



UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO
FACULTAD DE CIENCIAS EXACTAS Y
TECNOLOGÍAS



LICENCIATURA EN SISTEMAS DE INFORMACIÓN

TRABAJO FINAL DE GRADUACIÓN

DESARROLLO DE UNA
APLICACIÓN UBICUA
PERSONALIZADA DE APOYO AL
CURSO DE INGRESO DE LA
FCEYT-UNSE

Autor: **Daniel Sebastian Montoto**

Profesor Guía: **Ing. Silvina Unzaga**

Profesor Asesor: **Dra. Elena Durán**

Abril 2018

UNIVERSIDAD NACIONAL DE SANTIAGO DEL ESTERO
FACULTAD DE CIENCIAS EXACTAS Y TECNOLOGÍAS

TRABAJO FINAL DE GRADUACIÓN DE LA LICENCIATURA EN SISTEMAS DE
INFORMACIÓN

“DESARROLLO DE UNA APLICACIÓN UBICUA PERSONALIZADA DE APOYO AL CURSO DE INGRESO DE LA FCEYT- UNSE”

Autor:

.....
Daniel Sebastian Montoto

Profesor Guía:

.....
Ing. Silvina Unzaga

Profesor Asesor:

.....
Dra. Elena Durán

Aprobado el día del mes de del año 2018

por el Tribunal integrado por

.....
.....

∞ Santiago del Estero – Argentina ∞

AGRADECIMIENTOS

A la Ing. Silvina Unzaga y a la Dra. Elena Durán por la predisposición, dedicación y confianza brindada para culminar este trabajo.

A mis padres José y Alicia por apoyarme siempre. A mis hermanos Nico y Ana.

A mis sobrinos, Máximo, Bauti y Tomi por alegrarme los días.

A mis tías y primos por estar presente siempre.

A Marcela Domski, por ser un pilar fundamental en mis proyectos y sueños.

A mis amigos y compañeros de la facultad. Sin ellos, el camino hubiese sido más largo y complicado. En la última instancia de la carrera, gracias a aquellos amigos, que a la distancia decidieron mantener el brazo firme colaborando desinteresadamente.

Al Centro de Estudiantes de la FCEyT por tantos momentos y aprendizajes compartidos.

A Miguel G. Ávila por los consejos y ayuda brindada.

A la Facultad de Ciencias Exactas y Tecnologías de la Universidad Nacional de Santiago del Estero por la colaboración brindada a lo largo de todos estos años. Gracias por mantener siempre las puertas abiertas.

Daniel Sebastian Montoto
Santiago del Estero, Argentina
Abril 2018

ÍNDICE DE CONTENIDO

RESUMEN.....	IX
CAPÍTULO 1: INTRODUCCIÓN	1
1.1. PLANTEAMIENTO DEL PROBLEMA.....	2
1.2. TRABAJOS RELACIONADOS.....	4
1.3. JUSTIFICACIÓN	6
1.4. OBJETIVOS	7
1.5. ESTRUCTURA DEL TRABAJO	8
CAPÍTULO 2: MARCO TEÓRICO	9
2.1 COMPUTACIÓN UBICUA	10
2.2. EL APRENDIZAJE Y LAS TECNOLOGÍAS MÓVILES Y UBICUAS	12
2.2.1. Las Tecnologías Móviles y Ubicuas	12
2.2.2 Aprendizaje Ubicuo.....	13
2.2.2.1. Características del aprendizaje ubicuo	15
2.2.2.2. Motivaciones del Aprendizaje Ubicuo	18
2.3. PERSONALIZACIÓN EN U-LEARNING	20
2.3.1. Concepto de Personalización.....	20
2.3.2. Enfoques de Personalización.....	22
2.3.3. Modelo del Estudiante.....	23
CAPÍTULO 3: METODOLOGÍA DE DESARROLLO	27
3.1. INTRODUCCIÓN	27
3.2. SCRUM.....	28
3.3.1. Visión General.....	29
3.3.2. El Equipo Scrum (<i>scrum team</i>)	31
3.3.2.1. El Dueño del Producto (Product Owner)	32
3.3.2.2. El Equipo de Desarrollo (<i>Development Team</i>).....	32
3.3.2.3. El Scrum Master.....	33
3.3.3. Eventos de Scrum.....	35
3.3.3.1. El Sprint.....	35
3.3.3.2. Planificación del Sprint (<i>Sprint Planning</i>)	36
3.3.3.3. Objetivo del Sprint (<i>Sprint Goal</i>).....	38
3.3.3.4. Scrum Diario (<i>Daily Scrum</i>).....	39
3.3.3.5. Revisión de Sprint (<i>Sprint Review</i>)	40
3.3.3.6. Retrospectiva de Sprint (<i>Sprint Retrospective</i>)	41
3.3.4. Artefactos de Scrum.....	42
3.3.4.1. Lista de Producto (<i>Product Backlog</i>)	42

3.3.4.2. Lista de Pendientes del Sprint (<i>Sprint Backlog</i>).....	44
3.3.4.3. Incremento.....	45
3.3.5. Transparencia de los Artefactos	45
3.3.6. Definición de “Terminado” (<i>Definition of “Done”</i>).....	46
3.3. JUSTIFICACIÓN DEL USO DE SCRUM.....	46
CAPÍTULO 4: DESARROLLO DE LA APLICACIÓN.....	49
4.1. DESARROLLO DEL PROCESO SCRUM.....	50
4.2. LISTA DE PRODUCTO (PRODUCT BACKLOG)	52
4.3. DEFINICIÓN DE TERMINADO.....	55
4.4. SPRINT 1: CONTROL DE ACCESO APLICACIÓN MOVIL.....	56
4.4.1 Reunión de planificación del sprint 1.....	56
4.4.2. Objetivo del Sprint 1	56
4.4.3. Lista de Pendientes del Sprint 1	57
4.4.4. Incremento del Sprint 1	57
4.4.5. Revisión y Retrospectiva del Sprint 1	58
4.5. SPRINT 2: GEOLOCALIZACIÓN Y RECOMENDACIÓN DE PUNTOS DE INTERÉS	59
4.5.1. Reunión de planificación del sprint 2.....	59
4.5.2. Objetivo del Sprint 2	60
4.5.3. Lista de Pendientes del Sprint 2	60
4.5.4. Incremento del Sprint 2	60
4.5.5. Revisión y Retrospectiva del Sprint 2	61
4.6. SPRINT 3: MÓDULO DE PERSONALIZACIÓN	63
4.6.1. Reunión de planificación del sprint 3.....	63
4.6.2. Objetivo del Sprint 3	64
4.6.3. Lista de Pendientes del Sprint 3	64
4.6.4. Incremento del Sprint 3	65
4.6.5. Revisión y Retrospectiva del Sprint 3	68
4.7. SPRINT 4: CONTROL DE ACCESO APLICACIÓN WEB	70
4.7.1. Reunión de planificación del sprint 4.....	70
4.7.2. Objetivo del Sprint 4	70
4.7.3. Lista de Pendientes del Sprint 4	71
4.7.4. Incremento del Sprint 4	71
4.7.5. Revisión y Retrospectiva del Sprint 4	73
4.8. SPRINT 5: APLICACIÓN WEB.....	74
4.8.1. Reunión de planificación del sprint 5.....	74
4.8.2. Objetivo del Sprint 5	74
4.8.3. Lista de Pendientes del Sprint 5	75

4.8.4. Incremento del Sprint 5	77
4.8.5. Revisión y Retrospectiva del Sprint 5	80
4.9. SPRINT 6: DESPLIEGUE.....	81
4.9.1. Reunión de planificación del sprint 6.....	81
4.9.2. Objetivo del Sprint 6	81
4.9.3. Lista de Pendientes del Sprint 6	82
4.9.4. Incremento del Sprint 6	82
4.9.5. Revisión y Retrospectiva del Sprint 6	84
4.10 HERRAMIENTAS DE SOFTWARE UTILIZADAS	85
CAPÍTULO 5: PRUEBAS Y ANÁLISIS DE RESULTADOS	87
5.1. INTRODUCCIÓN	87
5.2. PRUEBAS MANUALES A LA APLICACIÓN ANDROID.....	88
5.3. PRUEBAS DE RENDIMIENTO.....	90
5.4. PRUEBAS DE ESTRÉS	93
5.5. PRUEBAS DE USABILIDAD	95
5.5.1. Cuestionario	96
5.5.2. Procesamiento de las Respuestas al Cuestionario	99
5.6. ANÁLISIS DE RESULTADOS	101
CAPÍTULO 6.....	103
CONCLUSIONES.....	103
REFERENCIAS	105
ANEXO I.....	109
MANUAL DE USUARIO DE LA APP MÓVIL	109

RESUMEN

El desarrollo de la tecnología en comunicaciones inalámbricas y dispositivos móviles ha experimentado un gran avance en los últimos tiempos, permitiendo el desarrollo de la computación ubicua. Por otra parte, al aprendizaje tradicionalmente se lo ha considerado como un proceso mediante el cual el aprendiz entra en contacto y adquiere conocimientos o destrezas de alguna fuente autorizada. Esta definición ya no es suficiente para describir al aprendizaje en la sociedad de la información. Las teorías actuales sostienen que los aprendices no absorben de forma pasiva el conocimiento, sino que más bien lo construyen de forma activa. Desde el enfoque de estas nuevas teorías del aprendizaje sumado al desarrollo de la computación ubicua que permite aprender en cualquier momento y lugar, emerge un nuevo modelo denominado aprendizaje ubicuo. Este aprendizaje se caracteriza por la movilidad, el reconocimiento de la ubicación, el conocimiento de la situación contextual, la adaptabilidad, la inmediatez, la accesibilidad y la personalización de la información. En el mismo sentido, los procesos educativos están relacionados con la situación que involucra al estudiante, posibilitando el acceso a los recursos educativos personalizados de acuerdo al perfil y al contexto de aprendizaje.

En el presente trabajo se abordaron conceptos relacionados principalmente al aprendizaje ubicuo y personalización en ambientes de aprendizaje ubicuo, los que dan sustento a lo aquí realizado. Asimismo, aprovechando las ventajas de la metodología ágil Scrum, se aplicó la misma para el desarrollo de una aplicación ubicua, basada en tecnología móvil para Smartphones que cuenten con el sistema operativo Android. La aplicación es capaz de recomendar puntos de interés cercanos dentro de la universidad a través de la geolocalización. Una vez que el estudiante selecciona un punto de interés, la aplicación recomienda objetos de aprendizaje en base al tiempo disponible del estudiante y los conocimientos previos que este posee. El contexto para él que fue diseñada la aplicación, es

el Taller de Ambientación Universitaria perteneciente al curso de ingreso de la Facultad de Ciencias Exactas y Tecnologías de la UNSE.

Se presentan, además, las cuatro pruebas realizadas a la aplicación móvil: prueba manual, prueba de rendimiento, prueba de estrés y prueba de usabilidad. El resultado de todas las pruebas fue exitoso, permitiendo detectar y corregir errores, medir el uso de la memoria y el procesador, verificar el funcionamiento al máximo rendimiento y por último conocer opiniones sobre la interfaz y el funcionamiento de la aplicación. Finalmente, se formula posibles líneas de trabajo a futuro.

Palabras claves: Aprendizaje ubicuo, Personalización, Geolocalización.

CAPÍTULO 1

INTRODUCCIÓN

En los últimos años se está explorando un nuevo modelo de educación llamado aprendizaje ubicuo (u-learning). Se desarrolla en un escenario caracterizado por la movilidad de los estudiantes que llevan los dispositivos móviles y los recursos de acceso a través de redes inalámbricas y necesitan un acceso uniforme e inmediato a la información. El desarrollo de aplicaciones u-learning requiere considerar las características y necesidades de los usuarios, las múltiples formas de movilidad, la tecnología móvil, la diversidad de los transportistas, y también, la diversidad de escenarios de aprendizaje que pueden presentarse.

Los ambientes de aprendizaje ubicuo superan las limitaciones de una clase o ambiente de aprendizaje tradicional, y extienden el aprendizaje haciendo realidad la idea de aprendizaje en cualquier lugar y en cualquier momento, permitiendo a las personas acceder a mejores experiencias de aprendizaje en sus ambientes de la vida diaria. El uso de dispositivos tales como teléfonos móviles generan nuevas oportunidades para los estudiantes permitiéndoles estar conectados intensamente. Por lo tanto, los contenidos educativos pueden accederse y las interacciones pueden concretarse donde los estudiantes lo necesiten, en diferentes campos de la vida diaria, sin restricción de espacio ni de tiempo (Sabine & Kinshuk, 2008).

En los ambientes ubicuos es fundamental proporcionar una educación personalizada. Personalización en el contexto de la Informática, se refiere a la habilidad de un sistema o aplicación de adaptarse a la situación actual de cada estudiante para satisfacer sus necesidades (Sabine & Kinshuk, 2008). Asimismo, es posible ofrecer información personalizada a través de Objetos de Aprendizaje (OA). Se entiende por objeto de

aprendizaje a una entidad digital, autocontenible y reutilizable que tiene un propósito educativo (Chiappe Laverde, Segovia Cifuentes, & Rincón Rodríguez, 2007).

1.1. PLANTEAMIENTO DEL PROBLEMA

La Universidad Nacional de Santiago del Estero (UNSE) realiza anualmente el curso de ingreso a las diferentes carreras que esta ofrece. Entre los talleres que forman parte del curso de ingreso de la FCEyT se encuentra el Taller de Ambientación Universitaria. “Este taller se desarrolla en dos instancias, el primero se desarrolla al inicio del Curso de Ingreso y tiene como objetivo introducir a los alumnos en aspectos organizativos y relativos a la dinámica peculiar de la institución universitaria, y el desarrollo del Curso de Ingreso. Se aborda información general de la UNSE y de la FCEyT e Introducción a la vida profesional. La segunda instancia se desarrolla con los Ingresantes de la FCEyT (aspirantes que aprueban el ingreso de las carreras de Grado y Pregrado) y tiene como finalidad permitir a los mismos incorporarse adecuadamente a la institución UNSE, a la FCEyT y particularmente a la carrera elegida. En esta instancia se propone brindar los conocimientos sobre los aspectos académicos y administrativos de la vida universitaria: Inscripciones, Sistema SiuGuaraní, Plan de estudio, Sistema de Correlatividades, Becas, Horarios de clases, etc.” (Honorable Consejo Directivo, 13 de Diciembre de 2016).

Estas instancias de aprendizaje se caracterizan por:

- La variedad de temas administrativos que se abordan.
- La cantidad de estudiantes que ingresan a las carreras de la Facultad.
- El breve tiempo áulico para el desarrollo de los temas.
- La amplia cantidad de secciones/sectores de la universidad donde se realizan diferentes funciones y actividades que los estudiantes deben conocer.
- La no obligatoriedad del taller lo cual implica que muchos estudiantes decidan no asistir.
- El acotamiento de contenidos que el taller realiza cada año a la situación inmediata de necesidad del estudiante.
- La entrega de una cartilla en formato impreso con información indispensable, la cual no brinda las posibilidades que permite un recurso digital en un dispositivo móvil.

Como se menciona anteriormente, en el marco del Taller de Ambientación Universitaria, los estudiantes requieren aprender aspectos académicos y administrativos de la vida universitaria. Posteriormente al taller, ante una necesidad específica los estudiantes deben remitirse a:

- Página web de la facultad, que ofrece información sobre el calendario académico, acceso al siu-guaraní, reglamentos en general, planes de estudio de cada carrera, etc.
- GaME (Gabinete de Mediación Educativa), ofrece un seguimiento continuo por parte del equipo conformado en dicha área. Además les proporciona a los ingresantes en el taller, una cartilla con información indispensable sobre inscripciones, autoridades de la facultad, síntesis del reglamento alumno, funciones de cada área, etc.

Tanto la página web como el GaME ofrecen información de carácter general. En caso de requerir información personalizada el estudiante debe recurrir a su tutor de par (estudiante avanzado de la carrera que presta asistencia a los estudiantes).

Además, se torna desafiante integrar tecnologías en los procesos de aprendizaje de los estudiantes que ingresan a la universidad, para favorecer la colaboración; la conexión de los espacios formales, no formales e informales; resituar la ubicación del aprendizaje, tanto dentro como fuera del aula; adaptar los contenidos, presentaciones y actividades a las características de los estudiantes y su contexto; entre otras cosas.

Ante lo expuesto, surge como una alternativa válida la aplicación del modelo de aprendizaje ubicuo. Este modelo designa al conjunto de actividades formativas, apoyadas en tecnología, que están accesibles en cualquier lugar y desde cualquier dispositivo permitiendo un aprendizaje situado, creativo, novedoso, con un alto grado de autonomía, donde el estudiante no está sujeto a las limitaciones de espacio y tiempo (Saadiyah, Arniza Ahmad, & Abd Jalil, 2010). Se debe notar que, las características más importantes del aprendizaje ubicuo son: la movilidad, el reconocimiento de ubicación, el conocimiento de la situación contextual, la adaptabilidad, la inmediatez y la accesibilidad.

Este concepto deriva de «computación ubicua» (Weiser, 1991), que es la integración de la informática en el entorno vital de las personas. El término “ubicuo” está relacionado con la noción “en cualquier momento y en cualquier lugar”, sumado el concepto

“aprendizaje”, se transforma en un conocimiento de “cualquier cosa” (Cope & Kalantzis, 2010).

El aprendizaje ya no sucede únicamente en las aulas, sino que puede desarrollarse en múltiples espacios, facilitando el conocimiento en el momento (Burbules N. C., 2014). De esta manera, estamos ante un conocimiento global, instantáneo e interconectado (VázquezCano, 2015), facilitado por la tecnología ubicua.

A partir de lo descrito anteriormente, y en el marco del proyecto SICyT-UNSE “*Métodos y Técnicas para desarrollos de Aplicaciones Ubicuas*”, este trabajo final propone desarrollar una aplicación de apoyo al aprendizaje ubicuo que permita a través de la geolocalización, recomendar puntos de interés (PI) cercanos dentro de la universidad. Entendiendo como PI aquellos sitios sobre los que el estudiante debe concretar algún aprendizaje. Cada PI tendrá asociados Objetos de Aprendizaje. Una vez seleccionado el PI por parte del estudiante, la aplicación recomendará los OA de forma personalizada en base a dos variables: los conocimientos previos del estudiante y el tiempo disponible que este tiene para realizar las tareas propuestas en el OA. Para ello, se trabajará con los metadatos de los OA y los datos del perfil del estudiante.

1.2. TRABAJOS RELACIONADOS

Debido al avance en la tecnología de los dispositivos móviles en los últimos años, se encuentra en gran difusión e investigación, el desarrollo de aplicaciones móviles que brinden apoyo al área educativa. Un ejemplo es el trabajo de Martín et al. en (Martín, y otros, 2010) que desarrolla una aplicación móvil mediante la cual se carga objetos de aprendizaje encapsulados con el estándar LOM según el lugar, momento y perfil que tenga el usuario. Dicha aplicación está desarrollada utilizando el framework M2Learn, diseñado para facilitar la creación de aplicaciones móviles que hagan uso de geolocalización, plataformas educativas, estándares, y herramientas de comunicación y colaboración.

Chen y Huang en (Chen & Huang, 2012) proponen un sistema de aprendizaje ubicuo (CAULS) sensible al contexto, basado en la identificación por radiofrecuencia (RFID), red inalámbrica, dispositivo de mano y las tecnologías de bases de datos para detectar y analizar el mundo real y aprender conductas de los estudiantes. Por otro lado, Ovalle et al. en (Ovalle, Salazar, & Duque, 2014) proponen un modelo de recomendación personalizada de recursos

educativos para cursos virtuales adaptativos que incorpora las bondades de la computación ubicua y de los agentes inteligentes. Dicho modelo busca proveer de información relevante y personalizada a los estudiantes sobre la planificación de sus cursos virtuales, evaluación en línea, búsqueda y recuperación de objetos de aprendizaje.

(El-Bishouty, Ogata, & Yano, 2008) presentan un entorno de aprendizaje ubicuo, colaborativo y personalizado (PERKAM) para apoyar al alumno a realizar una tarea. Este entorno puede reconocer los objetos ambientales que rodean al alumno, los que usa durante su práctica de estudio. Utiliza etiquetas RFID para detectar los objetos espaciales físicos circundantes. Combina temas de los materiales educativos con la tarea actual del alumno y luego recomienda los mejores materiales coincidentes de acuerdo con la cantidad que coinciden con la tarea actual del alumno y cuán cerca están sus ubicaciones físicas de la ubicación actual del alumno. El sistema también establece una correspondencia entre la necesidad actual del alumno y los intereses y ubicaciones de los otros alumnos, recomienda los mejores ayudantes pares disponibles y visualiza las distancias relativas entre la necesidad actual del alumno y los intereses y ubicaciones de los compañeros colaboradores. El alumno puede referirse a algunos materiales educativos o ponerse en contacto con compañeros ayudantes, interactuar y colaborar.

López Domínguez et al. en (López Domínguez, Rodríguez Matla, & Pomares Hernandez, 2016) presentan el diseño y desarrollo de un sistema de aprendizaje móvil consciente de contexto (SAMCC) que incluye objetos de aprendizaje móviles para el desarrollo de competencias disciplinares básicas de matemáticas e informática en alumnos de educación media. El sistema está compuesto por dos módulos: uno de seguimiento del aprendizaje y otro de personalización de aprendizaje. El módulo de seguimiento del aprendizaje permite enviar avisos por SMS y redes sociales. Por otro lado, el módulo de personalización permite que el sistema sugiera objetos de aprendizaje móviles con material educativo pertinente y adecuado a cada situación con base en el estilo del aprendizaje y la actividad que desarrolla (contexto) el estudiante.

Escutia y Torres en (Escutia & Torres, 2014) desarrollaron una aplicación que consiste en obtener la posición física del usuario por medio del GPS. Una vez obtenida, el dispositivo ubicará en su base de datos el punto histórico más cercano. Una vez encontrado, imprimirá en la pantalla la información acerca del sitio histórico, una imagen representativa y una pequeña reseña acerca de los eventos ocurridos en éste. Por otro lado, (Cui & Bull, 2005) introduce TenseITS, un entorno de aprendizaje de idiomas que adapta la interacción

a la comprensión individual del alumno, la cual se representa en un modelo de aprendizaje construido durante la interacción. También se adapta de acuerdo con las características contextuales de la ubicación del alumno que pueden afectar su capacidad de estudio: la probabilidad de que sean interrumpidos por otras personas; su capacidad general para concentrarse en ese lugar; y la cantidad de tiempo que tienen disponible para estudiar. Para el caso particular del tiempo disponible, se modela en base a cuatro opciones: menos de 15 minutos, entre 15 y 30 minutos, entre 30 minutos y 1 hora, y más de 1 hora. TenseITS se diseñó principalmente para usar durante períodos cortos de tiempo (hasta 30 minutos), pero las otras opciones están disponibles ya que puede haber ocasiones en las que el alumno desee usar el sistema por más tiempo.

Finalmente, Thompsen et al. en (Thompsen Primo, Behr, & Vicari, 2013) analiza el uso de técnicas de web semántica como una alternativa a los sistemas de recomendación educativos tradicionales. El modelo propuesto investiga el uso de la información contenida en metadatos de objetos de aprendizaje y el perfil académico del estudiante descrito en la ontología de conocimiento para luego proporcionar una estructura del conocimiento. Este conocimiento es almacenado en un archivo OWL, lo cual le otorga características de interoperabilidad y conocimiento del dominio.

La revisión bibliográfica realizada para conocer el actual estado del arte refleja por un lado la importancia de conocer la ubicación geográfica de una persona y en base a eso, sugerirle recursos educativos o algún tipo de información. Por otro lado, se destaca la importancia de personalizar la información entregada al usuario teniendo en cuenta diferentes variables como el estilo de aprendizaje, preferencias, actividades contextuales, etc. En contraste a los trabajos mencionados anteriormente, este trabajo combina las ventajas del aprendizaje ubicuo al utilizar el GPS y recomendar Objetos de Aprendizaje de forma personalizada teniendo en cuenta, la ubicación, los conocimientos previos del estudiante y el tiempo disponible que este posee.

1.3. JUSTIFICACIÓN

El desarrollo de la aplicación ubicua propuesta se justifica ya que:

- Promueve la aplicación de las tecnologías de la información para facilitar y optimizar la autogestión y el desempeño académico y cotidiano de cada alumno.

- Actualmente en la UNSE no existe una aplicación que permita a los estudiantes del Curso de Ingreso acceder de forma novedosa, dinámica e intuitiva a información personalizada. Solo existen aplicaciones de carácter informativo.
- Creará en los alumnos una experiencia de recreación educativa a partir de la interacción con la aplicación y el entorno.
- Generará alternativas de aprendizaje situado y sensible al contexto usando como recurso los dispositivos móviles.
- Sentará las bases para un desarrollo futuro de una aplicación ubicua, que no solo sea aplicable al Curso de Ingreso sino también al desarrollo de las carreras de la universidad, permitiendo a los estudiantes interactuar con Objetos de Aprendizaje independientemente del tiempo y espacio en el que se encuentren.
- Permitirá a los ingresantes a la universidad adquirir los conocimientos necesarios para realizar los trámites académicos y administrativos para esta nueva etapa de su formación.

En resumen, el desarrollo de esta aplicación busca implementar un cambio tecnológico en nuestra facultad, con el objetivo de generar interés en los alumnos y apoyar tanto a docentes como estudiantes en los procesos de enseñanza y aprendizaje.

1.4. OBJETIVOS

Teniendo en cuenta el proyecto de investigación mencionado en el apartado 1.1 sobre el que se desarrolla el presente trabajo, se definen los objetivos generales y específicos que se busca alcanzar:

- **Objetivos Generales**
 - Contribuir al proceso de aprendizaje de los ingresantes universitarios, aportando una aplicación ubicua, que facilite un aprendizaje situado y personalizado, sobre los aspectos académicos y administrativos de la vida universitaria.
- **Objetivos Específicos**
 - Diseñar y desarrollar una aplicación ubicua que asista al estudiante en sus aprendizajes en el Taller de Ambientación Universitaria perteneciente al Curso de Ingreso de la FCEyT de la UNSE y que permita:
 - Recomendar los puntos de interés cercanos de acuerdo a la ubicación geográfica del estudiante.

- Recomendar Objetos de Aprendizaje, del punto de interés seleccionado por el estudiante, en base al conocimiento previo y tiempo disponible del estudiante para realizar la tarea.
- Desarrollar el módulo web para gestión de la aplicación por parte del Docente.

1.5. ESTRUCTURA DEL TRABAJO

Previamente se describió el contexto donde se realiza el presente trabajo de investigación y desarrollo, como así también el planteamiento del problema, antecedentes, justificación y por último los objetivos que persigue este trabajo final.

En el capítulo dos, se definen los conceptos en los que se fundamenta este trabajo, relacionados principalmente al aprendizaje ubicuo y personalización en ambientes de aprendizaje ubicuo. En el tercer capítulo, se detalla la metodología ágil *Scrum* utilizada para guiar el desarrollo de software de este trabajo. En el capítulo cuatro, se presenta el diseño y desarrollo de la aplicación utilizando la metodología detallada en el capítulo anterior. En el quinto y último capítulo, se exponen las pruebas realizadas y el análisis de los resultados.

Finalmente, se presentan las conclusiones del trabajo en donde se incluyen algunas consideraciones sobre trabajo a futuro.

CAPÍTULO 2

MARCO TEÓRICO

Vivimos en un mundo cada vez más digital, poblado por una gran cantidad de dispositivos digitales diseñados para ayudar y automatizar tareas y actividades más humanas, para enriquecer la interacción social humana y mejorar la interacción en el mundo real. El entorno del mundo físico está cada vez más instrumentado digitalmente y está repleto de dispositivos integrados de control y sensores. Estos pueden detectar nuestra ubicación y pueden adaptarse automáticamente a ella, lo que facilita el acceso a servicios localizados, por ejemplo, puertas abiertas y luces encendidas a medida que nos acercamos a ellas. Los sistemas de posicionamiento pueden determinar nuestra ubicación actual a medida que avanzamos. Se pueden vincular a otros servicios de información, es decir, para proponer un mapa de una ruta a nuestro destino. Además, se pueden usar dispositivos como tarjetas y llaves sin contacto para obtener acceso a servicios protegidos ubicados en el entorno.

En muchas partes del mundo, existen altas velocidades de transferencia a través de redes cableadas e inalámbricas para transferir contenido multimedia (texto, audio y video), en el trabajo y hogar, y para el uso de usuarios móviles y dispositivos fijos. El uso cada vez mayor de redes inalámbricas permite que se agreguen más dispositivos e infraestructura de forma poco sistemática y menos disruptiva en el entorno físico. Los circuitos y dispositivos electrónicos se pueden fabricar para que sean más pequeños, más baratos y puedan operar de manera más confiable y con menos energía. Hay una proliferación de dispositivos móviles inteligentes multiusos para acceder a servicios locales y remotos. Los teléfonos móviles pueden actuar como múltiples cámaras y reproductores de audio y video, como dispositivos de información y consolas de juegos. La interacción puede personalizarse y hacerse consciente del contexto del usuario al compartir modelos de personalización en nuestros

dispositivos móviles con otros servicios cuando interactuamos con ellos, por ejemplo, los dispositivos de audio y video pueden pre-programarse para mostrar solo los contenidos favoritos de una persona.

Muchos tipos de servicios destinados para apoyar las actividades humanas cotidianas relacionadas con la alimentación, la energía, el agua, la distribución, el transporte y la salud dependen en gran medida de las computadoras. En el pasado, si queríamos acceder y combinar múltiples servicios para soportar múltiples actividades, necesitábamos usar dispositivos de acceso separados. Por el contrario, las ofertas de servicios de hoy en día pueden proporcionar una prestación de servicios más integrada, interoperable y ubicua, por ejemplo, el uso de redes de datos para ofrecer también transmisiones de video y servicios de voz. Hay un gran margen para este tipo de desarrollos.

El término "ubicuo", que significa aparecer o existir en todas partes, combinado con computación, se transforma en la Computación Ubicua. Esta se utiliza para describir los sistemas TIC que permiten que la información y las tareas estén disponibles en todas partes, de manera intuitiva e invisible para el usuario (Posland, 2009).

2.1 COMPUTACIÓN UBICUA

El término original de computación ubicua fue acuñado por Mark Weiser en 1988 en el Centro de Investigación de Palo Alto (PARC) de Xerox, mientras se desempeñaba como director del Laboratorio de Ciencias de la Computación. Imaginó un futuro en el que las tecnologías informáticas se integraban en los artefactos cotidianos, apoyando las actividades diarias e igualmente aplicables a nuestros lugares de trabajo, hogares y mucho más (Krumm, 2009). Weiser en (Weiser, 1991) se refería a dispositivos computacionales que “desaparecen” del entorno y que el usuario utiliza para llevar a cabo sus tareas cotidianas, sin siquiera percatarse de ello.

Esta visión inicial de Weiser está hoy aún lejos de realizarse, pero una definición práctica aceptada se refiere a: una nueva tecnología que permite a las personas la utilización de dispositivos electrónicos tales como smartphones, tablets y computadoras portátiles en cualquier momento y en cualquier lugar a través de conexiones de red. Los dispositivos están equipados con sensores y actuadores, lo que les permite interactuar e intercambiar datos con el entorno (Saadiah, Arniza Ahmad, & Abd Jalil, 2010) (Hwang, Tsai, & Yang, 2008).

La computación ubicua es una disciplina informática emergente que existe en la intersección de la computación, las redes y los sistemas informáticos integrados. Hoy, casi todas las computadoras están conectadas a Internet. Por lo tanto, la computación ubicua se puede describir como un método que conecta las cosas restantes a Internet con el fin de proporcionar información sobre cualquier cosa en cualquier momento y desde cualquier lugar. Debido a que las cosas se vuelven cada vez más pequeñas, la computación ubicua se caracteriza por la omnipresencia de pequeñas computadoras interconectadas de forma inalámbrica que se integran casi invisiblemente en cualquier tipo de objeto u objeto cotidiano. Además de las computadoras pequeñas, los sensores también son cada vez más pequeños, lo que da como resultado sistemas informáticos integrados inteligentes capaces de detectar su entorno y equiparlos con capacidades de comunicación y procesamiento de la información.

Por lo tanto, la computación ubicua permite aplicaciones completamente nuevas (por ejemplo, dispositivos cooperativos que crean nuevas funcionalidades emergentes que pueden tener enormes implicaciones económicas y sociales). También permite cuestiones relacionadas con la aceptación de la tecnología y la creación de un mundo lleno de tecnología en el que la realidad se acopla estrechamente al ciberespacio basado en la información o incluso se fusiona con él. Esto permite que los dispositivos inteligentes cierren la brecha entre los sistemas informáticos integrados y los sistemas ciber-físicos.

En el mismo sentido, la creación de pequeños dispositivos informáticos que desaparecen en el medio ambiente es esencial para la introducción de la computación ubicua. Además, todas las computadoras deben poder comunicarse entre sí desde cualquier lugar a todas partes las 24 horas del día, los 7 días de la semana, así como también conocer o recordar sus posiciones. También deben conocer las preferencias y posiciones de los usuarios, deben ser intuitivamente utilizables y pueden usar, en el futuro cercano, la forma humana natural de comunicación.

Finalmente, la computación ubicua está creando un paradigma completamente nuevo de un entorno informático para conjuntos heterogéneos de dispositivos, incluidas computadoras invisibles integradas en cosas u objetos cotidianos, como dispositivos de automatización, ropa, dispositivos domésticos, dispositivos móviles, dispositivos personales, dispositivos de seguridad, vehículos, dispositivos de pared, dispositivos portátiles, etc., ubicados en entornos, edificios habitados, áreas seguras, instalaciones de producción y más. Estos dispositivos pueden tener diferentes sistemas operativos, interfaces

de red con sus protocolos requeridos y capacidades de entrada, como detección, seguimiento, control, salida y más (Möller, 2016).

2.2. EL APRENDIZAJE Y LAS TECNOLOGÍAS MÓVILES Y UBICUAS

2.2.1. LAS TECNOLOGÍAS MÓVILES Y UBICUAS

El impacto más relevante en la educación en el aula se debe a la disponibilidad de tecnologías móviles. Las tecnologías móviles permiten que la fuerza de trabajo del futuro se capacite en nuevos conceptos de aprendizaje, aumentando su:

- Creatividad
- Iniciativa
- Capacidad de respuesta

Esto facilitará la adaptabilidad mejorando las habilidades de los agentes dentro del aula para: gestionar la incertidumbre, comunicarse a través y dentro de las culturas, subculturas y comunidades y por último, negociar conflictos.

El énfasis en el aprendizaje es la capacidad de seguir aprendiendo de forma duradera para toda la vida. Por lo tanto, el aprendizaje duradero o permanente utiliza oportunidades formales e informales de aprendizaje a lo largo de la vida de las personas para fomentar el desarrollo continuo, mejorar el conocimiento y las habilidades necesarias para el empleo y la realización personal a través de:

- Aprender a saber: dominar el aprendizaje en lugar de la adquisición de conocimiento estructurado.
- Aprender a hacer: capacitar a las personas para los tipos de trabajo que se necesitan ahora y en el futuro, incluida la innovación y la adaptación del aprendizaje sobre los entornos de trabajo futuros.
- Aprender a ser: educación que contribuye al desarrollo completo de una persona: mente y cuerpo, inteligencia, sensibilidad, apreciación estética y espiritualidad.

Utilizando tecnologías móviles, el aprendizaje puede considerarse como situado, colaborativo, ubicuo o de por vida, complementando los conocimientos actuales para

actualizarse con los últimos conocimientos en el área de la concentración. Además, las tecnologías móviles permiten compartir el conocimiento con otros, independientemente de su ubicación. Por lo tanto, el aprendizaje se vuelve ubicuo con respecto a las tecnologías móviles integradas en dispositivos digitales o unidades que realizan funciones orientadas a los humanos. Estos dispositivos o unidades también se vuelven más durables cuando se trata de almacenar contenido en cualquier formato o versión que permita crear una compatibilidad con versiones anteriores, permitiendo organizar y preservar los registros digitales del aprendizaje humano durante toda la vida.

El continuo crecimiento de las tecnologías móviles, el aumento de la disponibilidad de Internet en otros lugares y la transformación constante de los servicios de software y telecomunicaciones han brindado la oportunidad de conectar todo con cualquier cosa. Una de las primeras oportunidades para surgir fue el concepto de computación móvil y ubicua. Por lo tanto, la computación ubicua permite que las computadoras se integren en todas partes. Las primeras formas de redes informáticas ubicuas son evidentes en el uso generalizado de dispositivos móviles. Por lo tanto, por primera vez, hay más dispositivos en el mundo que personas, incluso un número creciente que solo se comunica con otras máquinas.

A medida que las computadoras se hicieron ubicuas, dejaron de ser el foco de actividad, lo que les permitió desaparecer en el fondo. La introducción de la computación ubicua al aprendizaje ubicuo (u-Learning) da como resultado que los estudiantes usen su tecnología móvil para conectarse con los múltiples dispositivos integrados digitales y servicios disponibles. Por lo tanto, en un aula de aprendizaje ubicua, los estudiantes exploran el espacio ubicuo y se conectan mediante la tecnología móvil para interactuar con los diversos dispositivos y servicios digitales integrados. Además, el aprendizaje ubicuo tiene el potencial de mejorar la educación de una manera sostenible y eliminar muchas de las limitaciones de la educación tradicional, por ejemplo, permitir la personalización en relación con las necesidades de los estudiantes y construir la base de una comunidad ubicua basada en la tecnología móvil (Möller, 2016).

2.2.2 APRENDIZAJE UBICUO

Tradicionalmente, el aprendizaje se ha configurado como un proceso mediante el que el aprendiz entra en contacto y absorbe como si fuera una esponja- conocimiento o destrezas,

de alguna fuente autorizada. Esta definición ya no es suficiente para describir la convergencia de las condiciones del conocimiento en la sociedad de la información. Las teorías más avanzadas sobre el aprendizaje sostienen que los aprendices no absorben de forma pasiva el conocimiento personalmente significativo, sino que más bien lo construyen de forma activa, a partir de su experiencia del mundo (Cope & Kalantzis, 2010). Al mismo tiempo, cada vez hay más dispositivos portátiles que funcionan como computadoras tradicionales tales como smartphones, smarttv, smartwatch, sistemas de geolocalización, etc. Los estudiantes interactúan con estos dispositivos todo el tiempo, son cada vez más baratos, pequeños y portátiles y están completamente interconectados.

En el mismo sentido, los avances recientes en tecnologías inalámbricas y de sensores han llevado al desarrollo de nuevos entornos de aprendizaje, denominados entornos de aprendizaje ubicuo sensibles al contexto, capaces de percibir la situación de los estudiantes y proporcionarles apoyo adaptativo.

En los últimos años investigadores abocados al e-learning observaron que, junto al avance de la tecnología, los temas de investigación evolucionaron desde e-learning hacia el aprendizaje móvil (m-learning) y actualmente está evolucionando desde m-learning al aprendizaje ubicuo (u-learning) (Hwang, Tsai, & Yang, 2008).

El aprendizaje móvil se caracteriza 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/o todo dispositivo de mano que tenga alguna forma de conectividad inalámbrica. Permite acceder a recursos de aprendizaje desde cualquier lugar y en cualquier momento (Romero & Molina, 2010).

El aprendizaje ubicuo, desprendido del concepto de computación ubicua, va un paso más allá del m-learning, al incorporar consciencia del contexto.

Teniendo en cuenta los conceptos anteriores, Saadiah (2010) define al aprendizaje ubicuo como *“un paradigma de aprendizaje que se lleva a cabo en un ambiente de computación ubicua que permite aprender lo correcto en el lugar y tiempo adecuado de la manera correcta”*.

Para especificar un mayor nivel de detalle sobre la definición amplia de aprendizaje ubicuo, que se refiere a, *“aprendizaje en cualquier momento y en cualquier lugar”*, (Hwang 2008) propone un esquema que visualiza la relación existente entre los diferentes tipos de aprendizaje. El esquema se presenta en la figura 2.1.

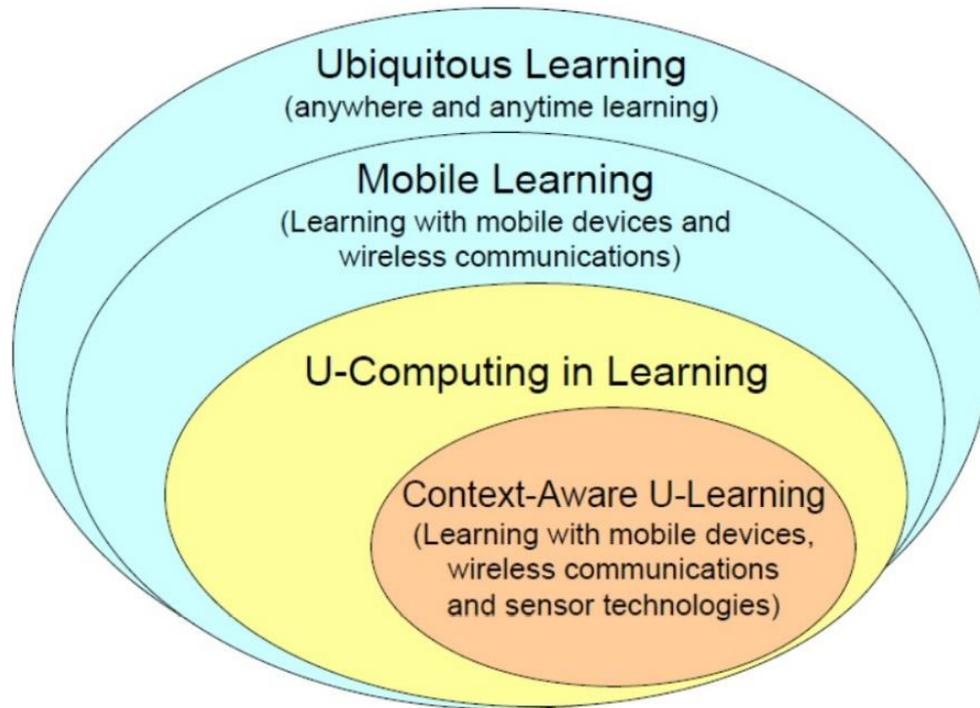


Figura 2.1: Relación entre aprendizaje ubicuo, aprendizaje móvil, computación ubicua en aprendizaje y “aprendizaje ubicuo sensible al contexto” (Hwang, Tsai, & Yang, 2008).

Por otro lado, en los ambientes de aprendizaje ubicuo los estudiantes se trasladan con sus dispositivos móviles que apoyan su proceso de aprendizaje al comunicarse con objetos embebidos y otros dispositivos del entorno. Los aspectos de personalización son muy importantes en los ambientes de aprendizaje ubicuo ya que permiten ofrecer servicios personalizados en función de las necesidades y preferencias de cada estudiante (Sabine & Kinshuk, 2008).

2.2.2.1. Características del aprendizaje ubicuo

(Chiu, Kuo, Huang, & Chen, 2008) propone las siguientes características:

- Urgencia de la necesidad de aprendizaje: el ambiente de aprendizaje ubicuo puede utilizarse para la urgencia del aprendizaje.
- Iniciativa de adquisición de conocimiento: los sistemas de u-learning pueden proporcionar la información necesaria a tiempo.

- **Interactividad del proceso de aprendizaje:** los estudiantes pueden comunicarse con sus compañeros, profesores y expertos de manera efectiva a través de las interfaces de los sistemas de aprendizaje ubicuo.
- **Situación de la actividad educativa:** en los ambientes de aprendizaje ubicuo, el proceso de aprendizaje puede integrarse en la vida cotidiana, así como los requisitos de conocimiento se presentan en un contexto auténtico.
- **Conciencia del contexto:** el ambiente de aprendizaje ubicuo es consciente del contexto, que se basa en los estados de los alumnos o las situaciones del entorno para proporcionar la información relacionada a los alumnos.
- **Proporciona servicios personalizado activamente:** teniendo en cuenta el contexto que rodea a los estudiantes, los sistemas de aprendizaje ubicuo brindan apoyo de manera activa en el camino y lugar correcto y en el momento adecuado.
- **Aprendizaje autorregulado:** en el cual los estudiantes pueden controlar activamente su progreso de aprendizaje por sí mismos.
- **Aprendizaje continuo:** el ambiente de aprendizaje ubicuo permite el aprendizaje sin interrupciones en cualquier lugar y cualquier momento. Los estudiantes pueden aprender mientras se trasladan de un lugar a otro.
- **Adaptar el contenido del tema:** el ambiente de aprendizaje ubicuo puede adaptar los contenidos de los temas a la capacidad de varios dispositivos de aprendizaje.
- **Comunidad de aprendizaje:** enriquecer la interacción de aprendizaje entre estudiantes y profesores a través de una comunidad en internet.

Por otro lado, y teniendo en cuenta las características propuestas por varios autores, (Saadiah, Arniza Ahmad, & Abd Jalil, 2010), presentan un resumen de las características principales del aprendizaje ubicuo. Se detallan a continuación:

- **Permanencia:** la información se encuentra almacenada y disponible a menos que los estudiantes decidan removerla.
- **Accesibilidad:** la información está siempre disponible cada vez que los estudiantes necesitan utilizarla.
- **Inmediatez:** permite a los estudiantes obtener información de manera inmediata.

- **Interactividad:** los estudiantes pueden interactuar con compañeros, profesores y expertos de manera eficiente y eficaz a través de diferentes medios de comunicación.
- **Conciencia de contexto:** es posible brindar información en base a la situación de los estudiantes y el entorno en el que se encuentran.

Por otra parte, (Burbules N. , 2012) destaca el aprendizaje situado como una característica relevante del aprendizaje ubicuo. El aprendizaje situado, en la perspectiva de la cognición situada es un producto de la interacción entre agentes y elementos del entorno; donde la utilización y adecuación de las herramientas con el medio circundante se tornan como premisas particulares en la construcción del conocimiento (Paz Penagos, 2007).

(Lave, 1991) y (Wenger, 2001) sostienen que el aprendizaje situado es un aspecto inseparable e integral de cualquier práctica social. La noción de aprendizaje situado indica el carácter contextualizado del aprendizaje que no se reduce a las nociones convencionales de aprendizaje in situ o aprendizaje haciendo, sino a la participación del aprendiz en una comunidad de práctica; esto es, en un contexto cultural, social, de relaciones, del cual se obtiene los saberes necesarios para transformarla y transformarse. (Paz Penagos, 2007) propone las siguientes características del aprendizaje situado:

- Es una experiencia social situada, que se enriquece con experiencias de otros, con recursos compartidos y con prácticas sociales comunes; en la cual el lenguaje juega un papel básico como herramienta mediadora.
- Se basa en el concepto que el conocimiento es contextual y situado, e influido por la actividad, el contexto y la cultura en la cual se utiliza.
- Desde la visión de la cognición situada, la enseñanza se debe centrar en prácticas educativas auténticas. La autenticidad de una práctica educativa está determinada por el grado de relevancia cultural de las actividades sociales, por las prácticas compartidas en las que participa el estudiante; así como del tipo y nivel de actividad social que éstas promueven.
- En la propuesta cognitiva situada es muy importante la mediación, la alineación por el adulto y otros compañeros, la negociación mutua de significados, la construcción

conjunta de los saberes y las estrategias que promuevan un aprendizaje, cooperativo, colaborativo o recíproco.

Burbules afirma que el aprendizaje ubicuo crea un recurso para apoyar el aprendizaje situado, a medida que la gente busca el conocimiento y las habilidades necesarias para hacer frente a situaciones reales e inmediatas a las que se enfrentan.

2.2.2.2. Motivaciones del Aprendizaje Ubicuo

A continuación, (Cope & Kalantzis , 2009) sugieren siete movimientos que son característicos del aprendizaje ubicuo. Cada uno explora y explota los potenciales de la informática ubicua.

- *Difuminar los límites tradicionales institucionales, espaciales y temporales de la educación.* En las instituciones de educación patrimonial de nuestro pasado reciente, los alumnos debían estar en el mismo lugar al mismo tiempo, realizando el mismo tema y manteniéndose en la misma página. Hoy en día, en la era de la grabación y la transmisión de bajo costo de cualquier contenido textual, visual y de audio en cualquier lugar, las “aulas tradicionales” no son imprescindibles. La educación puede ocurrir en cualquier lugar y en cualquier momento.
- *Cambiar el equilibrio.* En el aula tradicional, el docente y la pizarra se encuentran al frente de la sala. Los estudiantes se acomodan en filas rectas, responden preguntas uno a la vez o leen sus libros en silencio para completar el cuaderno de ejercicios. Este tipo de educación tenía cierto sentido para cierto tipo de mundo, un mundo donde los supervisores en el trabajo gritaban órdenes o transmitían memorandos sobre los aparentes intereses productivos de los trabajadores, donde los medios de comunicación contaban la historia principal que nos referíamos escuchar, y donde todos consumimos productos idénticos producidos en masa porque los ingenieros y empresarios habían decidido lo que sería bueno para nosotros. En los nuevos medios, la informática ubicua ha producido enormes transformaciones, ya no es necesario ser un receptor pasivo del conocimiento transmitido cuando los alumnos y los profesores pueden ser co-diseñadores colaborativos del conocimiento. Este es un fenómeno de difuminación entre las distinciones entre docentes y alumnos, y entre los creadores de conocimiento y los usuarios del conocimiento.

- *Reconocer las diferencias del alumno y usarlas como un recurso productivo.* Las sociedades modernas solían valorar la uniformidad: todos leíamos el mismo conjunto de periódicos y veíamos los mismos canales de televisión; todos consumimos los mismos productos. Y así fue en las escuelas: todos tenían que escuchar al docente al mismo tiempo, permanecer en el mismo mensaje en la misma página y hacer la misma prueba al final para ver si habían aprendido lo que el plan de estudios esperaba de ellos. El aprendizaje ubicuo ofrece una serie de posibilidades. No todos los estudiantes deben estar en la misma página; pueden estar en diferentes páginas de acuerdo a sus necesidades. Cada alumno puede conectar lo general y lo autoritativo con los detalles y detalles de sus propias experiencias e intereses de vida. Cada alumno puede ser un creador de conocimiento y un creador cultural, y en cada momento de creación y creación rehacen el mundo en el timbre de su propia voz y de una manera que se conecta con sus experiencias. Los estudiantes también pueden trabajar en grupos, como creadores de conocimiento colaborativo, donde la fortaleza del conocimiento del grupo surge de su capacidad de utilizar productivamente las complementariedades que surgen de sus diferencias. En este contexto, el docente deberá ser miembro activo de las comunidades de aprendizaje cosmopolitas y co-diseñadores, junto con los alumnos, de sus itinerarios de aprendizaje.
- *Ampliar y combinar los modos de representación.* La computación ubicua registra y transmite significados de manera multimodal: oral, escrita, visual y auditiva. Además, como nunca, no hay costo en la producción y transmisión de este material. Los educadores necesitarán comprender las diversas gramáticas de los múltiples modos de hacer significados que lo digital ha hecho posible, de la misma manera que las formas alfabéticas y simbólicas tradicionales.
- *Desarrollar capacidades de conceptualización.* El mundo de la computación ubicua está lleno de arquitecturas técnicas y sociales complejas que necesitamos poder leer para ser un usuario. Estos nuevos medios necesitan una sensibilidad particular de conceptualización, formas sofisticadas de reconocimiento de patrones y esquematización. Por estas razones (y por otras razones educativas mucho más antiguas, también), el aprendizaje ubicuo requiere una abstracción de orden superior y estrategias metacognitivas. Esta es la única manera de abrirse camino a través de las imposibilidades de la cantidad de información. Luego, los docentes deben convertirse en usuarios expertos de estas nuevas herramientas de creación de

significado, aplicando el metalenguaje que ellos y sus alumnos necesitan por igual para comprender sus posibilidades.

- *Conectar el propio pensamiento con la mente social de la cognición distribuida.* Hoy en día existe una inmediatez, inmensidad y navegabilidad del conocimiento que está disponible y accesible para los dispositivos que se han convertido más directamente en una extensión de nuestras mentes. Los educadores deberán crear nuevas medidas para evaluar las capacidades de los alumnos para saber cómo saber en este nuevo entorno.
- *Construir culturas colaborativas de conocimiento.* La informática ubicua invita a formas de reflexividad social que pueden crear "comunidades de práctica" para apoyar el aprendizaje. En el contexto del aprendizaje ubicuo, los docentes crean conocimientos con sus pares. Los estudiantes también involucran a personas que anteriormente se considerarían ajenas o incluso fuera de límites en el proceso de aprendizaje: padres y otros miembros de la familia, amigos críticos o expertos. Los espacios de trabajo digitales de las tecnologías de "redes sociales" son lugares ideales para este tipo de trabajo, a la vez simple y altamente transparente cuando se trata de auditar contribuciones diferenciales. Los docentes necesitarán habilidades superiores en la construcción de comunidades de aprendizaje si desean garantizar la inclusión y que todos los alumnos alcancen su potencial.

2.3. PERSONALIZACIÓN EN U-LEARNING

2.3.1. CONCEPTO DE PERSONALIZACIÓN

El aprendizaje personalizado presupone una enseñanza de alta calidad que se adapta a las diferentes formas en que los estudiantes logran sus conocimientos y habilidades. Por lo tanto, los cursos de enseñanza, los planes de estudios y las organizaciones escolares deben diseñarse de manera tal que lleguen al mayor número posible de estudiantes con diversas necesidades y experiencias durante el mayor tiempo posible. Los cursos personalizados involucran activamente a los alumnos al proporcionar estrategias de enseñanza y materiales que atraen el conocimiento y las preferencias de los alumnos, etc. Por su parte, un sistema educativo que responde a las necesidades individuales mediante la creación de un camino de aprendizaje personal permite a los estudiantes individuales experimentar la excelencia en su aprendizaje (Paneva & Zhelev, 2007).

La personalización incluye cómo encontrar y filtrar la información de aprendizaje que se ajusta a las preferencias y necesidades del usuario, cómo representarla y cómo dar herramientas al usuario para la reconfiguración de los sistemas.

El reto en un mundo rico en información no es proporcionar información en cualquier momento y en cualquier lugar, sino decir lo correcto en el momento adecuado y de la manera correcta para la persona adecuada. Como se mencionó en secciones anteriores, un entorno informático ubicuo permite a las personas aprender en cualquier momento y en cualquier lugar. Pero el problema fundamental es cómo proporcionar a los estudiantes la información correcta en el momento adecuado y de la manera correcta. Por lo tanto, un ambiente ubicuo debe personalizarse de acuerdo con la situación del alumno (El-Bishouty, Ogata, & Yano, 2008).

La personalización en el contexto de la Informática se refiere a la capacidad de un sistema o aplicación para adaptarse y satisfacer las necesidades de cada usuario; teniendo en cuenta, por ejemplo, su ubicación actual, conocimientos previos, tiempo disponible, nivel de conocimientos, estilos de aprendizaje, habilidades cognitivas, motivación, intereses, idioma preferido, etc.; y así lograr que los entornos de aprendizaje permitan al estudiante una experiencia más eficaz, conveniente y exitosa de aprendizaje (Sabine & Kinshuk, 2008).

(Cui & Bull, 2005) plantearon las condiciones para el sistema de aprendizaje personalizado de la siguiente manera:

- Individualizado de acuerdo con el conocimiento del alumno.
- Individualizado según la ubicación y las necesidades del alumno en esa ubicación.
- Movilidad, es decir, poder usarse en diferentes ubicaciones requeridas por el usuario.

El autor señala además que un aspecto relevante en la personalización es el tiempo que los estudiantes tendrán disponible para dedicar a una tarea, ya que a veces pueden tener solo un período muy corto de tiempo, mientras que otras veces pueden tener más tiempo. En el mismo sentido, (Ortigosa, Bravo, Carro, & Martin, 2010) indica que un sistema que permita gestionar el tiempo disponible para realizar actividades se convierte en un factor de especial importancia cuando los estudiantes solo disponen de unos pocos minutos. Uno de los bienes que resultan más escasos en esta sociedad es el tiempo, situación que obliga a los estudiantes a intentar sacar el máximo provecho de cualquier rato disponible.

Por su parte, (Paneva & Zhelev, 2007), plantea que la personalización puede examinarse en los siguientes aspectos:

- Personalización del contenido de aprendizaje, basado en las preferencias del alumno, formación y experiencia educativa, contenido de aprendizaje adaptado al estilo de aprendizaje individual del usuario;
- Personalización de la forma de representación y la forma del contenido de aprendizaje (por ejemplo, contenido de aprendizaje en forma de secuencias adaptativas de aprendizaje de objetos de aprendizaje).
- Personalización completa, que es una combinación de los dos tipos anteriores.

2.3.2. ENFOQUES DE PERSONALIZACIÓN

En los ambientes ubicuos es fundamental proporcionar una educación personalizada. Tornándose muy importantes aquellos sistemas que brindan la posibilidad de ofrecer servicios personalizados considerando la situación del estudiante. No obstante, en la mayoría de los sistemas de recomendación existentes, se utiliza sólo la información referida a las preferencias del usuario para proporcionar servicios personalizados, dejando de lado la información contextual (Park, Park, Kim, & Kang, 2010) que es altamente relevante en los ambientes ubicuos.

En cuanto a los enfoques de personalización para proporcionar servicios personalizados en entornos ubicuos, se consideran tres tipos principales (Park, Park, Kim, & Kang, 2010): el enfoque de personalización sensible al contexto, el enfoque de personalización basado en perfil de usuario y el enfoque híbrido. El primero ofrece servicios personalizados sobre la base de información contextual, siendo un enfoque estático. El segundo tipo de enfoque recomienda mediante el uso de similitud de usuarios e ítems. El tercero es una combinación de los dos anteriores.

Por otro lado, (Paneva & Zhelev, 2007) considera que los siguientes enfoques se pueden usar para aplicar la personalización del aprendizaje:

- *Personalización controlada por el alumno*. Requiere la entrada directa de las necesidades y preferencias del alumno al completar formularios de preguntas o al elegir opciones y alternativas.

- *Personalización, basada en un perfil de usuario existente y meta-descripciones del contenido de información.* En este caso, las preferencias de los alumnos se almacenan en su perfil.
- *Personalización mediante la búsqueda de una correlación entre los alumnos.* La correlación se realiza a través de los valores de los atributos, que describen el perfil del alumno. Si existe una fuerte correlación, existe la posibilidad de que el contenido para un perfil dado sea adecuado para aplicar a sus perfiles cercanos (adyacentes).

En este trabajo, para recomendar los Objetos de Aprendizaje, se adopta una estrategia basada en un enfoque de personalización híbrido, combinando la personalización basada en perfiles de usuario, que permite personalizar a las características relevantes de los estudiantes, en este caso los conocimientos previos; con la personalización sensible al contexto, que realiza la personalización en base al ambiente donde se concreta el aprendizaje, en este caso se considera la distancia. En ambos casos, se construyeron reglas lógicas para filtrar los aspectos relevantes a recomendar.

2.3.3. MODELO DEL ESTUDIANTE

Un aspecto clave para lograr la personalización en sistemas de aprendizaje ubicuo, es contar con un modelo de estudiante. Este modelo contiene información sobre conocimientos previos, comportamiento, ubicación, objetivos, estilos de aprendizaje, características personales, intereses y motivaciones, entre otras cosas (Froschl, 2005).

El modelo de estudiante permite que el sistema proporcione contenidos de cursos individualizados y orientación de estudio, sugiera objetivos de aprendizaje óptimos, determine los perfiles de los estudiantes y el conocimiento real que han adquirido, ensamble dinámicamente cursos basados en las necesidades individuales de capacitación y estilos de aprendizaje, y se unan profesores capaces de proporcionar apoyo en términos de orientación y motivación y, por lo tanto, para ayudar a los estudiantes con diferentes antecedentes y niveles de conocimiento a lograr sus objetivos de aprendizaje de manera efectiva.

A continuación, se presentan las categorías o dimensiones de información que forman parte del modelo del estudiante según distintos autores.

(Paneva & Zhelev, 2007) considera que el modelo de estudiante debe cubrir una cierta cantidad de información que se puede dividir en dos grupos principales:

- *Información general del alumno*, como objetivos de aprendizaje, aptitudes cognitivas, medidas para el estado de motivación, preferencias sobre el método de presentación, datos fácticos e históricos (información personal), etc.
- *Información sobre el comportamiento del alumno en el dominio del aprendizaje*, como nivel general de competencia del curso, nivel de competencia del módulo, nivel de competencia conceptual, tiempo de estudio del módulo, estado de resolución de pruebas, etc.

(Graf, Yang, Liu, & Kinshuk, 2009) tiene en cuenta las siguientes características:

- *Perfil*: el perfil del estudiante incluye solo información estática tal como, el nombre del estudiante, género, identificación del estudiante (ID), inicio del estudio, grado, programa de estudio y dirección de contacto.
- *Uso del sistema*: recopila información sobre cómo los estudiantes usan el sistema, qué servicios utilizan y cuándo utilizan el sistema o los servicios para el aprendizaje. Almacena el curso que está realizando el estudiante y un historial de cursos pasados, servicios y tiempos de aprendizaje preferidos (mañana, tarde, noche, etc.).
- *Progreso*: almacena información sobre los objetos de aprendizaje revisados y realizados como así también actividades de aprendizaje realizadas y revisadas. Además, se almacena el porcentaje de realización de objetos y actividades de aprendizaje. Por último, se registra la interacción en el foro, incluyendo preguntas, respuestas, etc.
- *Intereses y nivel de conocimiento*: los intereses y el nivel de conocimiento se almacenan a partir de medidas cualitativas. La información se puede extraer a partir de diferentes servicios como ser redes sociales, objetos y actividades de aprendizaje visitadas, desempeño en la solución de problemas de los cursos, etc.
- *Cercanía social*: incluye información sobre el nivel de familiaridad entre los estudiantes, indicando si se conocen entre sí, si ya han aprendido juntos y si están dispuestos a formarse juntos. Además, se almacena información sobre la preferencia general de colaboración de los estudiantes.

- *Habilidades para resolver problemas:* incluido el pensamiento crítico, el razonamiento científico, la motivación en la resolución de problemas, así como la capacidad general de resolución de problemas.
- *Estilos de aprendizaje:* a partir del modelo de Felder-Silverman que distingue cuatro dimensiones, activa / reflexiva, sensible / intuitiva, visual / verbal y secuencial / global.
- *Ubicación:* almacena información sobre la ubicación actual y ubicaciones pasadas sobre el estudiante. Se realiza a través de coordenadas GPS y pequeñas descripciones textuales sobre la ubicación.

CAPÍTULO 3

METODOLOGÍA DE DESARROLLO

3.1. INTRODUCCIÓN

La globalización ha afectado todos los sectores productivos y, en particular, la industria del software, lo cual ha requerido el desarrollo de nuevas metodologías que se adapten a las velocidades de los cambios y permitan construir rápidamente productos que satisfagan los requerimientos de los clientes. Es por esto por lo que el desarrollo de aplicaciones móviles se ha expandido en los últimos años debido, por un lado, a la masificación de los teléfonos móviles y al acceso a servicios de datos, y por otro, a la aparición de herramientas de software que facilitan su desarrollo (Babativa, Briceño, Nieto, & Salazar, 2016).

Asimismo, las metodologías ágiles han ganado popularidad desde hace algunos años, ya que constituyen una buena solución para proyectos a corto plazo, en especial, aquellos proyectos en dónde los requisitos están cambiando constantemente, un ejemplo de esto son las aplicaciones móviles, debido a que éstas tienen que satisfacer una serie de características y condicionantes especiales, tales como: canal, movilidad, portabilidad, capacidades específicas de las terminales, entre otras, y aun cuando existen miles de aplicaciones para dispositivos móviles que corren en diferentes sistemas operativos iOS, Android, BlackBerry y Windows Mobile (Amaya Balaguera, 2013). Si bien las metodologías ágiles han sido empleadas tradicionalmente en aplicaciones web y de escritorio, las características anteriores también las hacen apropiadas en el campo del software móvil.

Por otro lado, la aparición de las metodologías ágiles no puede ser asociada a una única causa, sino a todo un conjunto de ellas, si bien es cierto que la mayoría de autores lo relacionan con una reacción a las metodologías tradicionales, ¿cuáles fueron las causas de

esta reacción?, los factores que comúnmente se mencionan son la pesadez, lentitud de reacción y exceso de documentación, en definitiva, falta de agilidad de los modelos de desarrollo formales; otro punto importante sería la explosión de la red, las aplicaciones Web y las aplicaciones móviles, así como el crecimiento notorio del movimiento open source. A todo esto, se puede añadir un cambio bastante importante, en cuanto a la demanda del mercado del software, cada vez más orientada a la Web y a dispositivos móviles, con requisitos muy volátiles y en constante cambio, que requieren tiempos de desarrollo cada vez más cortos.

Las metodologías ágiles están basadas en un enfoque iterativo que se centra más en capturar mejor los requisitos cambiantes y la gestión de los riesgos, rompiendo el proyecto en iteraciones de diferente longitud, cada una de ellas generando un producto completo y entregable; e incremental donde un producto se construye bloque a bloque durante todo el ciclo de vida de desarrollo del producto, las iteraciones individuales deben producir alguna característica completamente funcional o mejorada. Su principal objetivo es reducir el tiempo de desarrollo (Amaya Balaguera, 2013).

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 e incrementos en el desarrollo a lo largo de todo el ciclo de vida del proyecto (Blanco, Camarero, Fumero, Werterski, & Rodríguez, 2009).

El presente capítulo está ordenado de la siguiente manera. En la segunda sección se detalla la metodología ágil *Scrum* a partir de la guía oficial de sus propios autores (Schwaber & Sutherland, 2016). En la tercera y última sección se fundamenta el uso de Scrum en el presente trabajo.

3.2. SCRUM

Scrum tiene sus orígenes a principios de los años 80, cuando Ikujiro Nonaka e Hirotaka Takeuchi identificaron y definieron el modelo a partir de analizar cómo desarrollaban los nuevos productos las principales empresas de manufactura tecnológica: Fuji-Xerox, Canon, Honda, Nec, Epson, Brother, 3M y Hewlett-Packard.

En su estudio, Nonaka y Takeuchi compararon la nueva forma de trabajo en equipo, con el avance en formación de scrum 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.

Scrum, como objetivo en los equipos de desarrollo de software, comenzó en la empresa Easel Corporation perteneciente a Jeff Sutherland en el año 1993. En 1995 Ken Schwaber formalizó el proceso al presentar “Scrum Development Process” en OOPSLA 95 (Object-Oriented Programming Systems & Applications conference), un marco de reglas para desarrollo de software, basado en los principios de scrum, y que Schwaber había empleado en el desarrollo de Delphi (Palacio, 2015).

Desde 1995 miles de proyectos en todo el mundo han utilizado Scrum para el desarrollo de productos, tanto en empresas pequeñas, “startups” con tan sólo 3 personas desarrollando un producto, como en multinacionales, entre las que se encuentran las siguientes: Motorola, Nokia, IBM, Intel, Microsoft, Amazon, Google, Yahoo, SAP, Philips Medical, Boeing, Ubisoft, Electronic Arts, 3M, Ferrari entre muchas otras más (Amaya Balaguera, 2013) (Albaladejo, 2017).

En la actualidad, Scrum se está utilizando en diferentes tipos de negocio y, especialmente, en el desarrollo de software. La Scrum Alliance es la organización sin ánimo de lucro que se encarga de difundir Scrum en este ámbito (Albaladejo, 2017)

3.3.1. VISIÓN GENERAL

Scrum no es un proceso o una técnica para construir productos; en lugar de eso, es un marco de trabajo dentro del cual se pueden emplear varios procesos y técnicas. Scrum muestra la eficacia relativa de las prácticas de gestión de producto y las prácticas de desarrollo de modo de poder mejorar. A nivel general está formado por el Equipo Scrum, eventos y artefactos.

El marco de trabajo Scrum es de tipo iterativo e incremental y se compone de iteraciones de duración fija (de una a cuatro semanas) llamadas *Sprints*. El Equipo Scrum está conformado por el Dueño de Producto, el Equipo de Desarrollo y el Scrum Master. El Dueño de Producto es responsable de lograr el valor del negocio y se encarga de la visión del producto representada por una lista priorizada de requisitos, tareas y errores evolutivos organizados en el *Product Backlog*. Los requisitos generalmente son descritos por historias

de usuarios. El Equipo de Desarrollo es el encargado de desarrollar el software de manera incremental a través de los Sprints. El Scrum Master es un líder que guía al equipo en el proceso y ayuda a resolver problemas. En cada Sprint se construye y entrega un Incremento del producto. Este debe cumplir con una Definición de Terminado (*Definition of Done*). Al comienzo de cada Sprint, durante la planificación del mismo (*Sprint Planning*), el equipo se compromete a completar un cierto número de tareas establecidas a partir del Product Backlog y documentos en un *Sprint Backlog* (conjunto de elementos del Product Backlog para implementarse). Las tareas son realizadas por los miembros del equipo y se estiman. Al final de cada Sprint hay una Revisión del Sprint (*Sprint Review*) y una Retrospectiva del Sprint (*Sprint Retrospective*). Durante la Retrospectiva del Sprint se evalúa la velocidad y la moral del equipo y la calidad del trabajo. Todos los días el equipo se reúne brevemente (*Scrum Daily*) y responden a tres preguntas: lo que se ha hecho desde la última reunión, lo que se espera hacer hasta la próxima reunión y cuáles son los impedimentos para alcanzar la meta. La cantidad de tiempo requerida para completar cada tarea se actualiza en el Sprint Backlog (Scharff & Verma, 2010).

Scrum se basa en la teoría de control de procesos empírica o empirismo. Está compuesto por tres pilares que soportan toda la implementación del control de procesos empírico: transparencia, inspección y adaptación. La transparencia asegura que todos los participantes comparten un lenguaje común en referencia al proceso y que aquellos que desempeñan el trabajo y aquellos que aceptan el producto de dicho trabajo deben compartir una definición común de “Terminado”. La inspección tiene como objetivo detectar variaciones indeseadas. Por último, la adaptación se refiere a que, si uno o más aspectos de un proceso se desvían de los límites aceptables y que el producto resultante será inaceptable, el proceso debe ajustarse cuanto antes (Schwaber & Sutherland, 2016).

En la figura 3.1 se ilustra la visión general del funcionamiento de Scrum.

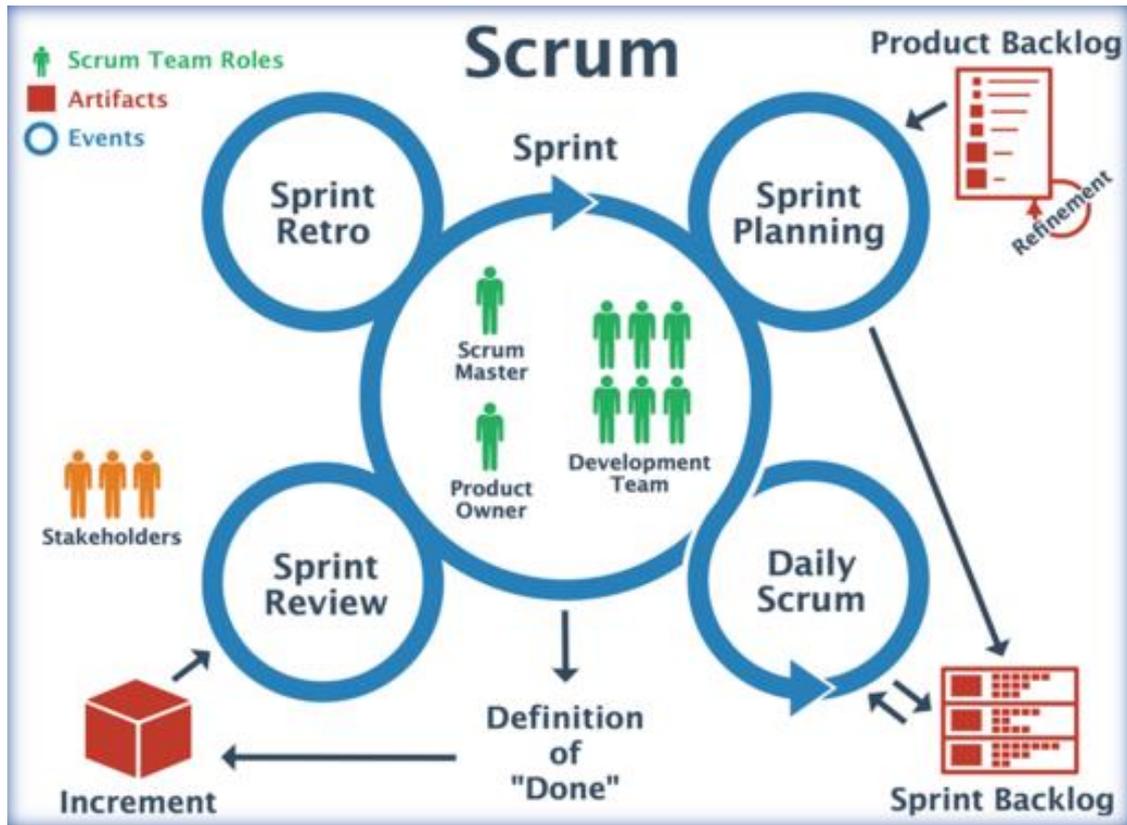


Figura 3.1: Visión general del funcionamiento de Scrum (Garg, 2017)

En los siguientes apartados se detalla cada uno de los roles, eventos y artefactos de la metodología.

3.3.2. EL EQUIPO SCRUM (*SCRUM TEAM*)

El Equipo Scrum consiste en un Dueño de Producto (Product Owner), el Equipo de Desarrollo (Development Team) y un Scrum Master. 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 de producto “Terminado” aseguran que siempre estará disponible una versión potencialmente útil y funcional del producto.

3.3.2.1. El Dueño del Producto (Product Owner)

El Dueño de Producto es el responsable de maximizar el valor del producto y el trabajo del Equipo de Desarrollo. El cómo se lleva a cabo esto podría variar ampliamente entre distintas organizaciones, Equipos Scrum e individuos.

El Dueño de Producto es la única persona responsable de gestionar la Lista del Producto (Product Backlog). La gestión de la Lista del Producto incluye:

- Expresar claramente los elementos de la Lista del Producto.
- Ordenar los elementos en la Lista del Producto para alcanzar los objetivos y misiones de la mejor manera posible.
- Optimizar el valor del trabajo que el Equipo de Desarrollo realiza.
- Asegurar que la Lista 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 Lista del Producto al nivel necesario.

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

Cabe destacar que en Scrum el Dueño de Producto es una única persona, no un comité. El Dueño de Producto podría representar los deseos de un comité en la Lista del Producto, pero aquellos que quieran cambiar la prioridad de un elemento de la Lista deben hacerlo a través del Dueño de Producto.

Para que el Dueño de Producto pueda hacer bien su trabajo, toda la organización debe respetar sus decisiones. Las decisiones del Dueño de Producto se reflejan en el contenido y en la priorización de la Lista del Producto.

3.3.2.2. El Equipo de Desarrollo (*Development Team*)

El Equipo de Desarrollo consiste en un grupo de profesionales que realizan el trabajo de entregar un Incremento de producto “Terminado” que potencialmente se pueda poner en producción al final de cada Sprint. Solo los miembros del Equipo de Desarrollo participan en la creación del Incremento.

Los Equipos de Desarrollo tienen las siguientes características:

- Son auto-organizados. Nadie (ni siquiera el Scrum Master) indica al Equipo de Desarrollo cómo convertir elementos de la Lista del Producto en Incrementos de funcionalidad potencialmente desplegados.
- Los Equipos de Desarrollo son multifuncionales, esto es, como equipo cuentan 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 tenerse 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. Tener menos de tres miembros en el Equipo de Desarrollo reduce la interacción y resulta en ganancias de productividad más pequeñas. 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 de Dueño de Producto y Scrum Master no cuentan en el cálculo del tamaño del equipo a menos que también estén contribuyendo a trabajar en la Lista de Pendientes de Sprint (Sprint Backlog).

3.3.2.3. El Scrum Master

El Scrum Master es el responsable de asegurar que Scrum se entienda y se adopte. 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 el Equipo Scrum pueden ser útiles 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 servicios al Dueño del Producto, al Equipo de Desarrollo y a la Organización.

En primer lugar, brinda servicios al Dueño del Producto de varias formas:

- Encontrar técnicas para gestionar la Lista de Producto de manera efectiva.
- Ayudar al Equipo Scrum a entender la necesidad de contar con elementos de Lista de Producto claros y concisos.
- Entender la planificación del producto en un entorno empírico.
- Asegurar que el Dueño de Producto conozca cómo ordenar la Lista de Producto para maximizar el valor.
- Entender y practicar la agilidad.
- Facilitar los eventos de Scrum según se requiera o necesite.

En segundo lugar, brinda servicios al Equipo de Desarrollo de varias formas:

- Guiar al Equipo de Desarrollo en ser auto-organizado y multifuncional.
- Ayudar al Equipo de Desarrollo a crear productos de alto valor.
- Eliminar impedimentos para el progreso del Equipo de Desarrollo.
- Facilitar los eventos de Scrum según se requiera o necesite.
- Guiar al Equipo de Desarrollo en entornos organizacionales en los que Scrum aún no haya sido adoptado y entendido por completo.

Por último, brinda servicios a la Organización de diferentes maneras:

- Liderar y guiar a la organización en la adopción de Scrum.
- Planificar las implementaciones de Scrum en la organización.
- Ayudar a los empleados e interesados a entender y llevar a cabo Scrum y el desarrollo empírico de producto.
- Motivar cambios que incrementen la productividad del Equipo Scrum.
- Trabajar con otros Scrum Masters para incrementar la efectividad de la aplicación de Scrum en la organización.

3.3.3. 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 se diseñaron específicamente para habilitar los pilares vitales de transparencia e inspección. La falta de alguno de estos eventos da como resultado una reducción de la transparencia y constituye una oportunidad perdida de inspección y adaptación.

3.3.3.1. 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 anterior.

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

Cabe destacar, que 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 clarificarse y renegociarse entre el Dueño de 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 lo que se construirá, un diseño y un plan flexible que guiará su construcción, el trabajo del equipo 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 incrementarse 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 aspecto importante a tener en cuenta es que un Sprint puede **cancelarse** antes de que el bloque de tiempo llegue a su fin. Solo el Dueño de 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. Sin embargo, debido a la corta duración de los Sprints, su cancelación rara vez tiene sentido.

Cuando se cancela un Sprint se revisan todos los Elementos de la Lista de Producto que se hayan completado y “Terminado”. Si una parte del trabajo es potencialmente entregable, el Dueño de Producto normalmente la acepta. Todos los Elementos de la Lista de Producto no completados se vuelven a estimar y se vuelven a introducir en la Lista de Producto. El trabajo finalizado en ellos pierde valor con rapidez y por lo general debe volverse a estimar.

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

3.3.3.2. Planificación del Sprint (*Sprint Planning*)

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

La 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 Planificación de Sprint responde a las siguientes preguntas:

- *¿Qué puede entregarse en el Incremento resultante del Sprint?*

El Equipo de Desarrollo trabaja para proyectar la funcionalidad que se desarrollará durante el Sprint. El Dueño de Producto discute el objetivo que el Sprint debería lograr y los Elementos de la Lista de 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 Lista de Producto, el último Incremento de 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 Lista de 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 planea los elementos de la Lista de Producto que 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 Lista de Producto y proporciona 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 Lista de Producto para el Sprint, el Equipo de Desarrollo decide cómo construirá esta funcionalidad para formar un Incremento de producto “Terminado” durante el Sprint. Los elementos de la Lista de Producto seleccionados para este Sprint, más el plan para terminarlos, recibe el nombre de Lista de Pendientes del Sprint (*Sprint Backlog*).

El Equipo de Desarrollo por lo general comienza diseñando el sistema y el trabajo necesario para convertir la Lista de Producto en un Incremento de producto funcional. El trabajo podría ser de tamaño o esfuerzo estimado variables. Sin embargo, durante la 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 Lista de Pendientes de Sprint, tanto durante la Planificación del Sprint como a lo largo del Sprint.

El Dueño de Producto puede ayudar a clarificar los elementos de la Lista de 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 Lista de Producto seleccionados con el Dueño de 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 Planificación del Sprint, el Equipo de Desarrollo debería ser capaz de explicar al Dueño de Producto y al Scrum Master cómo pretende trabajar como un equipo auto-organizado para lograr el Objetivo del Sprint y crear el Incremento esperado.

3.3.3.3. Objetivo del Sprint (*Sprint Goal*)

El Objetivo del Sprint es una meta establecida para el Sprint que puede lograrse mediante la implementación de la Lista de Producto. Proporciona una guía al Equipo de Desarrollo acerca de por qué está construyendo el incremento. Se crea durante la Planificación del Sprint. El objetivo del Sprint brinda al equipo de desarrollo cierta flexibilidad con respecto a la funcionalidad implementada en el Sprint. Los elementos de la Lista del Producto seleccionados ofrecen una función coherente que puede ser el objetivo del Sprint. El objetivo del Sprint puede representar otro nexos que haga que el Equipo de Desarrollo trabaje en conjunto y no en iniciativas separadas.

A medida que el equipo de desarrollo trabaja 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 Lista de pendientes del Sprint (*Sprint Backlog*).

3.3.3.4. Scrum Diario (*Daily Scrum*)

El Scrum Diario es una reunión con un bloque de tiempo de 15 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.

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?
- ¿Ve 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 Lista de Pendientes 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 tenga la reunión, pero es el Equipo de Desarrollo el responsable de dirigir el Scrum Diario. El Scrum Master enseña al Equipo de Desarrollo a mantener 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 realizar otras reuniones, identifican impedimentos a remover relativos al desarrollo, resaltan y promueven la toma de decisiones rápida y mejoran el nivel de conocimiento del Equipo de Desarrollo. El Scrum Diario es una reunión clave de inspección y adaptación.

3.3.3.5. Revisión de Sprint (*Sprint Review*)

Al final del Sprint se lleva a cabo una Revisión de Sprint para inspeccionar el Incremento y adaptar la Lista de 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 Lista de 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 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 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 de Producto.
- El Dueño de Producto explica qué elementos de la Lista de Producto se han “Terminado” y cuales no se han “Terminado”.
- El Equipo de Desarrollo habla acerca de qué estuvo bien durante el Sprint, qué problemas aparecieron y cómo fueron resueltos esos problemas.
- El Equipo de Desarrollo hace una demostración del trabajo que ha “Terminado” y responde preguntas acerca del Incremento.
- El Dueño de Producto habla acerca de la Lista de Producto en su 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 del 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 Lista de Producto revisada que define los elementos de la Lista de Producto posibles para el siguiente Sprint. Es posible además que la Lista de Producto reciba un ajuste general para enfocarse en nuevas oportunidades.

3.3.3.6. Retrospectiva de Sprint (*Sprint Retrospective*)

La Retrospectiva de Sprint es una oportunidad para el Equipo Scrum de inspeccionarse a sí mismo y de 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 Planificación de 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 usualmente más corto. El Scrum Master se asegura de que el evento se lleve a cabo y que los asistentes entiendan su propósito, 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.
- 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 y amenos para 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 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.

3.3.4. 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, necesaria para asegurar que todos tengan el mismo entendimiento del artefacto.

3.3.4.1. Lista de Producto (*Product Backlog*)

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

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

La Lista de Producto enumera todas las características, funcionalidades, requisitos, mejoras y correcciones que constituyen cambios a realizarse sobre el producto para entregas futuras. Los elementos de la Lista de Producto tienen como atributos la descripción, el orden, la estimación y el valor.

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

El refinamiento (*refinement*) de la Lista de Producto es el acto de añadir detalle, estimaciones y orden a los elementos de la misma. 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 la Lista. Durante el refinamiento se examinan y revisan sus elementos. 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 Lista de Producto pueden actualizarse en cualquier momento por el Dueño de Producto o a criterio suyo.

Los elementos de la Lista de 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 Lista de 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 pueda ser “Terminado” dentro de los límites del bloque de tiempo del Sprint. Los elementos de la Lista de 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 Lista de 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 ayudándoles a entender y seleccionar sus compromisos, 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 de 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 de tendencias se han utilizado para predecir el progreso, como trabajo pendiente (*burndown*), completado (*burnup*) y flujo acumulado (*cumulative flow*). Estas han probado ser útiles. Sin embargo, no reemplazan 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.

3.3.4.2. Lista de Pendientes del Sprint (*Sprint Backlog*)

La Lista de Pendientes del Sprint es el conjunto de elementos de la Lista de Producto seleccionados para el Sprint, más un plan para entregar el Incremento de producto y conseguir el Objetivo del Sprint. La Lista de Pendientes 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 Lista de Pendientes del Sprint hace visible todo el trabajo que el Equipo de Desarrollo identifica como necesario para alcanzar el Objetivo del Sprint. 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 Lista de Pendientes del Sprint durante el Sprint y esta emerge a lo largo del Sprint. Esto ocurre a medida que el Equipo de Desarrollo trabaja en lo planeado y aprende más acerca del trabajo necesario para conseguir el Objetivo del Sprint.

Cuando se requiere nuevo trabajo, el Equipo de Desarrollo lo adiciona a la Lista de Pendientes 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 se considera innecesario, es eliminado. Solo el Equipo de Desarrollo puede cambiar su Lista de Pendientes del Sprint durante un Sprint, es decir, 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 Lista de Pendientes 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.

3.3.4.3. Incremento

El Incremento es la suma de todos los elementos de la Lista de 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.

3.3.5. 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 de 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.

La labor del Scrum Master es 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.

3.3.6. DEFINICIÓN DE “TERMINADO” (*DEFINITION OF “DONE”*)

Cuando un elemento de la Lista de Producto o 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 de producto.

Esta misma definición guía al Equipo de Desarrollo en saber cuántos elementos de la Lista de Producto puede seleccionar durante la 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 de 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 especificar 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 de manera exhaustiva, 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.

3.3. JUSTIFICACIÓN DEL USO DE SCRUM

El mercado de aplicaciones móviles ha experimentado una rápida expansión durante los últimos 15 años, las plataformas móviles siguen mejorando su desempeño, y la necesidad de los usuarios de una amplia variedad de aplicaciones móviles va en aumento. El desarrollo

de software para las plataformas móviles viene con características únicas y restricciones que se aplican a la mayoría de las etapas del ciclo de vida. Las características distintivas más importantes son: un alto nivel de competitividad, corto tiempo de entrega, movilidad, portabilidad, capacidades específicas y constantemente cambiantes de las terminales, sistemas operativos diferentes e incompatibles, entre otras (Amaya Balaguera, 2013)

Las metodologías ágiles como son una excelente alternativa para guiar proyectos de desarrollo de software de tamaño reducido, como es el caso de las aplicaciones para dispositivos móviles, gracias a la gran facilidad de adaptación que poseen.

Para el caso particular de *Scrum*, los resultados obtenidos muestran que es una metodología que puede favorecer el desarrollo rápido de aplicaciones completamente funcionales. Su dinámica de uso de tipo iterativa e incremental permite realizar entregas de módulos funcionales y a la vez obtener una retroalimentación para futuras correcciones y mejoras.

El marco de trabajo a través de los Sprints permite obtener un incremento de producto terminado, utilizable y desplegable en aproximadamente un mes calendario o menos. Por otro lado, gracias a los eventos de Scrum, es posible realizar una inspección y adaptación del proceso a lo largo de todo el proyecto, con el objetivo de obtener el producto final que cumpla con todos los requerimientos.

Finalmente, es necesario destacar que la metodología Scrum fue implementada con éxito en el Trabajo Final de Graduación propuesto por (Loto , 2015), el cual desarrolla una aplicación móvil personalizada de apoyo al aprendizaje de redes de computadoras. El trabajo mencionado anteriormente se desarrolló en la misma institución universitaria en la cual se desarrolla el presente trabajo.

CAPÍTULO 4

DESARROLLO DE LA APLICACIÓN

En este capítulo se presenta el diseño y desarrollo de la aplicación móvil cliente-servidor, detallando todo el proceso a través de la metodología ágil *Scrum*.

En la primera sección se describe el comienzo del proceso Scrum definiendo la arquitectura del sistema y el *Equipo Scrum*. Luego, en la segunda sección, se presenta la *Lista de Producto* que almacena todos los requerimientos de este proyecto. En la tercera sección, se presenta la *Definición de Terminado*, documento que establece los compromisos asumidos para el proceso de desarrollo. Las siguientes secciones comprenden el desarrollo de cada *Sprint* surgidos a partir de la Lista de Producto. Por último, en la última sección, se detallan las *herramientas de software* utilizadas a lo largo del proyecto.

4.1. DESARROLLO DEL PROCESO SCRUM

Para comenzar con la metodología es necesario definir el *Equipo Scrum* y sus roles de la siguiente manera:

- ❖ Product Owner: Dra. Elena B. Durán (Directora del Proyecto de investigación en el que se enmarca el trabajo)
- ❖ Scrum Máster: Ing. Silvina I. Unzaga (Investigadora integrante del proyecto de investigación)
- ❖ Development Team: Daniel Sebastian Montoto

Además, como se describió en el apartado 3.2, Scrum es una metodología de tipo iterativa e incremental y su corazón es el *Sprint*. En el presente trabajo se considera un sprint adicional denominado *Sprint0*, el cual tiene como objetivo principalmente definir la arquitectura global del sistema y el Product Backlog. También se describen las herramientas de desarrollo utilizadas.

La arquitectura que se muestra en la figura 4.1 juega un papel imprescindible ya que se encarga de mostrar los elementos más importantes del sistema y su interacción entre ellos. Está compuesta por dos grandes componentes: Cliente y Servidor. El cliente es la aplicación desarrollada en Android que se ejecutará en los teléfonos móviles. El servidor está compuesto por una aplicación web desarrollada en PHP, MySQL y otros frameworks para administrar información relacionada a los usuarios, objetos de aprendizaje, puntos de interés, etc. Internet es el medio de comunicación entre el Cliente y el Servidor.

Cabe destacar que en el servidor además de la aplicación web, se encuentra almacenada la base de datos, webservices y los objetos de aprendizaje previamente generados. La base de datos almacena toda la información relacionada a los estudiantes y su historial de aprendizaje (Modelo de estudiante), docentes administradores, como así también todo lo referido a los puntos de interés y objetos de aprendizaje. Los webservices son scripts desarrollados en PHP y MySQL con el objetivo de intercambiar datos entre la aplicación móvil y el servidor.



Figura 4.1: Arquitectura global del Sistema

4.2. LISTA DE PRODUCTO (PRODUCT BACKLOG)

En esta lista se almacenan todas las funcionalidades y requerimientos de forma priorizada. Es una lista dinámica que fue evolucionando y actualizándose a lo largo del proyecto. El resultado final se muestra en la tabla 4.1.

Los requerimientos / historias de usuario se describen utilizando el siguiente formato: Como un [Rol], requiero [descripción de la funcionalidad], con la finalidad de [razón o resultado] adaptado de (Armijos, Fiallos, Villavicencio, & Abad, 2015).

Tabla 4.1: Lista de Producto final

ID	Requerimientos / Historias de usuario	Imp.	Estim.
1	Como <i>Administrador</i> requiero que la aplicación móvil permita iniciar sesión para identificar al usuario en el sistema.	99	5
2	Como <i>Administrador</i> requiero que la aplicación móvil permita cerrar sesión para que el usuario pueda salir del sistema.	97	4
3	Como <i>Administrador</i> requiero que la aplicación móvil geolocalice al alumno para luego recomendarle puntos de interés (PI) cercanos y facilitar un aprendizaje situado.	95	8
4	Como <i>Administrador</i> requiero que se almacenen las coordenadas geográficas de cada PI para calcular las distancias entre cada PI y el alumno.	94	2
5	Como <i>Administrador</i> requiero que la aplicación móvil ordene los PI cercanos de menor a mayor distancia para que el alumno pueda seleccionar el más cercano.	92	7
6	Como <i>Administrador</i> requiero que la aplicación móvil recomiende objetos de aprendizaje (OA) en base al tiempo disponible y conocimiento previo que el alumno posee para facilitar un aprendizaje personalizado.	90	5
7	Como <i>Administrador</i> requiero que los OA mantengan un estado (no realizado, aprobado y desaprobado) por cada alumno, para futuras personalizaciones asociadas al conocimiento previo.	88	3
8	Como <i>Administrador</i> requiero que al dar de alta un alumno se establezca por defecto No-Realizado el estado de todos los objetos de aprendizaje asociado a ese alumno.	87	2
9	Como <i>Administrador</i> requiero que al dar de alta un objeto de aprendizaje se establezca por defecto No-Realizado el estado de ese objeto para todos los alumnos.	85	2
10	Como <i>Administrador</i> requiero un cambio de estado del OA (asociado a un alumno) a Aprobado cuando el alumno apruebe la evaluación incluida en el OA.	83	2

11	Como <i>Administrador</i> requiero un cambio de estado del OA (asociado a un alumno) a Desaprobado cuando el alumno desaproebe la evaluación incluida en el OA.	81	2
12	Como <i>Administrador</i> requiero que la aplicación móvil permita visualizar OA previamente generados para que el estudiante pueda interactuar con ellos.	79	4
13	Como <i>Administrador</i> requiero que la aplicación móvil permita ingresar al alumno su tiempo disponible para recomendarle OA de forma personalizada.	78	2
14	Como <i>Administrador</i> requiero una aplicación web para administrar los parámetros generales del sistema.	65	10
15	Como <i>Administrador</i> requiero que la aplicación web permita iniciar sesión a docentes administradores para ingresar al sistema.	64	2
16	Como <i>Administrador</i> requiero que la aplicación web cuente con dos tipos de administradores para que el de mayor jerarquía pueda dar de alta nuevos administradores.	62	2
17	Como <i>Administrador</i> requiero que la aplicación web permita cerrar sesión para salir del sistema.	60	1
18	Como <i>Administrador</i> requiero que la aplicación web permita identificar el ingreso de un administrador para mostrarle el panel de control correspondiente.	57	3
19	Como <i>Administrador</i> requiero que la aplicación web permita dar de alta un alumno para que el mismo pueda iniciar sesión desde la aplicación móvil.	56	2
20	Como <i>Administrador</i> requiero que la aplicación web permita dar de baja un alumno y toda su información relacionada para eliminarlo del sistema	55	2
21	Como <i>Administrador</i> requiero que la aplicación web permita modificar información relacionada al alumno para actualizar el modelo del estudiante.	54	2
22	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel alumnos para visualizar información relacionada a ellos.	53	1
23	Como <i>Administrador</i> requiero que la aplicación web permita dar de alta la referencia a un OA, vinculado a un PI, para que esté disponible desde la aplicación móvil.	52	1
24	Como <i>Administrador</i> requiero que la aplicación web permita dar de baja la referencia a un OA para que no esté disponible desde la aplicación móvil.	51	1
25	Como <i>Administrador</i> requiero que la aplicación web permita modificar información relacionada a un OA para mantener actualizados los metadatos propios.	50	1

26	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel objetos de aprendizaje para visualizar información relacionada a ellos.	49	1
27	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel historia académica para visualizar información relacionada con alumnos y objetos de aprendizaje.	40	2
28	Como <i>Administrador</i> requiero que la aplicación web permita dar de alta un PI para que esté disponible desde la aplicación móvil.	37	1
29	Como <i>Administrador</i> requiero que la aplicación web permita dar de baja un PI para que no esté disponible desde la aplicación móvil.	35	1
30	Como <i>Administrador</i> requiero que la aplicación web permita modificar un PI para actualizar la información relacionada a él.	33	1
31	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel punto de interés para visualizar información relacionada a ellos.	31	1
32	Como <i>Administrador</i> requiero que una vez integrado el sistema se realicen pruebas de validación para corroborar que satisface todos los requerimientos.	27	10
33	Como <i>Administrador</i> requiero que la aplicación móvil cuente con un manual de usuario.	10	3

Referencia de la tabla 4.1:

- La primera columna hace referencia a un identificador único para cada requerimiento.
- La segunda columna es la descripción del requerimiento siguiendo el formato de (Armijos, Fiallos, Villavicencio, & Abad, 2015).
- La tercera columna refleja la importancia y/o prioridad de cada requerimiento.
- La cuarta columna hace referencia a la estimación de tiempo de cada requerimiento, definidos en puntos de historia (Kniberg, 2007) (Palacio, 2015).

4.3. DEFINICIÓN DE TERMINADO

La Definición de Terminado (*Definition of Done*) es un estándar de calidad que garantiza que se cumpla uno de los pilares fundamentales de Scrum, la *transparencia*. El objetivo de este documento es asegurar que todo el Equipo Scrum entienda cuando un *Incremento* de producto está terminado, es decir, cuando potencialmente se puede poner en producción. Para este proyecto de desarrollo de software se utilizó la siguiente Definición de Terminado:

- ✓ Todos los requerimientos de software deben estar codificados y compilados sin errores.
- ✓ Cada Incremento debe integrarse a los Incrementos anteriores sin provocar fallos y estar preparado para el despliegue.
- ✓ Cada Incremento debe contar con pruebas funcionales con el objetivo de depurar errores.
- ✓ Para cada Incremento realizar una demostración en la reunión de Revisión de Sprint (*Sprint Review*).
- ✓ Cada Incremento debe pasar la prueba de aceptación por parte del Dueño de Producto.
- ✓ Al final del proyecto debe realizarse una prueba de integración para garantizar que todos los Incrementos funcionan en conjunto.
- ✓ El código desarrollado debe estar documentado y almacenarse un repositorio de control de versiones.
- ✓ Se deben cumplir requisitos no funcionales:
 - ✓ La aplicación móvil debe ejecutarse correctamente en teléfonos móviles con sistema operativo Android desde la API 21 (Lollipop) hasta la API 24 (Nougat).
 - ✓ La aplicación web debe ejecutarse correctamente en navegadores de uso común tales como Google Chrome, Internet Explorer y Mozilla Firefox.
 - ✓ La aplicación móvil debe contar con un manual de usuario.
 - ✓ La aplicación móvil debe contar con una interfaz elegante y fácil de usar.

4.4. SPRINT 1: CONTROL DE ACCESO APLICACIÓN MÓVIL

4.4.1 REUNIÓN DE PLANIFICACIÓN DEL SPRINT 1

A continuación, se responden las preguntas asociadas a esta reunión.

- ¿Qué puede entregarse en el Incremento resultante del Sprint?

En este Sprint se diseña e implementa el sistema de control de acceso para la aplicación móvil. El control de acceso permite identificar a los estudiantes que inician sesión para brindarles los Objetos de Aprendizaje de forma personalizada en base al tiempo disponible y conocimiento previo. Se implementa por un lado la función para mantener la sesión iniciada y por otro lado la función para cerrar sesión. En esta reunión también se discute el objetivo del sprint, el cual es definido en la siguiente sección. La duración de este sprint es de 2 semanas.

- ¿Cómo se conseguirá completar el trabajo seleccionado?

La aplicación móvil es desarrollada íntegramente con el software Android Studio. En primer lugar, se realiza el diseño de la ventana para el inicio de sesión. Esta ventana permite ingresar el nombre de usuario y contraseña y luego enviarlos al servidor para ser validados. Se utilizan Webservices creados en PHP para la comunicación entre la aplicación móvil y el servidor.

En esta reunión se seleccionan los elementos de la Lista de Producto que pasarán a formar parte de la Lista de Pendientes del Sprint 1. Esta última se presenta en la sección 4.4.3.

4.4.2. OBJETIVO DEL SPRINT 1

El objetivo de este Sprint es desarrollar la primera versión de la aplicación móvil con la funcionalidad de control de acceso. Esto permitirá iniciar y cerrar sesión a los estudiantes.

4.4.3. LISTA DE PENDIENTES DEL SPRINT 1

En la tabla 4.2 se presenta la Lista de Pendientes del Sprint 1 donde se detallan las tareas a realizar.

Tabla 4.2: Lista de Pendientes Sprint 1

ID	Requerimientos / Historias de usuario	Tareas	Estim.
1	Como <i>Administrador</i> requiero que la aplicación móvil permita iniciar sesión para identificar al usuario en el sistema.	<ul style="list-style-type: none"> - Diseñar la interfaz gráfica de la pantalla inicial. - Implementar la librería Volley para la conexión y envío de datos al servidor. - Realizar pruebas y depuración. 	5
2	Como <i>Administrador</i> requiero que la aplicación móvil permita cerrar sesión para que el usuario pueda salir del sistema.	<ul style="list-style-type: none"> - Diseñar e implementar la función para cerrar sesión en la aplicación móvil. - Realizar pruebas y depuración verificando el correcto cierre de sesión. 	4

4.4.4. INCREMENTO DEL SPRINT 1

El incremento del producto de este Sprint es la versión inicial de la aplicación móvil. La misma cuenta con una pantalla de inicio que permite ingresar el nombre de usuario y contraseña del estudiante. Estos datos son enviados al servidor para ser validados y en el caso de ser correctos, permite el inicio de sesión. Por otro lado, la aplicación cuenta con un menú que permite el cierre de sesión.

Resultado del Incremento

Este módulo funcional permite que solo los estudiantes, dados de alta por docentes, puedan iniciar sesión y acceder al contenido del Taller de Ambientación. Asimismo, este módulo cuenta con la función para cerrar sesión. Las figuras 4.2 y 4.3 muestran las capturas de pantalla con el resultado obtenido al finalizar el Sprint 1.



Figura 4.2: Inicio de sesión

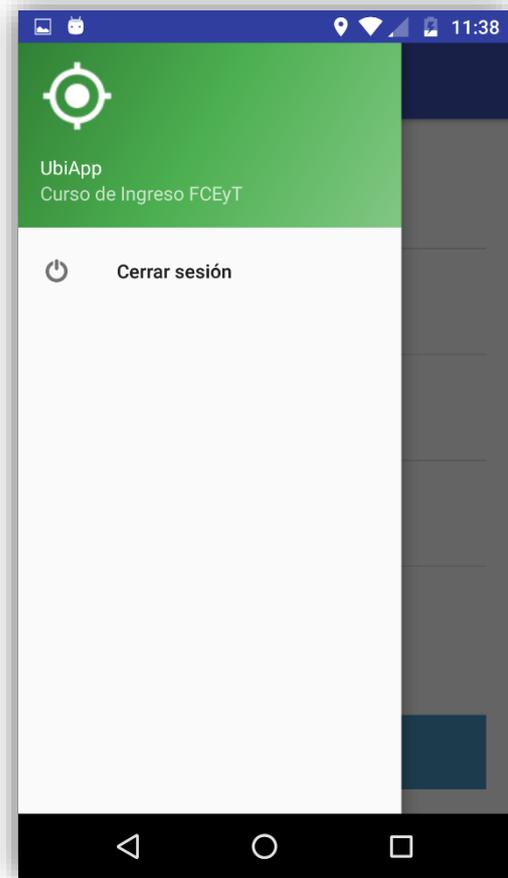


Figura 4.3: Cierre de sesión

4.4.5. REVISIÓN Y RETROSPECTIVA DEL SPRINT 1

En la reunión se presentó el incremento que consiste en la versión inicial de la aplicación móvil. Se realizó una demostración sobre el trabajo terminado que consistió en ingresar datos de prueba de usuario y comprobar que son autenticados correctamente por el servidor.

La Lista de Producto fue revisada para definir qué elementos se terminaron en este Sprint y cuáles son los posibles candidatos para el siguiente Sprint. Los ajustes en la Lista de Producto fueron mínimos.

Con respecto a los problemas surgidos en este Sprint, se debieron principalmente al uso de librerías para conexiones al servidor que actualmente ya no están recomendadas. Se solucionó satisfactoriamente incorporando modernas y más eficientes librerías en los módulos de conexión. Por último, en cuanto a personas, relaciones, procesos y herramientas no hubo cambios significativos.

4.5. SPRINT 2: GEOLOCALIZACIÓN Y RECOMENDACIÓN DE PUNTOS DE INTERÉS

4.5.1. REUNIÓN DE PLANIFICACIÓN DEL SPRINT 2

A continuación, se responden las preguntas asociadas a esta reunión.

- ¿Qué puede ser terminado en este Sprint?

En este Sprint se diseña e implementa el módulo de geolocalización y recomendación de puntos de interés. Este módulo permite obtener la ubicación a cielo abierto del estudiante que inicia sesión en la aplicación móvil y posteriormente, recomendarle los puntos de interés de la universidad ordenados por cercanía. En esta reunión también se discute el objetivo del sprint, el cual es definido en la siguiente sección. La duración de este sprint es de 4 semanas.

- ¿Cómo se conseguirá completar el trabajo seleccionado?

En primer lugar, se realiza una revisión y análisis de las librerías que provee Android para proporcionar geolocalización. Luego, se integra la librería adecuada y se realiza la codificación del módulo que permite obtener la posición geográfica del dispositivo móvil. Finalmente, se diseña la ventana que muestra los puntos de interés en forma de lista ordenada creciente, indicando una distancia aproximada en metros, hasta cada punto de interés.

En esta reunión se seleccionan los elementos de la Lista de Producto que pasarán a formar parte de la Lista de Pendientes del Sprint 2. Esta última se presenta en la sección 4.5.3.

4.5.2. OBJETIVO DEL SPRINT 2

El objetivo de este Sprint es desarrollar e implementar el módulo que permitirá geolocalizar el dispositivo móvil del estudiante y luego recomendarle puntos de interés cercanos.

4.5.3. LISTA DE PENDIENTES DEL SPRINT 2

En la tabla 4.3 se presenta la Lista de Pendientes del Sprint 2 donde se detallan las tareas a realizar.

Tabla 4.3: Lista de Pendientes Sprint 2

ID	Requerimientos / Historias de usuario	Tareas	Estim.
1	Como <i>Administrador</i> requiero que la aplicación móvil geocalice al alumno para luego recomendarle puntos de interés (PI) cercanos y facilitar un aprendizaje situado.	<ul style="list-style-type: none"> - Analizar librerías de geocalización y seleccionar la más adecuada. - Integrar las librerías al código. - Desarrollar el módulo de recomendación de PI cercanos. - Realizar pruebas y depuración. 	8
2	Como <i>Administrador</i> requiero que se almacenen las coordenadas geográficas de cada PI para calcular las distancias entre cada PI y el alumno.	<ul style="list-style-type: none"> - Recolectar las coordenadas geográficas (latitud y longitud) de cada PI. - Almacenar las coordenadas en formato decimal en la base de datos. 	2
3	Como <i>Administrador</i> requiero que la aplicación móvil ordene los PI cercanos de menor a mayor distancia para que el alumno pueda seleccionar el más cercano.	<ul style="list-style-type: none"> - Desarrollar el módulo de ordenación de los PI. - Realizar pruebas y depuración. 	7

4.5.4. INCREMENTO DEL SPRINT 2

El incremento del producto de este Sprint es la funcionalidad, integrada en la aplicación móvil, que permite obtener la ubicación geográfica del estudiante y luego recomendarle puntos de interés cercanos en el ámbito de la universidad.

Resultado del Incremento

Luego de iniciar sesión, se obtiene la ubicación geográfica del estudiante y la aplicación móvil es capaz de sugerirle los puntos de interés ordenados por cercanía.

En la figura 4.4, es posible notar una etiqueta “Exactitud”, la cual se encarga de mostrar constantemente la precisión de las señales obtenidas por el GPS. El botón “Actualizar ubicación” es requerido cuando el estudiante se desplaza de un lugar a otro con el fin de recalculer las distancias.

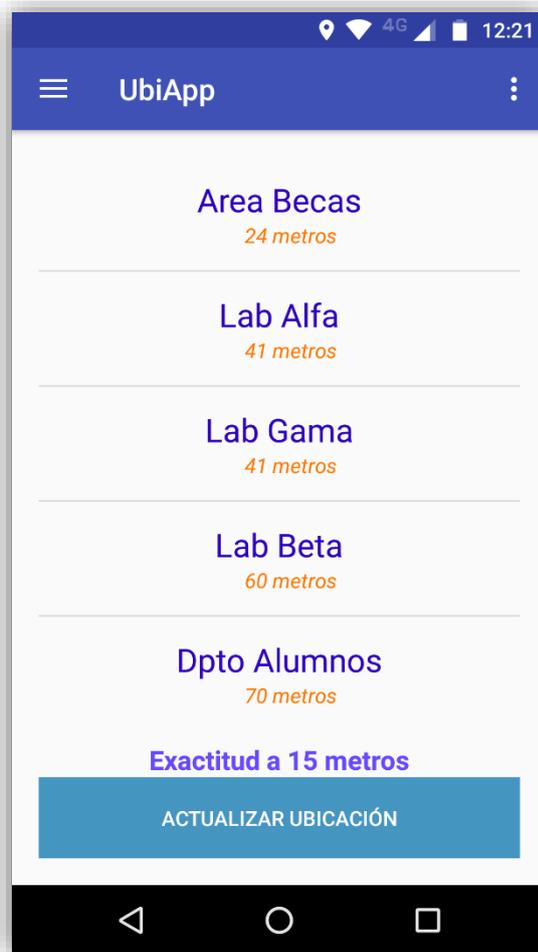


Figura 4.4: Recomendación de puntos de interés cercanos

4.5.5. REVISIÓN Y RETROSPECTIVA DEL SPRINT 2

En la reunión se presentó el incremento que consiste en la funcionalidad de geolocalización y recomendación de puntos de interés. Se realizó una demostración sobre el trabajo terminado que consistió básicamente, en recorrer diferentes lugares de la universidad

y comprobar que la aplicación recomienda correctamente los puntos de interés que se encuentran cerca del dispositivo móvil.

La Lista de Producto fue revisada para definir qué elementos se terminaron en este Sprint y cuáles son los posibles candidatos para el siguiente Sprint. Los ajustes en la Lista de Producto fueron mínimos.

Con respecto a los problemas surgidos en este Sprint, se debieron principalmente a la integración de la librería de geolocalización que combina métodos de GPS y redes de datos. Los mismos fueron solucionados satisfactoriamente. Otro inconveniente detectado fue que el GPS inserto en el dispositivo móvil demora un poco más de lo habitual en captar satélites cuando el día está nublado, hay obstáculos como árboles o se encuentra cerca de edificios altos. Por último, en cuanto a personas, relaciones, procesos y herramientas no hubo cambios significativos.

4.6. SPRINT 3: MÓDULO DE PERSONALIZACIÓN

4.6.1. REUNIÓN DE PLANIFICACIÓN DEL SPRINT 3

A continuación, se responden las preguntas asociadas a esta reunión.

- ¿Qué puede ser terminado en este Sprint?

En este Sprint se diseña e implementa el módulo más importante del sistema, el *módulo de personalización*. Este se encarga de recomendar los objetos de aprendizaje de forma personalizada en base a dos variables: los conocimientos previos del estudiante y el tiempo disponible que este tiene para realizar las tareas propuestas en el OA. Asimismo, el estudiante podrá visualizar e interactuar con los OA seleccionando el que desee. En esta reunión también se discute el objetivo del sprint, el cual es definido en la siguiente sección. La duración de este sprint es de 4 semanas.

- ¿Cómo se conseguirá completar el trabajo seleccionado?

En primer lugar, se diseña la interfaz que permite al estudiante ingresar su tiempo disponible. Luego se diseña la ventana donde se presentarán los OA recomendados en forma de lista. Por último, para visualizar e interactuar con los OA se implementa un contenedor nativo de Android (WebView). Este contenedor será capaz de visualizar el OA que se encuentra almacenado en el servidor en formato HTML.

A partir de la construcción de las reglas lógicas, se diseña y desarrolla el módulo que recomendará los objetos de aprendizaje en base a los conocimientos previos del estudiante y el tiempo que este dispone. El primer caso, se llevará a cabo mediante el almacenamiento de estados (No Realizado, Aprobado y Desaprobado) a medida que el estudiante interactúe con los Objetos. Para el segundo caso, se consultarán los metadatos del OA donde se consigna el tiempo requerido para la realización del OA el que luego, se contrastará con el tiempo disponible declarado por el estudiante.

En esta reunión se seleccionan los elementos de la Lista de Producto que pasarán a formar parte de la Lista de Pendientes del Sprint 3. Esta última se presenta en la sección 4.6.3.

4.6.2. OBJETIVO DEL SPRINT 3

El objetivo de este Sprint es desarrollar e implementar el módulo de personalización, que permitirá recomendar Objetos de Aprendizaje de forma personalizada teniendo en cuenta los conocimientos previos del estudiante y el tiempo disponible que este posee.

4.6.3. LISTA DE PENDIENTES DEL SPRINT 3

En la tabla 4.4 se presenta la Lista de Pendientes del Sprint 3 donde se detallan las tareas a realizar.

Tabla 4.4: Lista de Pendientes Sprint 3

ID	Requerimientos / Historias de usuario	Tareas	Estim.
1	Como <i>Administrador</i> requiero que la aplicación móvil recomiende objetos de aprendizaje (OA) en base al tiempo disponible y conocimiento previo que el alumno posee para facilitar un aprendizaje personalizado.	<ul style="list-style-type: none"> - Definir la regla que permita comparar el tiempo ingresado por el estudiante con el tiempo de aprendizaje de los OA, para filtrar los OA de tiempo menor o igual al declarado por el estudiante. - Definir la regla, asociada al conocimiento previo, que permita filtrar los OA <i>No-Realizados</i> en primer lugar, <i>Desaprobados</i> en segundo lugar y <i>Aprobados</i> en último lugar. - Implementar las reglas definidas anteriormente. 	5
2	Como <i>Administrador</i> requiero que los OA mantengan un estado (no realizado, aprobado y desaprobado) por cada alumno, para futuras personalizaciones asociadas al conocimiento previo.	<ul style="list-style-type: none"> - Implementar el requerimiento en el modelo del estudiante. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	3
3	Como <i>Administrador</i> requiero que al dar de alta un alumno se establezca por defecto No-Realizado el estado de todos los objetos de aprendizaje asociado a ese alumno.	<ul style="list-style-type: none"> - Implementar el requerimiento en el modelo del estudiante. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	2
4	Como <i>Administrador</i> requiero que al dar de alta un objeto de aprendizaje se establezca por defecto No-Realizado el	<ul style="list-style-type: none"> - Implementar el requerimiento en el modelo del estudiante. 	2

	estado de ese objeto para todos los alumnos.	<ul style="list-style-type: none"> - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	
5	Como <i>Administrador</i> requiero un cambio de estado del OA (asociado a un alumno) a Aprobado cuando el alumno apruebe la evaluación incluida en el OA.	<ul style="list-style-type: none"> - Implementar el requerimiento en el modelo del estudiante. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	2
6	Como <i>Administrador</i> requiero un cambio de estado del OA (asociado a un alumno) a Desaprobado cuando el alumno desapruebe la evaluación incluida en el OA.	<ul style="list-style-type: none"> - Implementar el requerimiento en el modelo del estudiante. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	2
7	Como <i>Administrador</i> requiero que la aplicación móvil permita visualizar OA previamente generados para que el estudiante pueda interactuar con ellos.	<ul style="list-style-type: none"> - Diseñar la interfaz gráfica donde se mostrará el OA. - Implementar un contenedor web (WebView) nativo en la ventana diseñada anteriormente. - Realizar pruebas con el servidor y depuración. 	4
8	Como <i>Administrador</i> requiero que la aplicación móvil permita ingresar al alumno su tiempo disponible para recomendarle OA de forma personalizada.	<ul style="list-style-type: none"> - Diseñar la interfaz gráfica que permite al estudiante ingresar su tiempo disponible. - Almacenar temporalmente el tiempo disponible hasta que el estudiante salga de la aplicación. - Realizar pruebas y depuración. 	2

4.6.4. INCREMENTO DEL SPRINT 3

El incremento del producto de este Sprint es la funcionalidad, integrada en la aplicación móvil, que permite recomendar Objetos de Aprendizaje de forma personalizada y posteriormente interactuar con ellos a través de un contenedor web nativo.

Resultados del Incremento

Es un módulo clave del sistema y se encarga de recomendar Objetos de Aprendizaje en base al tiempo disponible del estudiante y a su conocimiento previo. Además, la aplicación cuenta con un contenedor web nativo (*WebView*) para interactuar con los Objetos de Aprendizaje. Luego de iniciar sesión, el estudiante puede indicar su tiempo disponible a través de una *SeekBar* disponible en Android Studio (figura 4.5).

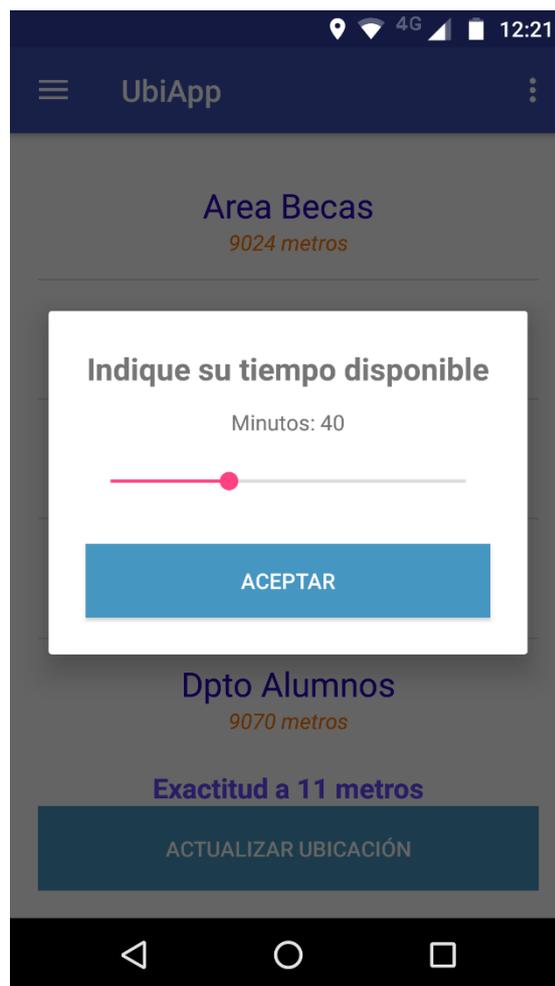


Figura 4.5: *SeekBar* para que el estudiante indique su tiempo disponible

Si el estudiante ingresa por primera vez a la aplicación, el estado de cada Objeto de Aprendizaje se encontrará en *NoRealizado* (figura 4.6). A modo de ejemplo, si un estudiante decide realizar el *ObjPrueba3* y aprueba la evaluación, el estado del *ObjPrueba3* se actualiza inmediatamente. Esta última interacción pasará a formar parte del conocimiento previo del estudiante (figura 4.7).

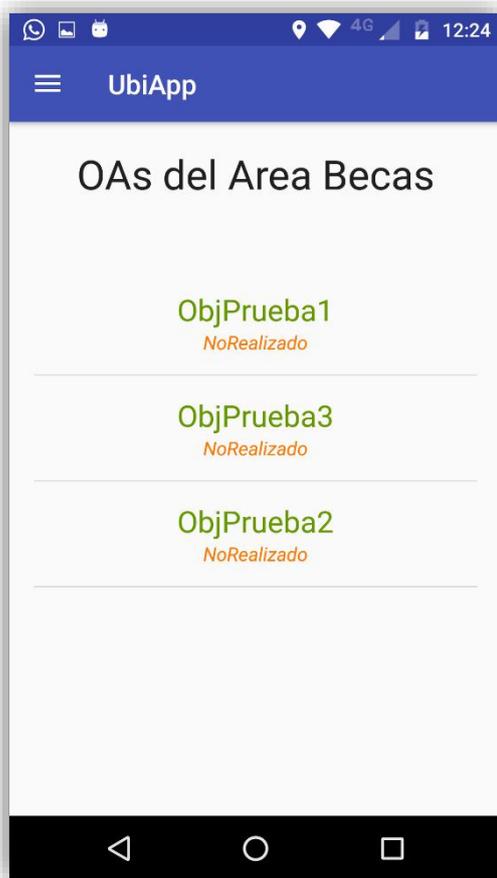


Figura 4.6: Listado de OAs

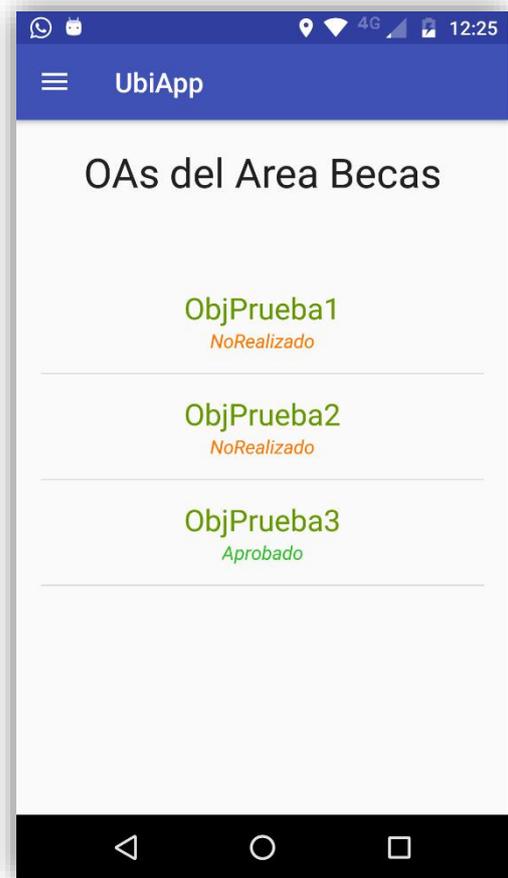


Figura 4.7: Listado de OAs actualizado

Un ejemplo de interacción con el *ObjPrueba1* se muestra en la figura 4.8 A través del contenedor web, es posible interactuar perfectamente con los Objetos de Aprendizaje. La herramienta *Exelearning*, posee de manera automática el diseño responsivo a fin de garantizar la correcta visualización en teléfonos móviles.

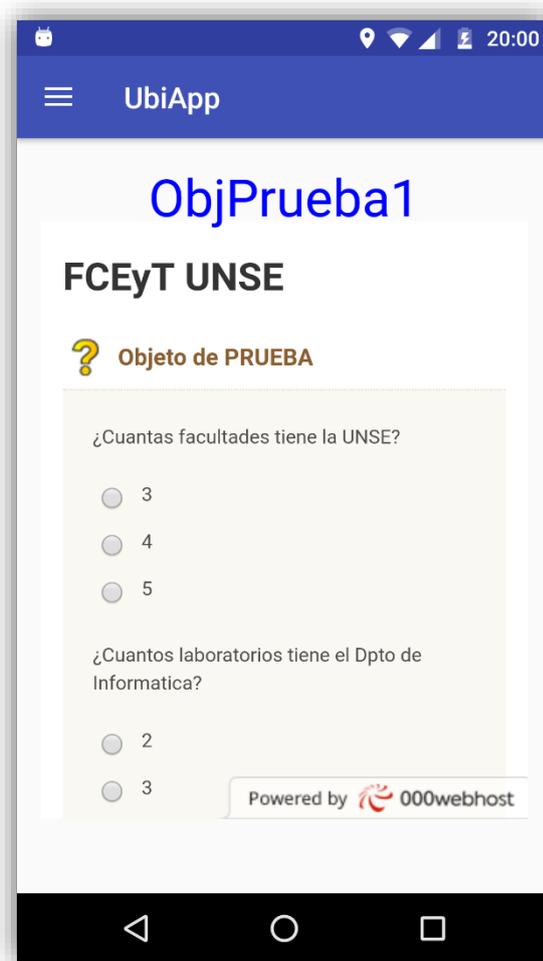


Figura 4.8: Interacción con un Objeto de Aprendizaje

4.6.5. REVISIÓN Y RETROSPECTIVA DEL SPRINT 3

En la reunión se presentó el incremento que consiste en la funcionalidad de recomendación y visualización de objetos de aprendizaje. Se realizó una demostración sobre el trabajo terminado que consistió en iniciar sesión con un usuario de prueba que ingresa por primera vez, y verificar como van cambiando el estado de los OA a medida que el usuario interactúa con ellos.

La Lista de Producto fue revisada para definir qué elementos se terminaron en este Sprint y cuáles son los posibles candidatos para el siguiente Sprint. Los ajustes en la Lista de Producto fueron mínimos.

Con respecto a los problemas surgidos en este Sprint, se debieron principalmente a al desarrollo de los módulos para obtener los resultados de la evaluación del Objeto de Aprendizaje y posteriormente definir el estado del objeto (aprobado/desaprobado) asociado a ese alumno. El resultado era generado en Javascript y fue necesario capturarlo en el momento que el estudiante envía las respuestas para ser evaluadas. Los mismos fueron solucionado satisfactoriamente. Por último, en cuanto a personas, relaciones, procesos y herramientas no hubo cambios significativos.

4.7. SPRINT 4: CONTROL DE ACCESO APLICACIÓN WEB

4.7.1. REUNIÓN DE PLANIFICACIÓN DEL SPRINT 4

A continuación, se responden las preguntas asociadas a esta reunión.

- ¿Qué puede ser terminado en este Sprint?

En este Sprint se diseña e implementa el sistema de control de acceso para la aplicación web. El control de acceso permite que solo los docentes administradores puedan gestionar los parámetros generales del sistema. Se implementa tanto la función para iniciar como para cerrar sesión en la aplicación web. En esta reunión también se discute el objetivo del sprint, el cual es definido en la siguiente sección. La duración de este sprint es de 4 semanas.

- ¿Cómo se conseguirá completar el trabajo seleccionado?

La aplicación web es desarrollada con PHP, HTML, jQuery y Javascript. En primer lugar, se realiza el diseño de la ventana para inicio de sesión. Esta ventana permite ingresar el nombre de usuario y contraseña y autenticarse con la base de datos.

En esta reunión se seleccionan los elementos de la Lista de Producto que pasarán a formar parte de la Lista de Pendientes del Sprint 4. Esta última se presenta en la sección 4.7.3.

4.7.2. OBJETIVO DEL SPRINT 4

El objetivo de este Sprint es desarrollar la primera versión de la aplicación web con la funcionalidad de control de acceso. Esto permitirá iniciar y cerrar sesión a los administradores.

4.7.3. LISTA DE PENDIENTES DEL SPRINT 4

En la tabla 4.5 se presenta la Lista de Pendientes del Sprint 4 donde se detallan las tareas a realizar.

Tabla 4.5: Lista de Pendientes Sprint 4

ID	Requerimientos / Historias de usuario	Tareas	Estim.
1	Como <i>Administrador</i> requiero una aplicación web para administrar los parámetros generales del sistema.	<ul style="list-style-type: none"> - Configurar servidor web. - Diseñar la interfaz gráfica (<i>front-end</i>) inicial con HTML. - Realizar pruebas y depuración. 	10
2	Como <i>Administrador</i> requiero que la aplicación web permita iniciar sesión a docentes administradores para ingresar al sistema.	<ul style="list-style-type: none"> - Diseñar el formulario para iniciar sesión. - Implementar las funciones de autenticación con PHP. - Realizar pruebas y depuración. 	2
3	Como <i>Administrador</i> requiero que la aplicación web cuente con dos tipos de administradores para que el de mayor jerarquía pueda dar de alta nuevos administradores.	<ul style="list-style-type: none"> - Configurar el acceso para el superadministrador. - Configurar el acceso para el administrador. 	2
4	Como <i>Administrador</i> requiero que la aplicación web permita cerrar sesión para salir del sistema.	<ul style="list-style-type: none"> - Diseñar e implementar un botón para cerrar sesión en la aplicación web. - Realizar pruebas y depuración verificando el correcto cierre de sesión. 	1

4.7.4. INCREMENTO DEL SPRINT 4

El incremento del producto de este Sprint es la versión inicial de la aplicación web. La misma cuenta con una pantalla de inicio que permite ingresar el nombre de usuario y la contraseña de los administradores. Por otro lado, al ser la primera versión, en la pantalla principal solo cuenta con un botón que permite el cierre de sesión.

Resultados del Incremento

Solo docentes administradores tienen acceso a esta aplicación para gestionar parámetros generales del sistema. En la figura 4.9 se muestra la captura de pantalla del inicio de sesión y en la figura 4.10 la del cierre de sesión.

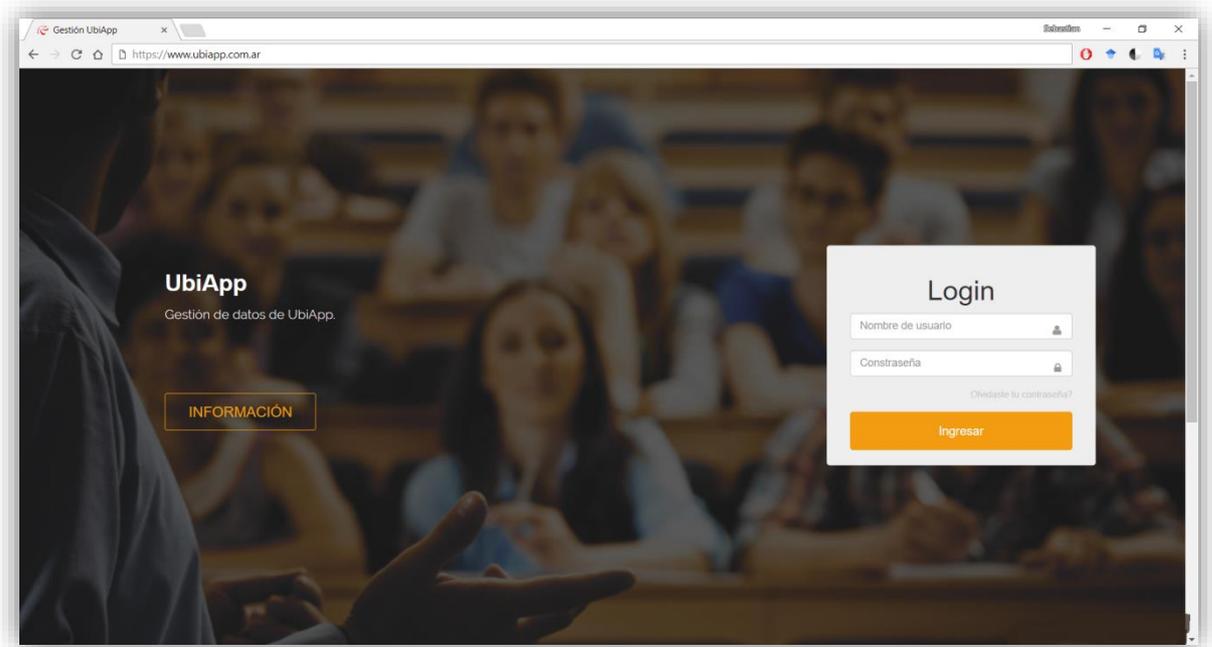


Figura 4.9: Inicio de sesión aplicación web



Figura 4.10: Cierre de sesión aplicación web

4.7.5. REVISIÓN Y RETROSPECTIVA DEL SPRINT 4

En la reunión se presentó el incremento que consiste en la versión inicial de la aplicación web. Se realizó una demostración sobre el trabajo terminado que consistió en ingresar datos de prueba de administradores y comprobar que son autenticados correctamente.

La Lista de Producto fue revisada para definir qué elementos se terminaron en este Sprint y cuáles son los posibles candidatos para el siguiente Sprint. Los ajustes en la Lista de Producto fueron mínimos.

Con respecto a los problemas surgidos en este Sprint, se debieron principalmente a la configuración de ciertas librerías de PHP, pero fueron solucionados satisfactoriamente.

4.8. SPRINT 5: APLICACIÓN WEB

4.8.1. REUNIÓN DE PLANIFICACIÓN DEL SPRINT 5

A continuación, se responden las preguntas asociadas a esta reunión.

- ¿Qué puede ser terminado en este Sprint?

En este Sprint se diseñan e implementan los módulos restantes de la aplicación web que permitirán la administración general del sistema. Los docentes administradores podrán gestionar toda la información relacionada a los estudiantes, objetos de aprendizaje y puntos de interés. En esta reunión también se discute el objetivo del sprint, el cual es definido en la siguiente sección. La duración de este sprint es de 4 semanas.

- ¿Cómo se conseguirá completar el trabajo seleccionado?

En primer lugar, se diseña una barra de menú horizontal en la parte superior de la ventana. Esta barra está compuesta por pestañas donde se podrá acceder tanto a información relacionada a la historia académica y datos personales de los estudiantes, como así también a información relacionada a objetos de aprendizaje y puntos de interés. Luego, se implementan las funciones que permiten realizar altas, bajas y modificaciones en la base de datos.

Cabe destacar que en la *Historia Académica* se muestran aquellos Objetos de Aprendizaje realizados por el estudiante, como así también los objetos no realizados. Esto tiene por finalidad mantener en un mismo panel toda la información entre estudiantes y objetos de aprendizaje.

En esta reunión se seleccionan los elementos de la Lista de Producto que pasarán a formar parte de la Lista de Pendientes del Sprint 5. Esta última se presenta en la sección 4.8.3.

4.8.2. OBJETIVO DEL SPRINT 5

El objetivo de este Sprint es desarrollar e implementar los módulos restantes de la aplicación web, que permitirán realizar altas, bajas y modificaciones en la base de datos.

4.8.3. LISTA DE PENDIENTES DEL SPRINT 5

En la tabla 4.6 se presenta la Lista de Pendientes del Sprint 5 donde se detallan las tareas a realizar.

Tabla 4.6: Lista de Pendientes Sprint 5

ID	Requerimientos / Historias de usuario	Tareas	Estim.
1	Como <i>Administrador</i> requiero que la aplicación web permita identificar el ingreso de un administrador para mostrarle el panel de control correspondiente.	<ul style="list-style-type: none"> - Diseñar la ventana y paneles tanto para el superadministrador como para el administrador. - Realizar pruebas y depuración. 	3
2	Como <i>Administrador</i> requiero que la aplicación web permita dar de alta un alumno para que el mismo pueda iniciar sesión desde la aplicación móvil.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	2
3	Como <i>Administrador</i> requiero que la aplicación web permita dar de baja un alumno y toda su información relacionada para eliminarlo del sistema	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	2
4	Como <i>Administrador</i> requiero que la aplicación web permita modificar información relacionada al alumno para actualizar el modelo del estudiante.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	2
5	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel alumnos para visualizar información relacionada a ellos.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Realizar pruebas y depuración. 	1

6	Como <i>Administrador</i> requiero que la aplicación web permita dar de alta la referencia a un OA, vinculado a un PI, para que esté disponible desde la aplicación móvil.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Crear los metadatos propios para esta aplicación. - Desarrollar e implementar la funcionalidad. - Realizar pruebas y depuración. 	1
7	Como <i>Administrador</i> requiero que la aplicación web permita dar de baja la referencia a un OA para que no esté disponible desde la aplicación móvil.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Realizar pruebas y depuración. 	1
8	Como <i>Administrador</i> requiero que la aplicación web permita modificar información relacionada a un OA para mantener actualizados los metadatos propios.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Realizar pruebas y depuración. 	1
9	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel objetos de aprendizaje para visualizar información relacionada a ellos.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Realizar pruebas y depuración. 	1
10	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel historia académica para visualizar información relacionada con alumnos y objetos de aprendizaje.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	2
11	Como <i>Administrador</i> requiero que la aplicación web permita dar de alta un PI para que esté disponible desde la aplicación móvil.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	1

12	Como <i>Administrador</i> requiero que la aplicación web permita dar de baja un PI para que no esté disponible desde la aplicación móvil.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	1
13	Como <i>Administrador</i> requiero que la aplicación web permita modificar un PI para actualizar la información relacionada a él.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración. 	1
14	Como <i>Administrador</i> requiero que la aplicación web cuente con un panel punto de interés para visualizar información relacionada a ellos.	<ul style="list-style-type: none"> - Diseñar la nueva funcionalidad. - Desarrollar e implementar la funcionalidad. - Verificar consistencia de los datos. - Realizar pruebas y depuración 	1

4.8.4. INCREMENTO DEL SPRINT 5

El incremento del producto de este Sprint es la aplicación web completamente funcional, con un menú de navegación, el cual permite gestionar información relacionada a estudiantes, objetos de aprendizaje y puntos de interés. Este incremento se integra al control de acceso desarrollado en el Sprint anterior para conformar la aplicación web completa.

Resultados del Incremento

Al iniciar sesión un docente administrador, el primer panel que muestra la aplicación web es “*Historial de Aprendizaje*”. Este panel muestra el estado de cada Objeto de Aprendizaje asociado a cada estudiante. Se pueden realizar búsquedas por el apellido de un estudiante y además, aplicar filtros por Objetos de Aprendizaje, Puntos de Interés y Estado (figura 13).

Es necesario destacar que la aplicación web cuenta con dos tipos de administradores para la gestión de los parámetros del sistema. Por un lado, un *Superadministrador* capaz de dar de alta *Administradores* y a su vez estudiantes que utilizarán la aplicación móvil. Por otro lado, un *Administrador* capaz de dar de alta solo estudiantes.

En la figura 4.11, al haber ingresado como *Superadministrador*, puede observarse el panel “*Usuarios*” desde donde puede darse de alta *Administradores*.

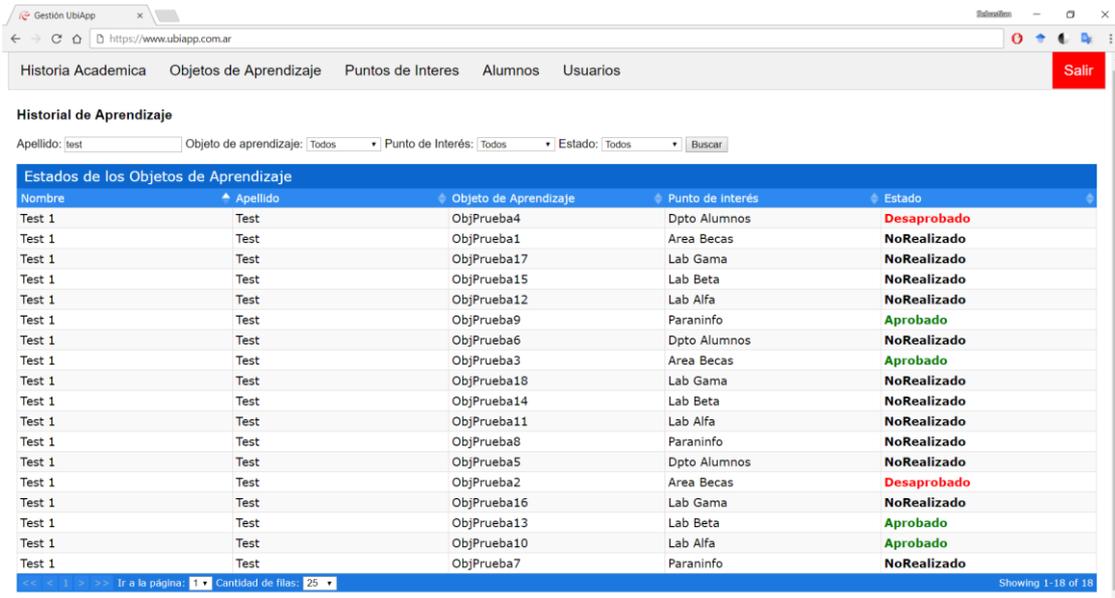


Figura 4.11: Panel Historia Académica

En las siguientes imágenes (figura 4.12, 4.13, 4.14, 4.15) se muestra los paneles restantes. En cada uno se pueden realizar consultas, altas, bajas y modificaciones.

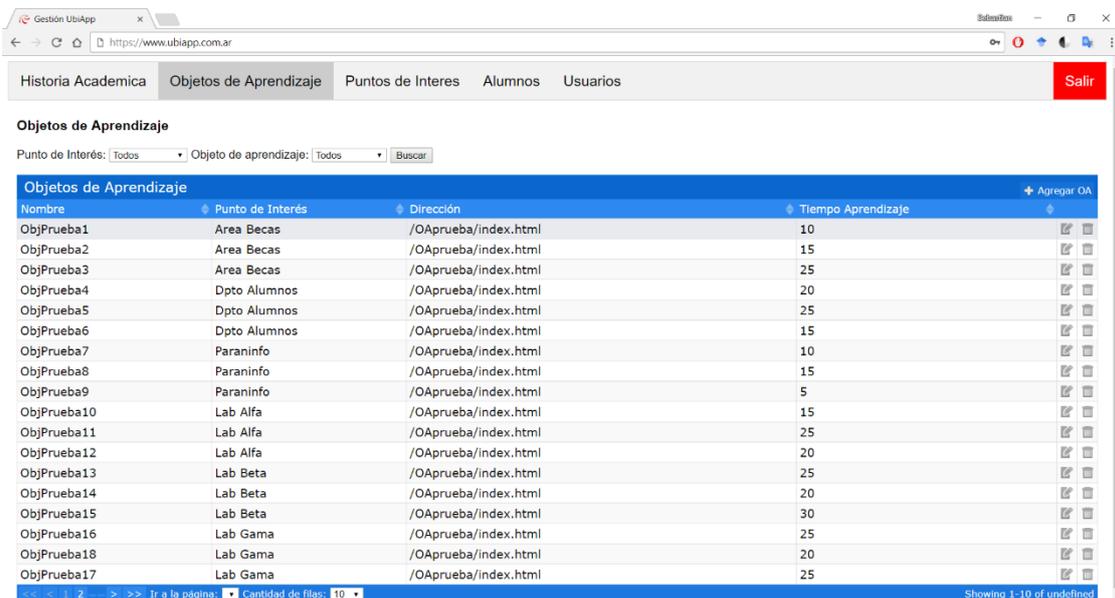


Figura 4.12: Panel Objetos de Aprendizaje

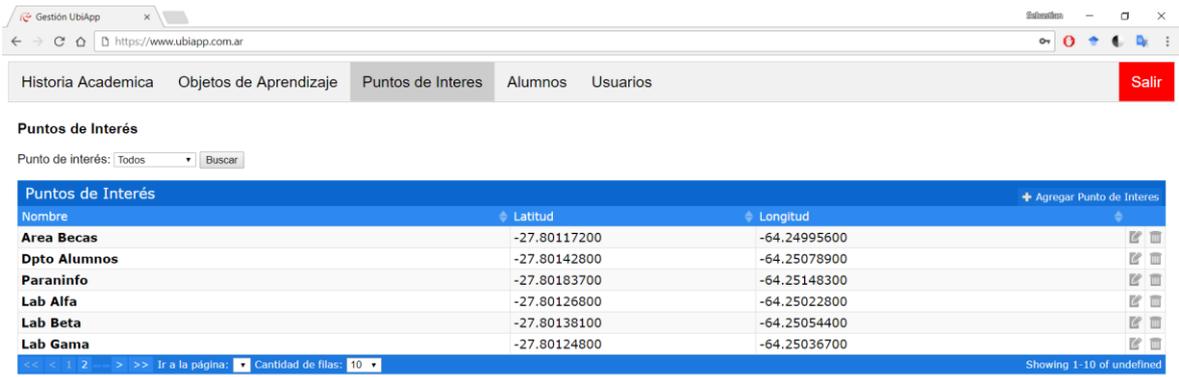


Figura 4.13: Panel Puntos de Interés

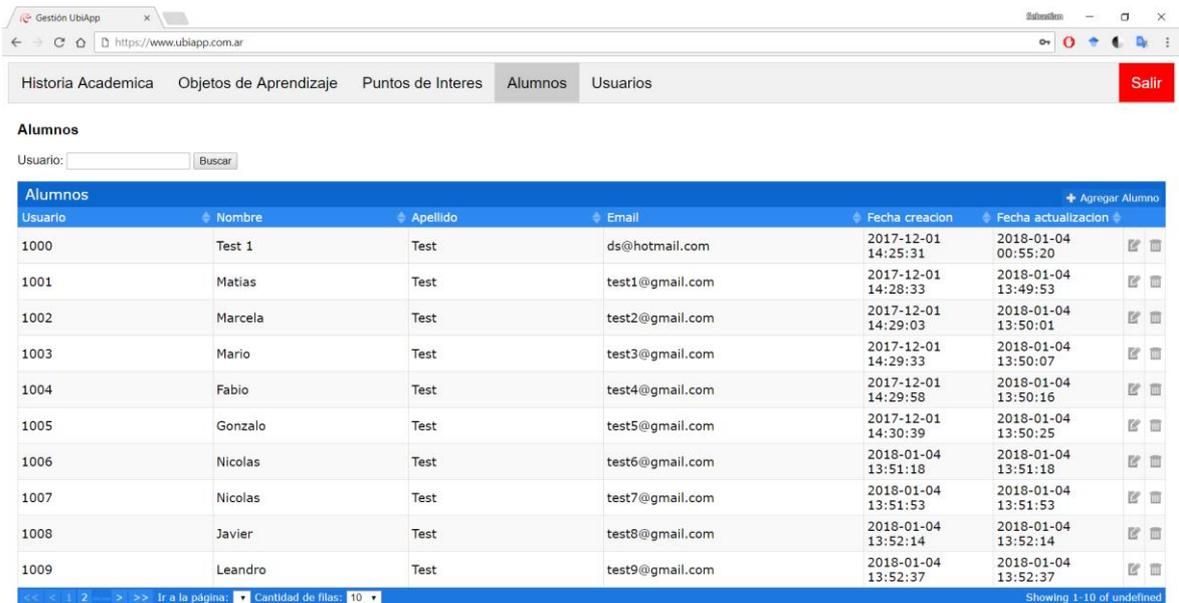


Figura 4.14: Panel Alumnos

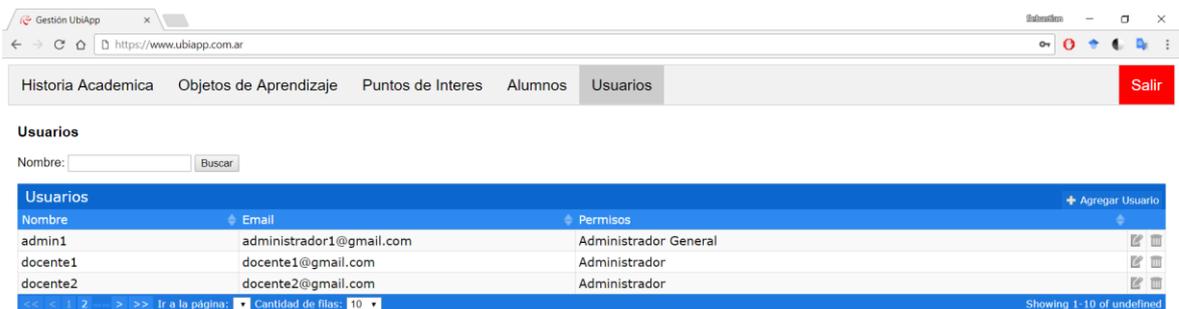


Figura 4.15: Panel Usuarios

4.8.5. REVISIÓN Y RETROSPECTIVA DEL SPRINT 5

En la reunión se presentó el incremento que consiste en la funcionalidad para gestionar información relacionada a estudiantes, objetos de aprendizaje y puntos de interés. Se realizó una demostración sobre el trabajo terminado que consistió en realizar diferentes altas, bajas y modificaciones y comprobar que funciona como debería hacerlo.

La Lista de Producto fue revisada para definir qué elementos se terminaron en este Sprint y cuáles son los posibles candidatos para el siguiente Sprint. Los ajustes en la Lista de Producto fueron mínimos.

Con respecto a los problemas surgidos en este Sprint, se debieron principalmente a la codificación de ciertas funciones en jQuery y Javascript. Los mismos fueron solucionados satisfactoriamente. Por último, en cuanto a personas, relaciones, procesos y herramientas no hubo cambios significativos.

4.9. SPRINT 6: DESPLIEGUE

4.9.1. REUNIÓN DE PLANIFICACIÓN DEL SPRINT 6

A continuación, se responden las preguntas asociadas a esta reunión.

- ¿Qué puede ser terminado en este Sprint?

En este Sprint, una vez integrado el sistema, se realizan las tareas finales para el despliegue de la aplicación. Estas tareas incluyen documentación y pruebas finales. En esta reunión también se discute el objetivo del sprint, el cual es definido en la siguiente sección.

- ¿Cómo se conseguirá completar el trabajo seleccionado?

En primer lugar, se contrata un hosting en internet y se configura la base de datos, webservices y la aplicación web para que funcione con salida a internet. Posteriormente, se suben todos los elementos al hosting y se activa la salida a través de internet. Por último, se instala la aplicación móvil en teléfonos de prueba.

En esta reunión se seleccionan los elementos de la Lista de Producto que pasarán a formar parte de la Lista de Pendientes del Sprint 2. Esta última se presenta en la sección 4.9.3.

4.9.2. OBJETIVO DEL SPRINT 6

El objetivo de este Sprint es realizar pruebas finales con el sistema integrado y posteriormente desplegarlo.

4.9.3. LISTA DE PENDIENTES DEL SPRINT 6

En la tabla 4.7 se presenta la Lista de Pendientes del Sprint 6 donde se detallan las tareas a realizar.

Tabla 4.7: Lista de Pendientes Sprint 6

ID	Requerimientos / Historias de usuario	Tareas	Estim.
1	Como <i>Administrador</i> requiero que una vez integrado el sistema se realicen pruebas de validación para corroborar que satisface todos los requerimientos.	<ul style="list-style-type: none"> - Configurar base de datos, webservices y aplicación web. - Subir al hosting los elementos para que la aplicación móvil pueda funcionar a través de internet. - Realizar pruebas y depuración. 	10
2	Como <i>Administrador</i> requiero que la aplicación móvil cuente con un manual de usuario.	<ul style="list-style-type: none"> - Redactar manual de usuario. - Incluirlo en el anexo de este trabajo. 	3

4.9.4. INCREMENTO DEL SPRINT 6

El incremento del producto de este Sprint es el sistema funcionando completamente, que incluye tanto la aplicación móvil como la aplicación web.

Resultado del Incremento

Se contrató el servicio de un hosting gratuito soportado por *Hostinger* y se procede a subir al servidor la base de datos (figura 4.16), webservices, la aplicación web y los Objetos de Aprendizaje de prueba (figura 4.17).

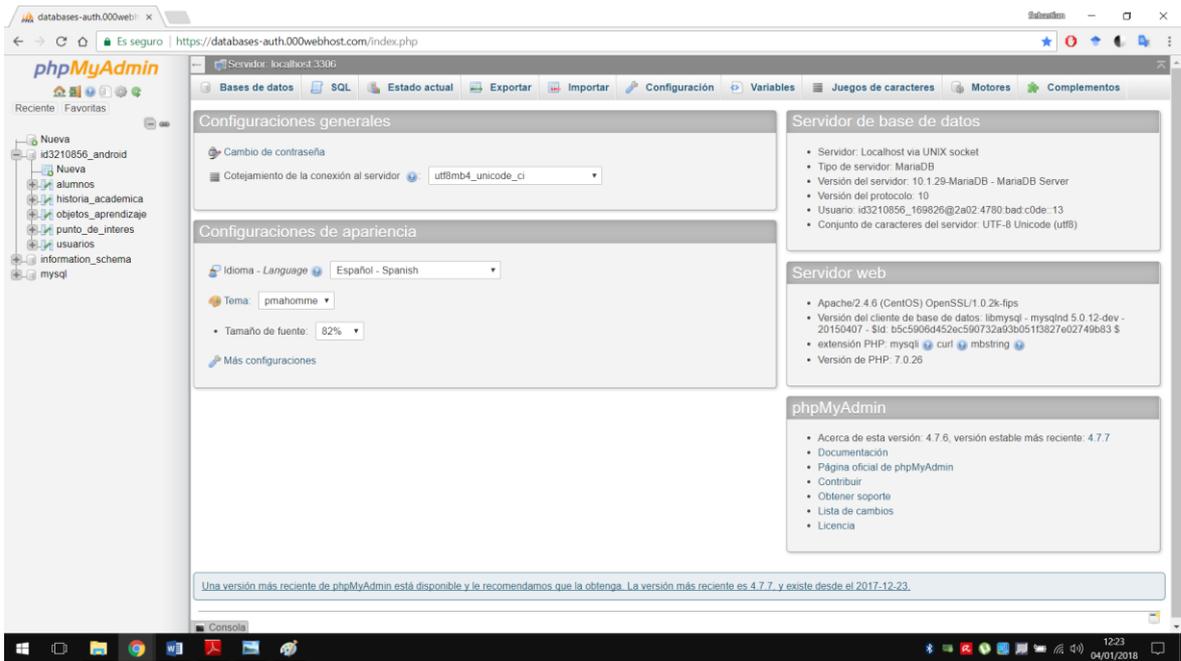


Figura 4.16: Carga de la BD a través de *phpMyAdmin*

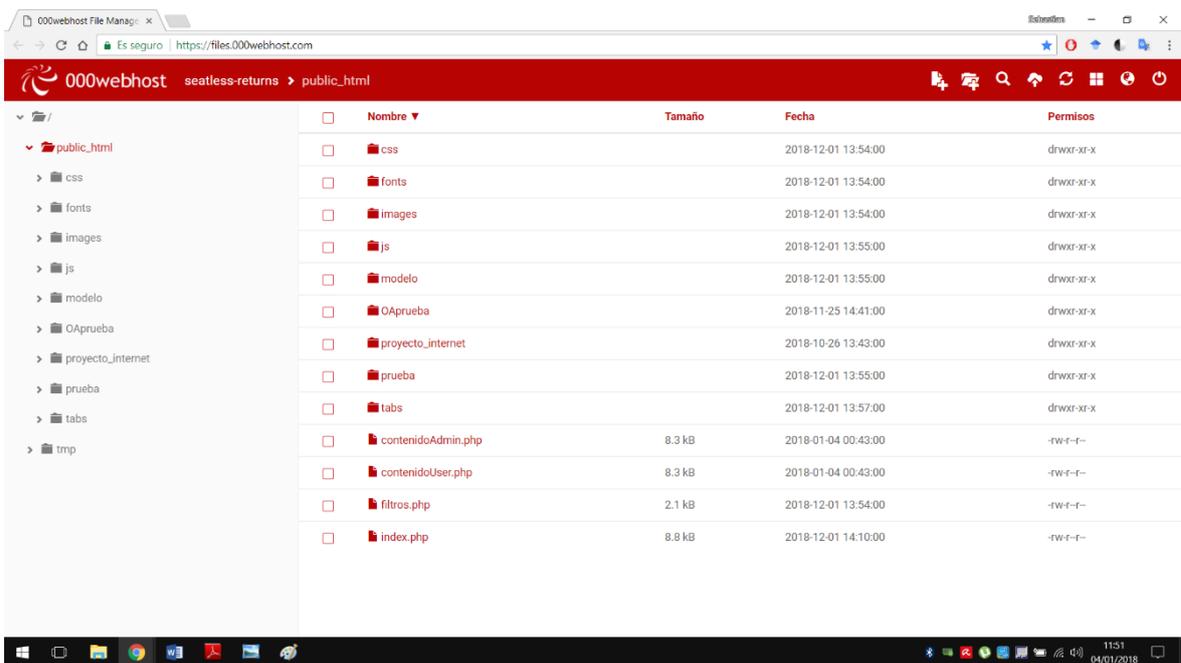


Figura 4.17: Carga de archivos a través de FTP

4.9.5. REVISIÓN Y RETROSPECTIVA DEL SPRINT 6

En la reunión se presentó el incremento que consiste en la aplicación móvil funcionando correctamente en el ámbito de la Universidad Nacional de Santiago del Estero. Se realizó una demostración del trabajo terminado que consistió en recorrer diferentes lugares de la universidad, comprobar que recomienda puntos de interés cercanos y luego poder interactuar con los objetos de aprendizaje perteneciente a cada punto de interés.

Con respecto a los problemas surgidos en este Sprint, fueron mínimos y se debieron en gran parte a configuraciones en el hosting. Los problemas fueron solucionados satisfactoriamente y se culminó con todos los elementos de la Lista de Producto.

4.10 HERRAMIENTAS DE SOFTWARE UTILIZADAS

Para llevar a cabo el desarrollo de software del presente trabajo se utilizaron las siguientes herramientas:

En lo que respecta a la aplicación móvil, fue desarrollada completamente con el software **Android Studio**, entorno de desarrollo integrado (IDE) oficial para el desarrollo de aplicaciones para Android desde diciembre de 2014 y se basa en **IntelliJ IDEA** (Google, 2017). Además del potente editor de códigos y las herramientas para desarrolladores de IntelliJ, Android Studio ofrece aún más funciones que aumentan la productividad durante la compilación de apps para Android, como las siguientes: un sistema de compilación basado en Gradle flexible, un emulador rápido con varias funciones, un entorno unificado en el que puedes realizar desarrollos para todos los dispositivos Android, entre muchas otras potentes funciones.

Por otro lado, la aplicación web fue desarrollada utilizando diferentes tecnologías. Dado que el objetivo principal de la aplicación web es realizar altas, bajas y modificaciones, resultó conveniente implementar **jTable**, que es un plugin de **jQuery** utilizado para crear tablas CRUD (Create, Read, Update, Delete) basado en **AJAX**. Para el desarrollo front-end se utilizó **HTML** y **Bootstrap** y para el back-end se utilizó **PHP** y **JavaScript**.

En lo que respecta a la base de datos se utilizó **MySQL**, el sistema de gestión de base de datos relacional opensource más famoso del mundo. Para probar continuamente tanto la aplicación móvil como la aplicación web se utilizó **XAMPP**, un servidor web de plataforma software libre, que está compuesto principalmente por el sistema de gestión de bases de datos MySQL y el servidor web **Apache**.

Como editor de código para desarrollar la aplicación web se utilizó **Brackets**, un moderno editor de texto opensource. Para el control de versiones del software se utilizó **Git** y **GitKraken**, interfaz gráfica para Git. El repositorio de control de versiones utilizado fue **GitLab**.

Finalmente, para el despliegue de la aplicación móvil se llevó a cabo con **000webhost**, un hosting gratuito soportado por Hostinger que ofrece herramientas tales como servidor web, ftp y base de datos.

CAPÍTULO 5

PRUEBAS Y ANÁLISIS DE RESULTADOS

5.1. INTRODUCCIÓN

En este capítulo se presenta los diferentes tipos de pruebas realizadas a la aplicación móvil Android. Debido a la dinámica de funcionamiento de los dispositivos móviles, es necesario llevar a cabo múltiples pruebas con el objetivo de depurar errores y brindar una experiencia de calidad en el uso de la aplicación.

La configuración de hardware particular incluida en los teléfonos y sus diferentes tipos de conexiones tales como GPS, conexiones de internet a través de redes de datos y Wi-Fi, conllevan a la necesidad de probar y experimentar las aplicaciones móviles tanto en su ambiente real de funcionamiento como dentro del laboratorio.

Para el caso particular de este trabajo, se desarrollaron cuatro tipos de pruebas. La primera, es una prueba manual realizada sobre diferentes dispositivos físicos y virtuales, con el fin de verificar el correcto funcionamiento de la aplicación cuando atraviesa las distintas etapas del ciclo de vida. La segunda prueba es de rendimiento y tiene como objetivo medir el consumo de memoria RAM y CPU durante el uso de la aplicación. La tercera, es una prueba de estrés, que exige al máximo la aplicación con el fin de que se detenga y poder comprobar donde ocurren los errores. La cuarta y última prueba, consiste en realizar una experimentación con un grupo reducido de estudiantes en el ámbito de la UNSE. El propósito de esta última prueba es determinar entre otras cosas, si la aplicación es intuitiva, atractiva y fácil de usar.

5.2. PRUEBAS MANUALES A LA APLICACIÓN ANDROID

Las principales características probadas en la aplicación móvil para asegurar un correcto funcionamiento fueron las siguientes:

- *Eventos del ciclo de vida de las Activities y Fragments:* La aplicación está compuesta por dos *Activities* principales, *LoginActivity* y *MainActivity*. Esta última es encargada de gestionar los *Fragments*: *ListarPiFragment*, *ListarOAFragment* y *MostrarOAFragment*.

Las pruebas consistieron en verificar que las *Activities* se desempeñan correctamente al atravesar los diferentes estados del ciclo de vida, por ejemplo, cuando la aplicación pasa a segundo plano o cuando se detiene. A continuación, en la tabla 5.1, se detallan las pruebas realizadas en cada *Activity* y *Fragment*:

Tabla 5.1: Pruebas realizadas sobre cada *Activity*

LoginActivity	<ul style="list-style-type: none"> • Control de inicio de sesión. • Control de inicio de sesión automático. • Control de cierre de sesión. • Conexión al servidor para autenticación. • Control sin conexión a internet. • Control al avanzar a la <i>MainActivity</i>. 	
MainActivity	ListarPiFragment	<ul style="list-style-type: none"> • Control al ingresar tiempo disponible. • Control al listar puntos de interés. • Verificar algoritmos para obtener las distancias entre el dispositivo y cada punto de interés. • Control al avanzar al <i>Fragment</i> <i>ListarOAFragment</i>.
	ListarOAFragment	<ul style="list-style-type: none"> • Control al listar OAs con conexión a internet. • Control al listar OAs sin conexión a internet. • Control al listar OAs cuando el tiempo disponible del estudiante sea menor que el tiempo de aprendizaje de los OAs. • Control del estado de los OAs antes y luego de interactuar con ellos. • Control al avanzar al <i>Fragment</i> <i>MostrarOAFragment</i>.

	MostrarOAFragment	<ul style="list-style-type: none"> • Control al cargar un OA seleccionado previamente. • Control al realizar una evaluación sobre un OA. • Control al actualizar un estado de un OA. • Control al retroceder al Fragment ListarOAFragment.
--	-------------------	--

- *Lectura y escritura en Base de Datos remota:* se realizaron pruebas para verificar el correcto funcionamiento de la aplicación móvil cuando realiza peticiones al servidor al momento de iniciar sesión y cuando se obtiene el listado de Objetos de Aprendizaje para cada estudiante. También se verifico que la base de datos se actualice correctamente luego de interactuar con un Objeto de Aprendizaje y definir un nuevo estado para ese objeto.
- *Diferentes tipos de dispositivos:* la aplicación móvil fue probada en diferentes dispositivos tanto físicos como emuladores con el fin de detectar errores funcionales y gráficos. En la tabla 5.2 se muestra en detalle los dispositivos sobre los cuales se probó la aplicación de manera exitosa.

Tabla 5.2: Dispositivos móviles usados para probar la aplicación

Dispositivo	Versión SO	Procesador	Resolución
Motorola Moto X Play	Marshmallow 6.0.1	Snapdragon 615 1.7GHz	1920 x 1080 px
Motorola Moto X Play	Nougat 7.1.1	Snapdragon 615 1.7GHz	1920 x 1080 px
Motorola G3	Marshmallow 6.0	Snapdragon 410 1.4GHz	1280 x 720 px
Samsung S6 Edge Plus	Nougat 7.0	Exynos 7420 2.1GHz	2560 x 1440 px
Samsung S8	Nougat 7.0	Exynos 8895 2.35GHz	2960 x 1440 px
Sony Ericsson ZL	Lollipop 5.1.1	Snapdragon S4 1.5GHz	1920 x 1080 px
Samsung J1 Ace 2016	Lollipop 5.1.1	Cortex A7 1.30 GHz	800 x 480 px
Motorola Moto E2	Marshmallow 6.0	Snapdragon 200 1.2 GHz	960 x 540 px
Motorola Moto E2	Lollipop 5.1	Snapdragon 200 1.2 GHz	960 x 540 px
LG Nexus 5X	Oreo 8.0.0	Snapdragon 808 1.8 GHz	1920 x 1080 px

Para concluir, las pruebas manuales realizadas en gran parte en el laboratorio y en una pequeña parte en el patio de la UNSE, permitieron detectar y corregir errores respecto al algoritmo para calcular las distancias entre el dispositivo Android y cada punto de interés. Como así también, fueron de suma utilidad para comprobar que la aplicación móvil obtiene toda la información necesaria para realizar la personalización a cada estudiante. Finalmente, se pudo comprobar que la aplicación funciona de manera adecuada en por lo menos 20 (veinte) smartphones con diferentes sistemas operativos.

5.3. PRUEBAS DE RENDIMIENTO

Para esta prueba, se utilizó una herramienta online denominada *Testdroid Cloud*, que permite medir el consumo de memoria y procesador de la aplicación móvil durante diez minutos de uso continuo. La herramienta provee de dos dispositivos virtuales de prueba, por un lado, un *LG Google Nexus 5 Android 6.0.1* y por otro lado un *LG Google Nexus 5 D820 Android 5.0*.

Los resultados arrojados por la herramienta *Testdroid Cloud* referidos al consumo de memoria RAM para cada dispositivo se detalla en la figura 5.1 y figura 5.3. Respecto al consumo de procesador para cada dispositivo se muestra en la figura 5.2 y figura 5.4.

- **LG Google Nexus 5**

Consumo de memoria RAM

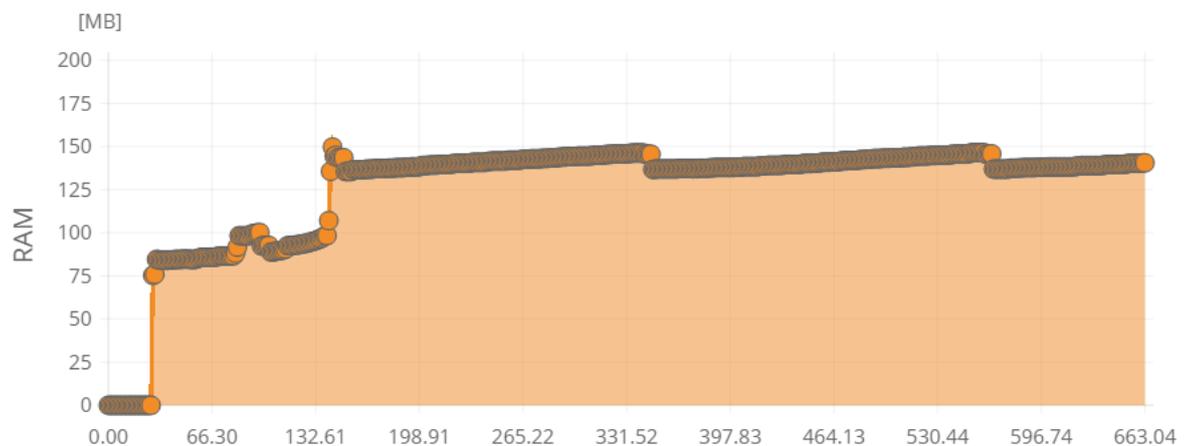


Figura 5.1: Consumo de RAM de Nexus 5

El consumo máximo registrado por la herramienta es de aproximadamente 150MB de memoria RAM, el cual se mantiene constante durante los siguientes minutos de uso de la aplicación.

Consumo de Procesador

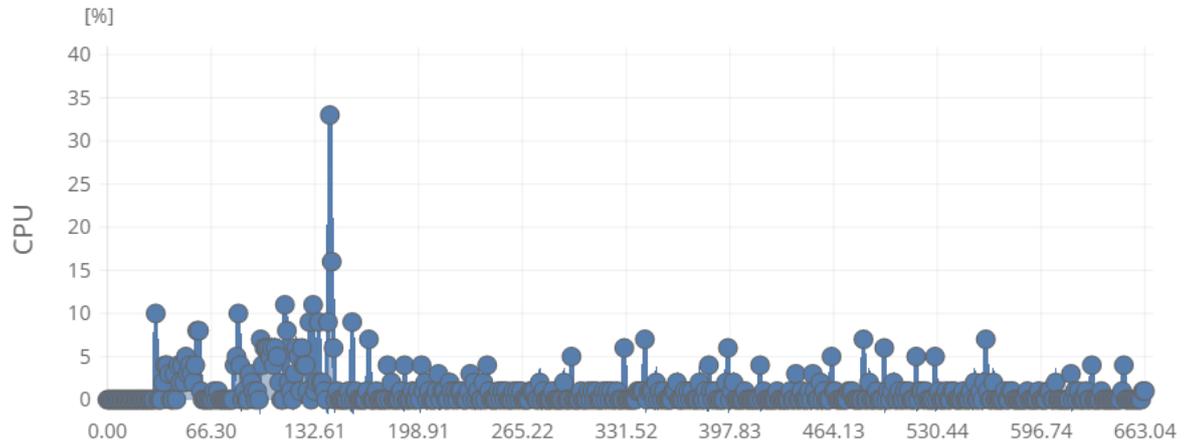


Figura 5.2: Consumo de Procesador de Nexus 5

El consumo máximo registrado fue del 33% de utilización del procesador durante un segundo aproximadamente, luego mantuvo picos de hasta un 11% durante los diez minutos de prueba.

- **LG Google Nexus 5 D820**

Consumo de memoria RAM

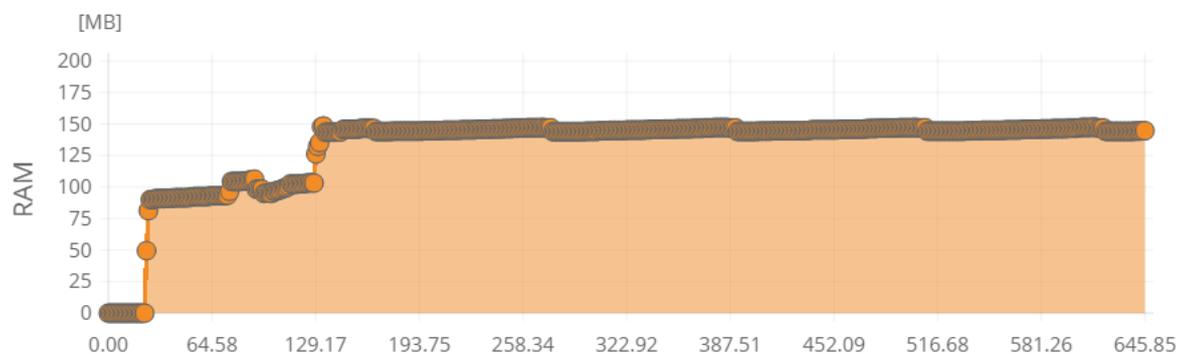


Figura 5.3: Consumo de RAM de Nexus 5 D820

El consumo máximo registrado por la herramienta es de aproximadamente 145MB de memoria RAM, similar al consumo medido con el primer dispositivo virtual. Este consumo se mantiene constante durante los siguientes minutos de uso de la aplicación.

Consumo de Procesador

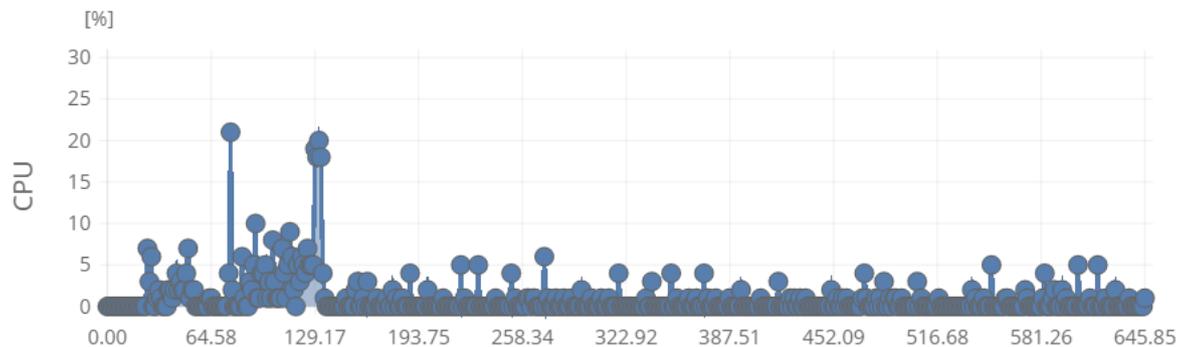


Figura 5.4: Consumo de Procesador de Nexus 5 D820

Para este caso, el consumo máximo registrado fue del 21% de utilización del procesador durante un segundo, luego mantuvo picos de hasta un 10% durante toda la prueba.

Cabe destacar, que además de haber medido el consumo de la aplicación con los dispositivos virtuales, también se realizó una medición con un dispositivo físico (Motorola Moto X Play) utilizando la herramienta de monitoreo continuo incluida en el IDE Android Studio. Los resultados arrojados por esta última fueron similares a los medidos con la herramienta online.

En conclusión, la prueba de rendimiento sirvió para conocer la utilización de memoria RAM y procesador cuando la aplicación ejecuta los procesos más complejos. Considerando que los teléfonos modernos vienen con 1GB de memoria como mínimo, que la aplicación móvil haya utilizado 150MB (15% del total sobre 1024MB) se considera aceptable. Respecto al uso del procesador (de un 10%), se considera también aceptable.

5.4. PRUEBAS DE ESTRÉS

Estas pruebas se realizan sobre la interfaz de la aplicación Android utilizando tanto dispositivos reales como emuladores. La herramienta utilizada para este tipo de pruebas es Monkey. Esta herramienta, provista por Android Studio bajo línea de comandos, genera secuencias pseudo-aleatorias de eventos de usuario como clics, toques o gestos, así como una serie de eventos a nivel del sistema. Con Monkey es posible esforzar la aplicación móvil para comprobar si se detiene o congela. Además, en el caso de encontrar algún error, es posible verificarlo a través de un log.

La prueba de estrés se realizó sobre las Actividades principales de la aplicación: *LoginActivity* y *MainActivity*. Bajo el control de la *MainActivity* se encuentran los fragmentos (*fragments*) encargados de listar los puntos de interés, listado y visualización de objetos de aprendizaje, entre otros.

Los dispositivos utilizados para esta prueba fueron dos, un teléfono real Motorola Moto X Play versión de Android 6.0.1 y un emulador Nexus 5 versión de Android 6.0.

La primera prueba se realizó sobre el *LoginActivity*. La prueba consistió en inyectar 5000 eventos sobre el formulario de login, con una duración aproximada de 40 segundos en total. El resultado de la prueba fue exitoso ya que la aplicación no se detuvo en ningún momento. Los eventos generados por Monkey se realizaron rápidamente y consistieron en ingresar diferentes caracteres en el formulario, presionar múltiples veces el botón de iniciar sesión, conectar y desconectar múltiples veces el GPS, Wi-fi, paquete de datos, entre muchos eventos más.

El resultado arrojado por Monkey puede verse en la figura 5.5. Cuando la herramienta indica la leyenda “Monkey Finished” significa que no hubo errores durante el procedimiento.

```

Administrador: Windows PowerShell
:Sending Touch (ACTION_UP): 0:(59.288555,383.14587)
:Sending Trackball (ACTION_MOVE): 0:(3.0,-4.0)
:Sending Touch (ACTION_DOWN): 0:(302.0,463.0)
:Sending Touch (ACTION_UP): 0:(309.57935,461.49716)
:Sending Touch (ACTION_DOWN): 0:(551.0,167.0)
:Sending Touch (ACTION_UP): 0:(592.1294,192.86716)
:Sending Touch (ACTION_DOWN): 0:(837.0,1565.0)
:Sending Touch (ACTION_UP): 0:(846.42664,1545.7518)
:Sending Touch (ACTION_DOWN): 0:(241.0,1096.0)
:Sending Touch (ACTION_UP): 0:(242.28278,1086.276)
:Sending Touch (ACTION_DOWN): 0:(406.0,1137.0)
:Sending Touch (ACTION_UP): 0:(402.82813,1139.5812)
:Sending Trackball (ACTION_MOVE): 0:(-5.0,-4.0)
:Sending Touch (ACTION_DOWN): 0:(898.0,495.0)
:Sending Touch (ACTION_UP): 0:(911.2949,498.09756)
:Sending Touch (ACTION_DOWN): 0:(975.0,1729.0)
:Sending Touch (ACTION_UP): 0:(980.759,1725.7316)
:Sending Trackball (ACTION_MOVE): 0:(-4.0,-4.0)
//[calendar_time:2017-12-26 21:57:47.748 system_uptime:171116611]
// Sending event #4900
:Sending Trackball (ACTION_MOVE): 0:(-2.0,-5.0)
:Sending Touch (ACTION_DOWN): 0:(874.0,889.0)
:Sending Touch (ACTION_UP): 0:(882.7313,902.2563)
:Sending Trackball (ACTION_MOVE): 0:(3.0,0.0)
:Sending Trackball (ACTION_MOVE): 0:(-2.0,-2.0)
:Sending Touch (ACTION_DOWN): 0:(202.0,1337.0)
:Sending Touch (ACTION_UP): 0:(199.76878,1341.516)
:Sending Touch (ACTION_DOWN): 0:(691.0,1506.0)
:Sending Touch (ACTION_UP): 0:(681.2868,1564.3055)
:Sending Touch (ACTION_DOWN): 0:(327.0,813.0)
:Sending Touch (ACTION_UP): 0:(354.70172,720.83215)
:Sending Trackball (ACTION_MOVE): 0:(-4.0,-1.0)
:Sending Touch (ACTION_DOWN): 0:(303.0,83.0)
:Sending Touch (ACTION_UP): 0:(302.9576,96.71295)
:Sending Touch (ACTION_DOWN): 0:(98.0,536.0)
Events injected: 5000
:Sending rotation degree=0, persist=false
:Dropped: keys=0 pointers=0 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=36495ms (0ms mobile, 0ms wifi, 36495ms not connected)
// Monkey finished
shell@lux:/$
PS E:\AndroidSDK\platform-tools>

```

Figura 5.5: Resultado de Monkey sobre LoginActivity

La segunda prueba se realizó sobre el *MainActivity*. Al igual que en la primera prueba, esta consistió en inyectar 5000 eventos sobre la pantalla principal inmediata al inicio de sesión. La prueba duró aproximadamente 40 segundos y el resultado fue exitoso, es decir, la aplicación no se detuvo en ningún momento.

A diferencia de la primera prueba, Monkey se encargó de ejercitar al máximo los diferentes módulos que dependen de *MainActivity*. Entre los más importantes se encuentran los módulos encargados de listar puntos de interés, listar e interactuar con objetos de aprendizaje, etc. El resultado arrojado por Monkey puede verse en la figura 5.6.

```

Administrador: Windows PowerShell
:Sending Touch (ACTION_UP): 0: (837.33563,553.4236)
:Sending Touch (ACTION_DOWN): 0: (1044.0,1393.0)
:Sending Touch (ACTION_UP): 0: (1080.0,1330.9497)
:Sending Touch (ACTION_DOWN): 0: (956.0,631.0)
:Sending Touch (ACTION_UP): 0: (935.48474,640.39874)
:Sending Trackball (ACTION_MOVE): 0: (-5.0,2.0)
:Sending Touch (ACTION_DOWN): 0: (855.0,1725.0)
:Sending Touch (ACTION_UP): 0: (854.48346,1722.7617)
:Sending Trackball (ACTION_MOVE): 0: (-4.0,-4.0)
:Sending Touch (ACTION_DOWN): 0: (108.0,304.0)
:Sending Touch (ACTION_UP): 0: (111.38354,307.31644)
// [calendar_time:2017-12-27 20:26:54.632 system_uptime:252063494]
// Sending event #4900
// [calendar_time:2017-12-27 20:26:54.633 system_uptime:252063494]
// Sending event #4900
:Sending Trackball (ACTION_MOVE): 0: (2.0,1.0)
:Sending Trackball (ACTION_MOVE): 0: (2.0,4.0)
:Sending Trackball (ACTION_MOVE): 0: (2.0,-1.0)
:Sending Trackball (ACTION_MOVE): 0: (0.0,3.0)
:Sending Trackball (ACTION_UP): 0: (0.0,0.0)
:Sending Trackball (ACTION_MOVE): 0: (-4.0,1.0)
:Sending Touch (ACTION_DOWN): 0: (962.0,909.0)
:Sending Touch (ACTION_UP): 0: (945.55707,924.29126)
:Sending Touch (ACTION_DOWN): 0: (659.0,1141.0)
:Sending Touch (ACTION_UP): 0: (643.3235,1131.1537)
:Sending Touch (ACTION_DOWN): 0: (800.0,405.0)
:Sending Touch (ACTION_UP): 0: (858.0108,483.40076)
:Sending Touch (ACTION_DOWN): 0: (68.0,1661.0)
:Sending Touch (ACTION_UP): 0: (30.35415,1590.173)
:Sending Touch (ACTION_DOWN): 0: (592.0,1081.0)
:Sending Touch (ACTION_UP): 0: (533.20514,1052.8168)
:Sending Touch (ACTION_DOWN): 0: (38.0,1740.0)
:Sending Touch (ACTION_UP): 0: (37.447083,1738.0964)
:Sending Touch (ACTION_DOWN): 0: (923.0,386.0)
Events injected: 5000
:Sending rotation degree=0, persist=false
:Dropped: keys=0 pointers=16 trackballs=0 flips=0 rotations=0
## Network stats: elapsed time=29227ms (0ms mobile, 0ms wifi, 29227ms not connected)
// Monkey finished
shell@lux: / $

```

Figura 5.6: Resultado de Monkey sobre MainActivity

Cabe destacar que se ejecutó tres veces la herramienta Monkey en cada dispositivo y sobre cada *Activity*. En todos los casos los resultados fueron exitosos. La figura 5.5 y figura 5.6 corresponden a la última prueba en ambos casos.

5.5. PRUEBAS DE USABILIDAD

El funcionamiento de la aplicación móvil fue validado mediante una experimentación reducida compuesta por 5 (cinco) estudiantes pertenecientes a la Facultad de Ciencias Exactas y Tecnologías. La experimentación se realizó en el ámbito de la Universidad Nacional de Santiago del Estero y consistió principalmente en recorrer a cielo abierto los diferentes puntos de interés que se tuvieron en cuenta para realizar la prueba.

Para realizar la experimentación se crearon 18 (dieciocho) Objetos de Aprendizaje de prueba con tiempos de aprendizaje diferentes y 6 (seis) puntos de interés, es decir, 3 Objetos de Aprendizaje por cada punto de interés. El detalle se muestra en la tabla 5.3:

Tabla 5.3: Objetos de Prueba para experimentación

Nombre	Punto de Interés	Tiempo
ObjPrueba1	Área Becas	10
ObjPrueba2	Área Becas	15
ObjPrueba3	Área Becas	25
ObjPrueba4	Dpto. Alumnos	20
ObjPrueba5	Dpto. Alumnos	25
ObjPrueba6	Dpto. Alumnos	15
ObjPrueba7	Parainfo	10
ObjPrueba8	Parainfo	15
ObjPrueba9	Parainfo	5
ObjPrueba10	Lab. Alfa	15
ObjPrueba11	Lab. Alfa	25
ObjPrueba12	Lab. Alfa	20
ObjPrueba13	Lab. Beta	25
ObjPrueba14	Lab. Beta	20
ObjPrueba15	Lab. Beta	30
ObjPrueba16	Lab. Gamma	25
ObjPrueba17	Lab. Gamma	25
ObjPrueba18	Lab. Gamma	20

Todas las intervenciones de los estudiantes fueron registradas, es decir, con que Objetos de Aprendizaje interactuaron y con cuáles no. Luego, finalizadas las experiencias de aprendizaje por parte de los estudiantes se recabaron opiniones de manera informal sobre si la aplicación presentó algún error al momento de la interacción.

De manera formal, se diseñó un cuestionario simple compuesto por 10 preguntas. El mismo se detalla en la siguiente subsección.

5.5.1. CUESTIONARIO

El cuestionario empleado tiene por finalidad, por un lado, recopilar opiniones sobre la aplicación en relación con el diseño de la interfaz (interfaz atractiva, interfaz intuitiva y organización del contenido). Por otro lado, permite recolectar opiniones sobre el

funcionamiento y uso de la aplicación (facilidad de uso, utilidad del tiempo disponible, utilidad de la geolocalización, facilidad de autogestión y utilidad de la aplicación a futuro).

Se compone por respuestas prefijadas dentro de las cuales el encuestado puede marcar la que mejor se ajuste a su opinión. La escala a tener en cuenta para cada pregunta es la siguiente:

- 1: Completamente en desacuerdo,
- 2: Desacuerdo,
- 3: Tal vez,
- 4: De acuerdo,
- 5: Totalmente de acuerdo

Cuestionario para estudiantes

1. ¿Cree que la aplicación es fácil de usar?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

2. ¿Cree que la aplicación es intuitiva?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

3. ¿Cree que la aplicación tiene una interfaz de usuario atractiva y fluida?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

4. ¿La organización del contenido y/o componentes de la aplicación le parece correcto?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

5. ¿La sugerencia de Objetos de Aprendizaje, dándole prioridad a los “¿No realizados”, es de su agrado?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

6. ¿Cree que es de utilidad establecer su tiempo disponible para que la aplicación sugiera los objetos que son posibles realizar?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

7. ¿Es de su agrado la forma en que la aplicación móvil le permite ingresar su tiempo disponible?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

8. ¿Considera que la geolocalización es útil para sugerir puntos de interés?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

9. ¿Considera que la aplicación facilitaría la autogestión a los ingresantes a la Facultad? Es decir, ubicar los puntos de interés sobre los que debe realizar algún aprendizaje en el curso de ingreso y acceder por ellos mismos a los conocimientos.

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

10. Teniendo en cuenta, que la aplicación móvil permite a los estudiantes interactuar con Objetos de Aprendizaje independientemente del lugar en el que se encuentren ¿Considera que la aplicación podría ser útil no solo para el curso de ingreso sino también para el desarrollo de las carreras de la universidad, brindando un elemento de soporte a las materias que se dictan en la misma?

	1	2	3	4	5	
Completamente en desacuerdo						Totalmente de acuerdo

5.5.2. PROCESAMIENTO DE LAS RESPUESTAS AL CUESTIONARIO

Como se mencionó anteriormente, luego de la experiencia con la aplicación móvil, se les solicitó a los estudiantes que respondieran una breve encuesta individual. El procesamiento consistió en analizar las respuestas brindadas por los 5 estudiantes que probaron la aplicación.

El resultado fue el siguiente:

- El 100% de los estudiantes está “Totalmente de acuerdo” en que la aplicación es fácil de usar.
- El 60% de los estudiantes está “De acuerdo” que la aplicación es intuitiva mientras que el 40% está “Totalmente de acuerdo”.
- El 80% de los estudiantes está “Totalmente de acuerdo” en que la aplicación tiene una interfaz de usuario atractiva y fluida mientras que el 20% opina que “Tal vez” la interfaz de usuario es atractiva y fluida.

- El 60% de los estudiantes está “Totalmente de acuerdo” en que la organización de los contenidos y componentes es correcta mientras que el 40% opina que está “De acuerdo”.
- El 80% de los estudiantes está “Totalmente de acuerdo” en darle prioridad a los Objetos de Aprendizaje *No realizados*, mientras que el 20% restante opina que está “De acuerdo”.
- El 60% de los estudiantes piensa que “Tal vez” es de utilidad indicar el tiempo disponible para sugerir Objetos de Aprendizaje, un 20% opina que está “De acuerdo” y el restante 20% opina que está “Totalmente de acuerdo” en indicar su tiempo disponible para personalizar las sugerencias de Objetos de Aprendizaje.
- El 80% de los estudiantes está “Totalmente de acuerdo” en la forma gráfica que la aplicación le permite ingresar el tiempo disponible, mientras que el 20% restante está “De acuerdo”.
- El 100% de los estudiantes está “Totalmente de acuerdo” en que la geolocalización es útil para sugerir puntos de interés a los ingresantes.
- El 60% de los estudiantes está “Totalmente de acuerdo” en que la aplicación facilitaría la autogestión para los ingresantes, un 20% está “De acuerdo” mientras que el 20% restante opina que “Tal vez” facilitaría la autogestión.
- El 60% de los estudiantes está “De acuerdo” en que la aplicación podría ser útil no solo para el curso de ingreso sino también para el desarrollo de las carreras de la universidad, un 20% está “Totalmente de acuerdo” mientras que el 20% restante piensa que “Tal vez” sería de utilidad para el desarrollo de las carreras de la universidad.

5.6. ANÁLISIS DE RESULTADOS

En este capítulo se presentaron cuatro tipos de pruebas realizadas a la aplicación móvil: prueba manual, prueba de rendimiento, prueba de estrés y prueba de usabilidad.

La primera prueba, fue de suma utilidad a lo largo de todo el desarrollo ya que permitió corregir diferentes tipos de errores. Cada vez que se presentaba un incremento de software, se realizaba una prueba manual para verificar el correcto funcionamiento de todos los módulos existentes.

Seguidamente, la prueba de rendimiento sirvió para conocer el porcentaje de utilización tanto de la memoria RAM como del procesador. Con respecto a la prueba de estrés, desarrollada al final de todo el desarrollo de software, fue de gran utilidad para comprobar que la aplicación móvil no se detiene llevándola al máximo de su rendimiento.

La última prueba, de usabilidad y experimentación, fue de gran importancia para determinar cómo responde las conexiones del GPS, datos e internet, como así también, para evaluar si la aplicación es atractiva y fácil de usar. Los resultados arrojados pudieron comprobar que la aplicación es fácil de usar, es intuitiva y atractiva, la organización del contenido es correcta, que la geolocalización es de utilidad para sugerir puntos de interés y además que una aplicación similar sería útil no solo para el curso de ingreso sino para el desarrollo de las carreras de la universidad.

Los aspectos débiles a tener en cuenta son, por un lado, que la recepción de señales satelitales del GPS se ve parcialmente afectada en algunos puntos específicos debido a los árboles y la proximidad de un edificio con otro; los días nublados, son también motivo de interferencia con las señales GPS. Por otro lado, las conexiones de internet Wi-Fi presentaron cierta inestabilidad por lo cual, fue necesario utilizar conexiones a través de redes de datos durante las pruebas.

De lo expresado se concluye que, si bien la aplicación móvil presenta aspectos a mejorar, los procesos implementados funcionan correctamente.

CAPÍTULO 6

CONCLUSIONES

El aprendizaje ubicuo más que un fenómeno de la tecnología es una visión que desafía a revisar el entorno educativo para proponer nuevos escenarios educativos, ya que permiten poner a disposición de los estudiantes una gran diversidad de recursos educativos, crear nuevos y variados entornos de formación, personalizar el aprendizaje y posibilitar la realización de un conjunto de actividades formativas desde cualquier momento y lugar y, desde cualquier dispositivo.

En este trabajo se ha presentado el diseño y desarrollo de una aplicación ubicua para el apoyo a un taller del curso de ingreso de la FCEyT-UNSE. Se ha buscado con el diseño planteado, alcanzar las características esenciales del aprendizaje ubicuo como son la *permanencia* (el estudiante tiene disponibles los OA referidos a las temáticas del taller permanentemente), *accesibilidad* (el estudiante puede acceder a los OA en cualquier momento y lugar desde su dispositivo móvil), *inmediatez* (el estudiante puede concretar su aprendizaje cuando lo requiera) y *conciencia del contexto* (la aplicación le permite realizar un aprendizaje situado, en función de la ubicación en la que se encuentra el estudiante).

Por otra parte, en esta aplicación, el aprendizaje ubicuo se ve enriquecido al considerar aspectos de personalización, ya que se le recomiendan al estudiante OAs en función a su tiempo disponible para realizar la tarea y a su conocimiento previo. De esta forma, combinando información contextual con información del modelo de estudiante, se logra proporcionar un soporte más conveniente y efectivo al estudiante.

Teniendo en cuenta el dinamismo en el desarrollo de aplicaciones móviles, la metodología *Scrum* fue de gran ayuda para guiar todo el proceso. Si bien los requisitos no

fueron excesivamente cambiantes debido a una precisa descripción desde el comienzo, los cambios que surgieron fueron considerados en los Sprints posteriores.

A lo largo del proceso de desarrollo de la aplicación, se persiguió siempre el cumplimiento de los objetivos planteados al comienzo.

Creemos que este trabajo contribuirá a los estudiantes del curso de ingreso, aportando una aplicación móvil desarrollada nativamente para Android. Esto último, garantiza un buen rendimiento y fluidez en la interacción con el usuario y, gracias a las conexiones de internet 4G actuales es posible obtener rápidamente la información almacenada en el servidor.

Si bien es cierto que la aplicación móvil se mostró estable durante la experimentación realizada con estudiantes, está limitada a temáticas relacionadas al curso de ingreso. Además, es necesario que un docente experto en el área cree los Objetos de Aprendizaje utilizando un determinado *iDevice* de la herramienta *Exelearning*, de manera que la aplicación móvil sea capaz de capturar el resultado de las evaluaciones incluidas en los OAs.

Como trabajo a futuro podría concretarse una prueba de usabilidad en el entorno real del curso de ingreso; y en un futuro un poco más lejano, podría ampliarse las funcionalidades para que se incluya la aplicación no solamente en el curso de ingreso sino también en el desarrollo de las carreras de la universidad, permitiendo a los estudiantes interactuar con Objetos de Aprendizaje independientemente del tiempo y espacio en el que se encuentren.

Publicaciones

La realización de este trabajo final, desarrollado en el marco del proyecto “*Métodos y Técnicas para desarrollos de Aplicaciones Ubicuas*”, permitió la presentación en dos jornadas. El artículo bajo el nombre “*Diseño de una aplicación ubicua personalizada de apoyo a estudiantes del curso de ingreso*” fue presentado en la XII Jornadas de Ciencia y Tecnología de Facultades de Ingeniería del NOA, realizada en San Fernando del Valle de Catamarca los días 10 y 11 de agosto del 2017. La segunda presentación, en modalidad poster, fue presentada en la Primera Jornada de Vinculación con el Medio realizada en la Universidad Nacional de Santiago del Estero el día 3 de octubre del 2017.

REFERENCIAS

- Froschl, C. (2005). User Modeling and User Profiling in Adaptive E-learning Systems. *Master's Thesis. Graz University of Technology.*
- Albaladejo, X. (2017). *Proyectos Agiles*. Abgerufen am 28. Agosto 2017 von *Proyectos Agiles*: www.proyectosagiles.org
- Amaya Balaguera, Y. D. (2013). Metodologías ágiles en el desarrollo de aplicaciones para dispositivos móviles. Estado actual. *Revista de Tecnología, Journal Technology*, 12(2), 111-124.
- Armijos, A., Fiallos, A., Villavicencio, M., & Abad, C. (Diciembre 2015). Aplicación de Scrum en la construcción de un simulador de Redis. *Revista Tecnológica ESPOL*, 28(5), 346-360.
- Babativa, A., Briceño, P., Nieto, C., & Salazar, O. (2016). Desarrollo Ágil de una Aplicación para Dispositivos Móviles. Caso de Estudio: Taxímetro Móvil. *INGENIERÍA*, 21(3), 260-275.
- 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.*
- Burbules, N. (2012). El aprendizaje ubicuo y el futuro de la enseñanza. *Encounters/Encuentros/Rencontres on Education*, 13, 3-14.
- Burbules, N. C. (2014). El aprendizaje ubicuo: nuevos contextos, nuevos procesos. *Entramados: educación y sociedad*. 1(131134).
- Chen, C.-C., & Huang, T.-C. (2012). Learning in a u-Museum: Developing a context-aware ubiquitous learning environment. *Computers & Education*, 873–883.
- Chiappe Laverde, A., Segovia Cifuentes, Y., & Rincón Rodríguez, H. (2007). Toward an instructional design model based on learning objects. *Education Tech Research Dev*, 55:671–681.
- Chiu, P.-S., Kuo, Y.-H., Huang, Y.-M., & Chen, T.-S. (2008). A Meaningful Learning based u-Learning Evaluation Model. *Eighth IEEE International Conference on Advanced Learning Technologies*, 77-81.

- Cope, B., & Kalantzis, M. (2009). Ubiquitous Learning: An Agenda for Educational Transformation. *Proceedings of the 6th International*, 576-582.
- Cope, B., & Kalantzis, M. (2010). Ubiquitous learning.
- Cui, Y., & Bull, S. (2005). Context and learner modelling for the mobile foreign language learner. *System* 33, 353-367.
- El-Bishouty, M. M., Ogata, H., & Yano, Y. (2008). A Model of Personalized Collaborative Computer Support Ubiquitous Learning Environment. *Eighth IEEE International Conference on Advanced Learning Technologies*, 97-101.
- Escutia, R. F., & Torres, M. M. (2014). Desarrollo de aplicaciones móviles para la enseñanza en ciencias. *ReCIBE, Año 3 No.1*.
- Garg, A. (2017). *Linkedin*. Abgerufen am 28. Agosto 2017 von LinkedIn: <https://www.linkedin.com/pulse/scrum-events-ankur-garg>
- Google. (2017). *Android Studio*. Abgerufen am Agosto 2017 von Android Studio: <https://developer.android.com/studio/index.html>
- Graf, S., Yang, G., Liu, T.-C., & Kinshuk. (2009). Automatic, Global and Dynamic Student Modeling in a Ubiquitous Learning Environment. *18 Knowledge Management & E-Learning: An International Journal*, 1(1), 18-35.
- Honorable Consejo Directivo. (13 de Diciembre de 2016). Anexo - Resolución N° 206/16.
- Hwang, G.-J., Tsai, C.-C., & Yang, S. (2008). Criteria, Strategies and Research Issues of Context-Aware Ubiquitous Learning. *Educational Technology & Society*, 11(2), 81-91.
- Kniberg, H. (2007). *Scrum and XP from the Trenches*. USA.
- Krumm, J. (2009). *Ubiquitous Computing Fundamentals*. Redmon, Washington, U.S.A.: CRC Press.
- Lave, J. (1991). Situated Learning Legitimate Peripheral Participation.
- López Domínguez, E., Rodríguez Matla, Y., & Pomares Hernández, S. (2016). MobiLearn: Context-Aware Mobile Learning System. *IEEE LATIN AMERICA TRANSACTIONS, VOL. 14, NO. 2.*, 958-964.
- Loto, M. (2015). *DISEÑO DE UNA APLICACIÓN MÓVIL PERSONALIZADA DE APOYO AL APRENDIZAJE DE REDES DE COMPUTADORAS*. Universidad Nacional de Santiago del Estero, Facultad de Ciencias Exactas y Tecnologías, Santiago del Estero.
- Martín, S., Díaz, G., Carpio, J., Colmenar, A., Peire, J., & Castro, M. (2010). Aplicación móvil sensible al contexto cargadora de objetos de aprendizaje LOM. *Congreso Iberoamericano de Informatica Educativa*, 7.
- Möller, D. (2016). Ubiquitous Computing. In *Guide to Computing Fundamentals in Cyber-Physical Systems* (S. 185-234). Springer International Publishing Switzerland 2016.

- Ortigosa, A., Bravo, J., Carro, R., & Martin, E. (2010). ENTORNOS DE APRENDIZAJE MÓVILES ADAPTATIVOS Y EVALUACIÓN: COMOLE Y GESES. *RIED*, 13(2), 167-207.
- Ovalle, D., Salazar, O., & Duque, N. (2014). Modelo de Recomendación Personalizada en Cursos Virtuales basado en Computación Ubicua y Agentes Inteligentes. *Información Tecnológica Vol. 25(6)*, 131-142.
- Palacio, J. (2015). *Gestión de proyectos Scrum Manager*. Scrum Manager®.
- Paneva, D., & Zhelev, Y. (2007). MODELS, TECHNIQUES AND APPLICATIONS OF E-LEARNING PERSONALIZATION. *International Journal "Information Technologies and Knowledge"*, 1, 244-250.
- Park, W.-I., Park, J.-H., Kim, Y.-K., & Kang, J.-H. (2010). An Efficient Context-Aware Personalization Technique in Ubiquitous Environments. *ICUIMC'10, January 14–15, 2010*,.
- Paz Penagos, H. (2007). EL APRENDIZAJE SITUADO COMO UNA ALTERNATIVA EN LA FORMACIÓN DE COMPETENCIAS EN INGENIERÍA. *Revista Educación en Ingeniería*, 1-13.
- Posland, S. (2009). *Ubiquitous Computing: Basics and Vision*. Wiley Telecom.
- Romero, D., & Molina, J. M. (2010). Ambiente de Aprendizaje Móvil Basado en Micro-Aprendizaje. *IEEE-RITA Vol. 5*, 159-166.
- S. G., & Kinshuk. (2008). Adaptivity and Personalization in Ubiquitous Learning Systems. *A. Holzinger (Ed.): USAB 2008, LNCS 5298, © Springer-Verlag Berlin Heidelberg*, 331–338.
- Saadiah, Y., A. E., & A. K. (2010). The definition and characteristics of ubiquitous learning: A discussion. *International Journal of Education and Development using Information and Communication Technology Vol. 6, Issue 1*, 117-127.
- Scharff, C., & Verma, R. (2010). Scrum to Support Mobile Application Development Projects in a Just-in-time Learning Context.
- Schwaber, K., & Sutherland, J. (2016). *La Guía Definitiva de Scrum: Las Reglas del Juego*. www.scrum.org.
- Thompson Primo, T., Behr, A., & Vicari, R. (2013). A Semantic Web Approach to Recommend Learning Objects. 340-350.
- VázquezCano, E. (2015). The challenge of teacher training for using of mobile digital devices in higher education. *Perspectiva Educacional*. 54(1)(149162).
- Weiser, M. (1991). *The Computer for the 21st Century*. (265, 94104).
- Wenger, E. (2001). *Comunidades de Práctica*.

ANEXO I

MANUAL DE USUARIO DE LA APP MÓVIL

En esta sección se incluye un sencillo manual de usuario para el uso de la aplicación móvil destinada a estudiantes del curso de ingreso.

- Paso 1: Iniciar sesión con el usuario y contraseña provisto por un administrador (figura A.1).
- Paso 2: Indicar tiempo disponible a través de la *SeekBar* (figura A.2).
- Paso 3: Esperar que el GPS tome señal y muestre su precisión en metros. Apretar el botón *Actualizar Ubicación* y seleccionar el Punto de Interés sobre el que se desea realizar un aprendizaje (figura A.3).
- Paso 4: En el caso de seleccionar *Área Becas*, la aplicación listará los Objetos de Aprendizaje de acuerdo al tiempo disponible (ingresado al iniciar) y al conocimiento previo (OAs sobre los cuales se realizó un aprendizaje) (figura A.4).
- Paso 5: En el caso de seleccionar el objeto de prueba *ObjPrueba1*, la aplicación cargará el OA a través de un *WebView* nativo (figura A.5).
- Paso 6: Luego de haber interactuado con el contenido del OA y haber realizado la evaluación, es necesario apretar el botón *Enviar Respuestas*. En ese instante se determina el resultado de la evaluación y se actualiza el estado (figura A.6 y figura A.7). Al volver al listado de OAs, ya se encuentra actualizado el estado de la interacción reciente (figura A.8).



Figura A.1

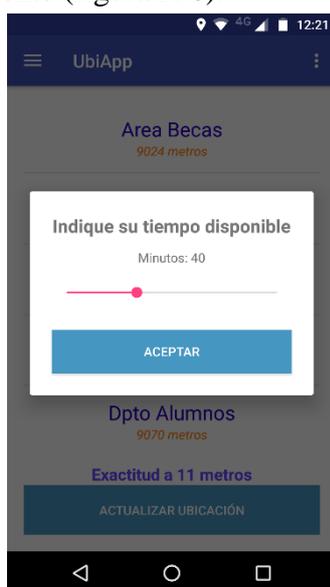


Figura A.2

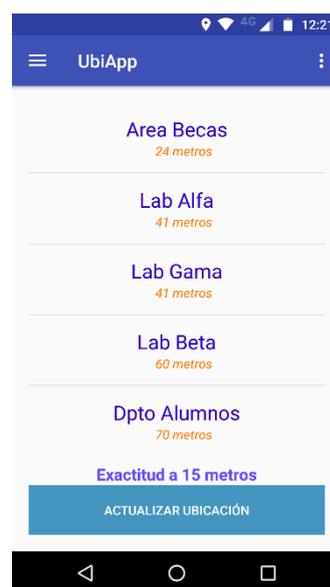


Figura A.3



Figura A.4

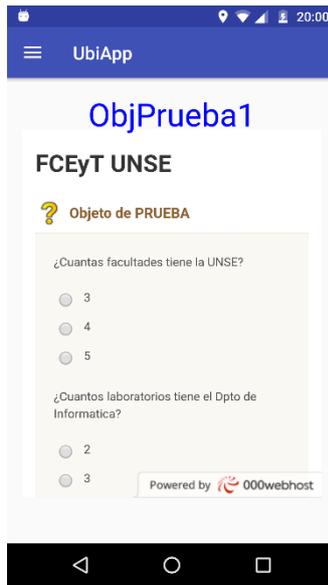


Figura A.5

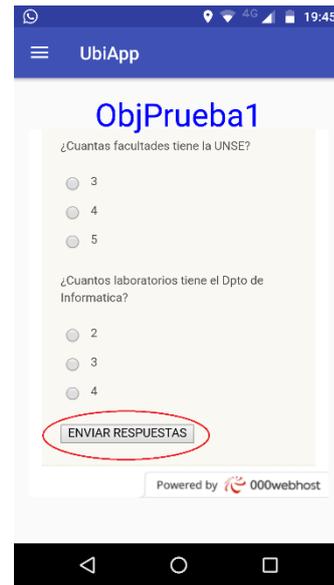


Figura A.6

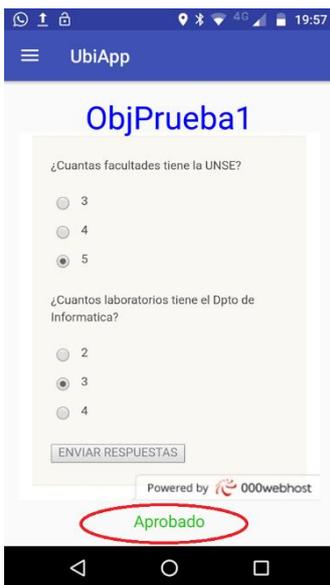


Figura A.7



Figura A.8

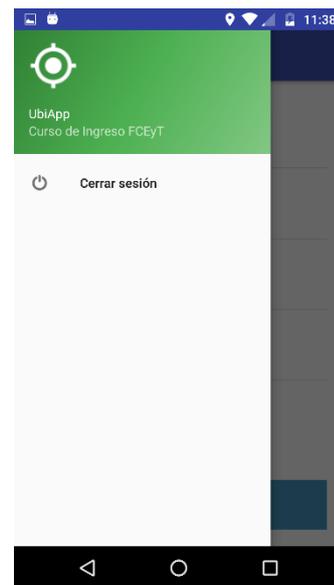


Figura A.9

En el caso de querer cerrar sesión, desplegar el menú lateral y luego apretar el botón *Cerrar Sesión* (figura A.9).