



UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO

FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS



LICENCIATURA EN SISTEMAS DE INFORMACIÓN

TRABAJO FINAL DE GRADUACIÓN

**DISEÑO DE UNA APLICACIÓN
MÓVIL PERSONALIZADA DE
APOYO AL APRENDIZAJE DE
REDES DE COMPUTADORAS**

Autor:

Matías Hernán Loto

Profesor Guía:

Dra. Elena B. Durán

Profesor Asesor:

Ing. Gregorio N. Tkachuk

Octubre de 2015

**TRABAJO FINAL DE GRADUACIÓN DE LA LICENCIATURA EN SISTEMAS DE
INFORMACIÓN**

**“DISEÑO DE UNA APLICACIÓN MÓVIL
PERSONALIZADA DE APOYO AL
APRENDIZAJE DE REDES DE
COMPUTADORAS”**

Autor:

.....

Matías Hernán Loto

Profesor Guía:

.....

Dra. Elena B. Durán

Asesor:

.....

Ing. Gregorio N. Tkachuk

* _____ *

Aprobado el día del mes de del año 20.....

por el Tribunal integrado por

.....

.....

∞ Santiago del Estero – Argentina ∞

DEDICATORIAS

A mis Padres Jaime y Maryleé.

A mis Hermanos Fernando, Ivana, Florencia, Antonella y Emanuel.

A mis Abuelas Amanda y Olga.

A mi Sobrinito Bauti, ese Hermoso Angelito en el Cielo que Iluminará

Eternamente Nuestros Corazones.

A su Mamá Anabella.

Matias Hernán Loto

AGRADECIMIENTOS

A la Dra. Elena B. Durán,
Por sus conocimientos, su apoyo, su predisposición, su paciencia, y por el
compromiso asumido para el desarrollo del presente trabajo.

Al Ing. Gregorio N. Tkachuk,
Por su disponibilidad y sugerencias brindadas para desarrollar el trabajo final.

A mis padres Jaime y Maryleé,
Por haberme apoyado incondicionalmente a lo largo de mi carrera
universitaria, por sus valores, sus consejos y sus ejemplos de fortaleza al
atravesar momentos difíciles en la vida.

A mis hermanos Fernando, Ivana, Florencia, Antonella y Emanuel,
Por la paciencia, el apoyo y la comprensión en todo momento.

A mi abuela Amanda,
Por su constante e invaluable apoyo.

A toda mi familia.

Matías Hernán Loto
Santiago del Estero, Argentina
Octubre 2015

INDICE DE CONTENIDOS

RESUMEN.....	XI
CAPÍTULO I.....	1
PLANTEAMIENTO DEL PROBLEMA.....	1
I.1. INTRODUCCIÓN	1
I.2. PLANTEAMIENTO DEL PROBLEMA.....	2
I.3. JUTIFICACIÓN DE LA RELEVANCIA DEL PROBLEMA.....	5
I.4. ANTECEDENTES	6
I.5. OBJETIVOS	11
I.5.1 Objetivos Generales.....	11
I.5.2 Objetivos Específicos	12
I.6. ESTRUCTURA DEL TRABAJO.....	12
CAPÍTULO II.....	13
MARCO CONCEPTUAL	13
II.1. COMPUTACION MOVIL	13
II.1.1 Definiciones	14
II.1.2 Atributos de la Computación Móvil.....	14
II.1.3 Desarrollo de Aplicaciones Móviles	15
II.1.4 Aprendizaje Móvil.....	16
II.2. ESTILOS DE APRENDIZAJE.....	18
II.2.1 Definiciones	19
II.2.2 Modelo de Felder y Silverman	20
II.2.3 Implicaciones Pedagógicas.....	22
II.3. SISTEMAS HIPERMEDIA ADAPTATIVOS.....	22
II.3.1 Sistemas Hipermedia Adaptativos.....	22
II.3.2 Componentes de un Sistema Hipermedia Adaptativo	23
II.3.3 Adaptación en los Sistemas Hipermedia Adaptativos.....	24
II.3.3.1 Métodos y Técnicas de Adaptación.....	24
II.3.3.1.1 Métodos de adaptación contenido	25
II.3.3.1.2 Técnicas de adaptación contenido.....	26
CAPÍTULO III.....	28
METODOLOGIA DE DESARROLLO.....	28
III.1. INTRODUCCIÓN	28
III.2. METODOLOGÍAS DE DESARROLLO ÁGIL	29
III.2.1 El Manifiesto Ágil.....	30
III.2.2 Por qué usar Metodologías Ágiles	31
III.2.3 Metodologías Ágiles en el Desarrollo de Software Móvil.....	32
III.3. SCRUM.....	34
III.3.1 Visión General	34

III.3.2 El Equipo Scrum (Scrum Team)	36
III.3.2.1 <i>El Dueño del Producto (Product Owner)</i>	37
III.3.2.2 <i>El Equipo de Desarrollo (Development Team)</i>	38
III.3.2.3 <i>El Scrum Master</i>	39
III.3.3 Eventos de Scrum.....	39
III.3.3.1 <i>El Sprint</i>	40
III.3.3.2 <i>Reunión de Planificación del Sprint (Sprint Planning Meeting)</i>	41
III.3.3.3 <i>Objetivo del Sprint (Sprint Goal)</i>	43
III.3.3.4 <i>Scrum Diario (Daily Scrum)</i>	43
III.3.3.5 <i>Revisión de Sprint (Sprint Review)</i>	44
III.3.3.6 <i>Retrospectiva de Sprint (Sprint Retrospective)</i>	45
III.3.4 Artefactos de Scrum.....	46
III.3.4.1 <i>Pila del Producto (Product Backlog)</i>	46
III.3.4.2 <i>Pila del Sprint (Sprint Backlog)</i>	48
III.3.4.3 <i>Incremento</i>	49
III.3.5 Transparencia de los Artefactos	49
III.3.6 Definición de Terminado (Definition of Done)	50
III.4 POR QUÉ ELEGIMOS SCRUM	51
<u>CAPÍTULO IV</u>	52
<u>DESARROLLO DE LA APLICACIÓN.....</u>	52
IV.1. INTRODUCCION	52
IV.2. APLICANDO SCRUM – ARQUITECTURA DE LA APLICACIÓN	52
IV.3. PILA DEL PRODUCTO (PRODUCT BACKLOG).....	54
IV.4. DEFINICION DE TERMINADO	56
IV.5. HERRAMIENTAS DE DESARROLLO	56
IV.6. SPRINT I – CONTROL DE ACCESO.....	58
IV.6.1 Reunión de Planificación del Sprint I	58
IV.6.2 Objetivo del Sprint I	59
IV.6.3 Pila del Sprint I	59
IV.6.4 Incremento	60
IV.6.5 Revisión y Retrospectiva del Sprint I	60
IV.7. SPRINT II – LECTURA DE CÓDIGOS QR.....	62
IV.7.1 Reunión de Planificación del Sprint II.....	62
IV.7.2 Objetivo del Sprint II	62
IV.7.3 Pila del Sprint II.....	63
IV.7.4 Incremento	63
IV.7.5 Revisión y Retrospectiva del Sprint II	63
IV.8. SPRINT III – ADMINISTRACIÓN GENERAL DEL SISTEMA	64
IV.8.1 Reunión de Planificación del Sprint III	64
IV.8.2 Objetivo del Sprint III.....	64

IV.8.3 Pila del Sprint III.....	65
IV.8.4 Incremento	66
IV.8.5 Revisión y Retrospectiva del Sprint III.....	66
IV.9. SPRINT IV – MÓDULO DE ADAPTACIÓN.....	67
IV.9.1 Reunión de Planificación del Sprint IV	67
IV.9.2 Objetivo del Sprint IV.....	68
IV.9.3 Pila del Sprint IV	68
IV.9.4 Incremento	69
IV.9.5 Revisión y Retrospectiva del Sprint IV	73
IV.10. SPRINT V – RELEASE.....	75
IV.10.1 Reunión de Planificación del Sprint V.....	75
IV.10.2 Objetivo del Sprint V	75
IV.10.3 Pila del Sprint V.....	76
IV.10.4 Incremento	76
IV.10.5 Revisión y Retrospectiva del Sprint V.....	76
<u>CAPÍTULO V.....</u>	<u>77</u>
<u>PRUEBAS Y ANALISIS DE RESULTADOS.....</u>	<u>77</u>
V.1. INTRODUCCION.....	77
V.2. PRUEBAS A LA APLICACIÓN WEB.....	77
V.2.1 Pruebas de Contenido.....	77
V.2.2 Pruebas de Interfaz.....	78
V.2.2.1 Prueba de Mecanismos de la Interfaz.....	78
V.2.2.2 Pruebas de Compatibilidad	81
V.2.3 Prueba de Navegación.....	82
V.2.4 Prueba de Rendimiento	82
V.3. PRUEBAS A LA APLICACIÓN MÓVIL	84
V.3.1 Estrategia de Testing en Android	84
V.3.2 Pruebas a la Aplicación Android.....	84
V.3.3 Pruebas del módulo de adaptación en la Aplicación.....	86
V.4. ANALISIS DE RESULTADOS	93
<u>CAPÍTULO VI.....</u>	<u>94</u>
<u>CONCLUSIONES.....</u>	<u>94</u>
<u>REFERENCIAS BIBLIOGRÁFICAS</u>	<u>96</u>
<u>ANEXO A</u>	<u>102</u>
<u>SCRUM – INCREMENTOS DEL PRODUCTO</u>	<u>102</u>
A.1. SPRINT I – CONTROL DE ACCESO	102
A.2. SPRINT II – LECTURA DE CÓDIGOS QR	104
A.3. SPRINT III – ADMINISTRACIÓN GENERAL DEL SISTEMA	106
A.4. SPRINT IV – MÓDULO DE ADAPTACIÓN.....	109
A.5. SPRINT V – RELEASE.....	113

RESUMEN

Las tecnologías móviles y las comunicaciones están afectando cada vez más la vida de los individuos y son un factor importante a tener en cuenta en todos los ámbitos, incluido el educativo. Estas tecnologías han transformado las metodologías de aprendizaje. Hoy el aprendizaje se produce en cualquier momento y en cualquier lugar, y se aprende durante toda la vida. En consecuencia han surgido nuevos modelos tales como el aprendizaje móvil, el aprendizaje ubicuo, el autoaprendizaje, etc. Es necesario apoyar estos paradigmas educativos con aplicaciones educativas personalizadas. En este trabajo se presenta el desarrollo de una aplicación móvil para Tablets y Smartphones con S.O. Android, que utiliza códigos QR para reconocer componentes en un laboratorio de redes de computadoras y presenta información personalizada acerca de los mismos, en base al estilo de aprendizaje y al nivel de conocimiento del estudiante. La aplicación ha sido desarrollada siguiendo la metodología de desarrollo ágil Scrum, caracterizada por el desarrollo iterativo e incremental. Esta aplicación busca ser una herramienta motivadora y útil que permita a los estudiantes mejorar sus procesos de aprendizaje y fortalecer su formación.

Palabras Clave: *Aprendizaje móvil, estilos de aprendizaje, personalización, códigos QR.*

CAPÍTULO I

PLANTEAMIENTO DEL PROBLEMA

I.1. INTRODUCCIÓN

El mundo se transforma a gran velocidad, pasamos de una economía industrial a otra basada en la información y orientada por los medios (Cope & Kalantzis, 2009). Un sistema educativo estático y monolítico no se adapta al mundo cambiante, complejo e innovador en el que actualmente vivimos y donde la tecnología cambia de forma dramática y constante. Es necesario adaptarse a este mundo más interconectado, más complejo y que se apoya en la innovación. Es preciso ajustar los métodos de enseñanza y apoyarlos con la tecnología digital de los tiempos que corren (García, 2009; Mockus et al., 2011).

La inclusión de la tecnología y particularmente de la tecnología móvil al proceso educativo es de vital importancia para crear un sistema educativo dinámico que se adapte a un mundo complejo de constante cambio y progreso (García, 2009; Nagella & Govindarajulu, 2008), además como plantea Vahey y Crawford (2002) el aprendizaje con tecnologías móviles puede ser personalizado, situado y auténtico, básicamente más centrado en el estudiante fomentando así la creatividad y la innovación. La inclusión de la tecnología móvil al área educativa, más que como una opción se presenta como una necesidad para apoyar las tareas de aprendizaje y a su vez motivar a los estudiantes; sin embargo, con el fin de obtener los mayores beneficios de la tecnología móvil, es fundamental el desarrollo de aplicaciones móviles adaptativas que permitan una mejor experiencia al usuario final mediante la presentación de contenido personalizado. Es aquí donde surgen dos conceptos claves de nuestro trabajo, por un lado los *Sistemas Hipermedia Adaptativos* que utilizan métodos y técnicas de adaptación para brindar información multimedia personalizada en base a una serie de factores almacenados en el modelo de usuario, como el nivel de conocimiento, preferencias, entre otros. Por otro lado, los *Estilos de Aprendizaje* representan las preferencias características en la forma en que un estudiante percibe y procesa la información Felder (1998). Basándonos en ellos podemos generar y ofrecer contenido personalizado para mejorar el proceso de aprendizaje de los estudiantes. La definición de Keefe (1988) recogida por Alonso, Gallego y Honey (1994) puntualiza que “*los estilos de aprendizaje son los rasgos cognitivos, efectivos y*

fisiológicos, que sirven como indicadores relativamente estables, de cómo los alumnos perciben, interaccionan y responden a sus ambientes de aprendizaje”.

Muchos investigadores en el área educativa consideran los estilos de aprendizaje como un factor importante en el proceso de aprendizaje y afirman que pueden mejorar el rendimiento académico de los estudiantes (Hong & Kinshuk, 2004). Felder, por ejemplo, sostiene que los estudiantes con una fuerte preferencia por un determinado estilo, pueden tener dificultades en el proceso de aprendizaje si el entorno educativo no se adapta a su estilo de aprendizaje particular (Felder & Silverman, 1988). Algunos ensayos demuestran que proporcionar información adaptada al Estilo de Aprendizaje del estudiante puede ser un factor importante en el resultado final del aprendizaje (Chen & Macredie, 2002; Bajraktarevic, Hall & Fullick, 2003). En base a estos trabajos se puede argumentar que incorporar los estilos de aprendizaje facilita en cierta medida el proceso de aprendizaje y así mismo aumenta la eficiencia de los estudiantes.

Este trabajo trata sobre el conocimiento del usuario, los estilos de aprendizaje, la adaptación y la tecnología móvil en la educación y tiene como objeto el desarrollo de una aplicación móvil para *Smartphones* y *Tablets* que asista en sus procesos de aprendizaje a los estudiantes que realizan un curso sobre Redes de Computadoras. La aplicación para la plataforma Android permitirá el reconocimiento de componentes de red en un laboratorio a través de la lectura de códigos QR y deberá presentar información adaptada al nivel de conocimiento y al estilo de aprendizaje del estudiante considerando el modelo de Felder-Silverman (1988). Se espera que asistir al estudiante mediante las tecnologías móviles y en forma personalizada permita mejorar sus procesos de aprendizaje y fortalecer su formación.

I.2. PLANTEAMIENTO DEL PROBLEMA

El rendimiento académico está relacionado con los procesos de aprendizaje. “Aprendizaje y rendimiento implican la transformación de un estado determinado en un estado nuevo, que se alcanza con la integración en una unidad diferente con elementos cognitivos y de estructuras no ligadas inicialmente entre sí” (De Natale, 1990). En todos los ámbitos educativos y sobre todo en las instituciones de Educación Superior se debe trabajar en el perfeccionamiento continuo de los procesos de aprendizaje para lograr preparar profesionales con una formación integral (Esquivel, González & Flores, 2013; Figueroa & Viglicca, 2006).

Los estudiantes perciben y adquieren conocimiento de manera distinta, tienen sus propias ideas, piensan y actúan de manera distinta. Además, tienen preferencias hacia determinadas estrategias cognitivas que permiten asimilar nuevo conocimiento. El término estilos de aprendizaje hace referencia a estas estrategias que son, de manera más específica, formas de recopilar, interpretar, organizar y pensar sobre la nueva información (Gentry & Helgesen, 1999).

Son varios los estudios que confirman la relación entre los Estilos de Aprendizaje y el rendimiento académico, como resultado de la respuesta de los alumnos a diferentes métodos de enseñanza. Varios investigadores encontraron evidencias de que al presentar la información mediante diferentes enfoques se consigue una instrucción más efectiva (Saarikoski, Salojärvi, Del Corso & Ovcin, 2001). Hong y Kinshuk (2004), Felder y Silverman (1988), Barragán (2008), Chen y Macredie (2002) y Bajraktarevic *et al.* (2003) entre otros, en sus respectivos trabajos consideran los estilos de aprendizaje como un factor importante en el proceso de aprendizaje y concuerdan que la incorporación de los mismos a la educación permite que el proceso de aprendizaje sea más eficiente y aumente el rendimiento académico de los estudiantes. Un concepto muy ligado a los estilos de aprendizaje es la **adaptación**, en los últimos años distintos investigadores han intentado integrar los estilos de aprendizaje en el diseño de sus aplicaciones adaptativas. Sin embargo, este no es un proceso sencillo.

Muchos Sistemas Hipermedia tradicionales presentan, en general, algunas limitaciones importantes. Éstas son básicamente tres: la desorientación que se produce en algunos estudiantes al recorrer sitios que contienen gran cantidad de información sobre los temas ofrecidos; la fatiga cognitiva que experimentan algunos usuarios al obtener contenidos en diferentes formatos de información y de manera simultánea y, por último, la utilización de una única estrategia instruccional, asumiendo que los estudiantes están dotados de las mismas capacidades, conocimientos, experiencias y estilos para procesar y percibir la información que se les proporciona (Prieto, Gros & García, 2003).

Una alternativa que disminuye el impacto de las limitaciones anteriormente descritas son los Sistemas Hipermedia Adaptativos (SHA), que esencialmente son sistemas hipermedia con capacidad de adaptar la presentación de contenidos, los formatos de información y las opciones de navegación a características de sus potenciales usuarios, esto con el fin de proporcionar la información más relevante y en los formatos más adecuados para cada uno de ellos. Inicialmente fueron modelados ciertos factores o características como el nivel de conocimientos, las experiencias previas, el idioma y los intereses para

producir la adaptación en los primeros SHA destinados a favorecer el aprendizaje, en un área específica del conocimiento (Prieto, Leighton & García, 2004). Posteriormente, estos tipos de sistemas han incorporado otras variables, los estilos cognitivos o los estilos de aprendizaje de los estudiantes. Esta nueva forma de basar la adaptación se fundamenta en que distintos estudiantes no logran un igual grado de desempeño frente a una única forma de enseñanza, que es la que esencialmente proporcionan los sistemas tradicionales, sino que requieren de estrategias instruccionales adecuadas a la forma como procesan y perciben la información (Prieto, Gros & García, 2004).

Algo muy considerado en los SHAs desde sus orígenes es la adaptación basada en el conocimiento de los usuarios, fue uno de los primeros factores en ser utilizados en el modelo de usuario y su importancia radica en que el conocimiento es fundamental al momento de presentar un determinado concepto, ya que permitirá incluir u ocultar parte de la información que se considere relevante o no para un determinado usuario en función de sus conocimientos. Los usuarios con escasos conocimientos sobre un determinado concepto necesitan explicaciones básicas innecesarias para los usuarios avanzados. Por el contrario éstos últimos pueden acceder a información más detallada sobre el mismo concepto, la cual podría resultar abrumadora para el caso de los usuarios que tienen conocimientos básicos.

Por otra parte, las tecnologías de la información han desarrollado los elementos de comunicación necesarios para romper los paradigmas de tiempo y espacio en los que se fundaban las relaciones humanas afectando todos sus ámbitos incluido el educativo (Ramos, Aguilar & Howlet, 2013).

Estudios relacionados al aprendizaje móvil (Chwabe & Göth, 2005; Vahey & Crawford, 2002) han demostrado que los estudiantes están más comprometidos en aprender, han incrementado la autoestima, la autodirección, toman la iniciativa y tienen mayor oportunidad de aprender a su propio ritmo. Las investigaciones han confirmado que el aprendizaje móvil es efectivo para aumentar la motivación de los participantes, ya que en última instancia, pone al alumno en el control de qué, cuándo y dónde aprenden (McCabe & Tedesco, 2012).

En consecuencia, puede resultar de interés el desarrollo de una aplicación móvil que ofrezca asistencia a los estudiantes a través de sus dispositivos móviles. Además, si se considera que cada estudiante aprende de una manera diferente, y tiene un nivel de conocimiento distinto al de otros compañeros, sería apropiado que las aplicaciones

informáticas que apoyan al aprendizaje potencialicen estas capacidades individuales de los estudiantes mediante características adaptativas.

En particular, la aplicación móvil que se propone está dirigida a estudiantes de carreras vinculadas a la Informática/Computación y Electrónica, que en sus currículas abordan el tema de Redes de Computadoras. Es común que estos estudiantes asistan a los Laboratorios de Redes para realizar sus prácticas. Puede ocurrir, en estos casos, que el docente no siempre se encuentre disponible para orientar al alumno y evacuar sus dudas. En este tipo de situaciones puede resultar de utilidad una aplicación informática que corra en el dispositivo móvil que utiliza el alumno y que lo asista en sus consultas para poder avanzar con la tarea.

I.3. JUTIFICACIÓN DE LA RELEVANCIA DEL PROBLEMA

El presente trabajo propone el desarrollo de una aplicación móvil que considera el nivel de conocimiento y el estilo de aprendizaje del estudiante para brindar información personalizada de manera transparente para el mismo. La aplicación será desarrollada para *Tablets* y *Smartphones* con S.O *Android*, que son dispositivos de uso común en la actualidad y un medio factible para realizar el reconocimiento de códigos QR si cuentan con una cámara.

El desarrollo de esta aplicación se justifica puesto que contribuirá al área educativa promoviendo el uso de dispositivos móviles y las tecnologías de la información y comunicación (TIC) en el ambiente universitario; permitirá crear en los alumnos una experiencia de recreación educativa a partir de la interacción con la aplicación y el entorno; generará alternativas de aprendizaje que apoyen al estudiante y usen como recurso dispositivos móviles que se encuentran al alcance de los mismos y finalmente sentará las bases para un desarrollo futuro de una aplicación ubicua, que permitirá a los estudiantes aprender independientemente del tiempo y el espacio en el que se encuentren.

En resumen, el desarrollo e implementación de esta aplicación intenta llenar un vacío tecnológico en nuestra facultad y a la vez traer un cambio positivo/favorable en la manera de aprender de los estudiantes de nuestra universidad, se intenta generar interés en los alumnos y a la vez apoyar a los profesores y a los mismos estudiantes en los procesos de enseñanza y de aprendizaje.

Si los resultados de implementar esta aplicación en la materia Redes de Computadoras son fructíferos, es posible a futuro, experimentar con aplicaciones similares

en otras asignaturas e inclusive en otras carreras con el fin de apoyar los procesos de enseñanza y de aprendizaje en el ámbito universitario.

I.4. ANTECEDENTES

La revisión bibliográfica se ha realizado poniendo el foco en tres aspectos: por un lado antecedentes de trabajos que demuestran la importancia de los sistemas educativos adaptativos y los diferentes factores de personalización que se pueden tener en cuenta para la adaptación; por otro, antecedentes de trabajos sobre aprendizaje móvil, seleccionando especialmente aquellos que utilizan tecnología de código QR; y finalmente trabajos de aprendizaje móvil que incluyan aspectos de adaptación.

Para el primer caso, podemos citar:

Yang, Hwang y Yang (2013) presentan el desarrollo de un sistema de aprendizaje adaptativo que incorpora un módulo de personalización y adapta la presentación de contenido basándose en los estilos cognitivos y los estilos de aprendizaje, además llevan a cabo un ensayo con el fin de evaluar el rendimiento del enfoque propuesto en un curso de ciencia del computador, el mismo muestra que el sistema mejora el rendimiento de aprendizaje de los alumnos.

Papanikolaou, Grigoriadou, Kornilakis y Magoulas (2003) presentan el sistema hipertexto INSPIRE donde implementan una técnica de adaptación de la presentación ampliamente utilizada llamada alteración de fragmentos que consiste en modificar un fragmento agregando o suprimiendo información, además adaptan las opciones de navegación. INSPIRE utiliza el Modelo de Honey y Mumford (1992) con los estilos de aprendizaje Activo, Reflexivo, Teórico y Pragmático para la adaptación de la presentación de contenidos; la adaptación de las opciones de navegación se determina en relación con el nivel de desempeño. Los mismos módulos de conocimiento son presentados a todos los estudiantes pero el orden y apariencia de los módulos difieren de un estudiante a otro. Así para estudiantes Activos (quienes son motivados por experimentos y tareas desafiantes) el módulo *Actividad* aparece al inicio de la página, seguido de los enlaces de ejemplos, teoría y ejercicios. En el caso de los Pragmáticos (quienes son motivados por probar con teorías y técnicas), el módulo *Ejercicio* se muestra en la parte superior de la página, seguido de enlaces a los ejemplos, teoría y actividades. De manera similar, en caso de los Reflexivos el orden de los módulos es: ejemplos, teoría, ejercicios y actividades, mientras que para los Teóricos el orden es: teoría, ejemplos, ejercicios y actividades. El sistema también ofrece la posibilidad de elegir el orden de estudio preferido.

Un tipo de adaptación distinta se observa en el trabajo de Cha, Kim, Lee y Yoon (2006) donde se personaliza la interfaz del sistema en base a las preferencias de usuario. Los criterios de adaptación se basan en el modelo de Felder-Silverman. La interfaz se personaliza de forma adaptativa: esta contiene tres pares de widgets (imagen/texto, audio/video, tablero de preguntas y repuestas/tablero de anuncios), cada par contiene un área primaria y secundaria de información. El espacio asignado en la pantalla para cada widget varía de acuerdo con el estilo de aprendizaje del estudiante. Por ejemplo para un estudiante Visual el widget de imagen se ubica en el área primaria de información, que es más grande que el widget de texto; los dos widgets son intercambiados en caso de un estudiante con estilo Verbal. De forma similar el widget de preguntas/respuestas y el de anuncios son intercambiados en caso de un estudiante Activo versus uno Reflexivo.

Un tipo de adaptación más compleja es implementada por Triantafillou, Pomportsis y Demetriadis (2003). Usan tanto técnicas de adaptación de la presentación como técnicas de adaptación de navegación para personalizar la información y la ruta de aprendizaje de los estudiantes dependientes de campo (DC) y los independientes de campo (IC). Específicamente este sistema adaptativo basado en los estilos cognitivos usa texto condicional y variantes de página para presentar la información en un estilo diferente: de lo específico a lo general en el caso de los alumnos IC (quienes tienen una preferencia analítica) y de lo general a lo específico para los alumnos DC (quienes tienen una preferencia global). El sistema ofrece además dos opciones de control: **control de programa** para estudiantes DC, por medio del cual el sistema guía al alumno a través del material de aprendizaje; **control de estudiantes** para los IC, por medio del cual los alumnos pueden elegir sus propias rutas de aprendizaje, a través de un menú. Puesto que los estudiantes DC se benefician más de instrucciones y retroalimentación, un marco adicional en la parte inferior de la página se utiliza para darles instrucciones y orientaciones explícitas. Este marco no se encuentra en el caso de los alumnos IC, que prefieren pocas instrucciones. Del mismo modo, en el caso de las pruebas de autoevaluación, la información proporcionada para alumnos IC es menos extensa que en el caso de los alumnos DC. Finalmente a los alumnos DC se les ofrece dos herramientas de navegación para ayudarlos a estructurar el material de aprendizaje y crear el panorama general: un mapa conceptual (una representación visual de los conceptos de dominio y las relaciones entre ellos) y un indicador de ruta gráfica (presentación del actual, el anterior y el siguiente tema). Además el sistema permite a los estudiantes modificar las opciones de adaptación,

haciendo sus propias elecciones entre control de programa/estudiante, retroalimentación mínima/máxima, etc.

Popescu, Badica y Moraret (2010) presentan un innovador sistema adaptativo llamado WELSA que utiliza un método de modelado implícito combinado con adaptación de navegación y técnicas adaptativas de anotaciones. Este sistema no está basado en un modelo de estilos de aprendizaje particular sino en un conjunto de características extraídas de varios modelos de estilos de aprendizaje. Cuando el usuario realiza un curso, sus acciones son registradas y analizadas por el sistema con el fin de crear un modelo preciso del alumno. Sobre la base de las preferencias de aprendizaje identificadas y las reglas de adaptación integradas, el sistema ofrece a los estudiantes cursos individualizados. WELSA también ofrece funcionalidades para los profesores, quienes pueden crear cursos por medio de la *herramienta de edición dedicada*; también pueden establecer ciertos parámetros del proceso de modelado, de manera que se adapte a las particularidades de su curso. Los autores además presentan los datos obtenidos de experimentar con el sistema donde se observa la precisión del método de modelado así como la eficiencia y eficacia de la adaptación en el proceso de aprendizaje.

Nguyen, Nguyen y Ho (2009) presentan ACGs (Adaptive Course Generation System) un sistema para cursos adaptativos, creado y diseñado en la Facultad de Tecnología de la Universidad Nacional de Vietnam, Hanoi. El mismo constituye un AES (Adaptive e-learning systems) que ofrece cursos web con navegación adaptativa. ACGs se implementa utilizando tecnología basada en agentes, a través del sistema Bee-gent (Kawamura, Tahara, Hasegawa, Ohsuga y Honiden, 2000). Los agentes son usados para el seguimiento de los estudiantes, la actualización del modelo de dominio, la preparación de pruebas y la evaluación de resultados. La adaptación se centra en los objetivos y en el nivel de conocimiento de los usuarios.

Un trabajo también relacionado con la adaptación al nivel de conocimiento ha sido elaborado por Chorfi y Jemmi (2004), con su sistema hipermedia adaptativo denominado PERSO basado en el reconocimiento y procesamiento del lenguaje natural. El sistema utiliza sofisticadas técnicas para entender la información y requerimientos ingresados por los estudiantes, y crea contenido de aprendizaje en base a esto. PERSO utiliza dos enfoques. Uno de ellos sirve para representar el espacio semántico y evaluar la similitud semántica entre las respuestas correctas y las del estudiante a través de LSA (Latent Semantic Analysis). El otro enfoque se basa en decisiones previas o problemas similares y genera una nueva solución usando CBR (case-based reasoning). Esto es usado por el

sistema para determinar que concepto mostrar al estudiante en base a su nivel de conocimiento. El conocimiento es representado a través de redes semánticas (grafos) donde los vértices representan partes del conocimiento y los arcos, relaciones entre estos conocimientos. El sistema se compone de cinco componentes: el plan de estudios, el modelo de usuario, el analizador semántico, el sistema CBR y el planificador. Debido a que PERSO realiza análisis del lenguaje natural y no es compatible con estándares y procesos, tiene un limitado potencial para la internalización.

Para el caso de aplicaciones educativas móviles que realizan reconocimiento de códigos QR se puede citar;

Rikala y Kankaanranta (2012) presentan un estudio que explora y analiza los procesos y métodos de enseñanza que incorporan códigos QR y dispositivos móviles en el aula. Proponen una clasificación de los modos en que se usan los códigos QR en el contexto de la educación y definen cinco categorías (actividades de rastreo, actividades de campo, tareas basadas en papel, contenido generado por estudiantes, instrucciones de trabajo) en las cuales basan sus experimentos y concluyen el trabajo con observaciones reflexivas sobre el uso de códigos QR en el aula.

Los autores McCabe, Tedesco (2012) y García (2009) presentan una línea de investigación similar a la anterior, donde se utilizan dispositivos móviles y códigos QR enlazados a videos en la web. McCabe y Tedesco trabajan con alumnos de séptimo grado para apoyar sus esfuerzos con las tareas de matemáticas en el hogar. Los autores complementan los antiguos cuadernos de trabajos con códigos QR que son enlazados a videos en *YouTube*, lo cual brinda a los alumnos y a sus padres soporte digital para resolver los problemas matemáticos. Se detectó que los videos refuerzan el aprendizaje, los dispositivos móviles motivan a los estudiantes, existe un aumento de la comunicación con sus maestros y además los dispositivos involucran a padres y hermanos en el proceso de aprendizaje. Por su parte García presenta un proyecto interdisciplinario de educación física e idioma alemán. En él se utilizan *tablets* y códigos QR además de una metodología basada en el aprendizaje por proyectos. En el trabajo los alumnos de 3° ESO (Educación Secundaria Obligatoria) preparan videos de calentamiento deportivo con instrucciones en alemán, dichos videos son subidos a *YouTube* o *Vimeo* y enlazados a códigos QR. Por otro lado los alumnos de 1° ESO utilizan *tablets* para hacer el reconocimiento de los códigos QR que los dirigen hacia los videos, esto permite a los alumnos de 1° aprender los movimientos de calentamiento básicos y a su vez practicar el idioma alemán.

Yusof, Goolamally, Latif y Fadzil (2012) presentan un proyecto de la Open University Malaysia que intenta incorporar el uso de los códigos QR como un recurso de educación complementario mediante la introducción de tres ejemplos específicos de actividades básicas orientadas por códigos QR en un curso de estadística elemental. Tres tipos de contenido diferentes se utilizan para mejorar la comprensión de los estudiantes: instrucción basada en texto, localizador uniforme de recursos de video conferencias (*iLectures*) y una web de actividades auto-prueba. La principal contribución de este proyecto es ayudar a los estudiantes a aprender, practicar y mejorar sus conocimientos sobre estadísticas.

Un proyecto completamente diferente presenta Power Robert (2012). *QR cache* es el proyecto que involucra el desarrollo de objetos de aprendizaje reusables móviles (*mobile reusable learning objects - RLOs*) que son accedidos a través del escaneo de códigos QR para aprender terminología en inglés de hardware de computadora. El estudio se basa en la teoría de la distancia transaccional (TDT) (Moore, 1989, 1991) y el modelo FRAME de Koole (2009) para proporcionar fundamentos teóricos, tanto para las decisiones de diseño educativo como para los RLO. Los primeros resultados muestran un mayor compromiso y la reducción de la distancia transaccional, también indican que los RLO muestran una fuerte convergencia de los tipos de actividades delimitadas por el modelo FRAME.

Considerando el tercer foco enunciado, se analizaron trabajos que señalan la importancia y el impacto de incluir la tecnología móvil en la educación (*m-learning*), considerando los estilos de aprendizaje para brindar contenido personalizado. Un ejemplo es el trabajo de Kinshuk (2004) que investiga cómo mejorar el proceso de aprendizaje mediante la adaptación del contenido de un curso a los estilos de aprendizaje del estudiante en un ambiente multiplataforma (para PC y PDA). El trabajo está basado en la teoría de estilo de aprendizaje de Felder-Silverman (1988). Se desarrolla un *framework* para modelar los estilos de aprendizaje de los estudiantes y presentar el tema adecuado y adaptado a cada estudiante. Este *framework* utiliza una arquitectura tradicional basada en web con dos componentes adicionales, el módulo de análisis de estilo de aprendizaje y módulo de análisis de dispositivo de acceso. El primero se encarga de modelar el estilo de aprendizaje del estudiante y se comunica con el modelo de estudiante, mientras que el segundo módulo identifica el perfil del dispositivo de acceso y proporciona esta información al módulo tutorial. Finalmente el módulo tutorial crea el contenido adaptado en base al modelo de estudiante y el perfil del dispositivo de acceso y se lo presenta al estudiante.

Un trabajo de este estilo es desarrollado por Mota Jorge (2008) que describe el diseño, desarrollo e implementación de un reproductor de curso adaptable que utiliza los estilos de aprendizaje de Kolb y redes neuronales para modelar a los alumnos. La aplicación genera dinámicamente rutas de navegación y adaptación de diseño. El sistema implementa la adaptación de las recomendaciones individuales y adaptación de contenido basándose en los estilos de aprendizaje, en los conocimientos previos del alumno, en el progreso del mismo y en la persistencia de sus propias preferencias. El autor describe la implementación de los métodos de adaptación para la secuenciación de contenidos y presentación adaptativa basada en preferencias de estilos de aprendizaje y un motor de IA (Inteligencia Artificial) utilizando una red neuronal que procesa las predicciones del mejor diseño de presentación del siguiente objeto de aprendizaje en una secuencia de navegación.

La revisión bibliográfica realizada para conocer el actual estado del arte refleja por un lado la importancia de los sistemas hipermedia adaptativos que utilizan diferentes factores o características del usuario como estilos de aprendizaje, preferencias, progreso del mismo y demás para realizar la adaptación de contenido o de las opciones de navegación. Por otro lado podemos observar el uso de aplicaciones móviles que utilizan códigos QR en ambientes educativos, y finalmente trabajos que revelan el valor de las aplicaciones *m-learning* que realizan diferentes tipos de adaptación en base a ciertos modelos de estilos de aprendizaje y otros factores. En contraste con los trabajos antes mencionados se puede decir que el trabajo que aquí se propone combina o integra aprendizaje móvil con tecnología de códigos QR y sistemas hipermedia adaptativos considerando estilos de aprendizaje y nivel de conocimiento de los estudiantes. Además no se han encontrado antecedentes de aplicaciones móviles que utilicen códigos QR para el reconocimiento de componentes de red en ambientes educativos, en consecuencia se espera que el trabajo sea un importante aporte para esta área educativa.

I.5. OBJETIVOS

A continuación se presentan los objetivos generales y específicos que se buscan alcanzar en el presente trabajo.

I.5.1 Objetivos Generales

Favorecer los procesos de aprendizaje en el nivel universitario, aportando como recurso educativo una aplicación informática personalizada, basada en tecnología móvil, para asistir a estudiantes en un Laboratorio de Redes de Computadoras.

I.5.2 Objetivos Específicos

- ✓ Diseñar y desarrollar la aplicación cliente para el dispositivo móvil.
- ✓ Diseñar y desarrollar la aplicación web de administración.
- ✓ Modelar el perfil de usuario y desarrollar el módulo de adaptación para la presentación de información.

I.6. ESTRUCTURA DEL TRABAJO

El trabajo se estructura como se detalla a continuación. En el Capítulo II se presenta el Marco Conceptual con definiciones claves vinculadas al trabajo. En el Capítulo III se describe la metodología de desarrollo seleccionada para el proyecto en cuestión. En el Capítulo IV se presenta el diseño y desarrollo de la aplicación. En el Capítulo V se exponen las pruebas realizadas y el análisis de resultados. Finalmente en el Capítulo VI se presentan las conclusiones del trabajo.

CAPÍTULO II

MARCO CONCEPTUAL

II.1. COMPUTACION MOVIL

En un ambiente de computación tradicional el usuario se sienta frente a su computadora que está conectada a otras máquinas, a redes y a servidores a través de cables. La necesidad de estar conectados de esta manera, limita el uso de la computadora y se hace difícil o imposible para las personas que se encuentran en movimiento. En particular, vendedores, reparadores, agentes de la ley, entre otros muchos otros, podrían ser más eficaces si utilizan la tecnología de la información mientras están en sus puestos de trabajo en el campo o en tránsito. Por lo tanto la computación móvil fue diseñada en principio para los trabajadores que viajaban fuera de los límites de sus organizaciones o para las personas que estaban fuera de sus hogares.

La primera solución a la necesidad de la *computación móvil* fue diseñar las computadoras lo suficientemente pequeñas como para que puedan ser fácilmente transportadas. En primer lugar, se inventó el ordenador portátil; más tarde surgieron equipos cada vez más pequeños, tales como PDAs (asistentes digitales personales) y otros dispositivos de mano, los cuales fueron llamados *dispositivo móviles*. Con el tiempo estos dispositivos se volvieron cada vez más livianos y potentes en cuanto a velocidad de procesamiento y almacenamiento. Al final de la jornada laboral, los trabajadores móviles podían descargar (o cargar) información desde o hacia una computadora de escritorio normal en un proceso conocido como *sincronización*.

La segunda solución a la necesidad de la computación móvil consistía en sustituir los cables por *medios de comunicación inalámbrica*. Los sistemas inalámbricos han estado en uso en radio, TV y teléfono por un largo tiempo. Así que era natural adoptarlos en el ambiente de la computación.

La tercera solución fue la combinación de las dos primeras, utilizar los dispositivos móviles en un entorno inalámbrico. Esto se conoce como *computación móvil inalámbrica*, esta combinación permite una conexión en tiempo real entre un dispositivo móvil y otros entornos de computación, tales como Internet o una intranet. Esta innovación revolucionó la manera en cómo la gente usa las computadoras y se ha extendido a todas las áreas entre ellas la educación, la salud, el entretenimiento y mucho más. Este nuevo modelo de

computación está básicamente, dando lugar a la ubicuidad, con esto nos referimos a la computación en cualquier momento y en cualquier lugar. (Turban, Rainer & Potter, 2005)

II.1.1 Definiciones

A continuación se presentan dos definiciones de computación móvil:

La Computación Móvil se define como el entorno de computación sobre la movilidad física. En este entorno, los usuarios pueden acceder a datos, información o cualquier otro objeto lógico desde cualquier dispositivo mediante cualquier red mientras se están moviendo (Talukder, Ahmed & Yavagal, 2010).

También ha sido definida como “Un servicio que se mueve con las personas, el cual permite que las mismas se puedan inscribir, recordar, comunicar y razonar independientemente de la posición de los dispositivos.” (Lyytinen & Yoo, 2002).

II.1.2 Atributos de la Computación Móvil

La computación móvil tiene dos características principales que la diferencian de otras formas de computación: la *Movilidad* y el *Amplio Alcance*.

Movilidad implica portabilidad. La computación móvil se basa en el hecho de que los usuarios llevan un dispositivo móvil con ellos y pueden ponerse en contacto en tiempo real con otros sistemas en donde quiera que se encuentren, siempre que puedan conectarse a una red inalámbrica.

Amplio alcance se refiere al hecho de que las personas pueden ser contactadas en cualquier momento. Por supuesto, los usuarios pueden bloquear ciertas horas o ciertos mensajes, pero cuando los usuarios llevan un dispositivo móvil, pueden ser contactados al instante.

Estas dos características, la movilidad y el amplio alcance, crean cinco atributos de valor añadido que rompen las barreras de la geografía y el tiempo: *ubicuidad*, *comodidad*, *conectividad instantánea*, *personalización* y *localización de productos y servicios*.

Un dispositivo móvil puede ajustarse a la necesidad de información y comunicación en tiempo real, independientemente de la ubicación del usuario (*ubicuidad*). Con un dispositivo móvil con conexión a Internet, es fácil y rápido acceder a la Web, intranets y otros dispositivos móviles sin tener que arrancar un PC o realizar una llamada a través de un módem (*conveniencia* y *conectividad instantánea*). La información puede ser personalizada y enviada a consumidores individuales (*personalización*). Sabiendo donde se

encuentra físicamente un usuario en cualquier momento es posible ofrecer productos y servicios pertinentes (*localización*). (Turban *et al.*, 2005).

II.1.3 Desarrollo de Aplicaciones Móviles

La proliferación de dispositivos móviles propició el surgimiento de un nuevo tipo de aplicaciones, que pudieran suplir las necesidades de los usuarios en movimiento. La tecnología móvil ha mostrado un crecimiento extraordinario y exponencial que no para de crecer, por tal motivo es primordial el desarrollo de aplicaciones móviles que brinden soluciones informáticas en los diferentes entornos por los que se mueve el usuario, para ello se han retomado los modelos arquitectónicos de software existentes, los cuales se han adaptado junto a las metodologías de desarrollo tradicionales aunque las más usadas son las metodologías de desarrollo ágil.

A continuación se presentan tres variantes de aplicaciones móviles que difieren en función del enfoque de desarrollo aplicado: un enfoque nativo y dos enfoques multiplataforma (web e híbrido). (Delia, Galdamez, Thomas & Pesado, 2013).

Aplicaciones Web: son diseñadas para ser ejecutadas en el navegador del dispositivo móvil. La principal ventaja de este tipo de aplicación es su independencia de la plataforma. Solo es necesario un navegador y conexión a Internet. Por contrapartida, esto disminuye la velocidad de ejecución y podrían llegar a ser menos atractivas que las aplicaciones nativas. Además, podrían tener baja performance por problemas de conectividad. Finalmente este tipo de aplicaciones no pueden utilizar todos los recursos de hardware del dispositivo, como por ejemplo, cámara, GPS, entre otros.

Aplicaciones Nativas: son aquellas que se conciben para ejecutarse en una plataforma específica, es decir, se debe considerar el tipo de dispositivo, el sistema operativo a utilizar y su versión. Cuando la aplicación está lista para ser distribuida debe ser transferida a las *App stores* específicas de cada sistema operativo. La principal ventaja de este tipo de aplicaciones es la posibilidad de interactuar con todas las capacidades del dispositivo (cámara, GPS, acelerómetro, agenda, entre otras). Además no es estrictamente necesario poseer acceso a Internet. Su ejecución es rápida, puede ejecutarse en modo background y notificar al usuario cuando ocurra un evento que necesite su atención.

Estas ventajas se pagan con un mayor costo de desarrollo, pues se debe utilizar un lenguaje de programación diferente según la plataforma.

Aplicaciones Híbridas: combinan lo mejor de los dos tipos de aplicaciones anteriores. Se utilizan tecnologías multiplataforma como HTML, Javascript y CSS, pero se

puede acceder a buena parte de las capacidades específicas de los dispositivos. En resumen, son desarrolladas utilizando tecnología web y son ejecutadas dentro de un contenedor web sobre el dispositivo móvil. Entre las principales ventajas se pueden mencionar la distribución de la aplicación a través de la tienda de aplicaciones, la reutilización de código para múltiples plataformas y la posibilidad de utilizar las características de hardware del dispositivo. Una de las desventajas es que, al utilizar la misma interfaz para todas las plataformas, la apariencia de la aplicación no será como la de una aplicación nativa. Finalmente la ejecución será mucho más lenta que la ejecución en una aplicación nativa.

En el presente trabajo utilizamos el enfoque nativo ya que se adapta perfectamente a nuestros intereses, por un lado brindar una experiencia de usuario única y personalizada con una aplicación elegante, rápida y de alto rendimiento que pueda ser ejecutada en Tablets y Smartphones con S.O. Android, por otro lado tener acceso nativo al hardware del dispositivo móvil, específicamente a la cámara para realizar las lecturas de Códigos QR.

Es importante destacar que nuestra aplicación además de ser una aplicación móvil nativa (ya que estará programada con Java para Android, no será multiplataforma como las híbridas) tendrá una arquitectura Cliente-Servidor que resulta necesaria para administrar y centralizar toda la información asociada al modelo de dominio y al modelo de usuario. Asimismo en el servidor estará implementado el módulo de adaptación encargado de generar la información personalizada para ser desplegada en el dispositivo móvil del usuario.

II.1.4 Aprendizaje Móvil

La tecnología móvil ha traído consigo un gran potencial para ser usado en el proceso de enseñanza y aprendizaje. La convergencia digital de la tecnología móvil y el aprendizaje electrónico (*e-learning*) han dado origen al aprendizaje móvil (*m-learning*) en respuesta a las necesidades de aprendizaje de una sociedad cada vez más dinámica, en donde el tiempo disponible para la adquisición o repaso de nuevos o existentes conocimientos en adaptación a un entorno continuamente cambiante parece reducirse. El aprendizaje móvil se caracteriza de esta forma por su capacidad de entregar contenidos de aprendizaje sin fronteras de tiempo ni espacio a través de dispositivos móviles, tales como teléfonos celulares, agendas electrónicas, pequeñas computadoras y todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica con la finalidad de maximizar los tiempos disponibles para el aprendizaje. Es así que el aprendizaje móvil representa una

nueva posibilidad para acceder a diversos recursos de aprendizaje desde cualquier lugar y en cualquier momento, dando la oportunidad al estudiante de aprender en el instante, escenario y contexto más adecuados en relación a su objetivo y estilo de aprendizaje. (Molina & Romero, 2010).

Entre las ventajas que nos ofrece el aprendizaje móvil se destacan las siguientes (Jair, 2009):

- *Uso eficiente del tiempo*: se habla de la utilización de los tiempos muertos, que son aquellos que se destinan en labores de transporte, espera para ser atendido, para ser empleados con fines de aprendizaje.
- *Expansión de la alfabetización digital*: muchos de los usuarios de la tecnología móvil utilizan esta herramienta como medio de entretenimiento y comunicación social, y aunque su uso como forma de aprendizaje no está muy difundido se considera de enorme potencial.
- *Accesibilidad*: los dispositivos están al alcance de casi cualquier persona, así como los servicios necesarios para su uso.
- *Contacto social*: el estudiante puede estar en contacto con compañeros y tutores en cualquier momento para recibir información y facilitar su aprendizaje.
- *Productividad*: al hacer uso efectivo de tiempos muertos y el poder recibir retroalimentación a pedido, permite que los procesos de enseñanza y aprendizaje se hagan más rápido y de modo efectivo, aumentando la productividad del estudiante.
- *Aprendizaje colaborativo*: el estar en contacto con los compañeros de curso redundará en beneficio al momento de realizar trabajos que involucren colaboración, ya que la misma se refuerza con la presencia digital de los integrantes permitiendo aportes y retroalimentación.
- *Incremento en el estudio individual*: los aparatos usados en este tipo de tecnología son personales y por lo tanto su uso implica constancia, responsabilidad y motivación por parte del estudiante para cumplir con sus deberes y trabajos.
- *Información eficaz*: los contenidos transmitidos a los estudiantes por este medio deben tener características especiales y una de ellas es su tamaño (Lee, 2005), nadie va a pasar horas leyendo en un celular o PDA. La información enviada debe ser sintética y completa.

- *Recursos de aprendizaje móvil*: los profesores pueden diseñar y poner a disposición de los alumnos materiales que contribuyan al aprendizaje de sus alumnos, pudiendo estos, disponer de dichos materiales de manera asíncrona (*m-Learning Resources*).

En la Tabla 2.1, Burgos y Lozano (2010), identifican los servicios, beneficios y características que ofrece la tecnología móvil en los diferentes enfoques de aprendizaje según (Sharples, Taylor & Vavoula, 2005).

Enfoques de Aprendizaje	Apoyo de la Tecnología Móvil
Aprendizaje centrado en la persona	Servicios dirigidos y personalizados
Aprendizaje colaborativo	Conectividad inalámbrica a servicios de red
Aprendizaje Situacional	Movilidad de recursos, materiales y contenidos educativos.
Aprendizaje Contextual	Conciencia contextual
Aprendizaje Ubicuo	Ubicuidad (Pervasive)
Aprendizaje Permanente (Long Life Learning)	Reusabilidad, durabilidad, y permanencia de recursos y objetos de aprendizaje.

Tabla 2.1: Ventajas del Aprendizaje Móvil. (Burgos & Lozano, 2010)

El Aprendizaje Móvil involucra un modelo tecnológico cuya única variante es la de reducir aún más las pocas limitaciones tempo-espaciales que puedan sufrir los sistemas de enseñanza y aprendizaje a través de internet, accesible desde ordenadores convencionales. Así, desde una perspectiva pedagógica el aprendizaje móvil apunta a una nueva dimensión en los procesos de educación, al poder atender necesidades urgentes de aprendizaje, ubicarse en escenarios móviles y posibilitar gran interactividad en estos procesos. (Aretio, 2004).

II.2. ESTILOS DE APRENDIZAJE

Si miramos a nuestro alrededor podremos observar que las personas ven las cosas de diferente manera en la que nosotros lo hacemos. La gente tiene formas muy distintas de ver la misma situación y por lo tanto, su respuesta puede ser diferente.

Si nos centramos en el campo educativo, podemos ver que los estudiantes varían enormemente en la velocidad y manera en la que asimilan nueva información y en aplicar este conocimiento a nuevas situaciones. Los estudiantes aprenden de formas muy diferentes. Por ejemplo, si observamos la forma en la que la gente aprende segundas lenguas, algunos prefieren escuchar la lengua, otros leer un libro, otros prefieren

aprenderla poco a poco en lecciones diferenciadas, otros se encuentran más cómodos a través del aprendizaje en grandes bloques, y muchos prefieren utilizarla en entornos reales. Otro ejemplo, es la forma en que los estudiantes aprenden un nuevo lenguaje de programación. Algunos estudiantes empiezan escribiendo código inmediatamente después de aprender lo más básico del lenguaje, mientras que otros necesitan leer un manual y mirar un buen número de ejemplos antes de escribir alguna línea.

Probablemente existen tantas formas de enseñar como las hay de aprender. Algunos profesores solo dan lecciones magistrales, otros involucran a los estudiantes en discusiones, otros se centran en los axiomas, otros proporcionan más ejemplos, algunos priman el que los estudiantes memoricen información, y otros la comprensión de los conceptos (Felder & Henriques, 1995).

Podemos ver estas diferencias individuales en muchos aspectos de la vida cotidiana. Los estilos de aprendizaje tratan de dar respuesta a cuales son los mecanismos que subyacen en estas diferencias individuales.

II.2.1 Definiciones

A continuación se presentan algunas de las definiciones de Estilos de Aprendizaje más significativas:

- Descripción de las actitudes y comportamiento que determina las preferencias individuales en la forma de aprender (Honey & Mumford, 1992).
- Compuesto de características cognitivas, afectivas y psicológicas que sirven como indicadores relativamente estables de como un estudiante percibe, interacciona y responde al entorno de aprendizaje (Keefe, 1979).
- Preferencias características en la forma en que un estudiante percibe y procesa la información (Felder, 1996).

En este trabajo se utiliza el termino Estilos de Aprendizaje tal y como lo define Felder (1996), asumiendo que los estilos de aprendizaje son un tipo especial de preferencias, ya que el estudiante muchas veces no es consciente de ellas y son relativamente estables en el tiempo y en distintas situaciones (Brusilovsky, 2001).

II.2.2 Modelo de Felder y Silverman

El modelo de estilos de aprendizaje de Felder y Silverman (FSLSM) publicado en 1988 (Felder & Silverman, 1988) distingue cinco dimensiones para los estilos de aprendizaje. Estas dimensiones son independientes unas de otras y muestran como los estudiantes prefieren organizar (inductivo/deductivo), procesar (activo/reflexivo), percibir (sensorial/intuitivo), recibir (visual/verbal), y entender (secuencial/global) nueva información.

Felder y Silverman describen los estilos de aprendizaje usando escalas que van de 11 a -11 para cada dimensión. Los autores justifican la eliminación de la dimensión inductivo/deductivo desde un punto de vista estrictamente pedagógico, ya que no desean proporcionar a profesores y alumnos una herramienta que justifique unos hábitos que no consideran positivos. En concreto, no desean que esa dimensión sirva para potenciar la enseñanza y el aprendizaje deductivos, claramente preferidos tanto por profesores como por alumnos: la mayoría de los alumnos conscientemente prefiere una enseñanza deductiva, en el sentido de que sólo tienen interés en aprender aquello de lo que van a ser examinados y también para la mayoría de los profesores este estilo de enseñanza es más sencillo que su opuesto. Por lo tanto, el estilo de aprendizaje de cada estudiante está representado por cuatro valores entre 11 y -11, uno por cada dimensión. Felder y Silverman consideran las preferencias como tendencias, esto es, que incluso un estudiante con una fuerte preferencia por un estilo de aprendizaje particular puede actuar algunas veces de forma diferente.

La dimensión activo/reflexivo es una de las cuatro dimensiones de este modelo. Los estudiantes activos aprenden mejor trabajando activamente con el material de aprendizaje, aplicándolo y probando cosas. Tienden a estar más interesados en la comunicación con los demás y prefieren aprender trabajando en grupos donde puedan discutir acerca del material aprendido. Por el contrario, los estudiantes reflexivos prefieren pensar acerca del material de forma individual, trabajando solos.

La dimensión sensorial/intuitivo es otra dimensión del modelo. A los estudiantes con un estilo de aprendizaje sensorial les gusta aprender hechos y material concreto, usando sus experiencias sensoriales de hechos particulares como la primordial fuente de información. Les gusta resolver problemas con aproximaciones estándar y tienden a ser cuidadosos con los detalles. Son considerados más realistas, sensatos y prácticos, y les gusta relacionar el material aprendido con el mundo real. Como polos opuestos están los

estudiantes intuitivos, éstos prefieren aprender materiales abstractos como teorías y sus significados subyacentes, con principios generales en vez de hechos concretos, siendo estos principios la principal fuente de información. Les gusta descubrir posibilidades y relaciones y tienden a ser más innovadores y creativos. Como resultado, obtienen mejores puntuaciones en los exámenes con preguntas abiertas que en los exámenes tipo test. Esta dimensión difiere de la dimensión activo/reflexivo en un aspecto importante: la dimensión sensorial/intuitivo se refiere a la fuente de información preferida mientras que la dimensión activo/reflexivo tiene que ver con el proceso de transformación en conocimiento de la información percibida.

La dimensión visual/verbal hace referencia a la preferencia en cuanto al formato de los datos. Diferencia entre estudiantes que recuerdan mejor lo que han visto (fotos, dibujos, diagramas, graficas, etc.) de los que prefieren representaciones textuales, independientemente de que sean escritas u orales.

En la cuarta dimensión, los estudiantes se diferencian en la forma en la que entienden ese material, secuencial o global. Los estudiantes secuenciales aprenden en pequeños pasos incrementales y, por lo tanto, tienen un progreso lineal en su aprendizaje. Tienden a seguir soluciones lógicas y estructuradas para encontrar soluciones. En oposición, los estudiantes globales usan un proceso de pensamiento holístico y aprenden en grandes saltos. Tienden a absorber el material de aprendizaje de una forma casi aleatoria, sin ver las conexiones que existen, para obtener de repente una visión general. Son capaces de resolver problemas complejos y poner cosas juntas de formas novedosas; sin embargo, tienen dificultades en explicar como lo han hecho. Ya que la visión general es tan importante para ellos, tienden a estar interesados en los resúmenes y en el conocimiento extenso, mientras que los estudiantes secuenciales están más interesados en los detalles.

El Inventory of Learning Styles (Soloman, 1992), es la primera versión de un cuestionario para identificar los estilos de aprendizaje del modelo de Felder. El estilo de aprendizaje del estudiante se calculaba a través de las respuestas a 28 preguntas. Posteriormente Felder y Soloman desarrollaron el cuestionario Index of Learning Styles (ILS) (Felder & Soloman, 2004), un cuestionario con 44 preguntas. La preferencia en cada dimensión se expresa como el resultado de las respuestas (“a” o “b”) a las 11 cuestiones. El resultado es la resta de las respuestas “b” a las respuestas “a”.

II.2.3 Implicaciones Pedagógicas

Los estilos de aprendizaje pueden tener importancia en diversos aspectos de la educación. En primer lugar, para hacer conscientes a los estudiantes de su estilo de aprendizaje y enseñarles cuáles son sus puntos fuertes y cuáles los más débiles. El conocimiento de su estilo de aprendizaje les puede ayudar a entender por qué el aprendizaje es a veces difícil para ellos. En segundo lugar, concientizar a los profesores que proporcionar a los estudiantes material y actividades que satisfagan sus preferencias puede mejorar el proceso de enseñanza y aprendizaje. En la enseñanza presencial es una tarea complicada pretender que el profesor adapte su estilo de enseñanza a los estilos de aprendizaje de muchos estudiantes, para ello, lo mejor sería proporcionar material y actividades lo suficientemente variados como para que todos los estudiantes se encuentren satisfechos con, al menos, algunos de ellos. Sin embargo, la enseñanza a través de Internet nos permite adaptar los contenidos y la estructura del curso de una forma individualizada (Barragán, 2008).

II.3. SISTEMAS HIPERMEDIA ADAPTATIVOS

Ted Nelson fue uno de los pioneros del hipertexto y lo definió como la combinación de texto en lenguaje natural con las aptitudes de los ordenadores para ramificarse de forma interactiva (Nelson, 1965). En otras palabras, el hipertexto se puede entender como texto no secuencial, que está conectado por vínculos. Hipermedia extiende el concepto de hipertexto a través de elementos multimedia como gráficos, audio, y video en vez de solo representaciones textuales.

II.3.1 Sistemas Hipermedia Adaptativos

El desarrollo de Sistemas Hipermedia Adaptativos (SHA) empezó en la década de los 90 como intersección de dos áreas, el Hipertexto y el Modelado de Usuario. El primer resumen de sistemas hipermedia adaptativos, métodos y técnicas usadas por estos sistemas fue el realizado por Peter Brusilovsky (Brusilovsky, 1996).

El año 1996 marcó un antes y un después en la investigación de hipermedia adaptativa (Brusilovsky, 2001). Antes de esta fecha la investigación en el área se había reducido a unos pocos grupos aislados. Después de 1996 este campo vivió una gran expansión con nuevos sistemas hipermedia adaptativos, técnicas, introducción de

tecnologías Web y la primera conferencia oficial en 2000. Brusilovsky hizo una nueva versión de su resumen de sistemas hipermedia adaptativos en 2001 (Brusilovsky, 2001).

El objetivo de los Sistemas Hipermedia Adaptativos es proporcionar contenidos hipermedia que encajen con las necesidades individuales de los usuarios. Los SHAs pueden ser definidos como sistemas hipermedia que reflejan algunos rasgos del usuario en el modelo de usuario y aplican este modelo para adaptar varios aspectos visibles del sistema al usuario. Dicho de otra forma, el sistema debe poseer tres características (Brusilovsky, 1996):

- ✓ debe ser un sistema hipermedia.
- ✓ debe disponer de un modelo de usuario.
- ✓ debe ser capaz de adaptarse usando este modelo.

El proceso de adaptación suele ser cíclico: primero, se construye un modelo de usuario, después este modelo se usa para generar recursos adaptativos, más tarde se actualiza el modelo de usuario a través de la información proporcionada por el usuario o por las interacciones del usuario con el sistema, y esto hace que el sistema se adapte dinámicamente a ese modelo de usuario actualizado.

II.3.2 Componentes de un Sistema Hipermedia Adaptativo

Un Sistema Hipermedia Adaptativo tal como lo expone Buitrago (2010) está formado por tres componentes: el Modelo de Dominio, el Modelo de Usuario, y el Modelo de Adaptación. La Figura 2.1, muestra un esquema de cómo interactúan estos tres componentes y a continuación se describen cada uno de ellos.

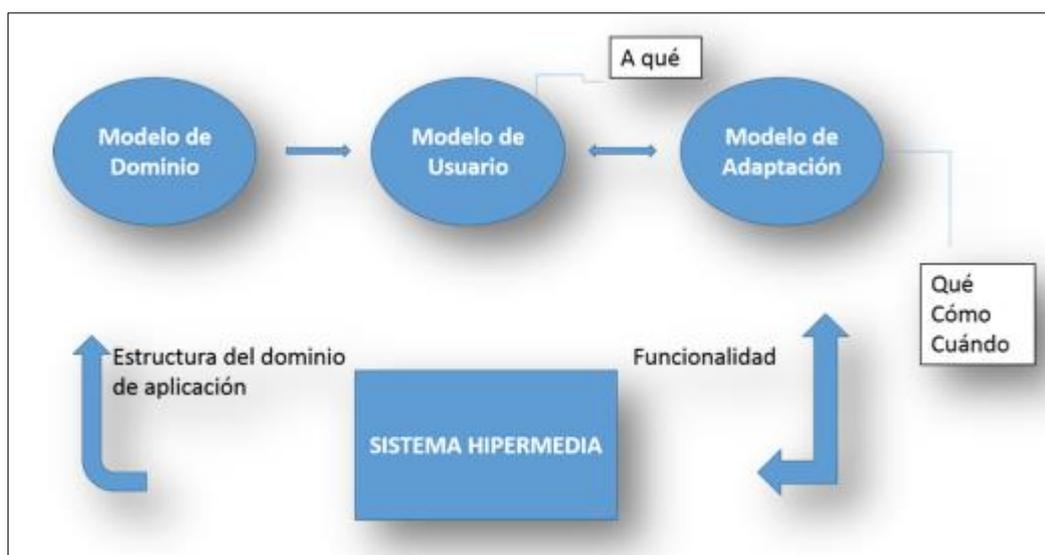


Figura 2.1: Componentes de los Sistemas Hipermedias Adaptativos (SHA). (Medina, Molina, García & Rodríguez, 2002)

Modelo del Dominio: estructura el conocimiento que se desea transmitir y almacena la información por conceptos, las relaciones de éstos con otros conceptos, y sus atributos. Un modelo de dominio o de contenido puede tener un conjunto de conceptos independientes, o puede poseer conceptos relacionados con otros, formando una clase de red semántica. Los sistemas educativos suelen utilizar requisitos previos que están vinculados al concepto; para avanzar a otro concepto se necesita haber aprendido una o varias nociones previas (Buitrago, 2010)

Modelo del Usuario: representa la relación de cada sujeto con el conocimiento que se le desea transmitir. Para ello, almacena y estructura aspectos relevantes de cada usuario. “Estos aspectos, llamados atributos, son almacenados en entidades tipo tabla que relacionan a cada usuario con sus características y con los conceptos del Modelo de Dominio” (Buitrago, 2010).

Modelo de Adaptación: está compuesto por un conjunto de reglas de adaptación que especifican qué y cómo se deben mostrar y comportar los elementos del sistema considerando el modelo de usuario.

II.3.3 Adaptación en los Sistemas Hipermedia Adaptativos

Para proporcionar adaptación en los sistemas hipermedia adaptativos se pueden tener en cuenta dos aspectos:

Un aspecto se refiere a “las características del usuario” a las que se adapta. Por ejemplo un sistema se puede adaptar al conocimiento previo, a los objetivos de aprendizaje, a las habilidades cognitivas, a las preferencias y habilidades de los usuarios, así como a los estilos de aprendizaje de los estudiantes.

Otro aspecto se refiere a “qué” elementos pueden ser adaptados por el sistema. Estos elementos son básicamente el contenido de las páginas y los enlaces de navegación. En base a esta separación, se definen dos tipos de adaptación denominadas adaptación de contenidos (o adaptación de la presentación) y adaptación en las opciones de navegación (o adaptación de enlaces) (Brusilovsky, 1996).

II.3.3.1 Métodos y Técnicas de Adaptación

Este aspecto es uno de los más utilizados para clasificar los sistemas hipermedias adaptativos, ya que permite distinguir los diferentes modos de adaptar la información a los usuarios.

Es importante diferenciar entre técnicas y métodos (Brusilovsky, 1996). La *técnica de adaptación* se define como un procedimiento que permite que los sistemas hipermedia se adapten. Una técnica es parte del nivel de implementación y puede ser descripta mediante una forma específica de representar el conocimiento y un algoritmo de adaptación. Por su parte, un *método de adaptación* se define como una generalización de una técnica de adaptación existente. Está basado en una idea clara de adaptación que puede representarse en el nivel conceptual. Un método puede ser implementado mediante diferentes técnicas. Al mismo tiempo, cada técnica puede ser usada para implementar distintos métodos utilizando la misma representación del conocimiento. Estos métodos y técnicas pueden ser combinados para crear adaptación en los sistemas hipermedias.

A continuación se presentan las técnicas y métodos para adaptar los contenidos que se presentan a los usuarios.

II.3.3.1.1 Métodos de adaptación contenido

Existen diferentes métodos que permiten adaptar los contenidos de un documento a cada tipo de usuario concreto.

El método de las *explicaciones adicionales* (Brusilovsky, 1996) tiene como objetivo incluir u ocultar aquellas partes de la información asociada a un concepto que se considere relevante o no para un determinado usuario en función de sus conocimientos acerca de tal concepto. Los usuarios con escasos conocimientos previos sobre un determinado tema necesitan explicaciones básicas innecesarias para otros usuarios que si tengan conocimientos sobre dicho tema. Por el contrario éstos últimos pueden acceder a información más detallada sobre el mismo concepto, la cual podría resultar abrumadora para el caso de los primeros. En general, algunos tipos o categorías de usuarios pueden obtener información adicional especialmente preparada para esa categoría, además de la información sobre los contenidos básicos. Este sistema ha sido aplicado en sistemas como C-Book (Kay & Kummerfeld, 1994).

Mediante los métodos de las *explicaciones de requisitos previos* y *explicaciones comparativas* (Brusilovsky, 1996) se puede adaptar la información presentada a un usuario sobre un concepto al grado de conocimientos de dichos usuario sobre los conceptos relacionados (C-Book (Kay & Kummerfeld, 1994)). El primer método consiste en insertar, antes de la explicación de un concepto determinado, explicaciones de todos los conceptos que se consideran requisitos previos y que el usuario no ha asimilado suficientemente. El método de las explicaciones comparativas consiste en incluir este tipo de explicaciones

entre el concepto que se presenta al estudiante y otros conceptos similares que haya asimilado previamente.

El método de las *variantes* (Brusilovsky, 1996) se basa en la afirmación de que no es suficiente mostrar u ocultar distintas partes de los contenidos a los distintos usuarios, dado que éstos pueden necesitar distinta información. En este caso, el sistema almacena distintas variaciones de unos mismos contenidos y el usuario obtiene la variante más apropiada para su modelo de usuario. (ANATOM-TUTOR (Beaumont, 1994), SyPros (Gonschorek & Herzog, 1995)).

Por último, mediante el método de *ordenación* (Brusilovsky, 1996) se muestran todos los fragmentos relacionados con el concepto que se desea presentar al usuario ordenados por relevancia, teniendo en cuenta la procedencia y los conocimientos del usuario como se implementa en Hypadapter (Hohl, Bocker & Gunzenhäuser, 1996).

II.3.3.1.2 Técnicas de adaptación contenido

Existen varias técnicas de adaptación de contenidos para la implementación de los distintos métodos de adaptación descritos en el apartado anterior.

En la técnica de *texto condicional* (Brusilovsky, 1996), la información relacionada con un concepto se divide en varias porciones de texto, cada una de las cuales lleva asociada una condición relacionada con el nivel de conocimiento requerido para acceder a ella que está representado en el modelo de usuario. Antes de presentar la información, el sistema selecciona las porciones de texto cuya condición se satisface. Esta técnica requiere programación por parte del autor, pero es muy flexible. Mediante su utilización se pueden implementar todos los métodos de adaptación mencionados anteriormente salvo la ordenación. Ha sido utilizada en sistemas como C-Book (Kay & Kummerfeld, 1994) y AHA (De Bra & Calvi, 1998), entre otros.

La técnica de *texto expandible* (*stretchtext*) sugerida en MetaDoc (Boyle & Encarnacion, 1994) e implementada en KN-AHS (Kobsa, Müller & Nill, 1994) es un tipo especial de hipertexto. En este caso, cuando se selecciona una palabra resaltada o una frase, ésta se reemplaza por una porción de texto, extendiendo el texto del documento actual. El texto extendido se puede colapsar y viceversa todas las veces que sea necesario. Aunque el sistema decide qué textos aparecen expandidos y cuáles comprimidos, el usuario puede modificarlo. En este caso, el sistema puede tomar nota de las preferencias del usuario para ofrecerle información de la manera más apropiada en sucesivas ocasiones.

El método de las variantes puede ser implementado a través de las técnicas de *variantes de fragmentos* y *variantes de páginas* (Brusilovsky, 1996), el sistema mantiene variaciones del mismo documento o fragmentos con distintas presentaciones del mismo contenido. Cada variación es apropiada para uno de los posibles estereotipos de usuarios. Al presentar un documento, el sistema selecciona la variante más apropiada para el usuario concreto, seleccionando un documento completo o componiendo el documento a partir de los fragmentos seleccionados, dependiendo de la técnica aplicada. La técnica de variantes de fragmentos se implementa en ANATOM-TUTOR (Beaumont, 1994). La técnica de variantes de páginas podemos observarla en ORIMUHS (Encarnação, 1995).

La más poderosa de todas las técnicas de adaptación de contenido es la *técnica basada en marcos* (Brusilovsky, 1996), implementada en Hypadapter (Hohl *et al.*, 1996) y en EPIAIM (De Rosis, Carolis & Pizzutilo, 1993). Con esta técnica toda la información referida a un concepto en particular se representa como un marco. Sus campos contienen distintas explicaciones para un concepto, enlaces a otros marcos, ejemplos, etc. Para decidir qué campos se deben presentar al usuario y en qué orden se utilizan reglas de presentación, que contienen condiciones relacionadas no sólo con el conocimiento del usuario acerca del concepto presentado, sino con cualquier característica representada en el modelo del usuario.

CAPÍTULO III

METODOLOGIA DE DESARROLLO

III.1. INTRODUCCIÓN

El desarrollo de software no es una tarea sencilla, prueba de ello es que existen numerosas propuestas metodológicas que inciden en distintas dimensiones del proceso de desarrollo. Por una parte tenemos aquellas propuestas más tradicionales que se centran especialmente en el control del proceso, estableciendo rigurosamente las actividades involucradas, los artefactos que se deben producir, y las herramientas y notaciones que se usarán. Estas propuestas han demostrado ser efectivas y necesarias en un gran número de proyectos (respecto a tiempos y recursos), donde por lo general se exige un alto grado de control en el proceso. Ante las dificultades para utilizar metodologías tradicionales con estas restricciones de tiempo y flexibilidad, muchos equipos de desarrollo se resignan a prescindir del “buen hacer” de la Ingeniería del Software, asumiendo el riesgo que ello conlleva. Ante esta situación, las metodologías ágiles aparecen como una posible respuesta para llenar este vacío metodológico. Por estar especialmente orientadas para proyectos pequeños y entornos cambiantes, las metodologías ágiles constituyen una solución a medida, con una elevada simplificación que a pesar de ello no renuncia a las prácticas esenciales para asegurar la calidad del producto. (Torres, López, Canós & Penadés, 2003).

Para muchos equipos de desarrollo el uso de metodologías tradicionales les resulta muy lejano a su forma de trabajo actual considerando las dificultades de su introducción e inversión asociada en formación y herramientas. Por otro, las características de los proyectos para los cuales las metodologías ágiles han sido especialmente pensadas se ajustan a un amplio rango de proyectos industriales de desarrollo de software; aquellos en los cuales los equipos de desarrollo son pequeños, con plazos reducidos, requisitos volátiles, y/o basados en nuevas tecnologías. (Torres *et al.*, 2003).

El presente trabajo trata sobre el desarrollo de una aplicación móvil basada en las nuevas tecnologías y con requisitos relativamente cambiantes por tal motivo se optó por aplicar una metodología de desarrollo ágil, lo cual es lo más adecuado considerando además que el equipo de desarrollo es pequeño y con plazos de desarrollo reducidos.

Este capítulo está organizado como sigue. En la sección dos se introducen las principales características de las metodologías ágiles, recogidas en el Manifiesto Ágil (Beck K. *et al.*, 2001). En la sección tres se presenta y detalla *Scrum*, la metodología ágil elegida para el desarrollo del presente trabajo y finalmente en la sección cuatro se fundamenta la elección de esta metodología.

III.2. METODOLOGÍAS DE DESARROLLO ÁGIL

La historia de las Metodologías Ágiles y su apreciación como tales en la comunidad de la Ingeniería de Software tiene sus inicios en la creación de una de las metodologías utilizada como arquetipo XP (*eXtreme Programming*), que nace de la mente de Kent Beck (2000), tomando ideas recopiladas junto a Ward Cunningham (Calderón, Dámaris, Rebaza & Carlos, 2007).

Durante 1996, Beck es llamado por Chrysler como asesor del proyecto Chrysler Comprehensive Compensation (C3) payroll system. Dada la poca calidad del sistema que se estaba desarrollando, Beck decide empezar de cero utilizando las prácticas que él había ido definiendo a lo largo del tiempo. El gran sistema, que administra la liquidación de aproximadamente 10.000 empleados, es puesto en operación en Mayo de 1997 casi respetando el calendario propuesto. Como consecuencia del éxito de dicho proyecto, Kent Beck dio origen a XP (Beck K., 2000) iniciando el movimiento de metodologías ágiles al que se anexarían otras metodologías surgidas mucho antes que el propio Beck fuera convocado por Chrysler.

Es así como este tipo de metodologías fueron inicialmente llamadas “metodologías livianas”, sin embargo, aún no contaban con una aprobación pues se las consideraba por muchos desarrolladores como meramente intuitiva. Luego, con el pasar de los años, en febrero de 2001, tras una reunión celebrada en Utah-EEUU, nace formalmente el término “ágil” aplicado al desarrollo de software. En esta misma reunión participan un grupo de 17 expertos de la industria del software (Beck K. *et al.*, 2001), incluyendo algunos de los creadores o impulsores de metodologías de software con el objetivo de esbozar los valores y principios que deberían permitir a los equipos desarrollar software rápidamente y respondiendo a los cambios que puedan surgir a lo largo del proyecto. Se pretendía ofrecer una alternativa a los procesos de desarrollo de software tradicionales, caracterizados por ser rígidos y dirigidos por la documentación que se genera en cada una de las actividades desarrolladas. (Calderón *et al.*, 2007)

Tras esta reunión se creó *The Agile Alliance*, una organización, sin fines de lucro, dedicada a promover los conceptos relacionados con el desarrollo ágil de software y ayudar a las organizaciones para que adopten dichos conceptos. El punto de partida fue el Manifiesto Ágil, un documento que resume la filosofía “ágil”. (Beck K. *et al.*, 2001).

III.2.1 El Manifiesto Ágil

El Manifiesto Ágil (Beck K. *et al.*, 2001) comienza enumerando los principales valores del desarrollo ágil. Según el Manifiesto se valora:

- ✓ ***Al individuo y las interacciones del equipo de desarrollo sobre el proceso y las herramientas.*** La gente es el principal factor de éxito de un proyecto software. Es más importante construir un buen equipo que construir el entorno. Muchas veces se comete el error de construir primero el entorno y esperar que el equipo se adapte automáticamente. Es mejor crear el equipo y que éste configure su propio entorno de desarrollo en base a sus necesidades
- ✓ ***Desarrollar software que funciona más que conseguir una buena documentación.*** La regla a seguir es “no producir documentos a menos que sean necesarios de forma inmediata para tomar una decisión importante”. Estos documentos deben ser cortos y centrarse en lo fundamental.
- ✓ ***La colaboración con el cliente más que la negociación de un contrato.*** Se propone que exista una interacción constante entre el cliente y el equipo de desarrollo. Esta colaboración entre ambos será la que marque la marcha del proyecto y asegure su éxito.
- ✓ ***Responder a los cambios más que seguir estrictamente un plan.*** La habilidad de responder a los cambios que puedan surgir a lo largo del proyecto (cambios en los requisitos, en la tecnología, en el equipo, etc.) determina también el éxito o fracaso del mismo. Por lo tanto, la planificación no debe ser estricta sino flexible y abierta.

Los valores anteriores inspiran los doce principios del manifiesto. Son características que diferencian un proceso ágil de uno tradicional. Los dos primeros principios son generales y resumen gran parte del espíritu ágil. El resto tienen que ver con el proceso a seguir y con el equipo de desarrollo, en cuanto metas a seguir y organización del mismo. Los principios son:

- I. *La prioridad es satisfacer al cliente mediante tempranas y continuas entregas de software que le aporte un valor.*
- II. *Dar la bienvenida a los cambios. Se capturan los cambios para que el cliente tenga una ventaja competitiva.*
- III. *Entregar frecuentemente software que funcione desde un par de semanas a un par de meses, con el menor intervalo de tiempo posible entre entregas.*
- IV. *La gente del negocio y los desarrolladores deben trabajar juntos a lo largo del proyecto.*
- V. *Construir el proyecto en torno a individuos motivados. Darles el entorno y el apoyo que necesitan y confiar en ellos para conseguir finalizar el trabajo.*
- VI. *El diálogo cara a cara es el método más eficiente y efectivo para comunicar información dentro de un equipo de desarrollo.*
- VII. *El software que funciona es la medida principal de progreso.*
- VIII. *Los procesos ágiles promueven un desarrollo sostenible. Los promotores, desarrolladores y usuarios deberían ser capaces de mantener una paz constante.*
- IX. *La atención continua a la calidad técnica y al buen diseño mejora la agilidad.*
- X. *La simplicidad es esencial.*
- XI. *Las mejores arquitecturas, requisitos y diseños surgen de los equipos organizados por sí mismos.*
- XII. *En intervalos regulares, el equipo reflexiona respecto a cómo llegar a ser más efectivo, y según esto ajusta su comportamiento.*

III.2.2 Por qué usar Metodologías Ágiles

Podemos afirmar que las metodologías tradicionales presentan algunos problemas a la hora de abordar ciertos proyectos, por ejemplo tienen costosas fases previas de especificación de requisitos, análisis y diseño. La corrección durante el desarrollo de errores introducidos en estas fases será costosa, es decir, se pierde flexibilidad ante los cambios. Por otro lado el desarrollo es más lento. Es difícil para los desarrolladores entender un sistema complejo en su globalidad y generalmente el proceso de desarrollo es abrumado por gran cantidad de documentación. (Calderón *et al.*, 2007).

Por el contrario, *las metodologías ágiles están especialmente indicadas en proyectos con requisitos poco definidos o cambiantes*. Estas metodologías se aplican bien en equipos pequeños que resuelven problemas concretos, lo que no restringe su aplicación en el desarrollo de grandes sistemas, ya que una correcta modularización de los mismos es

fundamental para su exitosa implementación. Las metodologías ágiles presentan diversas ventajas, entre las que podemos destacar:

- ❖ Capacidad de respuesta a cambios de requisitos a lo largo del desarrollo.
- ❖ Entrega continua de software funcional en plazos breves.
- ❖ Trabajo colaborativo entre el cliente y el equipo de desarrollo.
- ❖ Importancia de la simplicidad, eliminando el trabajo innecesario.
- ❖ Constante revisión de la técnica y el buen diseño.
- ❖ Mejora continua de los procesos y el equipo de desarrollo.

En definitiva, el desarrollo ágil de software intenta evitar los tortuosos y burocráticos caminos de las metodologías tradicionales, enfocándose en las personas y los resultados. Promueve iteraciones en el desarrollo a lo largo de todo el ciclo de vida del proyecto. Desarrollar software en cortos periodos de tiempo minimiza los riesgos, cada una de esas unidades de tiempo se llama iteración, la cual debe durar entre una y cuatro semanas. Cada iteración del ciclo de vida incluye: planificación, análisis de requerimientos, diseño, codificación, revisión y documentación. Cada iteración no debe añadir demasiada funcionalidad como para justificar el lanzamiento del producto al mercado, sino que la meta es conseguir una versión funcional sin errores. Al final de cada iteración, el equipo vuelve a evaluar las prioridades del proyecto. (Blanco, Camarero, Fumero, Warterski & Rodríguez, 2009)

III.2.3 Metodologías Ágiles en el Desarrollo de Software Móvil

A medida que el negocio de las aplicaciones móviles se va expandiendo y haciendo rentable, se tienen que investigar las metodologías óptimas para tales aplicaciones y entornos que permitan alcanzar el éxito de una forma atractiva y eficiente; Las metodologías tradicionales difieren bastante de las metodologías necesarias para el desarrollo de software móvil debido a que éste debe satisfacer requerimientos y restricciones especiales, además de exigir un alto nivel de calidad para operar adecuadamente en la enorme cantidad de terminales que llenan el mercado actual y el venidero. El mercado actual es altamente competitivo y cambiante. En ese contexto el desarrollo del software busca básicamente rapidez, calidad y reducción de costos en la ejecución de sus proyectos. Para asumir estos retos es necesario tener agilidad y flexibilidad. Estas características se constituyen en el fundamento mismo de las metodologías ágiles de desarrollo.

Las metodologías ágiles poseen ciertas características que las hacen totalmente aplicables al dominio del software móvil. Abrahamsson (2005) identifica los métodos ágiles como la solución potencial para el desarrollo de software en móviles. A continuación, en la Tabla 3.1 se presenta un análisis comparativo para probar la idoneidad de los métodos ágiles para el desarrollo de software móvil (Blanco *et. al.*, 2009).

Características Ágiles	Motivación Lógica	Desarrollo Móvil
Alta volatilidad del entorno	Debido a la alta frecuencia de cambios en los requerimientos, hay menos necesidad de diseño y planificación inicial y mayor necesidad de desarrollos incrementales e iterativos.	Alta incertidumbre, entornos dinámicos, cientos de nuevos terminales cada año.
Equipos de desarrollo pequeños	Capacidad de reacción más rápida, trabajo basado en la comunicación directa, menos documentación.	La mayor parte de los proyectos de desarrollo software para plataformas móviles se lleva a cabo en microempresas y PyME.
Cliente identificable	Desaparecen los malentendidos.	Existe un número ilimitado de usuarios finales, pero son fáciles de identificar.
Entornos de desarrollo orientados a objetos	La mayoría de las herramientas de desarrollo ágil existen bajo plataformas orientadas a objetos.	Se usan lenguajes O.O como Java y C++, entre muchos otros.
Software crítico no asegurado	Los fallos no causan gran impacto, como la pérdida de vidas. Es posible mayor agilidad en el desarrollo.	La mayoría del software es para entretenimiento. Los terminales no son fiables.
Software a nivel de aplicación	Sistemas embebidos grandes requieren comunicación exhaustiva y mecanismos de verificación.	Mientras los sistemas móviles son complejos y altamente dependientes, las aplicaciones son muy autónomas.
Sistemas pequeños	Menos necesidad de diseño inicial.	Las aplicaciones, aunque variables en tamaño, no suelen superar las 10.000 líneas de código.
Ciclos de desarrollo cortos	Propósito de realimentación rápida.	Periodos de desarrollo de 1 a 6 meses.

Tabla 3.1: Comparación entre las características ágiles y los rasgos en el desarrollo móvil (adaptado de Rahimian & Ramsin 2008).

El desarrollo de aplicaciones móviles difiere del desarrollo de software tradicional en muchos aspectos, lo que provoca que no todas las metodologías sean adecuadas para estos entornos. Esto es porque el software móvil tiene que satisfacer una serie de requerimientos y condicionantes especiales que lo hacen más complejo (Blanco *et. al.*, 2009). Como se ha visto en la Tabla 3.1, las metodologías ágiles aparecen como las más apropiadas para los proyectos de desarrollo móvil los cuales están basados en nuevas tecnologías y en donde los equipos de desarrollo son pequeños, los plazos reducidos y los requisitos poco definidos o cambiantes.

Dentro de las metodologías ágiles, existen diferentes propuestas como Crystal Clear, Extreme Programming, Lean o Kanban, que se adaptan excelentemente al desarrollo de software. Sin embargo, existe una metodología que ha probado ser un gran punto de partida y es ampliamente utilizada en el desarrollo de aplicaciones móviles. Su nombre es Scrum y será analizada en la siguiente sección. (Duran & Fajardo, 2012).

III.3. SCRUM

Scrum fue identificado y definido por Ikujiro Nonaka e Hirotaka Takeuchi a principios de los 80, al analizar cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard (Takeuchi & Nonaka, 1986).

En su estudio, Takeuchi y Nonaka (1986) compararon la nueva forma de trabajo en equipo, con el avance en formación de melé (scrum en inglés) de los jugadores de Rugby, a raíz de lo cual quedó acuñado el término “scrum” para referirse a ella. Aunque esta forma de trabajo surgió en empresas de productos tecnológicos, es apropiada para proyectos con requisitos inestables y para los que requieren rapidez y flexibilidad, situaciones frecuentes en el desarrollo de determinados sistemas de software.

En 1995 Ken Schwaber presentó “*Scrum Development Process*” en OOPSLA 95 (*Object-Oriented Programming Systems & Applications conference*) (Schwaber, 1995), un marco de reglas para desarrollo de software, basado en los principios de scrum, y que él había empleado en el desarrollo de Delphi, y Jeff Sutherland en su empresa Easel Corporation. (Palacio, 2014).

Hoy en día Scrum es usado por empresas de todos los tamaños tales como Yahoo!, Microsoft, Google, Lockheed Martin, Motorola, SAP, Cisco, GE, CapitalOne y la Reserva Federal de EE.UU. Muchos equipos que usan Scrum dicen haber obtenido mejoras sustanciales, y en algunos casos una completa transformación de la productividad y la moral. Para desarrolladores de productos (muchos de los cuales están cansados de los constantes cambios de tendencia en gestión) esto es significativo. Scrum es simple y poderoso. (Deemer, Benefield, Larman & Vodde, 2009).

III.3.1 Visión General

Scrum es un *marco de trabajo* donde equipos multifuncionales pueden crear productos o desarrollar proyectos de una forma iterativa. El Equipo de Scrum se conforma por el Dueño del Producto, el Scrum Master y el Equipo de Desarrollo. El Dueño del

Producto tiene la responsabilidad de decidir qué trabajo es necesario realizar y representa a todos los interesados en el producto. El ScrumMaster actúa como un líder servicial, ayudando al equipo y a la organización a usar lo mejor posible Scrum. El Equipo de Desarrollo construye el producto incrementalmente en ciclos de trabajo llamados *Sprints*. Un Sprint es un periodo de tiempo de una a cuatro semanas, con preferencias por los intervalos más cortos, que se suceden uno detrás de otro. En cada Sprint, el Equipo construye y entrega un *Incremento del Producto*. Cada incremento es un subconjunto del producto, reconocible y visualmente mejorado, que cumple con el criterio de aceptación y está construido con un nivel de calidad llamada *Definición de Terminado*. Al principio de cada Sprint, el Equipo de Scrum realiza la *Sprint Planning Meeting* y seleccionan elementos del *Product Backlog* (lista de historias de usuarios priorizadas) para construir el siguiente Incremento del Producto. A continuación el Equipo realiza un plan detallado para el desarrollo del próximo Sprint, dando lugar al *Sprint Backlog* (desglose de las historias de usuario para ser implementadas). Durante el Sprint no se podrán añadir nuevos elementos del Product Backlog; Scrum se adapta a los cambios en el siguiente Sprint, pero el Sprint actual está pensado para concentrarnos en un objetivo pequeño, claro y relativamente estable. Todos los días el Equipo se reúne brevemente (Daily Scrum Meeting) para inspeccionar su progreso y ajustar los siguientes pasos necesarios para completar el trabajo pendiente. Al final del Sprint, el Equipo revisa el Sprint con los diferentes Stakeholders (interesados e involucrados en el producto) y realiza una demostración de lo que han desarrollado (*Sprint Review* y *Sprint Retrospective*). Se obtiene un feedback que podrá ser incorporado en el siguiente Sprint. Scrum enfatiza la entrega de un Incremento del Producto al final del Sprint que cumple con el criterio de aceptación del Dueño del Producto y con la Definición de Terminado compartida por el Equipo de Desarrollo.

En la Figura 3.1 se resumen los principales roles, artefactos y eventos de Scrum.

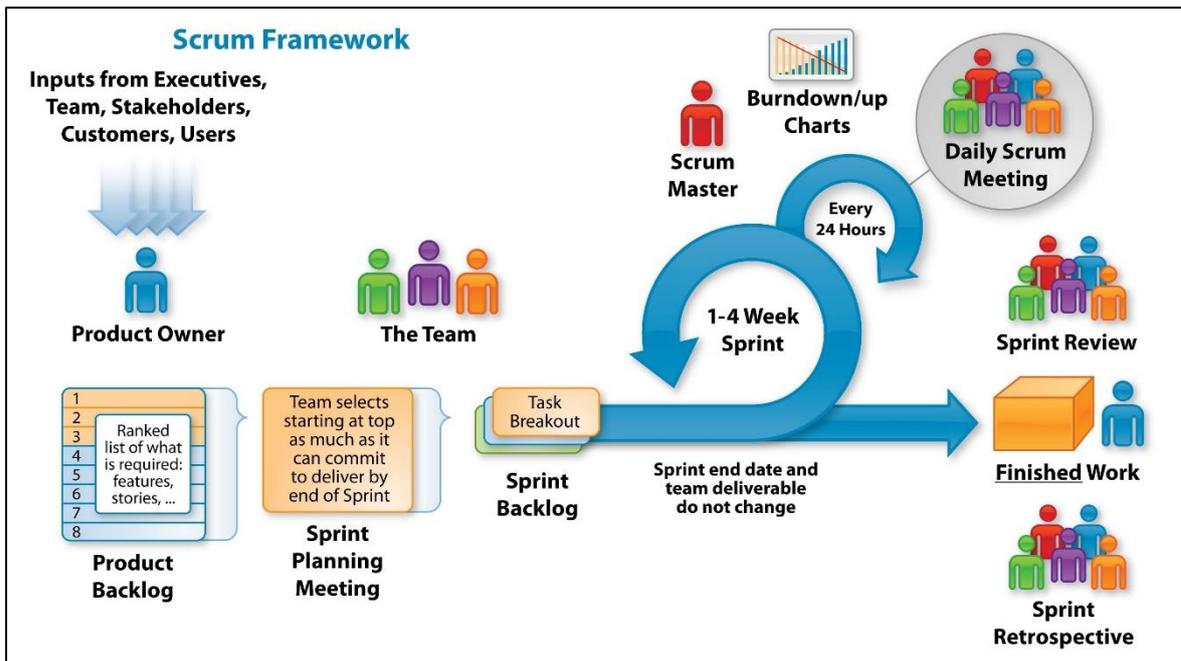


Figura 3.1: Visión General de Scrum (Hartman et al., 2015)

Un lema recurrente en Scrum es “inspección y adaptación”. Dado que el desarrollo implica de forma inevitable aprendizaje, innovación y sorpresas, Scrum enfatiza en dar pequeños pasos en el desarrollo, inspeccionando tanto el producto resultante como la eficacia de las prácticas actuales, y entonces adaptar los objetivos respecto al producto y las prácticas de los procesos y repetir indefinidamente.

A continuación, se profundiza en cada uno de los roles, artefactos y eventos de Scrum.

III.3.2 El Equipo Scrum (Scrum Team)

El Equipo Scrum está integrado por el Dueño del Producto, el Scrum Master y El Equipo de Desarrollo. Los Equipos Scrum son auto-organizados y multifuncionales. Los equipos auto-organizados eligen la mejor forma de llevar a cabo su trabajo y no son dirigidos por personas externas al equipo. Los equipos multifuncionales tienen todas las competencias necesarias para llevar a cabo el trabajo sin depender de otras personas que no son parte del equipo. El modelo de equipo en Scrum está diseñado para optimizar la flexibilidad, la creatividad y la productividad.

Los Equipos Scrum entregan productos de forma iterativa e incremental, maximizando las oportunidades de obtener retroalimentación. Las entregas incrementales

que cumple con la definición de terminado aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

III.3.2.1 El Dueño del Producto (*Product Owner*)

El Dueño de Producto es el responsable de maximizar el valor del producto y del trabajo del Equipo de Desarrollo, y es la única persona responsable de gestionar la Pila del Producto (Product Backlog). En el caso de un producto comercial el Dueño del Producto es responsable de maximizar el retorno de inversión (ROI) a base de identificar las funcionalidades del producto, trasladarlas a la Pila del Producto, decidir cuáles deberían estar al principio de la Pila para el siguiente Sprint, y repriorizar y refinar continuamente dicha Pila. En algunos casos el Dueño del Producto y el cliente son la misma persona; en otros, “el cliente” pueden ser millones de personas con necesidades muy variadas, en cuyo caso el rol del Product Owner es similar al del Product Manager (Director de Producto) o Product Marketing Manager (Director de Marketing de Producto). Sin embargo, el rol del Dueño del Producto es algo distinto del Product Manager tradicional, ya que interactúa de forma activa y regular con el Equipo, prioriza trabajando con todos los stakeholders y revisa los resultados de cada Sprint, en lugar de delegar las decisiones de desarrollo a un Jefe de Proyecto.

La gestión de la Pila del Producto por parte del Dueño del Producto implica:

- ✓ Expresar claramente los elementos de la Pila del Producto.
- ✓ Ordenar los elementos en la Pila del Producto para alcanzar los objetivos y misiones de la mejor manera posible.
- ✓ Optimizar el valor del trabajo desempeñado por el Equipo de Desarrollo.
- ✓ Asegurar que la Pila del Producto es visible, transparente y clara para todos, y que muestra aquello en lo que el equipo trabajará a continuación.
- ✓ Asegurar que el Equipo de Desarrollo entiende los elementos de la Pila del Producto al nivel necesario.

El Dueño del Producto podría hacer el trabajo anterior, o delegarlo al Equipo de Desarrollo. Sin embargo, en ambos casos el Dueño del Producto sigue siendo el responsable de dicho trabajo

Es muy importante notar que en Scrum el Dueño del Producto es una única persona, no un comité. El Dueño del Producto podría representar los deseos de un comité en la Pila del Producto, pero aquellos que quieran cambiar la prioridad de un elemento de

la Pila deben hacerlo a través del Dueño del Producto y para que éste pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Dueño del Producto se reflejan en el contenido y en la priorización de la Pila del Producto

III.3.2.2 *El Equipo de Desarrollo (Development Team)*

El Equipo de Desarrollo está formado por el grupo de profesionales que crean el Incremento del Producto con la definición de terminado y que potencialmente se puede poner en producción, al final de cada Sprint.

El Equipo de Desarrollo cumple con las siguientes características:

- ✓ Son auto-organizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Pila del Producto en Incrementos de funcionalidad potencialmente desplegables.
- ✓ Los Equipos de Desarrollo son multifuncionales, contando como equipo con todas las habilidades necesarias para crear un Incremento de producto.
- ✓ Scrum no reconoce títulos para los miembros de un Equipo de Desarrollo, todos son Desarrolladores, independientemente del trabajo que realice cada persona; no hay excepciones a esta regla.
- ✓ Scrum no reconoce sub-equipos en los equipos de desarrollo, no importan los dominios particulares que requieran ser tenidos en cuenta, como pruebas o análisis de negocio; no hay excepciones a esta regla.
- ✓ Los Miembros individuales del Equipo de Desarrollo pueden tener habilidades especializadas y áreas en las que estén más enfocados, pero la responsabilidad recae en el Equipo de Desarrollo como un todo.

El tamaño óptimo del Equipo de Desarrollo es lo suficientemente pequeño como para permanecer ágil y lo suficientemente grande como para completar una cantidad de trabajo significativa. Los Equipos de Desarrollo más pequeños podrían encontrar limitaciones en cuanto a las habilidades necesarias durante un Sprint, haciendo que el Equipo de Desarrollo no pudiese entregar un Incremento que potencialmente se pueda poner en producción. Tener más de nueve miembros en el equipo requiere demasiada coordinación. Los Equipos de Desarrollo grandes generan demasiada complejidad como para que pueda gestionarse mediante un proceso empírico. Los roles del Dueño del

Producto y Scrum Master no cuentan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en la Pila del Sprint (Sprint Backlog).

III.3.2.3 El Scrum Master

El Scrum Master es el responsable de asegurar que Scrum es entendido y aplicado. Los Scrum Masters hacen esto asegurándose de que el Equipo Scrum trabaja ajustándose a la teoría, prácticas y reglas de Scrum.

El Scrum Master es un líder que está al servicio del Equipo Scrum y ayuda a las personas externas al Equipo Scrum a entender qué interacciones con este Equipo pueden ser de ayuda y cuáles no. El Scrum Master ayuda a todos a modificar estas interacciones para maximizar el valor creado por el Equipo Scrum.

El Scrum Master proporciona servicio al Dueño del Producto de varias formas, incluyendo: encontrar técnicas para gestionar la Pila del Producto de manera efectiva; ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Pila del Producto claros y concisos; entender la planificación del producto en un entorno empírico; asegurar que el Dueño del Producto conozca cómo ordenar la Pila del Producto para maximizar el valor; entender y practicar la agilidad; y, facilitar los eventos de Scrum según se requiera o necesite.

Por otro lado, el Scrum Master apoya al Equipo de Desarrollo de varias formas: guiándolos en ser auto-organizado y multifuncional; ayudando a crear productos de alto valor; eliminando impedimentos para el progreso del Equipo; facilita los eventos de Scrum (reuniones o *meetings*) según se requieran o necesiten; y, guiando al Equipo en entornos de organizaciones donde Scrum aún no ha sido adoptado y entendido por completo.

Finalmente el Scrum Master ofrece servicio a la Organización de la siguiente manera: lidera y guía a la Organización en la aplicación de Scrum; planifica las implementaciones de Scrum en la organización; ayuda a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto; motiva cambios que incrementen la productividad del Equipo Scrum; y, trabaja con otros Scrum Masters para incrementar la efectividad de la aplicación de Scrum en la organización.

III.3.3 Eventos de Scrum

En Scrum existen eventos predefinidos con el fin de crear regularidad y minimizar la necesidad de reuniones no definidas en Scrum. Todos los eventos son bloques de tiempo (*time-boxes*), de tal modo que todos tienen una duración máxima. Una vez que comienza

un Sprint, su duración es fija y no puede acortarse o alargarse. Los demás eventos pueden terminar siempre que se alcance el objetivo del evento, asegurando que se emplee una cantidad apropiada de tiempo sin permitir desperdicio en el proceso.

Además del propio Sprint, que es un contenedor del resto de eventos, cada uno de los eventos de Scrum constituye una oportunidad formal para la inspección y adaptación de algún aspecto. Estos eventos están diseñados específicamente para habilitar la transparencia e inspección. La falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida para inspeccionar y adaptarse.

III.3.3.1 El Sprint

El corazón de Scrum es el Sprint, es un bloque de tiempo (time-box) de un mes o menos durante el cual se crea un incremento de producto “Terminado”, utilizable y potencialmente desplegable. Es más conveniente si la duración de los Sprints es consistente a lo largo del esfuerzo de desarrollo. Cada nuevo Sprint comienza inmediatamente después de la finalización del Sprint previo.

Los Sprints contienen y consisten de la Reunión de Planificación del Sprint (*Sprint Planning Meeting*), los Scrums Diarios (*Daily Scrums*), el trabajo de desarrollo, la Revisión del Sprint (*Sprint Review*), y la Retrospectiva del Sprint (*Sprint Retrospective*).

Durante el Sprint no se realizan cambios que puedan afectar al Objetivo del Sprint (*Sprint Goal*); Los objetivos de calidad no disminuyen; y, el alcance puede ser clarificado y renegociado entre el Dueño del Producto y el Equipo de Desarrollo a medida que se va aprendiendo más.

Cada Sprint puede considerarse un proyecto con un horizonte no mayor de un mes. Al igual que los proyectos, los Sprints se usan para lograr algo. Cada Sprint tiene una definición de qué se va a construir, un diseño y un plan flexible que guiará la construcción y el trabajo y el producto resultante.

Los Sprints están limitados a un mes calendario. Cuando el horizonte de un Sprint es demasiado grande la definición de lo que se está construyendo podría cambiar, la complejidad podría elevarse y el riesgo podría aumentar. Los Sprints habilitan la predictibilidad al asegurar la inspección y adaptación del progreso al menos en cada mes calendario. Los Sprints también limitan el riesgo al costo de un mes calendario.

Un Sprint puede ser cancelado antes de que el bloque de tiempo llegue a su fin. Solo el Dueño del Producto tiene la autoridad para cancelar el Sprint, aunque puede hacerlo bajo la influencia de los interesados, del Equipo de Desarrollo o del Scrum Master.

Un Sprint se cancelaría si el Objetivo del Sprint llega a quedar obsoleto. Esto podría ocurrir si la compañía cambia la dirección o si las condiciones del mercado o de la tecnología cambian. En general, un Sprint debería cancelarse si no tuviese sentido seguir con él dadas las circunstancias. Pero debido a la corta duración de los Sprints, rara vez la cancelación tiene sentido.

Cuando se cancela un Sprint, se revisan todos los Elementos de la Pila del Producto que se hayan completado y “Terminado”. Si una parte del trabajo es potencialmente entregable, el Dueño del Producto normalmente lo acepta. Todos los Elementos de la Pila del Producto no completados se vuelven a estimar y se vuelven a introducir en esta Pila.

Las cancelaciones de Sprint consumen recursos, ya que todos deben reagruparse en otra Reunión de Planificación de Sprint para empezar otro Sprint. Las cancelaciones de Sprint son a menudo traumáticas para el Equipo Scrum y son muy poco comunes.

III.3.3.2 Reunión de Planificación del Sprint (*Sprint Planning Meeting*)

El trabajo a realizar durante el Sprint se planifica en la Reunión de Planificación de Sprint. Este plan se crea mediante el trabajo colaborativo del Equipo Scrum completo.

La Reunión de Planificación de Sprint tiene un máximo de duración de ocho horas para un Sprint de un mes. Para Sprints más cortos, el evento es usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña al Equipo Scrum a mantenerse dentro del bloque de tiempo.

La Reunión de Planificación de Sprint responde a dos cuestiones:

❖ *¿Qué puede ser terminado en este Sprint?*

El Equipo de Desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint. El Dueño del Producto discute el objetivo que el Sprint debería lograr y los Elementos de la Pila del Producto que, si se completan en el Sprint, lograrían el Objetivo del Sprint. El Equipo Scrum completo colabora en el entendimiento del trabajo del Sprint.

La entrada a esta reunión está constituida por la Pila del Producto, el último Incremento del Producto, la capacidad proyectada del Equipo de Desarrollo para el Sprint, y el rendimiento pasado del Equipo de Desarrollo. El número de elementos

de la Pila del Producto seleccionados para el Sprint depende únicamente del Equipo de Desarrollo. Solo el Equipo de Desarrollo puede evaluar qué es capaz de lograr durante el Sprint que comienza.

Después de que el Equipo de Desarrollo programa qué elementos de la Pila del Producto entregará en el Sprint, el Equipo Scrum elabora un Objetivo del Sprint (*Sprint Goal*). El Objetivo del Sprint debería lograrse durante el Sprint a través de la implementación de la Pila del Producto, y provee una guía al equipo de desarrollo de por qué se está construyendo el incremento.

❖ *¿Cómo se conseguirá completar el trabajo seleccionado?*

Una vez que se ha establecido el objetivo y seleccionado los elementos de la Pila del Producto para el Sprint, el Equipo de Desarrollo decide cómo construirá esta funcionalidad para formar un Incremento de producto “Terminado”. Los elementos de la Pila del Producto seleccionados para este Sprint, más el plan para terminarlos, recibe el nombre de Pila del Sprint (Sprint Backlog).

El Equipo de Desarrollo por lo general comienza diseñando el sistema y el trabajo necesario para convertir la Pila del Producto en un Incremento de producto funcional. El trabajo podría ser de tamaño o esfuerzo estimado variables. Sin embargo, durante la Reunión de Planificación del Sprint, se planifica suficiente trabajo como para que el Equipo de Desarrollo pueda hacer una proyección de lo que cree que puede completar en el Sprint que comienza. Para el final de esta reunión, el trabajo planificado por el Equipo de Desarrollo para los primeros días del Sprint es descompuesto en unidades de un día o menos. El Equipo de desarrollo se auto-organiza para asumir el trabajo de la Pila del Sprint, tanto durante la reunión de Planificación de Sprint como a lo largo del Sprint.

El Dueño del Producto puede ayudar a clarificar los elementos de la Pila del Producto seleccionados y hacer concesiones. Si el Equipo de Desarrollo determina que tiene demasiado trabajo o que no tiene suficiente trabajo, podría renegociar los elementos de la Pila del Producto seleccionados con el Dueño del Producto. El Equipo de Desarrollo podría también invitar a otras personas a que asistan con el fin de que proporcionen asesoría técnica o relacionada con el dominio.

Al finalizar la Reunión de Planificación de Sprint, el Equipo de Desarrollo debería ser capaz de explicar al Dueño del Producto y al Scrum Master cómo pretende trabajar como un equipo auto-organizado para lograr el Objetivo del Sprint y crear el Incremento esperado.

III.3.3.3 *Objetivo del Sprint (Sprint Goal)*

El Objetivo del Sprint es una meta establecida para el Sprint que puede ser alcanzada mediante la implementación de la Pila del Producto. Proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento. Es creado durante la reunión de Planificación del Sprint. El objetivo del Sprint ofrece al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint. Los elementos de la Pila del Producto seleccionados ofrecen una función coherente, que puede ser el objetivo del Sprint. El objetivo del Sprint puede representar otro nexo de unión que haga que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas.

A medida que el equipo de desarrollo trabaja, se mantiene el objetivo del Sprint en mente. Con el fin de satisfacer el objetivo del Sprint se implementa la funcionalidad y la tecnología. Si el trabajo resulta ser diferente de lo que el Equipo de Desarrollo espera, ellos colaboran con el Dueño del Producto para negociar el alcance de la Pila del Sprint (Sprint Backlog).

III.3.3.4 *Scrum Diario (Daily Scrum)*

El Scrum Diario es una reunión con un bloque de tiempo de quince minutos para que el Equipo de Desarrollo sincronice sus actividades y cree un plan para las siguientes 24 horas. Esto se lleva a cabo inspeccionando el trabajo avanzado desde el último Scrum Diario y haciendo una proyección acerca del trabajo que podría completarse antes del siguiente Sprint.

El Scrum Diario se realiza a la misma hora y en el mismo lugar todos los días para reducir la complejidad. Durante la reunión, cada miembro del Equipo de Desarrollo explica:

- ¿Qué hice ayer que ayudó al Equipo de Desarrollo a lograr el Objetivo del Sprint?
- ¿Qué haré hoy para ayudar al Equipo de Desarrollo a lograr el Objetivo del Sprint?
- ¿Veo algún impedimento que evite que el Equipo de Desarrollo o yo logremos el Objetivo del Sprint?

El Equipo de Desarrollo usa el Scrum Diario para evaluar el progreso hacia el Objetivo del Sprint y para evaluar qué tendencia sigue este progreso hacia la finalización del trabajo contenido en la Pila del Sprint. El Scrum Diario optimiza las posibilidades de que el Equipo de Desarrollo cumpla el Objetivo del Sprint. Cada día, el Equipo de

Desarrollo debería entender cómo intenta trabajar en conjunto como un equipo auto-organizado para lograr el Objetivo del Sprint y crear el Incremento esperado hacia el final del Sprint. El Equipo de Desarrollo o los miembros del equipo a menudo se vuelven a reunir inmediatamente después del Scrum Diario, para tener discusiones detalladas, o para adaptar, o replanificar el resto del trabajo del Sprint.

El Scrum Master se asegura de que el Equipo de Desarrollo realice la reunión, pero el Equipo de Desarrollo es el responsable de dirigir el Scrum Diario. El Scrum Master enseña al Equipo de Desarrollo para que mantenga el Scrum Diario en los límites del bloque de tiempo de quince minutos y se asegura de que se cumpla la regla de que solo los miembros del Equipo de Desarrollo participan en el Scrum Diario.

Los Scrum Diarios mejoran la comunicación, eliminan la necesidad de mantener otras reuniones, identifican y eliminan impedimentos relativos al desarrollo, resaltan y promueven la toma de decisiones rápida, y mejoran el nivel de conocimiento del Equipo de Desarrollo. El Scrum Diario constituye una reunión clave de inspección y adaptación.

III.3.3.5 Revisión de Sprint (*Sprint Review*)

Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Pila del Producto si fuese necesario. Durante la Revisión de Sprint, el Equipo Scrum y los interesados colaboran acerca de lo que se hizo durante el Sprint. Basándose en esto, y en cualquier cambio a la Pila del Producto durante el Sprint, los asistentes colaboran para determinar las siguientes cosas que podrían hacerse para optimizar el valor. Se trata de una reunión informal, no una reunión de seguimiento, y la presentación del Incremento tiene como objetivo facilitar la retroalimentación de información y fomentar la colaboración.

Se trata de una reunión restringida a un bloque de tiempo de cuatro horas para Sprints de un mes. Para Sprints más cortos, se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito. El Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado.

La Revisión de Sprint incluye los siguientes elementos:

- ❖ Los asistentes son el Equipo Scrum y los interesados clave invitados por el Dueño del Producto.

- ❖ El Dueño del Producto explica qué elementos de la Pila del Producto se han “Terminado” y cuales no se han “Terminado”.
- ❖ El Equipo de Desarrollo habla acerca de qué fue bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos dichos problemas.
- ❖ El Equipo de Desarrollo demuestra el trabajo que ha “Terminado” y responde preguntas acerca del Incremento.
- ❖ El Dueño del Producto habla acerca de la Pila del Producto en el estado actual. Proyecta fechas de finalización probables en el tiempo basándose en el progreso obtenido hasta la fecha (si es necesario).
- ❖ El grupo completo colabora acerca de qué hacer a continuación, de modo que la Revisión de Sprint proporcione información de entrada valiosa para Reuniones de Planificación de Sprints subsiguientes.
- ❖ Revisión de cómo el mercado o el uso potencial del producto podría haber cambiado lo que es de más valor para hacer a continuación.
- ❖ Revisión de la línea de tiempo, presupuesto, capacidades potenciales y mercado para la próxima entrega prevista del producto.

El resultado de la Revisión de Sprint es una Pila del Producto revisada, que define los elementos posibles para el siguiente Sprint. Además es posible que la Pila del Producto reciba un ajuste general para enfocarse en nuevas oportunidades.

III.3.3.6 Retrospectiva de Sprint (*Sprint Retrospective*)

La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y crear un plan de mejoras que sean abordadas durante el siguiente Sprint.

La Retrospectiva de Sprint tiene lugar después de la Revisión de Sprint y antes de la siguiente Reunión de Planificación del Sprint. Se trata de una reunión restringida a un bloque de tiempo de tres horas para Sprints de un mes. Para Sprints más cortos se reserva un tiempo proporcionalmente menor. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, además el Scrum Master enseña a todos a mantener el evento dentro del bloque de tiempo fijado y participa en la reunión como un miembro del equipo ya que la responsabilidad del proceso Scrum recae sobre él.

El propósito de la Retrospectiva de Sprint es: inspeccionar cómo fue el último Sprint en cuanto a personas, relaciones, procesos y herramientas; identificar y ordenar los

elementos más importantes que salieron bien y las posibles mejoras; y finalmente se espera crear un plan para implementar las mejoras a la forma en la que el Equipo Scrum desempeña su trabajo.

El Scrum Master alienta al equipo para que mejore dentro del marco de proceso Scrum, su proceso de desarrollo y sus prácticas para hacerlos más efectivos en el siguiente Sprint. Durante cada Retrospectiva de Sprint, el Equipo Scrum planifica formas de aumentar la calidad del producto mediante la adaptación de la Definición de Terminado (*Definition of Done*) según sea conveniente.

Para el final de la Retrospectiva de Sprint, el Equipo Scrum debería haber identificado mejoras que implementará en el próximo Sprint. El hecho de implementar estas mejoras en el siguiente Sprint, constituye la adaptación subsecuente a la inspección del Equipo de Desarrollo a sí mismo. Aunque las mejoras pueden implementarse en cualquier momento, la Retrospectiva de Sprint ofrece un evento dedicado para este fin, enfocado en la inspección y la adaptación.

III.3.4 Artefactos de Scrum

Los artefactos de Scrum representan trabajo o valor en diversas formas que son útiles para proporcionar transparencia y oportunidades para la inspección y adaptación. Los artefactos definidos por Scrum están diseñados específicamente para maximizar la transparencia de la información clave, que es necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

III.3.4.1 Pila del Producto (*Product Backlog*)

La Pila del Producto es una lista ordenada de todo lo que podría ser necesario en el producto (historias de usuario), y es la única fuente de requisitos para cualquier cambio a realizarse en el producto. El Dueño de Producto es el responsable de la Pila del Producto, incluyendo su contenido, disponibilidad y ordenación.

Una Pila de Producto nunca está completa. El desarrollo más temprano de la misma solo refleja los requisitos conocidos y mejor entendidos al principio. La Pila del Producto evoluciona a medida que el producto y el entorno en el que se usará también lo hacen. La Pila del Producto es dinámica; cambia constantemente para identificar lo que el producto necesita para ser adecuado, competitivo y útil. Mientras el producto exista, su Pila del Producto también existe.

La Pila del Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a ser hechos sobre el producto para entregas futuras. Los elementos de esta Pila tienen como atributos la descripción, la ordenación, la estimación y el valor.

A medida que un producto es utilizado y se incrementa su valor, y el mercado proporciona retroalimentación, la Pila del Producto se convierte en una lista más larga y exhaustiva. Los requisitos nunca dejan de cambiar, así que la Pila del Producto es un artefacto vivo. Los cambios en los requisitos de negocio, las condiciones del mercado o la tecnología podrían causar cambios en la Pila del Producto.

A menudo, varios Equipos Scrum trabajan juntos en el mismo producto. Para describir el trabajo a realizar sobre el producto, se utiliza una única Pila del Producto. En ese caso podría emplearse un atributo de dicha Pila para agrupar los elementos.

El refinamiento (*refinement*) de la Pila del Producto es el acto de añadir detalle, estimaciones y orden a los elementos que la conforman. Se trata de un proceso continuo, en el cual el Dueño de Producto y el Equipo de Desarrollo colaboran acerca de los detalles de los elementos de esta Pila. Durante este refinamiento, se examinan y revisan cada uno de los elementos de la Pila del Producto. El Equipo Scrum decide cómo y cuándo se hace el refinamiento. Este usualmente consume no más del diez por ciento de la capacidad del Equipo de Desarrollo. Sin embargo, los elementos de la Pila del Producto pueden actualizarse en cualquier momento por el Dueño de Producto o a criterio suyo.

Los elementos de la Pila del Producto de orden más alto son generalmente más claros y detallados que los de menor orden. Se realizan estimaciones más precisas basándose en la mayor claridad y detalle; cuanto más bajo es el orden, menor es el detalle. Los elementos de la Pila del Producto de los que se ocupará el Equipo de Desarrollo en el siguiente Sprint tienen una granularidad mayor, habiendo sido descompuestos de forma que cualquier elemento puede ser “Terminado” dentro de los límites del bloque de tiempo del Sprint. Los elementos de la Pila del Producto que pueden ser “Terminados” por el Equipo de Desarrollo en un Sprint son considerados “preparados” o “accionables” para ser seleccionados en una reunión de Planificación de Sprint. Los elementos de la Pila del Producto normalmente adquieren este grado de transparencia mediante las actividades de refinamiento descritas anteriormente.

El Equipo de Desarrollo es el responsable de proporcionar todas las estimaciones. El Dueño de Producto podría influenciar al Equipo ayudándolos a entender y seleccionar

soluciones de compromiso, pero las personas que harán el trabajo son las que hacen la estimación final.

Seguimiento del Progreso Hacia un Objetivo

En cualquier momento, es posible sumar el trabajo total restante para alcanzar el objetivo. El Dueño del Producto hace seguimiento de este trabajo restante total al menos en cada Revisión de Sprint, donde compara esta cantidad con el trabajo restante en Revisiones de Sprint previas, para evaluar el progreso hacia la finalización del trabajo proyectado en el tiempo deseado para el objetivo. Esta información se muestra de forma transparente a todos los interesados.

Varias prácticas de proyección sobre tendencias se han utilizado para predecir el progreso, como trabajo consumido (*burndown*), avanzado (*burnup*) y flujo acumulado (*cumulative flow*). Estas se han revelado como útiles. Sin embargo, no remplazan la importancia del empirismo. En entornos complejos, se desconoce lo que ocurrirá. Solo lo que ya ha ocurrido puede utilizarse para la toma de decisiones con miras al futuro.

III.3.4.2 Pila del Sprint (*Sprint Backlog*)

La Pila del Sprint es el conjunto de elementos de la Pila del Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Pila del Sprint es una predicción hecha por el Equipo de Desarrollo acerca de qué funcionalidad formará parte del próximo Incremento y del trabajo necesario para entregar esa funcionalidad en un Incremento “Terminado”.

La Pila del Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint. Esta Pila es un plan con un nivel de detalle suficiente como para que los cambios en el progreso se puedan entender en el Scrum Diario. El Equipo de Desarrollo modifica la Pila del Sprint durante el Sprint y ésta emerge a lo largo del Sprint. Esto ocurre a medida que el Equipo de Desarrollo trabaja sobre el plan y aprende más acerca del trabajo necesario para conseguir el Objetivo del Sprint.

Según se requiere nuevo trabajo, el Equipo de Desarrollo lo añade a la Pila del Sprint. A medida que el trabajo se ejecuta o se completa, se va actualizando la estimación de trabajo restante. Cuando algún elemento del plan pasa a ser considerado innecesario, es eliminado. Durante un Sprint, solo el Equipo de Desarrollo puede cambiar su Pila del

Sprint y ésta es una imagen visible en tiempo real del trabajo que el Equipo de Desarrollo planea llevar a cabo durante el Sprint, y pertenece únicamente al Equipo de Desarrollo.

Seguimiento del Progreso del Sprint

En cualquier momento durante un Sprint, es posible sumar el trabajo restante total en los elementos de la Pila del Sprint. El Equipo de Desarrollo hace seguimiento de este trabajo restante total al menos en cada Scrum Diario para proyectar la posibilidad de conseguir el Objetivo del Sprint. Haciendo seguimiento del trabajo restante a lo largo del Sprint, el Equipo de Desarrollo puede gestionar su progreso.

III.3.4.3 Incremento

El Incremento es la suma de todos los elementos de la Pila del Producto completados durante un Sprint y el valor de los incrementos de todos los Sprints anteriores. Al final de un Sprint, el nuevo Incremento debe estar “Terminado”, lo cual significa que está en condiciones de ser utilizado y que cumple la Definición de “Terminado” del Equipo Scrum. El incremento debe estar en condiciones de utilizarse sin importar si el Dueño de Producto decide liberarlo o no.

III.3.5 Transparencia de los Artefactos

Scrum se basa en la transparencia. Las decisiones para optimizar el valor y controlar el riesgo se toman basadas en el estado percibido de los artefactos. En la medida en que la transparencia sea completa, estas decisiones tienen unas bases sólidas. En la medida en que los artefactos no son completamente transparentes, estas decisiones pueden ser erróneas, el valor puede disminuir y el riesgo puede aumentar.

El Scrum Master debe trabajar con el Dueño del Producto, el Equipo de Desarrollo y otras partes involucradas para entender si los artefactos son completamente transparentes. Hay prácticas para hacer frente a la falta de transparencia; el Scrum Master debe ayudar a todos a aplicar las prácticas más apropiadas si no hay una transparencia completa. Un Scrum Master puede detectar la falta de transparencia inspeccionando artefactos, reconociendo patrones, escuchando atentamente lo que se dice y detectando diferencias entre los resultados esperados y los reales.

Es responsabilidad del Scrum Master trabajar con el Equipo Scrum y la organización para mejorar la transparencia de los artefactos. Este trabajo usualmente

incluye aprendizaje, convicción y cambio. La transparencia no ocurre de la noche a la mañana, sino que es un camino.

III.3.6 Definición de Terminado (Definition of Done)

Cuando un Incremento se describe como “Terminado”, todo el mundo debe entender lo que significa “Terminado”. Aunque esto varía significativamente para cada Equipo Scrum, los miembros del Equipo deben tener un entendimiento compartido de lo que significa que el trabajo esté completado, para asegurar la transparencia. Esta es la *Definición de Terminado* para el Equipo Scrum y se utiliza para evaluar cuándo se ha completado el trabajo sobre el Incremento del producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos de la Pila del Producto puede seleccionar durante una reunión de Planificación del Sprint. El propósito de cada Sprint es entregar Incrementos de funcionalidad que potencialmente se puedan poner en producción, y que se ajustan a la Definición de Terminado actual del Equipo Scrum.

Los Equipos de Desarrollo entregan un Incremento de funcionalidad de producto en cada Sprint. Este Incremento es utilizable, de modo que el Dueño del Producto podría elegir liberarlo inmediatamente. Si la Definición de Terminado para un incremento es parte de las convenciones, estándares o guías de la organización de desarrollo, al menos todos los Equipos Scrum deben seguirla. Si “Terminado” para un incremento no es una convención de la organización de desarrollo, el Equipo de Desarrollo del Equipo Scrum debe definir una Definición de Terminado apropiada para el producto. Si hay múltiples Equipos Scrum trabajando en la entrega del sistema o producto, los equipos de desarrolladores en todos los Equipos Scrum deben definir en conjunto la Definición de Terminado.

Cada Incremento se integra con todos los Incrementos anteriores y es probado exhaustivamente, asegurando que todos los Incrementos funcionan en conjunto. A medida que los Equipos Scrum maduran, se espera que su Definición de Terminado se amplíe para incluir criterios más rigurosos para una mayor calidad. Cualquier producto o sistema debería tener una Definición de Terminado que es un estándar para cualquier trabajo realizado sobre él.

III.4 POR QUÉ ELEGIMOS SCRUM

Scrum es el marco de trabajo ágil más utilizado para el desarrollo móvil. Scrum utiliza una estrategia “flexible”, donde el cliente tiene más transparencia sobre el progreso actual del proyecto y puede incluso cambiar el plan de ejecución del proyecto en cualquier momento de acuerdo a los valores organizacionales o factores del entorno, que no es posible con el enfoque tradicional o secuencial.

Scrum como metodología ágil se ajusta de forma natural al desarrollo móvil ya que está preparado y responde adecuadamente a los cambios tecnológicos, frecuentes en el ambiente móvil. Por otro lado el modelo de Sprints brinda características claves entre las que se destacan: soporte del modelo de actualizaciones utilizado por las aplicaciones móviles; incremento de la fiabilidad al utilizar múltiples ciclos de pruebas y controles de calidad; y mejora la experiencia del usuario, ya que permite experimentar con opciones, ajuste de diseño y características de la aplicación a medida que se entregan los incrementos, de tal manera que la experiencia del usuario es rápida, lisa y transparente.

Asimismo, las aplicaciones móviles son diferentes de otras aplicaciones no móviles en diferentes aspectos:

- ✓ Necesitan tener un tamaño reducido.
- ✓ Actualizarse rápidamente.
- ✓ Interactuar correctamente con el backend del servidor cuando es necesario.
- ✓ Estar libres de errores y ser rápidas.

Scrum permite cumplir con estos requisitos a través de los Sprints, los Incrementos, los múltiples ciclos de pruebas y la participación frecuente de los interesados a lo largo del proceso de desarrollo, lo que la hace una metodología ideal para el desarrollo de aplicaciones móviles y es la metodología elegida para llevar a cabo el desarrollo de la aplicación móvil planteada en este trabajo.

CAPÍTULO IV

DESARROLLO DE LA APLICACIÓN

IV.1. INTRODUCCION

El presente trabajo involucra el desarrollo de una aplicación móvil cliente-servidor. Por un lado tendremos una aplicación nativa para la plataforma Android (Cliente) y por otra parte una aplicación web que trabajará en conjunto con la primera (Servidor). La aplicación móvil será utilizada por los estudiantes y permitirá la lectura de códigos QR asociados a componentes de red con el fin de obtener información personalizada. Por su parte la aplicación web será utilizada por los docentes para gestionar toda la información vinculada al modelo de dominio y al modelo de usuario. Asimismo, del lado del servidor estará implementado el módulo de adaptación encargado de generar información adaptada en base al estilo de aprendizaje y nivel de conocimiento del alumno.

Este capítulo abarca el diseño y desarrollo de la aplicación móvil cliente-servidor utilizando la metodología de desarrollo ágil Scrum.

A continuación se presenta la arquitectura global, la Pila del Producto, la Definición de Terminado, las herramientas de desarrollo y cada uno de los Sprints ejecutados.

IV.2. APLICANDO SCRUM – Arquitectura de la Aplicación

Los roles de Scrum son ejecutados de la siguiente manera:

Product Owner: Dra. Elena B. Durán (Profesor Guía del Trabajo Final).

Scrum Master: Ing. Gregorio N. Tkachuk (Asesor del Trabajo Final).

Team: Matías Hernán Loto (Alumno Autor del Trabajo Final).

Como se describió Scrum es un framework que nos permite el desarrollo de sistemas complejos de manera iterativa e incremental, sin embargo Scrum no habla explícitamente de todos los preparativos previos a comenzar el desarrollo ni de la arquitectura del sistema, por tal motivo en esta sección, se presenta algo vital en todo sistema, la arquitectura global que guiará el desarrollo. En esta etapa se definen además los entornos de desarrollo, se trabaja en el Product Backlog, se hace una previsión del reparto de historias de usuario por iteración; y finalmente se hace el diseño de la arquitectura.

La arquitectura del sistema se presenta en la Figura 4.1, la cual consta de dos grandes componentes: Cliente y Servidor. A grandes rasgos el cliente es la aplicación nativa para Android que se ejecuta en el dispositivo móvil. El servidor es la aplicación web programada usando PHP, JavaScript y MySQL que implementa el módulo de adaptación y permite gestionar todo lo relacionado al Modelo de Dominio y al Modelo de Usuario. Internet es el medio de comunicación entre ambos.

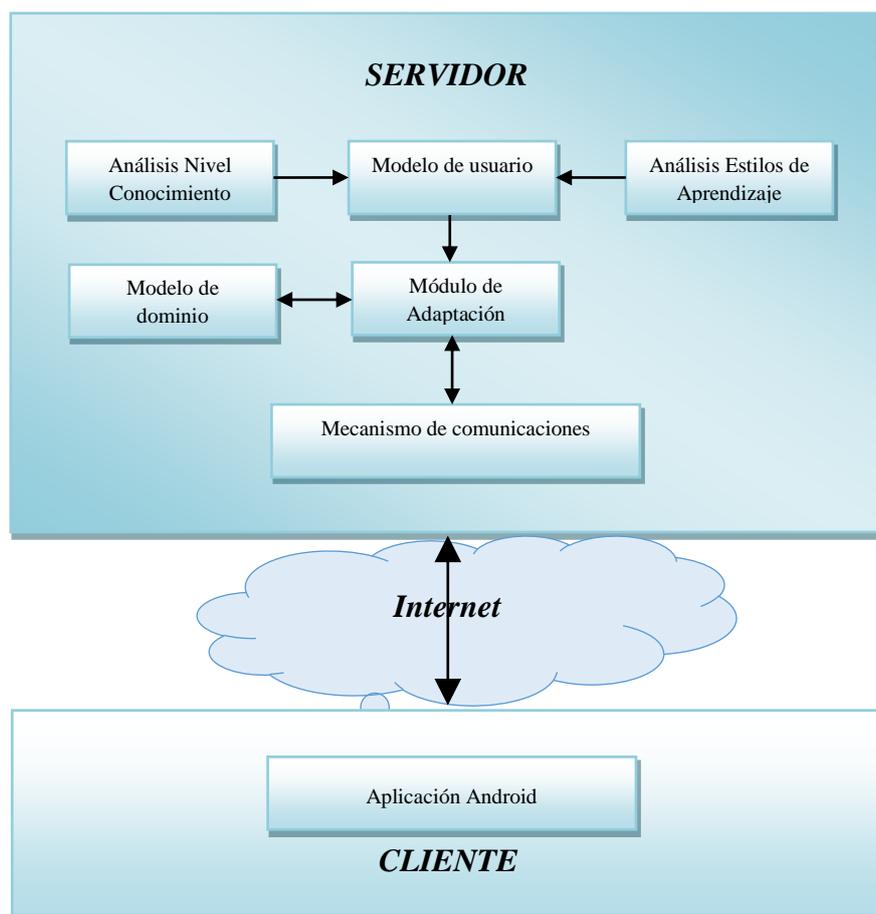


Figura 4.1 – Arquitectura Cliente-Servidor

El *modelo de usuario* contiene toda la información referida a un usuario (estudiante) y puede determinar su nivel de conocimiento y estilo de aprendizaje en base al *módulo de análisis nivel de conocimiento* encargado de recuperar y actualizar el nivel de conocimiento de un usuario y al *módulo de análisis estilos de aprendizaje* que permite determinar, definir y actualizar el estilo de aprendizaje de un usuario. Esta información junto con la información del *modelo de dominio*, encargado de recuperar y mantener información actualizada referida a los puntos de interés, son enviados al *módulo de adaptación*, que aplica diferentes técnicas y métodos de adaptación propias de los Sistemas Hipermedia Adaptativos, generando información personalizada a cada usuario. Finalmente,

ésta información es transmitida al *mecanismo de comunicaciones* y desplegada en la *interfaz de la aplicación* en el dispositivo móvil a través de Internet. Es importante señalar que el principal tipo de adaptación que se realiza en este trabajo es la adaptación de contenido.

IV.3. PILA DEL PRODUCTO (PRODUCT BACKLOG)

A continuación presentamos la Pila del Producto (Product Backlog) en la Tabla 4.1, la cual, a medida que avanzó el proyecto, evolucionó hasta llegar a este nivel.

ID	Historias de Usuarios	Prioridad	Estimación
1	COMO Administrador REQUIERO una aplicación web PARA administrar los parámetros generales del sistema.	99	3
2	COMO Administrador REQUIERO que la aplicación web permita iniciar sesión PARA ingresar al sistema.	95	5
3	COMO Administrador REQUIERO identificar el ingreso de un administrador a la aplicación web PARA para presentar un panel de control que permita gestionar los parámetros del sistema.	93	5
4	COMO Administrador REQUIERO que la aplicación web permita recuperar contraseñas.	90	4
5	COMO Administrador REQUIERO que la aplicación web permita cerrar sesión PARA salir del sistema.	89	4
6	COMO Administrador REQUIERO que la aplicación móvil permita iniciar sesión PARA identificar al usuario en el sistema.	85	9
7	COMO Administrador REQUIERO que la aplicación móvil permita cerrar sesión PARA salir del sistema.	83	6
8	COMO Administrador REQUIERO que la aplicación móvil lea Códigos QR PARA reconocer componentes de red etiquetados con los mismos.	82	10
9	COMO Administrador REQUIERO que la aplicación móvil permita visualizar información generada por el servidor cuando envía un código QR PARA presentar esta información a los estudiantes.	80	6
10	COMO Administrador REQUIERO dar de alta un estudiante PARA que el mismo pueda iniciar sesión en la aplicación móvil.	75	8
11	COMO Administrador REQUIERO dar de baja un estudiante y toda información relacionada PARA eliminarlo del sistema.	73	6
12	COMO Administrador REQUIERO visualizar todos los estudiantes desde un panel de control PARA tener una visión general de todos los usuarios y trabajar con ellos.	71	4
13	COMO Administrador REQUIERO actualizar el perfil de un estudiante	70	8

	PARA modificar o agregar datos relacionados al mismo.		
14	COMO Administrador REQUIERO visualizar los perfiles de los estudiantes PARA trabajar con ellos.	68	4
15	COMO Administrador REQUIERO dar de alta cursos, temas y subtemas PARA asociarlos a cada estudiante.	65	9
16	COMO Administrador REQUIERO dar de baja cursos, temas y subtemas.	64	9
17	COMO Administrador REQUIERO actualizar información relacionada a los cursos, temas y subtemas PARA mantener información actualizada acerca de los mismos.	63	7
18	COMO Administrador REQUIERO visualizar los cursos, temas y subtemas desde el panel de control PARA trabajar con ellos.	61	4
19	COMO Administrador REQUIERO que al dar de alta un estudiante se establezca por defecto el Estilo de Aprendizaje “Activo-Sensitivo” y el Nivel de Conocimiento “Bajo” PARA que el sistema pueda generar inmediatamente información personalizada.	60	10
20	COMO Administrador REQUIERO actualizar los Estilos de Aprendizaje de los estudiantes PARA realizar correcciones o actualizar esta información cuando se tengan datos más precisos acerca del estilo de un estudiante.	58	8
21	COMO Administrador REQUIERO actualizar el Nivel de Conocimiento de cada estudiante con respecto a un subtema PARA que el sistema pueda realizar la adaptación en base a dicho nivel.	57	8
22	COMO Administrador REQUIERO visualizar todos los Estilos de Aprendizaje asociados a los estudiantes PARA trabajar con ellos.	55	4
23	COMO Administrador REQUIERO visualizar todos los Niveles de conocimiento asociados a los estudiantes PARA trabajar con ellos.	54	4
24	COMO Administrador REQUIERO dar de alta componentes de red PARA ser reconocidos desde la Aplicación con el lector de códigos QR.	52	10
25	COMO Administrador REQUIERO dar de baja componentes de red PARA que no estén disponible desde la aplicación móvil.	51	8
26	COMO Administrador REQUIERO actualizar los componentes de red PARA mantener información actualizada acerca de los mismos.	50	10
27	COMO Administrador REQUIERO visualizar los componente de red cargados en la base de datos PARA para trabajar con ellos.	45	4
28	Como Administrador REQUIERO que el sistema implemente técnicas de adaptación PARA generar información personalizada basándose en el estilo de aprendizaje y nivel de conocimiento de un estudiante.	44	9
29	COMO Administrador REQUIERO scripts de recuperación PARA recuperar el sistema en casos de fallos inesperados.	35	8
30	COMO Administrador REQUIERO pruebas finales a la aplicación PARA validar el correcto funcionamiento de la misma.	28	9

31	COMO Administrador REQUIERO la documentación asociada a la aplicación.	20	5
32	COMO Alumno REQUIERO el manual de usuario de la aplicación móvil.	18	4

Tabla 4.1: Product Backlog

IV.4. DEFINICION DE TERMINADO

La Definición de Terminado (Definition of Done – DoD) es el estándar de calidad que debe cumplir el equipo Scrum, es aquello que garantiza que han cumplido lo que se han comprometido y que el “Incremento” del producto es realmente “potencialmente entregable”. A continuación se presenta la Definición de Terminado utilizada en este proyecto:

1. Cada historia de usuario/funcionalidad debe estar codificada, compilada y desplegada.
2. Las historias de usuario/funcionalidades construidas no deben afectar a las funcionalidades ya entregadas.
3. Las historias de usuario/funcionalidades comprometidas deben funcionar en el entorno real.
4. Las historias de usuario/funcionalidades deben contar con pruebas funcionales y el resultado debe ser exitoso.
5. El Código desarrollado debe estar comentado y estandarizado.
6. El Código debe estar en el repositorio de control de versiones.
7. Se deben cumplir los requisitos no funcionales:
 - a. La aplicación web debe ejecutarse sin problemas en las últimas versiones de los navegadores más utilizados entre ellos Mozilla Firefox, Google Chrome e Internet Explorer.
 - b. La aplicación móvil debe funcionar en Smartphones y Tablets con las últimas versiones del Sistema Operativo Android.
 - c. La aplicación móvil debe tener una interfaz elegante, intuitiva y fácil de usar, por lo cual se deben seguir los estándares de diseño de Android.

IV.5. HERRAMIENTAS DE DESARROLLO

Como se mencionó, nuestro trabajo implica del lado del cliente el desarrollo de una aplicación nativa para Android y del lado del servidor una aplicación web. A continuación se presentan las herramientas a ser utilizadas.

Para la parte cliente se hará uso del **SDK de Android** que incluye un conjunto de librerías y herramientas de desarrollo, un depurador de código, un simulador y documentación. El IDE a utilizar es **Eclipse** con el **plugin ADT** (Android Developer Tools). La selección de estas herramientas se basa en las excelentes experiencias con el IDE open source Eclipse en el pasado y debido a que, al momento de comenzar el presente trabajo, la página oficial Android Developers promueve el uso de estas herramientas brindando un paquete listo para comenzar a desarrollar, denominado **ADT bundle**.

Para la parte del servidor tenemos por un lado la programación del *backend* donde se utilizará **CodeIgniter**, un framework de código abierto para el desarrollo de aplicaciones web y sitios web dinámicos con PHP, entre sus ventajas se destacan el uso del patrón de desarrollo Modelo-Vista-Controlador (MVC), la programación orientada a objetos (POO), la curva de aprendizaje es muy corta, es muy ligero y compatible con casi cualquier hosting. Para programar el frontend se usará **Bootstrap**, un framework CSS de software libre para el diseño de sitios y aplicaciones web. La elección de este framework se basa en la facilidad de uso, la potencia para el diseño de todo tipo de páginas y a que posee una gran comunidad, es el proyecto más popular en GitHub y es usado por la NASA y la MSNBC junto a otras organizaciones. Para detalles menores como controles en formularios y solicitudes asíncronas se utilizará **Ajax**, **JSON** y algunas librerías **JavaScript** como **JQuery**.

Sublime Text 3 es un sofisticado editor de código multiplataforma, elegido como herramienta de programación para el desarrollo de la aplicación web por su potencia y amplia documentación. Como software de control de versiones se utilizará **Git**, desarrollado por Linus Torvalds. Finalmente como gestor de base de datos se hará uso de **MySQL**, ya que es la base de datos Open Source número uno del mundo con gran potencia y flexibilidad.

Todo el proyecto será llevado a cabo utilizando herramientas de Software Libre y Open Source. Se utilizará Linux como sistema operativo, específicamente la distribución **Ubuntu**.

IV.6. SPRINT I – Control de Acceso

IV.6.1 Reunión de Planificación del Sprint I

A continuación se responden a las cuestiones tratadas en esta reunión:

❖ *¿Qué puede ser terminado en este Sprint?*

En este Sprint se diseña e implementa el sistema de control de acceso completo, que involucra por un lado el control de acceso en la Aplicación Web y por otro el control de acceso en la Aplicación Móvil, brindando dos funciones claves, la primera permite que solo los administradores tengan acceso a las opciones para gestionar los parámetros generales del sistema desde el servidor y la segunda permite la identificación de los usuarios que inician sesión en la aplicación móvil con el fin de brindarles contenido personalizado en base a su estilos de aprendizaje y nivel de conocimiento. Se implementan las funciones para cerrar sesión tanto en la aplicación móvil como en la aplicación web. En esta reunión además se discute el objetivo del Sprint definido en la siguiente sección. La duración de este Sprint es de cuatro semanas.

❖ *¿Cómo se conseguirá completar el trabajo seleccionado?*

La aplicación web es programada principalmente con PHP, por tal motivo se analizan las librerías de autenticación existentes que nos permiten implementar opciones de seguridad claves como inicio y cierre de sesión, hashing de passwords, prevención de ataques por fuerza bruta y ataques por secuencias de comando (XSS), entre otras funciones. A continuación se procede con la implementación y pruebas del lado del servidor, cuando todo funcione adecuadamente, se procede al desarrollo de la aplicación móvil con la herramienta de Google, el *Android SDK ADT Bundle*, se desarrollan las primeras vistas para el control de acceso que permitan ingresar los datos del usuario y enviarlos al servidor para ser validados a través de JavaScript o Web Services. En esta reunión se discute y crea la Pila del Sprint I, presentada en la sección IV.6.3.

IV.6.2 Objetivo del Sprint I

El objetivo de este Sprint es crear la aplicación web inicial con las funcionalidades de control de acceso y la primera versión de la aplicación móvil con funciones de control de acceso que permita iniciar y cerrar sesión en ambos casos.

IV.6.3 Pila del Sprint I

A continuación se presenta en la Tabla 4.2 la Pila del Sprint I donde se detallan las tareas a realizar.

ID	Historia de Usuario	Tareas	Estimación
1	COMO Administrador REQUIERO una aplicación web PARA administrar los parámetros generales del sistema.	Configurar los framework frontend y backend a utilizar.	3
		Diseño inicial de la página.	
		Implementación y pruebas.	
2	COMO Administrador REQUIERO que la aplicación web permita iniciar sesión PARA ingresar al sistema.	Análisis y selección de la librería de autenticación a utilizar.	5
		Diseñar el formulario de autenticación.	
		Implementar la librería y realizar las modificaciones pertinentes.	
		Pruebas y depuración.	
3	COMO Administrador REQUIERO identificar el ingreso de un administrador a la aplicación web PARA para presentar un panel de control que permita gestionar los parámetros del sistema.	Implementar “acceso admin” en función de la librería o modificarla para cumplir con ello.	5
		Pruebas con diferentes usuarios.	
4	COMO Administrador REQUIERO que la aplicación web permita recuperar contraseñas.	Configurar el formulario para recuperar contraseñas.	4
		Configurar el servidor para el envío de emails.	
		Pruebas y depuración.	
5	COMO Administrador REQUIERO que la aplicación web permita cerrar sesión PARA salir del sistema.	Crear una barra superior con la opción para cerrar sesión.	4
		Implementar función para salir del sistema.	

		Pruebas analizando cookies.	
6	COMO Administrador REQUIERO que la aplicación móvil permita iniciar sesión PARA identificar al usuario en el sistema.	Bosquejo inicial de la aplicación. Diseñar vistas para el control de acceso. Implementar conexión con el servidor y autenticación de datos. Pruebas y depuración.	9
7	COMO Administrador REQUIERO que la aplicación móvil permita cerrar sesión PARA salir del sistema.	Diseñar e implementar función para cerrar sesión en la aplicación móvil. Realizar pruebas con el servidor.	6

Tabla 4.2: Sprint Backlog - Sprint I

IV.6.4 Incremento

El incremento del producto es por un lado una versión inicial de la aplicación móvil que presenta una pantalla de inicio de sesión para validar el usuario en el servidor, esto permite determinar exactamente que estudiante está usando la aplicación para que a futuro se pueda realizar la adaptación de la información al mismo. Por otro lado tenemos la aplicación web inicial con control de acceso que a futuro permitirá la gestión global del sistema por parte de los administradores (ver Anexo A para detalles).

IV.6.5 Revisión y Retrospectiva del Sprint I

Se presentó el incremento del producto que en esta etapa es una aplicación móvil básica que permite a los usuarios ingresar y ser autenticados por el servidor. Por otro lado se implementó una aplicación web inicial con control de acceso que permite al administrador iniciar y cerrar sesión. Los problemas surgidos en este Sprint fueron principalmente relacionados con la codificación de ciertos métodos en Android para la validación de los usuarios en el servidor, pero fueron solucionados satisfactoriamente.

Se realizó un ajuste general a la Pila del Producto para mostrar que elementos se han terminado y se analizaron cuales ítems podrían ser abordados en el próximo Sprint.

Con respecto a los procesos, personas, relaciones y herramientas no hubo cambios mayores.

IV.7. SPRINT II – Lectura de Códigos QR

IV.7.1 Reunión de Planificación del Sprint II

A continuación se responden a las cuestiones tratadas en esta reunión:

❖ *¿Qué puede ser terminado en este Sprint?*

En este Sprint se implementa el módulo para lectura de códigos QR en la Aplicación móvil y se integra una nueva vista (*View*) con un contenedor web nativo de Android (*WebView*) donde se podrá visualizar la respuesta del servidor (un documento HTML) cuando se envíe la información de un código QR. Este módulo es clave para nuestra aplicación. En esta reunión además se discute y define el objetivo de este Sprint presentado en la siguiente sección. La duración de este Sprint es de cuatro semanas.

❖ *¿Cómo se conseguirá completar el trabajo seleccionado?*

Se trabaja con Android del lado del cliente por tal motivo se realiza un estudio de las librerías Android disponibles para la lectura de código QR y los distintos tipos de implementación que se pueden realizar, una vez identificada la librería que mejor se adapta a nuestros requisitos, se realiza la integración y respectiva programación de funciones para el correcto funcionamiento del módulo. En esta reunión también se discute y se crea la Pila del Sprint II que se presenta en la sección IV.7.3.

IV.7.2 Objetivo del Sprint II

El objetivo del Sprint es implementar el módulo de lectura de Códigos QR e integrar un navegador web nativo en la aplicación móvil para visualizar la respuesta del servidor.

IV.7.3 Pila del Sprint II

A continuación se presenta en la Tabla 4.3 la Pila del Sprint II donde se detallan las tareas a realizar.

ID	Historias de Usuario	Tareas	Estimación
1	COMO Administrador REQUIERO que la aplicación móvil lea Códigos QR PARA reconocer componentes de red etiquetados con los mismos.	Análisis de librerías y selección de la más apropiada.	10
		Integración de la librería al código de la aplicación móvil.	
		Pruebas y depuración.	
2	COMO Administrador REQUIERO que la aplicación móvil permita visualizar información generada por el servidor cuando envía un código QR PARA presentar esta información a los estudiantes.	Análisis y diseño de las vistas.	6
		Implementar el contenedor web nativo en la aplicación móvil.	
		Pruebas con el servidor y depuración.	

Tabla 4.3: Sprint Backlog - Sprint II

IV.7.4 Incremento

El incremento del producto es la aplicación móvil con el escáner de códigos QR integrado y con un navegador web nativo embebido donde se visualizará la respuesta del servidor, lo que en futuros Sprint permitirá presentar la información personalizada generada por el módulo de adaptación.

IV.7.5 Revisión y Retrospectiva del Sprint II

Se presentó el incremento del producto que en esta etapa permite a los usuarios leer códigos QR y visualizar la respuesta del servidor. Los problemas surgidos en este Sprint fueron principalmente relacionados a la integración de la librería Zxing a nuestra aplicación móvil, pero fueron solucionados satisfactoriamente.

Se realizó un ajuste general a la Pila del Producto para mostrar que elementos se han terminado y se analizaron cuales ítem podrían ser abordados en el próximo Sprint.

Con respecto a los procesos, personas, relaciones y herramientas no hubo cambios mayores.

IV.8. SPRINT III – Administración general del Sistema

IV.8.1 Reunión de Planificación del Sprint III

A continuación se responden a las cuestiones tratadas en esta reunión:

❖ *¿Qué puede ser terminado en este Sprint?*

En este Sprint se diseña y desarrolla el sistema de administración que consiste en un panel de control en la Aplicación Web que permitirá la gestión global del sistema por parte del administrador (docente) incluyendo la gestión de usuarios, perfiles, cursos, temas y subtemas. En esta reunión además se discute y define el objetivo de este Sprint presentado en la siguiente sección. La duración de este Sprint es de cuatro semanas.

❖ *¿Cómo se conseguirá completar el trabajo seleccionado?*

El trabajo se reparte en tareas y luego se diseñan los módulos, en primer momento se realiza el diseño del panel de control con una barra lateral de navegación que contendrá las opciones de administración para gestionar usuarios, perfiles, cursos, temas y subtemas, cada uno de estos hacen llamadas a determinadas funciones que permiten las altas, bajas y modificaciones en la base de datos. En esta reunión también se discute y crea la Pila del Sprint III que se presenta en la sección IV.8.3.

IV.8.2 Objetivo del Sprint III

El objetivo de este Sprint es el desarrollo del panel de control en la aplicación web con las primeras opciones para gestionar los usuarios, perfiles, curso, temas y subtemas en la base de datos.

IV.8.3 Pila del Sprint III

A continuación se presenta en la Tabla 4.4 la Pila del Sprint III donde se detallan las tareas a realizar.

ID	Historias de Usuario	Tareas	Estimación
1	COMO Administrador REQUIERO dar de alta un estudiante PARA que el mismo pueda iniciar sesión en la aplicación móvil.	Análisis y diseño de la nueva función.	8
		Análisis y actualización del modelo de datos.	
		Implementar la función.	
		Pruebas y depuración.	
2	COMO Administrador REQUIERO dar de baja un estudiante y toda información relacionada PARA eliminarlo del sistema.	Análisis y diseño de la nueva función	6
		Análisis de repercusión en el modelo de datos, controles de integridad referencial.	
		Implementar la función.	
		Pruebas y depuración.	
3	COMO Administrador REQUIERO visualizar todos los estudiantes desde un panel de control PARA tener una visión general de los mismos y trabajar con ellos.	Análisis y diseño de la nueva función	4
		Implementación.	
		Pruebas y depuración.	
4	COMO Administrador REQUIERO actualizar el perfil de un estudiante PARA modificar o agregar datos relacionados al mismo.	Análisis y diseño de la nueva función	8
		Análisis y actualización del modelo de datos y controles de integridad referencial.	
		Implementar la función	
		Pruebas y depuración.	
5	COMO Administrador REQUIERO visualizar los perfiles de los estudiantes PARA trabajar con ellos.	Análisis y diseño de la nueva función	4
		Implementar la función.	
		Pruebas y depuración.	
6	COMO Administrador REQUIERO dar de alta cursos, temas y subtemas PARA asociarlos a cada estudiante.	Análisis y diseño de las nuevas funciones.	9
		Análisis y actualización del modelo de datos.	
		Implementar las funciones.	
		Pruebas y depuración.	

7	COMO Administrador REQUIERO dar de baja cursos, temas y subtemas.	Análisis y diseño de las nuevas funciones.	9
		Análisis de repercusión en el modelo de datos y controles de integridad referencial.	
		Implementar las funciones.	
		Pruebas y depuración.	
8	COMO Administrador REQUIERO actualizar información relacionada a los cursos, temas y subtemas PARA mantener información actualizada acerca de los mismos.	Análisis y diseño de las nuevas funciones.	7
		Análisis y actualización del modelo de datos y controles de integridad referencial.	
		Implementar las funciones.	
		Pruebas y depuración.	
9	COMO Administrador REQUIERO visualizar los cursos, temas y subtemas desde el panel de control PARA trabajar con ellos.	Análisis y diseño de las nuevas funciones.	4
		Implementar las funciones.	
		Pruebas y depuración.	

Tabla 4.4: Sprint Backlog - Sprint III

IV.8.4 Incremento

El incremento del producto es la aplicación web con la primera versión del panel de control para los administradores que permite la gestión de usuarios, perfiles, cursos, temas y subtemas (ver Anexo A para más detalles).

IV.8.5 Revisión y Retrospectiva del Sprint III

En esta reunión se presenta el incremento del producto, el mismo permite gestionar usuarios, perfiles y demás parámetros del sistema. Los problemas afrontados en este Sprint fueron menores, la mayoría relacionados con la configuración del framework CodeIgniter y la codificación en JavaScript, JQuery y PHP.

Se realizó un ajuste general a la Pila del Producto para mostrar que elementos se han terminado y se analizaron cuales ítem podrían ser abordados en el próximo Sprint.

Con respecto a los procesos, personas, relaciones y herramientas no hubo cambios.

IV.9. SPRINT IV – Módulo de Adaptación

IV.9.1 Reunión de Planificación del Sprint IV

A continuación se responden a las cuestiones tratadas en esta reunión:

❖ *¿Qué puede ser terminado en este Sprint?*

En este Sprint se diseña y desarrolla uno de los módulos claves del sistema, nos referimos al módulo de adaptación, el mismo permite generar información personalizada basándose en el estilo de aprendizaje y el nivel de conocimiento de un estudiante. Por otro lado se mejora la función alta de usuario permitiendo agregar automáticamente el nivel de conocimiento predeterminado (bajo) y el estilo de aprendizaje por defecto sensitivo-visual ya que predomina más que el intuitivo-verbal según el trabajo de Kinshuk (2004), luego esta información puede ser actualizada por el administrador. También se agrega al panel de control opciones para gestionar los componentes, los niveles de conocimiento y los estilos de aprendizaje. En esta reunión se discute y define el objetivo de este Sprint presentado en la siguiente sección. La duración del Sprint establecida es de cuatro semanas.

❖ *¿Cómo se conseguirá completar el trabajo seleccionado?*

En un principio se diseñan e implementan las funciones administrativas que permitan gestionar los estilos de aprendizaje, niveles de conocimiento y componentes de red en la base de datos, para luego diseñar y desarrollar el módulo de adaptación, el cual implementa los métodos y técnicas de adaptación utilizados en los Sistemas Hipermedia Adaptativos. Los métodos de adaptación utilizados en este trabajo fueron el método de las *explicaciones adicionales* y el método de las *variantes* y las técnicas implementadas fueron la técnica de *texto condicional*, la técnica de *texto expandible* y la técnica de *variante de páginas*. En esta reunión se discute y se crea la Pila del Sprint IV que se presenta en la sección IV.9.3.

IV.9.2 Objetivo del Sprint IV

El objetivo de este Sprint es desarrollar el módulo de adaptación y añadir nuevas opciones administrativas al panel de control para gestionar los componentes, los estilos de aprendizaje y los niveles de conocimientos asociados a cada estudiante.

IV.9.3 Pila del Sprint IV

A continuación se presenta en la Tabla 4.5 la Pila del Sprint IV donde se detallan las tareas a realizar.

ID	Historias de Usuario	Tareas	Estimación
1	COMO Administrador REQUIERO que al dar de alta un estudiante se establezca por defecto el Estilo de Aprendizaje “Activo-Sensitivo” y el Nivel de Conocimiento “Bajo” PARA que el sistema pueda generar inmediatamente información personalizada.	Análisis y diseño de las nuevas funciones de carga automática. Análisis y actualización del modelo de datos. Controles de integridad referencial. Implementación de las funciones. Pruebas y depuración.	10
2	COMO Administrador REQUIERO actualizar los Estilos de Aprendizaje de los estudiantes PARA realizar correcciones o actualizar esta información cuando se tengan datos más precisos acerca del estilo de aprendizaje de un estudiante.	Análisis y diseño de la nueva función. Actualización del modelo de datos. Controles de integridad referencial. Implementación de la función Pruebas y depuración.	8
3	COMO Administrador REQUIERO actualizar el Nivel de Conocimiento de cada estudiante con respecto a un subtema PARA que el sistema pueda realizar la adaptación en base a dicho nivel.	Análisis y diseño de la nueva función. Actualización del modelo de datos. Controles de integridad referencial. Implementación de la función. Pruebas y depuración.	8
4	COMO Administrador REQUIERO visualizar todos los Estilos de Aprendizaje asociados a los estudiantes PARA trabajar con ellos.	Análisis y diseño de la función. Implementación de la función. Pruebas y depuración.	4
5	COMO Administrador REQUIERO visualizar todos	Análisis y diseño de la nueva	4

	los Niveles de conocimiento asociados a los estudiantes PARA trabajar con ellos.	función. Implementación de la función. Pruebas y depuración.	
6	COMO Administrador REQUIERO dar de alta componentes de red PARA ser reconocidos desde la Aplicación con el lector de códigos QR.	Análisis y diseño de la nueva función. Análisis y actualización del modelo de datos. Controles de integridad referencial. Implementación de la función. Pruebas y depuración.	10
7	COMO Administrador REQUIERO dar de baja componentes de red PARA que no estén disponible desde la aplicación móvil.	Análisis y diseño de la función. Análisis de repercusión en el modelo de datos. Controles de integridad referencial al eliminar. Implementación de la función Pruebas y depuración.	8
8	COMO Administrador REQUIERO actualizar los componentes de red PARA mantener información actualizada acerca de los mismos.	Análisis y diseño de la nueva función. Análisis y actualización del modelo de datos. Implementación de la función. Pruebas y depuración.	10
9	COMO Administrador REQUIERO visualizar los componente de red cargados en la base de datos PARA para trabajar con ellos.	Análisis y diseño de la nueva función. Implementación de la función. Pruebas y depuración.	4
10	Como Administrador REQUIERO que el sistema implemente técnicas de adaptación PARA generar información personalizada basándose en el estilo de aprendizaje y nivel de conocimiento de un estudiante.	Análisis y diseño del nuevo módulo. Implementación del módulo de adaptación. Pruebas unitarias y de integración.	10

Tabla 4.5: Sprint Backlog - Sprint IV

IV.9.4 Incremento

El incremento del producto amplía las funciones del panel de control de la aplicación web permitiendo la gestión de los componentes, los estilos de aprendizaje y los

niveles de conocimiento de los estudiantes. Además se implementa el módulo de adaptación que permite generar información personalizada en base al estilo de aprendizaje y nivel de conocimiento del estudiante que ha iniciado sesión en la aplicación, permitiendo desplegar dicha información en su dispositivo móvil.

A continuación se presenta la estructura del módulo de adaptación y se describe el funcionamiento del mismo.

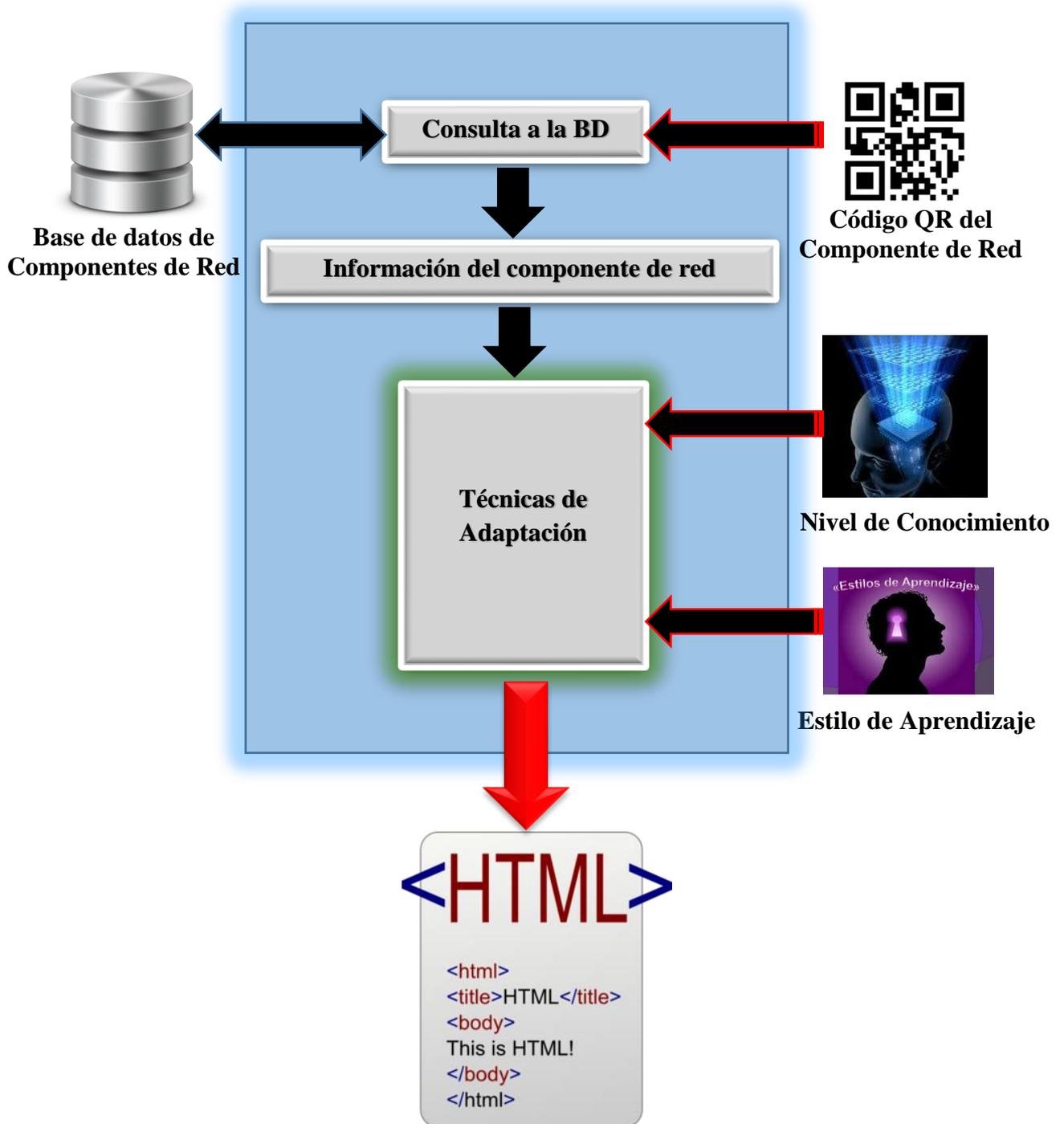


Figura 4.2 - Esquema general del módulo de adaptación.

En la Figura 4.2 podemos apreciar de manera general la estructura del módulo de adaptación. Como entrada el módulo recibe el nivel de conocimiento y estilo de aprendizaje del usuario que ha iniciado sesión en la aplicación móvil, además recibe la información de un código QR (por ejemplo 5476731478) con la cual realiza una consulta a la base de datos y recupera la información asociada a un determinado componente de red. Una vez que el módulo cuenta con la información necesaria aplica determinadas técnicas

de adaptación y genera un documento HTML que será desplegado en el dispositivo móvil. A continuación se presenta la estructura del documento y las técnicas de adaptación implementadas.

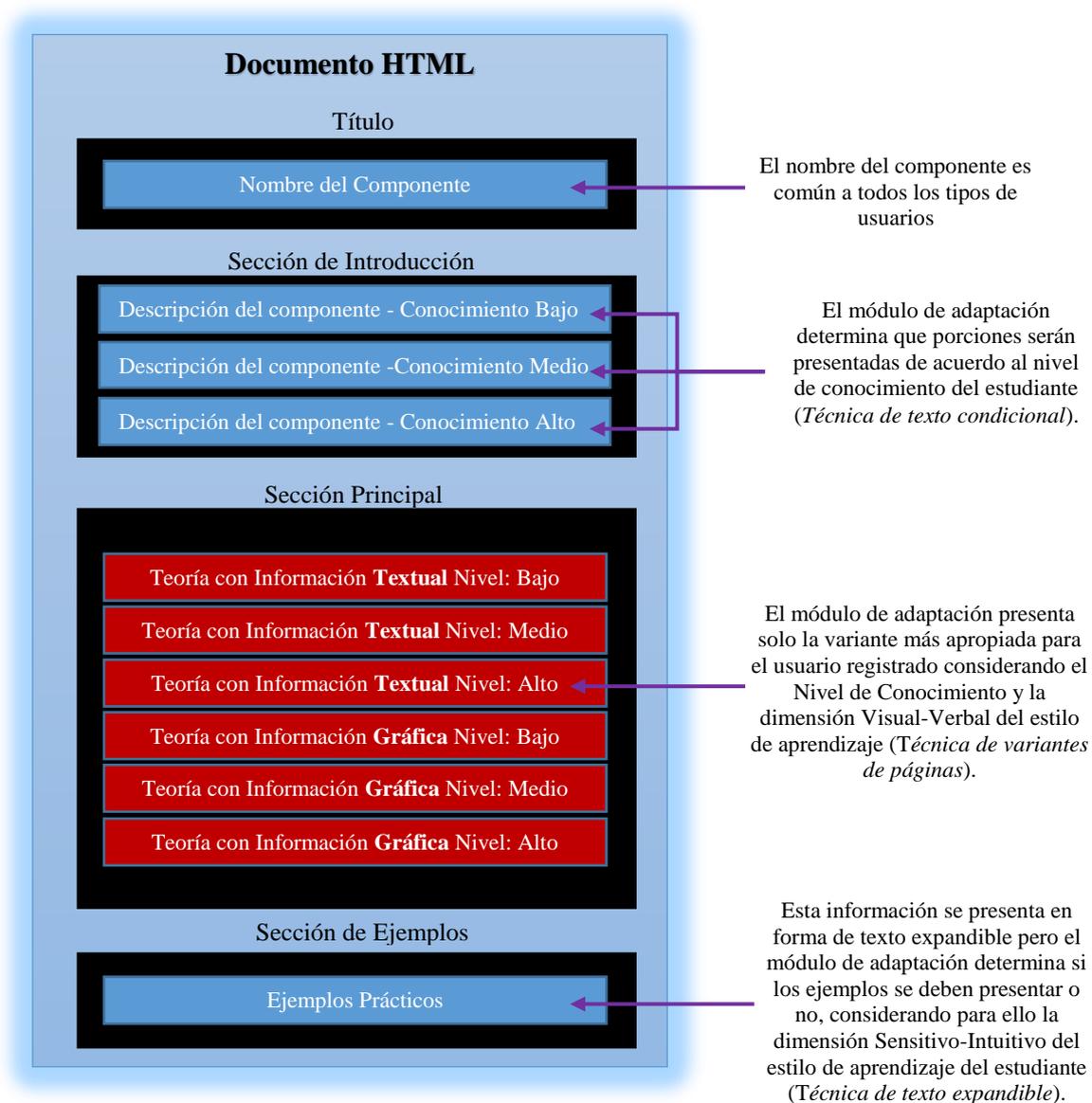


Figura 4.3 – Estructura del documento HTML y Técnicas de Adaptación implementadas.

El módulo utiliza la técnica de texto condicional para determinar que porciones de texto relacionadas a la descripción del componente debe presentar. Estas porciones se complementan entre si y el módulo determina cuáles satisfacen la condición del nivel de conocimiento del usuario registrado. Para un usuario con nivel de conocimiento bajo, solo presentará la porción de texto para el nivel bajo. Para un usuario con conocimiento medio presentará la porción del nivel bajo pero la complementa con la porción del nivel medio. Finalmente un alumno con conocimiento alto obtendría las tres porciones de texto para

todos los niveles, obteniendo así una mayor cantidad de detalles en la sección de introducción.

Para aplicar la técnica de variantes de páginas, el módulo cuenta con 6 variaciones con distintas presentaciones del mismo contenido, que resultan de combinar las dos dimensiones visual-verbal y los tres niveles de conocimiento bajo, medio y alto. Cada variación es adecuada para uno de los siguientes estereotipos de usuarios:

- ✓ Usuario con Estilo Visual– Conocimiento Bajo.
- ✓ Usuario con Estilo Visual – Conocimiento Medio.
- ✓ Usuario con Estilo Visual – Conocimiento Alto.
- ✓ Usuario con Estilo Verbal – Conocimiento Bajo.
- ✓ Usuario con Estilo Verbal – Conocimiento Medio
- ✓ Usuario con Estilo Verbal – Conocimiento Alto.

Al momento de generar el documento el sistema selecciona la variante más apropiada. Es decir, para aquellos usuarios con estilo verbal se presenta la variante con información puramente textual, por el contrario para los usuarios con estilo visual se presenta la variante con información gráfica, siempre considerando además el nivel de conocimiento del estudiante que ha iniciado sesión en la aplicación móvil.

Por otro lado el módulo considera la dimensión sensitivo-intuitivo para decidir si el documento generado presentará ejemplos prácticos o no. Los estudiantes sensitivos siempre prefieren tener ejemplos, ya que intentan relacionar la teoría con hechos concretos y prácticos. Para estos usuarios el módulo incluirá la sección de ejemplos con la técnica de texto expandible. Por el contrario para los estudiantes intuitivos esta sección estará oculta o mejor dicho no será incluida en el documento presentado, ya que este tipo de usuarios prefieren principios, conceptos teóricos e información abstracta, la cual obtienen en la sección principal mediante la técnica de variante de páginas.

Finalmente cuando el módulo termina de aplicar cada una de las técnicas de adaptación, genera un documento HTML que será presentado en la aplicación móvil del estudiante. (Ver sección V.3.3 *Pruebas del módulo de adaptación en la Aplicación*).

IV.9.5 Revisión y Retrospectiva del Sprint IV

En esta reunión se presenta el incremento del producto que como dijimos resulta en la aplicación web con el panel de control ampliado y en la implementación del módulo de adaptación. Los problemas afrontados en este Sprint fueron menores y estuvieron relacionados con la implementación de algunas técnicas de adaptación.

Se realizó un ajuste general a la Pila del Producto para mostrar que elementos se han terminado y se analizaron los ítems a ser abordados en el próximo Sprint.

Con respecto a los procesos, personas, relaciones y herramientas no hubo cambios mayores.

IV.10. SPRINT V – Release

IV.10.1 Reunión de Planificación del Sprint V

A continuación se responden a las cuestiones tratadas en esta reunión:

❖ *¿Qué puede ser terminado en este Sprint?*

En este Sprint se implementan las últimas tareas necesarias para el despliegue de la aplicación, entre ellas la generación de scripts de recuperación del sistema en caso de fallos, tareas relacionadas con las bases de datos en producción, documentación y pruebas finales. Además, en esta reunión se discute y define el objetivo del Sprint presentado en la siguiente sección.

❖ *¿Cómo se conseguirá completar el trabajo seleccionado?*

Por un lado se sube la versión final de la aplicación web al hosting y se modifican archivos de configuración en el framework para que la misma funcione correctamente a través de internet. Por otra parte, se instala la aplicación móvil en los dispositivos requeridos (Tablets y Smartphones). Además, se generan los scripts de recuperación y se realizan las últimas pruebas, entre ellas las pruebas de carga haciendo peticiones simultáneas con diferentes dispositivos móviles y con herramientas como ApacheBench para simular una gran cantidad de usuarios realizando peticiones simultáneamente. Finalmente se genera la documentación. En esta reunión también se discute y se crea la Pila del Sprint V que se presenta en la sección IV.10.3.

IV.10.2 Objetivo del Sprint V

El objetivo de este Sprint es ejecutar las pruebas finales, generar la documentación y desplegar la aplicación.

IV.10.3 Pila del Sprint V

A continuación se presenta en la Tabla 4.6 la Pila del Sprint V donde se detallan las tareas a realizar.

ID	Historias de Usuario	Tareas	Estimación
1	COMO Administrador REQUIERO scripts de recuperación PARA recuperar el sistema en casos de fallos inesperados.	Análisis de posibles fallos.	8
		Crear Scripts de recuperación.	
		Ejecutar pruebas.	
2	COMO Administrador REQUIERO pruebas finales a la aplicación PARA validar el correcto funcionamiento de la misma.	Análisis y diseño de pruebas.	9
		Ejecutar pruebas con el servidor.	
		Análisis de resultados y correcciones.	
3	COMO Administrador REQUIERO la documentación asociada a la aplicación.	Generar la documentación.	5
		Control y corrección de errores.	
4	COMO Alumno REQUIERO el manual de usuario de la aplicación móvil.	Redactar manual de usuario.	5
		Chequeo y corrección de errores.	

Tabla 4.6: Sprint Backlog - Sprint V

IV.10.4 Incremento

El incremento del producto es la aplicación final probada y funcionando correctamente, con la documentación correspondiente.

IV.10.5 Revisión y Retrospectiva del Sprint V

En esta reunión se presenta el producto final, la aplicación móvil que puede ser utilizada en el entorno real para la cual fue diseñada, en este caso el Laboratorio de Informática de la Universidad Nacional de Santiago del Estero. Los problemas afrontados en este Sprint estuvieron relacionados con la configuración de parámetros en el hosting y con el cacheo de páginas que producían retardos y anomalías en la aplicación móvil. Finalmente fueron solucionados y se completaron todos los ítems de la Pila del Producto.

CAPÍTULO V

PRUEBAS Y ANALISIS DE RESULTADOS

V.1. INTRODUCCION

En este capítulo se presenta un resumen de las pruebas realizadas tanto a la aplicación Android (Cliente) como a la aplicación web (Servidor) para garantizar un nivel de calidad que permita a los usuarios tener una experiencia agradable y fluida cuando utilicen la aplicación en sus dispositivos móviles, siempre con el objetivo de brindarles información personalizada acerca de un determinado punto de interés en base a sus estilos de aprendizaje y niveles de conocimiento.

V.2. PRUEBAS A LA APLICACIÓN WEB

Esta sección presenta un conjunto de pruebas que permiten entre otras cosas validar la forma en que se despliegan las páginas de nuestro sitio con el fin de determinar si cumplen con los estándares web. Se realizan pruebas relacionadas con la interfaz, la compatibilidad y el rendimiento, entre otras. Con ellas se espera asegurar un buen nivel de desempeño y calidad por parte de la aplicación web, ya sea cuando es utilizada por los administradores (docentes) o cuando interactúa con la aplicación móvil para presentar la información generada por el módulo de adaptación.

A continuación se presentan las pruebas realizadas.

V.2.1 Pruebas de Contenido

Esta prueba se realizó a través de un recorrido exhaustivo en la aplicación web poniendo énfasis en detectar tres tipos de errores: errores sintácticos (errores tipográficos o gramaticales); errores semánticos (errores en la precisión y completitud de la información) y errores en la estructura del contenido que se presenta al usuario.

Se puso especial énfasis en chequear que la información generada por la Aplicación Web y presentada a los usuarios en sus dispositivos móviles sea precisa, concisa y puntual.

En esta etapa se detectaron errores mínimos, sobre todo sintácticos y semánticos que fueron corregidos de manera satisfactoria.

V.2.2 Pruebas de Interfaz

La prueba de interfaz ejercita los mecanismos de interacción y valida los aspectos estéticos de la interfaz de usuario.

V.2.2.1 Prueba de Mecanismos de la Interfaz

En esta etapa se probaron los siguientes recursos:

- ✓ *Enlaces*: se chequeó cada vínculo de navegación para garantizar que se alcanza el recurso o función apropiados. Se utilizó el servicio de validación de enlaces del W3C *Check Link*, con el cual se detectaron algunos enlaces rotos y fueron satisfactoriamente corregidos.

A continuación se presentan algunos resultados.

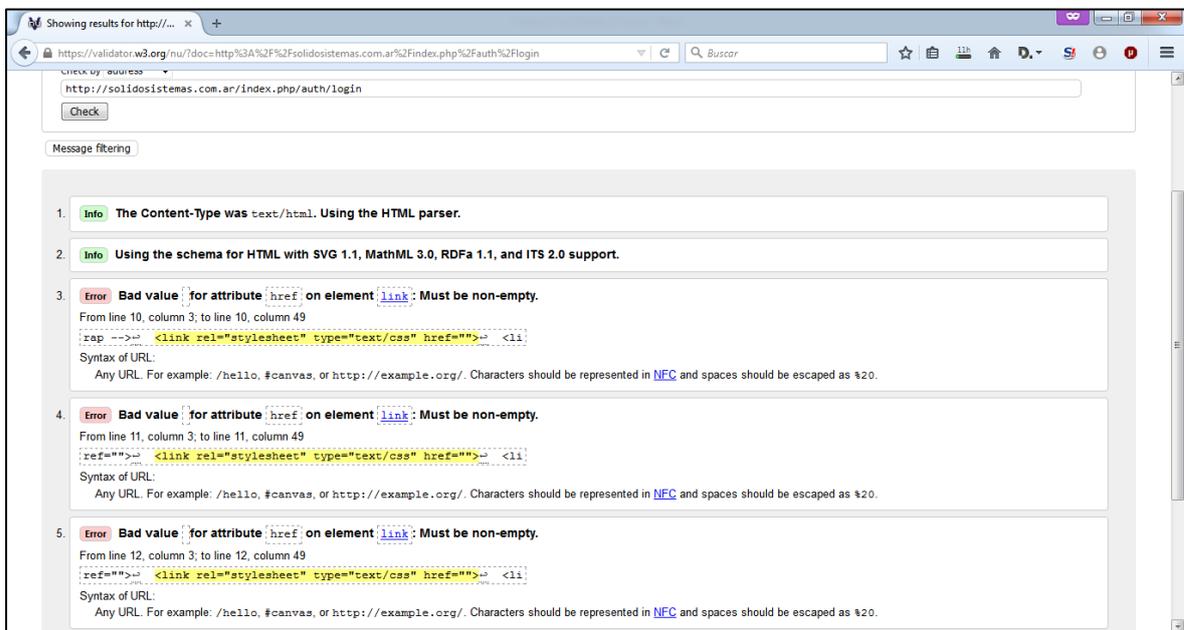


Figura 5.1 - Detección de enlaces rotos con Check Link.

Una vez realizadas las correcciones volvemos a ejecutar la prueba y obtenemos un resultado exitoso, nuestra aplicación web no contiene enlaces rotos.

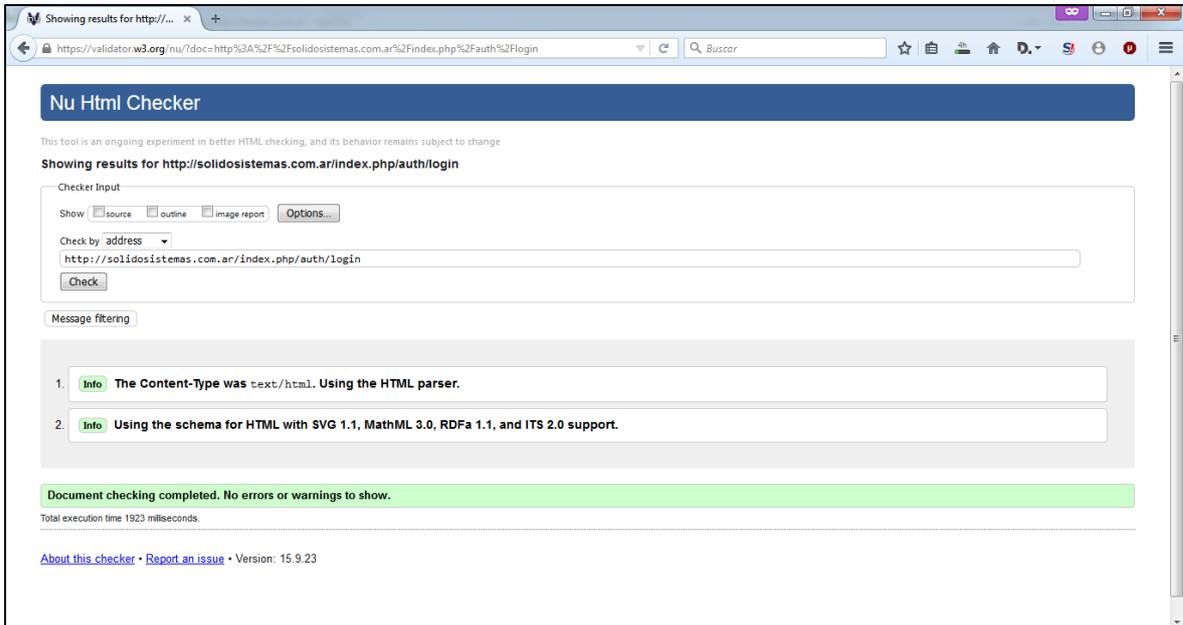


Figura 5.2 - Pruebas luego de realizar las correcciones pertinentes.

- ✓ *Formularios*: se corroboró que las etiquetas identifican correctamente los campos dentro de los formularios y que los campos obligatorios son reconocibles. Además, se verificó que toda la información cargada en los formularios es recibida por el servidor y ningún dato se pierde en la transmisión. Se chequeo que se usan valores por defecto cuando el usuario no selecciona de un menú desplegable y que las funciones JavaScript que realizan la comprobación de errores en los datos ingresados funcionan de manera adecuada.

A continuación se presenta una prueba ejecutando los controles en los formularios con un plugin JavaScript.

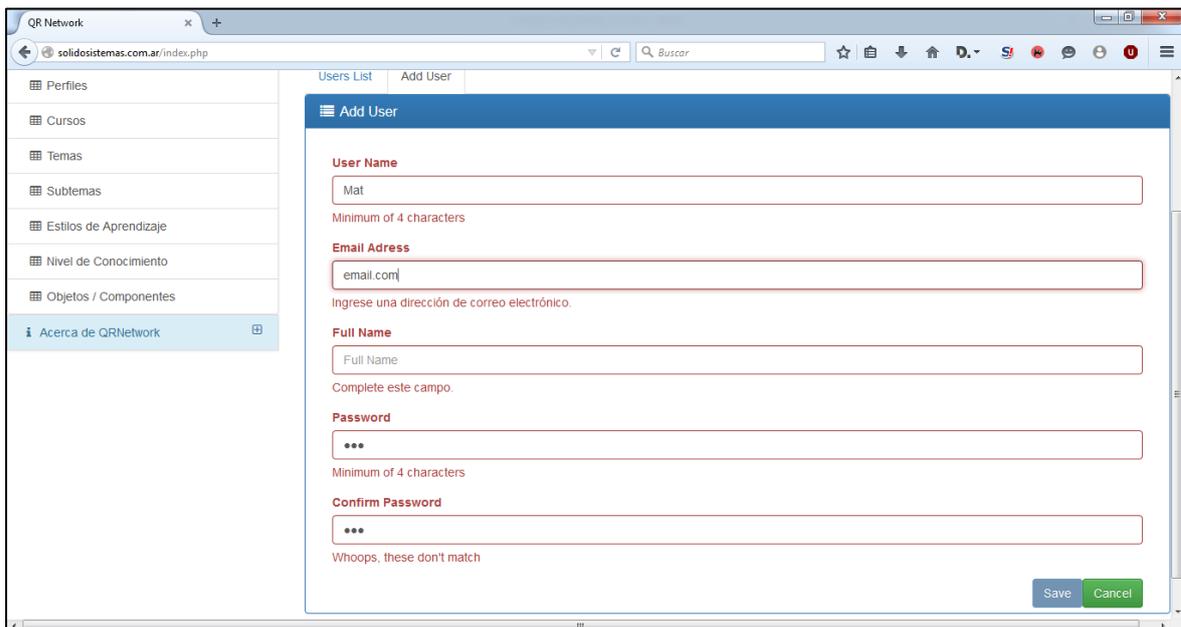


Figura 5.3 - Probando controles en el ingreso de datos del formulario alta de usuario.

- ✓ *Scripts del lado del Cliente:* se comprobó que el código JavaScript del lado del cliente se ejecuta correctamente y que los siguientes plugins basados en JQuery y JavaScript funcionan adecuadamente; DataTables Plugin para presentar tablas en forma refinada; Chosen Plugin para presentar menús desplegables (Dropdowns) con opciones de búsqueda; Validator Plugin para hacer controles de los datos de entrada en formularios, Slider Plugin utilizado en nuestro caso para las actualizaciones en las escalas de los estilos de aprendizaje; Summernote Plugin un potente editor WYSIWYG para los formularios. Cada uno de éstos fue chequeado a profundidad y se encontraron errores mínimos de configuración que fueron fácilmente solucionados.
- ✓ *HTML dinámico:* se desplegó cada página que contiene HTML dinámico y corroboró que cada despliegue es correcto. Las pruebas fueron realizadas en varias versiones de los navegadores Internet Explorer, Mozilla Firefox, Google Chrome y Opera y se presentan en la siguiente sección.
- ✓ *Cookies:* se comprobó que las cookies son construidas con los datos correctos y se transmiten adecuadamente al lado cliente, también se chequeo la persistencia para asegurar que las cookies expiran en la fecha adecuada. Se chequeó del lado del

servidor que las cookies se ligan adecuadamente cuando se realiza una solicitud al mismo.

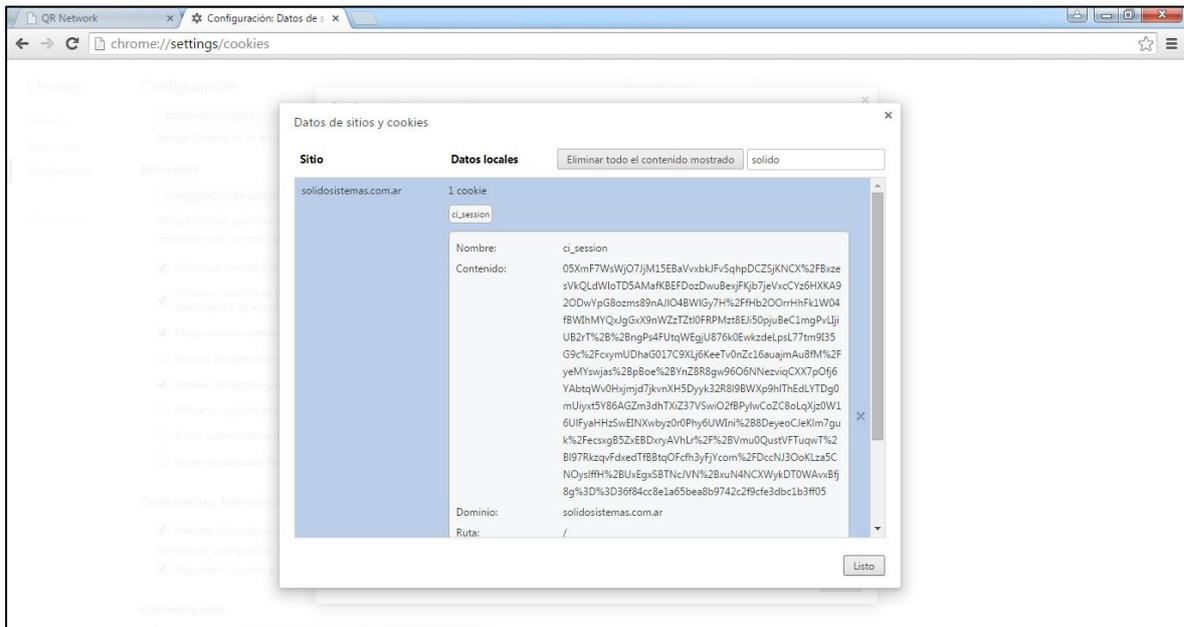


Figura 5.4 - Controles de cookies enviadas al cliente.

V.2.2.2 Pruebas de Compatibilidad

Diferentes computadoras, dispositivos de despliegue, sistemas operativos, navegadores y velocidades de conexión de red pueden tener influencia significativa sobre la operación de la Aplicación Web, por tal motivo en esta etapa se buscó descubrir problemas de compatibilidad antes que el sistema esté en línea.

A continuación se presenta el conjunto de configuraciones y sus variantes sobre las cuales fueron realizadas las pruebas.

Dispositivos	S.O.	Navegador Web	Resultado
Notebooks (15.6") Netbooks (10.1") PC de Escritorio	Windows 7 x86 Windows 7 x64	Internet Explorer 8.0.7601 & 9.0.8112	OK
		Mozilla Firefox 40.0.3 & 41.0	OK
		Google Chrome 45.0.2454.85	OK
		Opera 31.0	OK
	Linux Kernel 2.6	Mozilla Firefox 39.0.3	OK
		Google Chrome 27.0.1453.110	OK
		Epiphany 2.30.2	OK
Tablet	Android 4.1 Jelly Bean	Mozilla Firefox 26.0.1	OK
		Google Chrome 45.0.2454.94	OK
		Opera 32.0.1953	OK

Tabla 5.1 - Configuraciones sobre las que se realizaron las pruebas de la Aplicación Web.

V.2.3 Prueba de Navegación

La primera fase de la prueba de navegación en realidad comenzó durante la prueba de interfaz, en esta etapa se prueban los siguientes mecanismos de navegación:

- ✓ *Vínculos de navegación*: se hizo un chequeo de los enlaces de navegación para asegurar que todos alcanzan el contenido o funcionalidad adecuados.
- ✓ *Redirecciones*: se controló que las redirecciones sean las adecuadas cuando un usuario solicita una URL.
- ✓ *Marcas de página (favoritos, bookmarks)*: se comprobó que el sitio web garantiza la extracción de un título de página significativo cuando se crea una marca.
- ✓ *Marcos y framesets*: se probó que los frames se despliegan adecuadamente, la plantilla y tamaño son adecuados y el rendimiento de descarga permiten una experiencia fluida al utilizar la aplicación.
- ✓ *Mapas de sitio*: nuestro sistema cuenta con una barra lateral de navegación o *sidebar* que contiene los enlaces a las principales secciones del sitio para realizar la administración general del sistema de manera fácil y cómoda.

V.2.4 Prueba de Rendimiento

En esta etapa utilizamos una potente herramienta de Apache Software Foundation, denominada *Apache HTTP server benchmarking tool* que permite medir el rendimiento de un servidor generando peticiones con el nivel de concurrencia que indiquemos. En nuestro caso tomamos como punto de partida la cantidad aproximada de alumnos que utilizarán la aplicación en el entorno real, 35 alumnos (concurrencia), por otro lado cada usuario requerirá aproximadamente 17 solicitudes al servidor (peticiones aproximadas para solicitar todos los recursos de una página como imágenes, texto, archivos JS y CSS) para obtener una página HTML completa con información personalizada en su dispositivo móvil, por tal motivo se simulan 600 peticiones a nuestro servidor distribuidas en 35 hilos que representan la cantidad de usuarios concurrentes (cantidad de alumnos en la materia redes I), suponiendo que nuestro servidor va a trabajar bajo esa presión de carga. Precisamente esta capacidad de concurrencia nos permitirá comprobar condiciones de carrera o bloqueos, ya que el comportamiento de las peticiones es más natural que si se

realizan las 600 seguidas en un bucle. A continuación se presentan los datos obtenidos en una de las tantas pruebas realizadas.

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# ab -g resultados.csv -c 35 -n 600 http://solidosistemas.com.ar/index.php/qnetwork/adaptacion2/55545
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking solidosistemas.com.ar (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Finished 600 requests

Server Software:      nginx
Server Hostname:     solidosistemas.com.ar
Server Port:         80

Document Path:       /index.php/qnetwork/adaptacion2/55545
Document Length:     1266 bytes

Concurrency Level:   35
Time taken for tests: 12.446 seconds
Complete requests:   600
Failed requests:     33
  (Connect: 0, Receive: 0, Length: 33, Exceptions: 0)
Write errors:        0
Non-2xx responses:   567
Total transferred:   2346275 bytes
HTML transferred:    2233710 bytes
Requests per second: 48.21 [#/sec] (mean)
Time per request:    726.016 [ms] (mean)
Time per request:    20.743 [ms] (mean, across all concurrent requests)
Transfer rate:       184.10 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:     40  167 371.5    54  3051
Processing:  45  431 1124.1    65  9763
Waiting:     45  411 1114.4    65  9313
Total:       87  599 1175.2   126  9813

Percentage of the requests served within a certain time (ms)
 50%    126
 66%    161
 75%    202
 80%    921
 90%   3087
 95%   3116
 98%   3153
 99%   4141
100%   9813 (longest request)
    
```

Figura 5.5 – Prueba de rendimiento con ApacheBench.

Como podemos observar con esta carga de trabajo nuestro servidor responde 48 peticiones por segundo. Todo el test tomó 12.446 segundos para ser completado, por lo que se realizaron 600 peticiones en 12.446 segundos, esto nos da un tiempo de $12.446/600 = 20.74$ milisegundos por petición, pero no representa el tiempo real ya que tenemos peticiones concurrentes, por tal motivo $20.74 * 35 = 726.016$ milisegundos sería el tiempo promedio para completar una petición con el nivel de concurrencia actual. Lo que representa un tiempo de respuesta aceptable para los requerimientos de nuestra aplicación. También podemos observar que el mínimo tiempo de conexión fue 126 milisegundos y el máximo 9813 milisegundos.

Se realizaron repetidas pruebas con ApacheBench en diferentes horas del día y se obtuvieron resultados similares desde el servidor, lo que nos permite concluir que el rendimiento es más que aceptable para cumplir con los objetivos de nuestra aplicación.

V.3. PRUEBAS A LA APLICACIÓN MÓVIL

Las aplicaciones Android se ejecutan en dispositivos con recursos limitados, como memoria, potencia de procesador y fuente de alimentación. También se ejecutan en una variedad de dispositivos con diferentes versiones de Sistema Operativo Android, diferentes configuraciones de hardware y dependen de varios factores, como la conectividad de red, conexión con otros sistemas, etc. Por lo tanto es muy importante depurar, probar y optimizar una aplicación Android, ya que tener una cobertura de prueba razonable puede ayudar a mejorar y mantener la misma.

V.3.1 Estrategia de Testing en Android

Como no es posible probar una aplicación en todas las posibles configuraciones de dispositivos, es una práctica común ejecutar las pruebas en las configuraciones de dispositivos típicos en los cuales va a ser utilizada la aplicación, lo recomendable es probar la aplicación al menos en un dispositivo con la configuración más baja posible y en un dispositivo con la configuración más alta disponible, por ejemplo densidad de pixeles, resolución de pantalla, versión de sistema operativo, entre otros, para asegurarse que la aplicación funciona bien en estos dispositivos.

En nuestro caso las pruebas fueron realizadas en dispositivos virtuales y en dispositivos físicos como Smartphones y Tablets con diferentes características de hardware y software que se presentarán en las siguientes secciones.

V.3.2 Pruebas a la Aplicación Android

Las principales características probadas en nuestra aplicación para asegurar un perfecto desempeño y excelente experiencia para el usuario fueron:

- ❖ *Eventos del ciclo de vida de las Activities*: se probó que las activities de la aplicación manejan correctamente los eventos del ciclo de vida y que el estado de instancia (*instance state*) de los componentes de la interfaz son restaurados correctamente cuando se disparan los eventos de cambio de configuración. En nuestra aplicación tenemos dos activities claves LoginActivity y MainActivity, las

pruebas validaron que cada actividad ejecutaba correctamente las instrucciones en los eventos de ciclo de vida. Específicamente en nuestra aplicación el método *onCreate()* es el más relevante y el que mayor cantidad de código contiene. A continuación se detallan las pruebas realizadas en cada Activity.

➤ *LoginActivity*:

- ✓ Pruebas de inicio de sesión.
- ✓ Chequeo de inicio de sesión automático.
- ✓ Control de cierre de sesión.
- ✓ Control de acceso a la actividad principal.
- ✓ Pruebas de conexión con el servidor mediante JavaScript.
- ✓ Pruebas sin conexión a Internet.

➤ *MainActivity*:

- ✓ Control y Pruebas del *Navigation Drawer* (panel deslizante de navegación).
- ✓ Prueba de *fragments* y código asociado a los mismos.
- ✓ Pruebas del escáner de códigos QR con la librería Zxing;

❖ *Operaciones en la base de datos y el sistema de archivos*: se realizaron pruebas de lectura y escritura en el sistema de archivos para verificar el correcto almacenamiento y cifrado del nombre de usuario y contraseña necesarios para facilitar el inicio de sesión en nuestra aplicación. Específicamente se hizo uso de una clase *ObscuredSharedPreferences* que es una implementación de la clase *SharePreferences* de la API de Android que permite almacenar las preferencias de la aplicación usando ciertas características de encriptación para mantener un determinado nivel de seguridad. En este caso las pruebas consisten en verificar que el archivo de preferencia es correctamente escrito y leído en forma encriptada del sistema de archivo de Android.

❖ *Diferentes configuraciones de dispositivos*: con el fin de abarcar una amplia gama de dispositivos con S.O. Android, la aplicación fue probada en una gran cantidad de dispositivos virtuales, y también en una variedad de dispositivos físicos (*Smartphones* y *Tablets*), con diferentes configuraciones de hardware y software, de manera exitosa. Se espera que estos dispositivos o similares sean los utilizados por los estudiantes en el entorno real. A continuación se resumen los dispositivos físicos en los cuales fue testada la aplicación.

Dispositivo Móvil	Tipo	Versión de S.O.	Procesador	Resolución
Motorola RAZR i	Smartphone	Jelly Bean 4.1	Intel Atom Z2460 2GHz	540x960 píxeles, 4.3”
Samsung Galaxy Pocket	Smartphone	Jelly Bean 4.1	ARM 11 832 MHz	240x320 píxeles, 2.8”
Samsung Galaxy Fame Lite	Smartphone	Jelly Bean 4.1	Procesador de 850 MHz	320 x 480 píxeles, 3.5”
Samsung Galaxy S2	Smartphone	Ice Scream Sandwich 4.0	Exynos Cortex A9 Dual-Core 1.2 GHz	480x800 píxeles, 4.3”
Samsung Galaxy S3	Smartphone	Jelly Bean 4.3	Exynos 4 Quad-Core 1.4 GHz	720x1280 píxeles, 4.8”
Samsung Galaxy S4	Smartphone	Jelly Bean 4.3	Exynos 5410 Octa-Core 1.6 GHz	1080x1920 píxeles, 5.0”
Samsung Galaxy S5	Smartphone	KitKat 4.4	Snapdragon 801 Quad-Core 2.5 GHz	1080x1920 píxeles, 5.1”
Sony Xperia M	Smartphone	Jelly Bean 4.1	Qualcomm MSM8227 Dual-Core 1GHz	480x854 píxeles, 4.0”
Samsung Galaxy Tab2	Tablet	Jelly Bean 4.1	Cortex A9 Dual-Core 1GHz	800x1280 píxeles, 10.1”

Tabla 5.2 - Dispositivos móviles en los que fue testeada la aplicación.

V.3.3 Pruebas del módulo de adaptación en la Aplicación

En nuestro trabajo se consideran tres niveles de conocimiento (bajo, medio y alto) y dos dimensiones del modelo de Felder y Silverman (visual-verbal y sensitivo-intuitivo) para realizar la adaptación. La relación entre estas dimensiones y lo que podemos adaptar se muestra a continuación:

Dimensión	Sensitivo/Intuitivo	Visual/Verbal
¿Qué podemos adaptar?	Contenido	Presentación

La dimensión *Sensitivo/Intuitivo* está relacionada con el contenido, más concretamente con el tipo de contenido que se presenta. Los estudiantes intuitivos prefieren solo información conceptual, abstracta y teórica. Por el contrario, los estudiantes sensitivos además del contenido teórico prefieren que la información se complemente con ejemplos prácticos.

La dimensión *Visual/Verbal*: está relacionada con la presentación de la información, es una dimensión que puede resultar difícil de adaptar ya que puede eliminar material educativo, se debe ser muy cuidadoso con esto al cargar la base de datos, además

se debe tener en cuenta que todo contenido debe poder expresarse de ambas formas, mediante palabras y mediante imágenes.

Teniendo en cuenta las dimensiones Sensitivo/Intuitivo y Visual/Verbal, los posibles modelos de presentación de la información se presentan a continuación

	Sensitivo	Intuitivo
Visual	1) Teoría con imágenes, gráficos o tablas. 2) Ejemplos prácticos.	1) Teoría con imágenes, gráficos y tablas.
Verbal	1) Teoría con información textual. 2) Ejemplos prácticos.	1) Teoría con información textual.

Tabla 5.3 - Combinaciones de Estilos en base a las dimensiones Visual-Verbal y Sensitivo-Intuitivo.

Teniendo en cuenta los tres niveles de conocimiento y los cuatros estilos de aprendizaje se desprenden 12 tipos de usuarios posibles:

- ✓ Usuario Hipotético 1: Estilo Sensitivo-Visual / Nivel de conocimiento Bajo.
- ✓ Usuario Hipotético 2: Estilo Sensitivo-Visual / Nivel de conocimiento Medio.
- ✓ Usuario Hipotético 3: Estilo Sensitivo-Visual / Nivel de conocimiento Alto.
- ✓ Usuario Hipotético 4: Estilo Sensitivo-Verbal / Nivel de conocimiento Bajo.
- ✓ Usuario Hipotético 5: Estilo Sensitivo-Verbal / Nivel de conocimiento Medio.
- ✓ Usuario Hipotético 6: Estilo Sensitivo-Verbal / Nivel de conocimiento Alto.
- ✓ Usuario Hipotético 7: Estilo Intuitivo-Visual / Nivel de conocimiento Bajo.
- ✓ Usuario Hipotético 8: Estilo Intuitivo-Visual / Nivel de conocimiento Medio.
- ✓ Usuario Hipotético 9: Estilo Intuitivo-Visual / Nivel de conocimiento Alto.
- ✓ Usuario Hipotético 10: Estilo Intuitivo-Verbal / Nivel de conocimiento Bajo.
- ✓ Usuario Hipotético 11: Estilo Intuitivo-Verbal / Nivel de conocimiento Medio.
- ✓ Usuario Hipotético 12: Estilo Intuitivo-Verbal / Nivel de conocimiento Alto.

Para realizar las pruebas se cargaron los 12 usuarios hipotéticos abarcando todas las combinaciones posibles. A continuación se presentan los resultados obtenidos en algunas de las pruebas, donde se observa parte de la información obtenida por los usuarios cuando utilizan una Tablet.

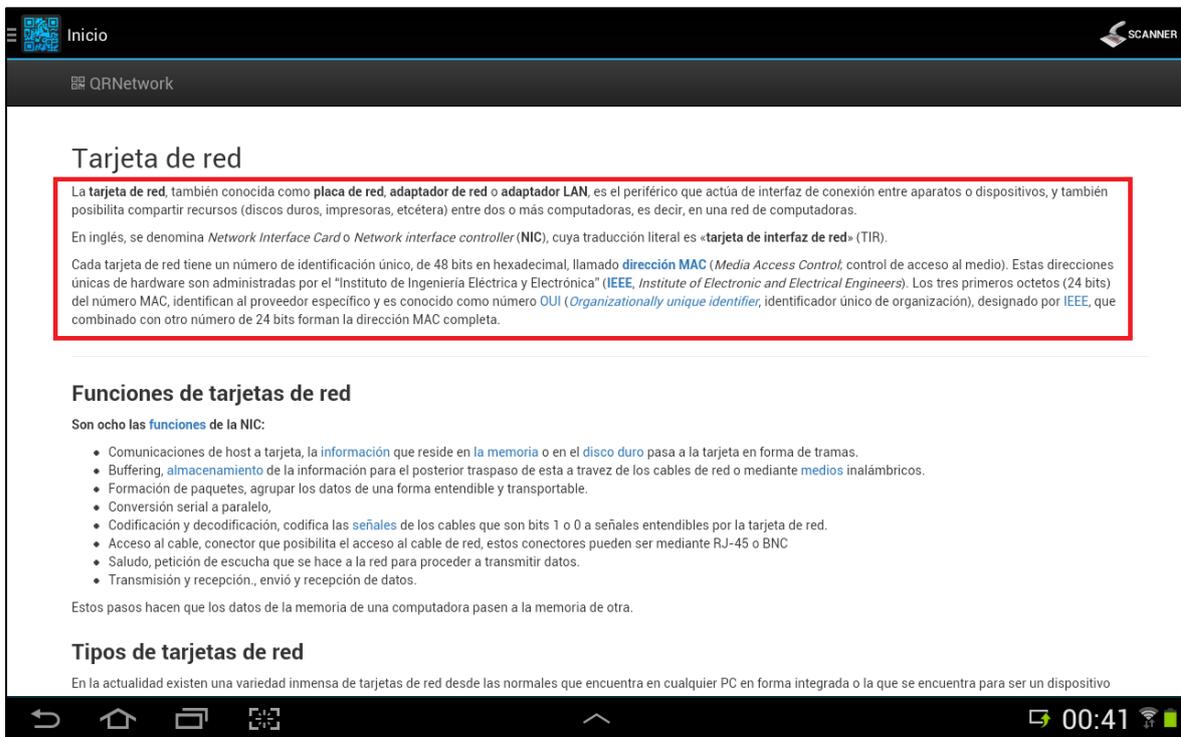


Figura 5.6 - Información obtenida por el usuario hipotético 10 (Intuitivo-Verbal, Bajo).

En la Figura 5.6 se resalta con un rectángulo rojo la porción de texto correspondiente a la Introducción, que ha resultado de aplicar la técnica de texto condicional. En este caso la descripción del componente de red que se presenta es mínima, ya que el nivel de conocimiento del usuario es bajo. El texto que se muestra debajo corresponde con la siguiente sección del documento generado (Sección principal).

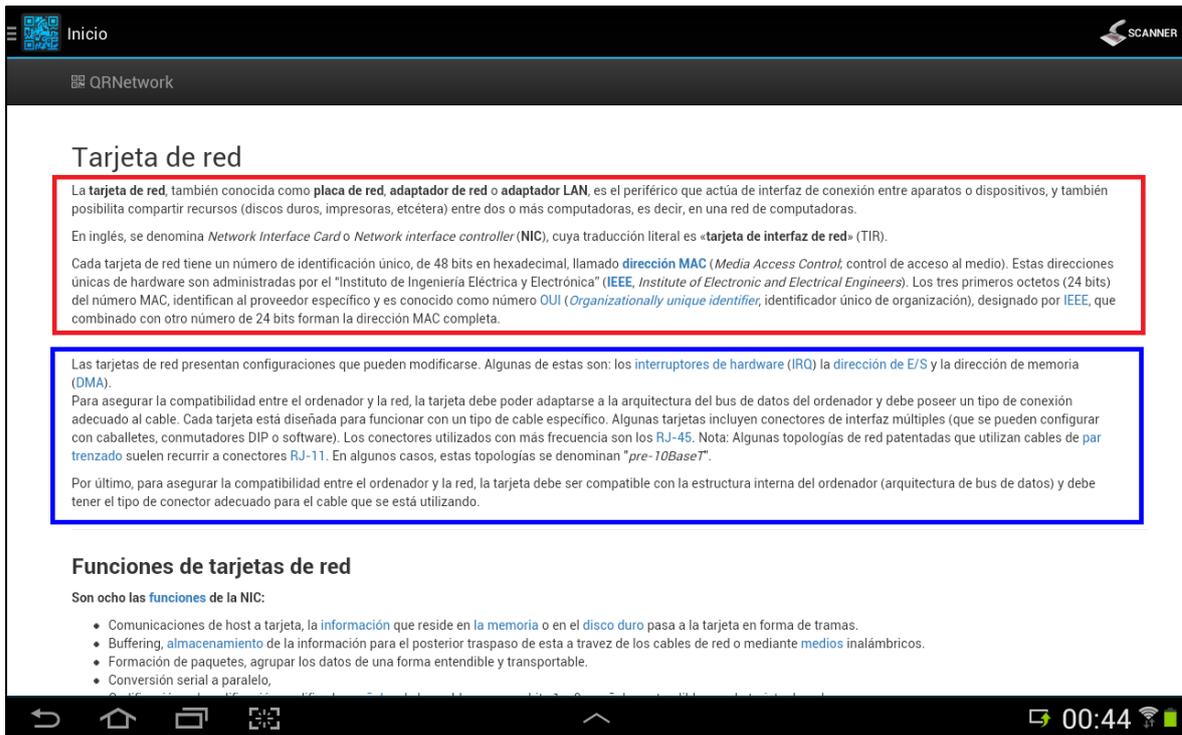


Figura 5.7 - Información obtenida por el usuario hipotético 11 (Intuitivo-Verbal, Medio).

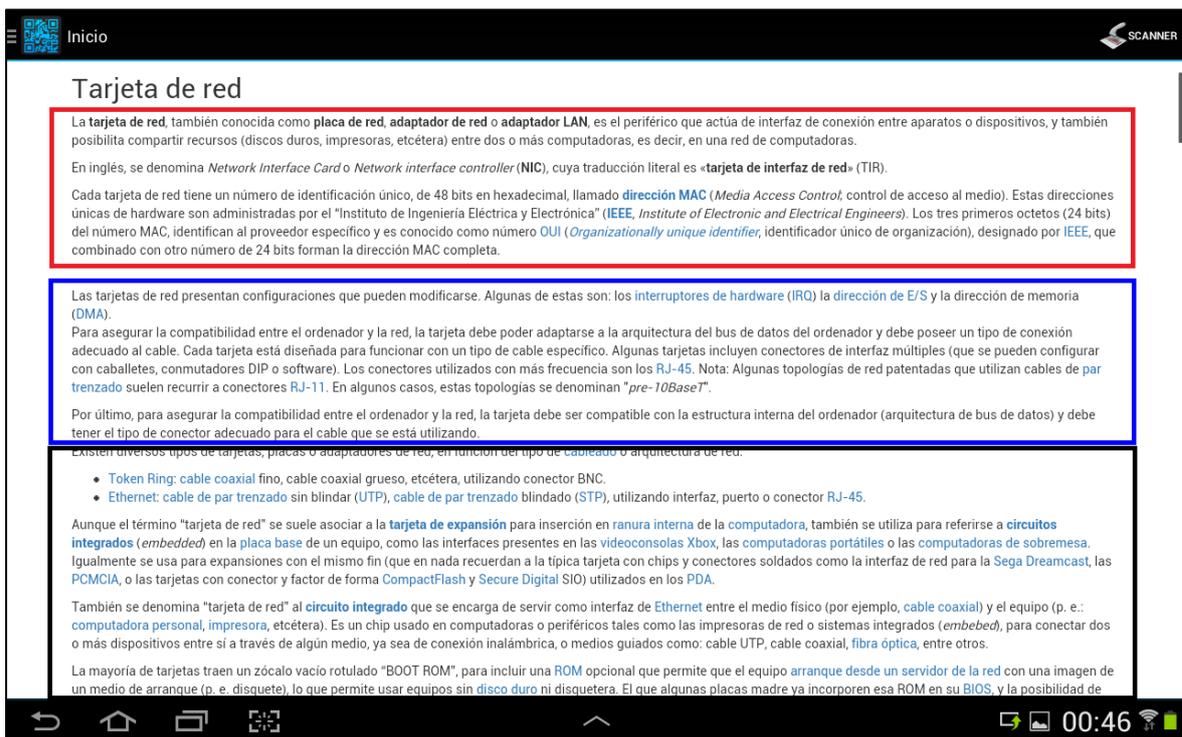


Figura 5.8 - Información obtenida por el usuario hipotético 12 (Intuitivo-Verbal, Alto)

En las Figuras 5.7 y 5.8 se puede observar claramente como el módulo de adaptación va complementando la sección de la introducción con porciones de texto cargadas para cada nivel de conocimiento. En las figuras se ha resaltado con un rectángulo rojo el texto para el nivel bajo, azul para el nivel medio y negro para el nivel alto con el fin

de mostrar la personalización, aunque esto resulta transparente para el usuario que hace uso de la aplicación móvil.

Como se puede observar en la figura 5.9 el usuario hipotético 8 con estilo de aprendizaje intuitivo-visual y nivel de conocimiento medio, obtiene información con gran cantidad de imágenes (en la sección principal del documento generado), por el contrario en la figura 5.10 el usuario hipotético 11 con un estilo intuitivo-verbal y nivel de conocimiento medio obtiene información exclusivamente textual acerca de un componente RJ-45.

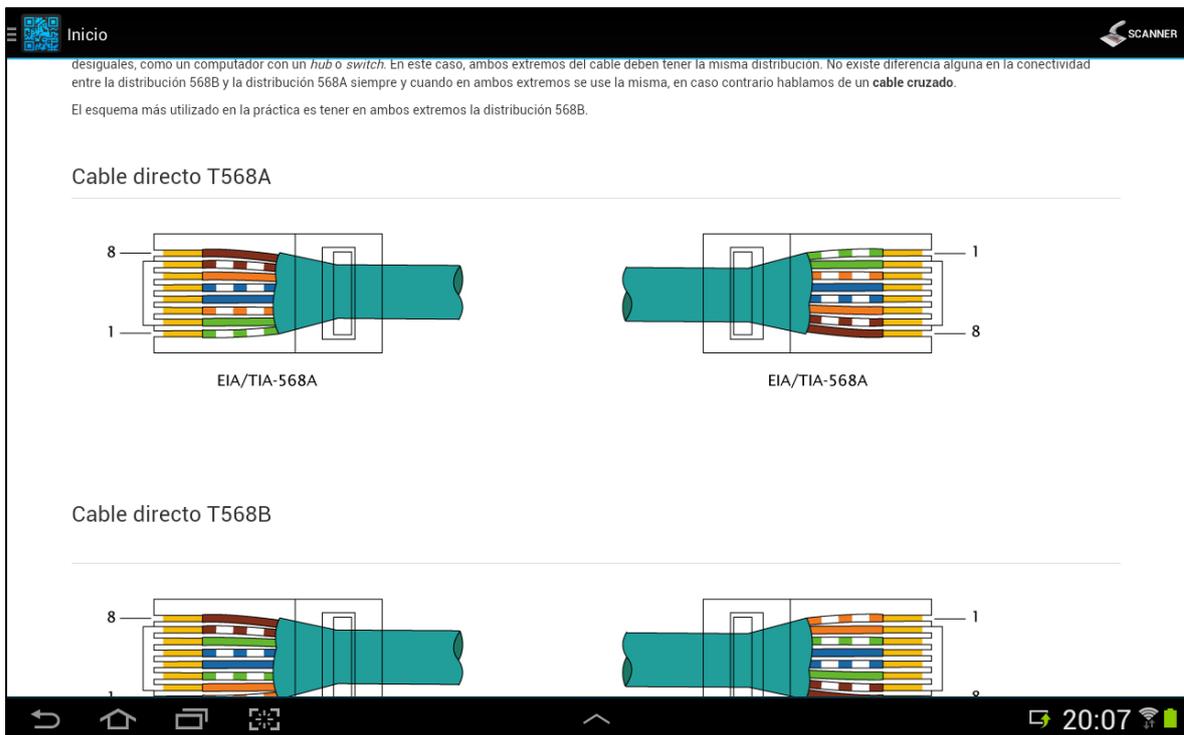


Figura 5.9 - Información obtenida por el usuario hipotético 8 (Intuitivo-Visual, Medio) para un RJ-45.

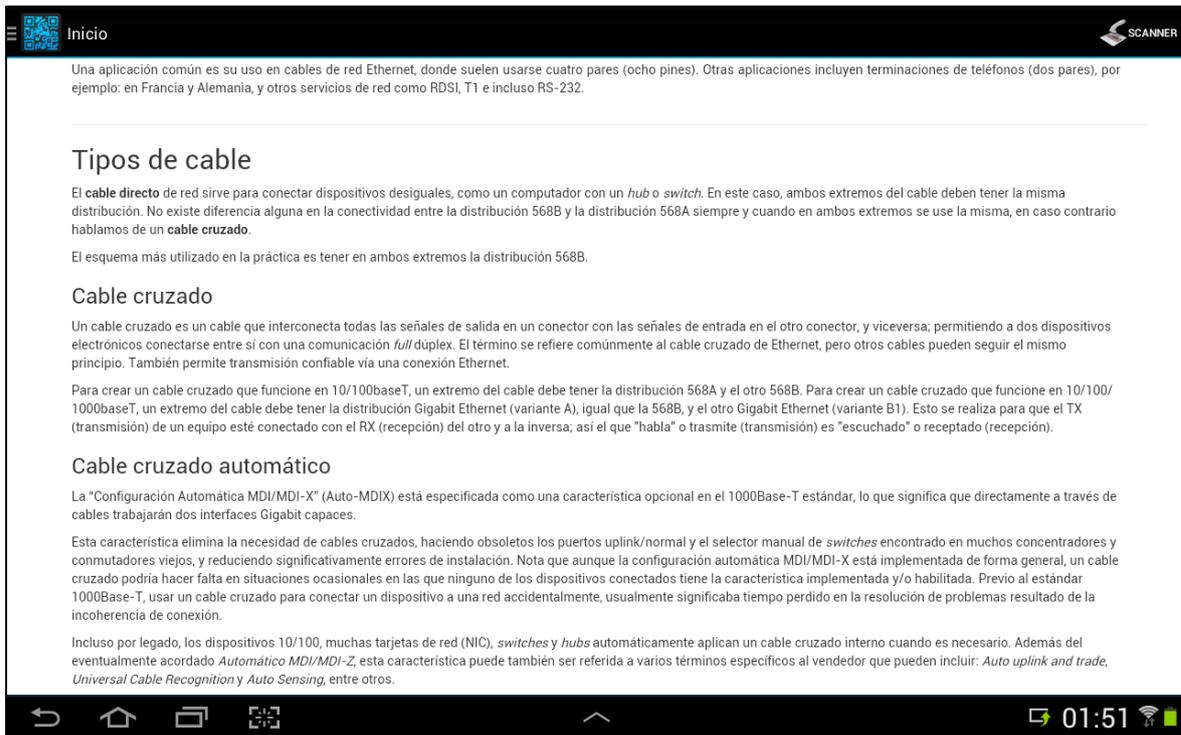


Figura 5.10 - Información obtenida por el usuario hipotético 11 (Intuitivo-Verbal, Medio) para un RJ-45

En la figura 5.11 se observa que el módulo de adaptación presenta al final del documento generado, la sección (contraíble y desplegable) de ejemplos prácticos para un usuario sensitivo. En este caso el estudiante obtiene información relacionada al armado de un cable de red con conectores RJ-45, por el contrario en la figura 5.12 se observa que la sección de ejemplo no ha sido embebida en el documento generado ya que un estudiante intuitivo tiene preferencias por la teoría y conceptos abstractos los cuales son presentados en la sección principal.

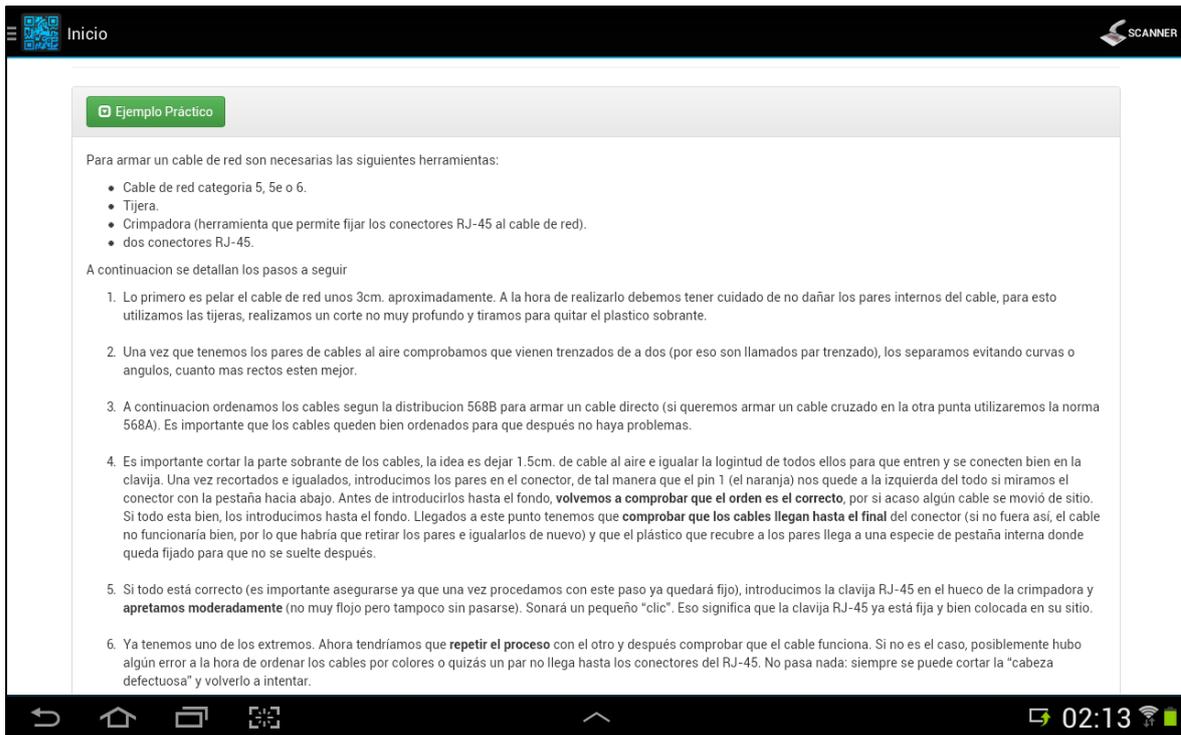


Figura 5.11 – Sección ejemplos obtenida por el usuario hipotético 6 (Sensitivo-Verbal, Alto) para un RJ-45

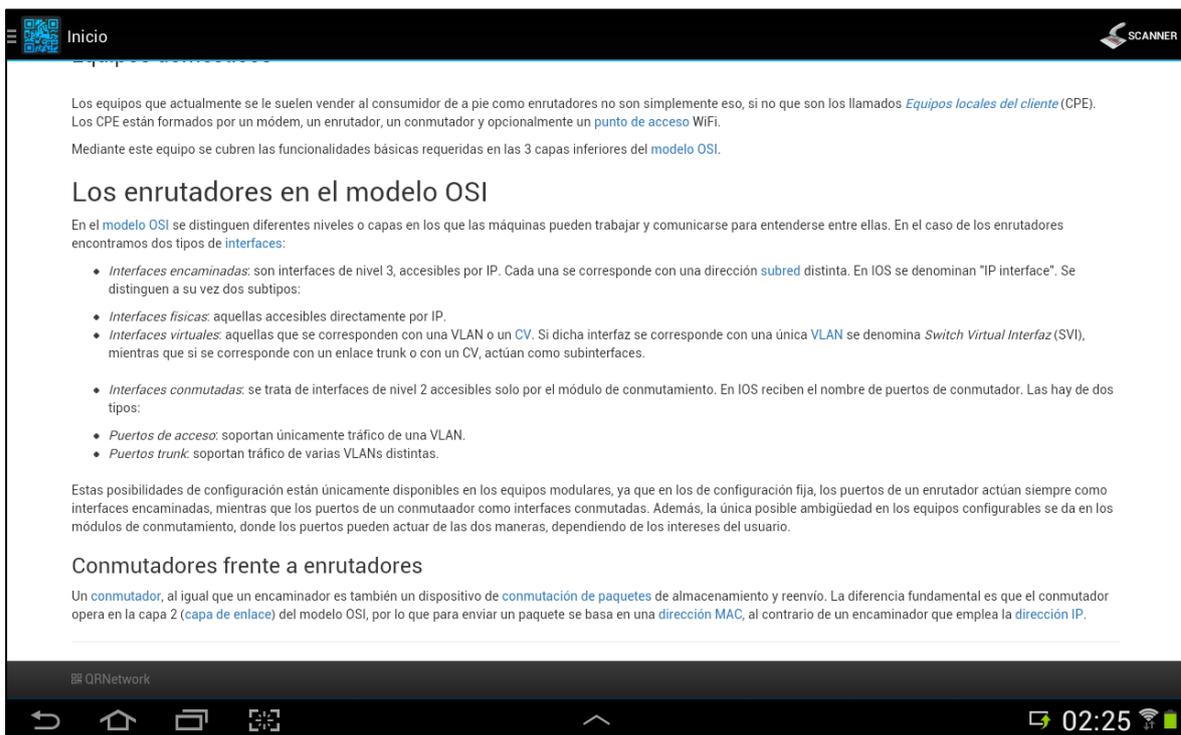


Figura 5.12 – Información obtenida por el usuario hipotético 12 (Intuitivo-Verbal, Alto) para un Router.

V.4. ANALISIS DE RESULTADOS

En base a las pruebas realizadas y a los datos obtenidos, podemos decir que tanto la Aplicación Móvil como la Aplicación Web están finalmente preparadas para trabajar en conjunto y brindar una gran experiencia de usuario cuando sean utilizadas. Resumiendo, podemos decir que la Aplicación Móvil se ejecuta correctamente cuando atraviesa los diferentes eventos del ciclo de vida, por otro lado las pruebas al hacer el cifrado, la lectura y la escritura en el sistema de archivos fueron satisfactorias y los resultados obtenidos en la prueba de configuración demostraron que la aplicación funciona adecuadamente en una amplia variedad de dispositivos móviles con S.O. Android. Por su parte, las pruebas a la Aplicación Web permitieron corregir errores menores, sobre todo de sintaxis, algunos semánticos relacionados a etiquetas HTML5 y algunos enlaces rotos que se resolvieron sin mayores inconvenientes y demostraron que la aplicación responde para personalizar el contenido y la presentación a los doce estereotipos de usuarios previstos. Además los valores obtenidos en la prueba de rendimiento aseguran buenos tiempos de respuesta por parte del servidor cuando se realiza una gran cantidad de peticiones con la concurrencia necesaria. Con toda esta información podemos concluir que finalmente la aplicación móvil está lista para ser utilizada en el entorno real con un rendimiento más que aceptable.

CAPÍTULO VI CONCLUSIONES

La tecnología móvil está siendo ampliamente incorporada en todos los ámbitos incluido el educativo, universidades de todo el mundo la incorporan en diferentes proyectos intentando mejorar los procesos de aprendizaje. La aplicación aquí presentada intenta formar parte de este modelo de educación basada en los dispositivos móviles, la conexión inalámbrica y los usuarios móviles, nos referimos al paradigma del aprendizaje móvil. Además, con el fin de potenciar las capacidades de personalización que posee la tecnología móvil y de brindar una experiencia de usuario individualizada, se integraron dos disciplinas: la Educación con los *Estilos de Aprendizaje*, en nuestro caso el modelo de Felder y Siverman (1998) y la Informática con los *Sistemas Hipermedia Adaptativos*, para obtener como resultado final una aplicación móvil que permite la lectura de códigos QR asociados a determinados puntos de interés (en nuestro caso componentes de red) y brinda información personalizada basándose en el estilo de aprendizaje y nivel de conocimiento del usuario.

Con respecto a la metodología Scrum podemos afirmar que se adaptó perfectamente al proyecto, entre las ventajas obtenidas están la facilidad de implementación y la agilidad en cuanto a cambios evitando la burocracia y la excesiva documentación. En nuestro caso no hubo mayores cambios respecto a los requisitos formulados en un principio, se hicieron pequeños cambios que fueron progresivamente incorporados en los diferentes Sprints.

Por su parte el producto final cumple con todos los requisitos planteados. Al ser una aplicación nativa posee un alto rendimiento y funciona de manera suave y fluida bajo los dispositivos móviles con S.O. Android. Cuando se realiza la lectura de un código QR la aplicación presenta casi instantáneamente la información personalizada generada por el módulo de adaptación en el servidor, donde se mantiene actualizada y centralizada toda la información. El módulo de adaptación es lo que le da a la aplicación una potencialidad desde lo pedagógico, que no se encuentra en la mayoría de las aplicaciones educativas móviles existentes. Dotar a la aplicación de la capacidad de mostrar contenido adaptado en función del estilo de aprendizaje y nivel de conocimiento del estudiante, transforma a la

aplicación en un recurso muy valioso para un aprendizaje centrado en el alumno, facilitando también el autoaprendizaje.

Si bien, por cuestiones de tiempo no se pudo concretar una prueba de usabilidad en entorno real de trabajo, pruebas informales realizadas con estudiantes permitieron recabar su opinión y adelantar que la interfaz de la aplicación resultó atractiva, intuitiva y fácil de usar. En consecuencia, se espera que la aplicación contribuya a mejorar los procesos de aprendizaje de los alumnos y sea una herramienta útil para los docentes de la Cátedra Redes I.

Entre los puntos débiles podemos mencionar que la aplicación está restringida a un determinado espacio ya que depende del uso de códigos QR para determinar que objeto se intenta reconocer. Por otro lado requiere un cierto esfuerzo por parte del docente al momento de realizar la carga de la base de datos con la información acerca de los componentes ya que se deben considerar todas las combinaciones posibles entre estilos de aprendizaje y niveles de conocimiento; sin embargo, esto se realiza una única vez para cada componente cargado.

Como trabajo futuro a corto plazo se espera concretar una prueba de usabilidad en entorno real; y a mediano plazo ampliar las funcionalidades de la aplicación con el fin de proporcionar reconocimiento de componentes no solo en los laboratorios donde trabajan los alumnos sino también en otros campos de trabajo. Además, se espera experimentar con variantes de la misma aplicación en otras asignaturas e inclusive en otras carreras con el fin de apoyar los procesos de enseñanza en el ámbito universitario.

REFERENCIAS BIBLIOGRÁFICAS

Abrahamsson P. (2005). *Keynote: Mobile software development - the business opportunity of today*. Proceedings of the International Conference on Software Development. Reykjavik, Iceland, pp. 20-23.

Alonso C., Gallego D. & Honey P. (1994). *Los Estilos de Aprendizaje. Procedimientos de diagnóstico y Mejora*. Bilbao: Ediciones Mensajero.

Aretio G. L. (2004). *Aprendizaje Móvil, m-learning*. Editorial del BENED.

Bajraktarevic N., Hall W. & Fullick P. (2003). *Incorporating learning styles in hypermedia environment: Empirical evaluation*. In P. de Bra, H. C. Davis, J. Kay, and M. Schraefel, editors, Proceedings of the Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems, pp. 41-52, Nottingham, UK. Eindhoven University.

Barragán P. P. (2008). *Una Propuesta de Incorporación de los Estilos de Aprendizaje a los Modelos de Usuario en Sistemas de Enseñanza Adaptativos*. Tesis Doctoral. Universidad Autónoma de Madrid, Madrid, España.

Beaumont I. H. (1994). *User Modelling in the Interactive Anatomy Tutoring System ANATOM-TUTOR*. User Modeling and User-Adapted Interaction 4(1), pp. 21-45. Reprinted in Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers. 1998, pp. 91-116.

Beck K. (2000). *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA. ISBN: 0321278658.

Beck K., Beedle M., Van Bennekum A., Cockburn A., Cunningham W., Fowler M.,... Thomas D. (2001). *Manifesto for Agile Software Development*. Utah. EEUU. Recuperado de <http://agilemanifesto.org/>

Blanco P., Camarero J., Fumero A., Werterski A. & Rodríguez P. (2009). *Metodología de desarrollo ágil para sistemas móviles. Introducción al desarrollo con Android y el iPhone*. Doctorado en Ingeniería de Sistemas Telemáticos. Universidad Politécnica de Madrid, España.

Boyle C. & Encarnacion A. O. (1994). *MetaDoc: an adaptive hypertext reading system*. User Models and User Adapted Interaction 4 (1), pp. 1-19.

Brusilovsky P. (1996). *Methods and techniques of adaptive hypermedia*. User Modeling and User-Adapted Interaction: The Journal of Personalization Research, 6 (2-3). pp. 87-129.

Brusilovsky P. (2001). *Adaptive hypermedia*. User Modeling and User-Adapted Interaction: The Journal of Personalization Research, vol. 11, N° 1 & 2, pp. 87-110.

- Buitrago A. (2010). *Sistemas hipermedia Adaptativos (SHA) IV*. Universidad de Castilla-La Mancha. España.
- Burgos Aguilar J. V. (2010). *Aprendizaje Móvil: el potencial educativo en la palma de la mano*. En J. V. A. Burgos & R. Lozano (Coords). Tecnología educativa y redes de aprendizaje de colaboración: retos y realidades de innovación en el ambiente educativo. México: Trillas.
- Calderón A., Dámaris S., Rebaza V. & Carlos J. (2007). *Metodologías Ágiles*. Escuela de Informática. Facultad de Ciencias Físicas y Matemáticas. Universidad Nacional de Trujillo. Perú.
- Cha H. J., Kim Y. S., Lee J. H. & Yoon T. B. (2006). *An Adaptive Learning System with Learning Style Diagnosis based on Interface Behaviors*. Workshop Proceedings of Intl. Conf. E-learning and Games (Edutainment 2006).
- Chen S. & Macredie R. (2002). *Cognitive styles and hypermedia navigation: Development of a learning model*. Journal of the American Society for Information Science and Technology, 53(1). pp. 3-15.
- Chorfi H. & Jemni M. (2004). *PERSO: Towards an adaptive e-learning system*. Journal of Interactive Learning Research, 15 (2004), N° 4, pp. 433-447.
- Chwabe G. & Göth C. (2005). *Mobile learning with a mobile game: design and motivational effects*. Journal of Computer Assisted Learning, 21(3), pp. 204-216. doi:10.1111/j.1365-2729.2005.00128.x.
- Cope B. & Kalantzis M. (2009). *Ubiquitous Learning*. University of Illinois Press. 279 págs.
- De Bra P. & Calvi L. (1998). *AHA! An open Adaptive Hypermedia Architecture*. The New Review of Hypermedia and Multimedia, Vol. 4, Taylor Graham Publishers, pp. 115-139.
- De Natale M. L. (1990). *Rendimiento escolar*. En Flores, G y Gutiérrez, I. Diccionario de Ciencia de la Educación. Madrid: Paulinas.
- De Rosis F., de Carolis B. & Pizzutilo S. (1993). *User tailored hypermedia explanations*. Workshop Adaptive Hypertext and Hypermedia at 4th International Conference on User Modeling, Hyannis, MA.
- Deemer P., Benefield G., Larman C. & Vodde B. (2009). *Información básica de Scrum (The Scrum Primer)*. Versión 1.1. ScrumTraining Institute. Traducción de Leo Antoli. Agile-Spain. Recuperado de http://www.goodagile.com/scrumprimer/scrumprimer_es.pdf
- Delia L., Galdamez N., Thomas P. & Pesado P. (2013). *Un análisis experimental de tipo de aplicaciones para dispositivos móviles*. Facultad de Informática. Universidad Nacional de la Plata. XVIII Congreso Argentino de Ciencias de la Computación. (CACIC2013).
- Duran K. S. & Fajardo T. M. A. (2012). *GINDEV: Evaluación de la metodología Scrum en el desarrollo de videojuegos independientes*. Tesis de Licenciatura en Ingeniería en

Sistemas y Tecnologías de Información. Universidad de las Américas Puebla. Cholula, Puebla, México.

Encarnação L. M. (1995). *Adaptivity in graphical user interfaces: an experimental framework*. Computers & Graphics 19 (6), pp. 873-884.

Esquivel F. P. C., González G. M. R. & Flores D. A. (2013). *Estilos de aprendizaje la importancia de reconocerlos en el aula*. Revista Iberoamericana para la Investigación y el Desarrollo Educativo. Vol. 10. ISSN 2007-2619.

Felder R. M. (1996). *Matters of style*. ASEE Prism, 6 (4). pp. 18-23.

Felder R. M. & Henriques E. R. (1995). *Learning and teaching styles in foreign and second language education*. Foreign Language Annals, 28 (1). pp. 21-31.

Felder R. M. & Silverman L. K. (1988). *Learning styles and teaching styles in engineering education*. Engineering Education, 78 (7). pp. 674-681.

Felder R. M. & Soloman B. A. (2004). *Index of learning styles*.

Figuroa N. & Vigliecca M. E. (2006). *Reflexiones sobre nuevos enfoques de enseñanza en ingeniería a partir de las experiencias con estilos de aprendizaje*. Revista de Informática Educativa y Medios Audiovisuales, Vol. 3(7), págs. 32-36.

García A.I. (2009). *Códigos QR Flexibles: Un Proyecto con Dispositivos Móviles para el Trabajo de Calentamiento en Educación Física*. EmásF, Revista Digital de Educación Física. ISSN 1989-8304. Año 4, Núm. 23. 2013.

Gentry J. A. & Helgesen M. G. (1999). *Using learning style information to improve the core financial management course*. Financial Practice and Education, Spring-Summer 1999.

Gonschorek M. & Herzog C. (1995). *Using hypertext for an adaptive help system in an intelligent tutoring system*. En Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, J. Greer (Eds.). Washington, DC: AACE. pp. 274-281.

Hartman B., Lawrence R., Saddington P., Green P., Calabrese J., Dittmer H. & Broderick T. (2015). *Agile for all Company*. Recuperado de <http://www.agileforall.com/intro-to-agile/#ScrumFrameworkImage>.

Hohl H., Bocker D., Gunzenhäuser R. (1996). *Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming*. User Modeling and User-Adapted Interaction 6 (2-3). pp. 131-156.

Hong H. & Kinshuk D. (2004). *Adaptation to student learning styles in web based educational systems*. In World Conference on Educational Multimedia, Hypermedia and Telecommunications, pp. 491-496, Lugano, Switzerland.

Honey P. & Mumford A. (1992). *The Manual of Learning Styles*. (3a Ed.) Peter Honey Publications, Maidenhead.

- Kawamura T., Tahara Y., Hasegawa T., Ohsuga A., Honiden S. (2000). *Bee-gent: bonding and encapsulation enhancement agent framework for development of distributed systems*. Systems and Computers in Japan, 31 (2000), N° 13, pp. 42-56.
- Kay J. & Kummerfeld R. J. (1994). *An Individualised Course for the C Programming Language*. 2nd International WWW Conference "Mosaic and the Web" 1994.
- Keefe J. W. (1979). *Nassp's student learning styles: Diagnosing and prescribing programs*. Technical report, National Association of Secondary School Principals, Reston, VA.
- Keefe J. W. (1988). *Profiling and Utilizing Learning Style*. Reston, Virginia: NASSP.
- Kinshuk T. L. (2004). *Application of Learning Styles Adaptivity in Mobile Learning Environments*. Conference: Third Pan Commonwealth Forum on Open Learning. pp. 4-8.
- Kobsa A., Müller D. & Nill A. (1994). *KN-AHS: An adaptive hypertext client of the user modeling system BGP-MS*. 4-th International Conference on User Modeling, Hyannis, MA, pp. 31-36.
- Koole M. L. (2009). *A model for framing mobile learning*. In M. Ally (Ed.), *Mobile learning: Transforming the delivery of education and training*, 25-47. Edmonton, AB: AU Press. Retrieved from <http://www.aupress.ca/index.php/books/120155>.
- Lee M. (2005). *Getting a move on with mobile learning*. Training Journal, 13.
- Lyytinen K. & Yoo Y. (2002). *Issues and challenges in ubiquitous computing: Introduction*. In Communications of ACM, Vol. 45, N° 12, pp. 62-65.
- McCabe M. & Tedesco S. (2012). *Using QR Codes and Mobile Devices to Foster an Inclusive Learning Environment for Mathematics Education*. International Journal of Technology and Inclusive Education (IJTIE), Vol. 1, Issue 1, pp. 37-43.
- Medina N., Molina F., García L. & Rodríguez M. (2002). *SEM-HP: Un Modelo para el Desarrollo de Sistemas Hipermedia Adaptativos*. Dep. LSI, Universidad de Granada; Dep. Informática, Universidad de Jaén.
- Mockus L., Dawson H., Edel-Malizia S., Shaffer D., Sung A.J. & Swaggerty A. (2011). *The Impact of Mobile Access on Motivation: Distance Education Student Perceptions*. Learning Design at Penn State's World Campus.
- Molina M. J. & Romero D. (2010). *Ambiente de Aprendizaje Móvil Basado en Micro-Aprendizaje*. Revista Iberoamericana de Tecnologías del/da Aprendizaje/Aprendizagem. IEEE-RITA Vol. 5, Núm. 4, Nov. 2010. pp. 159-166.
- Moore M. G. (1989). *Three types of interaction*. The American Journal of Distance Education, 3(2), pp.1-6.
- Moore M. G. (1991). *The American Journal of Distance Education*. Editorial: Distance education theory. 5(3), pp. 1-6.

- Mota Jorge. (2008). *Using Learning Styles and Neural Networks as an Approach to eLearning Content and Layout Adaptation*. PRODEI - Faculty of Engineering of University of Porto, Portugal.
- Nagella U.B. & Govindarajulu P. (2008). *Adaptive Approaches to Context Aware Mobile Learning Applications*. International Journal of Computer Science and Security. Vol. 2. Issue 2. pp. 15-26. India.
- Nelson T. H. (1965). *Complex information processing: a file structure for the complex, the changing and the indeterminate*. In ACM Annual Conference/Annual Meeting. Proceedings of the 1965 20th national conference, pp. 84-100, Cleveland, Ohio, United States.
- Nguyen V. A., Nguyen V. H. & Ho S. D. (2009). *Developing adaptive hypermedia system based on learning design level B with rules for adaptive learning activities*. VNU. Journal of Science, (Natural Sciences and Technology), 25, N°1, pp. 1-12.
- Palacio J. (2014). *Gestión de Proyectos Scrum Manager*. (Scrum Manager I y II) Versión 2.5. Propiedad de Iubaris Info 4 Media S.L.
- Papanikolaou K.A., Grigoriadou M., Kornilakis H. & Magoulas G.D. (2003). *Personalizing the interaction in a Web-based educational hypermedia system: the case of INSPIRE*. User-Modeling and User-Adapted Interaction, 13, pp. 213-267.
- Popescu E., Badica C. & Moraret L. (2010). *Accommodating Learning Styles in an Adaptive Educational System*. Informatica - an International Journal of Computing and Informatics. 2010, pp. 451-462.
- Power Robert. (2012). *QR Cache: Connecting mLearning practice with theory*. MLearn 2012: International Conference on Mobile and Contextual Learning 2012 Proceedings, vol 955, pp. 346-349. CEUR-WS.org.
- Rahimian V. & Ramsin R. (2008). *Designing and agile methodology for mobile software development: a hybrid method engineering approach*. Second International Conference on Research Challenges in Information Science. RCIS 2008, 3-6 June 2008, Marrakech. pp. 337-342.
- Ramos C.M., Aguilar S.L.A. & Howlet P.C.L. (2013). *Uso del Teléfono Inteligente con fines académicos. Caso de estudio: FCA de la UACH*. XVI Congreso Internacional sobre Innovaciones en Docencia e Investigación en Ciencias Económico Administrativas.
- Rikala J. & Kankaanranta M. (2012). *The Use of Quick Response Codes in the Classroom*. MLearn 2012: International Conference on Mobile and Contextual Learning 2012 Proceedings, Vol. 955, pp. 148-155. CEUR-WS.org.
- Saarikoski L., Salojärvi S., Del Corso D. & Ovcin, E. (2001). *The 3DE: An Environment for the Development of Learner-Oriented Customised Educational Packages*. ITHET, pp. 4-6 Julio, 2001, Kumamoto.

Schwaber K. (1995). *SCRUM Development Process*. OOPSLA '95 Workshop Proceedings 16 October 1995, Austin, Texas.

Sharples M., Taylor J. & Vavoula G. (2005). *Towards a Theory of Mobile Learning*. Journal: Proceedings of mLearn 2005. Vol 1. Issue 1. pp. 1-9.

Soloman B. A. (1992). *Inventory of learning styles*. North Carolina State University.

Takeuchi H. & Nonaka I. (1986). *The New New Product Development Game*. Harvard Business Review 64, N° 1 (January–February 1986).

Talukder A. K., Ahmed H. & Yavagal R. R. (2010). *Mobile Computing: Technology, Applications, and Service Creation*. Second Edition. McGraw-Hill Professional.

Torres P. L., López E. A. S., Canós J. H. & Penadés M. C. (2003). *Metodologías Ágiles en el Desarrollo de Software*. VIII Jornadas de Ingeniería del Software y Bases de Datos, JISBD. Alicante, 12 al 14 de Noviembre de 2003.

Triantafillou E., Pomportsis A. & Demetriadis S. (2003). *The design and the formative evaluation of an adaptive educational system based on cognitive styles*. Computers & Education, 41, pp. 87-103.

Turban E., Rainer K. R. & Potter R. E. (2005). *Introduction to Information Technology*. Third Edition. ISBN: 0-471-34780-9, 608 Pages. pp. 169-173.

Vahey P. & Crawford V. (2002). *Palm Education Pioneers Program Final Evaluation Report*. Menlo Park, CA: SRI International.

Yang T.-C., Hwang G.-J. & Yang, S.J.-H. (2013). *Development of an adaptive learning system with multiple perspectives based on students' learning styles and cognitive styles*. Educational Technology & Society, 16 (4), pp. 185-200.

Yusof S., Goolamally N., Latif L.A. & Fadzil M. (2012). *Using QR codes to enhance learning of elementary statistics*. 12th International Conference on Information (ICI12). pp. 312-321.

SCRUM – INCREMENTOS DEL PRODUCTO

A.1. SPRINT I – Control de Acceso

En este Sprint se diseña e implementa el sistema de control de acceso, que involucra por un lado el control de acceso en la Aplicación Web y por otro el control de acceso en la Aplicación móvil, brindando dos funciones claves. El primero permite que solo los administradores (docentes) puedan gestionar los parámetros generales del sistema desde el servidor y el segundo permite la identificación de los usuarios que inician sesión en la aplicación móvil con el fin de brindarles contenido personalizado. Se implementan las funciones para cerrar sesión tanto en la Aplicación móvil como en la Web.

A continuación se presentan las pantallas del *Incremento del Producto* obtenido en el Sprint I.

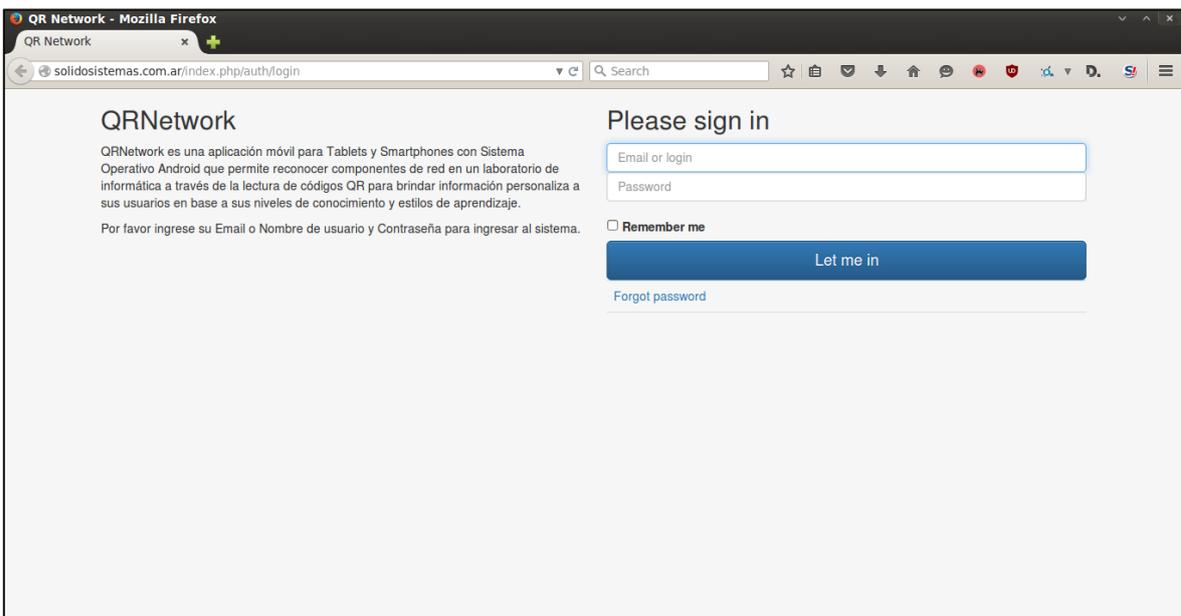


Figura A.1 - Control de Acceso en la Aplicación Web.

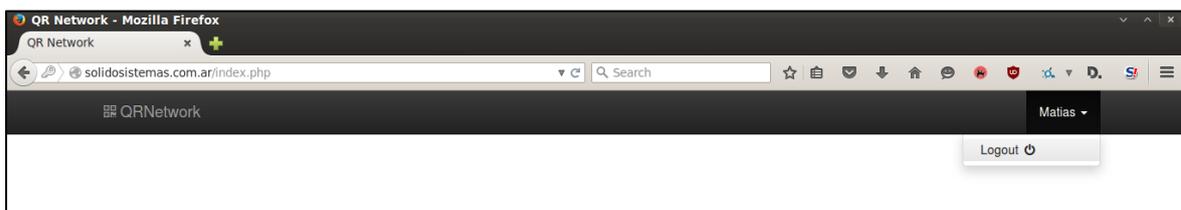


Figura A.2 - Opción para cerrar sesión.

Inicio y cierre de sesión en Smartphones y Tablets.

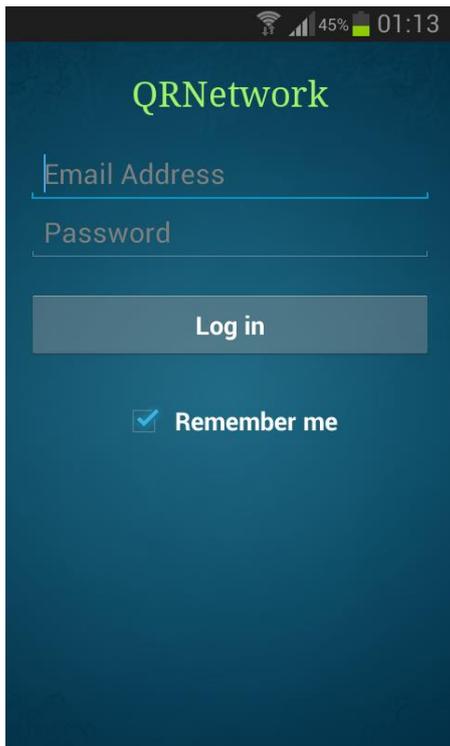


Figura A.3 – Inicio de Sesión en la Aplicación.

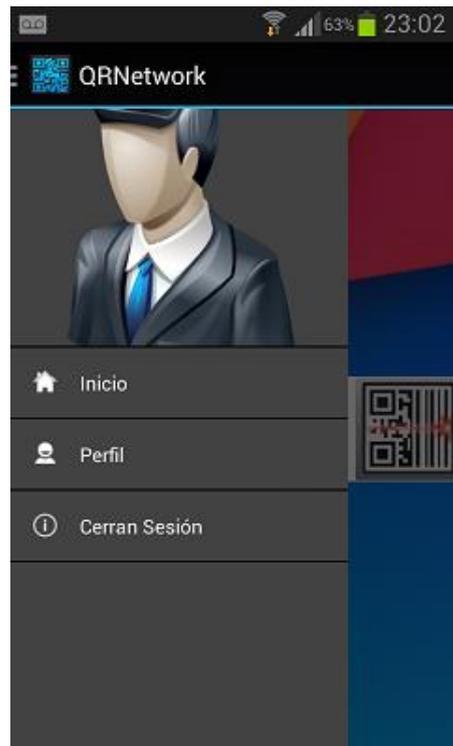


Figura A.4 - Opción para cerrar sesión.

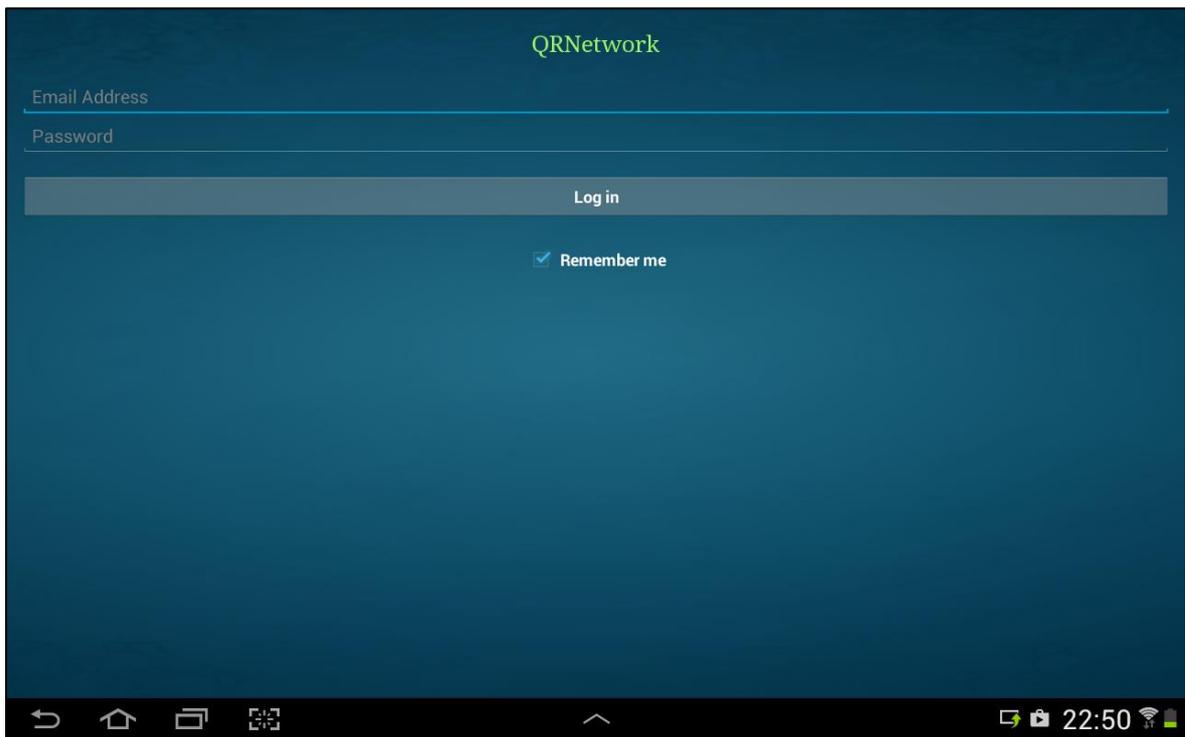


Figura A.5 – Inicio de sesión en una Tablet.

A.2. SPRINT II – Lectura de Códigos QR

En este Sprint se implementa el módulo para lectura de códigos QR en la Aplicación móvil y se integra una nueva vista (*View*) con un contenedor web nativo de Android (*WebView*) donde se podrá visualizar la respuesta del servidor (un documento HTML) cuando se envíe la información de un código QR. Este módulo es clave para nuestra aplicación. El incremento resultado de este Sprint es la aplicación móvil con el escáner integrado junto al navegador web nativo, como se observa a continuación.



Figura A.6 - Escáner de la Aplicación Móvil.

Algunas pruebas enviando los datos al servidor...



Figura A.7 - Envío al servidor de un código QR inexistente en la base de datos.

En el contenedor web (*webview*) de la Aplicación móvil obtenemos...

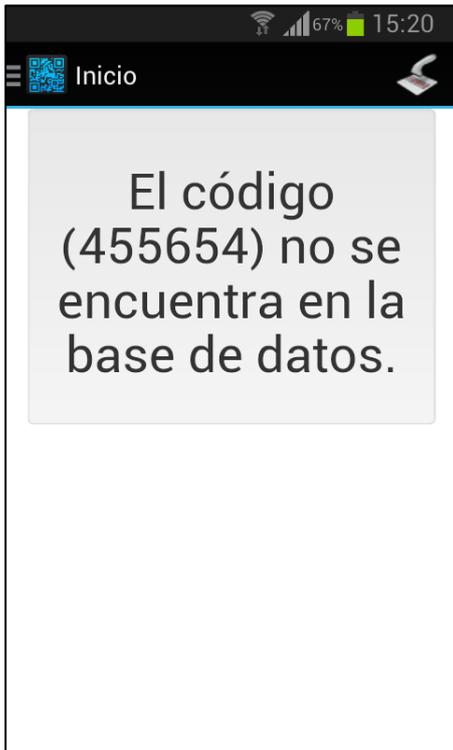


Figura A.8 – respuesta del servidor al recibir un código QR inexistente en la base de datos.

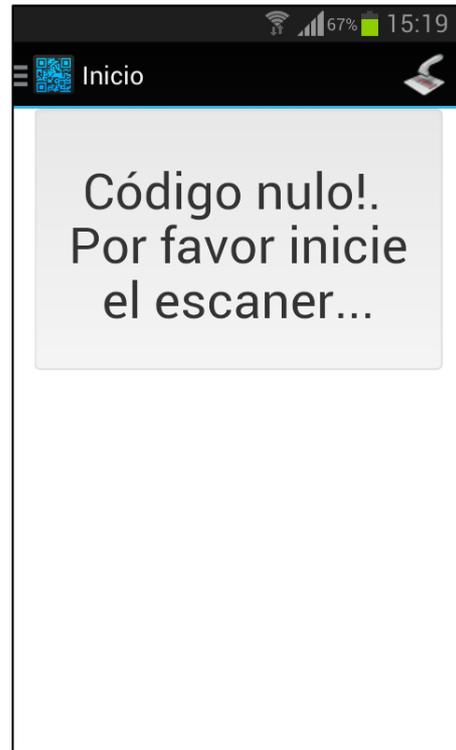


Figura A.9 – respuesta del servidor al recibir un código nulo o vacío.

A.3. SPRINT III – Administración General del Sistema

En este Sprint se diseña y desarrolla el sistema de administración. El incremento del producto consiste de un panel de control en la aplicación web que permite la gestión global del sistema por parte de un administrador (docente), incluyendo inicialmente la gestión de usuarios, perfiles, cursos, temas y subtemas.

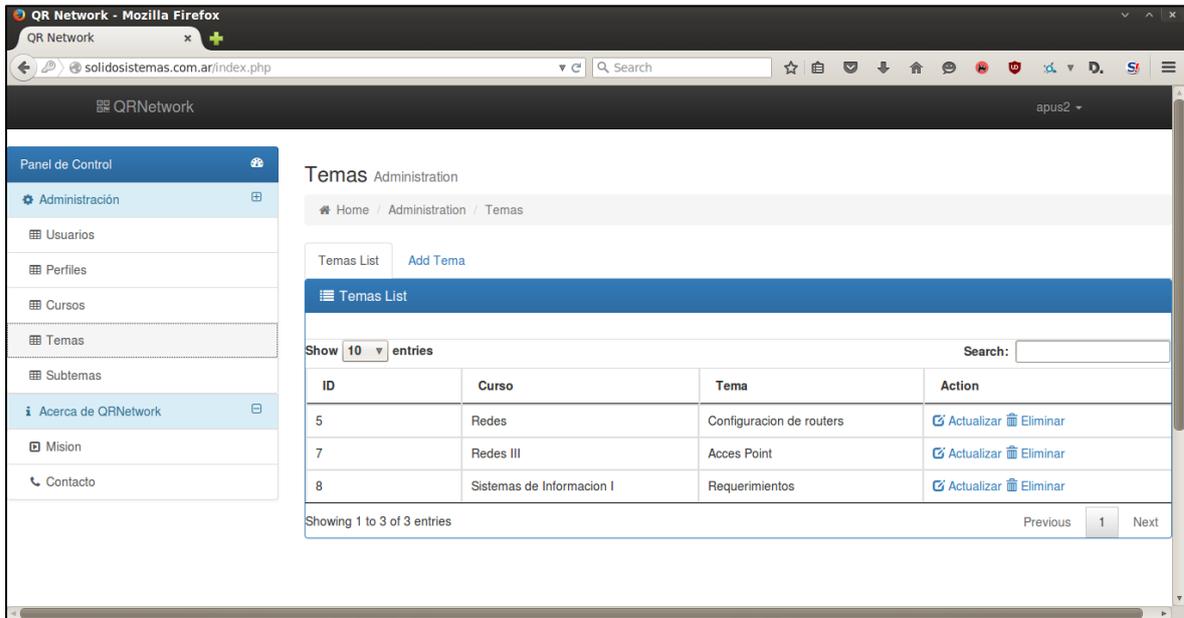


Figura A.10 - Panel de control con las primeras opciones de administración.

En la figura A.10 podemos observar la barra lateral de navegación (*sidebar*) que nos brinda acceso a las opciones de gestión. Inicialmente podemos listar, crear, modificar y eliminar cursos, temas y subtemas. Además podemos agregar usuarios (estudiantes) y actualizar información de sus perfiles.

A continuación se presenta algunas de las opciones descriptas.

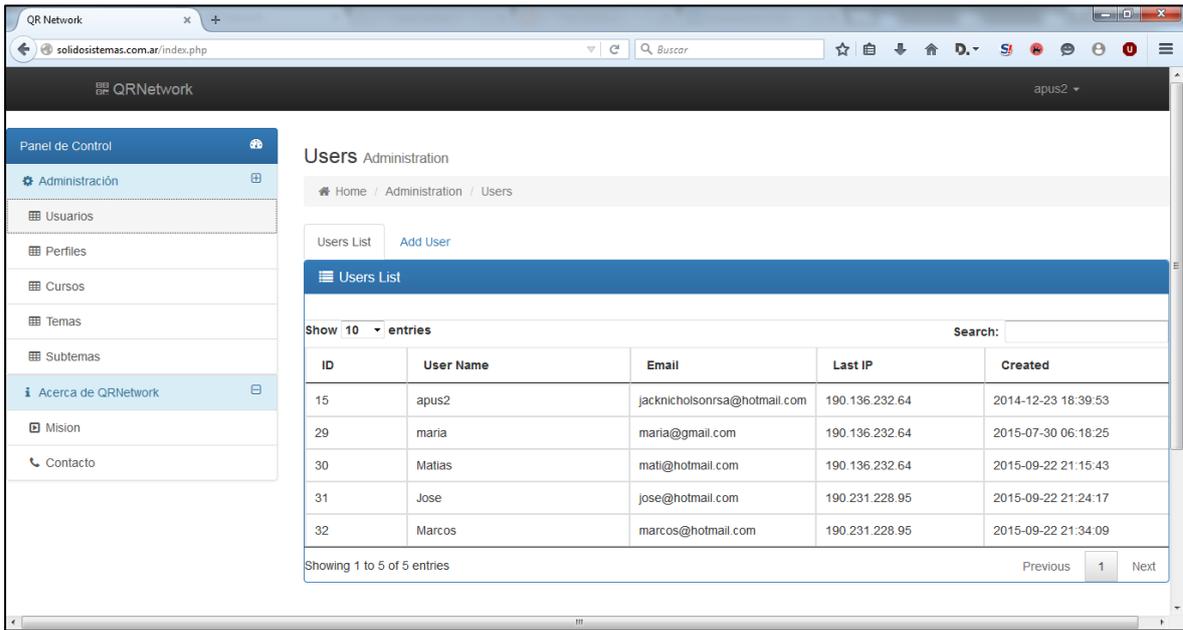


Figura A.11 - Administración de usuarios (Estudiantes).

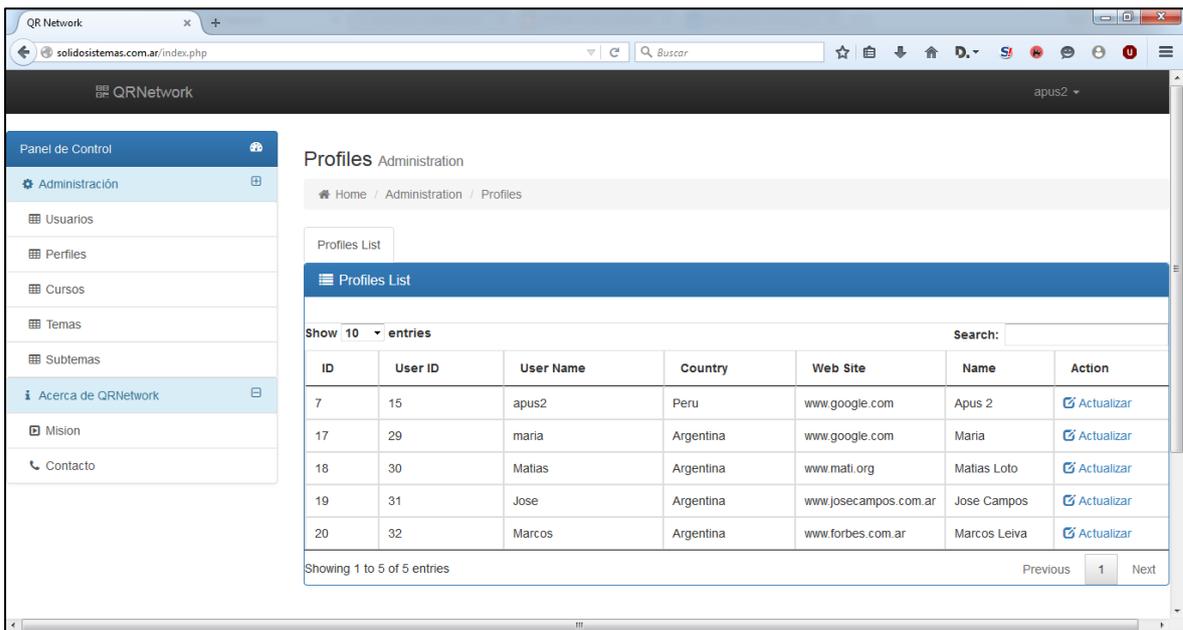
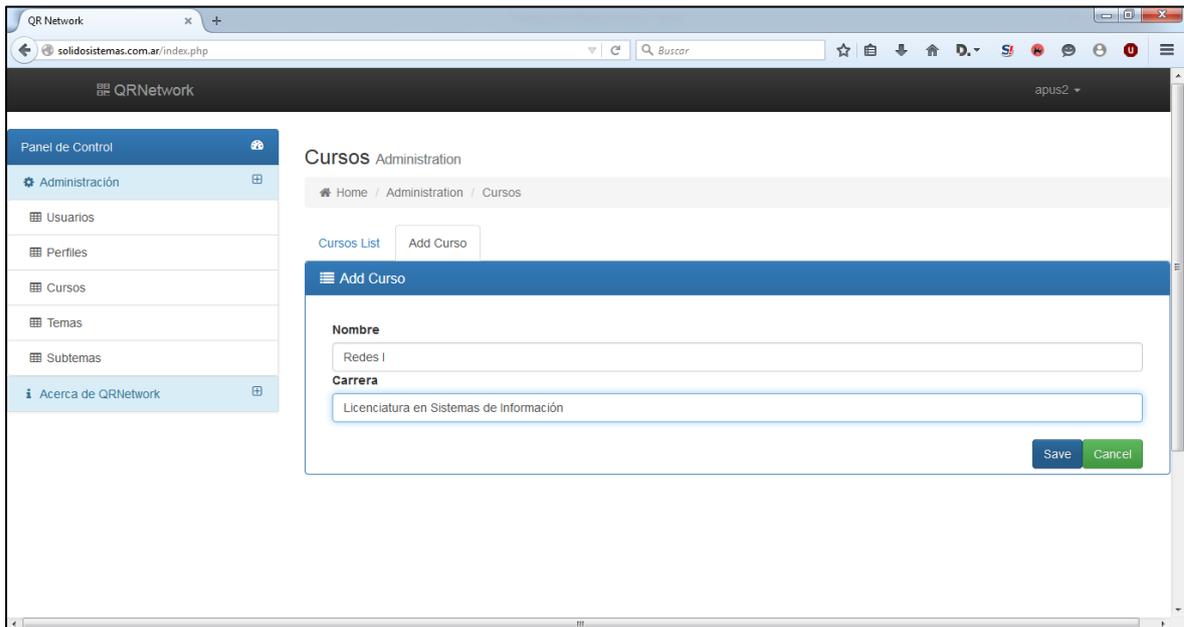


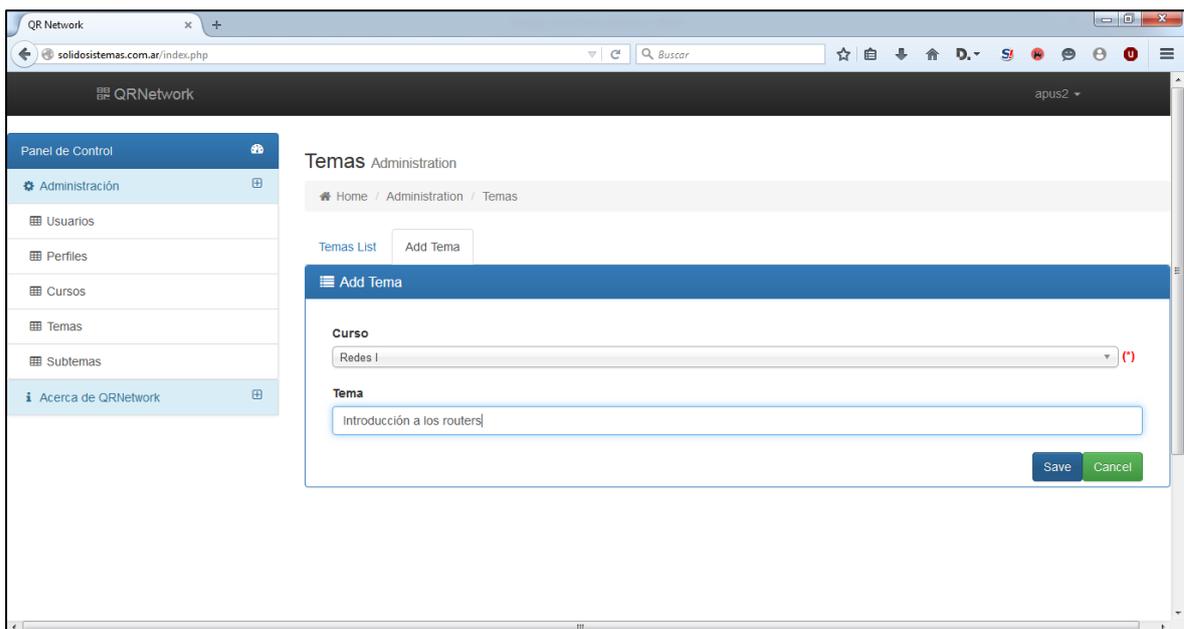
Figura A.12 - Administración de perfiles.

En las siguientes capturas observamos la carga de un curso y un tema relacionado al mismo.



The screenshot shows the 'Cursos Administration' page in the QR Network application. The left sidebar contains a 'Panel de Control' with options for 'Administración', 'Usuarios', 'Perfiles', 'Cursos', 'Temas', 'Subtemas', and 'Acerca de QRNetwork'. The main content area is titled 'Cursos Administration' and includes a breadcrumb trail 'Home / Administration / Cursos'. Below the breadcrumb are two buttons: 'Cursos List' and 'Add Curso'. The 'Add Curso' form is displayed, featuring two input fields: 'Nombre' with the value 'Redes I' and 'Carrera' with the value 'Licenciatura en Sistemas de Información'. At the bottom right of the form are 'Save' and 'Cancel' buttons.

Figura A.13 - Administración de Cursos (agregando el curso Redes I).



The screenshot shows the 'Temas Administration' page in the QR Network application. The left sidebar is identical to the previous screenshot. The main content area is titled 'Temas Administration' and includes a breadcrumb trail 'Home / Administration / Temas'. Below the breadcrumb are two buttons: 'Temas List' and 'Add Tema'. The 'Add Tema' form is displayed, featuring two input fields: 'Curso' with a dropdown menu showing 'Redes I' and a red asterisk icon, and 'Tema' with the value 'Introducción a los routers'. At the bottom right of the form are 'Save' and 'Cancel' buttons.

Figura A.14 - Agregando un nuevo Tema asociado al curso Redes I.

A.4. SPRINT IV – Módulo de Adaptación

El Incremento de este Sprint se compone por un lado del panel de control ampliado con las opciones para administrar los estilos de aprendizaje, los niveles de conocimiento y los componentes de red. Además, se mejora la función alta de usuario permitiendo agregar automáticamente el nivel de conocimiento predeterminado (bajo) y el estilo de aprendizaje por defecto sensitivo-visual ya que predomina más que el intuitivo-verbal según el trabajo de Kinshuk (2004), luego esta información puede ser actualizada por el administrador (docente). Por otra parte, obtenemos el módulo de adaptación que implementa la técnica de *texto condicional*, la técnica de *texto expandible* y la técnica de *variante de páginas*. A continuación se presentan algunos resultados.

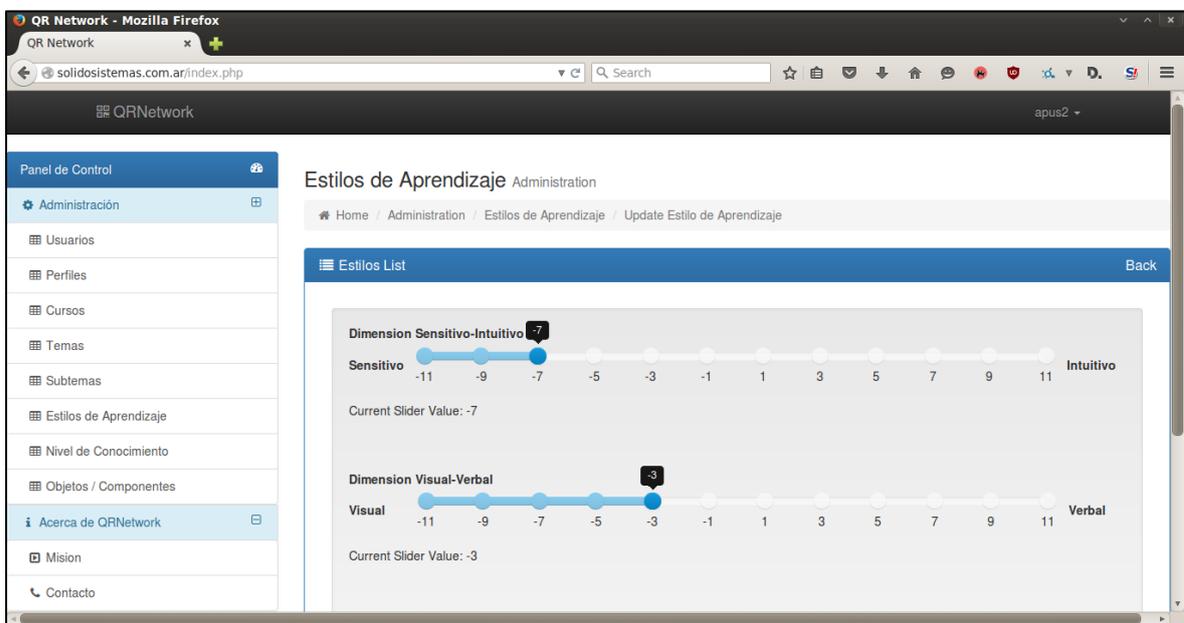


Figura A.15 – Actualizando el estilo de aprendizaje de un estudiante.

En la Figura A.15 se presenta la actualización de los valores asociados al estilo de aprendizaje de un estudiante.

En las Figuras A.16 y A.17 se muestra la actualización del nivel de conocimiento y la carga de la información para un componente de red.

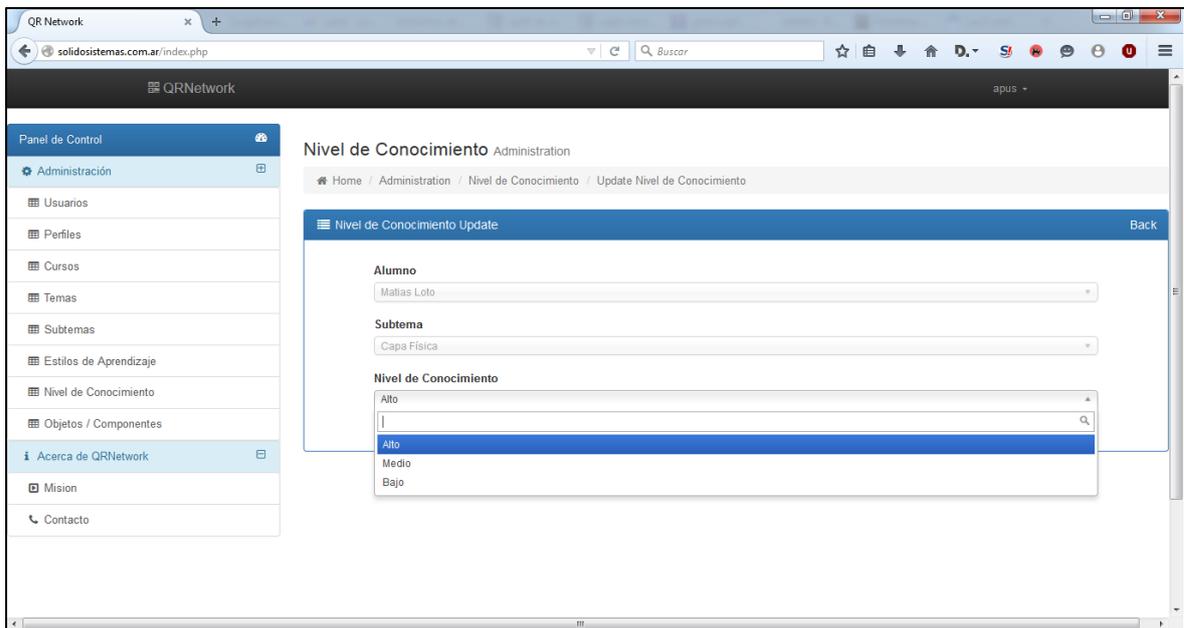


Figura A.16 – Actualizando el Nivel de Conocimiento de un estudiante respecto a un Subtema.

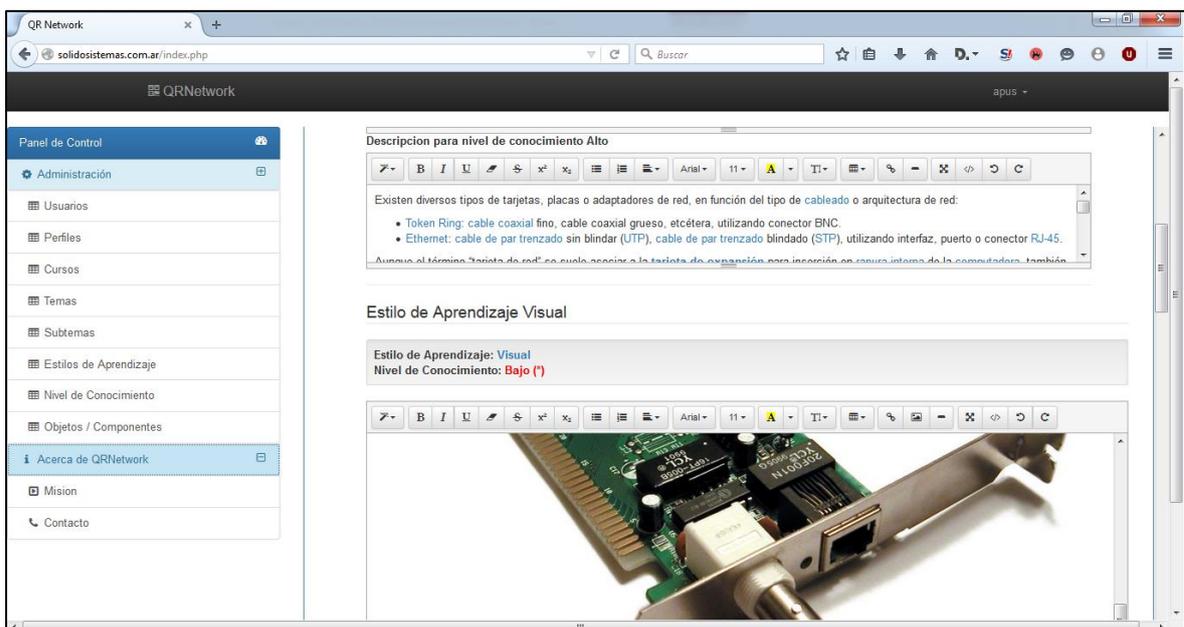


Figura A.17 - Cargando información de un componente (Tarjeta de red).

En las figuras A.18 y A.19 se observa a la aplicación presentando la información generada por el módulo de adaptación para un alumno con estilo de aprendizaje sensitivo y visual.



Figura A.18 – Información de una Tarjeta de red generada por el módulo de adaptación.



Figura A.19 – Información personalizada presentada para un componente RJ-45.

A.5. SPRINT V – Release

En este Sprint se implementan las últimas tareas necesarias para el despliegue de la aplicación, entre ellas la generación de scripts de recuperación del sistema en caso de fallos, tareas relacionadas con las bases de datos en producción, documentación y pruebas finales. El incremento de este Sprint es la aplicación terminada completamente funcional.

A continuación se presentan algunas pantallas con las pruebas finales y la carga de componentes en la base de datos.

```

root@bt: ~
File Edit View Terminal Help
root@bt:~# ab -g resultados.csv -c 35 -n 600 http://solidosistemas.com.ar/index.php/qrnetwork/adaptacion2/55545
This is ApacheBench, Version 2.3 <$Revision: 655654 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking solidosistemas.com.ar (be patient)
Completed 100 requests
Completed 200 requests
Completed 300 requests
Completed 400 requests
Completed 500 requests
Completed 600 requests
Finished 600 requests

Server Software:      nginx
Server Hostname:     solidosistemas.com.ar
Server Port:         80

Document Path:       /index.php/qrnetwork/adaptacion2/55545
Document Length:     45936 bytes

Concurrency Level:   35
Time taken for tests: 169.041 seconds
Complete requests:   600
Failed requests:     0
Write errors:        0
Total transferred:   28078705 bytes
HTML transferred:   27678793 bytes
Requests per second: 3.55 [#./sec] (mean)
Time per request:    9860.737 [ms] (mean)
Time per request:    281.735 [ms] (mean, across all concurrent requests)
Transfer rate:       162.21 [Kbytes/sec] received

Connection Times (ms)
              min  mean[+/-sd] median  max
Connect:     400  999 558.1   836  5625
Processing:  2657 8090 5993.0  6098  74845
Waiting:     862  1810 1200.5  1377  10524
Total:       3162 9089 6109.1  7240  76673

Percentage of the requests served within a certain time (ms)
50%    7240
66%    9591
75%   10964
80%   11922
90%   16398
95%   18942
98%   23471
99%   32226
100%  76673 (longest request)

```

Figura A.20 - Pruebas de rendimiento con ApacheBench.

Carga de la base de datos con información de los componentes a ser reconocidos a través de códigos QR.

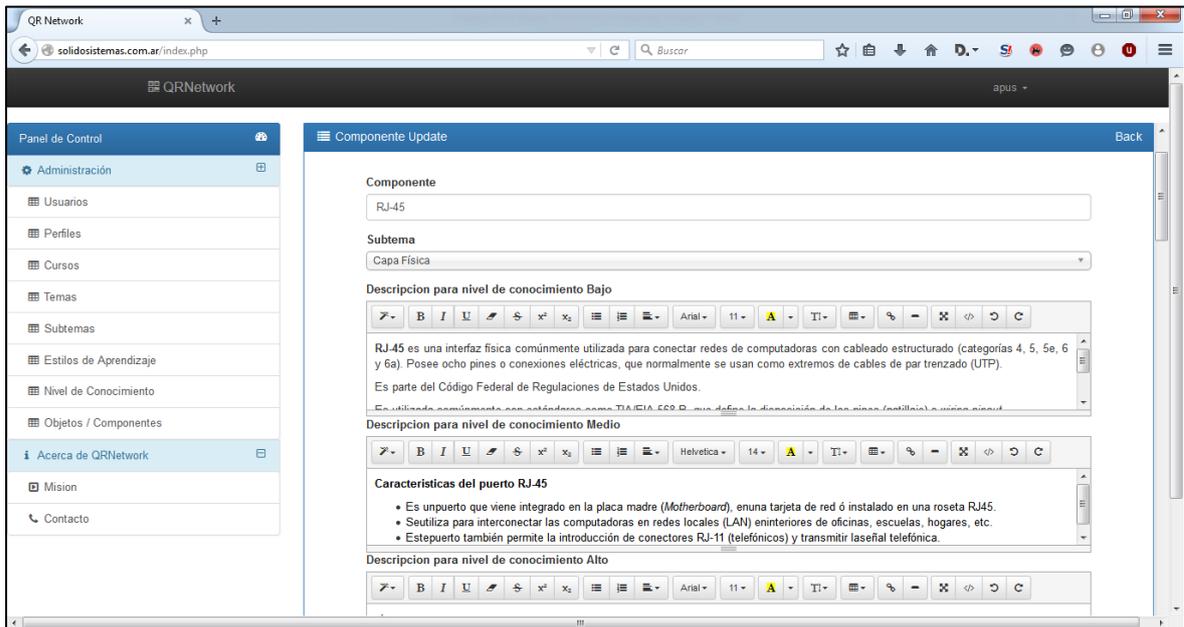


Figura A.21 - Cargando información de un conector RJ-45.

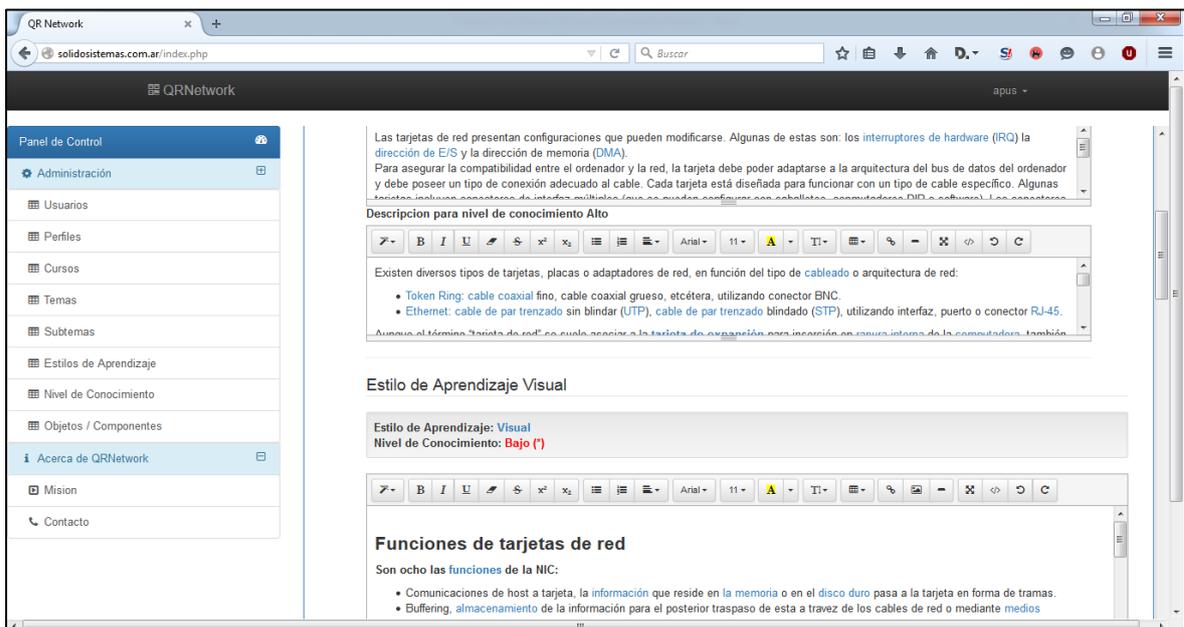


Figura A.22 - Cargando información de una Tarjeta de red.

