de diapositiva o nodo según el recorrido activo y la diapositiva actual:

- Cuando no hay recorrido activo, si se hace el gesto dentro de una diapositiva, la aplicación cambia a la diapositiva anterior o siguiente, dependiendo del sentido de la flecha del gesto.
- Si se ha definido un recorrido activo, se avanza o retrocede en el recorrido según el sentido de la flecha del gesto, independiente si la página actual es una diapositiva o no.

La transición se realiza por medio de animación que muestra la nueva página poniéndose encima de la página actual (como se ve en la Figura 25).

## 4.7. Definición de grupos

Como una manera de favorecer el trabajo colaborativo entre los alumnos, el profesor puede formar grupos de trabajo. Así, los alumnos pueden responder de manera grupal las preguntas que el profesor les mande (para más detalles, ver Sección 4.8.3).

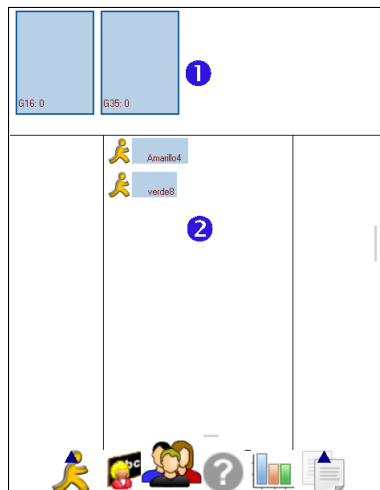


Figura 26: Modo de grupos

Este modo, que solamente puede ser utilizado por el profesor, define dos áreas especiales en la pantalla : un área de grupos y otra de estudiantes. En el área de grupos se pueden ver, crear y eliminar grupos, mientras que en el área de estudiantes se pueden ver los estudiantes que hay en la sección actual. La interacción entre las dos áreas permite la definición de grupos de trabajo.

Cuando hay varios grupos se puede utilizar el gesto de desplazamiento para poder navegar por los grupos. De igual manera, se puede utilizar el gesto para poder ver la lista de participantes. El gesto es reconocido cuando se inicia el movimiento del stylus en la parte inferior del área de grupos o en el costado izquierdo del área de estudiantes.

Para entrar en el modo de creación de grupos, se debe hacer click en el icono  de la barra de modos.

### 4.7.1. Creación de grupos

Para poder crear grupo, se debe hacer el gesto de una “L”, iniciando el movimiento en el recuadro superior de la pantalla (el área de grupos). Se creará un grupo con su número identificador (ver Figura 27).

### 4.7.2. Asignación de estudiantes a un grupo

Ya con los grupos creados, el siguiente paso es asignar integrantes a ellos. Para ello, se deben seleccionar a los participantes, arrastrarlos hasta el grupo correspondiente, y soltar el

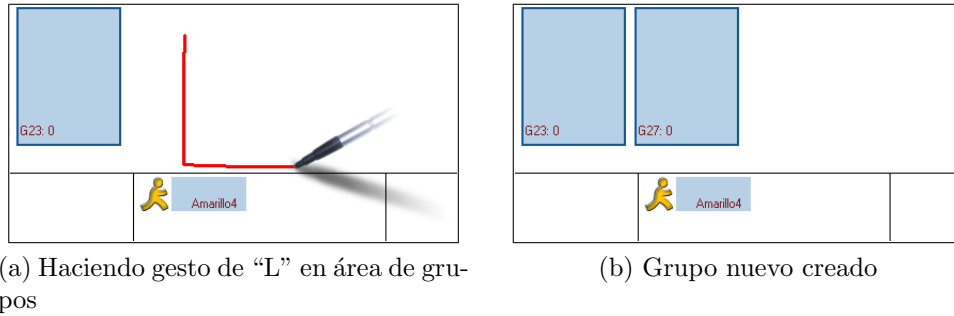


Figura 27: Creación de grupos

stylus (ver Figura 28).

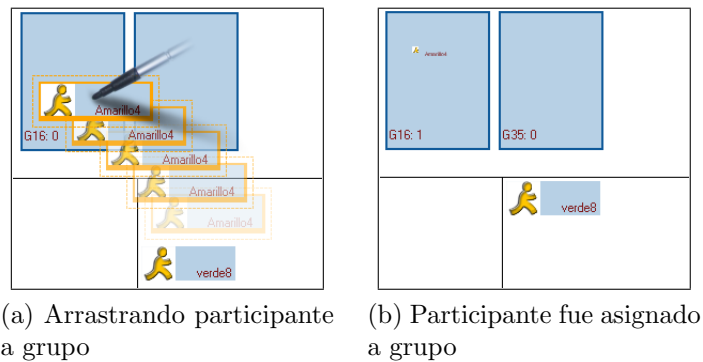


Figura 28: Asignación de grupos

Haciendo un tap en un grupo, éste se muestra con un borde adicional para indicar que está seleccionado. Cuando un grupo está seleccionado se muestra en la parte central de la pantalla los integrantes que componen a ese grupo (ver Figura 29). En ese estado, si se selecciona un participante y se arrastra a otro grupo, ese estudiante deja el grupo actual y pasa a formar parte del grupo donde fue arrastrado.

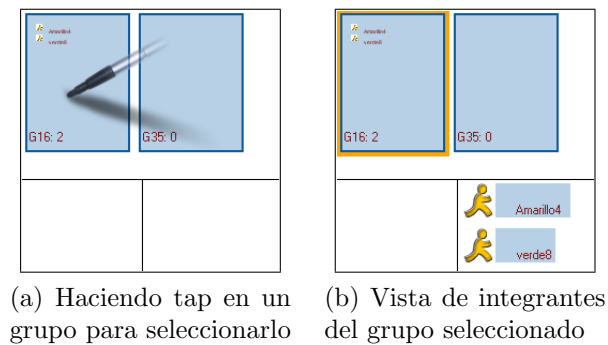
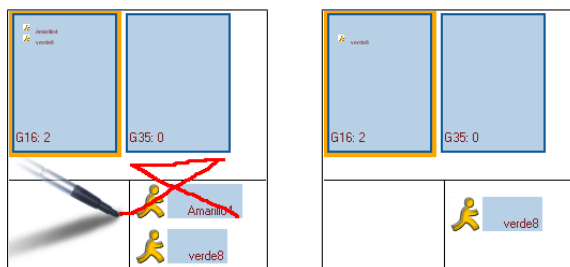


Figura 29: Ver integrantes de un grupo

### 4.7.3. Sacar a estudiantes de un grupo

Cuando hay un grupo seleccionado, realizando el gesto de la cruz sobre los participantes (ver Figura 30) ocasionará que esos participantes dejen el grupo al cual pertenecían.



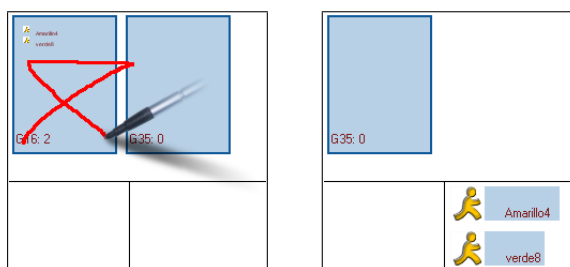
(a) Haciendo gesto de la cruz sobre un participante

(b) Resultado de aplicar gesto

Figura 30: Sacando un estudiante de un grupo

### 4.7.4. Eliminación de grupos

Haciendo el gesto de la cruz sobre un grupo causa su eliminación (ver Figura 31). Si el grupo contenía integrantes, todos ellos se sacan del grupo, con lo cual quedan libres para poder ser asignados a otros grupos posteriormente.




(a) Haciendo gesto de cruz sobre un grupo

(b) Grupo eliminado

Figura 31: Eliminación de un grupo

## 4.8. Definición de preguntas

Si el profesor quiere ver el nivel de aprendizaje de los alumnos en medio de la clase, tiene la opción de crear preguntas para que los alumnos las contesten, facilitando el *assessment*. El profesor puede hacer que las preguntas se respondan por los estudiantes de manera individual o grupal. Para entrar al modo, se debe hacer click en el icono  de la barra de modos.

### 4.8.1. Creación de una nueva pregunta

Para ello, el profesor se debe dirigir al “modo de definición de preguntas”, ir a la página donde va a crear la pregunta y seleccionar qué tipo de pregunta hacer a los alumnos. Hay 4 tipos de preguntas posibles en la aplicación:

- **Selección simple:** sólo una alternativa entre todas es la correcta.
- **Selección múltiple:** un conjunto de una o más alternativas conforman la respuesta correcta. Para que la respuesta sea correcta, la respuesta del alumno debe ser igual a ese conjunto de alternativas.
- **Secuencia:** la respuesta correcta es una secuencia ordenada de las alternativas expuestas en la pregunta. Si un alumno tiene en su respuesta las mismas alternativas que la respuesta correcta pero en otro orden, la respuesta dada por el alumno se considera incorrecta.
- **Libre:** en este tipo de pregunta no existen alternativas. Los estudiantes dibujan su respuesta libremente. Dado el carácter de la pregunta, la aplicación no puede determinar automáticamente si la respuesta es correcta o no, para ello es necesario que el profesor sea quien confirme que la respuesta es correcta o incorrecta.

El tipo de pregunta se puede cambiar en el menú de la parte superior de la pantalla (ver Figura 32). Un marco de color negro indica cuál es la opción seleccionada referente al tipo de pregunta.



Figura 32: Barra de menú de tipo de pregunta. De izquierda a derecha: selección simple, selección múltiple, secuencia y libre.

Las preguntas usualmente constan de 3 partes: enunciado (que indique al alumno cuál es la pregunta que debe responder), respuesta(s) correcta(s) y respuesta(s) incorrecta(s). En el caso de las preguntas libres la pregunta solamente tiene enunciado. El profesor debe seleccionar un nodo haciendo un tap en él, y aparecerán 3 iconos (ver Figura 33a): una cruz (para hacer que ese nodo sea respuesta incorrecta), un victo (para que sea correcta) o un signo de interrogación (para que sea parte del enunciado). No hay un límite máximo en la cantidad de respuestas correctas (en el caso de preguntas de secuencia o selección múltiple), respuestas incorrectas o enunciados, pero siempre se necesita de un enunciado, y en el caso de las preguntas no libres, al menos una respuesta correcta. Definir cuáles son las alternativas correctas/incorrectas permite al sistema indicar al estudiante de forma inmediata si la respuesta que dio al problema era la acertada o no, cuando este último ha respondido una pregunta no libre.

En las preguntas de secuencia, el orden en que se designen las alternativas correctas define el orden correcto de la respuesta. En el caso de las preguntas de selección simple, a lo más puede haber una respuesta correcta, por lo cual si se define un dibujable como respuesta correcta siendo que ya existía otro que poseía esa cualidad, el último dibujable queda como respuesta correcta, y el otro queda como incorrecta.

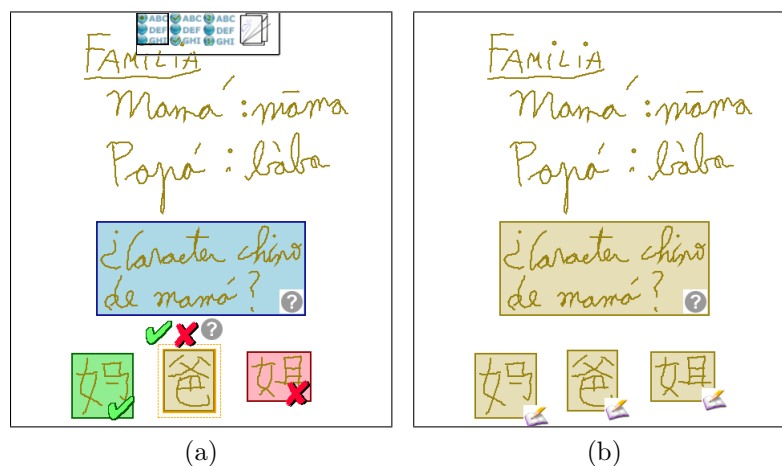



Figura 33: Vista de creación de preguntas (a) , vista del estudiante de la pregunta enviada por el profesor (b).


Si se quiere hacer una variante de la pregunta, hay que copiar el nodo o la diapositiva que contiene a la pregunta. Esa versión “duplicada” de la pregunta sí es modificable, e inicialmente su estado es “no enviada”. Después de hacer los cambios respectivos se puede enviar la pregunta a los estudiantes para que puedan responderla.

#### 4.8.2. Envío de preguntas a los estudiantes en forma individual

Una vez que la pregunta ya fue creada, se debe enviar a los alumnos para que puedan responderla. Para ello se debe expandir la “barra de opciones adicionales del modo” y el profesor debe presionar en la opción “Enviar” , con lo cual la pregunta es enviada a cada alumno para que sea respondida de manera individual. Una vez que pregunta fue enviada no se puede modificar el tipo de pregunta, o cuáles son partes del enunciado o respuesta (correcta/incorrecta). Para dar *awareness* que la pregunta fue enviada, el marco de la barra de selección del tipo de pregunta (que en la Figura 32 es negro) cambia a color rojo.

#### 4.8.3. Envío de preguntas a los estudiantes en forma grupal

Para favorecer el trabajo colaborativo entre los alumnos, el profesor puede formar grupos de trabajo y esos grupos asignarles preguntas que deben ser respondidas de forma colaborativa. Los grupos se forman usando el modo de definición de grupos, descrito en la Sección 4.7.

Cuando los grupos han sido formados, el profesor debe ir al “modo de definición de preguntas” y presionar sobre el icono de “Envío grupal”  para enviar la pregunta a los grupos. Al igual que en el envío de pregunta individual, no se puede modificar la pregunta una vez enviada.

#### 4.8.4. Confirmar respuestas libres como correctas/incorrectas

En las preguntas libres el sistema no puede determinar si la respuesta es certera o no, por lo cual el profesor debe revisar las respuestas de los estudiantes y marcarla como correctas o incorrectas, por medio de un gesto de victo o una cruz, respectivamente (ver Figura 34).

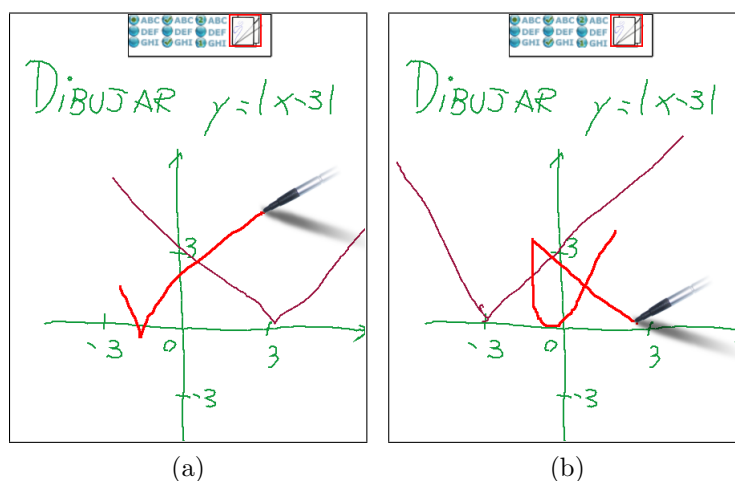


Figura 34: Haciendo el gesto de un victo (a), provoca que la respuesta a la pregunta libre sea marcada como correcta. Con el gesto de una cruz (b), la respuesta queda como incorrecta.

#### 4.8.5. Ver respuestas de los alumnos

Otra de las funcionalidades que provee este modo es la posibilidad de ver las respuestas que han dado los alumnos a una pregunta en particular. Cuando en la página actual hay una pregunta, el profesor dispone de una “barra de estudiantes” que aparece al costado derecho de la pantalla, que es utilizada para ver las respuestas que han dado los estudiantes. Esa barra contiene una lista con los estudiantes que han enviado la respuesta a la pregunta.

En el caso de las preguntas cerradas, permite ver la respuesta de un estudiante a la vez o de grupo de estudiantes a la vez según si la respuesta era de carácter individual o grupal, respectivamente. La respuesta se dibuja con color verde para diferenciarla de los otros nodos (ver Figura 35).

Si la pregunta es libre, se pueden superponer los diferentes respuestas en la misma página, como se explica en detalle en la Sección 4.8.6.



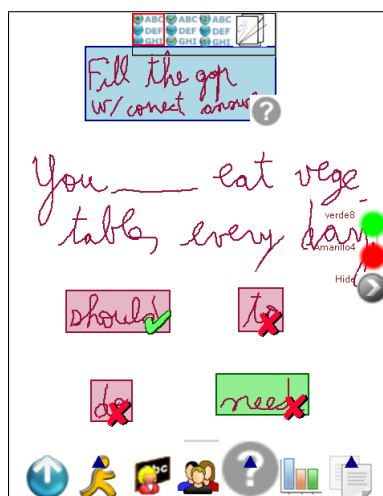
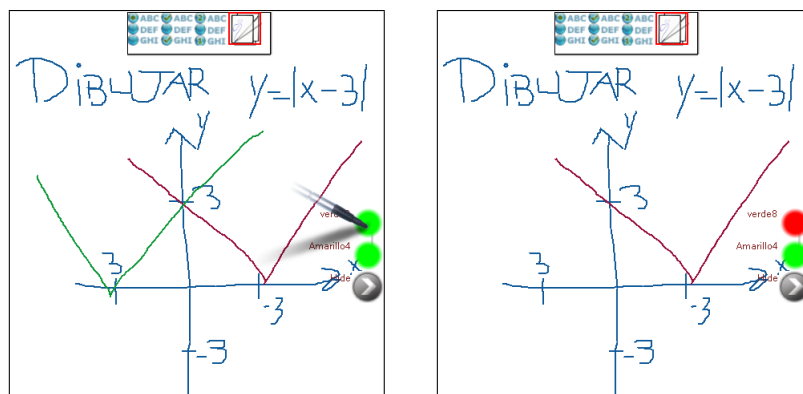


Figura 35: Vista de la respuesta de un estudiante a una pregunta

#### 4.8.6. Superposición trazos en preguntas libres

En las preguntas libres, por defecto se superponen todas las respuestas que han dado los alumnos sobre la pregunta (ver Figura 36). Usando la “barra de estudiantes” se puede filtrar los trazos de los estudiantes por dispositivo. Para preguntas donde la respuesta correcta tiene una forma o una ubicación bien definida, la superposición de respuestas permite al profesor detectar si hay respuestas correctas o incorrectas de forma inmediata.



(a) Haciendo tap en la barra de estudiantes

(b) Ocultados los trazos de un estudiante

Figura 36: Vista combinada de las respuestas de los estudiantes a una pregunta libre

### 4.9. Responder preguntas

El modo de “Responder preguntas” permite a los alumnos contestar las preguntas que el profesor les ha enviado. Además este modo incluye una vista que muestra cuáles son las preguntas que ha recibido el alumno, indicando cuáles ha respondido y junto con su desempeño

a la hora de responderlas.

En este modo (ver Figura 33b), cuando en la página actual hay una pregunta, el icono del modo cambia de imagen para indicar al estudiante qué tipo de pregunta está respondiendo: selección simple, selección múltiple, secuencia o libre.

Cuando la pregunta es grupal, en el icono de *Following* se muestra el nombre del grupo al cual pertenece el alumno (ver Figura 37).

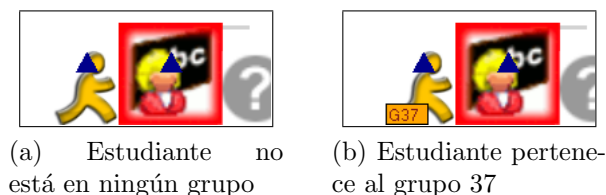




Figura 37: Indicador de pertenencia a un grupo


Para que el estudiante sepa cuál es el enunciado o las alternativas de la pregunta, los elementos que forman parte del enunciado tienen el icono de una pregunta en una de sus esquinas. Los elementos que forman parte de las alternativas (correcta o incorrecta) tienen el icono  indicando esa condición, para no confundir visualmente al estudiante con otros elementos que puede tener la página (como se ve más adelante en la Figura 38a).


Cuando se envía la respuesta al profesor, sobre el icono del modo aparecerá un círculo rojo tachado, indicando que no se puede volver a responder el problema.

#### 4.9.1. Responder de manera individual

La forma de responder una pregunta cambia dependiendo del tipo de pregunta que fue enviada por el profesor.

Para responder una pregunta libre, el alumno sólo tiene que dibujar hasta que estime conveniente, y entonces debe presionar el botón “Enviar” , accesible expandiendo la “barra de opciones adicionales del modo”. Para saber si la respuesta dada es correcta o no, el profesor tiene que ir a la página de la respuesta y marcarla como correcta o incorrecta. En el dispositivo del alumno en la esquina superior izquierda se dibujará un icono indicando el estado de la pregunta (si fue correcta o incorrecta).

Para los otros tipos de pregunta (selección simple, múltiple o secuencia), el alumno debe hacer tap o click sobre alguno de los nodos de alternativas de respuesta (indicadas mediante el icono  en su costado) para agregarlo/quitarlo a su respuesta (ver Figura 38). Los elementos que pertenecen a la respuesta del estudiante se dibujan con fondo verde para que sea más fácil identificarlos visualmente. Al igual que en caso de una pregunta libre, el alumno debe

presionar el botón “Enviar”  para mandar la respuesta al profesor. Al enviar la respuesta de este tipo de pregunta (Figura 38c), la aplicación indicará al usuario si la respuesta dada es correcta o incorrecta, mostrando el icono de un victo o de una cruz en la esquina superior izquierda de la pantalla. Adicionalmente, en la esquina inferior derecha de las alternativas se mostrará un icono indicando cuál es la respuesta correcta.




(a) Haciendo tap sobre una alternativa la marca como respuesta (b) Presionando en el botón Enviar para mandar la respuesta al profesor (c) La respuesta es incorrecta. Se muestra cuál era la alternativa correcta

Figura 38: Respondiendo a una pregunta individual de selección simple

Una vez que el alumno ha enviado su respuesta a un determinado problema no la puede cambiar. Es decir, no puede cambiar la alternativa que ha marcado en una pregunta cerrada, o agregar, quitar o mover objetos en una pregunta libre.

#### 4.9.2. Responder de manera grupal

Cuando la pregunta es de carácter grupal, cuando los alumnos están en la página de esa pregunta se les indica a qué grupo pertenecen, en el icono de *Following* de la barra de menú.

El proceso de responder es similar que el caso de la pregunta individual, salvo que cuando se presiona el botón “Enviar”  se envía la respuesta al resto del grupo en vez de ser enviada directamente al profesor (ver Figura 39). Para que la respuesta que dan los estudiantes de un grupo sea efectivamente enviada al profesor, todos los estudiantes deben marcar la misma respuesta (en el caso de una pregunta cerrada) o estar de acuerdo sobre la misma versión de la respuesta (en el caso de una pregunta libre). Cuando los alumnos presionan en el botón “Enviar”, mandan la respuesta al resto del grupo. Si hay consenso en la respuesta, ésta es enviada al profesor.

Cuando no hay consenso en la respuesta, aparecerán unos iconos en la parte superior izquierda de la pantalla (ver Figura 39b) indicando la cantidad de personas del grupo que están

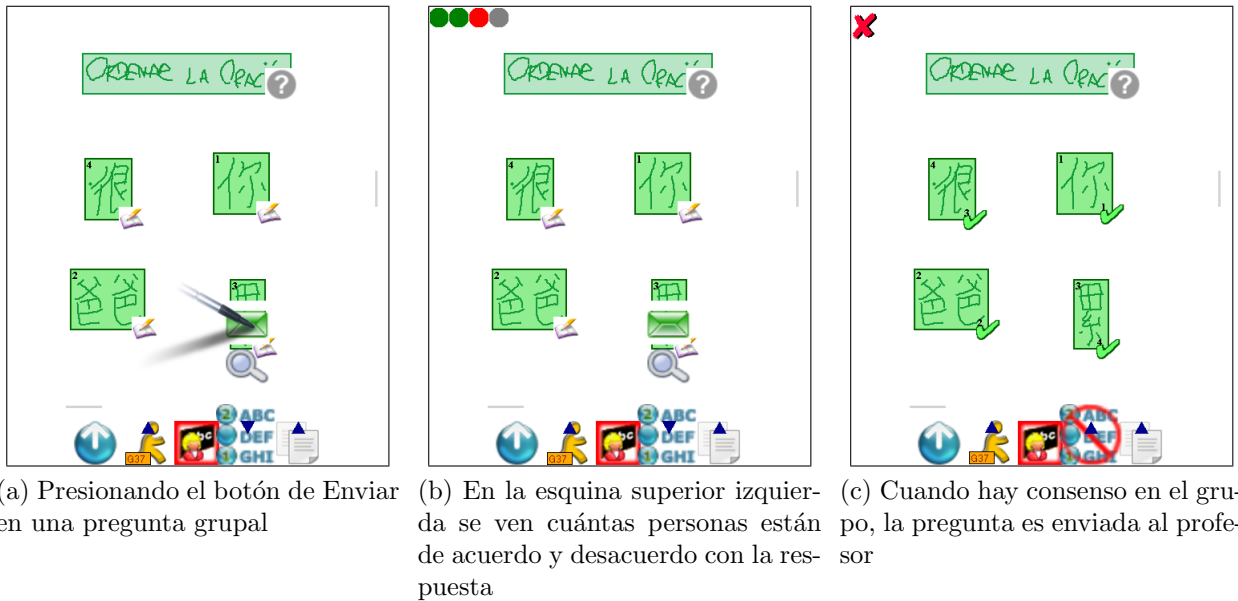


Figura 39: Contestar una pregunta grupal

de acuerdo, en desacuerdo con la respuesta que dio el alumno y los que no han respondido aún. Los alumnos del grupo pueden cambiar su respuesta y presionar en el botón “Enviar” nuevamente, hasta que todos estén de acuerdo en la respuesta.

Al igual que en caso individual, cuando la respuesta ha sido enviada al profesor, en la esquina superior izquierda de la pantalla aparecerá un icono indicando si la respuesta dada es correcta o no. En el caso de las preguntas libres, inicialmente aparecerá un icono de pregunta indicando que es necesaria la confirmación del profesor sobre la correctitud de la respuesta dada por los alumnos.

#### 4.9.3. Ver preguntas recibidas

Para que el alumno pueda ver las preguntas que se han recibido hasta el momento, existe una vista que permite ver todas las preguntas que el profesor ha enviado al alumno, indicando cuáles preguntas han sido respondidas correcta e incorrectamente. Esto permite al estudiante saber cómo ha sido su desempeño al responder preguntas, y le permite ir directamente a las páginas de las preguntas (haciendo doble tap sobre las preguntas). Para poder ingresar a ese modo, se debe hacer un tap en el icono del “modo de responder preguntas”, y hacer tap en el icono . Para salir de esa vista, se debe hacer doble tap en un área vacía.

Para mejorar la navegación, esta vista implementa el gesto de scroll/zoom (ver Figura 40) análogo al descrito en la Sección 4.5.6. Para poder ingresar a la página de una pregunta en particular, se debe hacer doble tap en la vista previa de la página, ocasionando que la aplicación pase al modo de “Responder preguntas” y vaya a la página de la pregunta.

En la parte superior de la pantalla hay una franja que muestra, de manera proporcional,

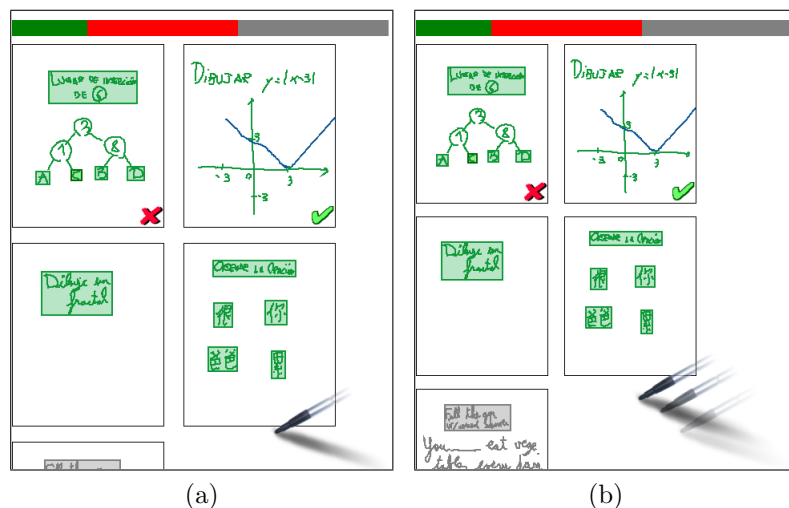


Figura 40: Arrastrando el stylus se aplica el gesto de scroll/zoom en el modo de “Ver preguntas”

cuántas preguntas el alumno ha contestado correctamente, incorrectamente, o no han sido contestadas, en colores verde, rojo y gris, respectivamente . Por ejemplo, si las respuestas incorrectas corresponden al 25 % del total de preguntas, entonces la barra que representa las respuestas incorrectas ocupa un 25 % del ancho de la pantalla . En la Figura 40 se puede ver tal franja, que muestra en ese caso particular que hay más respuestas incorrectas que respuestas correctas.

## 4.10. Resultados de las preguntas

El modo para ver los resultados de las preguntas presenta una vista de todas las respuestas que han dado los alumnos a las preguntas, junto con gráficos para informar al profesor sobre el desempeño de los alumnos y la dificultad de las preguntas, para dar apoyo al proceso de *assessment*. Este modo sólo puede ser utilizado por el profesor.

### 4.10.1. Vista de las respuestas de los alumnos

Este modo va mostrando las respuestas de los estudiantes a medida que están siendo recibidas por el dispositivo del profesor. Las preguntas están dispuestas en una tabla, como se explica en la Figura 41. Las columnas representan los estudiantes, y las filas representan las preguntas. Así, para poder saber la respuesta de un estudiante a un determinada pregunta, hay que buscar la celda donde la fila de la pregunta intersecta a la columna del estudiante.

En las celdas de la tabla se muestran las respuestas dadas por los estudiantes a las preguntas. En el caso de las preguntas cerradas, las alternativas marcadas por los estudiantes aparecen con fondo verde. En las preguntas abiertas, se verá lo que dibujó el alumno. En un costado de la pregunta se dibuja un icono que indica si la respuesta dada por el estudiante es correcta, incorrecta, o que no ha sido contestada.

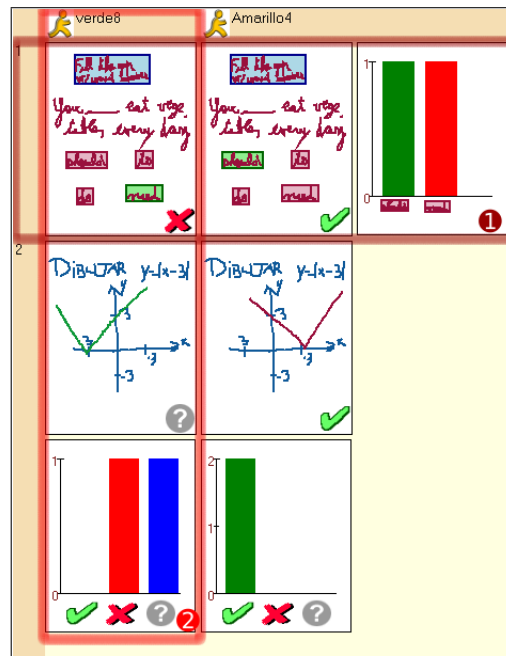


Figura 41: En una fila (marcada como 1 en la imagen) se ven las respuestas de los estudiantes a las preguntas, mientras que en una columna (2) se ven las respuestas que ha dado un estudiante a las diferentes preguntas.

Al hacer doble tap en la respuesta de un alumno a una pregunta ocasiona que la aplicación vaya a la página de la pregunta, mostrando la respuesta del estudiante (ver Figura 42).

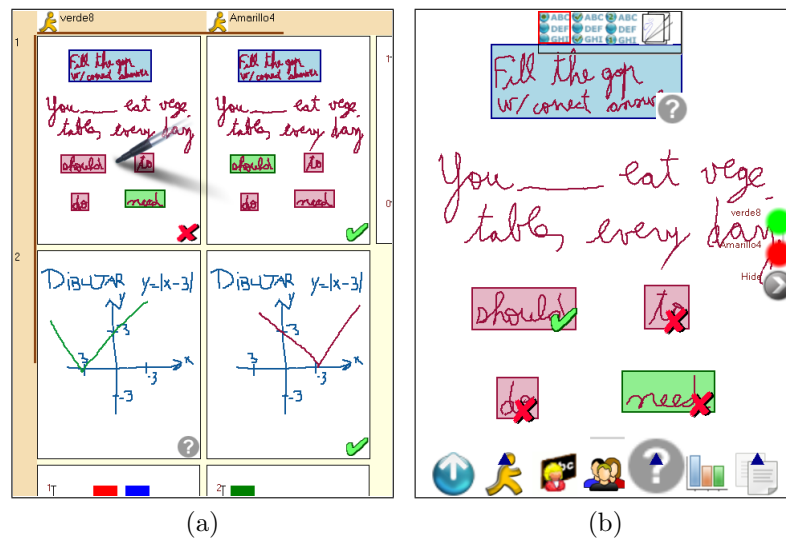


Figura 42: Al hacer doble tap sobre una respuesta (a) se va a la página de la pregunta, con los nodos de la respuesta de ese estudiante pintados con otro color (b).

### 4.10.2. Gráficos por estudiante y por pregunta

Al final (extremo derecho) de cada columna se muestra un gráfico de barras que da información sobre cómo ha sido el desempeño del alumno al responder las preguntas. En ese gráfico se pueden apreciar cuántas respuestas correctas e incorrectas ha tenido el estudiante. Análogamente, al final (extremo inferior) de cada fila existe un gráfico de barras que muestra por cuáles respuestas se han inclinado los estudiantes en una determinada pregunta (en el caso de las preguntas cerradas). En el eje X del gráfico se dibujan las alternativas que representan a cada barra del gráfico.

En cualquiera de los gráficos de barras se puede hacer doble tap para poder verlo con mayor detalle en pantalla completa. Para dejar esa vista del gráfico, basta con sólo hacer un tap en la pantalla.

### 4.10.3. Scroll/zoom

Dado que en este modo la información a mostrar puede no caber en la pantalla, se implementó un gesto que permite realizar desplazamiento y zoom a la vez. Este gesto funciona ligeramente diferente a los otros gestos de desplazamiento/zoom mencionados anteriormente, ya que en este modo además se necesita un desplazamiento en sentido horizontal: mover el stylus en diagonal provoca cambios en el zoom, mientras que moverlo sólo horizontal o verticalmente cambia el desplazamiento horizontal y vertical, respectivamente (ver Figura 43 y 44). Cerca de los costados de la pantalla aparecerán unas barras de scroll que avisa al usuario cuánta es el área no visible y en qué posición relativa a la tabla se encuentra.

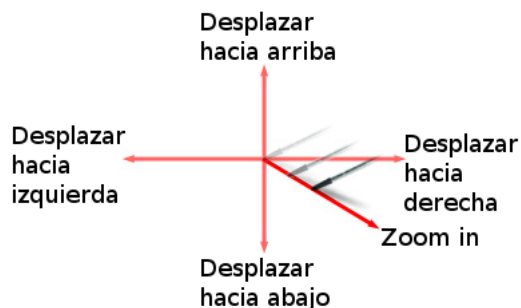


Figura 43: Arrastrando el stylus en la vista de preguntas permite cambiar el zoom y desplazar sin necesidad de gestos adicionales. (Versión modificada de imagen de [15])

## 5. Implementación

En esta sección se describe la implementación de las características nombradas en los capítulos anteriores de este documento.

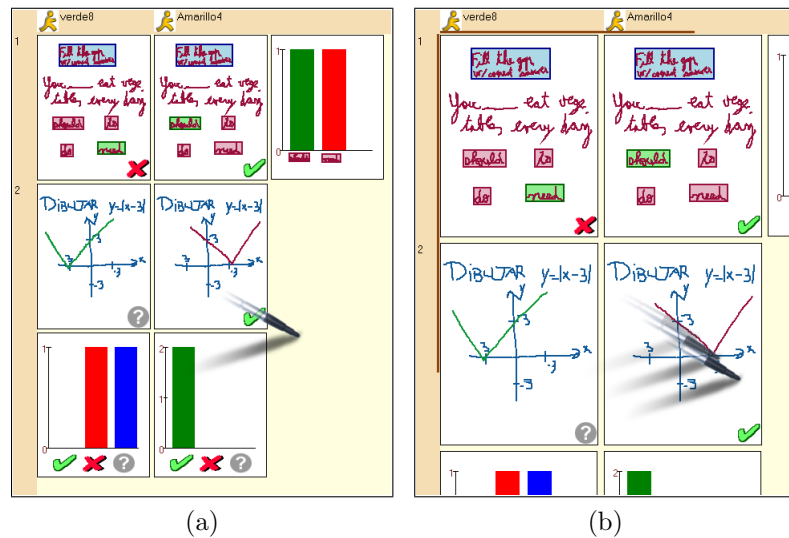


Figura 44: Arrastrando el stylus en forma diagonal cambia el nivel de zoom del modo.

## 5.1. Ambiente de programación

Con el fin de desarrollar el sistema se utilizó un computador Pentium 4 de 1.6 Ghz, con 1GB de RAM. Los dispositivos móviles utilizados fueron unas PDAs Dell modelos x50v y x51v, cuyos sistemas operativos son Windows Mobile versiones 2003 Second Edition y 5.0 respectivamente. Las PDAs poseen pantalla táctil, conectividad Wi-fi, Bluetooth y comunicación a través de un puerto infrarrojo (IrDa).

El entorno de desarrollo fue Visual Studio 2005, utilizando .NET Framework y .NET Compact Framework, ambos en su versión 2.0 . El language de programación utilizado fue C# , debido a que la plataforma SmartCore está implementada en ese lenguaje.

## 5.2. Arquitectura de la plataforma SmartCore

La plataforma SmartCore está programa íntegramente en C#, versión 2.0 ( de .NET Framework y .NET Compact Framework, para PC y dispositivos móviles respectivamente). La plataforma consta de varios módulos (ver Figura 2), explicados a continuación [15]:

- **SmartCore:** Núcleo de la aplicación, es el encargado de iniciar todos los demás módulos de la plataforma. Maneja el comportamiento global de la aplicación.
- **Módulo de comunicaciones:** También llamado módulo de red, realiza el descubrimiento de los dispositivos presentes en la misma red y la posterior conexión entre ellos. El módulo adicionalmente se encarga del intercambio de datos por TCP/IP e IrDA (infrarrojo).
- **Módulo de actividad:** Define el comportamiento de la aplicación, dependiendo de su estado actual. Este módulo aplica el reconocimiento de gestos, se encarga de la interfaz gráfica, y recibe los eventos del mouse o stylus para ser posteriormente procesados por la aplicación.



A continuación se describen los aspectos más relevantes de la plataforma.

### 5.2.1. SmartDraw

**SmartDraw** es la clase principal de la plataforma SmartCore. Esa extensa clase contiene métodos y variables estáticas que modifican el comportamiento global del sistema. Esta clase inicializa los demás componentes del sistema y la interfaz gráfica. Contiene una lista con todos los modos (ver Sección 5.2.8) de la aplicación actual.

Esta clase provee métodos para el envío de objetos a otros dispositivos (el método **Share** y sus sobrecargas), los cuales usan el módulo de red de la plataforma para realizar los envíos respectivos. Además, esa clase inicializa el thread de refresco de pantalla (ver Sección 5.2.9).

### 5.2.2. Postprocesables

Para poder enviar un objeto por red de un participante a otro, el objeto debe heredar necesariamente de la clase **PostProcessable**. Cuando un dispositivo recibe un objeto que extiende de la clase **PostProcessable** se le invoca el método **PostProcess()** con el fin de realizar acciones adicionales en el dispositivo receptor. Es posible redefinir ese método para poder realizar tareas según lo requiera el programador.

Cada instancia de esta clase almacena quién fue la persona que lo creó y un **string** identificador que se construye a partir del nombre de la clase del objeto y su fecha de creación. El identificador es utilizado para evitar la creación de más de una instancia representando al mismo objeto cuando se recibe en más de una ocasión.

Los objetos de esta clase pueden representar objetos persistentes (como los dibujables, explicados en la sección 5.2.4) u objetos creados con la finalidad de ejecutar acciones en equipos remotos (los cuales se denominarán **mensajes**).

### 5.2.3. Módulo de red

El descubrimiento y la comunicación entre los diferentes dispositivos es labor del módulo de red de la plataforma. El módulo utiliza protocolos TCP y UDP sobre IP e IrDa.

Cada dispositivo anuncia su existencia mandando mensajes UDP por multicast cada cierto tiempo. De esta manera, los dispositivos pueden descubrirse sin la necesidad de tener un servidor central donde los participantes se tengan que registrar.

Cuando un dispositivo descubre a otro por UDP, realiza una conexión TCP/IP con él, y entre ambos se envían la lista de participantes que están conectados con ellos. Así se agiliza el proceso de descubrimiento, puesto que se evita que cada dispositivo tenga que ir descubriendo a los demás equipos uno por uno.

Cada vez que se realiza una conexión nueva con un dispositivo se crean threads encargados del envío y recepción de objetos con ese dispositivo. El thread de recepción es el encargado de deserializar los objetos que se van recibiendo, y de entregarlos a la capa de aplicación para

su procesamiento posterior.

El módulo de red mantiene una lista con los dispositivos que están conectados, y a qué grupo(s) pertenecen. Un grupo contiene una lista de dispositivos, y un dispositivo puede estar en más de un grupo a la vez. Por defecto los dispositivos no pertenecen a ningún grupo. El módulo de red no hace otro uso de los grupos, por eso las aplicaciones que se construyan sobre SmartCore pueden utilizarlos según sus requerimientos.

#### 5.2.4. Dibujables

Todo los objetos que se despliegan en la pantalla (a excepción de los menús) son “dibujables”, este tipo de objetos heredan de la clase **Drawable**. La plataforma SmartCore mantiene una lista de todos los dibujables que se han creado hasta el momento y que no se han borrado.

Cada dibujable almacena un **string** como identificador, un rectángulo que indica la ubicación y dimensiones del dibujable, y una página asociada que es donde le corresponde dibujarse. Esa página es en realidad una instancia de la clase **DrawablesContainer**, explicada con mayor detalle en la sección 5.2.5.

La clase **Drawable** tiene varios métodos que pueden redefinidos que permiten indicar si el dibujable puede ser seleccionado, borrado, enviado por red, dibujado, etc. Esto permite modificar el comportamiento del dibujable en la aplicación, permitiendo flexibilidad sobre las operaciones que es posible realizar sobre él.

La clase **Drawable** permite que sus clases derivadas puedan redefinir el método **Draw**, que es llamado cuando se quiere pintar ese dibujable. Hay dibujables que contienen páginas en su interior, como es el caso de un **PageOwner**, que representa un nodo al cual uno puede entrar, lo que lleva al usuario a la página al interior del nodo. **PageOwner** redefine el método de dibujado, así que cuando es pintado en pantalla dibuja en su interior el contenido de su página interior.

Al guardar la sesión actual en un archivo XML, lo único que se almacena al interior del archivo son los dibujables. Si hay alguna información adicional que se desee guardar en el XML que no esté contenida en ningún **Drawable**, se debe crear un objeto **Drawable** que la contenga, asegurando así su almacenamiento en disco.

Una clase especial de dibujables son los trazos o *strokes*. Los trazos se crean basándose en el recorrido que haya hecho el stylus o el mouse en la pantalla. Debido a que los puntos de aquellos trazos pueden ser muy grandes debido a la resolución de algunos equipos móviles [15], a veces se realiza una compresión del trazo, reduciendo el número de puntos de ella. Esto permite que los trazos ocupen menos espacio en memoria, logrando transferirlos por red más rápidamente, y facilita de detección de gestos.

### 5.2.5. Contenedor de dibujables

Como fue dicho anteriormente, cada dibujable está asociado a una página. Las páginas de la plataforma son en realidad contenedores de dibujables, es decir, una instancia de la clase `DrawablesContainer` que almacena una lista de dibujables que están asociadas con ella. Los dibujables de esa lista son pintados en la pantalla cuando el contenedor es la página actual de la aplicación. Cada página además almacena cuáles son los dibujables que están seleccionados, y cuál es su dibujable “padre”. Estos contenedores nunca se almacenan a disco, lo que sí se guarda son los dibujables que contienen.

Al iniciar la aplicación, la página actual es la llamada “página principal” o *main*. Todo el contenido que se genere se organiza como un árbol jerárquico de nodos, donde la página principal actúa como raíz del árbol.

### 5.2.6. Widgets

Todos los objetos que no son dibujables pero se despliegan en pantalla son *widgets*. La barras de menú que aparecen a los costados de la pantalla son widgets. Los *widgets* son elementos que heredan de la clase `vObject`. Los objetos que hereden de esa clase pueden redefinir los métodos que se ejecutan cuando el usuario interactúa con ellos a través del stylus.

A diferencia de los dibujables, la información contenida en estos objetos no se guarda en disco. Además, cuando el usuario interactúa con la pantalla, los elementos que tienen mayor prioridad para procesar ese evento son los *vWidgets*. Si no es procesado, entonces se consulta al modo actual (y por ende, a los gestos que use el modo) para manejar aquel evento.

Para que un *widget* se dibuje en pantalla, debe estar en un contenedor visible. Existe un contenedor principal que siempre está visible, con lo cual sólo agregando componentes a él, se dibujan en pantalla. Los contenidos del contenedor principal por omisión se dibujan independientemente de la página y del modo en que se encuentre la aplicación.

### 5.2.7. Gestos

Los gestos son clases que heredan de `Gesture`, encargados de analizar los eventos que genera el stylus en la pantalla táctil o el mouse, en los métodos `AnalyzeMouseDown`, `AnalyzeMouseMove` y `AnalyzeMouseUp`. Estos métodos devuelven `true` si se ha detectado el gesto asociado, con lo cual la plataforma deja de buscar gestos que encajen con el gesto del lápiz. Cuando un gesto es detectado se ejecuta la acción asociada a él. Por ejemplo, en el gesto de borrar, si se detecta que se hizo una cruz con el stylus se procede a borrar los dibujables que estaban bajo ella.

Los gestos pueden actuar en estas situaciones:

- Cuando se presiona el stylus contra la pantalla o el botón del mouse, llamando al método `AnalyzeMouseDown`.

- Cuando se mueve el stylus o mouse a través de la pantalla, ejecutando el método `AnalyzeMouseMove`.
- Cuando se suelta el stylus o el botón del mouse, invocando al método `AnalyzeMouseUp`.

La plataforma provee gestos básicos para seleccionar, mover, copiar, cambiar de tamaño, rotar, etc. Los gestos creados para las aplicaciones que se construyen sobre SmartCore generalmente están muy acoplados a los modos, lo cual hace difícil que puedan ser usados en modos diferentes para los cuales fueron diseñados.

Los modos usualmente van preguntando gesto por gesto si se ha reconocido el trazo del stylus. El orden en que se llaman a los gestos tiene relevancia, se debe tener en cuenta el orden para evitar que no se llegue a reconocer un gesto porque fue erróneamente reconocido como otro gesto.

### 5.2.8. Modos

Como se dijo en la Sección 2.4, las aplicaciones construidas a partir de SmartCore se basan sistema de modos para funcionar. La aplicación puede contener cualquier número de modos disponibles, pero sólo puede tener un único modo activo a la vez. A nivel de implementación, un modo es una instancia de una clase que hereda de `Mode`. Los principales métodos que provee la clase `Mode` para ser redefinidos en las clases derivadas son:

- `EnterMode`: Este método se ejecuta cuando el modo pasa a estar activo.
- `ExitMode`: Este método es llamado cuando el modo deja de estar activo.
- `AnalyzeMouseDown`, `AnalyzeMouseMove`, `AnalyzeMouseUp`: Estos métodos se llaman en el objeto que representa al modo activo cuando el stylus se posa en la pantalla, cuando se arrastra el stylus por la pantalla y cuando se levanta el stylus de la pantalla, respectivamente. Redefinir estos métodos permite cambiar el comportamiento de la aplicación ante eventos del stylus generado por el usuario. Estos métodos se encargan de ocupar los métodos `AnalyzeMouseDown`, `AnalyzeMouseMove` y `AnalyzeMouseUp` de los gestos.
- `Receive`: Este método recibe un `PostProcessable` como parámetro. Cuando la capa de red de la aplicación recibe un objeto de otro dispositivo, ejecuta su método `PostProcess()` y notifica a todos los modos de la aplicación la llegada de un nuevo objeto, ejecutando el método `modo.Receive()` en todos los modos conocidos, pasando ese nuevo objeto como parámetro del método.
- `DoesOverrideDraw`: Método que retorna verdadero o falso, que indica si el propio modo se encarga del dibujado de la pantalla en vez de delegar esa función a plataforma SmartCore. Este método es llamado por el thread encargado de dibujar la pantalla, si este método devuelve verdadero, entonces el thread de dibujado llama al método `FullDraw` del modo.

- **FullDraw**: Este método recibe un objeto **Graphics** como parámetro, y se encarga de dibujar la pantalla. Algunos modos pueden querer redefinir este método para dibujar esquemas o diagramas que sería muy difícil de representar con objetos derivados de **Drawable**.
- **ManageContextualOptions**: Se llama cuando hay uno o varios objetos **Drawable** seleccionados en la página actual y el thread de dibujado desea dibujar botones alrededor de ellos. Redefinir este método permite agregar botones extra alrededor de los dibujables para realizar funciones específicas.
- **PageChanged**: Se llama a este método cuando dentro del modo se ha cambiado de página.
- **AfterSnakePaste**: Se llama a **AfterSnakePaste** cuando se han pegado nuevos dibujables en la página actual usando el gesto de copiado y pegado (**SnakeSelectionGesture**). Redefinir este método permite ver cuáles son los objetos recién pegados y manipularlos.

Cuando se inicia la aplicación, la plataforma busca todas las clases que hereden de **Mode** por medio de reflexión<sup>1</sup>, y las agrega a la barra de modos del extremo inferior de la pantalla. Así, no es necesario que el programador registre el modo en algún lado para que se pueda ser mostrado en la aplicación.

### 5.2.9. Dibujado de la pantalla

La plataforma **SmartCore** (en particular, la clase **SmartDraw**) por defecto se encarga de dibujar todo lo que hay en la pantalla: el fondo de la pantalla, dibujables (trazos, nodos, diapositivas, entre otros), barras de menú, etc. Sin embargo, se pueden configurar los modos para que dibujen en la pantalla, aunque de manera general se deja que **SmartDraw** se encargue del dibujado.

Cuando el thread de dibujado quiere pintar los elementos en la pantalla, verifica si el modo actual se encarga del dibujado. Si es así, se llama al método **FullDraw** del modo, en caso contrario se obtiene el **DrawablesContainer** actual, recorre su lista de dibujables y evalúa si los tiene que dibujar o no.

Otra forma para dibujar en la pantalla es usar **SmartDraw.MyGraphics**. Este objeto **Graphics** es público y puede ser accedido desde cualquier lugar de la aplicación. Usar ese objeto es más ineficiente porque la pantalla se va refrescando a medida que se va dibujando con el objeto.

## 5.3. Detalles de la implementación

A continuación se explica la implementación de los modos descritos anteriormente.

---

<sup>1</sup>Reflexión es la habilidad de un programa de examinar y modificar su estructura de alto nivel en tiempo de ejecución. *C#* posee cualidades introspectivas de reflexión. [15]

### 5.3.1. Comentarios generales

En el sistema se hace una diferenciación entre el rol del alumno y el rol del profesor. La aplicación almacena quién es el profesor, el resto de los dispositivos toma automáticamente el rol de alumno.

Como se ha dicho anteriormente, SmartCore se basa en un sistema de modos para funcionar. MCPresenter contiene sus propios modos para organizar las funcionalidades. Los modos de MCPresenter se pueden ver en la Figura 45. Cabe notar que `SlideshowMode` y `StudentNotesMode` no son modos, pero son utilizados por `TeacherMode`, evitando que `TeacherMode` concentrara todas las funcionalidades, haciendo menos mantenible su implementación.

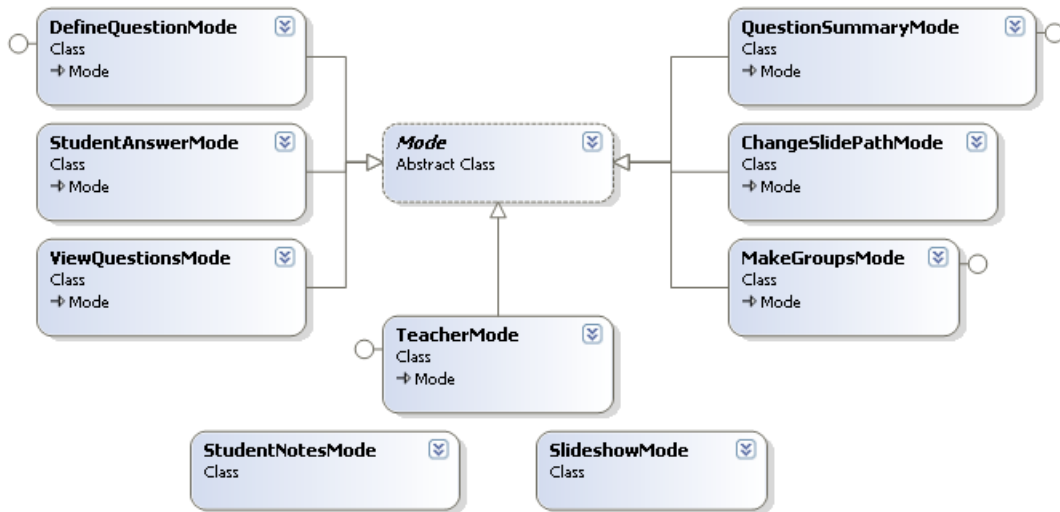


Figura 45: Diagrama de clases de los modos de MCPresenter

A continuación se describen los aspectos más relevantes de la implementación, por cada modo de la aplicación.

### 5.3.2. Área del pizarrón

La clase `TeacherMode` es la que implementa la funcionalidad del área del pizarrón. Esta clase es la que guarda quién tiene el rol de profesor en la sesión actual. Cuando un dispositivo toma el rol del profesor, se envía un mensaje al resto, todos los que reciben ese mensaje pasan automáticamente a ser estudiantes. Para lograr que los dispositivos que entren posteriormente a la sesión puedan saber quién es el profesor, se utiliza un dibujable del tipo `TeacherInfoDrawable`, el cual almacena quién es el profesor. Cada vez que un nuevo participante se conecta, se le envían todos los dibujables de la sesión actual (incluido el `TeacherInfoDrawable`) y se post-procesan. Al post-procesar el dibujable recién mencionado se fija quién es el profesor en el dispositivo receptor.

Para el manejo de las peticiones de escritura, el modo en el dispositivo del profesor, almacena una tabla de hash que relaciona el identificador de un alumno con un objeto que

contiene dos variables de instancia: el tiempo cuando esa persona realizó la última petición, y el estado (una enumeración del tipo `StudentStatus`), que puede tomar uno de los tres valores:

- **Initial**: El alumno no tiene permiso para escribir, y no lo ha solicitado tampoco.
- **AskedForWrite**: El alumno no tiene permiso para escribir, pero ha hecho una solicitud al profesor.
- **HasPermissionToWrite**: El alumno fue autorizado para escribir en el área del profesor.

El diagrama de estados del alumno se puede apreciar en la Figura 46. El alumno por sí solo únicamente puede cambiarse al estado **AskedForWrite** desde el estado inicial por medio de una petición de escritura al profesor. El profesor cuando le da permiso a escribir al alumno, lo lleva al estado **HasPermissionToWrite**. Cuando al alumno se le quita el permiso de escritura, vuelve al estado inicial.

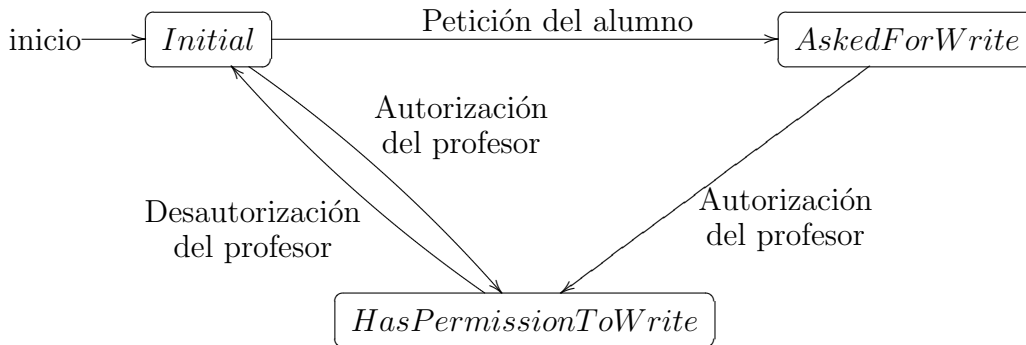


Figura 46: Diagrama de estados del alumno en TeacherMode

### 5.3.2.1. Dar permiso a los alumnos para escribir

La barra de estudiantes (que se muestra en el lado izquierdo de la pantalla) se construye en función de la tabla de hash mencionada anteriormente. Esa es la barra que permite al profesor dar o quitar permiso de escritura en el área del pizarrón.

Tanto al dar como al quitar permiso a un estudiante, se manda un mensaje por red dirigido a ese estudiante para que se entere de su nueva situación. El método `PostProcess()` de ese mensaje hace que se muestre el *awareness* visual al estudiante y ocasiona que se cambie el icono del “modo del pizarrón” para reflejar su nuevo estado.

Cuando los estudiantes tienen permiso para escribir sólo pueden borrar sus propios trazos. Para lograr esa funcionalidad, se creó un nuevo gesto de borrado, `SelectiveCutGesture`, que recibe como parámetro adicional en el método `AnalyzeMouseUp` un delegado<sup>2</sup> de tipo `Predicate<Drawable>` (ese delegado hace referencia a una función que recibe a un `Drawable`

<sup>2</sup>Un *delegado* es un tipo de referencia utilizado para encapsular un método. Son similares a los punteros a funciones de C o C++. [47]

como parámetro y devuelve falso o verdadero). Ese gesto decide si borrar un dibujable o no llamando al delegado: si ésta devuelve verdadero, se borra ese dibujable.

### 5.3.2.2. *Filtrar trazos por estudiante*

La forma de implementar esa funcionalidad es la siguiente: los trazos de los alumnos tienen redefinido el método `IsVisible()`, cuyo valor de retorno depende de un objeto del tipo `HideManager`. El funcionamiento de dicho objeto se explica en la Sección 5.3.10.3.

### 5.3.2.3. *Traer alumnos a la página actual*

Este modo provee una opción que permite al profesor forzar a todos los demás dispositivos que se cambien a la página donde está él (por ejemplo, en el caso que el profesor quiere mostrar algo importante). Ello es posible gracias a que se manda un red un mensaje de la clase `EnterTeacherModeMessage`, que tiene redefinido el método `PostProcess()`, de manera que cuando se ejecuta ese método en el dispositivo receptor del mensaje, se cambia la vista actual a la página del profesor, y el modo actual al “modo del pizarrón”.

### 5.3.2.4. *Pedir permiso de escritura al profesor*

En el caso de un estudiante, se consulta una variable en `TeacherMode` que indica si él puede escribir o no. Cuando un estudiante intenta escribir en el área del profesor, en el método `AnalyzeMouseUp` del modo se verifica si el alumno tiene permiso para hacerlo. Si no lo tiene, se le envía por red un mensaje (un post-procesable) al profesor representando la petición de escritura. En caso contrario, el alumno puede realizar sus acciones de manera normal.

Por el lado del profesor, cuando se recibe el post-procesable de la petición de escritura, se dibuja el *awareness* en pantalla usando `SmartDraw.MyGraphics`, y se cambia el estado del alumno a `AskedForWrite`.

### 5.3.3. **Toma de notas**

El modo de toma de notas, `NotesMode`, se encarga de proveer las funcionalidades de notas. Las notas que se hacen sobre una página no se realizan sobre la misma página, para evitar interferir con el contenido escrito originalmente escrito en la página. La página de notas corresponde a la página interior de un dibujable `StudentNoteOwner` que hereda de `PageOwner` (dibujable que contiene una página en su interior).

Cuando se entra al modo la página interior del `StudentNoteOwner` que se encuentre en la página actual pasa a ser la página actual (si no existe ese dibujable, se crea en el momento) y se dibuja como fondo la página en la cual se encontraba la aplicación antes de entrar al modo de notas (ver Figura 47). Al salir del modo se realiza el proceso inverso.

De esta manera, se evita que el usuario pueda modificar contenido que no sean notas (pues ese contenido está dibujado en el fondo, por lo que no es modificable), y además se logra que las notas sólo sean visibles dentro del modo de notas.



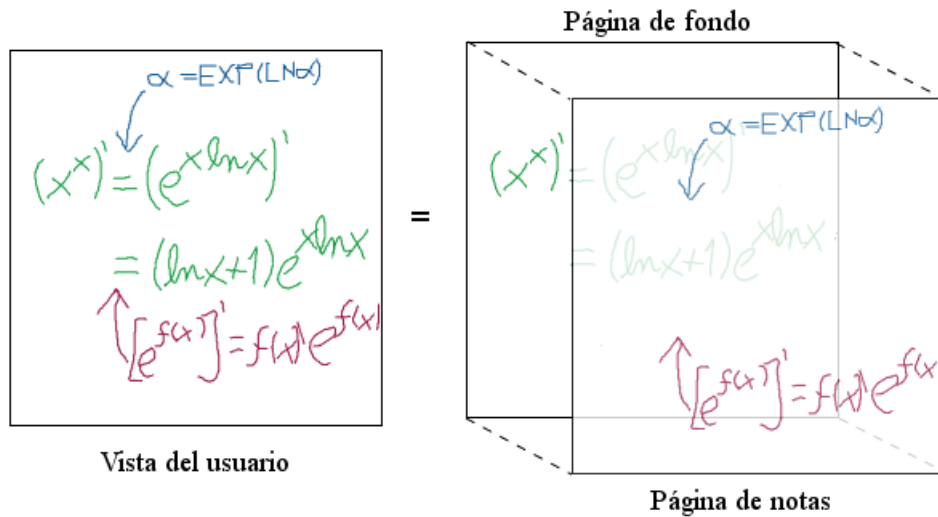


Figura 47: División por capas de la vista del “modo de notas”.

### 5.3.3.1. Escribir notas

Para escribir notas, en el `AnalyzeMouseUp()` del modo se utilizan los gestos para crear trazos y nodos. Para evitar sobrecargar la red cuando un usuario decide compartir notas con otro, todas las notas que hace una persona se comparten con el resto a medida que se su creador las va dibujando (incluso si esa persona no está compartiendo esas anotaciones con nadie), pero sólo pueden ser visibles en otros dispositivos si la persona que creó esas notas decide compartirlas.

El borrado es selectivo, sólo se pueden borrar los dibujables creados por uno mismo. Eso se logra mediante el gesto de borrado parametrizable descrito en la Sección 5.3.2.1.

### 5.3.3.2. Compartir notas

Cuando se hace tap en el botón de compartir notas en la barra de participantes, se llama a una función que crea un un mensaje (de la clase `ShareNotesMessage`), que es enviado al dispositivo al cual que le han compartido notas. El mismo mensaje se manda cuando se deja de compartir notas con algún participante.

El `PostProcess()` del `ShareNotesMessage` se encarga de dar *awareness* al usuario que alguien ha decidido compartir o dejar de compartir notas, además de actualizar los iconos de la barra de participantes. Se lanza un thread encargado de cambiar el color de la barra de participantes del modo de notas, cuando está contraída. Cuando se expande esa barra, se mata a ese thread.

### 5.3.3.3. Ver notas de otros

Para poder filtrar de manera visual cuáles notas son las que se están viendo, el modo contiene una lista de los participantes cuyas notas están ocultas (en un objeto de clase

`HideManager`, descrito en la Sección 5.3.10.3), y los trazos de las notas tienen redefinido el método que las hace visibles (método `IsVisible`), de manera que sólo se hacen visibles si no están en la lista mencionada anteriormente.

#### 5.3.4. Manejo de presentaciones

El funcionamiento de este modo se basa en dos dibujables: uno que representa una presentación (`PresentationDrawable`) y otro que representa una diapositiva (`SlideDrawable`). Una presentación contiene una secuencia ordenada de diapositivas. Para mantener el orden de las diapositivas que están al interior de la presentación, los objetos `PresentationDrawable` contiene una lista ordenada con los identificadores de las diapositivas contenidas al interior de ella. Esa lista se va actualizando conforme se van agregando, moviendo o eliminando diapositivas. Si bien en vez de usar esa lista se podría haber consultado la lista de dibujables del `DrawablesContainer` que representa a su página, esa lista no siempre guarda las diapositivas en el orden deseado.

##### 5.3.4.1. Creación de presentaciones

Para poder crear una diapositiva, en el método `AnalyzeMouseUp` del modo se detecta si es que se realizó el gesto de una “L”. Si es así, se crea una presentación en la página actual, y se crea una diapositiva vacía cuya página es la página interior de la presentación. Posteriormente se comparte la presentación recién creada a todos los estudiantes para que puedan verla y navegar a través de ella.

##### 5.3.4.2. Creación de diapositivas

Para evitar problemas con los otros modos, se restringe la creación de diapositivas sólo a la página interior de una presentación. Para detectar dónde se debe crear la diapositiva, cuando se hace click en la pantalla, se busca si el punto actual, y los puntos a 10 píxeles a los costados de ese punto tocan a una diapositiva. Si es así, la posición de la nueva diapositiva que se agrega es en función de esa diapositiva.

##### 5.3.4.3. Mover diapositivas

La línea divisoria que se dibuja en la pantalla se pinta al estar arrastrando diapositivas cerca del costado de otra diapositiva.

Se usa el método `VerColisiones` del modo para poder detectar cuál es la diapositiva más cercana al punto actual. Con ese dato, se pone un widget que representa a la línea divisoria en el lugar correspondiente. Al soltar el stylus, se llama a ese mismo método para ver en qué posiciones se tiene que actualizar la lista de identificadores de la presentación. Primero se eliminan de esa lista los identificadores de las diapositivas movidas, y después de insertan en el lugar correcto, evitando así que un mismo identificador se repita más de una vez.

#### 5.3.4.4. *Duplicación de diapositivas*

El duplicar diapositivas se basa en el gesto de SmartCore, `SnakePasteGesture`, que permite copiar y pegar. Sin embargo, después de pegar diapositivas éstas se deben ubicar en el lugar que les corresponde dentro de la presentación. Para ello, se redefinió el método `AfterSnakePaste` del modo, que es llamado después de pegar elementos. En ese método se hace uso del método `VerColisiones` (ver Sección 5.3.4.3) para ver cuál es la diapositiva más cercana al punto donde se realizó el gesto de pegado. Con esa información, se agregan los identificadores de diapositivas pegadas a la presentación en el lugar que corresponde dentro del arreglo ordenado de diapositivas.

#### 5.3.4.5. *Eliminación de diapositivas*

El gesto de borrado utilizado es el mismo gesto de borrado de SmartCore. Sin embargo, se notó que se detectaba el gesto del scroll/zoom antes de detectarse el gesto de borrado. Para evitar confusiones en la detección de ambos gestos, cuando el movimiento del stylus parte en un punto donde se toca una diapositiva, no se considera el gesto de scroll/zoom en la detección de gestos. Para ello en el `AnalyzeMouseDown` se verifica que el punto donde se inicia el movimiento toque a alguna diapositiva.

#### 5.3.4.6. *Scroll/zoom en la vista de diapositivas*

Cuando se están viendo las diapositivas al interior de una presentación, se puede utilizar el gesto de scroll/zoom para poder verlas todas (para más detalles del gesto, ver Sección 5.3.10.2). Al estar arrastrando el stylus usando ese gesto, se calcula el nuevo factor de zoom en función del zoom anterior y el desplazamiento horizontal del stylus. Con el dato del nuevo factor de zoom se calcula cuántas diapositivas caben en una fila, y se dibujan las diapositivas en la pantalla. El desplazamiento vertical se calcula en función del desplazamiento vertical del stylus.

#### 5.3.4.7. *Importar diapositivas de PowerPoint*

Para lograr leer los datos de una presentación PowerPoint, se utilizan las DLL de interoperatividad de Microsoft PowerPoint 2003. Éstas permiten, entre otras funcionalidades, abrir una presentación y exportar las diapositivas a archivos como PNG y JPG. Sin embargo, es necesario que en el equipo que se usen esas DLL esté instalado PowerPoint, ya que para usar esas funcionalidades, es necesario abrir una instancia de PowerPoint (usualmente sin mostrar la interfaz gráfica para no confundir al usuario).

Las diapositivas de cada presentación son exportadas en un directorio como imágenes PNG. El nombre del directorio depende del nombre del archivo. Por ejemplo, si el archivo de la presentación se llamaba *Clase.ppt* entonces las diapositivas se guardarán como imágenes PNG en un directorio llamado *Clase*.

La aplicación lee los archivos PNG ese directorio, crea un `PresentationDrawable`, y adentro de él crea tantos `SlideDrawable` como imágenes haya en ese directorio, y dentro de

cada diapositiva se crea un dibujable del tipo `ResizableBackgroundDrawable` que está vinculado a una imagen de ese directorio. Ese último dibujable recibe una ruta de archivo en el constructor, y cuando se dibuja, consulta primero un caché LRU <sup>3</sup> de imágenes para ver si está previamente cargada en memoria, si no es así despliega el archivo de imagen de aquella ruta y esa imagen pasa a estar en el caché. Las imágenes del caché están indexadas por la ruta completa del archivo de imagen.

Para lograr que se cambie el zoom de la página actual, cuando un dibujable del tipo `ResizableBackgroundDrawable` está seleccionado se almacena la posición del recuadro que se muestra en su esquina inferior derecha. En el `MouseDown` se verifica que el stylus haya tocado ese recuadro, si es así en el `AnalyzeMouseMove` se va cambiando el zoom del `DrawablesContainer` actual según la posición del stylus en la pantalla. Se limitó el zoom mínimo a 20% para evitar que el dibujable se hiciera tan pequeño tal que no se pudiera seleccionar nuevamente, mientras que no se impusieron cotas superiores para el factor de zoom de la página.

### 5.3.5. Edición de recorridos

Para implementar la noción de recorridos, se creó un dibujable del tipo `PresentationPathDrawable`, que almacena toda la información relativa a él. Cada vez que se modifica el recorrido, se reflejan los cambios en tal dibujable y se manda un mensaje por red a los demás participantes notificando los cambios hechos. Si bien la información de los recorridos se podría haber guardado como variables de instancia en el modo, esa información no podría haberse almacenado en disco porque los modos no son `Drawables`.

La edición propiamente tal de los recorridos es realizada mediante el modo de “Definición de recorridos”, que corresponde a la clase `ChangeSlidePathMode`. Este modo tiene redefinidos el método `DoesOverrideDraw` y `FullDraw`, lo que significa que cuando el modo está activo, él se encarga de dibujar la pantalla. En el método `FullDraw` se dibujan:

- El árbol de nodos: Árbol que tiene como raíz la página principal y muestra de forma jerárquica todos los nodos de la sesión actual. Se dibuja en la parte izquierda de la pantalla.
- El recorrido: Lista ordenada de nodos. Se dibuja como una lista, al lado derecho de la pantalla, en sentido vertical.
- Nodo arrastrado: Cuando se está arrastrando un nodo, en la posición del stylus o mouse se dibuja una copia del nodo para dar la sensación que se está “arrastrando” ese nodo por la pantalla. Si bien la plataforma `SmartCore` ya provee esa funcionalidad, al redefinir `FullDraw` se pierde la posibilidad de usarla de forma directa (puesto que lo que se está dibujando no corresponde a ningún `DrawablesContainer`).

Se decidió que este modo de encargara de dibujar porque de otra forma habría sido necesario traer todos los dibujables del recorrido y del árbol a la página actual (y almacenar cuál

---

<sup>3</sup>Un caché LRU (*Least Recently Used*) elimina el elemento usado menos recientemente cuando el caché se llena para dejar espacio para nuevos elementos.

era su página inicial), o crear unos dibujables que representaran a los nodos del árbol y del recorrido (aumentando innecesariamente la memoria ocupada por la aplicación).

Dado que dibujar el árbol de nodos puede ser muy costoso, la mitad derecha y la mitad izquierda se dibujan “bajo demanda”. Si hay cambios sólo en una mitad de la pantalla se actualiza esa mitad, evitando redibujar la otra mitad. Por ejemplo, cuando se está arrastrando un nodo dentro del lado derecho de la pantalla, sólo la parte derecha de la pantalla es repintada y la parte izquierda no se refresca, lo que evita llamar a la función recursiva que dibuja el árbol de nodos.

Un lado es actualizado si:

- Un nodo es eliminado (por ejemplo, cuando otra persona borra un nodo).
- Se está arrastrando un nodo dentro de ese lado.
- Se está aplicando el gesto de scroll/zoom dentro de ese lado.

El peor caso de este algoritmo es cuando se está arrastrando un nodo, y ése nodo toca ambas mitades de la pantalla, eso ocasiona que se repinte la pantalla completa.

Para poder saber qué regiones de la pantalla corresponden a cuáles nodos, cada vez que se dibujan los nodos en la pantalla (tanto del árbol de nodos como del recorrido) se almacena en una tabla de hash el identificador del nodo (como llave), y el rectángulo que ocupa en la pantalla. Así, al hacer un click en la pantalla (más específicamente, en el método `AnalyzeMouseUp` del modo), se recorren los rectángulos de la tabla de hash para determinar si el click fue realizado encima de un nodo. En un modo que no tiene redefinido el método `FullDraw` (como los otros modos mencionados anteriormente en este trabajo), lo que hace usualmente en la plataforma SmartCore es obtener el `DrawablesContainer` actual, obtener la lista de todos los dibujables contenidos en ella y verificar si donde se hizo click está contenido en el rectángulo de alguno de aquellos `Drawables`.

#### 5.3.5.1. Creación de recorridos

La creación de recorridos se basa en la detección del gesto de la flecha horizontal (realizado por la clase `HorizontalArrowGesture`, en el método `AnalyzeMouseUp` en el “modo Slideshow”). La detección del gesto se muestra en Código 1.

```
1 public new static bool AnalyzeMouseUp(DrawablesContainer gc, Stroke s)
2     {
3         if (s.Rect.Width < minWidth || s.Rect.Height > thresholdY)
4             return false;
5         s.Compress(Math.PI / 3, 0.3);
6         ArrayList v = s.GetPointsList();
7         if (v.Count != 3)
8             return false;
9         Point p1 = (Point)v[0], p2 = (Point)v[1], p3 = (Point)v[2];
10        if ((p3.Y < p1.Y && p3.Y < p2.Y) ||
11            (p3.Y > p1.Y && p3.Y > p2.Y))
12        {
13            //el x del último punto no debe quedar fuera del área
```

```

14         //marcada por los otros dos puntos
15         if ((p1.X + p2.X) / 2 < p3.X && p3.X < p2.X)
16             return true;
17         if ((p1.X + p2.X) / 2 > p3.X && p3.X > p2.X)
18             return true;
19     }
20     return false;
21 }

```

Código 1: Código de método AnalyzeMouseUp de HorizontalArrowGesture

En la línea 5 se comprime el trazo, permitiendo que el trazo sólo tenga a lo más ángulos de 60°, mientras que la línea 7 asegura que para la detección del gesto sólo se consideren trazos comprimidos de 3 puntos. La instrucción de la líneas 10-11 verifica que el último punto esté arriba o abajo de los otros dos. Por último, la líneas 15 y 17 verifican que el valor x del último punto esté acotado por el valor x de los otros dos puntos, con un sesgo hacia el valor x del segundo punto.

Cuando se ha detectado el gesto, se crea un nuevo dibujable que representa al recorrido en la página actual, en la posición final del gesto realizado.

#### 5.3.5.2. Edición de recorridos

Esta funcionalidad es propia del modo de “Definición de recorridos” . Como este modo se encarga del dibujado, para reproducir el efecto de “ir arrastrando un nodo” se tiene que dibujar en la posición del cursor o el stylus el nodo que se está moviendo. En el método `AnalyzeMouseDown` del modo, se detecta si se ha tocado algún nodo (consultando las tablas de hash que tienen como llave los rectángulos y los nodos como valor). Si es así, el modo almacena ese nodo y a qué lado de la pantalla estaba ese nodo (izquierda o derecha), de tal manera que cuando se mueva el stylus por la pantalla, el `AnalyzeMouseMove` se encarga de dibujar ese nodo en la posición donde el stylus está tocando la pantalla. En el `AnalyzeMouseUp` se realizan diferentes acciones dependiendo de qué lado de la pantalla provenía el objeto arrastrado y en qué parte de la pantalla se soltó. Por ejemplo, si se estaba arrastrando un nodo del lado izquierdo al lado derecho, se agrega al recorrido actual, pero si se hace en orden inverso, se saca del recorrido actual.

Al ir arrastrando un nodo en el lado de los recorridos, se muestra una línea negra para dar *awareness* del lugar donde va a quedar el nodo dentro del recorrido. Para poder determinar la posición de dicha línea, se recorre una lista que contiene los dibujables del recorrido que se están mostrando en la pantalla, se determina cuál es el más cercano, y dependiendo de la posición relativa vertical la línea se dibuja encima o bajo el dibujable, con unos pocos pixeles de separación con ese dibujable.

#### 5.3.5.3. Contracción/expansión de árbol de nodos

Cuando se va recorriendo los nodos para ser dibujados en el árbol de la izquierda, se guardan en una tabla de hash los nodos que tiene hijos como llave y un booleano como valor, de manera que los hijos de ese nodo puedan ser mostrados u ocultos por el usuario. En el método recursivo encargado de dibujar el árbol se consulta la tabla de hash usando el nodo

como llave, el valor que se obtenga se usa para decidir si se dibujan los hijos de ese nodo. Al hacer tap sobre los símbolos (+) o (-) se actualiza el valor en la tabla de hash.

#### 5.3.5.4. *Scroll/zoom*

Para que ambos lados tengan un factor de scroll y un factor de desplazamiento vertical independientes entre sí, se tiene un objeto por cada mitad que almacena esos valores. Dependiendo en qué mitad de la pantalla esté el punto donde se haga un `MouseDown`, el gesto encargado del scroll/zoom (ver Sección 5.3.10.2) va a usar el objeto asociado a esa mitad y sólo va a realizar su trabajo en esa mitad de la pantalla, sin afectar a la otra mitad.

#### 5.3.5.5. *Ir a página de un nodo*

Al hacer doble tap sobre un nodo, la aplicación se va a la página interior de ese nodo. Para lograr eso, se usa el gesto `DoubleClickGesture` que provee `SmartCore` para detectar los doble click. Cuando se detecta un doble click, se busca en las tablas de hash si el punto actual toca a algún nodo en la pantalla. Si es así, se obtiene ese nodo y se cambia la página por la página interior del nodo.

#### 5.3.5.6. *Definición de recorrido activo*

Para asegurar que sólo pueda haber a lo más un recorrido activo, se guarda en el modo una variable que indica cuál es el recorrido activo. Si esa variable es nula, no hay ningún recorrido que esté activo en ese momento.

Al seleccionar un recorrido, se dibujan iconos contextuales a sus costados por medio del método `ManageContextualOptions`. En ese método se agrega el botón que permite promover el recorrido al estado de recorrido activo.

#### 5.3.5.7. *Navegación usando el recorrido*

El gesto para cambiar diapositivas es una flecha horizontal, que corresponde a la clase `HorizontalArrowGesture`. La detección del gesto se realiza en el “modo del pizarrón”. Cuando el gesto es detectado, primero se determina el sentido del gesto, es decir, si la flecha va a la izquierda o a la derecha. Eso se logra comparando la coordenada X del primer punto con la coordenada X del último punto del gesto. Después, se verifica que nodo actual esté presente en el recorrido activo. Si es así, se cambia de página adelante o atrás del recorrido, según el sentido de la flecha del gesto.

Para mostrar la animación cuando se está realizando el cambio de una página a otra, se crea un objeto `Bitmap` sobre el cual se dibuja la nueva página llamando al método `DrawThumbnail()` de `DrawablesContainer`. Ese mapa de bits es dibujado en la pantalla cada cierto tiempo, posicionado cada vez más a la izquierda o la derecha según el sentido de la transición, hasta que alcanza el otro extremo de la pantalla. Se utilizó un mapa de bits intermedio en vez de dibujar la nueva página directamente a la pantalla, debido a que esa última opción provocaba parpadeos y retraso en la animación de la transición.

### 5.3.6. Definición de grupos

El modo encargado de la definición de grupos es el `MakeGroupsMode`. Este modo está mayoritariamente basado en un modo similar que existe en `SmartCore`.

Los grupos están representados de forma visible por dibujables del tipo `GroupDrawable` que provee `SmartCore`. Pero la noción de grupos es en realidad provista por la capa de red de `SmartCore`. Los métodos de la capa de red son accesibles por el objeto de la clase `SmartNode` que contiene `SmartDraw`. `SmartNode` contiene los métodos `join(groupId)` y `leave(groupId)` que permiten unirse y salirse de un grupo, respectivamente.

#### 5.3.6.1. Creación de grupos

Para crear un grupo, se debe detectar el gesto de la “L” en el `AnalyzeMouseUp` y el punto inicial del gesto debe estar en el área de grupos. El número del grupo se determina al azar, dentro de un intervalo fijo. Dado que es un intervalo finito, hay un límite superior a la cantidad de grupos que pueden haber a la vez, que es igual a la cantidad de números enteros diferentes dentro de ese intervalo.

#### 5.3.6.2. Asignación de estudiantes a un grupo

En el `AnalyzeMouseUp` del modo, se verifica si se está arrastrando un participante y el punto actual está sobre un grupo. Si es así, el dibujable que representa al participante se cambia a la página interior del grupo, y se manda un mensaje al participante para indicar que fue ingresado a un determinado grupo.

#### 5.3.6.3. Sacar a estudiantes de un grupo

Para lograr esta funcionalidad, es necesario disponer de una forma de ver los integrantes que contiene un grupo: al hacer click sobre un grupo, se crean dibujables que representan a los participantes que contiene el grupo, y se ponen en el área de participantes. Los participantes que no han sido asignados a ningún grupo se ocultan, poniéndolos en la “lista de dibujables ocultos” de `SmartDraw`. Los dibujables que estén en esa lista no son dibujados, incluso si su método `isVisible()` devuelve verdadero. Cuando un grupo se deselecciona, se vacía la “lista de dibujables ocultos”, y se eliminan los dibujables creados para representar a los estudiantes ya asignados a grupos.

Cuando se detecta el gesto de la cruz, se busca a todos los dibujables de participantes que tocó. A cada uno de ellos, se le saca de su grupo, y se manda un mensaje `GroupMessage` al participante indicando que ya no pertenece a ningún grupo.

#### 5.3.6.4. Eliminación de grupos

Cuando se hace el gesto de la cruz, por cada grupo que haya tocado, se sacan a todos los participantes de ese grupo, se manda un mensaje a esos participantes informando su eliminación del grupo, y se procede a borrar el dibujable.



### 5.3.7. Definición de preguntas

Toda la información referente a una pregunta se almacena en un `QuestionDrawable` ubicado en la página que contiene una “pregunta”. Ese dibujable contiene los identificadores de los dibujables que forman parte de la respuesta correcta, respuesta incorrecta y enunciado. Se guardan los identificadores y no los dibujables en sí evitar enviar información de más al serializar la pregunta y mandarla a los estudiantes. Cabe notar que los `QuestionDrawable` nunca se dibuja en la pantalla, es invisible para evitar su manipulación y borrado accidental.

La aplicación trata de no crear más de un `QuestionDrawable` a la vez en la misma página, porque podría ocasionar que dos alumnos estén en la misma página, pero tratando de contestar preguntas diferentes.

Una instancia de `QuestionDrawable` contiene la siguiente información:

- Tipo de la pregunta: si es de selección simple, múltiple, secuencia o libre. Una enumeración guarda esos valores.
- Identificadores de los dibujables que son parte del enunciado, de las alternativas correctas o de las alternativas incorrectas.
- Un booleano de nombre `Ready` que indica si la pregunta ha sido enviada por el profesor. Si bien este dibujable puede ser recibido por los alumnos antes que el profesor permita contestar la pregunta, si ese valor booleano es verdadero entonces los alumnos están facultados para responder a esa pregunta.

En el caso de una pregunta de secuencia, el orden de los identificadores de la respuesta correcta es el orden correcto de la respuesta.

Las respuestas de los alumnos son objetos del tipo `QuestionAnswerMessage`, que heredan de `PostProcessable`. Este objeto almacena el identificador de la pregunta, las respuestas dadas por el alumno y el estado de la pregunta. El estado puede ser (desde el punto de vista de los alumnos):

- `Unshared`: No se ha enviado la respuesta ni al profesor ni al grupo.
- `AnsweringAndNegotiating`: Dentro de un mismo grupo, los alumnos han dado diferentes respuestas a problema. Se deben poner de acuerdo.
- `AgreedAndSending`: Alumnos del grupo se pusieron de acuerdo en una respuesta y la han enviado al profesor.
- `ConfirmedAsWrong`: La respuesta dada por los alumnos es incorrecta.
- `ConfirmedAsCorrect`: La respuesta de los alumnos es correcta.

Los cambios de estados del `QuestionAnswerMessage` depende si la pregunta a la que corresponde es libre o no.

- En el caso de la pregunta libre, el diagrama de estados se puede apreciar en la Figura 48. Sólo cuando los alumnos han enviado la respuesta al profesor (estado *AgreedAndSending*), él puede marcar la respuesta como correcta o incorrecta.
- En una pregunta no libre, el diagrama de estados correspondiente es el de la Figura 49. Dado que el dibujable de la pregunta tiene un arreglo que contiene cuáles son las respuestas correctas, no es necesario pasar por el estado *AgreedAndSending*, pues la aplicación puede automáticamente determinar si la respuesta fue correcta o no.

Cabe notar que si la pregunta es individual, el “grupo” queda reducido a una sola persona, por lo cual al enviar una respuesta es considerado que hay consenso en el “grupo”.

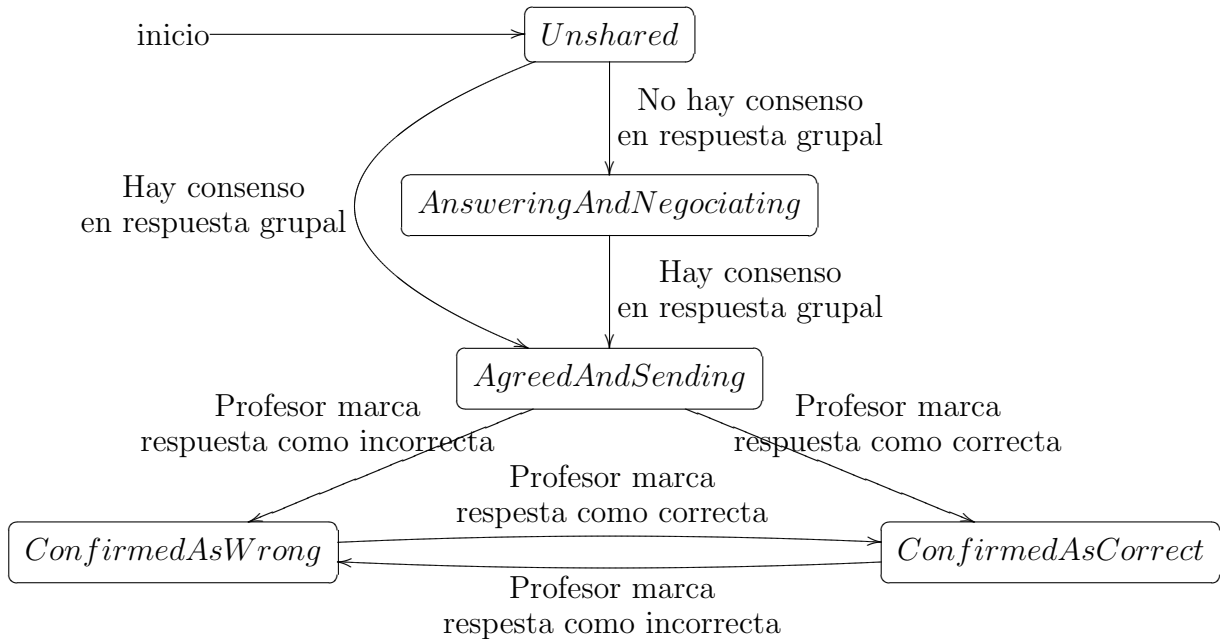


Figura 48: Diagrama de estados de una pregunta libre

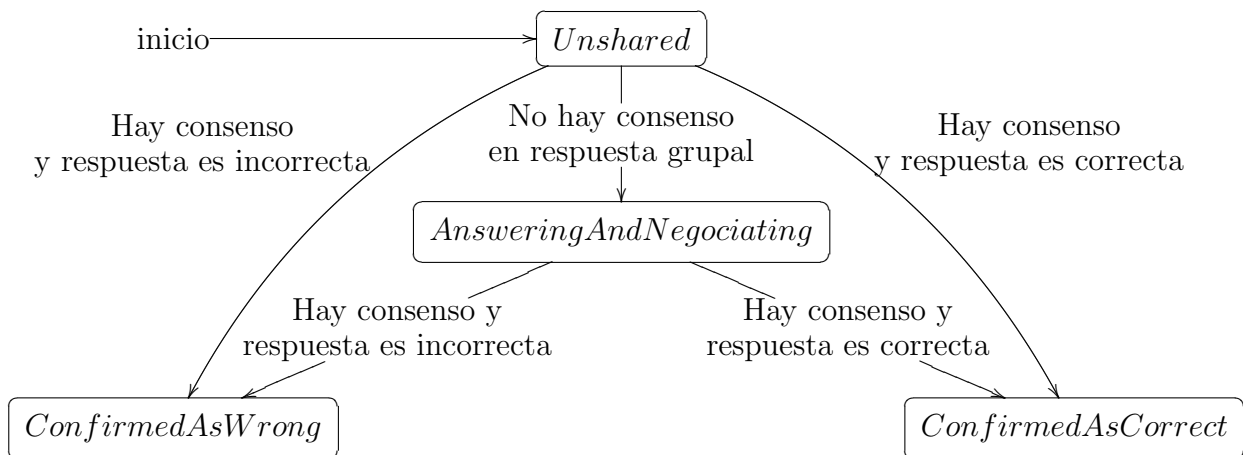


Figura 49: Diagrama de estados de una pregunta no libre

#### 5.3.7.1. *Creación de una nueva pregunta*

Cuando se entra a una página estando en el modo de “Definición de preguntas”, se crea un dibujable de la pregunta si es que no había uno con anterioridad, asegurando que en una página a lo más haya un `QuestionDrawable`.

Cuando se cambia el tipo de la pregunta o se define un dibujable como parte de la pregunta (ya sea como alterativa correcta/incorrecta o enunciado), esos cambios se ven guardados en el dibujable de la pregunta.

#### 5.3.7.2. *Envío de preguntas a los estudiantes en forma individual*

Cuando se presiona el botón Enviar, se les envía a los estudiantes un mensaje para indicarles que se puede responder a la pregunta. Que los estudiantes tengan un `QuestionDrawable` en su área de trabajo no implica necesariamente que es una pregunta que pueda ser respondida, puesto que el profesor puede estar todavía modificando su contenido antes de dejar que los alumnos la respondan. Cuando `questionDrawable.Ready` es verdadero, entonces la pregunta puede ser respondida por los estudiantes.

#### 5.3.7.3. *Envío de preguntas a los estudiantes en forma grupal*

Cuando se presiona el botón de Envío grupal, se les envía a los estudiantes un mensaje para indicarles que se puede responder a la pregunta. A diferencia de una pregunta individual, cuando un `QuestionDrawable` de una pregunta grupal es recibido, en el dispositivo receptor se obtienen cuáles son los integrantes del grupo.

#### 5.3.7.4. *Confirmar respuestas libres como correctas/incorrectas*

Cuando en la página actual hay una pregunta libre, y se hace el gesto de un victo o una cruz, se recorren las respuestas visibles en esos momentos y se cambia su estado a correcto (si se usó el gesto del victo) o cruz (si se utilizó el gesto de la cruz), modificando el estado del `QuestionAnswerMessage` respectivo. Ambos gestos vienen implementados en `SmartCore`.

#### 5.3.7.5. *Ver respuestas de los alumnos*

Cada `QuestionDrawable` tiene una tabla de hash que contiene información relativa a qué han contestado los alumnos. Si un alumno no está en ese hash, no ha enviado su respuesta al profesor. Esa tabla de hash permite llenar la barra de participantes.

En el caso de las preguntas libres, se utiliza un `HideManager` (ver Sección 5.3.10.3) para filtrar los trazos de los alumnos. Los trazos de los estudiantes son del tipo `StudentAnswerStroke` que se muestran según lo diga el `HideManager` del modo.

Para las preguntas no libres, sólo se muestra la respuesta de un estudiante a la vez. El modo contiene una variable que indica la respuesta de quién se está viendo. Para indicar la respuesta del alumno, se pinta con otro color el fondo del nodo de la respuesta, y el resto de las alternativas se pintan con color de fondo normal.

### 5.3.7.6. *Superposición trazos en preguntas*

En las preguntas libres, es posible superponer los trazos de las respuestas dadas por los estudiantes. Para lograr eso, los trazos que hacen los estudiantes al responder una pregunta libre es del tipo `StudentAnswerStroke`, que tiene redefinido el método `IsVisible` de tal forma que los compañeros del grupo del creador del dibujable y el profesor pueden verlo.

### 5.3.8. **Responder preguntas**

Cuando la página actual es una pregunta, este modo consulta al `QuestionDrawable` respectivo para saber cuáles son las alternativas correctas e incorrectas.

En el método `DrawAfter` se dibujan los iconos que van a los costados de las preguntas. Como este método es llamado cuando ya se ha dibujado toda la pantalla (fondo de pantalla, dibujables y widgets), no se corre el riesgo que se dibuje encima de los iconos que van a los costados de las preguntas, tapándolos de la vista del usuario.

#### 5.3.8.1. *Responder de manera individual*

En una pregunta cerrada, cuando los alumnos presionan el botón de “Enviar”, se envía un mensaje que contiene una lista de todos los identificadores de los dibujables que el estudiante marcó como respuesta. En el caso de una pregunta libre, esa lista se deja vacía, pues los trazos que el estudiante ha usado para responder se han enviado al profesor a medida que los va dibujando.

Una vez que el estudiante ha enviado la respuesta al profesor, el estado de la pregunta pasa a ser `AgreedAndSending` en el caso de una pregunta libre, o `ConfirmedAsCorrect` o `ConfirmedAsWrong` en el caso de una pregunta no libre. (como se ve en la Figuras 48 y 49)

El icono del modo cambia de acuerdo al tipo de pregunta que haya en la página actual, y si el alumno la ha respondido. Así, por ejemplo, un alumno puede saber si tiene que responder una pregunta de selección simple o de selección múltiple. Para lograr ese cambio, se redefinió el método `PageChanged` del modo, de manera que al entrar a una página nueva se obtiene información acerca de la pregunta y se cambia al icono que represente al tipo de la pregunta y si la pregunta ha sido respondida o no por el alumno.

#### 5.3.8.2. *Responder de manera grupal*

Cada vez que un estudiante de un grupo envía su respuesta, los otros integrantes del grupo la almacenan en una tabla de hash, tomando como llave el nombre del estudiante. Así, un alumno puede saber cuántas personas tienen la misma respuesta que él. Para que la respuesta sea enviada al profesor, algún alumno del grupo debe detectar que todos los estudiantes del grupo han enviado su respuesta y la respuesta es la misma. En tal caso, se envía un `QuestionAnswerMessage` al profesor y al resto del grupo.

La forma de detectar si hay consenso en la respuesta grupal depende si la pregunta es libre o no.

- Si la pregunta no es libre, para que haya consenso es necesario que la cantidad de respuestas correctas en el grupo (incluyendo al propio estudiante) sea igual a la cantidad de estudiantes en el grupo.
- Si la pregunta es libre, para que haya consenso sólo basta que no se hayan dibujado o modificado trazos desde que alguien haya enviado la respuesta.

#### 5.3.8.3. *Ver preguntas recibidas*

Se creó un modo adicional para que el alumno pudiera ver las preguntas que ha recibido. Ese modo se encarga del dibujado completo de la pantalla. Al entrar al modo se hace una búsqueda de preguntas por todos los nodos de la sesión actual, y se almacenan en una lista. Cada vez que se dibuja en pantalla las preguntas se consulta esa lista.

Para el dibujado del gráfico, a medida que se van dibujando las preguntas se va almacenando información acerca de la cantidad de respuestas correctas e incorrectas que ha dado el alumno. Ya dibujadas las preguntas, se calcula el porcentaje de cada una de las opciones, y se dibuja esa opción como una fracción del ancho de la pantalla. Un ejemplo: si las preguntas correctas son el corresponden al 25 %, entonces la barra de preguntas correctas ocupa 25 % del ancho de la pantalla.

Este modo también hace uso del gesto de scroll/zoom descrito en la Sección 5.3.10.2.

### 5.3.9. **Resultados de las preguntas**

A medida que el profesor va recibiendo los resultados de las preguntas, éstas se van almacenando en el `QuestionDrawable` respectivo, en una tabla de hash donde la llave es el identificador del estudiante que ha enviado su respuesta.

En una pregunta libre, se pueden superponer las respuestas de todos los estudiantes para que el profesor pueda identificar alguna tendencia de manera rápida.

#### 5.3.9.1. *Vista de las respuestas por alumno*

Cuando se entra al modo, se hace una búsqueda de todas las preguntas que hay en la sesión actual. Cada pregunta guarda cuáles han sido las respuestas que han dado los alumnos.

Como una misma fila representa a una misma pregunta, antes de dibujar la página con la respuesta de un alumno se realiza la acción que corresponda según el tipo de pregunta:

- Si la pregunta no es libre, las alternativas seleccionadas por ese alumno se dibujan con color de fondo de otro color. Antes de dibujar la página de la respuesta, se cambia el `SolidBrush` (objeto utilizado por C# para pintado de superficies) de las alternativas marcadas por el estudiante. Ese `SolidBrush` es usado para pintar el fondo del dibujable.
- En el caso de una pregunta libre, se necesita mostrar los dibujables que corresponden a la pregunta junto a los dibujables de respuesta del alumno. El modo hace uso de

un `HideManager` (ver Sección 5.3.10.3) con el cual se ocultan los trazos de los otros estudiantes excepto los trazos del estudiante del cual se está viendo la respuesta.

### 5.3.9.2. Gráficos por estudiante y pregunta

Hay un *widget* que permite visualizar en forma de gráfico los resultados en el caso de una pregunta no libre. El gráfico de barras queda representado por la clase `vBarGraph`, que hereda de la clase abstracta `vGraph`. En esa clase están dissociados los datos y el componente que grafica los datos. El gráfico recibe un objeto del tipo `AbstractModel` que contiene los datos a graficar. Entonces, en el modo de resultados, en vez de tener en memoria varios `vBarGraph`, cada uno representando a un gráfico que se dibuja en pantalla, se tiene un `AbstractDataModel` por cada gráfico en pantalla, y una sola instancia del objeto de gráfico de barras.

Haciendo una analogía con el modelo MVC (Modelo-Vista-Controlador), `vBarGraph` correspondería a la vista y al controlador mientras que `AbstractDataModel` representa al modelo. En la Figura 50 se muestra un diagrama de clases de las clases `vGraph`, `AbstractDataModel`, `AbstractRenderer` y derivadas.

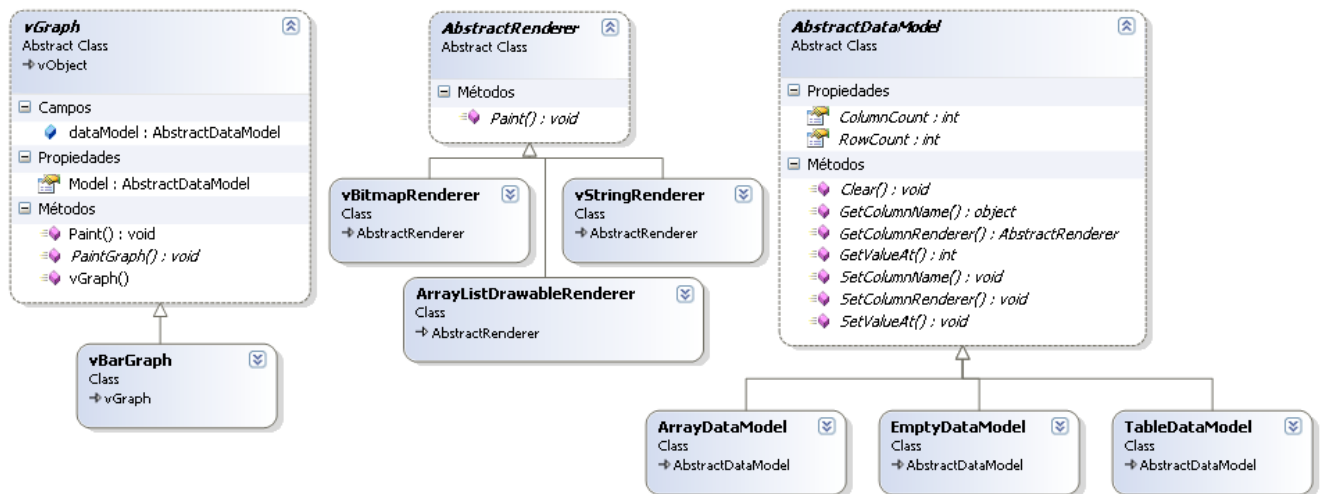


Figura 50: Diagrama de clases de VGraph y clases relacionadas

En el modelo `AbstractRenderer`, los datos son vistos como una matriz, donde cada fila representa una serie y cada lugar de la matriz (determinado por el par  $(fila, columna)$ ) representa un dato (como número entero). Los nombres que tienen los valores en el eje X del gráfico se pueden fijar y obtener mediante los métodos `SetColumnName` y `GetColumnName`, respectivamente. Los valores del modelo se pueden fijar con `SetValueAt` y obtener llamando al método `GetValueAt`.

Como el nombre de cada fila puede ser un objeto de cualquier clase, cada columna del modelo tiene un `AbstractRenderer` encargado de dibujarla en pantalla, a través del método

abstracto `Paint(object)`. Por ejemplo, `vStringRenderer` dibuja lo que devuelve el método `ToString()` del objeto, mientras que `vBitmapRender` trata el objeto como una imagen. `ArrayListDrawableRenderer` se encarga de pintar dibujables de `SmartDraw`, hacia abajo (como es en el caso del gráfico de respuestas a una pregunta de selección múltiple o de secuencia).

En la aplicación específicamente se utilizó la clase `ArrayDataModel`. Esa clase tiene como representación interna una matriz bidimensional para los datos.

### 5.3.9.3. *Scroll/zoom*

Este modo también tiene un gesto de scroll/zoom (para más detalles ver Sección 5.3.10.2), pero funciona de manera diferente. Dado que en este modo, además del desplazamiento vertical y el zoom, es necesario que el gesto realice desplazamiento en sentido horizontal.

## 5.3.10. Funciones generales

En esta sección se describen funcionalidades implementadas que son utilizadas por más de un modo. Esas funcionalidades fueron implementadas fuera de los modos para minimizar el acoplamiento entre ellos.

### 5.3.10.1. *Diferenciación de trazos de participante por color*

La primera vez que un dispositivo inicia alguna aplicación basada en `SmartCore` se le pide al usuario escoger un color y un nombre para poder identificar al usuario. La plataforma `SmartCore` no refleja en pantalla quiénes son los creadores de los trazos, mientras que los nodos son pintados del color de su creador. La manera usada para cambiar el color de los trazos según su creador es crear trazos que hereden de `Stroke` y redefinir el método de pintado, fijando el lápiz dibujador con el color asociado al participante, y posteriormente llamando al método de dibujar de `Stroke`.

### 5.3.10.2. *Gesto de scroll/zoom*

El gesto de scroll/zoom es utilizado por varios modos para poder escalar y mover el contenido en la pantalla. El algoritmo usado para calcular el nuevo factor de scroll y zoom es parte de `SmartCore`.

Para poder calcular el factor de zoom mínimo se necesita saber cuánto es el tamaño total (ancho y alto) de los elementos a los cuales se les aplica del gesto de zoom/scroll. El factor de zoom mínimo es el que logra que todos aquellos elementos se vean a la vez en la pantalla.

El desplazamiento se limita de tal forma de evitar que ninguno de los dibujables que están afectos al desplazamiento queden ubicados fuera de los límites de la pantalla. Para ese cálculo se necesita saber cuál es el alto total de los elementos a los cuales se les aplica el gesto.

El `ScrollZoomGesture` se encarga de aplicar el gesto de scroll y zoom. Ese gesto es llamado en el `AnalyzeMouseMove` del modo, el gesto llama de vuelta al modo para saber cuál es el rectángulo que ocupan los elementos afectados por el gesto. Finalmente, el gesto calcula el factor de zoom y llama al modo para que actualice la escala y desplazamiento de sus dibujables.

### 5.3.10.3. Ocultamiento de dibujables

Ciertos modos de la aplicación proveen mecanismos para mostrar u ocultar dibujables, según un criterio dado. Para lograr eso, se crearon unos dibujables cuyo método `isVisible()` respondía verdadero o falso según lo dijera el `HideManager` del modo.

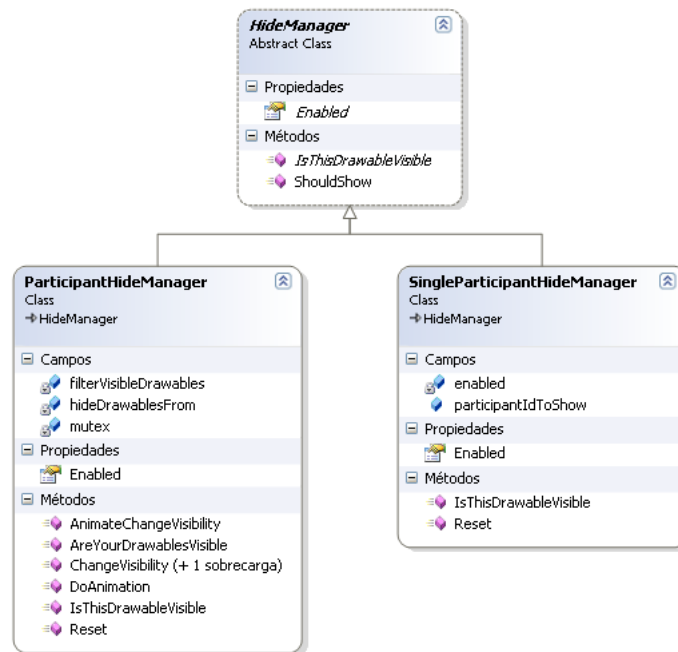


Figura 51: Diagrama de clases de `HideManager` y sus clases derivadas

`HideManager` (ver Figura 51) es una clase abstracta que determina si un dibujable debe ser mostrado en la pantalla. Redefiniendo el método `IsThisDrawableVisible` permite definir criterios personalizados para decidir si mostrar en pantalla: ese método devuelve verdadero si se debe pintar el dibujable que recibe como parámetro.

Las clases que heredan de `HideManager` son:

- `ParticipantHideManager` filtra dibujables según su creador. Las instancias de esa clase contienen una lista de participantes cuyos dibujables se ocultan. Esta clase es utilizada en el filtrado de trazos por estudiante del “modo del pizarrón” (ver Sección 5.3.2.2) y el filtrado de trazos para respuestas de preguntas libres (ver Sección 5.3.7.6).
- `SingleParticipantHideManager` filtra los dibujables de todos los participantes excepto uno. La idea es que sólo los dibujables de una persona sean visibles.



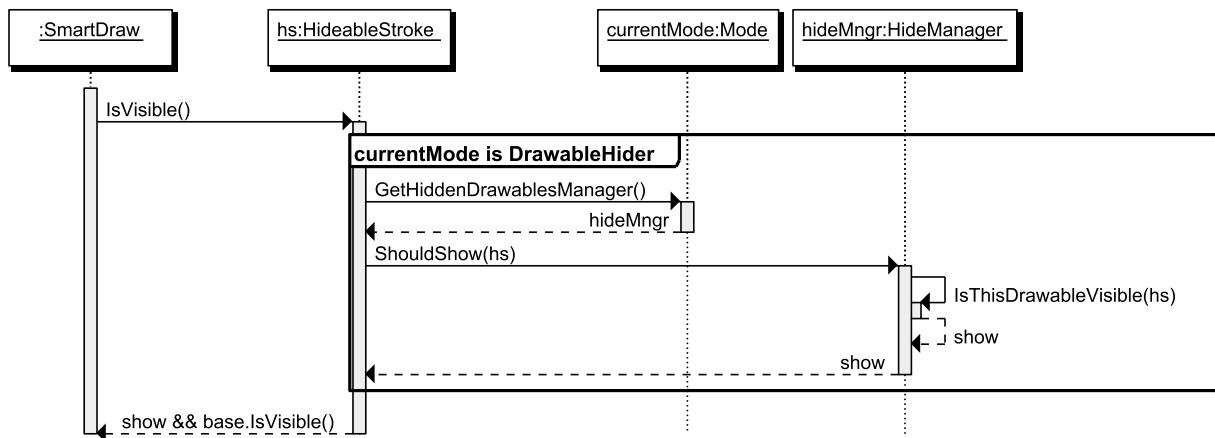


Figura 52: Diagrama de secuencia de HideManager

El proceso de ver si un dibujable se despliega en pantalla preguntando al HideManager es el siguiente (ver Figura 52):

- El thread de dibujado de SmartDraw llama al método isVisible del dibujable para decidir si lo tiene que desplegar en pantalla.
- El dibujable (como por ejemplo, un HideableStroke) invoca al método estático ShouldShow(this) del HideManager.
- Se ve si el modo actual implementa la interfaz DrawableHider. Si no la implementa, devuelve true, con lo cual el dibujable es pintado según diga el método isVisible de su clase padre.
- Si modo implementa la interfaz anteriormente mencionada, del modo actual se obtiene una instancia de HideManager (esto pues cada modo puede tener oculto diferentes dibujables, para mantener independencia entre ellos), y se llama al método IsThisDrawableVisible(), y se le pasa como parámetro el dibujable. Si este método retorna false, no se dibuja el objeto. En caso contrario, se muestra en pantalla si y sólo si base.isVisible() retorna verdadero.

#### 5.3.10.4. Gestos genéricos

En cierto punto en el desarrollo de la aplicación se tuvieron que crear gestos para poder crear trazos. La plataforma SmartCore provee la clase Stroke que representa a un trazo, pero en más de una ocasión se crearon trazos heredando de Stroke para agregarle más funcionalidades. Para evitar la creación de nuevos gestos cada vez que se quisiera usar un trazo diferente, se optó por crear una gesto genérico o parametrizable GenericStrokeGesture aprovechando los generics de C#.

Para implementar ese gesto se realizó una variación del gesto implementado en SmartCore que es usado para crear trazos (Stroke).

La declaración de la clase es la siguiente:

```
public class GenericStrokeGesture<T,U> : Gesture
    where T : Stroke where U : PointDrawable
```

El gesto es parametrizable por dos clases: una clase (T) que sea un dibujable representando un trazo , y la otra (U) que sea un punto. La cláusula **where T : Stroke** fuerza a que el trazo a usar herede de la clase **Stroke**. De igual manera, el dibujable del punto debe heredar de **PointDrawable**.

```
1 public new static bool AnalyzeMouseUp(DrawablesContainer gc, Stroke sg)
2 {
3     long ticks = Environment.TickCount - LastStrokeEnd;
4     if (gc.BlockedForEditing || (ticks > MaxTicksForPoint && sg.Rect.Size.Width + sg.Rect.
5         Size.Height < 8))
6         return false;
7
8     Drawable addedStroke;
9     if (sg.Rect.Size.Width + sg.Rect.Size.Height >= 8)
10    {
11        (...)
12        Point[] pts = (Point[])sg.MyPoints.ToArray(typeof(Point));
13        (...)
14        Type tipo = typeof(T);
15        ConstructorInfo constructor = tipo.GetConstructor(new Type[]{typeof(Point[])});
16        if (constructor != null)
17            addedStroke = (Stroke) (constructor.Invoke(new Object[] {pts }));
18        else
19            addedStroke = sg;
20    }
21    (...)
22    return true;
23 }
```

Código 2: Método AnalyzeMouseUp de la clase GenericStrokeGesture

Lo que hace el método de creación de trazos (mostrado en Código 2) es obtener el tipo del dibujable del trazo, obtener el constructor que reciba un arreglo de puntos como parámetro (que siempre va a existir pues **Stroke** tiene ese constructor), y llamarlo para obtener una nueva instancia del objeto. Un ejemplo de uso de esta clase sería:

```
GenericStrokeGesture<SomeStroke , SomePointDrawable >.AnalyzeMouseUp(
    drawableContainer , stroke );
```

También se creó un gesto para la creación de nodos, basándose en el gesto creador de nodos (**CreateRectNodeGesture**, que creaba dibujables del tipo **RectDrawable**), ese gesto es **GenericCreateRectNodeGesture**, cuya declaración es la siguiente:

```
public class GenericCreateRectNodeGesture<T> : Gesture
    where T : RectDrawable
```

Se usa de manera análoga al gesto genérico descrito anteriormente, sólo que el parámetro genérico del gesto tiene que heredar de **RectDrawable** (clase que representa nodos). El gesto utiliza el hecho que la clase de los nodos tiene un constructor que recibe las dimensiones (alto y ancho) del objeto, por lo cual el gesto busca ese constructor en la clase que le pasaron como parámetro.

## 6. Conclusiones

Este trabajo presenta el desarrollo de una aplicación que facilita a un profesor realizar prácticas pedagógicas en la sala de clases, mediar actividades destinadas a gestionar el conocimiento de los alumnos, recopilar información acerca de su desempeño, así como a la gestión del contenido a exponer, anotar o intercambiar en una sala de clase y en tiempo real. La aplicación fue diseñada para que tanto alumnos como profesores puedan utilizarla y permite, a diferencia de las aplicaciones similares existentes a la fecha, mostrar contenido a los alumnos, permitir el trabajo colaborativo, ejecutarse en dispositivos como PCs, Tablet-PCs y PDAs, mientras están interconectados a través de redes *ad-hoc* inalámbricas. El profesor tiene mayores funcionalidades de interacción: definición de grupos, filtrar contenido visible, otorgar permisos para edición de contenido, etc., mientras que los alumnos pueden hacer anotaciones, compartirlas, responder preguntas individuales o grupales y escribir en el área del pizarrón para complementar las explicaciones del profesor o para indicar que no entendió algún tema en particular.

Esta misma aplicación se puede utilizar en otros contextos y ámbitos de acción, ya que permite la creación, validación, distribución, revisión y aplicación del conocimiento a través del manejo de los datos e información de usuarios que trabajan de manera individual o colaborativa, pudiendo uno de ellos convertirse en moderador que tiene mayores funcionalidades de interacción (como lo tiene un usuario profesor).

Respecto a los objetivos planteados inicialmente:

- Se identificaron las características relevantes para un software usado en la sala de clases por medio de una revisión bibliográfica. Los resultados de esta revisión revelaron que las aplicaciones destinadas para su uso en la sala de clases deben ser fáciles de usar, separar las funcionalidades del profesor y los alumnos, permitir la comunicación entre el profesor y sus estudiantes (por ejemplo, cuando el profesor da *feedback* a sus alumnos sobre cómo han respondido a una pregunta o cuando el alumno complementa las explicaciones del profesor), y favorecer el trabajo colaborativo. Por otro lado, la utilización de software basado en diapositivas ocasiona que la enseñanza sea entregada de forma lineal y con poco nivel de interactividad por parte de los alumnos, haciendo difícil al profesor estimar cuánto saben los alumnos de la clase, por lo cual es necesario tener un método para que el profesor pueda modificar la forma de mostrar el contenido. La revisión bibliográfica también reveló que el software usado en la sala de clases debe permitir al profesor definir de qué manera exponer los contenidos a la clase y permitir la creación de preguntas destinadas a los alumnos en tiempo real para que el profesor haga *assessment* a los alumnos y los alumnos hagan *reflection* al ver cómo han respondido.
- Se identificaron características que deben poseer la interfaz humano-computador de un software que va ser utilizado en la sala de clases. La interfaz tiene que ser lo más simple posible, y una forma de lograr simplicidad es reducir la cantidad de componentes gráficos en dispositivos de pantalla táctil. Una manera natural de interactuar con dispositivos que tienen pantallas táctiles es mediante gestos del stylus, puesto que se

acerca al modelo del lápiz y papel, y la interfaz basada en gestos permite ahorrar espacio en la pantalla puesto que se prescindieron de componentes gráficos (menús, botones) para poder activar funcionalidades. Sin embargo, algunas componentes de interfaces tradicionales, tales como menús y botones, se mantuvieron debido a la dificultad que representaba sustituirlos por gestos. Los gestos implementados en la aplicación tienen varios propósitos: edición (crear, mover, duplicar, eliminar y borrar contenido), cambios en la visualización (zoom in/out, desplazamiento horizontal/vertical), navegación (entrar al interior de los nodos, utilización de recorrido para cambiar de página) y específicos de la aplicación (alumnos responden preguntas seleccionando la alternativa correcta o profesor marca respuestas como correctas/incorrectas). La plataforma provee una homologación de los tamaños de pantalla de los diferentes dispositivos presentes en la sesión, permitiendo compatibilizar las interfaces de usuario, los dispositivos que tengan menor tamaño de pantalla puedan ver lo que está anotando alguien en un computador con la aplicación ejecutándose a pantalla completa.

- Se diseñó e implementó una aplicación capaz de dar apoyo tanto al profesor como a los alumnos en la sala de clases, se logró que la aplicación funcionara tanto en Tablet-PC, PDA y pizarra electrónica. La aplicación, al estar basada en un framework llamado SmartCore, utilizó componentes para la compatibilidad de Tablet-PC, PDA y pizarra electrónica, la comunicación entre dispositivos, el manejo de gestos, y la definición de grupos de trabajo colaborativo. Se hicieron pruebas de la aplicación a cada uno de los componentes para asegurar su correcto funcionamiento.
- La evaluación del uso del software con los usuarios reales del sistema (profesores y alumnos) no se realizó, dado que la evaluación es un proceso complejo que debe ser instanciado en una sala de clases. La evaluación tiene como fin descubrir qué funcionalidades de la aplicación deben trabajarse más a fondo o cómo mejorar la interfaz de usuario.

La plataforma SmartCore es capaz de detectar al resto de los participantes que están ejecutando el programa en la misma red, sin intervención por parte del usuario. Además maneja los casos cuando un participante se sale de la red y vuelve, o cuando un participante se conecta de forma tardía a la sesión. No es necesaria una infraestructura compleja de red (como tener una conexión a Internet o acceso a un router), puesto que la aplicación puede utilizar una red inalámbrica *ad-hoc*.

Los requerimientos de memoria son satisfechos por el hardware de las PDAs. Nunca ocurrieron problemas de memoria al ejecutar la aplicación, pero lo que sí se vio afectado fue el rendimiento de la aplicación cuando habían muchos objetos de la aplicación creados en la sesión actual. En ese último escenario la aplicación respondía de forma tardía a las interacciones por parte del usuario. En el caso de las Tablet-PC no se apreciaron problemas de memoria o rendimiento dado su mayor poder de procesamiento y capacidad de memoria, en comparación con los dispositivos móviles.

## 7. Trabajo futuro

En el desarrollo del sistema quedan algunas características que requieren mejoras, como la existencia de un rol de proyector, mejora en la serialización de objetos, diferenciación de trazos por usuario, simplificación del diseño de los modos, uso de clases genéricas o parametrizables en SmartCore, el aviso a los modos de una nueva sesión, mejoras a la edición a mano alzada, inserción de objetos en las presentaciones y la manipulación de contenido de las diapositivas PowerPoint. A continuación se detallan los puntos nombrados anteriormente.

### 7.1. Rol de proyector

Un problema aparece cuando se quiere utilizar el sistema con el computador del profesor conectado directamente con un proyector: si el profesor está creando preguntas, todos los alumnos van a poder el proceso de creación, especialmente cuáles son las respuestas correctas, antes que el problema sea enviado a ellos. Una forma de arreglar eso sería en agregar un tercer rol al sistema: un rol “público”, y que el computador con ese rol esté conectado con el proyector. Ese rol muestra lo que está haciendo el profesor, pero evita mostrar la creación de preguntas.

Actualmente, una manera de evitar esto es tener las preguntas creadas antes de la clase.

### 7.2. Mejora en serialización de objetos

Otro aspecto a estudiar es la serialización binaria. Actualmente la serialización de los objetos que realiza la plataforma SmartCore es serialización XML, pero los tags XML ocupan gran parte del espacio en el archivo, ocasionando que la recepción de los objetos grandes se demore más de lo normal, dado que se tiene que procesar entero el archivo XML antes de la creación local del objeto. Habría que experimentar con la serialización binaria de objetos para ver si hay mejoras en velocidad y rendimiento.

### 7.3. Pruebas de usabilidad del sistema

Falta realizar pruebas de usabilidad con los usuarios reales del sistema (profesores y alumnos) para obtener retroalimentación sobre el sistema, y ver qué aspectos mejorar, en términos de interfaz y funcionamiento. La idea es que la interfaz de usuario no confunda a los alumnos y profesores. Además aquello permitiría vislumbrar si el los usuarios encuentran valor agregado al sistema.

### 7.4. Diferenciación de trazos de personas

La plataforma SmartCore no permite diferenciar de manera clara quiénes fueron los autores de los trazos o *sketches* dibujados en la pantalla. Si bien cada persona tiene asociado un color para los trazos y nodos que dibuje, los trazos de una misma página se dibujan de un solo color. En cambio, los nodos sí se dibujan con los colores de sus creadores.

Para que MCPresenter pudiera mostrar los trazos con los colores de sus autores, cada clase que representaba un trazo se encargaba de obtener el color respectivo antes de dibujarse en la pantalla. Esto último podría ser usado en SmartCore para el beneficio de todas las aplicaciones que ya se han escrito sobre esa plataforma.

## 7.5. Simplificar el diseño de los modos

En el desarrollo de la aplicación se constató que los modos concentran gran parte de las tareas: reconocimiento de gestos, dibujo de pantalla, acciones posteriores a la recepción de objetos por red, agregación de iconos contextuales a los objetos seleccionados, etc. Esto ocasiona que los modos, dada la gran cantidad de funciones que realizan, sean clases con muchas líneas de código, complejas y fuertemente acopladas a los objetos que usan, impidiendo su reutilización y extensibilidad.

## 7.6. Utilización de clases parametrizables en SmartCore

La introducción de clases parametrizables o genéricas en .NET Framework 2.0 permite la creación de clases más reusables y la prevención de errores ocasionados por el tipo de las variables. Existen objetos en SmartCore que podrían ser de su contraparte parametrizable (por ejemplo, usar `List<T>` en vez de `ArrayList`). Desde el punto de vista del desarrollador que ocupa SmartCore, sería más fácil detectar errores en tiempo de compilación asociado al uso de esos objetos.

Sin embargo, no todos los objetos que pueden pasarse a clases genéricas deberían hacerlo, puesto que el .NET Compact Framework no puede deserializar bien objetos de estas clases, lo que traería problemas al ejecutar aplicaciones de SmartCore en dispositivos móviles.

## 7.7. Aviso a modos de nueva sesión

Cuando en la aplicación se limpia la sesión actual, los modos no son informados de ese evento. Eso puede traer problemas porque los modos pueden tener como variables de instancia referencias a dibujables que estaban en la anterior sesión, ocasionando que la aplicación pase un estado inconsistente. Una solución sería introducir un mecanismo para avisar a los modos de tal evento, para que ellos puedan realizar los cambios necesarios para que la aplicación siga funcionando correctamente.

## 7.8. Mejorar la edición a mano alzada

La creación de los trazos a mano alzada no es tan fluida como en algunas aplicaciones similares que se utilizan en la sala de clases. Aquellas aplicaciones utilizan un ensamblado llamado Microsoft.Ink. Se debería investigar el funcionamiento de esa biblioteca, ver si ofrece características que la aplicación actual no tiene (por ejemplo, detección de la presión del lápiz sobre la pantalla para hacer trazos de grosor variable) y verificar si es viable acoplarlo a la plataforma SmartCore.

## **7.9. Inserción de objetos en las presentaciones**

Si bien la aplicación puede importar diapositivas desde PowerPoint para insertarlas al interior de una presentación, se podría agregar mayor variedad de contenido a las presentaciones. Por ejemplo, se podría insertar archivos de texto, documentos Word, enlaces a sitios web o contenido multimedial (como sonidos o animaciones gráficas) a las presentaciones, como una manera de complementar el contenido que el profesor va exponiendo a sus alumnos.

## **7.10. Edición de contenido de diapositivas PowerPoint**

Actualmente, las diapositivas PowerPoint que se importan a la aplicación se convierten a imágenes, lo que no posibilita la interacción con los elementos individuales que están presentes en las diapositivas, como los textos y los diagramas. Para ello primero habría que investigar cómo obtener los elementos individuales desde una diapositiva PowerPoint y crear en la aplicación dibujables que representen a cada uno de los posibles tipos de elementos presentes en aquellas diapositivas.

## 8. Bibliografía y referencias

- [1] G. Zurita, N. Baloian, F. Baytelman, "Mobile Collaborative Knowledge Management System," Computer Supported Cooperative Work in Design (CSCWD)'2008, pp. 588-593
- [2] Á. Arbonés, "Conocimiento para innovar," 2da edición, Ed. Díaz de Santos, 2006, pp. 131
- [3] E. Golub, "Supporting faculty goals during student presentations via electronic note-taking," Frontiers in Education, 2004. FIE 2004. 34th Annual, 20-23 Octubre 2004, vol. 2, pp. F4E-13-16
- [4] E. Tufte, "The Cognitive Style of PowerPoint". Graphics Press, Cheshire, Connecticut, 2003
- [5] G. Zurita, N. Baloian, F. Baytelman, "Supporting rich interaction in the classroom with mobile devices", Fifth IEEE International Conference on Wireless, Mobile, and Ubiquitous Technology in Education (*WMUTE*), pp. 115-122, 23-26, Marzo 2008
- [6] R.J. Anderson, R. Anderson, T. Vandegrift, S. Wolfman, K. Yasuhara, "Promoting interaction in large classes with computer-mediated feedback", CSCL 2003
- [7] M. Ratto, R. Shapiro, T. Truong, W. Griswold, "The activeclass project: Experiments in encouraging classroom participation". En CSCL 2003.
- [8] J. V. Wertsch, "Vygotsky and the Social Formation of Mind", Harvard University Press, 1988.
- [9] R.J. Anderson, R. Anderson, P. Davis, N. Linnell, C. Prince, V. Razmov, F. Videon, "Classroom Presenter: Enhancing Interactive Education with Digital Ink," Computer, vol. 40, no. 9, pp. 56-61, Septiembre, 2007.
- [10] D. Berque, "An evaluation of a broad deployment of DyKnow software to support note taking and interaction using pen-based computers," Journal of Computing Sciences in Colleges, vol. 21, no. 6, Junio 2006, pp. 204-216.
- [11] K. Koile, K. Chevalier, C. Low, S. Pal, A. Rogal, D. Singer, J. Sorensen, K. Seng Tay, K. Wu, "Supporting Pen-Based Classroom Interaction: New Findings and Functionality for Classroom Learning Partner," First International Workshop on Pen-Based Learning Technologies (PLT 2007), pp. 1-7, 2007.
- [12] C. Cobo, H. Pardo, "Planeta Web 2.0. Inteligencia colectiva o medios fast food", Grup de Recerca d'Interaccions Digitals, Universitat de Vic, Flacso México, Barcelona / México DF, 2007. <http://www.planetaweb2.net/>. Accedida el 4/Marzo/09
- [13] N. Baloian, J. Pino, H. Hoppe, "Dealing with students' attention problem in computer supported face-to-face lecturing", Educational Technology and Society, vol. 11, no. 2, 2008, pp. 192-205



- [14] G. Zurita, N. Baloian, F. Baytelman, "A collaborative face-to-face design support system based on sketching and gesturing," *Advanced Engineering Informatics*, vol. 22, no. 3, Julio 2008, pp. 340-349.
- [15] F. Baytelman, "Sistema de Soporte para Simulaciones Participativas mediante Dispositivos Móviles", memoria de título, Departamento de Ciencias de la Computación, Universidad de Chile, 2006.
- [16] J. Hernández, "Plataforma de comunicación sobre redes WiFi", memoria de título, Departamento de Ciencias de la Computación, Universidad de Chile, 2007.
- [17] Y. Otsuki, H. Bandoh, N. Kato, M. Nakagawa, "Development of an interactive electronic whiteboard system with multiple electronic pens for educational applications," *Proceedings of ICCE 2001*, vol. 3, pp. 1438-1445
- [18] E. Ambikairajah, J. Epps, M. Sheng, B. Celler, P. Chen, "Experiences with an electronic whiteboard teaching laboratory and Tablet PC-based lecture presentations," en *Proc. IEEE ICASSP*, vol. 5, 2005, pp. 565-568.
- [19] A. Vizcaino, J. Olivas, M. Prieto, "Modelos de Estudiante en entornos de Aprendizaje Colaborativo", Taller Internacional de Software Educativo, TISE, 1998, <http://www.c5.cl/ntic/docs/colaborativo/modelos.pdf>. Accedido el 4/Marzo/09
- [20] R. Dufresne, W. Gerace, W. Leonard, P. Mestre, L. Wenk, "Classtalk: A classroom communication system for active learning", *Journal of Computing in Higher Education*, 7, 1996, pp. 3-47
- [21] C. Collazos, L. Guerrero, A. Vergara, "Aprendizaje Colaborativo: un cambio en el rol del profesor", *Memorias III Congreso Educación Superior en Computación, Jornadas Chilenas de la Computación*, Punta Arenas, Chile, Noviembre, 2001. <http://www.dcc.uchile.cl/~luguerre/papers/CESC-01.pdf>. Accedido el 4/Marzo/09
- [22] A. Schmidt, M. Lauff, M. Beigl, "Handheld C.S.C.W.," Workshop on handheld, en: *Proceedings of the Computer Supported Collaborative Work, CSCW'98* Noviembre, Seattle, 1998.
- [23] C. Liu, L. Kao, "Handheld Devices with Large Shared Display Groupware: Tools to Facilitate Group Communication in One-to-One Collaborative Learning Activities," *IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE'05)*, pp. 128-135, 2005.
- [24] H. Edwards, B. Smith, G. Webb, "Lecturing: Case studies, experience and practice", London:Kogan, 2001, pp. 123-129.
- [25] S. Kopf, N. Schelee, W. Effelsberg, "The Interactive Lecture: Teaching and Learning Technologies for Large Classrooms", Department for Mathematics and Computer Science, University of Mannheim, Mannheim, GDR, 2005.

- [26] J. Heines, W. Liang, "Combining, Storing, and Sharing Digital Ink," Proceedings of SIGSE'07, Covington, Kentucky, EEUU. Marzo 7-10, 2007
- [27] B. Simon, R. Anderson, S. Wolfman, "Activating Computer Architecture with Classroom Presenter", 10th Workshop on Computer Architecture Education, Junio 2003.
- [28] UW Classroom Presenter, <http://classroompresenter.cs.washington.edu>. Accedido el 19/Nov/08.
- [29] M. Wilkerson, W. Griswold, B. Simon, "Ubiquitous Presenter: Increasing Student Access and Control in a Digital Lecturing Environment", SIGCSE '05: Proceedings of the 36th SIGCSE technical symposium on Computer science education, pp. 116-120
- [30] G. Singh, L. Denoue, A. Das, "Collaborative note taking using PDAs", Journal of Information Science and Engineering 21, 2005, pp. 835-848
- [31] M. Kam, J. Wang, A. Iles, E. Tse, J. Chiu, D. Glaser, O. Tarshish, J. Canny, "Livenotes: a system for cooperative and augmented note-taking in lectures". En *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA, Abril 02 - 07, 2005). CHI '05. ACM, Nueva York, NY, 531-540.
- [32] J. Sneller., "The Tablet PC classroom: Erasing borders, stimulating activity, enhancing communication," Frontiers in education conference - global engineering: knowledge without borders, opportunities without passports, 2007. FIE '07. 37th annual, pp. S3J-5-S3J-10, 10-13 Oct. 2007
- [33] S. Brewster, "Overcoming the Lack of Screen Space on Mobile Computers," Personal Ubiquitous Computing, vol. 6, no. 3, Enero 2002, pp. 188-205
- [34] J. Landay, B. Myers, "Sketching Interfaces: Toward More Human Interface Design". IEEE Computer, 2001, Vol 34, n°3, pp. 56-64
- [35] E. Lank, S. Phan, "Focus+Context sketching on a pocket PC." CHI '04 Extended Abstracts on Human Factors in Computing Systems, pp. 1275-1278
- [36] R. van der Lugt, "Functions of sketching in design idea generation meetings," Proceedings of the 4th Conference on Creativity & Cognition, 2002, pp. 72-79
- [37] L. Yang, J.A. Landay, Z. Guan, X. Ren, G.Dai, "Sketching informal presentations," Proceedings of the 5th international Conference on Multimodal interfaces, Vancouver, 2003, pp. 234-240
- [38] K. Forbus, J. Usher, V. Chapman, "Sketching for military courses of action diagrams," Proceedings of the 8th international Conference on intelligent User interfaces IUI '03, Miami, pp. 61-68
- [39] G. Zurita, P. Antunes, N. Baloian, L. Carriço, F. Baytelman, M. Sá, "Using PDAs in meetings: Patterns, Architecture and Components," Journal of Universal Computer Science, 2008. vol. 14, no. 1, Special Issue on Groupware: Issues and applications, pp. 123-146

- [40] T. Denning, W. G. Griswold, B. Simon, M. Wilkerson 2006. "Multimodal communication in the classroom: what does it mean for us?". SIGCSE Bull. vol. 138, no.1, Marzo 2006, pp. 219-223.
- [41] G. Zurita, N. Baloian, F. Baytelman, A. Farias, P. Antunes, "A Framework like a tool for specify motivating Collaborative Learning Participatory Simulations", Computer Supported Cooperative Work in Design *CSCWD'2007*, pp. 1016-1021.
- [42] G. Zurita, M. Nussbaum, M. Shaples, "Encouraging Face-to-Face Collaborative Learning through the Use of Handheld Computers in the Classroom," Mobile HCI 2003, Fifth International Symposium on Human Computer Interaction with Mobile Devices and Services, Udine, Italia.
- [43] C. Lee, A. Helal, N. Desai, V. Verma, B. Arslan, "Konark: A system and protocols for device independent, peer-to-peer discovery and delivery of mobile services," IEEE Transactions on Systems, Man and Cybernetics, Parte A, vol.33, no.6, pp. 682-696, Noviembre 2003
- [44] N. Segall, T. Doolen, J. David Porter, "A usability comparison of PDA-based quizzes and paper-and-pencil quizzes," Computers & Education, vol. 45, no. 4, Diciembre 2005, pp. 417-432.
- [45] N. Ward, H. Tatsukawa, "Software for taking notes in class," Frontiers in Education, FIE 2003. 33rd Annual , vol.3, pp. S2E-2-8
- [46] D. Holman, P. Stojadinovic, T. Karrer, J. Borchers, "Fly: An Organic Presentation Tool", CHI 2006, Abril 22-27, Canadá, pp.864-868.
- [47] MSDN: Referencia de C#, delegate. [http://msdn.microsoft.com/es-es/library/900fyy8e\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/900fyy8e(VS.80).aspx). Accedido el 27/Feb/09.