



**UNIVERSIDAD DEL BÍO-BÍO
FACULTAD DE CIENCIAS EMPRESARIALES
ESCUELA DE INGENIERÍA CIVIL INFORMÁTICA**

“Desarrollo de una versión móvil de la plataforma educativa Adecca UBB, para el sistema operativo Android”.

Por Rocío Sepúlveda Carriel

Memoria para optar al título de Ingeniero Civil en Informática

Profesora Guía : Maria Antonieta Soto Chico

Diciembre 2017
Chillán, Chile

Para Margarita y Alicia,
mis pilares fundamentales.

Agradecimientos

Quiero agradecer a las siguientes personas, por los distintos aportes que han realizado tanto en este proyecto, como en mis estudios de Ingeniería Civil en Informática, permitiendo la culminación de la etapa universitaria. Sin ellas, nada de esto sería posible.

Agradezco a mi familia por todo el apoyo, cariño y protección que me entregan. Agradezco especialmente a mi madre, por estar para mí incondicionalmente y entregarme sus valores. Agradezco a mi Ita, porque a pesar de que ya no esté, sigue dándome apoyo y motivación. Agradezco a mis amigos, por enseñarme las cosas que la Universidad no enseña, por escucharme y darme momentos inolvidables.

También agradezco a mis compañeros egresados, por los buenos consejos dados y el material entregado. Agradezco a cada uno de mis profesores, por ser un aporte significativo en mi educación; especialmente a aquellos profesores y profesoras que han dejado lecciones trascendentales. Agradezco a la profesora María Antonieta Soto, por darse el tiempo de revisar este documento en detalle y guiarme con su vasto conocimiento. Agradezco a Carlos San Juan, por desarrollar la mitad del trabajo en el servidor. Por último, agradezco al Departamento de Servicios Tecnológicos de la Universidad, por todas las oportunidades ofrecidas, por el buen recibimiento de cada uno de sus miembros y por ayudarme con este proyecto.

Resumen

Este proyecto se presenta para dar conformidad a los requisitos exigidos por la Universidad de Bío-Bío en el proceso de titulación para la carrera de Ingeniería Civil en Informática.

El proyecto titulado “Desarrollo de una versión móvil de la plataforma educacional Adecca UBB, para el sistema operativo Android”, busca agilizar los procesos de enseñanza e innovar las tecnologías de la Universidad del Bío-Bío, mediante el desarrollo de una versión de la plataforma educacional Adecca UBB, que permita tanto a docentes como estudiantes ingresar a la plataforma mediante un dispositivo móvil con sistema operativo Android. Debe ofrecer las principales funcionalidades que tiene la plataforma, tales como permitir el inicio de sesión de usuario, mostrar cursos y mostrar el contenido de estos. Aparte, debe permitir el acceso a distintas páginas web de la Universidad del Bío-Bío.

Para realizar el sistema descrito, se utiliza una metodología de desarrollo ágil con el objetivo de acercarse a la forma en que las empresas trabajan hoy en día. Scrum, específicamente, se ha escogido por ser una metodología adaptable, comprensible y accesible.

En términos de desarrollo, se utiliza información otorgada por el servidor de Adecca UBB mediante la arquitectura REST, permitiendo la virtualización de cursos y perfiles por cada usuario de la plataforma. Además, se muestran notificaciones ante distintos eventos que ocurran en los cursos de los usuarios, utilizando los servicios de Firebase Cloud Messaging. Por otra parte, el sistema se construye usando la arquitectura limpia, lo que implica el uso del patrón Modelo-Vista-Presentador y la librería Retrofit, para capturar los datos del servidor.

Finalmente, se obtiene un producto de software para dispositivos móviles con sistema Android, que contiene todas las funcionalidades solicitadas por el cliente, de modo que puede ser usada por todos los docentes y estudiantes de la Universidad del Bío-Bío. Así, se cumplió con todos los objetivos del proyecto, sin exceder el tiempo de duración acordado.

Abstract

This project is presented to accomplish the requirements demanded by the Universidad del Bío-Bío, in the titulation process for the career of Ingeniería Civil en Informática.

The project titled "Development of a mobile version of the educational platform Adecca UBB, for the Android operating system", seeks to streamline the teaching processes and innovate the technologies of the Universidad del Bío-Bío, by developing a version of the educational platform Adecca UBB, allowing teachers and students to access the platform through a mobile device with Android operating system. It must offer the main functionalities that the platform has, such as allowing user login, showing courses and displaying the content of these. Besides, it must allow the access to different web sites of the Universidad del Bío-Bío.

To implement the described system, an agile development methodology is used in order to approach the way in which companies work today. Scrum, specifically, has been chosen because it is an adaptable, comprehensible and accessible methodology.

In terms of development, information provided by the Adecca UBB server is used through the REST architecture, allowing the virtualization of courses and profiles for each user of the platform. In addition, notifications are shown to different events that occur in user courses, using the services of Firebase Cloud Messaging. On the other hand, the system is built using clean architecture, which implies the use of the Model-View-Presenter pattern and the Retrofit library, to capture the server data.

Finally, a software product is obtained for mobile devices with Android, which contains all the functionalities required by the costumer, so that it can be used by all the professors and students of the Universidad del Bío-Bío. Thus, all the objectives of the project were accomplished, without exceeding the agreed duration time.

Índice General

1	Introducción.....	14
2	Definición de la institución	16
2.1	Descripción de la institución	16
2.1.1	Antecedentes generales	16
2.1.2	Misión, Visión y Estructura organizativa	16
2.2	Descripción del área de estudio.....	18
2.2.1	Cargos relacionados con el proyecto.....	18
2.3	Descripción de la problemática	19
2.3.1	Descripción del sistema actual	19
2.3.2	Descripción del sistema a desarrollar	20
3	Definición proyecto	21
3.1	Objetivos del proyecto.....	21
3.1.1	Objetivo general	21
3.1.2	Objetivos específicos.....	21
3.2	Ambiente de Ingeniería de Software	22
3.2.1	Metodología de desarrollo.....	22
3.2.2	Técnicas y notaciones.....	22
3.3	Modelos, estándares y herramientas.....	24
3.3.1	Modelo de datos	24
3.3.2	Estándares.....	25
3.3.3	Herramientas de apoyo al desarrollo de software.....	25
3.4	Arquitectura de Software.....	26
3.4.1	MVP y Android	26
3.4.2	Arquitectura limpia.....	27
3.4.3	REST	29
4	Especificación de requisitos de Software	30
4.1	Propósito.....	30
4.2	Alcance	30
4.2.1	Objetivo del software	30
4.3	Visión general del producto.....	31
4.3.1	Producto en perspectiva.....	31
4.3.2	Interfaz de usuario	31
4.3.3	Funciones del producto.....	32
4.3.4	Limitaciones	33
4.4	Requisitos específicos	34

4.4.1	Requisitos funcionales.....	34
4.4.2	Requisitos no funcionales.....	36
5	Factibilidad.....	37
5.1	Factibilidad técnica.....	37
5.2	Factibilidad operativa.....	38
5.3	Factibilidad económica.....	39
5.3.1	Inversión.....	39
5.3.2	Costos.....	40
5.3.3	Beneficios tangibles.....	41
5.3.4	Beneficios intangibles.....	46
5.4	Conclusión del estudio de factibilidad.....	46
6	Análisis.....	47
6.1	Historias de usuario.....	47
6.2	Product Backlog.....	49
6.3	Esfuerzo requerido.....	50
6.4	Enfoque de la solución.....	53
6.5	Modelo de datos.....	54
7	Diseño.....	55
7.1	Diseño orientado al consumo de servicios.....	55
7.1.1	Estructura de Adecca UBB.....	55
7.1.2	Recursos y servicios.....	57
7.2	Diseño de arquitectura.....	60
7.2.1	MVP.....	60
7.2.2	REST.....	63
7.3	Sistema de notificaciones.....	66
7.4	Diagrama de clases.....	68
7.5	Funcionalidades adicionales.....	70
7.6	Diseño interfaz y navegación.....	71
7.6.1	Esquema de especificación de interfaz.....	71
7.6.2	Jerarquías de menú.....	72
8	Pruebas.....	76
8.1	Plan de pruebas.....	76
8.1.1	Cobertura del test.....	76
8.1.2	Metodología del test.....	76
8.2	Pruebas funcionales.....	78
8.3	Pruebas no funcionales.....	79

8.3.1 Usabilidad.....	79
8.3.2 Portabilidad.....	79
8.3.3 Seguridad.....	79
8.3.4 Eficiencia.....	79
8.4 Conclusiones de Pruebas	80
9 Conclusiones.....	82
Bibliografía.....	83
Anexo 1 : Factibilidad	84
1.1 Factibilidad técnica.....	84
1.2 Factibilidad económica.....	85
1.2.1 Pruebas de tiempo.....	85
1.3 Formato de encuesta aplicada a docentes	86
1.3.1 Resultados de encuesta aplicada a docentes	87
1.4 Formato de encuesta aplicada a estudiantes	91
1.4.1 Resultado de encuesta aplicada a estudiantes.....	93
1.5 Resumen de encuestas aplicadas	98
Anexo 2 : Diseño.....	99
2.1 Ejemplos de Adecca UBB.....	99
2.2 Especificación de servicios.....	100
Anexo 3 : Pruebas Unitarias	113
Anexo 4 : Resultados de Sprints.....	114
4.1 Sprint 1	114
4.1.1 Scrum Task Board	114
4.1.2 Cumplimiento de las historias de usuario.....	115
4.1.3 Primer periodo de pruebas.....	119
4.1.4 Segundo periodo de pruebas.....	124
4.2 Sprint 2	125
4.2.1 Scrum Task Board	125
4.2.2 Cumplimiento de las historias de usuario.....	127
4.2.3 Primer periodo de pruebas.....	135
4.2.4 Segundo periodo de pruebas.....	144
4.3 Sprint 3	146
4.3.1 Scrum Task Board	146
4.3.2 Cumplimiento de las historias de usuario.....	147
4.3.3 Primer periodo de pruebas.....	151
4.3.4 Segundo periodo de pruebas.....	155

4.4	Sprint 4	156
4.4.1	Scrum Task Board	156
4.4.2	Cumplimiento de las historias de usuario.....	158
4.4.3	Primer periodo de pruebas.....	166
4.4.4	Segundo periodo de pruebas.....	172
4.5	Sprint 5	173
4.5.1	Scrum Task Board	173
4.5.2	Cumplimiento de las historias de usuario.....	174
4.5.3	Primer periodo de pruebas.....	178
4.5.4	Segundo periodo de pruebas.....	182
4.6	Sprint 6	183
4.6.1	Scrum Task Board	183
4.6.2	Cumplimiento de las historias de usuario.....	184
4.6.3	Primer periodo de pruebas.....	189
4.6.4	Segundo periodo de pruebas.....	193
Anexo 5	: Pruebas de eficiencia	194

Índice Tablas

Tabla 1. Capas de software en arquitectura limpia.....	28
Tabla 2. Requisitos funcionales del sistema.....	35
Tabla 3. Requisitos de hardware y Software de Android Studio.....	37
Tabla 4. Resumen de la inversión.....	39
Tabla 5. Resumen de los costos.....	40
Tabla 6. Cálculo del tiempo de ahorro usando el sistema, en términos monetarios.....	42
Tabla 7. Cálculo del ahorro de hardware por estudiantes, en términos monetarios.....	43
Tabla 8. Cálculo del ahorro de hardware por docentes, en términos monetarios.....	44
Tabla 9. Total de ahorro por reducción del uso de hardware de la Universidad.....	44
Tabla 10. Total de ahorro.....	44
Tabla 11. Flujo de caja.....	45
Tabla 12. Historias de Usuario.....	48
Tabla 13. Product Backlog.....	49
Tabla 14. Puntos de historia.....	51
Tabla 15. Definición de Sprints.....	51
Tabla 16. Cálculo de la duración de un Sprint.....	52
Tabla 17. Recursos y Servicios.....	58
Tabla 18. Ejemplo de petición POST.....	64
Tabla 19. Ejemplo de petición GET.....	65
Tabla 20. Ejemplo de prueba de sistema.....	77
Tabla 21. Resumen de pruebas de sistema.....	78
Tabla 22. Ejemplo de características técnicas de un computador para desarrollo de software.....	84
Tabla 23. Prueba 1, tiempo que tarda un docente en responder la entrada de un foro.....	85
Tabla 24. Prueba 2, tiempo que tarda un estudiante en responder la entrada de un foro.....	85
Tabla 25. Servicio 1.....	100
Tabla 26. Servicio 2.....	102
Tabla 27. Servicio 3.....	103
Tabla 28. Servicio 4.....	104
Tabla 29. Servicio 5.....	106
Tabla 30. Servicio 6.....	107
Tabla 31. Servicio 7.....	108
Tabla 32. Servicio 8.....	109
Tabla 33. Servicio 9.....	110
Tabla 34. Servicio 10.....	111
Tabla 35. Servicio 11.....	112
Tabla 36. Pruebas de Sistema en HU_01.....	120
Tabla 37. Pruebas de Sistema en HU_02.....	121
Tabla 38. Pruebas de Sistema en HU_03.....	122
Tabla 39. Pruebas de Sistema en HU_04.....	123
Tabla 40. Pruebas de Sistema en HU_01.....	124
Tabla 41. Pruebas de Sistema en HU_03.....	124
Tabla 42. Pruebas de Sistema en HU_05.....	135
Tabla 43. Pruebas de Sistema en HU_06.....	136
Tabla 44. Pruebas de Sistema en HU_07.....	138
Tabla 45. Pruebas de Sistema en HU_08.....	139
Tabla 46. Pruebas de Sistema en HU_09.....	140
Tabla 47. Pruebas de Sistema en HU_10.....	141
Tabla 48. Pruebas de Sistema en HU_11.....	142

Tabla 49. Pruebas de Sistema en HU_12	143
Tabla 50. Pruebas de Sistema en HU_07	144
Tabla 51. Pruebas de Sistema en HU_10	145
Tabla 52. Pruebas de Sistema en HU_11	145
Tabla 53. Pruebas de Sistema en HU_13	151
Tabla 54. Pruebas de Sistema en HU_14	152
Tabla 55. Pruebas de Sistema en HU_15	153
Tabla 56. Pruebas de Sistema en HU_16	154
Tabla 57. Pruebas de Sistema en HU_15	155
Tabla 58. Pruebas de Sistema en HU_16	155
Tabla 59. Pruebas de Sistema en HU_17	166
Tabla 60. Pruebas de Sistema en HU_18	167
Tabla 61. Pruebas de Sistema en HU_19	168
Tabla 62. Pruebas de Sistema en HU_20	169
Tabla 63. Pruebas de Sistema en HU_21	170
Tabla 64. Pruebas de Sistema en HU_22	170
Tabla 65. Pruebas de Sistema en HU_23	170
Tabla 66. Pruebas de Sistema en HU_24	171
Tabla 67. Pruebas de Sistema en HU_17	172
Tabla 68. Pruebas de Sistema en HU_20	172
Tabla 69. Pruebas de Sistema en HU_25	178
Tabla 70. Pruebas de Sistema en HU_26	179
Tabla 71. Pruebas de Sistema en HU_27	180
Tabla 72. Pruebas de Sistema en HU_28	181
Tabla 73. Pruebas de Sistema en HU_25	182
Tabla 74. Pruebas de Sistema en HU_29	189
Tabla 75. Pruebas de Sistema en HU_30	190
Tabla 76. Pruebas de Sistema en HU_31	191
Tabla 77. Pruebas de Sistema en HU_32	192
Tabla 78. Pruebas de Sistema en HU_33	192
Tabla 79. Pruebas de Sistema en HU_31	193
Tabla 80. Pruebas de tiempo de respuesta para Q_Info.....	194
Tabla 81. Pruebas de tiempo de respuesta para A_Perfil	194

Índice Figuras

Figura 1. Organigrama de la Universidad del Bío-Bío.....	17
Figura 2. Ejemplo de Scrum Task Board [5].....	23
Figura 3. Diagrama de Entidad-Relación en la notación de Peter Chen [8].....	24
Figura 4. Modelo Vista Presentador.....	26
Figura 5. Diagrama de arquitectura limpia [9].....	27
Figura 6. Colores oficiales de la Universidad del Bío-Bío.....	31
Figura 7. Ícono de la plataforma Adecca UBB.....	32
Figura 8. Logotipo de la plataforma Adecca UBB.....	32
Figura 9. Modelo Entidad-Relación del sistema.....	54
Figura 10. Esquema representativo de la estructura de Adecca UBB.....	55
Figura 11. Ejemplo de carpetas y recursos didácticos.....	56
Figura 12. Secuencia de servicios.....	59
Figura 13. Ejemplo de capas en la aplicación.....	61
Figura 14. Proceso de FCM [14].....	66
Figura 15. Diagrama de clases del componente que implementa HU_01.....	68
Figura 16. Diagrama de clases del componente que implementa HU_02.....	69
Figura 17. Ejemplo de especificación de interfaz 1.....	71
Figura 18. Ejemplo de especificación de interfaz 2.....	71
Figura 19. Jerarquía de menú 1.....	72
Figura 20. Pantalla de inicio de usuario sin autenticar.....	73
Figura 21. Menú de usuario sin autenticar.....	73
Figura 22. Jerarquía de menú 2.....	74
Figura 23. Pantalla de inicio de usuario autenticado.....	75
Figura 24. Menú 1, de usuario autenticado. Parte 1.....	75
Figura 25. Menú 1, de usuario autenticado. Parte 2.....	75
Figura 26. Gráfico de respuestas a pregunta 1, encuesta a docentes.....	87
Figura 27. Gráfico de respuestas a pregunta 2, encuesta a docentes.....	87
Figura 28. Gráfico de respuestas a pregunta 3, encuesta a docentes.....	88
Figura 29. Gráfico de respuestas a pregunta 4, encuesta a docentes.....	88
Figura 30. Gráfico de respuestas a pregunta 5, encuesta a docentes.....	89
Figura 31. Gráfico de respuestas a pregunta 6, encuesta a docentes.....	89
Figura 32. Gráfico de respuestas a pregunta 7, encuesta a docentes.....	90
Figura 33. Gráfico de respuestas a pregunta 8, encuesta a docentes.....	90
Figura 34. Gráfico de respuestas a pregunta 1, encuesta a estudiantes.....	93
Figura 35. Gráfico de respuestas a pregunta 2, encuesta a estudiantes.....	93
Figura 36. Gráfico de respuestas a pregunta 3, encuesta a estudiantes.....	94
Figura 37. Gráfico de respuestas a pregunta 4, encuesta a estudiantes.....	94
Figura 38. Gráfico de respuestas a pregunta 5, encuesta a estudiantes.....	95
Figura 39. Gráfico de respuestas a pregunta 6, encuesta a estudiantes.....	95
Figura 40. Gráfico de respuestas a pregunta 7, encuesta a estudiantes.....	96
Figura 41. Gráfico de respuestas a pregunta 8, encuesta a estudiantes.....	96
Figura 42. Gráfico de respuestas a pregunta 9, encuesta a estudiantes.....	97
Figura 43. Gráfico resumen.....	98
Figura 44. Gráfico resumen.....	98
Figura 45. Ejemplo de pantalla principal de Adecca UBB.....	99
Figura 46. Ejemplo de pantalla principal de un curso de Adecca UBB.....	99
Figura 47. Scrum Task Board, Sprint 1.....	114
Figura 48. Captura de pantalla HU_01.....	115

Figura 49. Captura de pantalla HU_01	115
Figura 50. Captura de pantalla HU_02	116
Figura 51. Captura de pantalla HU_03	117
Figura 52. Captura de pantalla HU_04	118
Figura 53. Scrum Task Board, Sprint 2	126
Figura 54. Captura de pantalla HU_05	127
Figura 55. Captura de pantalla HU_06	128
Figura 56. Captura de pantalla HU_07	129
Figura 57. Captura de pantalla HU_08	130
Figura 58. Captura de pantalla HU_09	131
Figura 59. Captura de pantalla HU_10	132
Figura 60. Captura de pantalla HU_11	133
Figura 61. Captura de pantalla HU_12	134
Figura 62. Scrum Task Board, Sprint 3	146
Figura 63. Captura de pantalla HU_13	147
Figura 64. Captura de pantalla HU_14	148
Figura 65. Captura de pantalla HU_15	149
Figura 66. Captura de pantalla HU_16	150
Figura 67. Scrum Task Board, Sprint 4	157
Figura 68. Captura de pantalla HU_17	158
Figura 69. Captura de pantalla HU_18	159
Figura 70. Captura de pantalla HU_19	160
Figura 71. Captura de pantalla HU_20	161
Figura 72. Captura de pantalla HU_21	162
Figura 73. Captura de pantalla HU_22	163
Figura 74. Captura de pantalla HU_23	164
Figura 75. Captura de pantalla HU_24	165
Figura 76. Scrum Task Board, Sprint 5	173
Figura 77. Captura de pantalla HU_25	174
Figura 78. Captura de pantalla HU_26	175
Figura 79. Captura de pantalla HU_27	176
Figura 80. Captura de pantalla HU_28	177
Figura 81. Scrum Task Board, Sprint 6	183
Figura 82. Captura de pantalla HU_29	184
Figura 83. Captura de pantalla HU_30	185
Figura 84. Captura de pantalla HU_31	186
Figura 85. Captura de pantalla HU_32	187
Figura 86. Captura de pantalla HU_33	188

1 Introducción

En la actualidad, los dispositivos móviles se han convertido en una necesidad para muchas personas en todo el mundo, ya que agilizan la comunicación, mejoran la productividad, informan a los usuarios de los distintos sucesos que ocurren a su alrededor, entre muchos otros beneficios. A esto se le suma lo fácil que es adquirir un dispositivo móvil, ya que la oferta creciente de los últimos años ha hecho que su precio disminuya considerablemente.

Del mismo modo, los beneficios que otorgan los dispositivos móviles están siendo usados en el ámbito de la educación, agilizando los procesos que esta involucra. Hoy en día, los estudiantes pueden acceder a información de sus cursos, comunicarse con sus profesores o compañeros de clase, realizar trámites, contestar evaluaciones, entregar tareas y gestionar toda la información que utilizan en sus estudios. Desde el punto de vista de la docencia, los profesores pueden gestionar cursos, horarios, agendar reuniones o actividades, calificar estudiantes, entregar material, entre muchas acciones.

Por este motivo, la Universidad del Bío-Bío quiere entregar beneficios a la comunidad universitaria ofreciendo los servicios de la plataforma educativa Adecca UBB, a través de dispositivos móviles. Esto le permite agilizar y mejorar los procesos de enseñanza, además de proyectar una imagen que llame la atención del público estudiantil.

Adecca UBB es una Plataforma Educativa que tiene por finalidad prestar servicios de apoyo a la docencia de Pregrado y Postgrado, favoreciendo la utilización de estrategias activas de enseñanza para alcanzar ambientes de trabajo colaborativos. Permite virtualizar los cursos de un usuario, de modo que incluyan contenidos y funcionalidades útiles, tanto para estudiantes como docentes. Entre los contenidos se encuentran archivos, tareas, trabajos grupales, foros, etc., a los que los usuarios pueden acceder y participar en conjunto con otros miembros del curso.

Este proyecto busca desarrollar una versión reducida de la plataforma educacional Adecca UBB para dispositivos móviles con sistema operativo Android, que incorpore parte de las funcionalidades que la plataforma pone a disposición de estudiantes y docentes y que, además, permita acceder a los sitios web e información general de la Universidad del Bío-Bío.

Para cumplir con lo anterior, se ha escogido una serie de metodologías y técnicas que permiten desarrollar un producto de calidad. Entre ellas se encuentra la metodología ágil de desarrollo Scrum, conocida por el éxito que suele tener entre las empresas que la usan. Por otra parte, se hace uso de buenas prácticas como la arquitectura limpia, que permite aumentar la calidad del producto y facilitar el trabajo de los desarrolladores ante nuevas funciones o cambios en el sistema.

En este documento se encuentra descrito el proceso realizado para obtener el producto que requiere la Universidad del Bío-Bío. El siguiente capítulo presenta el contexto en el que se desarrolla este proyecto, el problema que se pretende resolver y de qué forma se debe solucionar. Posteriormente, el tercer capítulo estipula los objetivos, metodología y herramientas de desarrollo, notaciones y estándares de documentación, y las arquitecturas necesarias para construir el proyecto. El cuarto capítulo especifica los requerimientos de software definidos junto

al cliente, usando para ello el estándar ISO/IEC/IEEE 29148:2011. Luego, el quinto capítulo corresponde al estudio de factibilidad que determina qué tan realizable es el sistema, técnica, operativa y económicamente. El sexto capítulo analiza el sistema a desarrollar, usando las herramientas y técnicas de Scrum; se incluye el análisis que determina el tipo de software a construir, además del modelo de datos que representa a la plataforma. El séptimo capítulo, por su parte, corresponde al diseño del software donde se define el sistema como un consumidor de servicios, incluyendo la especificación del diseño de la arquitectura y el diseño de la interfaz y navegación. Posteriormente, el octavo capítulo corresponde a las pruebas realizadas en el sistema, que permiten verificar la calidad del mismo. Finalmente, el noveno capítulo presenta las conclusiones de este proyecto. Además, se incluyen las fuentes bibliográficas usadas y diversos anexos, que detallan información crucial sobre la cual se construyó este documento.

2 Definición de la institución

En este capítulo, se presenta información general de la institución para la cual se realiza este proyecto, incluyendo datos del área que plantea la problemática, representantes, el problema que enfrentan y qué necesitan para solucionarlo.

2.1 Descripción de la institución

A continuación, se presentan datos de la Universidad del Bío-Bío, para contextualizar el problema que se define más adelante.

2.1.1 Antecedentes generales

Nombre	:	Universidad del Bío-Bío
RUT	:	60.911.006-6
Representante legal	:	Rector, Dr. Héctor Guillermo Gaete Féres
Ubicación	:	Ciudades de Concepción y Chillán, zona centro sur de Chile.
Sitio Web	:	www.ubiobio.cl
Servicio	:	Educación universitaria

2.1.2 Misión, Visión y Estructura organizativa

Misión

“La Universidad del Bío-Bío, a partir de su naturaleza pública, responsable socialmente y estatal, tiene por misión, desde la Región del Biobío, aportar a la sociedad con la formación de personas integrales, a través de una Educación Superior de excelencia. Comprometida con los desafíos de la región y del país, contribuye a la movilidad e integración social por medio de; la generación y transferencia de conocimiento avanzado, mediante la docencia de pregrado y postgrado de calidad, la investigación fundamental, aplicada y de desarrollo, la vinculación bidireccional con el medio, la formación continua y la extensión. Asimismo, impulsa el emprendimiento y la innovación, el fortalecimiento de la internacionalización y el desarrollo sustentable de sus actividades, basada en una cultura participativa centrada en el respeto a las personas.” [1].

Visión

“Ser reconocida a nivel nacional e internacional como una Universidad pública, responsable socialmente y regional que, comprometida con su rol estatal, desde la Región del Bio-Bío, forma personas integrales de excelencia y aporta a través de su quehacer al desarrollo sustentable de la región y el país.” [1].

Estructura organizativa

La estructura organizacional de la Universidad del Bío-Bío se muestra a continuación, en la Figura 1.

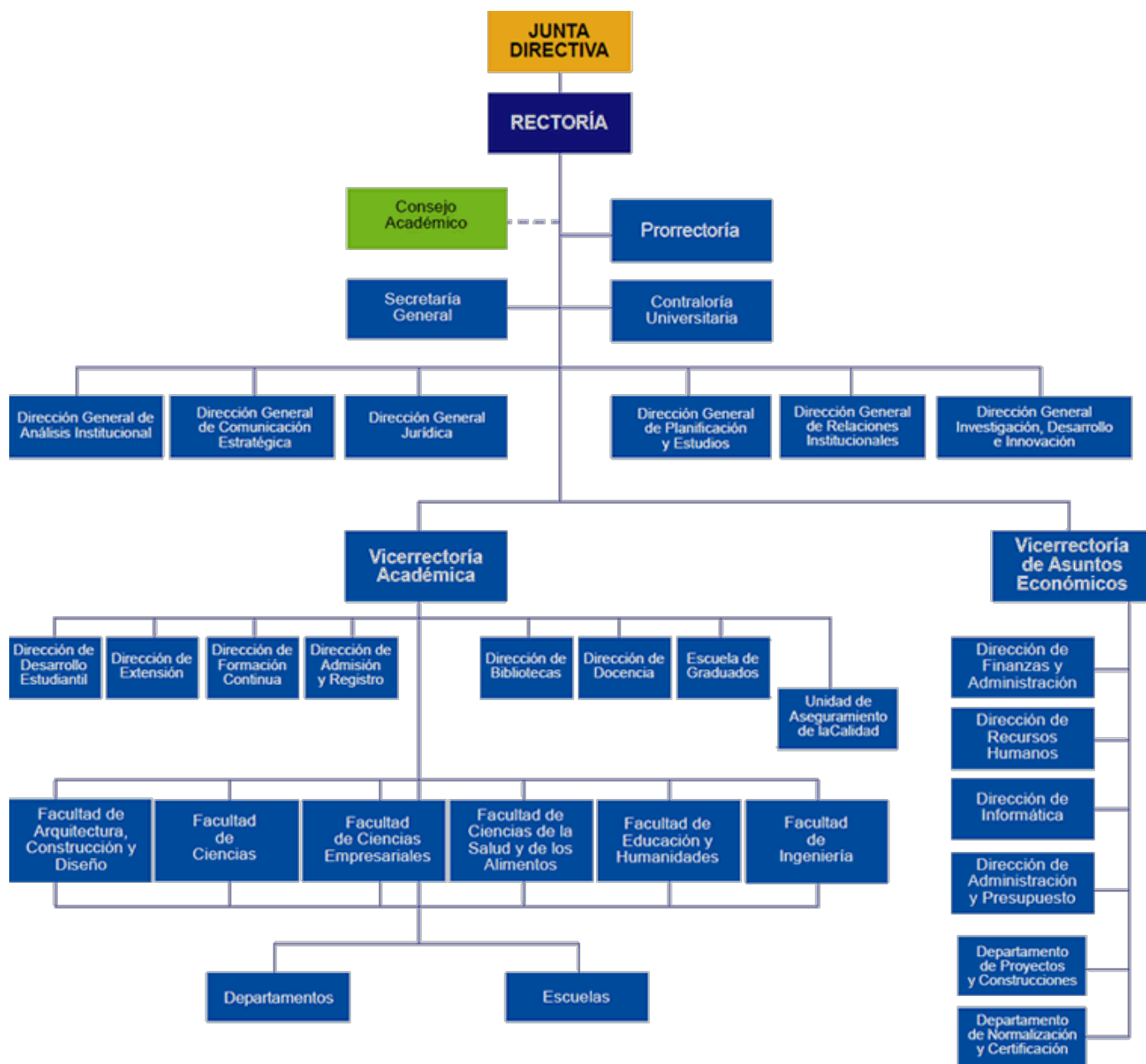


Figura 1. Organigrama de la Universidad del Bío-Bío

2.2 Descripción del área de estudio

Departamento de Servicios Tecnológicos

“El Departamento de Servicios Tecnológicos, tiene como finalidad administrar los recursos de redes, datos, software, hardware y telefónicos de la Sede Chillán. Para cumplir con su misión debe coordinarse con la Dirección de Informática, para prestar los servicios que demanden los usuarios, velando por el cumplimiento de las políticas informáticas de la Institución y apoyando el accionar de ambas Vicerreorías y las distintas unidades de la Universidad.” [2].

El proyecto académico de Adecca UBB es administrado por la Dirección de Informática de la Universidad del Bío-Bío, pero mantiene un Equipo Técnico-Pedagógico, responsable del desarrollo y de la implementación de estrategias activas de enseñanza [3]. Ha tenido la participación de estudiantes en práctica y memoristas de la Carrera de Ingeniería Civil en Informática de la Universidad del Bío-Bío, fundamentalmente en la revisión de las funcionalidades y elaboración de prototipos de estrategias de enseñanza [3].

2.2.1 Cargos relacionados con el proyecto

A continuación se indican los responsables de los cargos relacionados con el proyecto:

1. Coordinación Académica: Fernando Toledo Montiel
2. Coordinación Pedagógica: Raquel Aburto Godoy
3. Coordinación Técnica: Natanael Guerrero Carrasco
4. Encargado Desarrollo: Carlos San Juan Contreras

2.3 Descripción de la problemática

En esta sección, se contextualiza y describe el problema que aborda este proyecto, además del resultado que espera obtener el Departamento de Servicios Tecnológicos de la Universidad.

2.3.1 Descripción del sistema actual

Adecca es una Plataforma Educativa que tiene por finalidad prestar servicios de apoyo a la docencia de Pregrado y Postgrado, favoreciendo la utilización de estrategias activas de enseñanza para alcanzar ambientes de trabajo colaborativos.

Adecca UBB permite virtualizar los cursos de un usuario, de modo que incluyan contenidos y funcionalidades útiles, tanto para estudiantes como docentes. Entre los contenidos se encuentran archivos, tareas, trabajos grupales, foros, etc., a los que los usuarios pueden acceder y participar en conjunto con otros miembros del curso.

El origen de Adecca se remonta al año 1997, cuando el Departamento de Ciencias Básicas y el Proyecto Enlaces-UBB dirigidos por el académico Fernando Toledo Montiel, en función de sus requerimientos de servicio docente y de capacitación, determinan utilizar una Plataforma Virtual de apoyo.

Luego de investigar y comparar varias posibles herramientas de la época, se opta por la plataforma de fuente abierta Learnloop, dada su filosofía modular, intuitiva y bajo costo técnico para los usuarios, entre otras razones.

Al mismo tiempo, el Equipo Técnico y Pedagógico del Departamento de Ciencias Básicas interviene algunas funcionalidades e incorpora otras, determinando que la Plataforma se denomine EV&C UBB. En el año 2003 queda bajo el nombre de Adecca-UBB, “Administrador de Cursos de Código Abierto” y es utilizada masivamente por las Universidades adscritas al proyecto.

Durante el año 2012, se decide construir una nueva versión de Adecca conservando las características modulares y filosofía de trabajo que la habían hecho popular en la Universidad del Bío-Bío y otras instituciones. Técnicamente se actualiza la versión del lenguaje y bases de datos que la soportan, incorporando el Framework 'Codeigniter', permitiendo el fácil desarrollo de nuevos módulos y plugins.

Actualmente, la plataforma está disponible solo para navegadores web mediante la dirección adecca.ubiobio.cl. Cuando un usuario inicia sesión, esta caduca después de unos minutos, lo cual implica que el usuario debe ocupar tiempo en iniciar sesión constantemente. Por otra parte, si un usuario desea acceder a una función de la plataforma, debe acceder a través de un computador o a través del navegador del dispositivo móvil; la primera situación toma demasiado tiempo al usuario si no se encuentra cerca de un computador, la segunda situación toma menos tiempo, pero resulta incómoda ya que la web de Adecca UBB no es responsiva, dificultando la navegación en la página.

Por otra parte, la Universidad del Bío-Bío no posee aplicaciones móviles orientadas a los estudiantes y docentes, a diferencia de otras universidades estatales o de la zona. Esto implica una desventaja en la imagen que proyecta la Universidad, desde el punto de vista de la innovación.

2.3.2 Descripción del sistema a desarrollar

Se necesita un sistema que permita a los usuarios de Adecca UBB ingresar a una aplicación móvil de la plataforma para dispositivos Android. Esta aplicación debe contener varias de las funcionalidades de la plataforma, tales como permitir el inicio de sesión, mostrar cursos, mostrar el contenido de los cursos, permitir participación en el foro, permitir la descarga de archivos, recibir notificaciones, ver el perfil de usuario y personalizarlo. También debe permitir un acceso rápido a páginas y servicios de la Universidad, como la web de Intranet, biblioteca, Desarrollo Estudiantil y calendario académico. La aplicación debe operar con los mismos datos de la versión web, ya que acceden a los mismos recursos que dispone el servidor de Adecca UBB. Lo anterior implica que la aplicación móvil no necesita una base de datos propia.

3 Definición proyecto

En este capítulo se presenta la definición del proyecto, que permite guiar el desarrollo y la documentación del mismo.

3.1 Objetivos del proyecto

A continuación, se presenta el objetivo general y los objetivos específicos del proyecto, que deben ser cumplidos al término de este.

3.1.1 Objetivo general

Desarrollar una versión reducida de la plataforma educativa Adecca de la Universidad del Bío-Bío para dispositivos móviles con sistema operativo Android, que incorpore una parte de las funcionalidades que Adecca UBB pone a disposición de estudiantes y docentes y que, además, incorpore datos útiles de la Universidad del Bío-Bío, permitiendo un acceso rápido a la información.

3.1.2 Objetivos específicos

Los objetivos específicos corresponden a:

1. Permitir el acceso a Adecca UBB mediante dispositivos móviles Android.
2. Permitir la interacción con cursos de Adecca UBB.
3. Permitir la interacción con información personal de cada usuario de Adecca UBB.
4. Permitir el acceso a información general de la Universidad del Bío-Bío.

3.2 Ambiente de Ingeniería de Software

A continuación, se define la metodología de desarrollo que se sigue en este proyecto, además de las herramientas y notaciones que se usan en la documentación y desarrollo del mismo.

3.2.1 Metodología de desarrollo

En la actualidad, las metodologías ágiles se han vuelto muy populares entre las empresas debido al éxito que se ha conseguido en los proyectos que las han utilizado. Por definición, las metodologías ágiles son aquellas que permiten adaptar la forma de trabajo a las condiciones del proyecto, consiguiendo flexibilidad e inmediatez en la respuesta para amoldar el proyecto y su desarrollo a las circunstancias específicas del entorno [4]. Se ha escogido para este proyecto adoptar una metodología ágil, con el objetivo de acercarse a la forma en que las empresas trabajan hoy en día, y finalizar el proyecto de manera exitosa.

Scrum

Dentro del mundo de las metodologías ágiles se opta por Scrum, que se usa para el desarrollo de productos y servicios innovadores. Se escogió esta metodología principalmente por su adaptabilidad a los cambios en los requisitos, además de su fácil comprensión y aplicación, mediante distintas herramientas que son de fácil acceso.

3.2.2 Técnicas y notaciones

A continuación, se presentan técnicas y notaciones que son altamente usadas por quienes trabajan con Scrum, y que se utilizan en este proyecto.

Roles de Scrum

Dentro de esta metodología, existen distintos roles dedicados a diferentes tareas. Uno de estos es el Product Owner, quien solicita y aprueba las funcionalidades que desea en su producto, y las ordena según su prioridad. Otro rol es el de Scrum Master, que actúa como entrenador liderando el proceso y ayudando con la organización. Finalmente, está el equipo de desarrollo, quienes son los encargados de diseñar, construir y probar las funcionalidades que solicitó el Product Owner.

En este proyecto el rol de Product Owner corresponde al Departamento de Servicios Tecnológicos de la Universidad del Bío-Bío, el rol de Scrum Master corresponde al encargado de coordinación técnica de DST, y el equipo de desarrollo corresponde a la autora de este documento.

Product Backlog

Es una lista de trabajo ordenada según la prioridad desde el punto de vista del Product Owner. En este proyecto se utiliza como parte de Scrum, para planificar el desarrollo.

Sprints

En Scrum, un Sprint es una iteración o ciclo que suele durar entre una a cuatro semanas. Idealmente, cada Sprint debería tener la misma duración, y fechas de inicio y término fijadas. El trabajo completado en cada Sprint debe crear algo de valor para el cliente. Dado que trabaja en conjunto con el Product Backlog también es necesario utilizarla, como parte de la metodología y planificación del desarrollo.

Historias de Usuario

Para concordar con la metodología de desarrollo, se deben escribir los requisitos como historias de usuario. Los requisitos facilitan una comprensión compartida de lo que se necesita construir, y permiten comunicar efectiva y eficazmente lo que el cliente solicita a la persona o equipo que creará su solicitud. Las historias de usuarios se elaboran de una manera que hace que los requisitos sean comprensibles para los empresarios y técnicos.

Un formato de plantilla común para escribir historias de usuarios consiste en especificar una clase de usuario, qué quiere lograr (la meta), y por qué quiere lograr esa meta (el beneficio): Como <rol de usuario>, quiero <objetivo>, para <beneficio> [5].

Scrum Task Board

También llamada tabla de tareas de Scrum, se utiliza para visualizar el Backlog del proyecto. Como se aprecia en la Figura 2, las filas de la tabla corresponden a una historia de usuario, y dentro de ellas se encuentran las tareas que completan la funcionalidad solicitada en la historia de usuario, divididas según su estado en el proceso: “Por hacer” (To Do), “en proceso” (In Process), “por verificar” (To Verify) y “hecho” (Done). En este proyecto se utilizará una versión acotada de Task Board, para trabajar visualmente con la planificación de cada Sprint, que incluirá las columnas de historia de usuario, por hacer, en proceso y hecho o finalizado.

Story	To Do	In Process	To Verify	Done
As a user, I... 8 points	Code the... 9 Code the... 2 Test the... 8	Test the... 8 Code the... 4 Test the... 8	Test the... 6	Code the... 8 Test the... 8 Test the... 8 Test the... 6
As a user, I... 5 points	Code the... 8 Code the... 4	Test the... 8 Code the... 6 Code the... 8		Test the... 8 Test the... 6

Figura 2. Ejemplo de Scrum Task Board [5]

3.3 Modelos, estándares y herramientas

En este punto, se muestran los modelos y estándares que sirven para analizar el proyecto, además de las herramientas necesarias para desarrollarlo y documentarlo.

3.3.1 Modelo de datos

Modelo entidad relación

El Modelo de Entidad Relación es un modelo de datos, basado en una percepción del mundo real que consiste en un conjunto de objetos básicos llamados entidades y las relaciones entre estos objetos, implementándose en forma gráfica a través del Diagrama Entidad Relación [6].

Un diagrama entidad-relación (ER) es un tipo de diagrama de flujo ideado por Peter Chen, que ilustra cómo las "entidades", como personas, objetos o conceptos, se relacionan entre sí dentro de un sistema. Se usan a menudo para diseñar o depurar bases de datos relacionales en la ingeniería de software y otros campos similares [7].

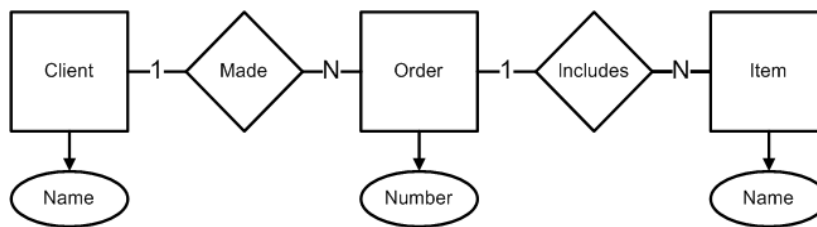


Figura 3. Diagrama de Entidad-Relación en la notación de Peter Chen [8]

En detalle [7]:

- Entidad: Algo que se puede definir, como una persona, objeto, concepto u evento, que puede tener datos almacenados acerca de este. Las entidades son el equivalente a los sustantivos gramaticales y se representan a través de un cuadrado o rectángulo.
- Atributo: Una propiedad o característica de una entidad. A menudo se muestra como un óvalo o círculo.
- Relación: Cómo las entidades interactúan o se asocian entre sí. Las relaciones son el equivalente a los verbos o asociaciones. Una relación puede ser definida según el número de entidades asociadas con ella, conocido como el grado. Se representa a través de un rombo.

3.3.2 Estándares

ISO/IEC/IEEE 29148:2011. Systems and software engineering — Life cycle processes — Requirements engineering. Este estándar internacional proporciona un tratamiento unificado de los procesos y productos que intervienen en la ingeniería de requisitos, en todo el ciclo de vida de los sistemas. Define la construcción de un buen requisito, proporciona atributos y características de ellos, y discute la aplicación iterativa y recursiva de los requisitos a lo largo del ciclo de vida. En detalle, establece la construcción y lenguaje que se debe usar, además de los puntos a tratar que debe incluir la documentación, tales como propósito, alcance, visión general, y requisitos específicos.

3.3.3 Herramientas de apoyo al desarrollo de software

A continuación, se indican las herramientas que tienen relación con el desarrollo de este proyecto:

- **Android Studio:** Android Studio es el entorno de desarrollo integrado (IDE) indispensable para crear y mantener una aplicación móvil. El uso de este software es gratuito y está disponible para el público en general mediante su descarga desde la web.
- **Java Development Kit:** Software que provee herramientas de desarrollo para la creación de programas en Java.
- **PdaNet+:** Comparte el acceso a Internet del dispositivo Android con el computador. En este proyecto se usa para emular la aplicación en un dispositivo físico.
- **RealtimeBoard:** Es una aplicación en línea que ofrece un espacio de trabajo altamente personalizable. Es usado como Scrum Task Board, a través de una plantilla dispuesta para ello.
- **ERDPlus:** Es una herramienta de modelado de base de datos que permite crear diagramas de entidad-relación, esquemas relacionales, esquemas de estrella y sentencias SQL DDL.
- **Creately:** Herramienta en línea para modelado de objetos, como diagramas UML.
- **JSON Formatter:** Herramienta en línea que permite formatear elementos JSON, mostrándolos con mayor claridad, además de cargar sentencias JSON desde urls, entre otras funciones.
- **Jschema2pojo:** Herramienta en línea que permite transformar elementos JSON a objetos en Java, ajustándose al formato GSON.
- **Firebase Cloud Messaging:** También llamado FCM, es un servicio de Google cuya función popular es el envío de notificaciones push desde una aplicación servidor hacia una aplicación cliente, por medio del servicio de mensajería del servidor de Firebase.
- **JUnit:** Es un framework de pruebas que usa anotaciones para identificar métodos que especifican una prueba.
- **RxJava:** Es una implementación de Java VM de ReactiveX (Reactive Extensions): una biblioteca para componer programas asincrónicos y basados en eventos, mediante el uso de secuencias observables.

3.4 Arquitectura de Software

A continuación, se presenta la arquitectura que usa el sistema en su construcción.

3.4.1 MVP y Android

El Modelo-Vista-Presentador (MVP) es un patrón derivado del Modelo-Vista-Controlador. MVP rompe el controlador para que el acoplamiento natural de la vista pueda ocurrir sin vincularlo al resto de las responsabilidades del controlador; cuando se usa MVC o no se usa algún patrón en particular, pueden existir clases de tipo Actividades y Fragmentos, que suelen controlar tanto el modelo como las vistas, provocando que el código sea muy extenso y complejo ante modificaciones y pruebas.

Usando este patrón se facilita el uso de pruebas de caja blanca, y en caso de que dos o más componentes posean la misma lógica, es posible reutilizar los recursos del sistema. Se debe considerar que este no es un patrón de arquitectura de aplicaciones, pero se usa como tal.

Los componentes de esta arquitectura se indican a continuación:

- **Modelo:** Es una interfaz responsable de manejar los datos. Sus atribuciones incluyen el uso de APIs, manejo de bases de datos, servicios web, etc. Además, se comunica con otros componentes que cumplen funciones similares, como los “interactores”, definidos más adelante.
- **Vista:** Sólo es responsable de presentar los datos de la forma en que solicita el presentador. La vista puede ser implementada a través de actividades y fragmentos.
- **Presentador:** Esto es esencialmente el controlador de MVC, excepto que no está en absoluto vinculado a la vista, sólo a una interfaz que comunica al presentador con ella. Por ende, actúa como coordinador entre la vista y el modelo. Esto aborda los problemas de factibilidad de pruebas, así como la modularidad/flexibilidad que se tiene con MVC. Para mejorar las pruebas en la aplicación, es necesario que el presentador sea independiente de cualquier elemento de Android, es decir, que sólo use dependencias de Java.

A continuación, la Figura 4 muestra un esquema de cómo funciona MVP en un software. Se puede apreciar que toda la información pasa por el presentador; la vista y el modelo no tienen comunicación directa.

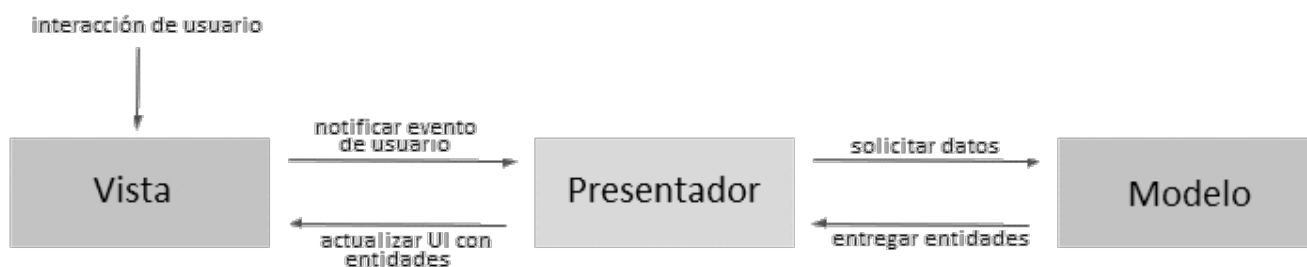


Figura 4. Modelo Vista Presentador

3.4.2 Arquitectura limpia

La arquitectura limpia consiste en escribir código de alta calidad, ya que no solo es cuestión de cumplir los requisitos del cliente; el código debe ser robusto, mantenible, flexible y fácil de probar ante los cambios y el crecimiento del software.

El propósito de esta arquitectura es que un sistema sea independiente de frameworks, fácil de probar, independiente de la UI, independiente de la base de datos, e independiente de cualquier agencia externa.

La Figura 5 integra variadas arquitecturas modernas en una idea individual y factible, que cumpla con los propósitos anteriores. Algunas de las capas que involucra la arquitectura limpia, en general, son:

- Las entidades (entities), son los objetos de negocio de la aplicación.
- Los casos de uso (use cases), manejan el flujo de datos desde y hacia las entidades. A ellos se les llama “interactores”.
- Los adaptadores de interfaz (interface adapters), convierten los datos desde el formato más conveniente para los casos de uso y entidades. Aquí pertenecen los controladores y presentadores.
- Frameworks y drivers contiene los detalles de la UI, herramientas, frameworks como Android, etc.

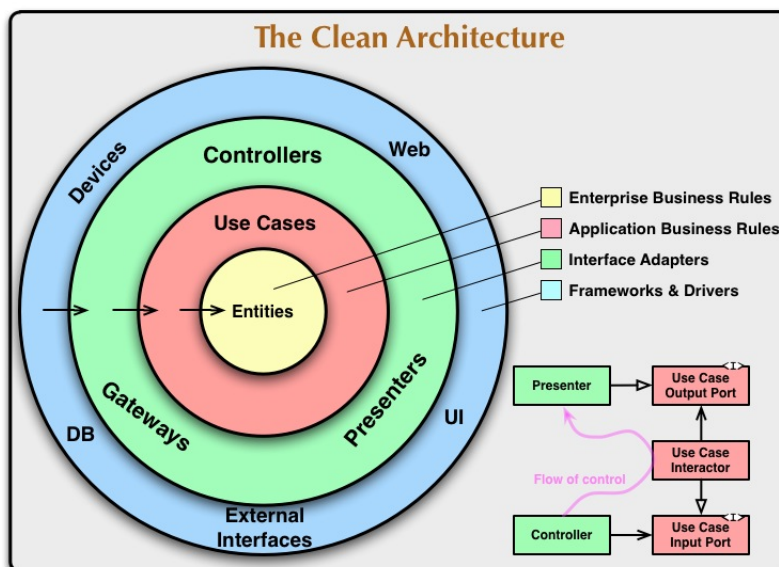


Figura 5. Diagrama de arquitectura limpia [9]

La arquitectura limpia se puede conseguir dividiendo el sistema en 3 capas, como se muestra en la Tabla 1, donde cada una cumple un propósito específico y funciona independiente de las otras. Cabe mencionar que cada capa usa su propio modelo de datos, de modo que esta independencia pueda ser alcanzada.

Capa	Responsabilidad
Presentación	Maneja la interacción entre el usuario y el software, usualmente a través de una interfaz de usuario (UI). En esta capa se encuentran los presentadores de MVP.
Dominio	Es la lógica de negocio. Trabaja con los datos realizando cálculos, verificaciones, transformaciones, etc.
Datos	Comunicación con bases de datos, web services, sistemas de mensajería, etc.

Tabla 1. Capas de software en arquitectura limpia

Interactores

Los interactores se ubican entre las capas de presentación y dominio ya mencionadas. Entre las capas de presentación y dominio ya mencionadas se ubican los interactores. Consisten en la implementación de clases e interfaces encargadas de procesar los datos que entrega la capa de dominio, desde bases de datos, servidores web o cualquier otra fuente. Luego de procesar los datos, un interactor los envía al presentador, para que sean mostrados en la vista o se hagan cambios en ella.

3.4.3 REST

La Transferencia de Estado Representacional (REST) es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en todos los formatos posibles, como XML y JSON. Es una alternativa en auge a otros protocolos estándar de intercambio de datos como SOAP (Simple Object Access Protocol), que disponen de una gran capacidad pero también mucha complejidad [10].

Algunas características de REST son [10]:

- Protocolo cliente/servidor sin estado: cada petición HTTP contiene toda la información necesaria para ejecutarla, lo que permite que ni cliente ni servidor necesiten recordar ningún estado previo para satisfacerla.
- Las operaciones más importantes relacionadas con los datos en cualquier sistema REST y la especificación HTTP son cuatro: POST (crear), GET (leer y consultar), PUT (editar) y DELETE (eliminar).
- Los objetos en REST siempre se manipulan a partir de la URI (identificador de recursos uniformes; es una cadena de caracteres que identifica los recursos de una red de forma unívoca). Es la URI y ningún otro elemento el identificador único de cada recurso de ese sistema REST. La URI facilita acceder a la información para su modificación o borrado.

Lo que permite que REST tenga ciertas ventajas [10]:

- El protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos, lo que se acopla muy bien a MVC y la arquitectura limpia.
- Al utilizar REST se garantiza que los métodos HTTP son seguros, lo que significa que al solicitar un recurso este requerimiento no modifica o causa ningún tipo de cambio en su estado.
- La API REST es independiente al tipo de lenguaje o tecnología con las que se esté trabajando, lo que ofrece libertad a la hora de escoger un lenguaje o tecnología que se adapte mejor al problema, siempre que permita el formato JSON.

4 Especificación de requisitos de Software

En este capítulo, se presentan los requisitos de software y sistema basados en el estándar ISO/IEC/IEEE 29148:2011. Se detalla el propósito, alcance, visión general, y requisitos específicos que cubre este proyecto.

4.1 Propósito

Este proyecto fue solicitado desde el Departamento de Servicios Tecnológicos de la Universidad del Bío-Bío, como parte del mejoramiento continuo en los procesos e imagen corporativa de la Universidad. Por este motivo, DST ha decidido participar de esa mejora continua mediante la actualización de uno de los servicios que presta a la comunidad universitaria, la plataforma educativa Adecca UBB.

4.2 Alcance

El alcance de este sistema se indica a continuación:

- El nombre del sistema debe ser “Adecca UBB”, al igual que el nombre de la plataforma.
- El sistema sólo estará disponible para dispositivos con sistema operativo Android Ice Cream Sandwich o superior.
- La documentación del sistema no deberá mostrar claves de servidor y claves de firma de aplicación.
- El sistema tendrá funcionalidades exclusivas de Adecca UBB y no de otras plataformas.

A continuación, se define el objetivo del software, para definir mejor el alcance del sistema.

4.2.1 Objetivo del software

Objetivo general

Mejorar el acceso a la plataforma Adecca UBB para los docentes y estudiantes de la Universidad del Bío-Bío a través de dispositivos móviles Android.

Objetivos específicos

- Facilitar la visualización del contenido de los cursos de un usuario.
- Facilitar la descarga de archivos.
- Facilitar la visualización de información personal.
- Facilitar la modificación de la información personal.
- Facilitar la entrega de información instantánea.
- Facilitar el acceso a información en línea de la Universidad del Bío-Bío.

4.3 Visión general del producto

En este punto, se presentan los requisitos del sistema en términos generales, incluyendo las limitaciones del mismo.

4.3.1 Producto en perspectiva

El sistema ha de ser una versión de la plataforma Adecca UBB, que permita tanto a docentes como estudiantes ingresar a la plataforma mediante un dispositivo móvil con sistema Android. Las funcionalidades que el sistema debe ofrecer a los usuarios corresponden a las funciones principales que les ofrece la plataforma, tales como tener una sesión de usuario personalizada, para ver los cursos en que participan y su contenido. Aparte, debe permitir el acceso a distintas páginas web de la Universidad.

4.3.2 Interfaz de usuario

La Universidad del Bío-Bío estipula los requisitos de apariencia en su Manual de Comunicación Corporativa, sin embargo, no se detallan los colores que deben ser usados en sistemas informáticos. En términos muy generales, especifica que “en cualquier representación gráfica de la Universidad, se privilegiará el uso del azul, el gris y los espacios en blanco, preferentemente sobre fondos claros” [11]. Por lo tanto, la apariencia del sistema se basa en los colores oficiales de la Universidad del Bío-Bío (azul, gris, rojo y amarillo), que se muestran en la Figura 6, variando ligeramente las tonalidades y la luminosidad.


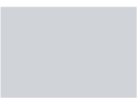


Uso	Color 1	Color 2	Color 3	Color 4
				
Imprenta	PANTONE 293 CVC	PANTONE 428 40%	PANTONE Warm red	PANTONE 123 CVC
Web	#0040B0	#D3DBDE	#FF3333	#FFCC33
RGB	r0-g0-b153	r211-g219-b222	r255-g54-b23	r255-g178-b15
CMYK	c100-m50-y0-k0	c21-m9-y10-k0	c0-m79-y91-k0	c0-m30-y94-k0

Figura 6. Colores oficiales de la Universidad del Bío-Bío

Otra consideración es el logotipo de la plataforma Adecca UBB, el cual debe mostrarse como ícono de la aplicación móvil (Figura 7), al abrir la aplicación y en el menú de esta (Figura 8).



Figura 7. Ícono de la plataforma Adecca UBB



Figura 8. Logotipo de la plataforma Adecca UBB

Considerando que los usuarios del sistema son diversos en sus capacidades de uso de tecnología, es necesario que la aplicación a desarrollar muestre claramente los textos en la pantalla, y que al inicio se muestren los botones con un nombre de la actividad que realizan para que la interacción sea intuitiva.

Además, la vista de la aplicación debe ser optimizada tanto para dispositivos con pantalla vertical, horizontal, teclado visible y oculto.

4.3.3 Funciones del producto

Las funciones que debe cumplir el sistema actual son las siguientes:

- El sistema gestionará el proceso de autenticación de usuarios, mediante las credenciales de Adecca UBB, que permita mostrar información personalizada a cada usuario.
- El sistema manejará información sobre el proceso de virtualización de cursos, que permita al usuario participar en los cursos en los que está inscrito en la plataforma Adecca UBB.
- El sistema gestionará el proceso de participación de usuarios en el recurso didáctico de tipo foro, para que puedan comunicarse con el resto de participantes.
- El sistema manejará el proceso de descargas de contenido, que permita a los usuarios obtener archivos adjuntos en los cursos y recursos didácticos.
- El sistema gestionará el proceso de recepción de notificaciones por parte de los usuarios, para que estén enterados en tiempo real de eventos que ocurran en sus cursos inscritos.

4.3.4 Limitaciones

Las limitaciones del sistema se indican a continuación:

- No usará librerías que provoquen un mayor impacto en el sistema o que sean incompatibles con las versiones de Android implicadas.
- El sistema no incorporará funciones que sean solo para docentes o funcionarios de la Universidad.
- El sistema no permitirá la inscripción de cursos.
- El sistema no incluirá funciones de otras plataformas educativas o administrativas de la Universidad.

4.4 Requisitos específicos

Los requisitos presentados a continuación, corresponden a lo solicitado por los Coordinadores del Departamento de Servicios Tecnológicos de la Universidad del Bío-Bío.

4.4.1 Requisitos funcionales

A continuación, la Tabla 2 presenta los requisitos funcionales del sistema. En primera instancia, se presentan los requisitos que dependen de la autenticación de un usuario, y a partir del requisito funcional N° 18 (RF_18), se encuentran los que están disponibles para los usuarios en general.

ID	Nombre	Descripción
RF_01	Inicio de sesión	El sistema debe permitir iniciar una sesión de usuario con el mismo Rut y contraseña de Adecca UBB.
RF_02	Visualización de cursos	El sistema debe mostrar un listado de los cursos en que participa el usuario (RF_01) en el tiempo actual.
RF_03	Visualización de cursos anteriores	El sistema debe mostrar un listado de los cursos en que participó el usuario (RF_01), organizados por el año en que se dictaron.
RF_04	Visualización de información de un curso	El sistema debe mostrar información propia de un curso (RF_02 y RF_03). Es decir, descripción del curso, avisos del docente a cargo y tiempo límite de tareas.
RF_05	Visualización de participantes de un curso	El sistema debe mostrar todos los participantes de un curso (RF_02 y RF_03).
RF_06	Envío de correo electrónico	El sistema debe permitir al usuario enviar correo electrónico a los integrantes de un curso (RF_05).
RF_07	Visualización de la estructura de un curso	El sistema debe mostrar la estructura del directorio de carpetas y recursos didácticos que tiene un curso (RF_02 y RF_03).
RF_08	Visualización de recurso didáctico	El sistema debe mostrar la descripción de un recurso didáctico, fecha de apertura y cierre (si tiene) y los enlaces adjuntos (RF_07).
RF_09	Descarga de archivos	El sistema debe mostrar los archivos que hay dentro de un recurso didáctico (RF_08) y descargarlos.
RF_10	Descarga de guía didáctica	El sistema debe permitir la descarga de la guía didáctica (RF_02 y RF_03).
RF_11	Descarga de programa de un curso	El sistema debe permitir la descarga del programa de un curso (RF_02 y RF_03).
RF_12	Visualización del foro	El sistema debe mostrar las entradas y respuestas del recurso didáctico tipo "Foro" (RF_08).
RF_13	Envío de entradas al foro	El sistema debe permitir al usuario enviar una nueva entrada al foro (RF_11), y responder a las entradas y respuestas que ya existen. Esto sólo si la fecha de cierre aún no se cumple o no tiene fecha.

RF_14	Envío de notificaciones	El sistema debe enviar notificaciones al móvil en caso de: <ol style="list-style-type: none"> 1. Nuevas tareas en un curso (RF_02). 2. Nuevas respuestas a una entrada del usuario en el foro (RF_12). 3. Nuevo tiempo límite en una tarea (RF_08). 4. Nuevos archivos en un recurso didáctico (RF_10). 5. Nuevo aviso por parte del docente o los docentes de un curso.
RF_15	Visualización de perfil de usuario	El sistema debe mostrar el perfil del usuario (RF_01).
RF_16	Edición del perfil de usuario	El sistema debe permitir editar el alias, sitio web, dirección y teléfono personal del perfil de usuario (RF_14).
RF_17	Cierre de sesión	El sistema debe permitir al usuario cerrar su sesión (RF_01).
RF_18	Visualización de calendario académico	El sistema debe mostrar el calendario académico del año en curso de la Universidad del Bío-Bío.
RF_19	Visualización de la web de Intranet UBB	El sistema debe mostrar la web de Intranet de la Universidad del Bío-Bío.
RF_20	Visualización la web de la biblioteca de la universidad	El sistema debe mostrar la web de la biblioteca de la Universidad del Bío-Bío.
RF_21	Visualización la web de DDE UBB	El sistema debe mostrar la web de la Dirección de Desarrollo Estudiantil de la Universidad del Bío-Bío.
RF_22	Visualización de “Acerca de Adecca UBB”.	El sistema debe mostrar información relativa a la plataforma Adecca UBB.
RF_23	Visualización de Noticias UBB en pantalla principal	El sistema debe mostrar la web de Noticias UBB en la pantalla principal de la aplicación.

Tabla 2. Requisitos funcionales del sistema

4.4.2 Requisitos no funcionales

A continuación, se indican los atributos o requisitos no funcionales del sistema:

- Usabilidad – Atracción

El sistema debe usar los colores de la Universidad del Bío-Bío según el Manual de Comunicación Corporativa, haciendo un uso atractivo de ellos de acuerdo al criterio del cliente.

- Portabilidad – Adaptabilidad

El sistema debe funcionar sin problemas ni errores en cualquier dispositivo móvil con sistema Android versión 4.0.3 (Ice Cream Sandwich) o superior.

- Funcionalidad – Seguridad

Los datos del sistema deben ser accedidos sólo a través de un usuario autenticado mediante el RUT y contraseña que se utiliza en la plataforma Adecca UBB.

- Eficiencia – Tiempo de respuesta

El sistema debe tardar menos de 15 segundos en mostrar las pantallas de información con los datos solicitados, de lo contrario, mostrar mensajes de error.

5 Factibilidad

En este capítulo, se presentan los resultados del estudio de factibilidad realizado para determinar cuán viable es el proyecto desde el punto de vista técnico, operativo y económico.

Dichos resultados están sustentados en un estudio realizado en la Universidad del Bío-Bío, en el cual participaron 20 docentes y 80 estudiantes de distintos departamentos y carreras, mediante la contestación de una encuesta vía correo electrónico para los estudiantes, y personalmente para los docentes. La documentación del estudio se encuentra en el Anexo 1, secciones 1.3, 1.4 y 1.5.

5.1 Factibilidad técnica

Se considera que para el desarrollo y mantención de este proyecto, se debe usar un computador con sistema operativo Microsoft Windows ya que es el sistema que la Universidad usa por defecto y, por lo tanto, tiene licencias disponibles para ello. Además, Microsoft Windows es compatible con las aplicaciones que se usan en este proyecto, tales como Android Studio, Java y PdaNet+.

Dada la relevancia de Android Studio, a continuación en la Tabla 3, se muestran los requisitos de hardware y software que solicita Android Studio para ser usado en Windows¹.

Requerimientos	
Sistema Operativo	Microsoft® Windows® 7/8/10 (32- o 64-bit).
RAM	3 GB RAM mínimo, 8 GB RAM recomendada; además de 1 GB para el Emulador de Android Studio.
Disco Duro	2 GB mínimos de espacio disponible en el disco, 4 GB recomendados (500 MB para el IDE + 1.5 GB para Android SDK y el emulador de imágenes del sistema).
Gráficos	1280 x 800 mínimos para la resolución de la pantalla.

Tabla 3. Requisitos de hardware y Software de Android Studio

En el Anexo 1.1, se muestra la ficha técnica de un computador portátil², cuyas características se ajustan a los requerimientos de hardware y software de Android Studio. Es necesario mencionar que se ha escogido un computador portátil en vez de un computador de escritorio, debido a que el primero es significativamente menos costoso y, por lo tanto, más conveniente para la Universidad.

Otro aspecto a considerar es la aptitud del recurso humano. Quien desarrolle y mantenga la aplicación móvil debe saber programar en lenguaje Java, y conocer aspectos generales del funcionamiento de Android Studio, cuya amplia documentación se encuentra en la web, en conjunto con numerosas guías, artículos y foros que simplifican el desarrollo.

¹ Datos obtenidos de <https://developer.android.com/studio/index.html>

² Ficha obtenida desde la web de pcfactory.cl

Tanto el hardware, software, y el recurso humano están disponibles para realizar este proyecto, por lo tanto, el proyecto es factible técnicamente.

5.2 Factibilidad operativa

El sistema se construye para usuarios que sólo necesitan saber utilizar un Smartphone. Según el estudio aplicado, el 82% de los encuestados utiliza un dispositivo móvil con sistema operativo Android.

En cuanto al uso de la plataforma Adecca UBB, un 80% de los estudiantes afirma utilizarla con frecuencia, y de esos usuarios un 95% indica que ha intentado abrirla desde un dispositivo móvil, principalmente para descargar archivos de un curso. Luego, un 95% de los estudiantes encuestados que usan la plataforma, están a favor de la existencia de una aplicación para dispositivos móviles de Adecca UBB que permita ver los cursos, descargar archivos, enviar correos y recibir notificaciones.

Para los docentes, un 65% de los encuestados afirma utilizar la plataforma, y de esos usuarios un 46% admite haber intentado ingresar desde un dispositivo móvil. De quienes utilizan la plataforma, un 83% afirma que le gustaría disponer de una versión móvil de ella.

En conclusión, el estudio aplicado revela que este proyecto es factible operativamente, debido que la mayoría de los encuestados usan la plataforma Adecca UBB y además, la mayoría, usa dispositivos móviles con sistema operativo Android.

5.3 Factibilidad económica

A continuación se presenta la factibilidad económica, incluyendo la inversión, los costos del sistema, los beneficios tangibles (en términos de ahorro) y los beneficios intangibles.

5.3.1 Inversión

Los ítems de la inversión son los siguientes:

- Sueldo de Ingeniero Civil en Informática: El sueldo de un Ingeniero Civil en Computación es de \$1.096.9483 al mes (aproximadamente).
- Hardware: Un computador como con las características necesarias para el desarrollo del proyecto cuesta alrededor de \$300.000 pesos chilenos, según el mercado en 2017.
- Play Store de Google: El costo de registrar la aplicación en Play Store de Google para que esté disponible al público es de \$25 dólares y se paga sólo una vez. El valor del dólar es de \$674 pesos chilenos al 30 de mayo del 2017.

A continuación, en la Tabla 4, se muestra un cuadro resumen con los conceptos de la inversión. El valor de cada concepto se encuentra expresado en pesos chilenos.

Concepto	Monto
Sueldo de Ingeniero Civil en Informática	\$4.387.792 (por 4 meses de trabajo)
Hardware	\$300.000
Play Store de Google	\$16.861
Total Inversión	\$4.704.653

Tabla 4. Resumen de la inversión

En total, la Universidad debería invertir \$4.704.653.

³ Fuente: <http://conexioningenieros.com/wp-content/uploads/2016/11/Estudio-de-Sueldos-Conexi%C3%B3n-Ingenieros-2016.pdf>

5.3.2 Costos

El único costo asociado al proyecto corresponde a la mantención para actualizar el software, que se debe realizar anualmente.

Mantención

La mantención ya indicada, se debe ejecutar por un Ingeniero Civil en Informática, que trabaje 8 horas diarias durante 15 días al año (120 horas mensuales). En la Tabla 5, se muestra un cuadro resumen con el cálculo de la mantención.

Detalle	
Sueldo de Ingeniero Civil en Informática	\$6.856 (por hora)
Horas de trabajo	120
Total Costos	\$822.720

Tabla 5. Resumen de los costos

En total, la Universidad debería usar \$822.720 pesos chilenos, para cubrir los costos en mantención.

5.3.3 Beneficios tangibles

Se ha determinado que los factores que influyen en el cálculo de los beneficios tangibles son dos. El primer factor corresponde al ahorro en el tiempo de trabajo de los docentes, y el segundo factor corresponde ahorro del uso de hardware de la Universidad, si se usa la aplicación móvil. Antes de presentar los factores, se describen datos generales que se deben usar para realizar los cálculos en ambos casos.

Datos para realizar los cálculos

Para calcular los beneficios tangibles se usan los siguientes datos:

- Estudiantes de pregrado: 10.597⁴
- Docentes: 567
- Estudiantes: La Universidad no tiene gastos asociados a remuneraciones de estudiantes. Los estudiantes sólo acceden a la plataforma a través de computadores en la Universidad o computadores fuera de esta.
- Docentes: Los docentes de la Universidad del Bío-Bío son remunerados. Trabajan en promedio 37 horas mensuales, y el sueldo promedio por hora es de \$12.000 pesos chilenos⁵.
- A través del estudio realizado, se calcula que el 65% de los docentes utilizan Adecca UBB y un 80% de los estudiantes la usan con frecuencia. Además, entre docentes y estudiantes, un 82% de la población encuestada utiliza un dispositivo con sistema operativo Android.
- Los cálculos anuales se basan en once meses, que es la cantidad de tiempo al año en que la Universidad permanece operativa.

⁴ Dato obtenido de https://es.wikipedia.org/wiki/Universidad_del_B%C3%ADo-B%C3%ADo “Universidad del Bío-Bío, Wikipedia”.

⁵ Datos obtenidos de http://ubiobio.cl/miweb/web2012.php?id_pagina=4353 “Dotación Personal de Planta, Transparencia Activa, Universidad del Bío-Bío”.

I. Se reduciría el tiempo de acceso a la plataforma Adecca UBB para los usuarios

Para calcular lo anterior se deben considerar los siguientes aspectos:

- Como se mencionó anteriormente, los estudiantes no se consideran en este ítem puesto que la Universidad no tiene gastos asociados a sus remuneraciones.
- Según los datos del estudio, se calcula que el 35% usaría el sistema para dispositivos Android. El resto de los docentes no se considera puesto que no utiliza la plataforma, o no utiliza un dispositivo Android, o no desea usar una aplicación móvil.
- De acuerdo al estudio realizado, un profesor accede a la plataforma desde un computador de la Universidad 20 veces al mes.
- Según una prueba aplicada a un docente del Departamento de Actividad Física de la Universidad (ver Tabla 23), tarda 53,5 segundos en responder una entrada en un foro. Este tiempo será considerado como tiempo estándar en realizar una operación básica dentro de la plataforma.
- Considerando las dos variables anteriores, en promedio un docente dedica 17,8 minutos de su tiempo mensualmente para trabajar en la plataforma desde el computador, realizando una operación básica.
- Se estima que existiendo el sistema en el dispositivo, realizar una operación básica tardará alrededor de 30 segundos. Si la misma operación básica probada en el ítem anterior se realiza a través del dispositivo, mensualmente un docente ocuparía 10 minutos.

La Tabla 6 muestra el cálculo de este factor, según los datos anteriores:

Variable	Detalle	Valor
V	Tiempo de trabajo en Adecca UBB en minutos	17,8 min.
W	Tiempo de ahorro de trabajo en minutos	10 min.
X	Costo por minutos de trabajo en pesos chilenos	\$200
Y	Cantidad de docentes	567
Z	Porcentaje de docentes que usarían el sistema	35%
F	Función de costo	$(V-W) * X * (Y * Z)$
Total mensual		\$308.880
Total anual		\$3.397.680

Tabla 6. Cálculo del tiempo de ahorro usando el sistema, en términos monetarios

La Universidad ahorraría \$3.397.680 pesos al año, valor que corresponde al tiempo que ahorrarían los docentes utilizando la plataforma en su versión móvil, en vez de la versión web.

II. Se reduciría el uso de hardware de la Universidad

Para calcular lo anterior se deben considerar los siguientes aspectos:

- Según el estudio realizado, un 65% de los estudiantes utilizaría la plataforma Adecca UBB en su versión móvil. El resto de los estudiantes no se considera puesto que no utiliza la plataforma, o no utiliza un dispositivo Android, o no desea usar una aplicación móvil.
- Un estudiante que utiliza la plataforma accede a ella desde la Universidad 3,6 veces al mes.
- En el caso de los docentes, y como se mencionaba en el ítem anterior, un 35% usaría la plataforma en su versión móvil.
- Como se mencionó previamente, los docentes usuarios de Adecca UBB en su versión web dedican aproximadamente 17,8 minutos al mes para utilizar la plataforma desde la Universidad.
- Mediante prueba (ver Anexo 1, Sección 1.2.1), se determinó que un estudiante tarda alrededor de 143 segundos en realizar una operación básica en la plataforma desde la Universidad.
- Según la Comisión Nacional de Energía, el valor de la electricidad es de \$61,872 /kWh en Chile⁶.
- Para este cálculo, se supone que un equipo de gama media consume 180W y un único monitor consume 40W⁷.

La Tabla 7, muestra el cálculo del monto que ahorraría la Universidad, si los estudiantes dejan de realizar operaciones básicas en la plataforma Adecca UBB, a través del hardware disponible en los campus.

Variable	Detalle	Valor
W	Veces en que un estudiante accede a la plataforma al mes, desde la Universidad	3,6
X	Tiempo de uso de Adecca UBB por estudiante en minutos	2,38 min.
Y	Watts que consume un computador	220W
Z	Costo de la electricidad por Kilowatt	\$61,872
A	Cantidad de estudiantes	10.597
B	Porcentajes de estudiantes que usarían el sistema	65%
F	Función de costo	$(((W * X) / 60) * Y) / 1000 * Z * (A * B)$
Total mensual		\$13.389
Total anual		\$147.276

Tabla 7. Cálculo del ahorro de hardware por estudiantes, en términos monetarios

⁶ Dato obtenido de <https://www.cne.cl/precio-medio-de-mercado-2/> "Precio Medio de Mercado, Comisión nacional de energía".

⁷ Datos obtenidos de <https://hardzone.es/2015/03/31/cuanto-cuesta-la-electricidad-que-consume-tu-pc/>

La Tabla 8, por su parte, muestra el cálculo del monto que ahorraría la Universidad, si los docentes dejaran de ocupar la electricidad de la misma para realizar operaciones básicas en la plataforma Adecca UBB.

Variable	Detalle	Valor
V	Tiempo de trabajo en Adecca UBB en minutos	17,8 min.
W	Watts que consume un computador	220W
X	Costo de la electricidad por Kilowatt	\$61,872
Y	Cantidad de docentes	567
Z	Porcentajes de estudiantes que usarían el sistema	35%
F	Función de costo	$((V / 60) * W / 1000) * X * (Y * Z)$
Total mensual		\$801
Total anual		\$8.815

Tabla 8. Cálculo del ahorro de hardware por docentes, en términos monetarios

Entonces, el cálculo del total de ahorro por reducción del uso de hardware de la Universidad, se muestra en la Tabla 9 a continuación:

Total estudiantes	\$147.276
Total docentes	\$8.815
Total	\$156.091

Tabla 9. Total de ahorro por reducción del uso de hardware de la Universidad

De acuerdo a lo anterior, la Universidad ahorraría \$156.091 pesos al año, valor que corresponde al ahorro en electricidad, producido por estudiantes y docentes que dejarían de usar el hardware de la Universidad para realizar operaciones básicas en Adecca UBB.

Total beneficios tangibles

La Tabla 10, a continuación, muestra la suma del resultado de los factores previamente presentados.

Reducción del tiempo en los accesos a la plataforma Adecca UBB	\$3.397.680
Reducción del uso de hardware de la Universidad	\$156.091
Total	\$3.553.771

Tabla 10. Total de ahorro

Por lo tanto, la Universidad podría ahorrar \$3.553.771 pesos chilenos al año, al disponer de la versión móvil de la plataforma Adecca UBB.

Valor actual neto

A continuación, la Tabla 11 muestra el cálculo del flujo de caja del proyecto y el valor actual neto, para determinar si el proyecto es factible económicamente.

	Años				
	1	2	3	4	5
Flujo de caja					
Ingresos	3.553.771	3.553.771	3.553.771	3.553.771	3.553.771
Costos	-822.720	-822.720	-822.720	-822.720	-822.720
Flujo de caja neto	2.731.051	2.731.051	2.731.051	2.731.051	2.731.051

Tabla 11. Flujo de caja

La fórmula para calcular el flujo de caja es la siguiente:

$$VAN = -I_0 \sum_{t=1}^n \frac{Ft}{(1+k)^t}$$

Donde, Ft son los flujos de dinero en cada periodo t, I₀ es la inversión realizada en el momento inicial (t = 0), n es el número de periodos de tiempo y k es el tipo de descuento o tipo de interés exigido a la inversión.

En este caso, la tasa de descuento⁸ corresponde a un 12%, y la inversión, como se presentó anteriormente, corresponde a \$4.704.653. Por lo tanto, el VAN a cinco años corresponde a \$5.140.175 pesos, lo que quiere decir que este proyecto es factible.

⁸ Según criterio de evaluación de la Vicerrectoría de Asuntos Económicos

5.3.4 Beneficios intangibles

Las mejoras en el acceso de la plataforma Adecca UBB pueden repercutir en los siguientes beneficios:

Alumno

- Mejor arraigo con la Universidad del Bío-Bío.
- Puede recomendar la Universidad a otras personas mostrando o mencionando la aplicación.

Universidad

- Lineamiento con las tecnologías móviles actuales.
- Mayor prestigio en niveles de servicio estudiantil.

5.4 Conclusión del estudio de factibilidad

Dado que el proyecto resulta factible técnica, operativa y económicamente, se puede concluir, por lo tanto, que el proyecto es factible, y resulta un beneficio para la Universidad del Bío-Bío.

Cabe mencionar que, debido al tamaño y la complejidad del entorno en que se evalúa la factibilidad operativa y económica, es probable que los resultados varíen.

6 Análisis

En este capítulo, se presenta el análisis previo al desarrollo del software, para comprender claramente la necesidad del cliente. Siguiendo la metodología de desarrollo, a continuación se muestran las historias de usuario obtenidas de los requisitos funcionales, además del Product Backlog definido para el proyecto y la planificación de Sprints, como parte del esfuerzo requerido.

Otro punto importante presentado aquí, corresponde a la definición de cómo se solucionará el problema, a través del enfoque de la solución y el modelo de datos.

6.1 Historias de usuario

A continuación, la Tabla 12 muestra las historias de usuario obtenidas de los requisitos, y según la especificación descrita en la sección 3.2.2. Considerando que en el Product Backlog (presentado más adelante) se ordenan las historias por prioridad, en este punto también se han definido en ese orden, para que exista una vinculación directa entre ambos. Con el fin de cumplir con la metodología de desarrollo, el resto de la documentación trabaja sólo con las historias de usuario en vez de los requisitos, ya que son más específicas e individuales.

ID	Requisito funcional	Descripción
HU_01	RF_01	Como usuario registrado, quiero iniciar una sesión con los datos de Adecca UBB para acceder a mi información personal.
HU_02	RF_02	Como usuario registrado, quiero ver un listado del nombre de los cursos en que participo actualmente para acceder a su información.
HU_03	RF_07	Como usuario registrado, quiero ver la estructura de carpetas y recursos didácticos de un curso para comprender mejor la organización del mismo.
HU_04	RF_04	Como usuario registrado, quiero ver fechas de tareas por vencer de un curso para saber cuánto tiempo tengo para subir cada tarea.
HU_05	RF_04	Como usuario registrado, quiero ver la descripción de un curso para saber de qué se trata.
HU_06	RF_08	Como usuario registrado quiero ver la descripción de un recurso didáctico para saber de qué se trata.
HU_07	RF_09	Como usuario registrado, quiero ver y descargar los archivos que hay dentro de un recurso didáctico para acceder a su contenido.
HU_08	RF_08	Como usuario registrado, quiero saber la fecha de apertura y la fecha de cierre de un recurso didáctico para saber si debo participar en él antes de que cierre.
HU_09	RF_08	Como usuario registrado, quiero ver enlaces adjuntos de un recurso didáctico para acceder a su contenido.
HU_10	RF_10	Como usuario registrado, quiero descargar la guía didáctica de un curso para acceder a su contenido.
HU_11	RF_11	Como usuario registrado, quiero descargar el programa de un curso para acceder a su contenido.
HU_12	RF_17	Como usuario registrado, quiero cerrar mi sesión de usuario para no exponer mis datos ante otros usuarios que usen el mismo dispositivo.

HU_13	RF_12	Como usuario registrado, quiero ver las entradas del recurso didáctico tipo Foro para leerlas.
HU_14	RF_12	Como usuario registrado, quiero ver las respuestas a las entradas del recurso didáctico tipo Foro para leerlas.
HU_15	RF_13	Como usuario registrado, quiero enviar una nueva entrada al foro para agregar información.
HU_16	RF_13	Como usuario registrado, quiero responder una entrada existente para hacer consultas, comentarios o añadir información.
HU_17	RF_03	Como usuario registrado, quiero ver un listado del nombre de los cursos en los que participé, ordenados por año, para acceder a su información.
HU_18	RF_05	Como usuario registrado, quiero ver un listado de los nombres de los participantes de un curso para saber quiénes son.
HU_19	RF_06	Como usuario registrado, quiero enviar un correo electrónico a cualquier participante para poder comunicarme con él o ella.
HU_20	RF_15	Como usuario registrado, quiero ver mi información pública para saber qué datos pueden ver otras personas desde la web de Adecca UBB.
HU_21	RF_19	Como usuario, quiero ver la web de Intranet UBB para acceder a ella rápidamente.
HU_22	RF_18	Como usuario, quiero ver el calendario académico de la Universidad del Bío-Bío para acceder a su información rápidamente.
HU_23	RF_21	Como usuario, quiero ver la web de Desarrollo Estudiantil UBB para acceder a ella rápidamente.
HU_24	RF_20	Como usuario, quiero ver la web de la biblioteca UBB para acceder a ella rápidamente.
HU_25	RF_04	Como usuario registrado, quiero ver avisos importantes de un curso para seguir indicaciones del docente a cargo.
HU_26	RF_14	Como usuario registrado, quiero recibir una notificación cuando hay una nueva tarea en un curso para saber de qué se trata y cuál es el plazo de entrega.
HU_27	RF_14	Como usuario registrado, quiero recibir una notificación cuando alguien responda a mi entrada en un foro para ver qué dice instantáneamente.
HU_28	RF_14	Como usuario registrado, quiero recibir una notificación en caso de que el plazo de entrega de una tarea se cambie para estar al tanto de ello instantáneamente.
HU_29	RF_14	Como usuario registrado, quiero recibir una notificación cuando se suba un nuevo archivo a un recurso didáctico para acceder a su contenido instantáneamente.
HU_30	RF_14	Como usuario registrado, quiero recibir una notificación cuando un docente ponga un aviso en un curso para estar al tanto de esa información instantáneamente.
HU_31	RF_16	Como usuario registrado, quiero actualizar mi alias, sitio web, dirección y teléfono para que otros usuarios puedan acceder a la última versión de esos datos desde la web de Adecca UBB.
HU_32	RF_23	Como usuario, quiero ver las noticias de la Universidad del Bío-Bío para estar al tanto de los últimos acontecimientos en la Universidad.
HU_33	RF_22	Como usuario, quiero ver información relativa a la plataforma Adecca UBB para saber sobre su historia y desarrollo.

Tabla 12. Historias de Usuario

6.2 Product Backlog

A continuación, la Tabla 13 muestra el Product Backlog del proyecto, según Scrum. Como se dijo anteriormente, la identificación de las historias de usuario está estrechamente relacionada con la prioridad de las mismas, según el criterio del cliente.

Prioridad	ID
1	HU_01
2	HU_02
3	HU_03
4	HU_04
5	HU_05
6	HU_06
7	HU_07
8	HU_08
9	HU_09
10	HU_10
11	HU_11
12	HU_12
13	HU_13
14	HU_14
15	HU_15
16	HU_16
17	HU_17
18	HU_18
19	HU_19
20	HU_20
21	HU_21
22	HU_22
23	HU_23
24	HU_24
25	HU_25
26	HU_26
27	HU_27
28	HU_28
29	HU_29
30	HU_30
31	HU_31
32	HU_32
33	HU_33

Tabla 13. Product Backlog

6.3 Esfuerzo requerido

En Scrum, los puntos de historia son una de las formas más utilizadas para medir el tamaño de un proyecto. Estos puntos están influenciados por varios factores, como complejidad y tamaño físico. Ambos factores no están directamente relacionados entre sí, ya que el desarrollo de una historia de usuario puede involucrar pocas líneas de código, por ejemplo, pero componen un algoritmo cuya lógica es muy compleja [5].

Para el cálculo de los puntos de historia en esta metodología no se ocupan unidades de medida, sin embargo, se suele utilizar la escala de Fibonacci modificada (1, 2, 3, 5, 8, 13, 20, 40, y 100), propuesta por Mike Cohn⁹.

El proceso del cálculo se indica a continuación:

- Se presentan todas las historias de usuario al equipo de desarrollo.
- Se escoge la historia de usuario más pequeña y fácil de realizar, y se le asigna 1 punto.
- Se compara el resto de las historias de usuario con la anterior, y se le asignan los puntos de acuerdo al grado de dificultad. Por ejemplo, la segunda historia de usuario escogida es tres veces más difícil de realizar que la primera, se le asignan 3 puntos.

Si una historia de usuario tiene demasiados puntos (según el límite definido por el equipo de desarrollo), entonces se debe dividir en historias más pequeñas [12].

A continuación, la Tabla 14 muestra los puntos calculados por historia de usuario:

ID	
HU_01	13
HU_02	8
HU_03	20
HU_04	5
HU_05	5
HU_06	5
HU_07	13
HU_08	5
HU_09	8
HU_10	3
HU_11	3
HU_12	1
HU_13	13
HU_14	20

⁹ Uno de los contribuyentes a la invención de Scrum.

HU_15	8
HU_16	8
HU_17	13
HU_18	5
HU_19	3
HU_20	13
HU_21	3
HU_22	3
HU_23	3
HU_24	3
HU_25	5
HU_26	13
HU_27	13
HU_28	13
HU_29	13
HU_30	13
HU_31	13
HU_32	3
HU_33	2
Total	272

Tabla 14. Puntos de historia

La historia correspondiente al cierre de sesión de usuario (HU_12), es considerada la más pequeña y fácil de todas, ya que sólo implica mostrar la vista de usuario sin autenticar.

Se ha determinado junto al Product Owner que, la implementación de las historias de usuario debería realizarse en seis Sprints. Para que los Sprints estén balanceados, deben cumplir aproximadamente 45 puntos cada uno (272 dividido en 6). Entonces, la asignación de historias por Sprint según la restricción anterior, se establece a continuación en la

Tabla 15.

Sprint	Rango de historias según prioridad	Total de puntos
1	1 – 4	46
2	5 – 12	43
3	13 – 16	49
4	17 – 24	46
5	25 – 28	44
6	29 – 33	44

Tabla 15. Definición de Sprints

Se estima que 1 punto de historia tardaría 1,5 horas en ser desarrollado (considerando el trabajo que se debe realizar en HU_12, y en base a proyectos anteriores). Entonces, para determinar la duración de los Sprints, se ha hecho un cálculo con los datos mostrados en la Tabla 16 .

Variable	Detalle	Valor
H	Tiempo que tarda en ser construido 1 punto de historia.	1,5
P	Puntos por Sprint (aprox.).	45
T	Horas de trabajo diarias	8
F	Función de duración	$(H * P) / T$
Resultado		8,44 días

Tabla 16. Cálculo de la duración de un Sprint

De acuerdo a lo anterior, cada Sprint necesita alrededor de 9 días para ser completado. Considerando que la semana laboral es de 5 días, y que se debe dejar un tiempo para realizar las pruebas necesarias, se determina que los Sprints deben durar 2 semanas cada uno, para cumplir con los puntos acordados y mantener comunicación constante con el Product Owner.

6.4 Enfoque de la solución

Considerando que la plataforma Adecca UBB es un sistema web, para cumplir las historias de usuario se debe usar una arquitectura que obtenga los servicios que ofrece la plataforma, y así mostrarlos en la versión móvil de ella.

El servidor de Adecca UBB posee una API propia, por la cual se obtienen los servicios. Esta API permite que los datos se transmitan de manera privada y segura, a través de una arquitectura REST ya establecida.

De acuerdo a lo anterior, la aplicación móvil debe ser un consumidor de servicios web, con arquitectura REST.

6.5 Modelo de datos

A continuación, la

Figura 9 muestra el modelo de datos de la aplicación, mediante un diagrama de entidad-relación que utiliza la Notación de Peter Chen (Sección 3.3.1). Ya que la aplicación es un consumidor de servicios web, no dispone de una base de datos propia, por lo tanto el modelo es una referencia a los datos que debe recibir la aplicación desde el servidor.

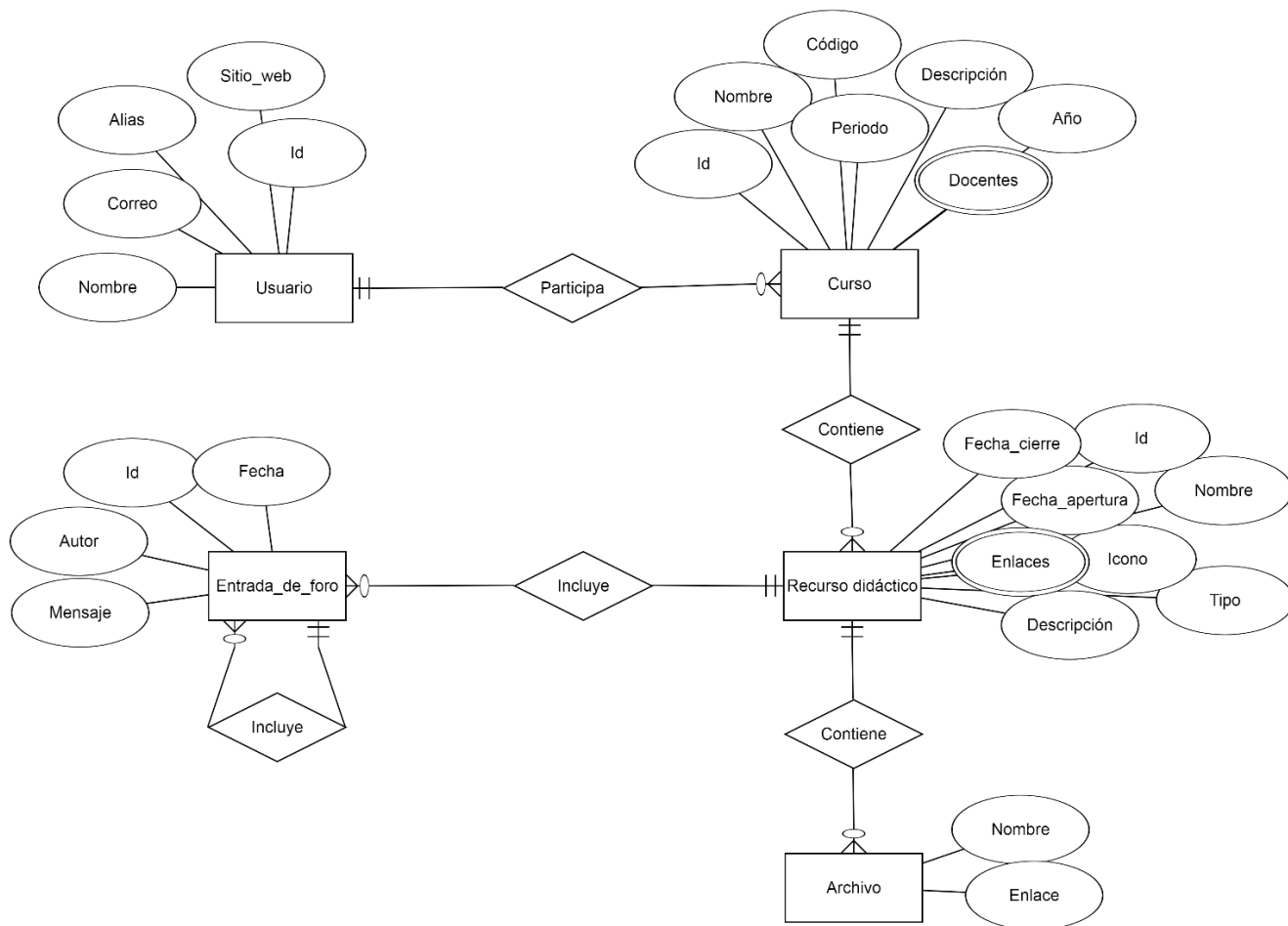


Figura 9. Modelo Entidad-Relación del sistema

Dentro de cada curso pueden existir distintos tipos de recursos didácticos. Los recursos didácticos pueden ser de tipo tarea, cuestionario, foro, archivos, trabajo en parejas, JIT (simulación de tecleras), entre otros. Cada uno de ellos, además, puede o no incluir archivos adjuntos, descripción, fechas de apertura y cierre, y enlaces adjuntos.

Cuando un recurso didáctico es de tipo foro, se incluyen todas las entradas que este contiene. Las entradas pueden tener distintas respuestas, y las respuestas también pueden ser respondidas, provocando una serie de mensajes en cascada. Por este motivo, el esquema muestra un ciclo en la entidad “Entrada_de_foro”.

7 Diseño

En este capítulo se presenta el diseño de la aplicación, enfocado en el desarrollo de la misma. Los puntos tratados a continuación corresponden a la implementación de las definiciones y estipulaciones hechas en los capítulos anteriores, como arquitectura, diseño de interfaces, entre otros.

7.1 Diseño orientado al consumo de servicios

En este capítulo se muestra la forma en que opera la plataforma Adecca UBB, además de los recursos y servicios que deben ser consumidos para cumplir con las historias de usuario, asimiladas al proceso de la plataforma.

7.1.1 Estructura de Adecca UBB

En la Figura 10 se muestra la estructura que usa la plataforma Adecca UBB para organizar sus contenidos. A partir de este esquema, se define qué información es relevante en el proceso para determinar los recursos y servicios que debe consumir la aplicación de este proyecto.

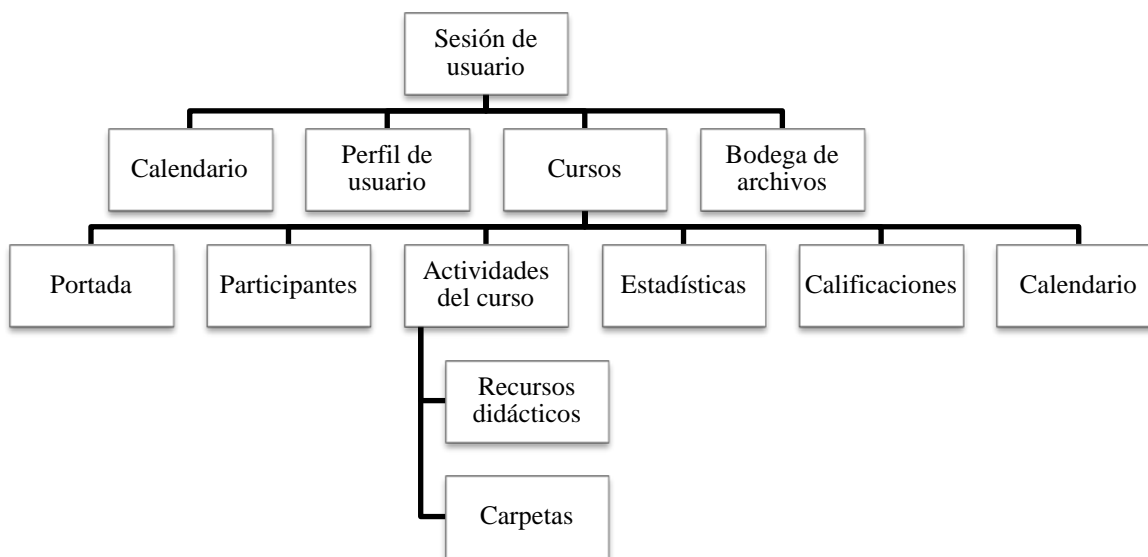


Figura 10. Esquema representativo de la estructura de Adecca UBB

Cuando se ingresa a la plataforma Adecca UBB mediante la dirección www.adecca.ubiobio.cl, antes que todo, se solicita la autenticación del usuario mediante RUT y contraseña. Una vez que el usuario se ha autenticado, es decir, ha iniciado su sesión, se muestra en su pantalla principal todos los cursos a los que pertenece el usuario en ese semestre, seguido de una opción que permite ver los cursos anteriores divididos por año y semestre. Además, se muestran opciones para ver el perfil de usuario, ver el calendario personal y la bodega, donde se encuentran todos los archivos personales y entregados en actividades de cursos. Un ejemplo de ello se encuentra en el Anexo 2, Figura 45.

Al ingresar a un curso, se muestra la portada de este, que contiene su descripción y fechas por vencer. Estas últimas corresponden a las fechas de los recursos didácticos que están por cerrarse dentro de una semana. También se muestran opciones para ver la portada del curso, ver los participantes, el calendario, las calificaciones y las estadísticas. A la izquierda de la pantalla se muestra el contenedor de las actividades del curso. En el Anexo 2, Figura 46 se muestra un ejemplo de lo que se ve al ingresar a un curso.

Las actividades del curso están organizadas de manera flexible, dependiendo del criterio del docente o administrador del curso. Existen carpetas, y recursos didácticos o “módulos”, estos últimos pueden ser de tipo “Trabajo o tarea individual”, “Trabajo o tarea grupal”, “Cuestionario”, “Foro”, “Recursos” (archivos), “Revisiones entre pares”, “Revisiones entre pares grupal”, “Evaluación fuera de línea”, “Justo a tiempo” (JIT), “Autoevaluación Estrategias de Aprendizaje”, “Chat”, “Diario reflexivo” y “Entrega Simple”. Los recursos didácticos pueden ir tanto fuera como dentro de las carpetas.

Como se muestra en la Figura 11, un ejemplo de organización y estructura de actividades, podrían ser carpetas que representen las distintas unidades en las que se divide el curso; dentro de ellas se podría incluir, por ejemplo, un recurso didáctico que contenga los archivos y enlaces de esa unidad, otro recurso didáctico que sea una tarea, u otro que sea una evaluación, etc.

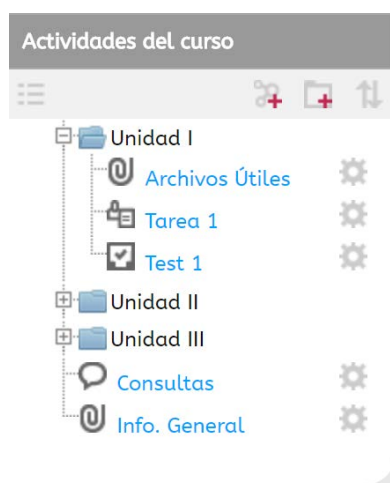


Figura 11. Ejemplo de carpetas y recursos didácticos

7.1.2 Recursos y servicios

Para cumplir con las historias de usuario, y seguir la lógica de Adecca UBB, en la Tabla 17 se han definido los recursos y servicios que la aplicación ha de consumir.

ID	Recurso	Servicio	Observaciones	HU asociada
Q_User	Usuario	Consulta que, a partir del RUT y la contraseña de un usuario retorna código, ID, mensaje, nombres, apellidos y correo del usuario.	La respuesta está cifrada en Base64. Si el código es 0 la autenticación fue incorrecta. Perfil es la ID del usuario. Mensaje corresponde a “éxito” o “fracaso” para la consulta. Nombres, Apellidos y Correo oficial corresponden a los registrados por la Universidad.	HU_01
Q_Curso	Curso	Consulta que, a partir del ID del usuario y del tipo de curso retorna id, nombre, código, periodo, año y docentes de un curso, por cada curso que tiene el usuario.	Tipo de curso corresponde a: 0, si es curso anterior. 1, si es curso actual. Código puede contener más de un elemento ya que cada uno corresponde a las secciones de ese curso. Periodo se refiere al semestre en que se dicta el curso. Docentes corresponde al o los nombres de los docentes que dictan el curso.	HU_02 y HU_17
Q_Guia	Guía Didáctica	Consulta que, a partir del ID de un curso retorna el enlace de descarga del archivo de la guía didáctica.	El enlace contiene nombre, id y ruta del archivo. Parte de la respuesta está cifrada en Base64. Ruta corresponde a la dirección del archivo en el servidor.	HU_10
Q_Prog	Programa de asignatura	Consulta que, a partir del ID de un curso retorna el enlace de descarga del archivo del programa de asignatura.	El enlace contiene nombre, id y ruta del archivo. Parte de la respuesta está cifrada en Base64. Ruta corresponde a la dirección del archivo en el servidor.	HU_11
Q_Info	Información de curso	Consulta que, a partir del ID de un curso y del ID del usuario retorna la descripción del curso, avisos y fechas de eventos próximos, además de id, nombre, icono y tipo de un recurso didáctico, por cada recurso que tiene un curso. También retorna el id y el nombre de las carpetas que contienen los recursos didácticos.	Se entiende por eventos evaluaciones o actividades con una fecha de término fija. Puede no contener descripción del curso. Puede no contener fechas de eventos próximos. Puede no contener avisos. Los tipos de recurso pueden ser: “Trabajo o tarea individual”, “Trabajo o tarea grupal”, “Cuestionario”, “Foro”, “Recursos” (archivos), “Revisiones entre pares”, “Revisiones entre pares grupal”, “Evaluación fuera de línea”, “Justo a tiempo” (JiTT), “Autoevaluación Estrategias de Aprendizaje”, “Chat”, “Diario reflexivo” y “Entrega Simple”. Puede no contener carpetas. Puede no contener recursos didácticos.	HU_03, HU_04, HU_05, HU_06, HU_25

Q_Party	Participante	Consulta que, a partir del ID de un curso retorna el nombre, apellido y correo de cada participante de ese curso.	Dependiendo de la posición en el arreglo que se retorna como respuesta, se puede identificar qué tipo de participante es. Cada posición contiene una lista de participantes. En caso de que no haya participantes, la posición del arreglo es vacía. Las posiciones corresponden a los siguientes tipos de participantes: 0, Docentes. 1, Ayudantes. 2, Estudiantes.	HU_18, HU_19
Q_Rec	Recurso didáctico	Consulta que, a partir del ID de un recurso didáctico retorna la id, nombre, descripción, fecha de activación, fecha de cierre, archivos y enlaces de ese recurso didáctico.	Puede no contener descripción. Puede no contener fecha de cierre. Archivos puede contener cero o varios elementos. Vínculos puede contener cero o varios elementos.	HU_07, HU_08, HU_09
Q_Foro	Foro	Consulta que, a partir del ID de un recurso didáctico, si este es de tipo "Foro", retorna el id, mensaje, fecha y autor de todas las entradas que hay dentro de ese foro.	Puede no contener descripción. Puede no contener entradas.	HU_13
Q_Answ	Respuesta a entrada de foro	Consulta que, a partir del ID de una entrada de foro o de la ID de una respuesta, retorna id, mensaje, fecha, autor de todas las respuestas a esa entrada o respuesta.	Dentro de un recurso didáctico de tipo foro en la plataforma, está permitido responder respuestas, por lo tanto una entrada puede contener respuestas anidadas infinitamente con los mismos atributos.	HU_14
Q_Perfil	Perfil	Consulta que, a partir del ID del usuario, retorna el alias, correo, web, dirección y teléfono del usuario.	Exceptuando el correo, puede que no contenga el resto de la información.	HU_20
A_Entry	Nueva entrada de foro	Agregación que, añade una nueva entrada o respuesta a un foro mediante la id del recurso didáctico, id del usuario, id de la entrada padre, id de la respuesta a la entrada padre y el mensaje de la nueva entrada. Retorna 1 si la respuesta fue agregada correctamente, 0 si no lo fue.	Cuando la entrada nueva no responde ninguna otra entrada, id de entrada padre e id de respuesta a entrada padre son parámetros nulos. Cuando la entrada nueva es una respuesta a una entrada, id de entrada padre corresponde a la id de la entrada a responder, e id de respuesta a la entrada padre es un parámetro nulo. Cuando la entrada nueva responde la respuesta de una entrada padre, se incluyen todos los parámetros.	HU_15, HU_16
A_Perfil	Nuevo perfil	Agregación o Actualización que, mediante la id del usuario añade o edita el sitio web, su dirección y teléfono.	El usuario puede escoger qué atributos agrega y edita. Los atributos no escogidos son nulos.	HU_31
A_Noti	Notificación	Consulta que, a partir del ID de un usuario, Token e IMEI del dispositivo, retorna el estado de la petición.	Token corresponde a una clave única, generada por la plataforma Firebase, para cada usuario. IMEI corresponde al código del dispositivo móvil. El estado de la petición corresponde a success o failure (éxito o fracaso).	HU_26, HU_27, HU_28, HU_29, HU_30

Tabla 17. Recursos y Servicios

Fuera de los servicios definidos, HU_12 corresponde a una función que cambia la vista de un usuario autenticado, a un usuario no autenticado, por lo tanto, no utiliza servicios web. Las historias HU_21, HU_22, HU_23, HU_24 y HU_32, se implementan a través de una función de Android que permite mostrar una página web dentro de la aplicación, o abrir la página en el navegador del dispositivo, usando sólo la URL. Por otra parte, HU_33 no utiliza servicios web, ya que la información de Adecca UBB no varía con frecuencia, por lo tanto, puede incluirse directamente en la aplicación, a modo de texto.

Entonces, de acuerdo a lo anterior, todas las historias de usuario quedan cubiertas por los recursos y servicios definidos, y por las funcionalidades explicadas en el párrafo anterior.

Aunque los servicios son independientes, normalmente siguen una secuencia de peticiones para poder ser llamados, ya que necesitan los datos que se obtuvieron en una petición anterior (usualmente identificadores). La Figura 12 muestra un esquema de la secuencia que se debe seguir, comenzando por Q_User, el único servicio que no necesita los datos de otras peticiones.

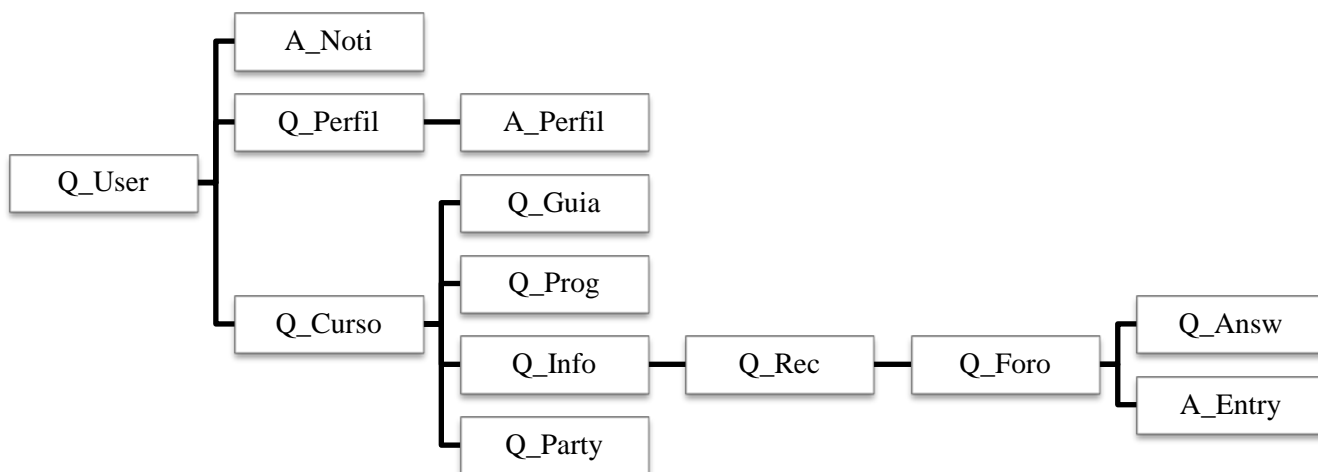


Figura 12. Secuencia de servicios

7.2 Diseño de arquitectura

A continuación, se presentan las distintas arquitecturas que utiliza el sistema, incluyendo el sistema de notificaciones.

7.2.1 MVP

La aplicación desarrollada en este proyecto ocupa el Modelo Vista Presentador para organizar sus capas, como se explica en la definición del proyecto (Sección 3.4.1).

A continuación, se presenta un ejemplo del proceso que es realizado para mostrar los cursos de un usuario. Se parte cuando el usuario envía su RUT y contraseña a través de la pantalla de inicio de sesión del sistema. Luego, aquellos datos se envían al servicio Q_User como parámetros, junto con el IMEI o serie del dispositivo. Una vez que la petición de este servicio ha sido exitosa, se devuelven los datos del usuario al controlador de la vista, entre ellos el ID de usuario, que se guarda en la memoria interna del dispositivo.

Posteriormente, el sistema muestra la vista del menú principal y página de inicio, desde donde se accede a la opción que muestra el listado de los cursos del usuario, mediante la petición del servicio Q_Curso. Para realizar esta petición, el sistema obtiene el ID de usuario desde la memoria interna y lo envía como parámetro en el servicio. El resultado se pasa hasta el controlador de la vista, que ordena los datos para que se muestren en un listado.

La Figura 13 muestra este proceso con mayor detalle, a través de las capas de arquitectura limpia y MVP; las flechas azules representan el flujo de datos desde las acciones del usuario hasta la petición del servicio, por el contrario, las flechas naranjas representan el flujo desde que la petición retorna los datos hasta que los muestra en pantalla.

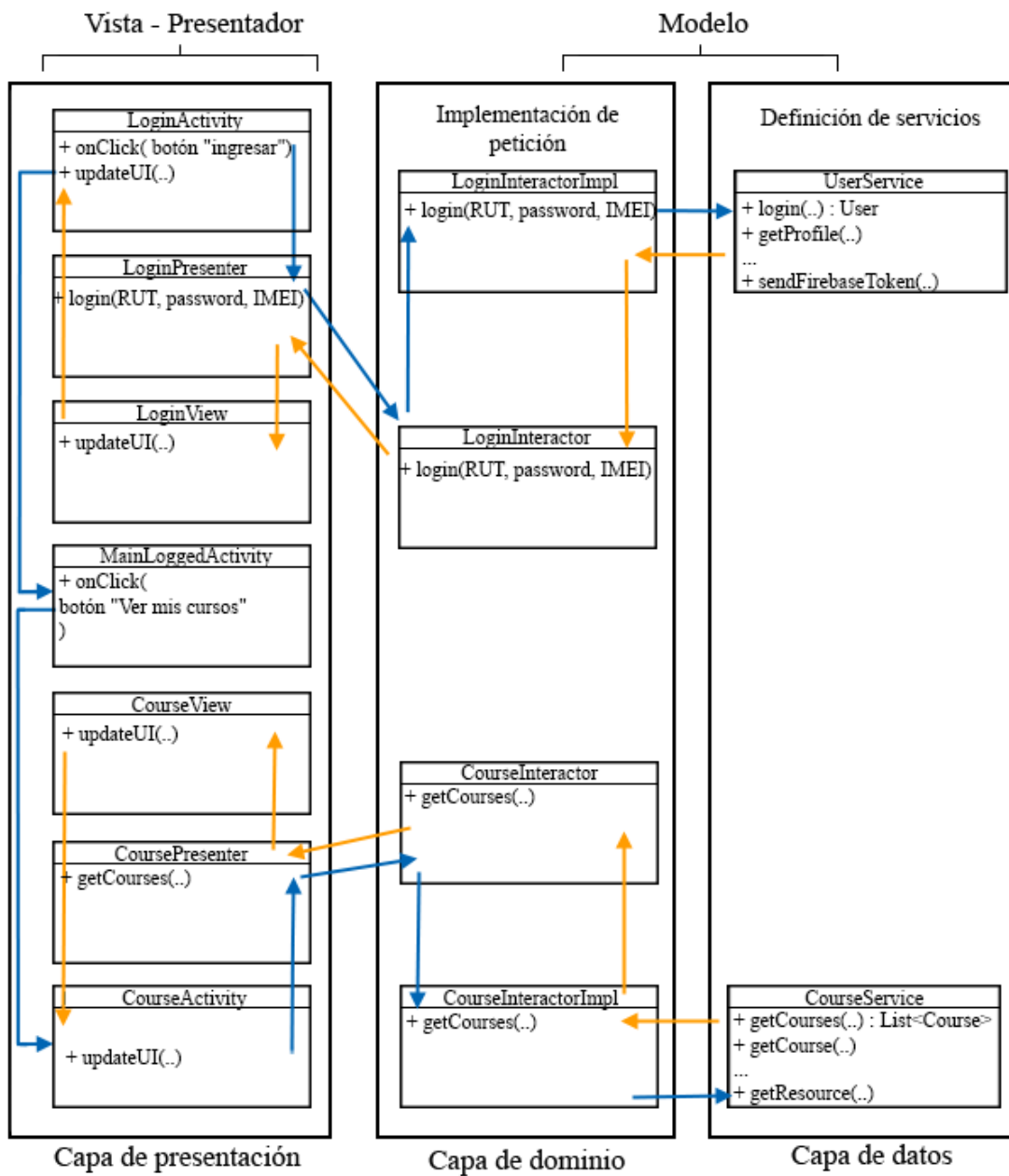


Figura 13. Ejemplo de capas en la aplicación

A continuación se indican las implementaciones que se realizan de manera general en cada capa:

- En la capa de datos, se encuentran interfaces que implementan los servicios previamente descritos, para proveer datos al resto de las capas.
- Posteriormente, y cumpliendo con la capa de dominio, por cada servicio se crea un interactor, que extrae los datos recibidos desde la capa de datos, y los procesa a través de bibliotecas de código abierto como lo es GSON de Google, para transformarlos a objetos.
- Se crea además un presentador ligado a una interfaz de la vista. El presentador captura el objeto que retorna el interactor y lo pasa a la vista, la cual finalmente, es llamada por el controlador de la vista que corresponde a una actividad o fragmento. Para presentar el objeto extraído, la vista puede utilizar distintos recursos como lo son adaptadores de listas, árboles u otras estructuras que representen al objeto de mejor manera. Es claro que en este punto se encuentra la capa de presentación.

En conclusión, el proceso de desarrollo cumple con la estructura de capas, y con las propuestas de MVP, ya que la vista no extrae ni procesa datos en bruto; el presentador sólo es un conector entre el modelo y la vista, y el modelo por sí solo está libre de frameworks, por lo tanto, los datos que extrae y procesa pueden ser probados fácilmente.

7.2.2 REST

Las peticiones de servicios, según la arquitectura REST vista en la Sección 3.4.3, se deben realizar mediante librerías como Retrofit y Volley, que convierten la API HTTP en una interfaz de Java. En este caso, no es posible usar Volley, ya que utiliza el “contexto” de la aplicación para realizar las peticiones, el cual es un elemento de Android, y su uso en el modelo es un incumplimiento a las condiciones de MVP. Retrofit, en cambio, sólo usa Java para comunicarse con el servidor, adaptándose bien al Modelo-Vista-Presentador, por lo que su uso es apropiado en esta aplicación.

En la implementación, las clases que tienen relación directa con los servicios corresponden a los interactores de cada uno de ellos, y dos interfaces llamadas CourseService y UserService. Estas interfaces definen un método por cada servicio, donde se incluye los parámetros a enviar y el tipo de objeto que se debe obtener. Existe, además, una clase que intercepta todas las peticiones, que agrega la cabecera correspondiente a la clave de la API de Adecca UBB, para que no sea incluida en cada método y de este modo eliminar la redundancia. La clave de API corresponde a una secuencia de caracteres que solo conoce el cliente y el servidor. Se usa como un método de seguridad para evitar que intrusos accedan a los datos.

La interfaz de UserService contiene Q_User, Q_Perfil, A_Perfil y A_Noti, mientras que CourseService contiene todos los servicios restantes. De este modo, los datos que son directamente del usuario, quedan separados de los datos que son directamente de los cursos. Se hace esta separación para que ante un posible crecimiento del sistema, quien desarrolle tenga claridad sobre la ubicación de los servicios, a través de la identificación del tipo de cada uno.

A continuación, la Tabla 18 y la Tabla 19 muestran las peticiones de Q_User y Q_Curso ya vistas, que corresponden a un método POST y un método GET respectivamente. Ambas tablas muestran un identificador del servicio, la URL de los datos en el servidor, el método de envío de los datos, y los parámetros que se deben enviar, incluyendo la clave privada del servidor. Además, se muestran los tipos de respuesta que retorna la petición al servicio. En caso de que la petición sea exitosa (código 200), se retorna un objeto en formato JSON que contiene los datos que requiere la aplicación. Se incluye también un ejemplo acotado del código de la petición, en Retrofit. El resto de las peticiones se encuentran en el Anexo 2, Sección 2.2.

ID	Q_User
URL	http://146.83.196.237/wadecca/index.php/api/android/login/format/json/
Método	POST
Parámetros de la URL	rut/{rut}/password/{password}/imei/{imei}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>{ "code": 1, "perfil": 4288, "nombres": "Rocío Macarena", "apellidos": "Sepúlveda Carriel", "correo": "rocsepul@alumnos.ubiobio.cl" }</pre>
Respuesta errónea	<p>Ejemplo 1:</p> <p>Código: 400, solicitud incorrecta</p> <p>Contenido: null</p> <p>Ejemplo 2:</p> <p>Código: 403, acceso denegado</p> <p>Contenido: null</p>
Llamada de ejemplo	<pre>public interface UserService { @POST("rut/{rut}/password/{password}/imei/{imei}") Call<Login> login(@Path("rut") String rut, @Path("password") String password, @Path("imei") String imei); } public LoginInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_LOGIN) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()) .build(); service = retrofit.create(UserService.class); }</pre>

Tabla 18. Ejemplo de petición POST

ID	Q_Curso
URL	http://146.83.196.237/wadecca/index.php/api/android/getCourses/format/json/
Método	GET
Parámetros de la URL	id_usuario/{id_usuario}/tipo_cursos/{tipo_cursos}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>[{ "id": 4864, "nombre": "Prueba 2017", "año": "2017", "codigo": ["634007-1"], "periodo": "Segundo Semestre", "docentes": ["Carlos Eduardo San Juan Contreras", "David Abdías Sepúlveda Riquelme"] }, {...}]</pre>
Respuesta errónea	<p>Ejemplo 1:</p> <p>Código: 400, solicitud incorrecta</p> <p>Contenido: null</p> <p>Ejemplo 2:</p> <p>Código: 403, acceso denegado</p> <p>Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_usuario/{id_usuario}/tipo_cursos/{tipo_cursos}") Call<List<Course>> getCourses(@Path("id_usuario") String id_usuario, @Path("tipo_cursos") int tipo_curso); } public CourseInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_CURSOS) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()).build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 19. Ejemplo de petición GET

7.3 Sistema de notificaciones

Una notificación push es un paquete de datos que aparece como alerta en el dispositivo móvil de un usuario cuando no está usando la aplicación [13].

Las notificaciones en este sistema se envían de manera personalizada a cada usuario y se generan ante las siguientes situaciones:

- Nueva tarea en un curso.
- Nueva entrada en un foro.
- Cambio del tiempo límite en una tarea.
- Nuevo archivo en un recurso didáctico.
- Nuevo aviso por parte de un docente.

Como se menciona en la sección 3.3.3, FCM (Firebase Cloud Messaging) permite enviar notificaciones push desde el servidor a sus clientes. La Figura 14 muestra el proceso que sigue un paquete de datos, para ser mostrado como notificación.

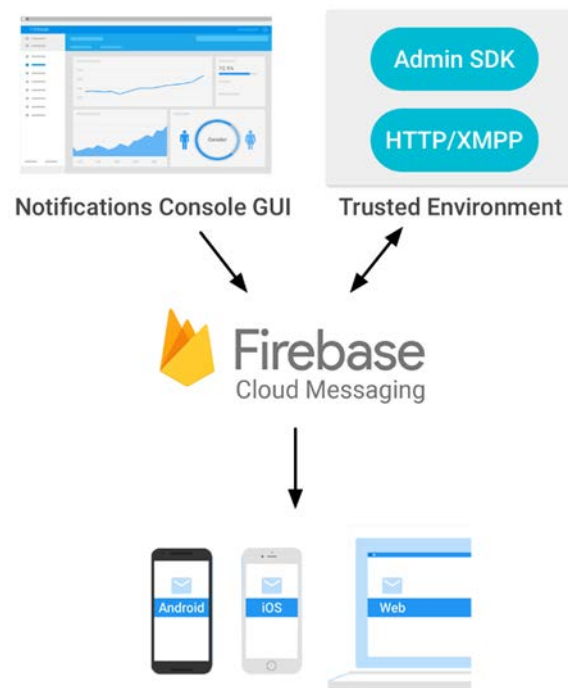


Figura 14. Proceso de FCM [14]

El servidor HTTP, pasa el paquete de datos al servidor de FCM, el cual lo envía a todos los dispositivos móviles suscritos a las notificaciones del servidor.

La suscripción de notificaciones enviadas desde el servidor, se realiza a través de Tokens, que corresponden a una clave de 52 caracteres, la cual vincula cada dispositivo con la aplicación. Un Token se obtiene a través de la función `FirebaseInstanceIdService` de FCM, que se ejecuta en el cliente. Una vez que se obtiene el Token, se debe enviar al servidor por medio de un servicio web (`A_Noti`, para este sistema), junto con la ID de usuario, para que sean registrados en la base de datos.

Si el servidor HTTP desea enviar notificaciones a ciertos tipos de usuario, busca los tokens de esos usuarios en su base de datos, y envía el mensaje al servidor de FCM, junto con los tokens.

En este proyecto, la suscripción de notificaciones se realiza automáticamente, justo después que el usuario inicia una sesión. También existe una función que monitorea el Token, de modo que si este se actualiza por orden de FCM o alguna condición del dispositivo, pueda reenviar el nuevo Token al servidor HTTP.

La implementación de lo anterior se realiza mediante las siguientes clases y se definen como servicios en la configuración de la aplicación:

- `MyFirebaseInstanceIdService`: Permite detectar el Token actualizado y retornarlo para ser usado en el servidor.
- `MyFirebaseMessagingService`: Esta clase es el controlador de los mensajes recibidos a través de las notificaciones.
- `MyJobService`: Esta clase permite pasar el recibimiento de una notificación a segundo plano, y reprogramar su envío si es que el sistema no pudo recibirla de inmediato.

A continuación, se muestra un ejemplo de la definición del servicio de `MyFirebaseInstanceIdService`:

```
<service android:name="com.ubb.adecca.Utils.Notifications.MyFirebaseInstanceIdService">
  <intent-filter>
    <action android:name="com.google.firebase.INSTANCE_ID_EVENT"/>
  </intent-filter>
</service>
```

De este modo, la aplicación puede recibir notificaciones en cualquier momento, mientras exista conexión con los servidores.

7.4 Diagrama de clases

A continuación, se muestran dos diagramas de clases, correspondientes a los componentes que implementan HU_01 y HU_02, respectivamente.

No es posible representar el sistema completo a través de un solo diagrama de clases, ya que existen catorce componentes que implementan los servicios descritos anteriormente, y alrededor de cuarenta clases que implementan el resto de historias de usuario, o están relacionadas con esas implementaciones. Sin embargo, todos los componentes que implementan servicios siguen la misma estructura que aquí se presenta, por lo tanto, los diagramas son representativos de todos esos componentes.

La Figura 15, corresponde al diagrama de clases del componente que permite ingresar a un usuario registrado (HU_01). Se puede distinguir el uso de la vista (ActivityLogin y LoginView), el presentador (LoginPresenter), y el modelo, que incluye el interactor (LoginInteractor y LoginInteractorImpl) y el servicio (LoginService).

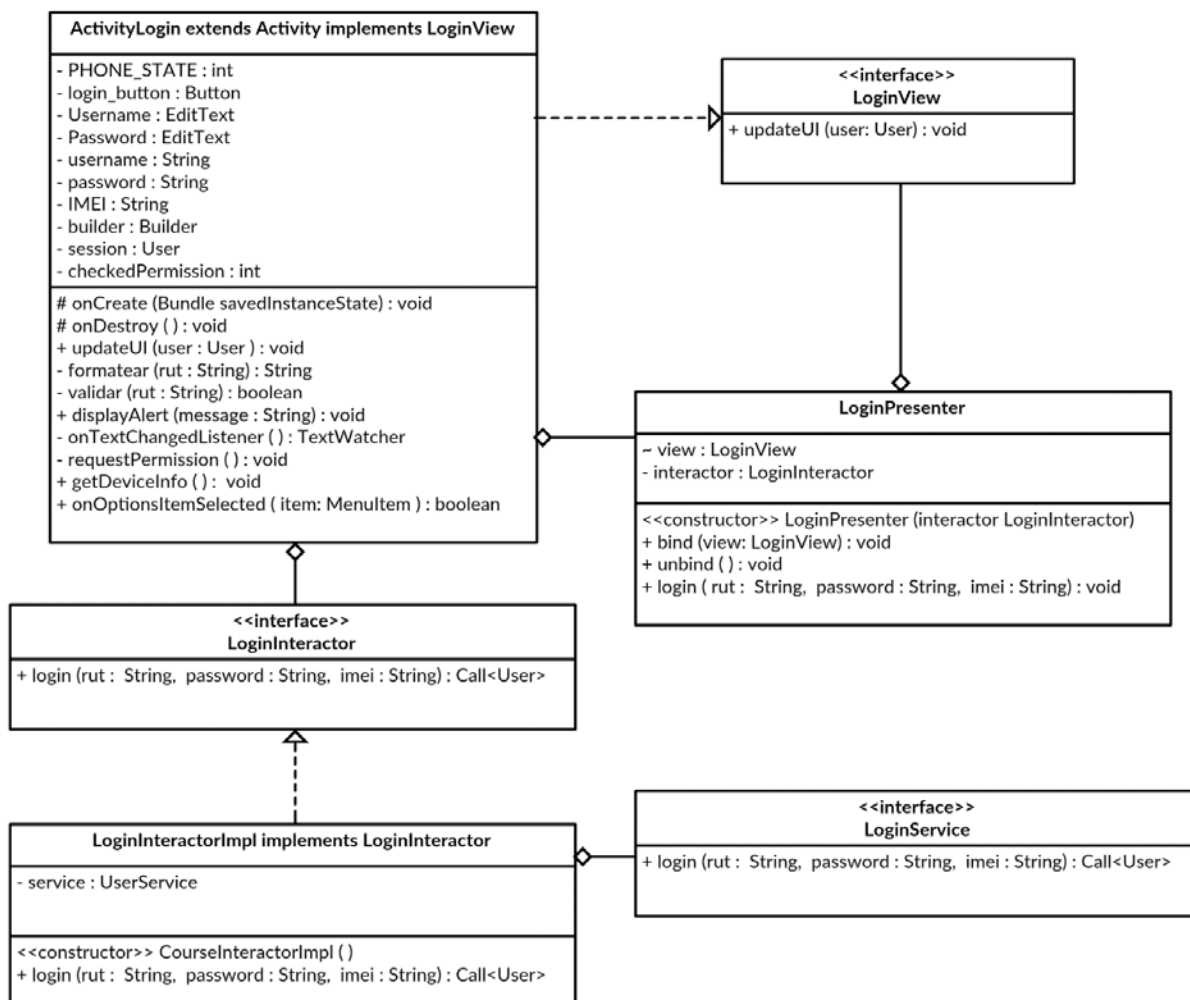


Figura 15. Diagrama de clases del componente que implementa HU_01

Por su parte, la Figura 16, corresponde al diagrama de clases del componente que permite mostrar los cursos de un usuario (HU_02). Tal como en el ejemplo anterior, se puede distinguir el uso de la vista (ActivityCourse y CourseView), el presentador (CoursePresenter), además del modelo, que incluye el interactor (CourseInteractor y CourseInteractorImpl) y el servicio (CourseService).

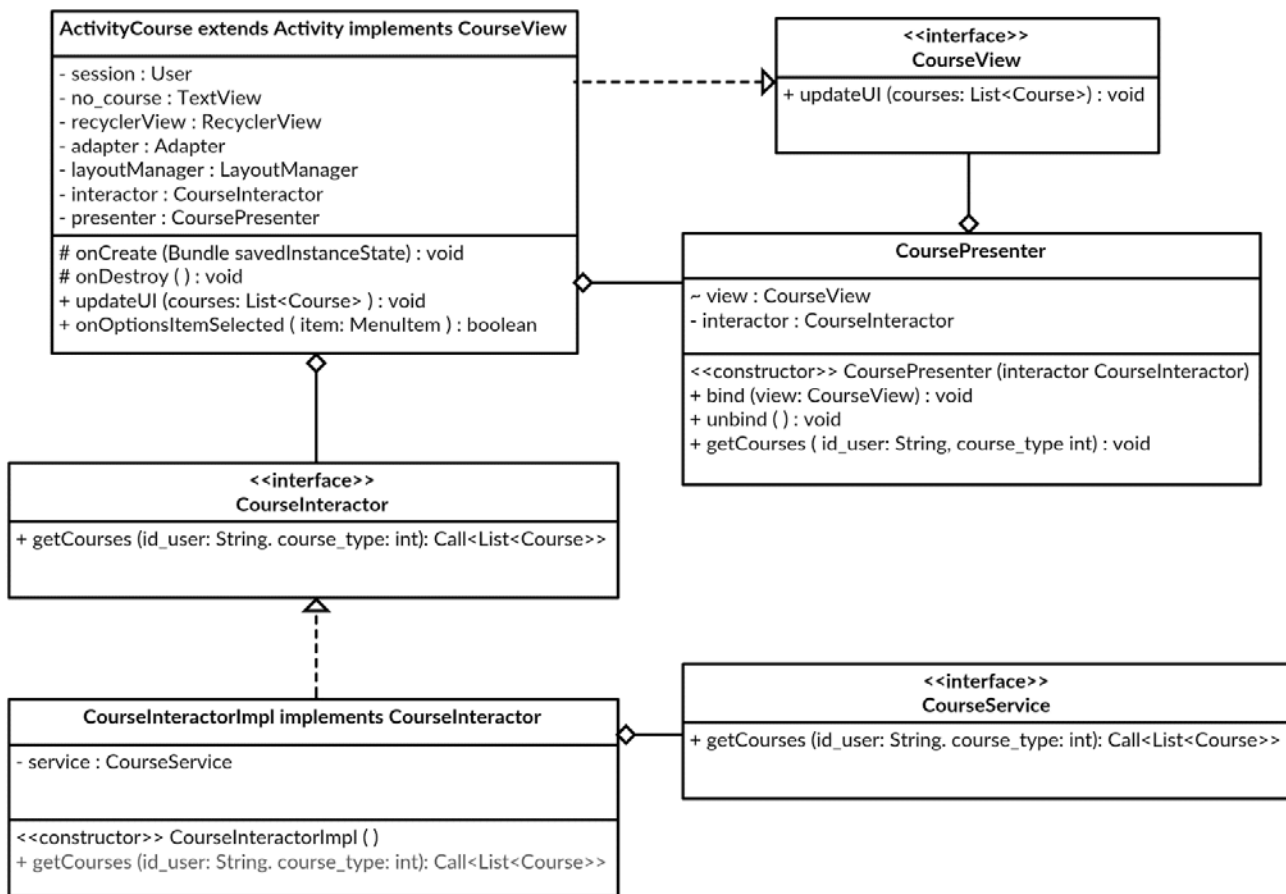


Figura 16. Diagrama de clases del componente que implementa HU_02

7.5 Funcionalidades adicionales

En el producto final existen tres funcionalidades, aparte de las especificadas en las historias de usuario, que han sido desarrolladas en una etapa posterior al término del último Sprint, durante la elaboración de este documento. Estas funcionalidades no se encuentran especificadas del mismo modo que el resto, con el objetivo de que el documento cumpla con un alto nivel de formalidad e integridad. Sin embargo, para reflejar la realización de este trabajo, a continuación se encuentra una breve explicación de cada función adicional.

Bodega de archivos personales

Esta funcionalidad muestra una lista de los archivos personales que tiene un usuario autenticado en la bodega de Adecca UBB, permitiendo la descarga de cada archivo. La obtención de los datos se realiza mediante una petición GET al servidor de Adecca UBB, usando el ID del usuario como parámetro. Se puede acceder a esta función mediante una opción en el menú principal de un usuario autenticado.

Mapas de la Universidad del Bío-Bío

Se muestra una pantalla con botones que permiten acceder al mapa de las distintas zonas de la Universidad, dichas zonas son los campus Fernando May, La Castilla y Concepción, así como del Centro de Extensión. Los mapas se muestran a través de Google Maps en la aplicación Maps, si es que está habilitada en el dispositivo. Esta funcionalidad puede ser accedida a través de una opción disponible en el menú principal de un usuario autenticado así como de uno sin autenticar.

Web UBB

Se muestra la página web de la Universidad del Bío-Bío, del mismo modo que se realiza en HU_22, por ejemplo. Esta función está disponible tanto en el menú principal de usuario autenticado como en el menú de un usuario sin autenticar.

7.6 Diseño interfaz y navegación

En esta sección se presenta la especificación de la interfaz y las jerarquías de menú de la aplicación, además de ejemplos en base a lo que se ha hecho en el sistema.

7.6.1 Esquema de especificación de interfaz

La interfaz del sistema debe seguir un patrón similar en los contenidos que muestre. Por lo general, un elemento del patrón corresponde a una barra en la parte superior de la pantalla, la cual es de color azul de acuerdo a lo indicado en la especificación de requisitos del software, y de un largo de 45 pixeles de densidad independiente (dp), como se muestra en la Figura 17. Dentro de ella debe ir el título del contenido que se desea mostrar, o el nombre de la aplicación, en el caso de las pantallas de inicio. Además, debe contener la opción de regreso o menús, si corresponden.

Otro elemento del patrón, corresponde a una segunda barra azul, compuesta de una o más opciones en forma de pestañas, que muestran distintos contenidos. La opción o pestaña escogida debe diferenciarse del resto usando un texto de color claro, para no confundir al usuario. La Figura 18 muestra un ejemplo de este elemento, junto al primero.

El contenido que se desee mostrar (cursos, perfil, webs, etc.), corresponde a la parte blanca de ambas figuras.

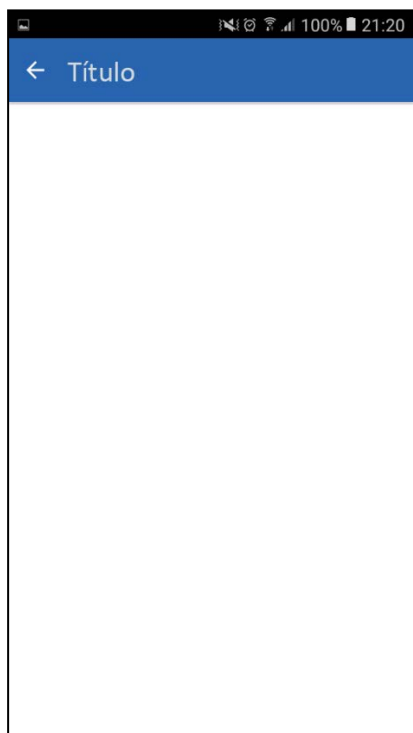


Figura 17. Ejemplo de especificación de interfaz 1

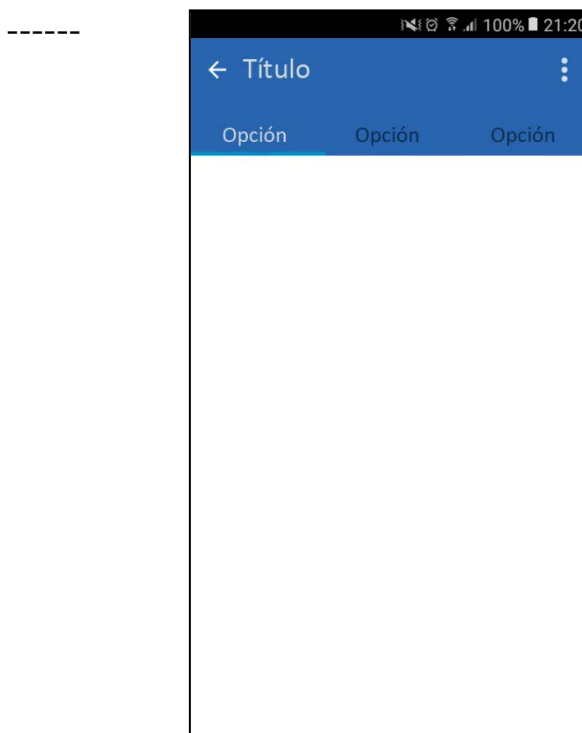


Figura 18. Ejemplo de especificación de interfaz 2

7.6.2 Jerarquías de menú

El sistema de navegación se compone de dos partes, correspondientes al área de un usuario sin autenticar y a un usuario autenticado. Se ha diseñado de este modo, para que los usuarios que no están registrados en Adecca UBB (postulantes, por ejemplo) puedan acceder a algunos de los contenidos web que ofrece la Universidad, a modo de promocionar la misma.

Cada contenido dentro del sistema debe tener una opción que permita al usuario regresar a la página anterior, ya que algunos dispositivos no tienen disponible el botón que realiza esa acción.

A continuación, la Figura 19 muestra la jerarquía de menú que corresponde a un usuario sin autenticar. Notar que entre las opciones del Menú 1 sólo hay accesos a páginas web de la Universidad del Bío-Bío, además de la opción de “Ingresar”. Esta opción permite autenticar al usuario, y se encuentra en ambos menús para que sea más accesible.

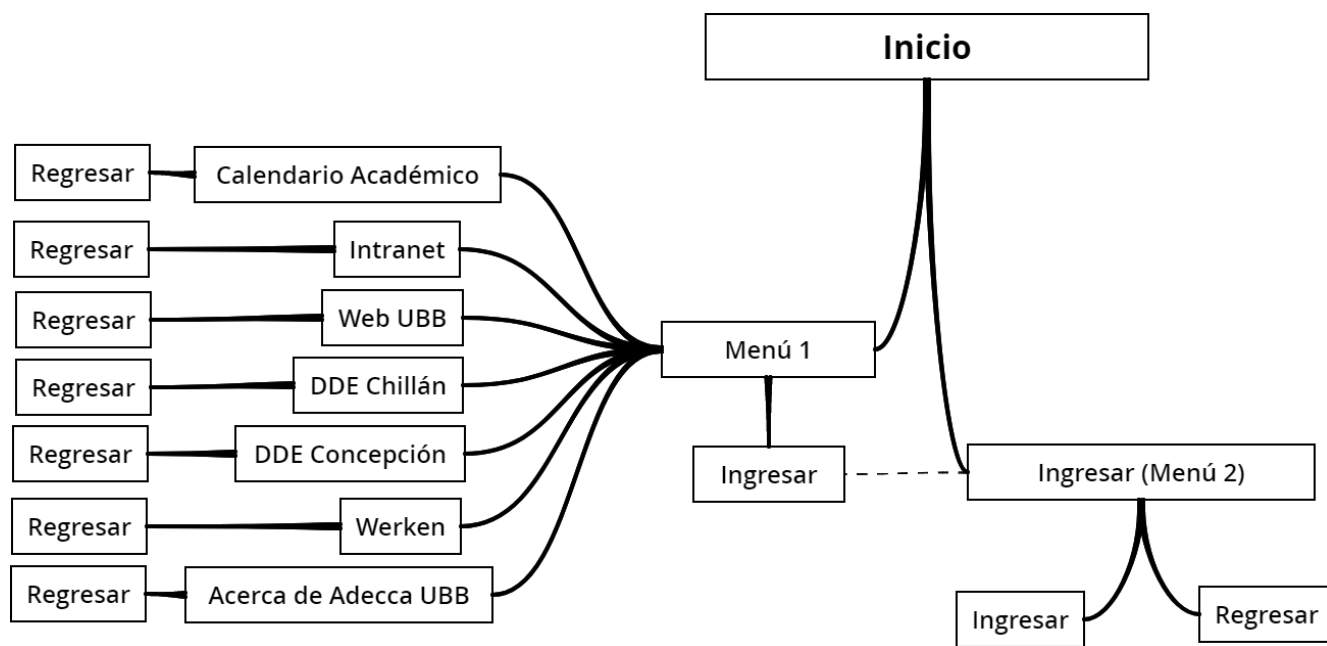


Figura 19. Jerarquía de menú 1

Las figuras a continuación, corresponden a capturas de pantalla de la aplicación, que muestran el menú en el caso de un usuario sin autenticar. El primer círculo de la Figura 20 muestra la opción que despliega el Menú 1, mientras que el segundo círculo muestra la opción que permite ir a la pantalla de inicio de sesión. La Figura 21, por su parte, muestra todas las opciones del Menú 1, tal y como se especifica en la jerarquía.

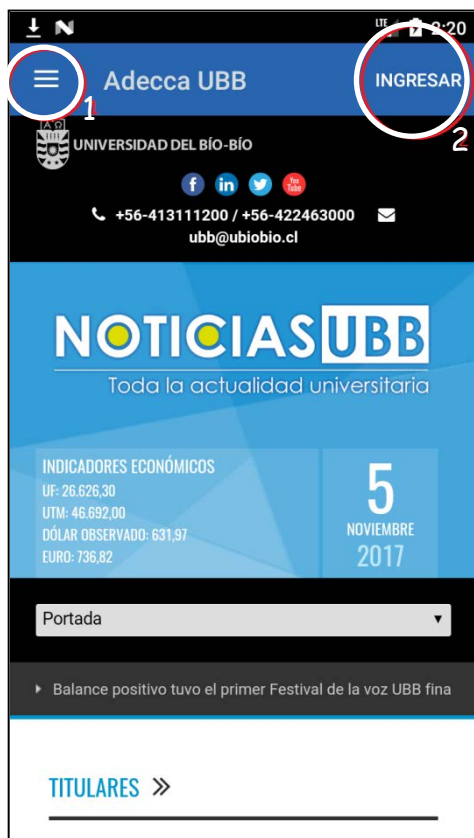


Figura 20. Pantalla de inicio de usuario sin autenticar

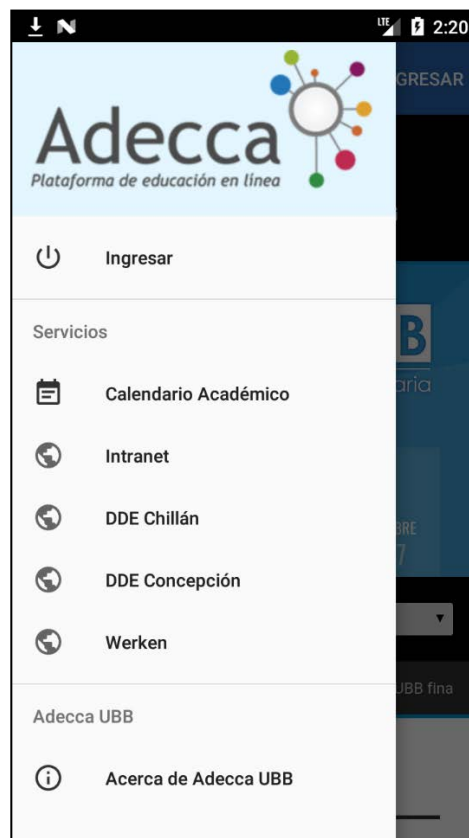


Figura 21. Menú de usuario sin autenticar

A continuación, la Figura 22 muestra la jerarquía de menú que corresponde a un usuario autenticado. El Menú 1 contiene todas las opciones que cubre esta área, pero para mejorar la accesibilidad al usuario, fuera del menú se incluye una opción que muestra los cursos de inmediato (ver mis cursos), además de una opción que permite cerrar la sesión de usuario a través del Menú 2, de este modo, no es necesario localizarlas dentro del menú 1 y se puede acceder a ellas rápidamente.

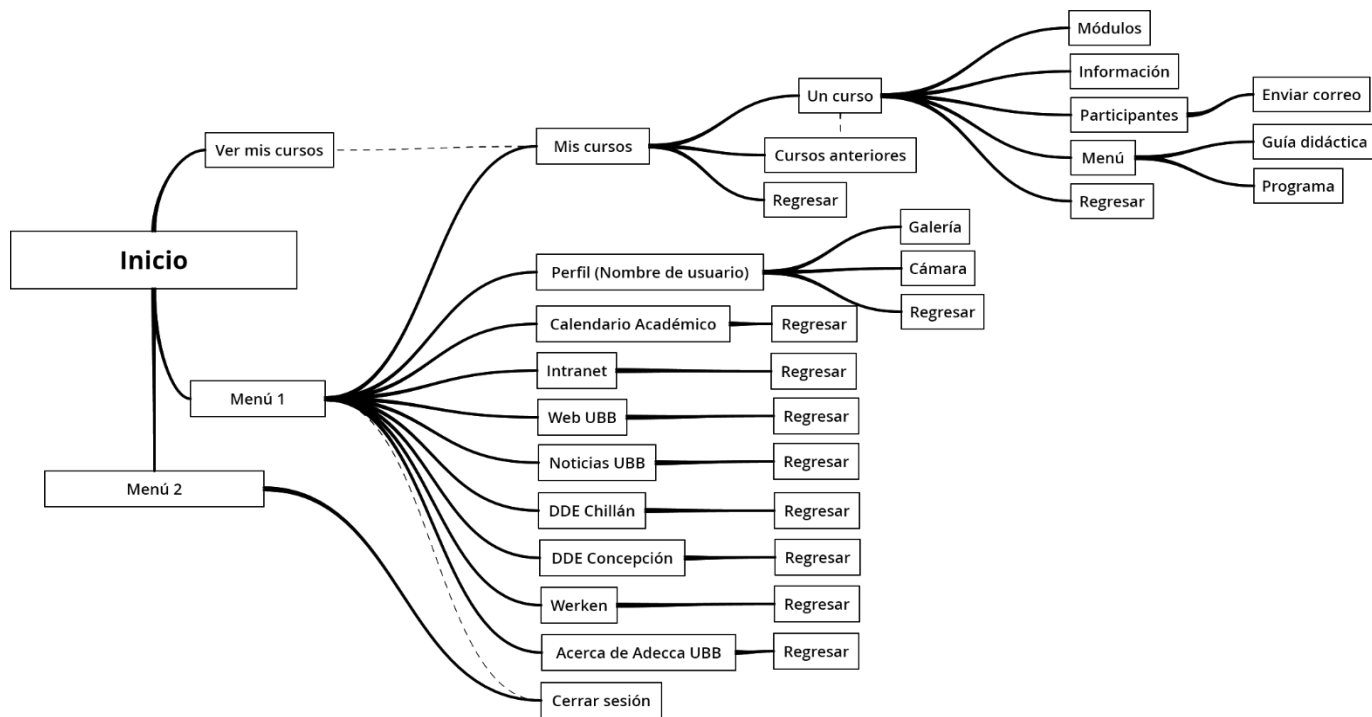


Figura 22. Jerarquía de menú 2

La Figura 23 corresponde a la pantalla de inicio del área de un usuario autenticado. El primer círculo, muestra la opción que despliega el menú principal de esta área, el cual se muestra en la Figura 24 y Figura 25. El segundo círculo, muestra la opción que permite cerrar sesión, mientras que el tercer círculo, muestra una opción para ver los cursos del semestre, sin tener que pasar por el menú principal, como se había explicado.

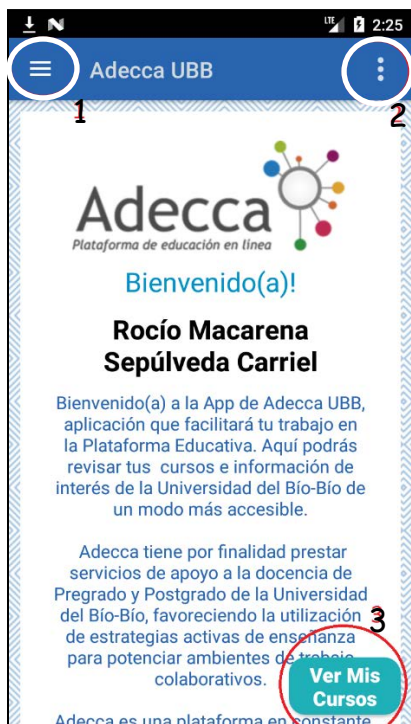


Figura 23. Pantalla de inicio de usuario autenticado

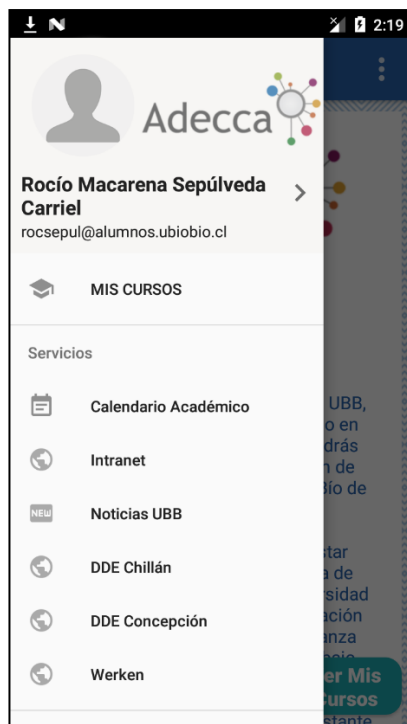


Figura 24. Menú 1, de usuario autenticado. Parte 1

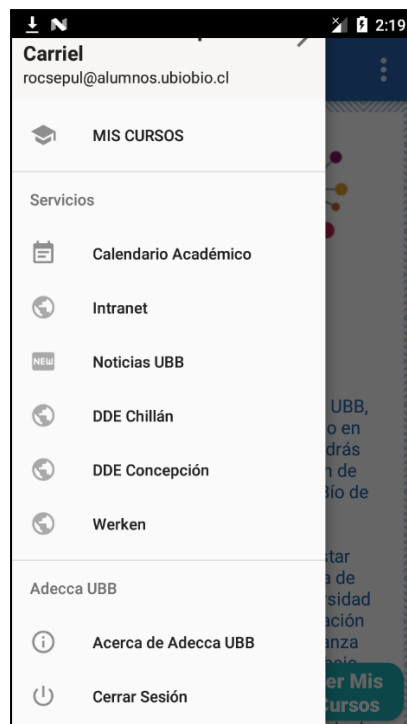


Figura 25. Menú 1, de usuario autenticado. Parte 2

8 Pruebas

“En Scrum, la calidad no es algo que un grupo de testing “prueba” al final; es algo que un equipo multifuncional de Scrum posee y construye continuamente y verifica en cada Sprint. Cada incremento de valor que se crea, se completa con un alto nivel de confianza y tiene el potencial de ser puesto en producción o enviado a clientes (...). Como resultado, se reduce sustancialmente la necesidad de cualquier ensayo tardío significativo para adherir a la calidad” [5].

8.1 Plan de pruebas

Un plan de pruebas es un documento que sistemáticamente describe el enfoque de testing, para evitar cuestionamientos durante el desarrollo que podrían afectar los plazos de entrega. En general, el plan define la cobertura del test (qué se prueba), la metodología del test (cómo se prueba), y la responsabilidad del test (quién prueba) [15].

El plan de pruebas para este proyecto se define a continuación.

8.1.1 Cobertura del test

De acuerdo a la metodología de desarrollo, se debe probar cada una de las historias de usuario contenidas en un Sprint, antes de que este finalice. Las pruebas se deben realizar por periodos; si en un periodo se encuentra un fallo bajo una determinada condición del sistema, se debe corregir el error y volver a aplicar la prueba bajo esa condición, en un siguiente periodo.

Si el proceso de pruebas y correcciones excede la fecha de término del Sprint, las historias pendientes pasan al siguiente Sprint. Por otra parte, tanto pruebas como errores deben registrarse en el Scrum Task Board, como parte de las tareas que se realizan.

8.1.2 Metodología del test

Pruebas unitarias

Se encuentran en un nivel de prueba de software donde se prueban unidades o componentes individuales de un software (pruebas de caja blanca). El propósito es validar que cada unidad del software funciona según lo diseñado [16].

En este caso, se usan pruebas unitarias para probar el modelo; por cada servicio web definido anteriormente se prueba que los datos sean recibidos y convertidos a objetos correctamente. Adicionalmente, se agregan pruebas para el formateo y validación de RUT.

Para probar los servicios, por cada uno se han definido dos casos de pruebas:

- Conexión exitosa: Por ejemplo, testCurrentCourses_ConnectionCorrect ().
- Conexión errónea: Por ejemplo, testCurrentCourses_ConnectionWrong ().

En el Anexo 3, se muestra un ejemplo de la implementación de estas pruebas.

Pruebas de sistema

Es principalmente una prueba de tipo caja negra. Esta prueba evalúa el funcionamiento del sistema desde el punto de vista del usuario. No requiere ningún conocimiento interno del sistema, como el diseño o la estructura del código. [17]

Para probar el sistema en su conjunto, los requisitos y expectativas deben ser claros y quien prueba también debe comprender el uso de la aplicación en tiempo real. Por lo tanto, al probar el sistema, puede ser útil tener una idea clara de cómo se usará la aplicación y qué tipo de problemas puede enfrentar en tiempo real. [17]

Entonces, para probar la funcionalidad que implica cada historia, se definen distintos tipos de escenarios que un usuario podría encontrarse al momento de usar esa funcionalidad. Entre los escenarios, se incluyen pruebas con sistemas Android Ice Cream Sándwich y Nougat, ya que corresponden a la menor y mayor versión de sistema operativo soportados por la aplicación. La documentación de cada prueba se realiza según el ejemplo de la Tabla 20.

La documentación de las pruebas se encuentra en el Anexo 3, por cada Sprint realizado.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_01	1	Determinar si un usuario puede iniciar sesión con un RUT cuyo dígito verificador es K	El RUT y la contraseña se encuentran en la BD y están asociados	RUT con dígito verificador K y contraseña	Se muestra la pantalla de bienvenida	Al presionar el botón "Ingresar" de la pantalla de inicio de sesión	Se muestra la pantalla de bienvenida con el nombre de usuario	Éxito

Tabla 20. Ejemplo de prueba de sistema

8.2 Pruebas funcionales

La

Tabla 21 muestra un resumen de las pruebas funcionales de sistema realizadas en un primer periodo, por historia de usuario. El resumen del segundo periodo de pruebas no se incluye, ya que todas las historias de usuario que pasaron a esta etapa tuvieron un porcentaje de éxito del 100%.

	Historia de Usuario	N° de escenarios probados	N° de escenarios exitosos	Porcentaje de éxito
Sprint 1	HU_01	9	8	88,8%
	HU_02	5	5	100%
	HU_03	5	2	40%
	HU_04	5	5	100%
	HU_05	5	5	100%
Sprint 2	HU_06	5	5	100%
	HU_07	6	5	83%
	HU_08	4	4	100%
	HU_09	6	6	100%
	HU_10	5	3	60%
	HU_11	5	3	60%
	HU_12	1	1	100%
Sprint 3	HU_13	5	5	100%
	HU_14	5	5	100%
	HU_15	4	3	75%
	HU_16	4	3	75%
Sprint 4	HU_17	5	4	80%
	HU_18	4	4	100%
	HU_19	3	3	100%
	HU_20	4	3	75%
	HU_21	1	1	100%
	HU_22	1	1	100%
	HU_23	2	2	100%
	HU_24	1	1	100%
Sprint 5	HU_25	4	3	75%
	HU_26	3	3	100%
	HU_27	3	3	100%
	HU_28	3	3	100%
Sprint 6	HU_29	3	3	100%
	HU_30	3	3	100%
	HU_31	4	3	75%
	HU_32	1	1	100%
	HU_33	1	1	100%
Porcentaje de éxito en promedio				90%

Tabla 21. Resumen de pruebas de sistema

Los fallos ocurridos en las pruebas se deben a la incorrecta validación de datos en las vistas del sistema, por no considerar ciertas situaciones al momento de desarrollar, como lo es verificar la conexión a Internet. Por su parte, las correcciones consistieron en aplicar las validaciones correspondientes, lo cual tomó poco tiempo. Así, tanto pruebas como correcciones se realizaron dentro la duración de cada Sprint.

En el caso de los fallos graves, la corrección fue más compleja, sin embargo, tampoco se excedió el tiempo límite del Sprint involucrado.

Según los resultados mostrados en la Tabla 21, las pruebas en las primeras historias de usuario fracasaron más que en las últimas, tanto en frecuencia como en cantidad de fallos. Por otra parte, todos los Sprints tuvieron al menos 1 falla, debiendo aplicar el segundo periodo de pruebas en cada uno de ellos.

8.3 Pruebas no funcionales

A continuación, se presentan los elementos y pruebas que permiten determinar el cumplimiento de los requisitos no funcionales, estipulados en la sección 4.4.2.

8.3.1 Usabilidad

La aplicación utiliza en su interfaz los colores correspondientes al Manual de Comunicación Corporativa de la Universidad del Bío-Bío, con ligeras variaciones en las tonalidades. Por otra parte, ha sido catalogada como “atractiva”, por la encargada de Coordinación Pedagógica y el encargado de Coordinación Técnica del Departamento de Servicios Tecnológicos de la Universidad. Lo anterior implica que el requisito se ha cumplido.

8.3.2 Portabilidad

De acuerdo a las distintas pruebas presentadas en el Anexo 4, la aplicación funciona correctamente en los dispositivos con sistema Android Ice Cream Sandwich (versión 4.0.3), por lo tanto, este requisito no funcional está cubierto.

8.3.3 Seguridad

El control de acceso, como parte de la seguridad, queda implementado a través de HU_01, que permite que los datos sean accedidos sólo por usuarios autenticados con las credenciales de Adecca UBB. Por otra parte, todas las peticiones realizadas al servidor incluyen en la cabecera una clave secreta del servidor de Adecca UBB, evitando que los datos sean accedidos por terceros. Además, los datos más sensibles se envían codificados para aumentar la seguridad. Así, se evidencia que el requisito de seguridad se ha cumplido.

8.3.4 Eficiencia

En este punto, se mide el tiempo que tarda la aplicación en mostrar los datos desde el servidor, a través de distintas pruebas. La primera prueba consiste en acceder al servicio Q_Info definido en la sección 7.1.2, seleccionando el curso con más contenido en la lista de los cursos del usuario. La segunda prueba consiste en

acceder al servicio A_Perfil ya definido, ingresando a la pantalla de edición del perfil de usuario, y rellenando los campos de alias, web, dirección y teléfono con los mismos datos para cada prueba. Se usan estos servicios porque pueden requerir más tiempo que el resto, debido al tamaño de los datos que obtienen o envían.

Todas las pruebas se realizan en los Laboratorios de Computación del Campus Fernando May, usando la red Wi-Fi de la Universidad, y la red privada del dispositivo. Además, se prueba en distintos horarios (8:00, 11:00, 16:00 y 18:00 aproximadamente). Esto, con el objetivo de determinar el tiempo de carga de los datos bajo distintas condiciones que se dan dentro de la Universidad.

De acuerdo al resultado de las pruebas, el tiempo de carga por cada operación es menor a 15 segundos, tal como lo solicita el requisito. Alrededor de las 11:30 horas el tiempo de carga es mayor, ya que es el horario en que más docentes y estudiantes utilizan el servidor de Adecca UBB, por lo tanto el tiempo para todas las peticiones aumenta.

Por su parte, en el caso de la red pública, el tiempo es aún mayor, debido a que gran cantidad de estudiantes se reúne alrededor del mediodía en la Universidad para hacer tareas por Internet¹⁰.

Teniendo en consideración lo anterior, el requisito no funcional de eficiencia se ha cumplido para condiciones externas usuales.

8.4 Conclusiones de Pruebas

El hecho de que las historias de usuario tengan un 90% de éxito en el primer periodo de pruebas, indica que el sistema fue desarrollado casi correctamente. Esto se debe al uso de pruebas unitarias para probar el modelo, además de la capacidad del equipo de desarrollo para pensar en los distintos eventos que podrían ocurrir a un usuario. Otro resultado positivo es que la planificación de los Sprints permitió realizar pruebas y corregir fallos dentro del tiempo establecido, evitando el envío de historias de usuario a un siguiente Sprint. También, el uso del plan de pruebas fue beneficioso para este proyecto, porque permitió probar de manera expedita y fortalecer la calidad del sistema.

Los fallos que se presentan en mayor cantidad en las primeras historias de usuario, suceden por la poca familiaridad con el lenguaje de Android, situación que se mejora a medida que se avanza en el código. Por otra parte, los fallos por validaciones indican que el desarrollo de la vista no estuvo al mismo nivel que el desarrollo de los presentadores o modelos.

Se puede concluir entonces, que el uso de pruebas unitarias junto a una buena planificación y plan de pruebas, permite mejorar la calidad del sistema, por lo tanto, son elementos necesarios al momento de realizar un proyecto de software. En contraste, para disminuir los fallos es conveniente dar la misma prioridad a todos los elementos del sistema, usando pruebas unitarias tanto para modelos, como vistas y presentadores. Además, los fallos

¹⁰ Según el encargado de los Laboratorios Centrales de Computación de la Universidad del Bío-Bío.

producidos por la falta de experiencia en Android, se disminuyen mediante la capacitación del equipo de desarrollo antes del inicio del proyecto.

9 Conclusiones

Scrum fue una buena elección al momento de escoger la metodología de desarrollo, puesto que con tiempo reducido, y poca experiencia del equipo de desarrollo, se logró finalizar el proyecto a tiempo, cumpliendo con todos los requisitos y preferencias del Product Owner. Además, no se retrasó el proyecto tratando de aplicar esta metodología, dado que las técnicas y herramientas que se usan en Scrum son fáciles de comprender y acceder. Por otra parte, el hecho de que los requisitos no cambiaron durante el desarrollo implicó una ventaja que ayudó a terminar el proyecto a tiempo.

La planificación de Sprints se realizó de manera exitosa, ya que no hubo atrasos en los entregables del proyecto. Si bien no hubo holguras en el tiempo de desarrollo, se pudo construir el sistema y probarlo tal como se esperaba. Es necesario mencionar el importante papel que jugaron los puntos de historia, ya que por ellos se pudo determinar una duración acertada de cada Sprint.

En el caso de la arquitectura usada, el Modelo-Vista-Presentador fue una excelente opción, ya que permitió aplicar la arquitectura limpia y realizar pruebas unitarias sin modificar el código. El mayor desafío en este proyecto fue aplicar y comprender las arquitecturas usadas, puesto que no se tenía conocimiento de ellas antes de comenzar el proyecto. Aparte, las definiciones, explicaciones, y ejemplos de las arquitecturas varían dependiendo de la fuente, provocando confusión.

Aunque las pruebas en el sistema arrojaron resultados positivos, para minimizar errores y aumentar la calidad del producto en los futuros cambios que se realicen, se podría hacer un plan de pruebas donde la cobertura de las pruebas unitarias sea mayor, o aplicar desarrollo guiado por pruebas de software, por ejemplo.

Considerando lo anterior, se comprueba que Scrum como metodología ágil ayuda en la construcción exitosa en los proyectos. Además, es comprensible que el uso de puntos de historia sea una técnica muy usada en las metodologías ágiles y, sin duda, debería ser tratada cada vez que se trabaje en un proyecto de este tipo. Otra observación importante, es que usar arquitectura limpia es una excelente práctica que debe ser considerada al momento de desarrollar, puesto que ayuda a mejorar significativamente la calidad del software.

Personalmente, esta ha sido una experiencia enriquecedora, puesto que se tenía poco conocimiento sobre desarrollo en Android y un total desconocimiento sobre arquitectura limpia, MVP y REST. Por otra parte, aunque ya se había trabajado con Scrum en proyectos anteriores, en esta oportunidad se siguió la metodología de manera más rigurosa, pudiendo comprender mejor cómo operan las empresas en el mercado actual. Como última observación, el trabajo intensivo y dedicado otorga buenos resultados.

Finalmente, se concluye que el proyecto fue realizado de manera exitosa, cumpliendo con los objetivos definidos, los requerimientos del cliente, y las expectativas personales.

Bibliografía

- [1] Universidad del Bío-Bío, «UBB,» [En línea]. Available: http://www.ubiobio.cl/w/#Vision_y_Mision.
- [2] Universidad del Bío-Bío, «Departamento de Servicios Tecnológicos,» [En línea]. Available: <http://www.ubiobio.cl/dst/>.
- [3] Universidad del Bío-Bío, «Adecca UBB,» 2017. [En línea]. Available: <http://adecca.ubiobio.cl/>.
- [4] E. Martínez, «IEBS School,» 7 Febrero 2014. [En línea]. Available: <http://www.iebschool.com/blog/que-es-agile-agile-scrum/>.
- [5] K. S. Rubin, «Essential Scrum,» de *Essential Scrum*, Michigan, Addison-Wesley, 2012, p. 83.
- [6] G. Ríos, «Modelo Entidad Relación,» 2007. [En línea]. Available: http://www.belgrano.esc.edu.ar/matestudio/carpeta_de_access_introduccion.pdf.
- [7] Lucid Software Inc., «Lucidchart,» 2017. [En línea]. Available: <https://www.lucidchart.com/pages/es/qu%C3%A9-es-un-diagrama-entidad-relaci%C3%B3n>.
- [8] Quassnoi, «Explain Extended,» 18 Octubre 2009. [En línea]. Available: <https://explainextended.com/2009/10/18/what-is-entity-relationship-model/>.
- [9] R. Martin, «8th Light,» 13 Agosto 2012. [En línea]. Available: <https://8thlight.com/blog/uncle-bob/2012/08/13/the-clean-architecture.html>.
- [10] BBVAOPEN4U, «BBVA Open4U,» 23 Marzo 2016. [En línea]. Available: <https://bbvaopen4u.com/es/actualidad/api-rest-que-es-y-cuales-son-sus-ventajas-en-el-desarrollo-de-proyectos>.
- [11] Universidad del Bío-Bío, «Dirección General De Comunicación Estratégica,» 2014. [En línea]. Available: http://www.ubiobio.cl/mcc/papeleria/Manual_de_Comunicacion_Corporativa.pdf.
- [12] D. Kocurek, «ScrumDesk,» 16 Noviembre 2011. [En línea]. Available: <https://www.scrumdesk.com/effort-vs-time/>.
- [13] P. Leddy, «Pulsate HQ,» [En línea]. Available: <http://academy.pulsatehq.com/push-notifications-explained>.
- [14] Firebase, «Firebase,» 17 Octubre 2017. [En línea]. Available: <https://firebase.google.com/docs/cloud-messaging/?hl=es-419>.
- [15] XBOSoft Inc., «Scrum Test Planning White Paper,» Santa Clara.
- [16] Software Testing Fundamentals, «Software Testing Fundamentals,» [En línea]. Available: <http://softwaretestingfundamentals.com/unit-testing/>.
- [17] B. Mehta, «Software Testing Help,» 28 Octubre 2017. [En línea]. Available: <http://www.softwaretestinghelp.com/system-testing/>.
- [18] Wikipedia, «Wikipedia,» [En línea]. Available: https://es.wikipedia.org/wiki/Universidad_del_B%C3%ADo-B%C3%ADo.
- [19] Corporación Universidad de Concepción, «UDEEC,» Julio 2012. [En línea]. Available: <http://www.udec.cl/memoria2011/>.
- [20] J. F. Echeverría, «AméricaEconomía Intelligence,» 2016. [En línea]. Available: <https://rankings.americaeconomia.com/2016/universidades-chile/ranking>.
- [21] OBS Business School, «OBS Business School,» 14 Febrero 2014. [En línea]. Available: <http://www.obs-edu.com/int/blog-project-management/diagramas-de-gantt/que-es-un-diagrama-de-gantt-y-para-que-sirve>.

Anexo 1: Factibilidad

1.1 Factibilidad técnica

Especificaciones del Equipo	
ID	25572
Marca	Asus
Modelo	X540SA-XX416T
Color	Negro Chocolate
Especificaciones del Procesador	
Procesador	INTEL
Línea Procesador	INTEL Pentium
Modelo Procesador	N3710
Velocidad Procesador	1.60 GHz hasta 2,56 GHz
Cantidad Núcleos	Quad Core
Especificaciones de Memoria	
Memoria RAM	8GB
Memoria máxima soportada	8GB
Velocidad de Memoria	DDR3
Especificaciones Almacenamiento	
Almacenamiento	500GB
Almacenamiento adicional	No incluye
Especificaciones de Pantalla	
Tamaño Pantalla	15.6"
Resolución Máxima	1366x768
Touch	No
Especificaciones Gráficas	
Tarjeta Video	Integrada
Marca Tarjeta Video	INTEL
Modelo de Tarjeta de Video	HD Graphics 405
Sistemas Operativos	
Sistema Operativo	Windows®
Versión	10

Tabla 22. Ejemplo de características técnicas de un computador para desarrollo de software

1.2 Factibilidad económica

1.2.1 Pruebas de tiempo

Prueba 1	Tiempo que tarda un docente en responder la entrada de un foro.
Profesor en prueba	Melandro Serrano Gutiérrez.
Condiciones de prueba	Computador encendido, red conectada, navegador cerrado sin datos de IP ni sesión registrados en el navegador.
Resultado	53,25 segundos.

Tabla 23. Prueba 1, tiempo que tarda un docente en responder la entrada de un foro

Prueba 2	Tiempo que tarda un estudiante en responder la entrada de un foro.
Estudiante en prueba	Rocío Sepúlveda Carriel
Condiciones de prueba	A 30 metros del computador aproximadamente, sin fila para solicitar el computador, computador encendido, red conectada, navegador cerrado sin datos de IP ni sesión registrados en el navegador.
Resultado	143 segundos.

Tabla 24. Prueba 2, tiempo que tarda un estudiante en responder la entrada de un foro

1.3 Formato de encuesta aplicada a docentes

Uso de Adecca UBB en smartphones

La siguiente encuesta tiene por objetivo recopilar información acerca del uso de Adecca UBB a través de los smartphones, por parte de los docentes de la Universidad del Bío-Bío en Chillán.

*Obligatorio

1. **¿Qué tipo de smartphone usa? ***

Selecciona todos los que correspondan.

- Android
- iPhone
- No uso smartphone

2. **¿Utiliza la plataforma Adecca UBB? ***

Marca solo un óvalo.

- Sí
- No

3. **¿En promedio, cuántas veces a la semana accede a la plataforma desde un computador de la Universidad?**

4. **¿Ha intentado usar Adecca UBB desde un dispositivo móvil?**

Marca solo un óvalo.

- Sí
- No

5. **¿Para qué?**

6. **¿Le gustaría que hubiera una aplicación para acceder a Adecca UBB desde su smartphone?**

Marca solo un óvalo.

- Sí

7. **¿Qué funcionalidades le gustaría que tuviera la aplicación?**

Marca solo un óvalo.

- Ver cursos
- Descargar archivos
- Enviar correos
- Revisar y contestar el foro
- Otro: _____

1.3.1 Resultados de encuesta aplicada a docentes

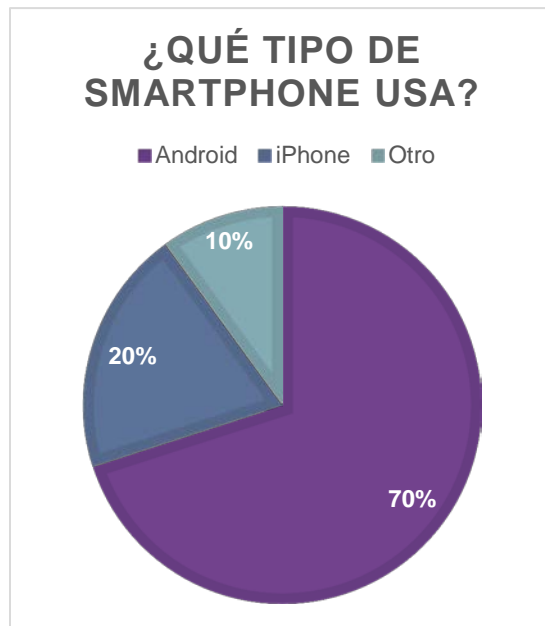


Figura 26. Gráfico de respuestas a pregunta 1, encuesta a docentes

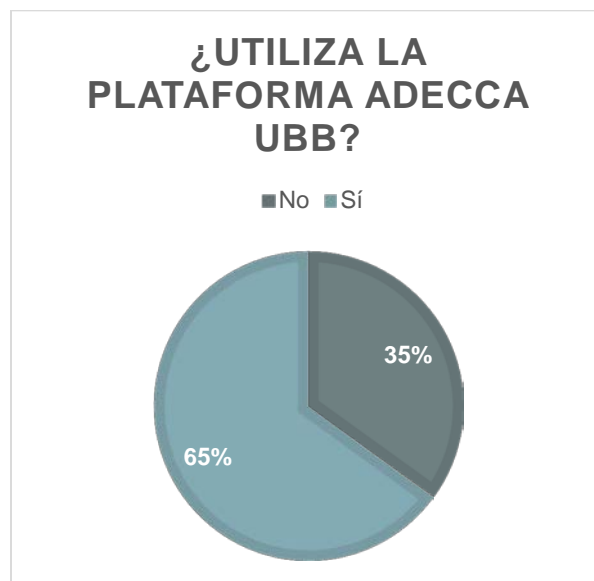


Figura 27. Gráfico de respuestas a pregunta 2, encuesta a docentes

A continuación, se muestran los resultados de quienes contestaron sí a la segunda pregunta.



Figura 28. Gráfico de respuestas a pregunta 3, encuesta a docentes

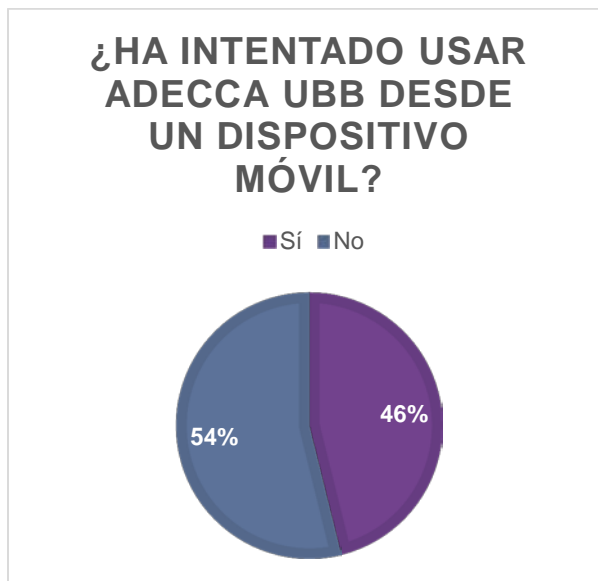


Figura 29. Gráfico de respuestas a pregunta 4, encuesta a docentes



Figura 30. Gráfico de respuestas a pregunta 5, encuesta a docentes

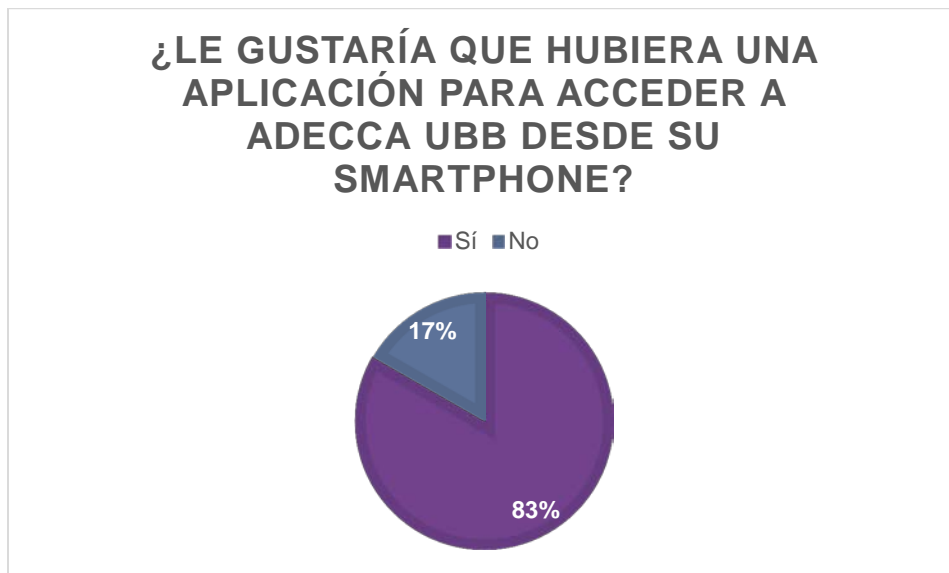


Figura 31. Gráfico de respuestas a pregunta 6, encuesta a docentes

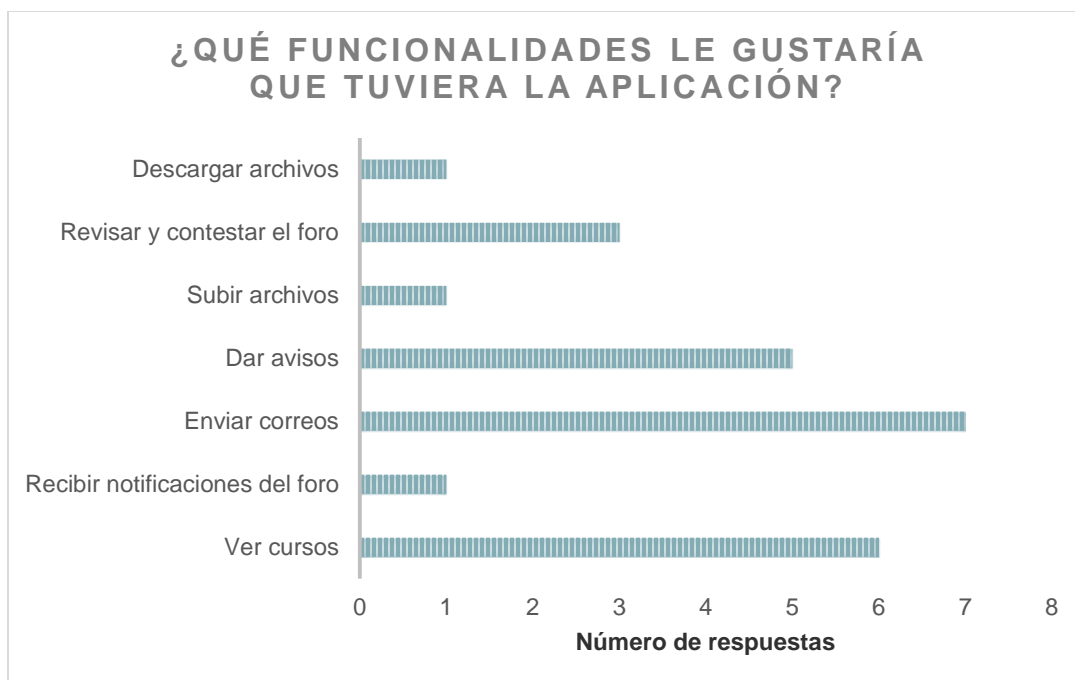


Figura 32. Gráfico de respuestas a pregunta 7, encuesta a docentes

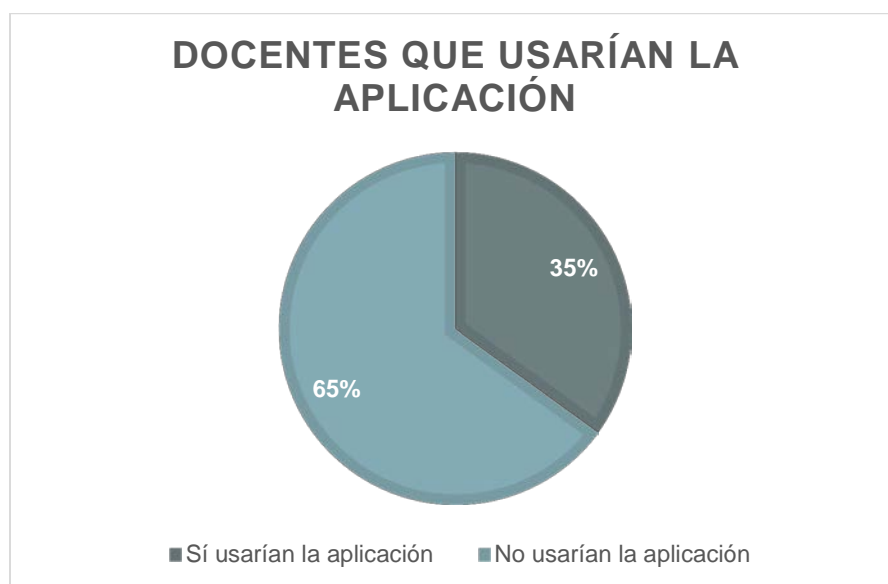


Figura 33. Gráfico de respuestas a pregunta 8, encuesta a docentes

1.4 Formato de encuesta aplicada a estudiantes

Uso de Adecca UBB en smartphones

La siguiente encuesta tiene por objetivo recopilar información acerca del uso de Adecca UBB a través de los smartphones, por parte de los estudiantes de pregrado de la Universidad del Bío-Bío en Chillán.

*Obligatorio

1. ¿A qué carrera perteneces?

Marca solo un óvalo.

- Bachillerato en Ciencias
- Contador Público y Auditor
- Diseño Gráfico
- Enfermería
- Fonoaudiología
- Ingeniería Civil en Informática
- Ingeniería Comercial
- Ingeniería en Alimentos
- Ingeniería en Recursos Naturales
- Nutrición y Dietética
- Pedagogía en Castellano y Comunicación
- Pedagogía en Ciencias Naturales
- Pedagogía en Educación Física
- Pedagogía en Educación General Básica
- Pedagogía en Educación Matemática
- Pedagogía en Educación Parvularia
- Pedagogía en Historia y Geografía
- Pedagogía en Inglés
- Psicología
- Trabajo Social

2. ¿Qué tipo de smartphone usas? *

Selecciona todos los que correspondan.

- Android
- iPhone
- No uso smartphone

3. ¿Utilizas la plataforma Adecca UBB con frecuencia? *

Marca solo un óvalo.

- Sí
- No

4. **¿Cuántas veces al mes accedes a Adecca desde un computador de la Universidad?**

5. **¿Has intentado usar Adecca UBB desde un dispositivo móvil?**

Marca solo un óvalo.

- Sí
 No

6. **¿Para qué?**

7. **¿Te gustaría que hubiera una aplicación para acceder a Adecca UBB desde tu smartphone?**

Marca solo un óvalo.

- Sí
 No

8. **¿Qué funcionalidades te gustaría que tuviera la aplicación?**

Marca solo un óvalo.

- Ver cursos
 Descargar archivos
 Enviar correos
 Revisar y contestar el foro
 Otro: _____
- Pedagogía en Educación Parvularia
 Pedagogía en Historia y Geografía
 Pedagogía en Inglés
 Psicología
 Trabajo Social

2. **¿Qué tipo de smartphone usas? ***

Selecciona todos los que correspondan.

- Android
 iPhone
 No uso smartphone

3. **¿Utilizas la plataforma Adecca UBB con frecuencia? ***

Marca solo un óvalo.

- Sí
 No

1.4.1 Resultado de encuesta aplicada a estudiantes

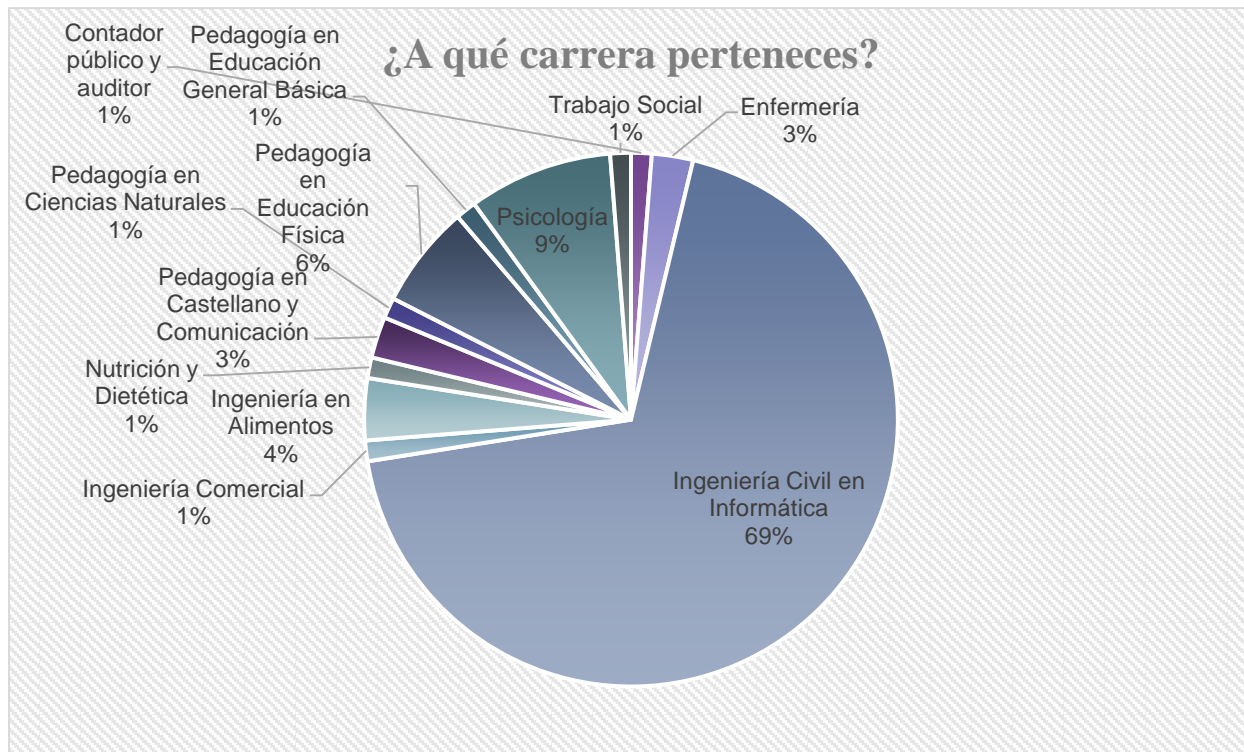


Figura 34. Gráfico de respuestas a pregunta 1, encuesta a estudiantes

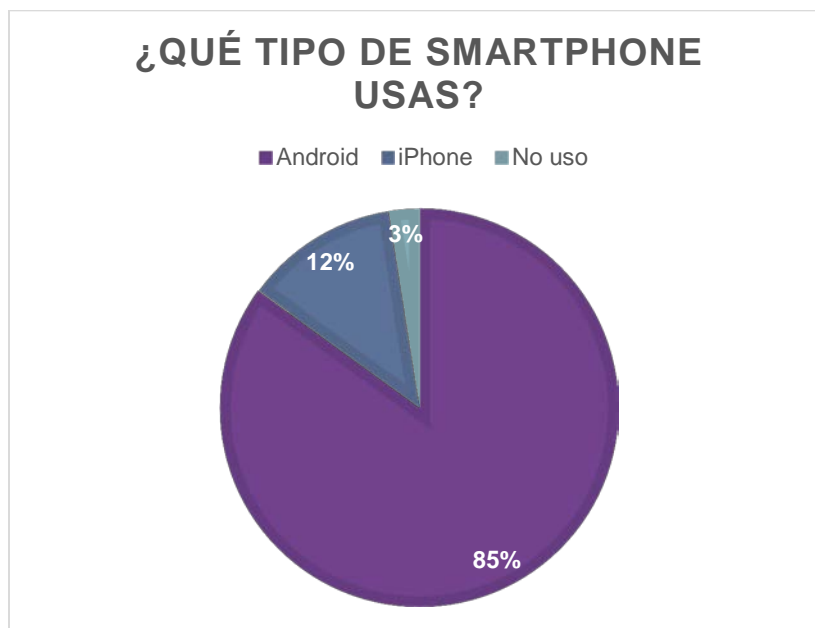


Figura 35. Gráfico de respuestas a pregunta 2, encuesta a estudiantes

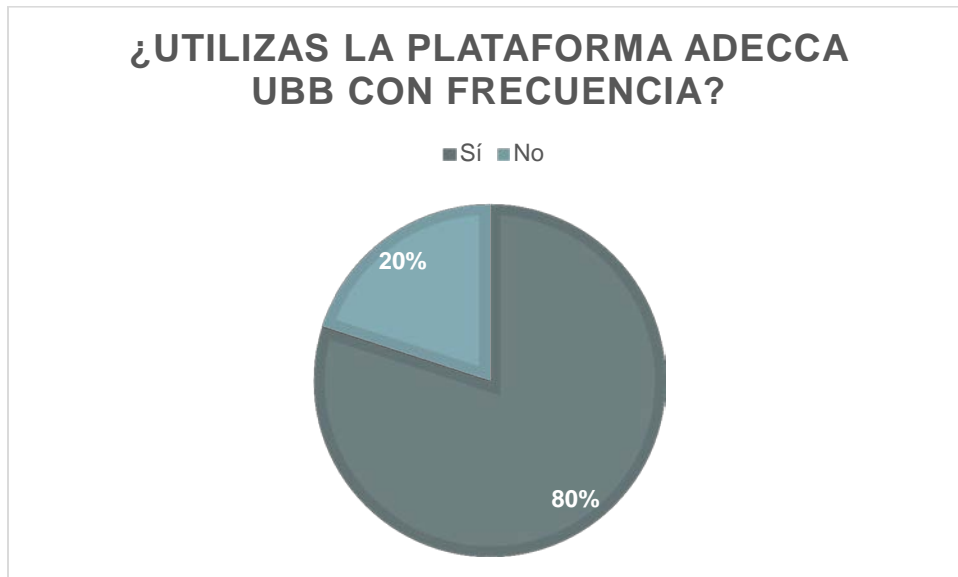


Figura 36. Gráfico de respuestas a pregunta 3, encuesta a estudiantes

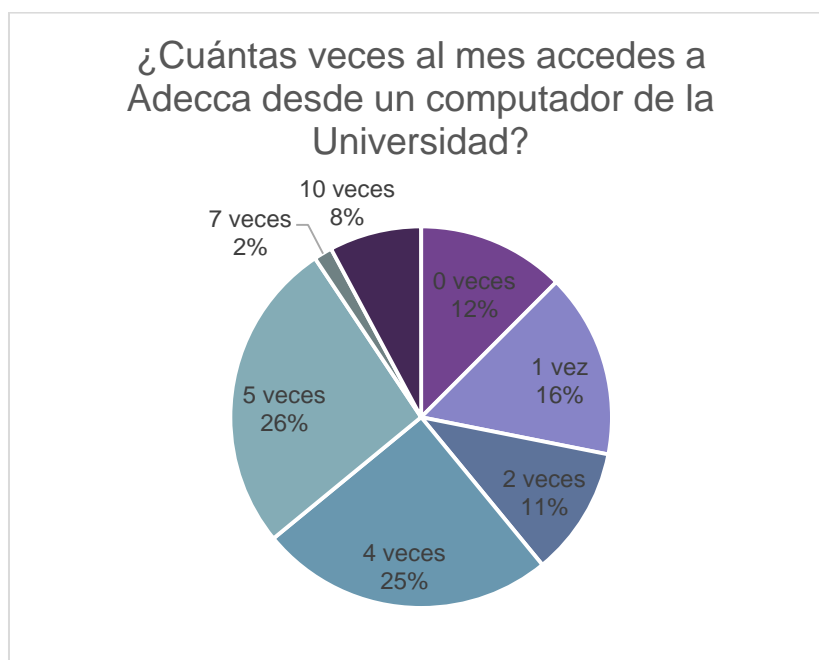


Figura 37. Gráfico de respuestas a pregunta 4, encuesta a estudiantes

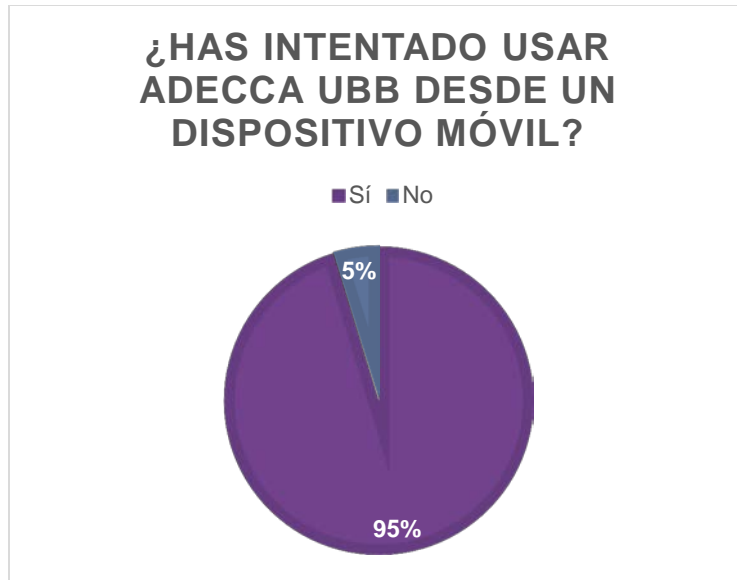


Figura 38. Gráfico de respuestas a pregunta 5, encuesta a estudiantes



Figura 39. Gráfico de respuestas a pregunta 6, encuesta a estudiantes



Figura 40. Gráfico de respuestas a pregunta 7, encuesta a estudiantes

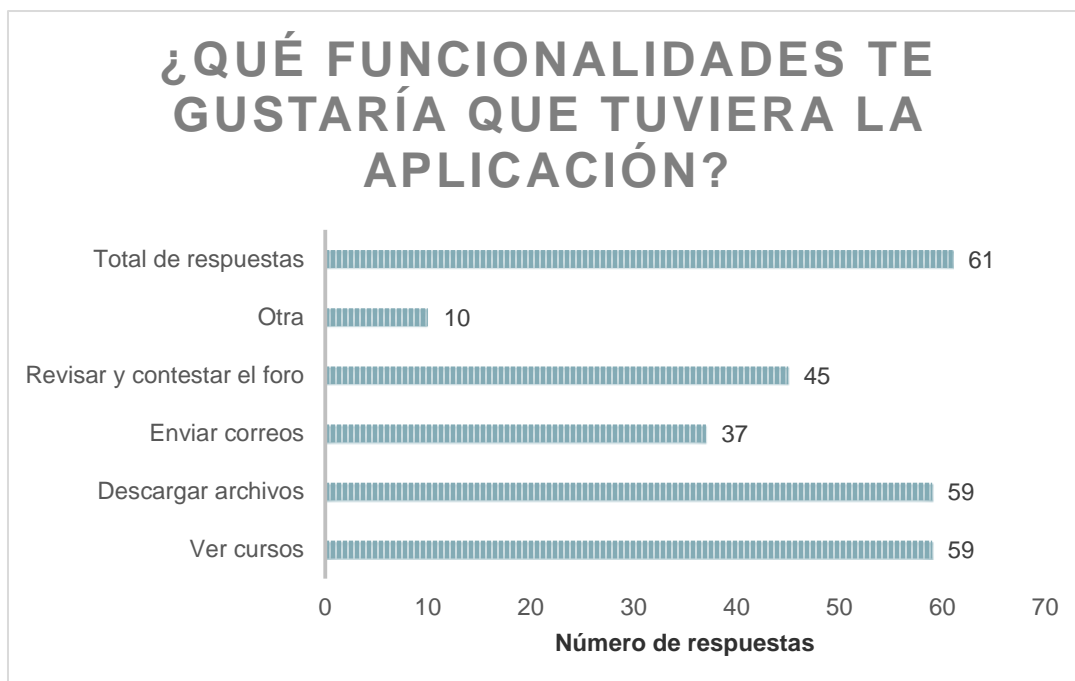


Figura 41. Gráfico de respuestas a pregunta 8, encuesta a estudiantes

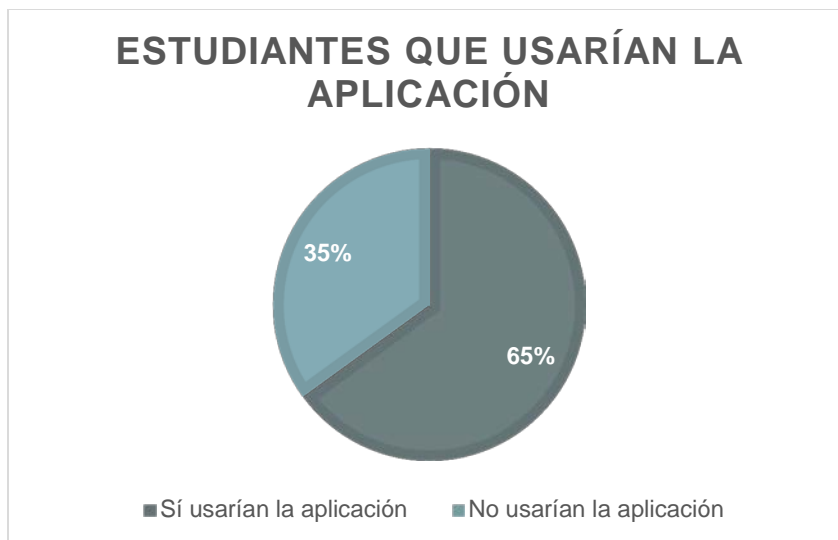


Figura 42. Gráfico de respuestas a pregunta 9, encuesta a estudiantes

1.5 Resumen de encuestas aplicadas

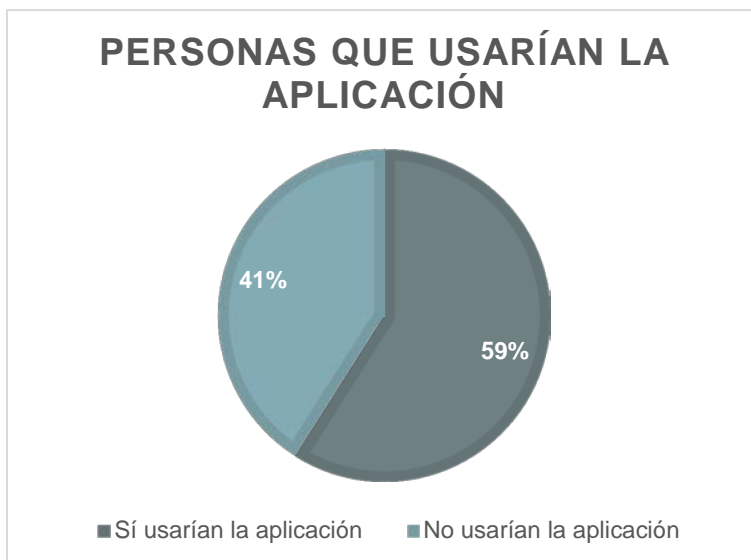


Figura 43. Gráfico resumen



Figura 44. Gráfico resumen

Anexo 2: Diseño

2.1 Ejemplos de Adecca UBB

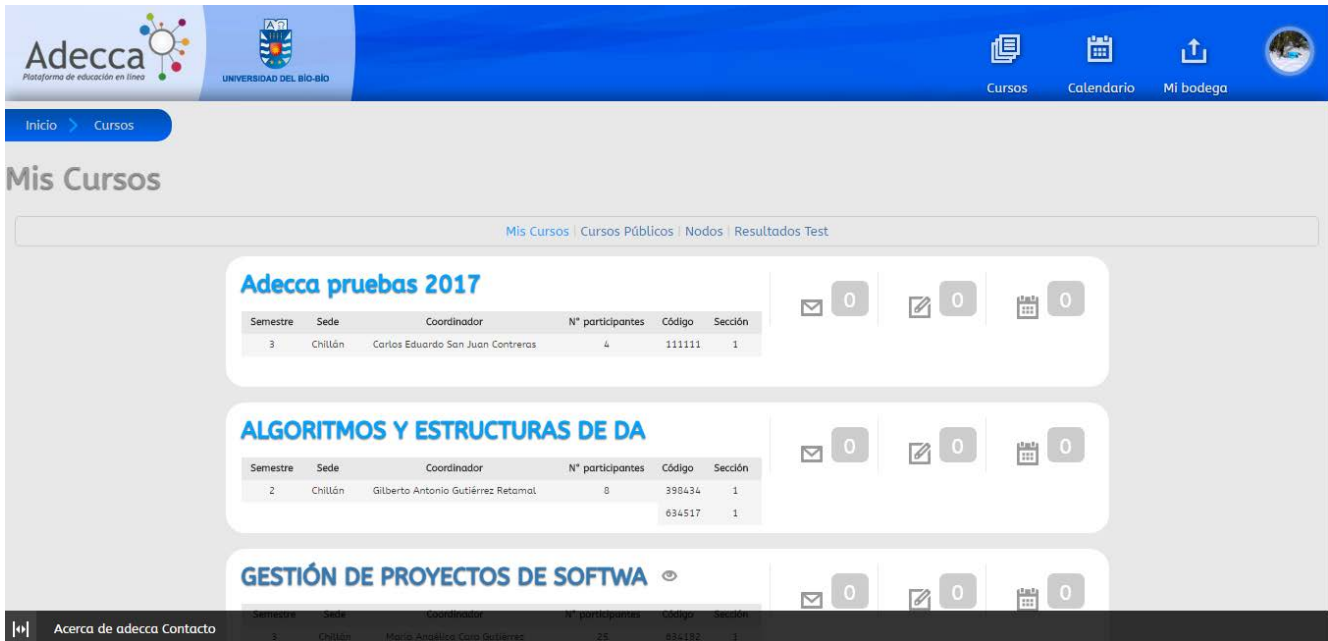


Figura 45. Ejemplo de pantalla principal de Adecca UBB



Figura 46. Ejemplo de pantalla principal de un curso de Adecca UBB

2.2 Especificación de servicios

ID	Q_User
URL	http://146.83.196.237/wadecca/index.php/api/android/login/format/json/
Método	POST
Parámetros de la URL	rut/{rut}/password/{password}/imei/{imei}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo: Código: 200 Contenido: { "code": 1, "perfil": 4288, "nombres": "Rocío Macarena", "apellidos": "Sepúlveda Carriel", "correo": "rocsepul@alumnos.ubiobio.cl" }</p>
Respuesta errónea	<p>Ejemplo: Código: 400, solicitud incorrecta Contenido: null</p> <p>Ejemplo: Código: 403, acceso denegado Contenido: null</p>
Llamada de ejemplo	<pre>public interface UserService { @POST("rut/{rut}/password/{password}/imei/{imei}") Call<Login> login(@Path("rut") String rut, @Path("password") String password, @Path("imei") String imei); } public LoginInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_LOGIN) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()) .build(); service = retrofit.create(UserService.class); }</pre>

Tabla 25. Servicio 1

ID	Q_Curso
URL	http://146.83.196.237/wadecca/index.php/api/android/getCourses/format/json/
Método	GET
Parámetros de la URL	id_usuario/{id_usuario}/tipo_cursos/{tipo_cursos}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo: Código: 200 Contenido:</p> <pre>[{ "id": 4864, "nombre": "Prueba 2017", "año": "2017", "codigo": ["634007-1"], "periodo": "Segundo Semestre", "docentes": ["Carlos Eduardo San Juan Contreras", "David Abdías Sepúlveda Riquelme"] }, { ... }]</pre>
Respuesta errónea	<p>Ejemplo: Código: 400, solicitud incorrecta Contenido: null</p> <p>Ejemplo: Código: 403, acceso denegado Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_usuario/{id_usuario}/tipo_cursos/{tipo_cursos}") Call<List<Course>> getCourses(@Path("id_usuario") String id_usuario, @Path("tipo_cursos") int tipo_curso); }</pre>

```
public CourseInteractorImpl() {  
    OkHttpClient.Builder httpClient = new OkHttpClient.Builder();  
    httpClient.addNetworkInterceptor(new AddHeaderInterceptor());  
    Retrofit retrofit = new Retrofit.Builder()  
        .baseUrl(Constants.URL_CURSOS)  
        .addConverterFactory(GsonConverterFactory.create())  
        .client(httpClient.build())  
        .build();  
    service = retrofit.create(CoursesService.class);  
}
```

Tabla 26. Servicio 2

ID	Q_Guia
URL	http://146.83.196.237/wadecca/index.php/api/android/downloadGD/format/json/
Método	GET
Parámetros de la URL	id_curso/{id_curso}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo: Código: 200 Contenido: { "url": "http://146.83.196.237/wadecca/file/download/eyJub21icmUiOiJ2aXZvX2FsX2xpbWl0ZS5qcGciLCJhcmNoaXZvIjoiMTUwMzUyMDQyOTA0ODkuannIiwicGF0aCI6IjIwMTdcL2N1cnNvc1wvNDg2NFwvMTUwMzUyMDQyOTA0ODkuannBnIn0="</p>
Respuesta errónea	<p>Ejemplo: Código: 400, solicitud incorrecta Contenido: null</p> <p>Ejemplo: Código: 403, acceso denegado Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_curso/{id_curso}") Call<ActivityModules.simpleURL> getGuide(@Path("id_curso") String id_curso); }</pre> <pre>public CourseGuideInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constantes.URL_GUIDE) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()) .build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 27. Servicio 3

ID	Q_Prog
URL	http://146.83.196.237/wadecca/index.php/api/android/downloadP/format/json/
Método	GET
Parámetros de la URL	id_curso/{id_curso}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo: Código: 200 Contenido: { "url": "http://146.83.196.237/wadecca/file/download/eyJub21imUiOiJ2aXZvX2FsX2xpbWl0ZS5qcGciLCJhcmNoaXZvIjoieMTUwMzUyMDQyOTA0ODkuanBnIiwicGF0aCI6IjIwMTdcL2N1cnNvc1wvNDg2NFwvMTUwMzUyMDQyOTA0ODkuanBnIn0="</p>
Respuesta errónea	<p>Ejemplo: Código: 400, solicitud incorrecta Contenido: null</p> <p>Ejemplo: Código: 403, acceso denegado Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_curso/{id_curso}") Call<ActivityModules.simpleURL> getProgram(@Path("id_curso") String id_curso); }</pre> <pre>public CourseProgramInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_PROGRAM) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()) .build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 28. Servicio 4

ID	Q_Info
URL	http://146.83.196.237/wadecca/index.php/api/android/getCourseById/format/json/
Método	GET
Parámetros de la URL	id_user/{id_usuario}/id_curso/{id_curso}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>{ "descripcion": "<!DOCTYPE html>\n<html>\n<head>\n</head>\n<body>\n<p>Esta es la descripcion del curso.</p>\n<div id=\"lipsum\">\n<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit.</p>\n</div>\n</body>\n</html>", "modulos": { "content": [{ "content": [{ "id": 29053, "nombre": "diapos", "icono": "recursos", "tipo_modulo": "Recursos" }], "nombre": "Apuntes", "id": 8128 }, { "content": [{ "content": [{ "id": 29062, "nombre": "tarea", "icono": "recursos", "tipo_modulo": "Recursos" }], "nombre": "tareas", "id": 8131 }], "nombre": "Notas", "id": 8129 }], "nombre": "/" }, "eventos": [] }</pre>

<p>Respuesta errónea</p>	<p>Ejemplo: Código: 400, solicitud incorrecta Contenido: null</p> <p>Ejemplo: Código: 403, acceso denegado Contenido: null</p>
<p>Llamada de ejemplo</p>	<pre>public interface CoursesService { @GET("id_user/{id_usuario}/id_curso/{id_curso}") Call<CourseContent> getCourseDetail(@Path("id_usuario") String id_usuario, @Path("id_curso") String id_curso); } public ModulesInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_SINGLE_COURSE) .addConverterFactory(gsonConverter()) .client(httpClient.build()) .build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 29. Servicio 5

ID	Q_Party
URL	http://146.83.196.237/wadecca/index.php/api/android/getUsersByCourse/format/json/
Método	GET
Parámetros de la URL	id_curso/{id_curso}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>[[{ "nombres": "Carlos Eduardo", "apellidos": "San Juan Contreras", "correo": "csanjuan@ubiobio.cl" }], [], [{ "nombres": "Usuario", "apellidos": "De Prueba", "correo": "prueba@gmail.com" }]]</pre>
Respuesta errónea	<p>Ejemplo:</p> <p>Código: 400, solicitud incorrecta</p> <p>Contenido: null</p> <p>Ejemplo:</p> <p>Código: 403, acceso denegado</p> <p>Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_curso/{id_curso}") Call<List<Participant>> getParticipants(@Path("id_curso") String id_curso); } public ParticipantsInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_PARTICIPANTS) .addConverterFactory(gsonConverter()) .client(httpClient.build()).build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 30. Servicio 6

ID	Q_Rec
URL	http://146.83.196.237/wadecca/index.php/api/android/getModuleById/format/json/
Método	GET
Parámetros de la URL	id_modulo/{id_modulo}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>{ "id": 29065, "nombre": "tarea de prueba", "descripcion": "<!DOCTYPE html>\n<html>\n<head>\n</head>\n<body>\n<p>tarea con archivos adjuntos</p>\n</body>\n</html>", "fecha_activacion": "06-09-2017 09:09:00", "fecha_cierre": "00-00-0000 00:00:00", "recursos": [{ "link": "PGEgIGhyZWY9Imh0dHA6Ly8xNDYuODMuMTk2LjIzNy93YWRIY2NhL2ZpbGUvZG93bmxvYWQvZXIKdWIyMWljbVVpT2" }], "vinculos": [] }</pre>
Respuesta errónea	<p>Ejemplo:</p> <p>Código: 400, solicitud incorrecta</p> <p>Contenido: null</p> <p>Ejemplo:</p> <p>Código: 403, acceso denegado</p> <p>Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_modulo/{id_modulo}") Call<ModuleContent> getModuleContent(@Path("id_modulo") String id_modulo); } public ModuleContentInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_SINGLE_MODULE) .addConverterFactory(gsonConverter()) .client(httpClient.build()).build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 31. Servicio 7

ID	Q_Foro
URL	http://146.83.196.237/wadecca/index.php/api/android/getTopicsByForum/format/json/
Método	GET
Parámetros de la URL	id_modulo/{id_modulo}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>[{ "id": 5323, "entrada": "entrada 1", "fecha": "2017-10-18 17:13:28", "autor": "Sepúlveda Carriel, Rocío Macarena" }]</pre>
Respuesta errónea	<p>Ejemplo:</p> <p>Código: 400, solicitud incorrecta</p> <p>Contenido: null</p> <p>Ejemplo:</p> <p>Código: 403, acceso denegado</p> <p>Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_modulo/{id_modulo}") Call<List<Entries>> getForumEntries(@Path("id_modulo") String id_modulo); } public EntriesInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_ENTRIES) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()) .build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 32. Servicio 8

ID	Q_Answ
URL	http://146.83.196.237/wadecca/index.php/api/android/getAnswersByTopic/format/json
Método	GET
Parámetros de la URL	id_topic/{id_topic}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>[{"id": 5324, "entrada": "respuesta 1", "fecha": "18-10-2017 17:13:51", "autor": "Sepúlveda Carriel, Rocío Macarena", "answers": [{ "id": 5325, "entrada": "<p>sub-respuesta 1</p>", "fecha": "31-10-2017 00:51:21", "autor": "Sepúlveda Carriel, Rocío Macarena" }] }]</pre>
Respuesta errónea	<p>Ejemplo:</p> <p>Código: 400, solicitud incorrecta</p> <p>Contenido: null</p> <p>Ejemplo:</p> <p>Código: 403, acceso denegado</p> <p>Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @GET("id_topic/{id_topic}") Call<List<ExpandableListAdapter.Answer>> getForumAnswers(@Path("id_topic") String id_topic); } public AnswersInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_FORUM_ANSWERS) .addConverterFactory(gsonConverter()) .client(httpClient.build()).build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 33. Servicio 9

ID	Q_Perfil
URL	http://146.83.196.237/wadecca/index.php/api/android/getPerfil/format/json/
Método	GET
Parámetros de la URL	id_usuario/{id_usuario}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo:</p> <p>Código: 200</p> <p>Contenido:</p> <pre>{ "alias": "Maca", "oficial_mail": "rocsepul@alumnos.ubiobio.cl", "web": "http://google.com ", "direccion": "Volcanes 1", "telefono": "55555555", "imagen": "/9j/4AAQSk...EnmhAf/Z" }</pre>
Respuesta errónea	<p>Ejemplo:</p> <p>Código: 400, solicitud incorrecta</p> <p>Contenido: null</p> <p>Ejemplo:</p> <p>Código: 403, acceso denegado</p> <p>Contenido: null</p>
Llamada de ejemplo	<pre>public interface UserService { @GET("id_usuario/{id_usuario}") Call<Profile> getProfile(@Path("id_usuario") String id_usuario); } public ProfileInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_PROFILE) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()) .build(); service = retrofit.create(UserService.class); }</pre>

Tabla 34. Servicio 10

ID	A_Entry
URL	http://146.83.196.237/wadecca/index.php/api/android/addPostToForum/format/json/
Método	POST
Parámetros de la URL	id_modulo/{id_modulo}/mensaje/{mensaje}/id_padre/{id_padre}/id_usuario/{id_usuario}/id_entrada/{id_entrada}
Parámetros de datos	adecca-api-key/{key}
Respuesta exitosa	<p>Ejemplo: Código: 200 Contenido: { "result": "success", }</p>
Respuesta errónea	<p>Ejemplo: Código: 400, solicitud incorrecta Contenido: null</p> <p>Ejemplo: Código: 403, acceso denegado Contenido: null</p>
Llamada de ejemplo	<pre>public interface CoursesService { @POST("id_modulo/{id_modulo}/mensaje/{mensaje}/id_padre/{id_padre}/id_usuario/{id_usuario}/id_entrada/{id_entrada}") Call<Void> sendEntry(@Path("id_modulo") String id_modulo, @Path("mensaje") String mensaje, @Path("id_padre") String id_padre, @Path("id_usuario") String id_usuario, @Path("id_entrada") String id_entrada); } public SendEntriesInteractorImpl() { OkHttpClient.Builder httpClient = new OkHttpClient.Builder(); httpClient.addNetworkInterceptor(new AddHeaderInterceptor()); Retrofit retrofit = new Retrofit.Builder() .baseUrl(Constants.URL_SEND_ENTRY) .addConverterFactory(GsonConverterFactory.create()) .client(httpClient.build()) .build(); service = retrofit.create(CoursesService.class); }</pre>

Tabla 35. Servicio 11

Anexo 3: Pruebas Unitarias

```
@RunWith(JUnit4.class)
public class CoursesModelTest {

    CourseInteractor interactor;

    @Before
    public void setUp() {
        interactor = new CourseInteractorImpl();
    }

    @Test
    public void testCurrentCourses_ConnectionCorrect() throws Exception {
        TestSubscriber<List<Course>> subscriber = TestSubscriber.create();
        interactor.getCourses("4288", 1).subscribe(subscriber);
        subscriber.assertNoErrors();
        subscriber.assertCompleted();
    }

    @Test
    public void testCurrentCourses_ConnectionWrong() throws Exception {
        TestSubscriber<List<Course>> subscriber = TestSubscriber.create();
        interactor.getCourses("x", 1).subscribe(subscriber);
        subscriber.assertNotCompleted();
    }
}
```

Anexo 4: Resultados de Sprints

4.1 Sprint 1

Durante este Sprint se logró implementar las cuatro historias de usuario especificadas en el Product Backlog. El registro de lo que se ha realizado por cada historia se muestra a continuación.

4.1.1 Scrum Task Board

Las tareas realizadas en cada historia de usuario se muestran a continuación, en la Figura 47.

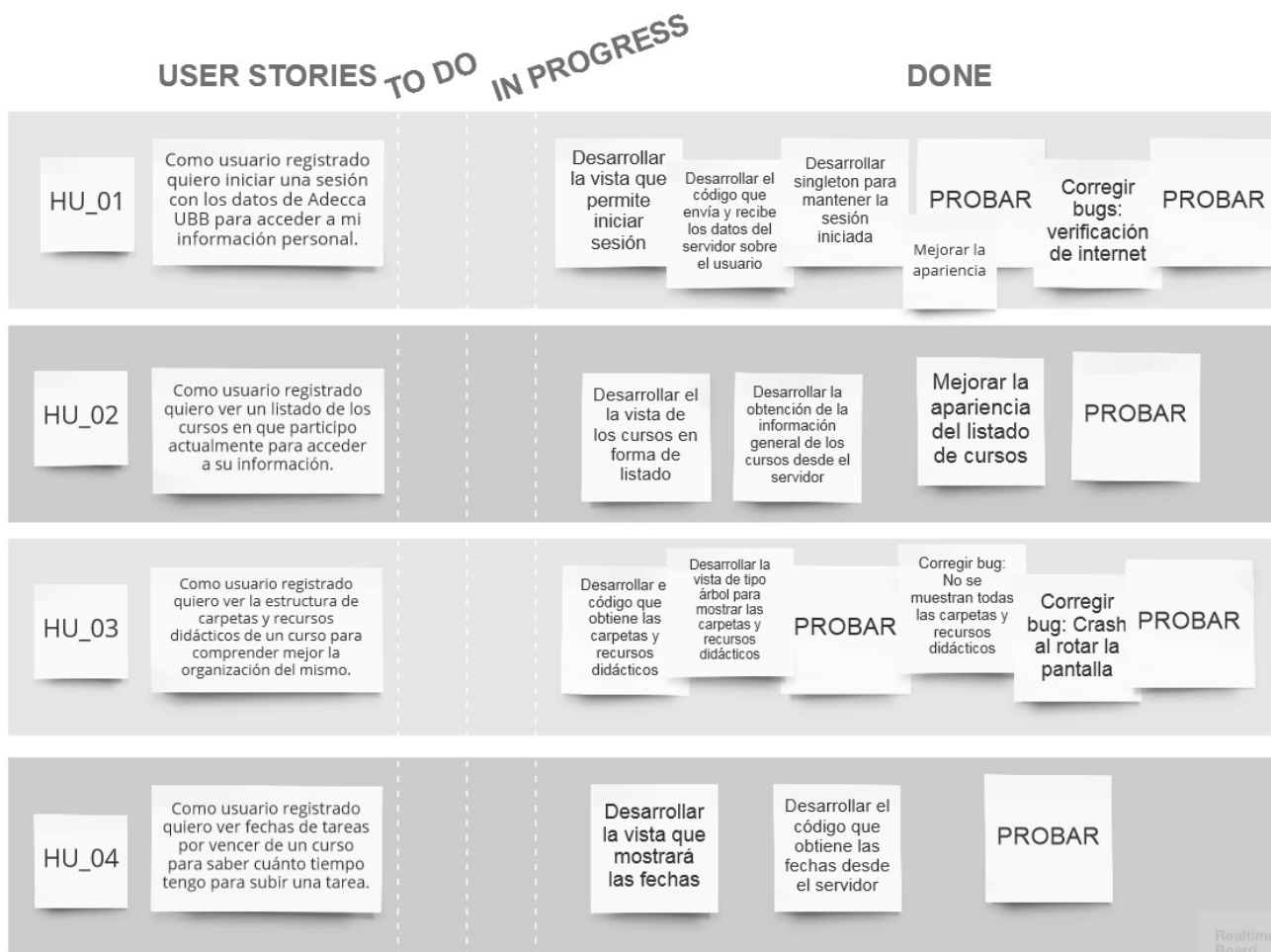


Figura 47. Scrum Task Board, Sprint 1

4.1.2 Cumplimiento de las historias de usuario

A continuación, se evidencia el cumplimiento de cada historia de usuario, mediante capturas de pantalla de la aplicación, además de una breve explicación de la misma.

HU_01

La Figura 48, muestra la pantalla que permite al usuario iniciar sesión. Si los datos se ingresan correctamente, la aplicación muestra la pantalla de bienvenida, como en la Figura 49. Entonces, se evidencia el cumplimiento de HU_01. Además, mediante esta implementación, se evidencia parte del cumplimiento del requisito no funcional de seguridad.



Figura 48. Captura de pantalla HU_01



Figura 49. Captura de pantalla HU_01

HU_02

La Figura 50, corresponde a la pantalla que muestra el nombre de los cursos actuales del usuario, incluyendo el nombre de los docentes a cargo, el periodo y el código. De este modo, se evidencia que HU_02 se ha cumplido correctamente.



Figura 50. Captura de pantalla HU_02

HU_03

La Figura 51, corresponde a la pantalla que muestra la estructura de carpetas y recursos didácticos de un curso. De este modo, se evidencia que HU_03 se ha cumplido correctamente.

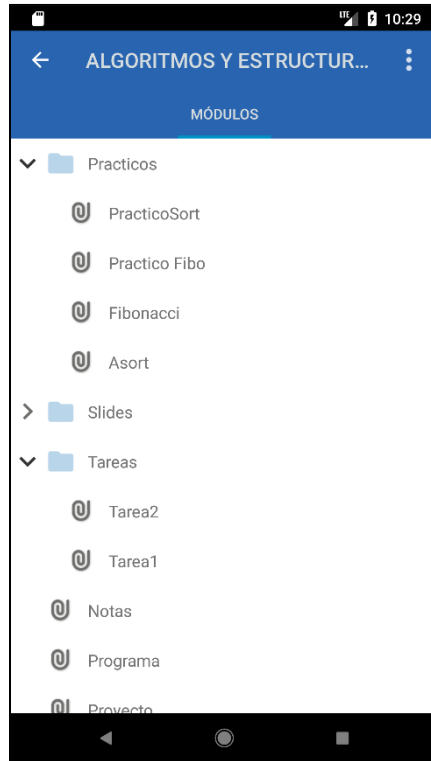


Figura 51. Captura de pantalla HU_03

HU_04

La Figura 52, corresponde a la pantalla que muestra las fechas por vencer de un curso. De este modo, se evidencia que HU_04 se ha cumplido correctamente.



Figura 52. Captura de pantalla HU_04

4.1.3 Primer periodo de pruebas

La documentación de las primeras pruebas de sistema, aplicadas a cada historia de usuario en este Sprint, se encuentra a continuación.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_01	1	Determinar si un usuario puede iniciar sesión sin Internet	No hay conexión a Internet	Ninguno	Mensaje indicando que no hay Internet	Al presionar el botón “Ingresar” de la pantalla de inicio	No despliega ningún mensaje	FRACASO
	2	Determinar si un usuario puede iniciar sesión sin ingresar datos	Los campos de RUT y contraseña no contienen caracteres	Ninguno	Mensaje indicando que se debe ingresar usuario y contraseña	Al presionar el botón “Ingresar” de la pantalla de inicio de sesión	Se muestra mensaje indicando que debe ingresar datos	Éxito
	3	Determinar si un usuario puede iniciar sesión ingresando sólo un RUT	El campo de contraseña no contiene caracteres	RUT	Mensaje indicando que se debe ingresar usuario y contraseña	Al presionar el botón “Ingresar” de la pantalla de inicio de sesión	Se muestra mensaje indicando que debe ingresar datos	Éxito
	4	Determinar si un usuario puede iniciar sesión ingresando sólo una contraseña	El campo de RUT no contiene caracteres	Contraseña	Mensaje indicando que se debe ingresar usuario y contraseña	Al presionar el botón “Ingresar” de la pantalla de inicio de sesión	Se muestra mensaje indicando que debe ingresar datos	Éxito
	5	Determinar si un usuario puede iniciar sesión ingresando un RUT valido y contraseña incorrecta	La contraseña no está asociada al RUT en la base de datos del servidor	RUT existente en la BD y contraseña incorrecta	Mensaje indicando que se debe ingresar usuario y contraseña válidos	Al presionar el botón “Ingresar” de la pantalla de inicio de sesión	Se muestra mensaje indicando que los datos no son válidos	Éxito
	6	Determinar si un usuario puede iniciar sesión con un RUT cuyo dígito verificador es K	El RUT y la contraseña se encuentran en la BD y están asociados	RUT con dígito verificador K y contraseña	Se muestra la pantalla de bienvenida	Al presionar el botón “Ingresar” de la pantalla de inicio de sesión	Se muestra la pantalla de bienvenida con el nombre de usuario	Éxito
	7	Determinar si un usuario puede iniciar sesión con RUT cuyo dígito verificador es un número	El RUT y la contraseña se encuentran en la BD y están asociados	Rut con dígito verificador numérico y contraseña	Se muestra la pantalla de bienvenida	Al presionar el botón “Ingresar” de la pantalla de inicio de sesión	Se muestra la pantalla de bienvenida con el nombre de usuario	Éxito

8	Determinar si un usuario puede iniciar sesión con la pantalla horizontal en Android Ice Cream Sandwich	El RUT y la contraseña se encuentran en la BD y están asociados Sistema Android 4.0 La pantalla está horizontal	RUT y contraseña	Se muestra la pantalla de bienvenida	Al presionar el botón "Ingresar" de la pantalla de inicio de sesión	Se muestra la pantalla de bienvenida	Éxito
9	Determinar si un usuario puede iniciar sesión con la pantalla horizontal en Android Nougat	El RUT y la contraseña se encuentran en la BD y están asociados Sistema Android 7.0 La pantalla está horizontal	RUT y contraseña	Se muestra la pantalla de bienvenida	Al presionar el botón "Ingresar" de la pantalla de inicio de sesión	Se muestra la pantalla de bienvenida	Éxito

Tabla 36. Pruebas de Sistema en HU_01

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_02	1	Determinar si un usuario puede ver sus cursos sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar el botón "Mis Cursos" del menú principal	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede saber que no está participando en cursos	El usuario no participa en cursos en el semestre actual Sesión iniciada	Ninguno	Mensaje indicando que no participa en cursos ese semestre	Al presionar el botón "Mis Cursos" del menú principal	Se muestra mensaje indicando que no hay cursos ese semestre	Éxito
	3	Determinar si un usuario puede ver los cursos en los que está participando en el semestre	Sesión iniciada Usuario participa en cursos del semestre	Ninguno	Lista de cursos en los que participa el usuario en el semestre actual	Al presionar el botón "Mis Cursos" del menú principal	Se muestra la lista de cursos del usuario en el semestre actual	Éxito
	4	Determinar si un usuario puede ver los cursos en los que está participando en el semestre, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada El usuario participa en cursos Sistema Android 4.0 La pantalla está horizontal	Ninguno	Lista de cursos en los que participa el usuario en el semestre actual	Al presionar el botón "Mis Cursos" del menú principal	Se muestra la lista de cursos del usuario en el semestre actual	Éxito
		Determinar si un usuario puede ver los cursos en los que está participando en el semestre, en Android Nougat con pantalla horizontal	Sesión iniciada El usuario participa en cursos Sistema Android 7.0 La pantalla está horizontal	Ninguno	Lista de cursos en los que participa el usuario en el semestre actual	Al presionar el botón "Mis Cursos" del menú principal	Se muestra la lista de cursos del usuario en el semestre actual	Éxito

Tabla 37. Pruebas de Sistema en HU_02

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_03	1	Determinar si un usuario puede ver la estructura de carpetas y recursos didácticos de un curso sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar un curso de la lista de cursos del usuario	Se muestra mensaje que indica que no hay Internet	Éxito
	2	Determinar si un usuario puede ver la estructura de carpetas y recursos didácticos de un curso	El usuario participa en un curso Sesión iniciada	Ninguno	Estructura de carpetas y recursos didácticos de ese curso	Al presionar un curso de la lista de cursos del usuario	La estructura no muestra todos los recursos	FRACASO
	3	Determinar si un usuario puede saber que aún no hay estructura de carpetas y recursos didácticos de un curso	El usuario participa en un curso que no contiene recursos didácticos Sesión iniciada	Ninguno	Mensaje indicando que el curso aún no tiene recursos didácticos	Al presionar un curso de la lista de cursos del usuario	Se muestra un mensaje indicando que no hay recursos didácticos	Éxito
	4	Determinar si un usuario puede ver la estructura de carpetas y recursos didácticos de un curso, en Android Ice Cream Sandwich con pantalla horizontal	El usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Estructura de carpetas y recursos didácticos de ese curso	Al presionar un curso de la lista de cursos del usuario	La aplicación se cierra con un mensaje de error	ERROR GRAVE
	5	Determinar si un usuario puede ver la estructura de carpetas y recursos didácticos de un curso, en Android Nougat con pantalla horizontal	El usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Estructura de carpetas y recursos didácticos de ese curso	Al presionar un curso de la lista de cursos del usuario	La aplicación se cierra con un mensaje de error	ERROR GRAVE

Tabla 38. Pruebas de Sistema en HU_03

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_04	1	Determinar si un usuario puede ver las fechas por vencer disponibles sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de información de un curso	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver las fechas por vencer de un curso	El usuario participa en un curso que contiene fechas por vencer Sesión iniciada	Ninguno	Lista de las fechas por vencer del curso	Al presionar la pestaña de información de un curso	Se muestra la lista de fechas por vencer del curso	Éxito
	3	Determinar si un usuario puede saber que no hay fechas por vencer en un curso	El usuario participa en un curso que no contiene fechas por vencer Sesión iniciada	Ninguno	Mensaje indicando que no hay fechas por vencer	Al presionar la pestaña de información de un curso	Se muestra mensaje indicando que no hay fechas para mostrar	Éxito
	4	Determinar si un usuario puede ver las fechas por vencer en un dispositivo con sistema Android Ice Cream Sandwich, con la pantalla horizontal	El usuario participa en un curso con fechas por vencer Sistema Android 4.0 La pantalla está horizontal	Ninguno	Lista de las fechas por vencer del curso	Al presionar la pestaña de información de un curso	Se muestra la lista de fechas por vencer del curso	Éxito
	5	Determinar si un usuario puede ver las fechas por vencer en un dispositivo con sistema Android Nougat, con la pantalla horizontal	El usuario participa en un curso con fechas por vencer Sistema Android 7.0 La pantalla está horizontal	Ninguno	Lista de las fechas por vencer del curso	Al presionar la pestaña de información de un curso	Se muestra la lista de fechas por vencer del curso	Éxito

Tabla 39. Pruebas de Sistema en HU_04

4.1.4 Segundo periodo de pruebas

En este punto se aplica la misma prueba que fracasó en el primer periodo de pruebas, de acuerdo a la historia de usuario y el escenario, con el objetivo de verificar que el problema se ha resuelto. A continuación se presenta la documentación del segundo periodo de pruebas.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_01	1	Determinar si un usuario puede iniciar sesión sin Internet	No hay conexión a Internet	Ninguno	Mensaje indicando que no hay Internet	Al presionar el botón "Ingresar" de la pantalla de inicio	Se muestra mensaje indicando que no hay conexión a Internet	Éxito

Tabla 40. Pruebas de Sistema en HU_01

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_03	2	Determinar si un usuario puede ver la estructura de carpetas y recursos didácticos de un curso	El usuario participa en un curso Sesión iniciada	Ninguno	Estructura de carpetas y recursos didácticos de ese curso	Al presionar un curso de la lista de cursos del usuario	Se muestra la estructura de carpetas y recursos	Éxito
	4	Determinar si un usuario puede ver la estructura de carpetas y recursos didácticos de un curso, en Android Ice Cream Sandwich con pantalla horizontal	El usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Estructura de carpetas y recursos didácticos de ese curso	Al presionar un curso de la lista de cursos del usuario	Se muestra la estructura de carpetas y recursos	Éxito
	5	Determinar si un usuario puede ver la estructura de carpetas y recursos didácticos de un curso, en Android Nougat con pantalla horizontal	El usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Estructura de carpetas y recursos didácticos de ese curso	Al presionar un curso de la lista de cursos del usuario	Se muestra la estructura de carpetas y recursos	Éxito

Tabla 41. Pruebas de Sistema en HU_03

4.2 Sprint 2

Durante este Sprint se logró implementar las ocho historias de usuario especificadas en el Product Backlog. El registro de lo que se ha realizado por cada historia se muestra a continuación.

4.2.1 Scrum Task Board

Las tareas realizadas en cada historia de usuario se muestran a continuación, en la Figura 53.

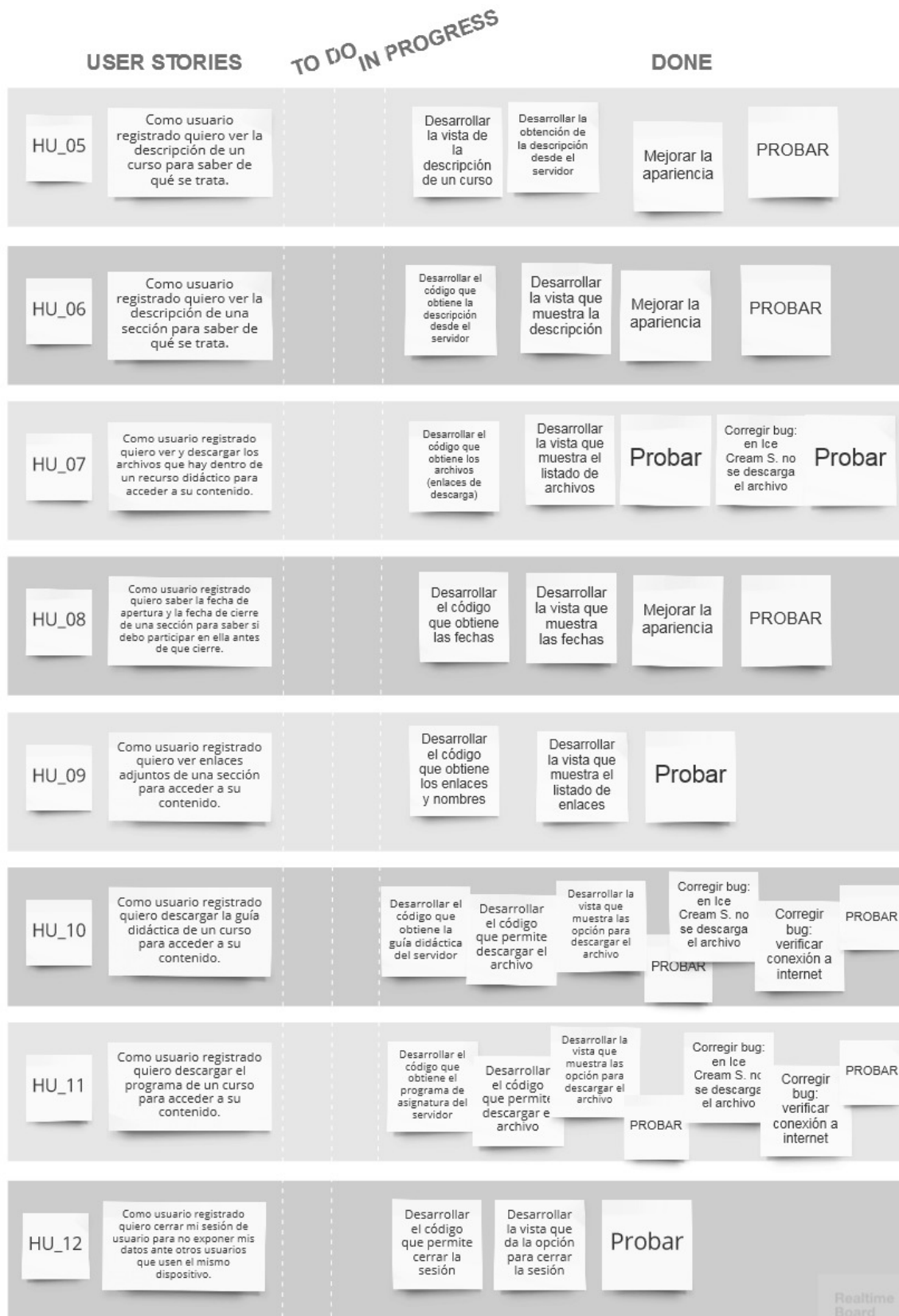


Figura 53. Scrum Task Board, Sprint 2

4.2.2 Cumplimiento de las historias de usuario

A continuación, se evidencia el cumplimiento de cada historia de usuario, mediante capturas de pantalla de la aplicación, además de una breve explicación de la misma.

HU_05

La Figura 54, corresponde a la pantalla que muestra la descripción de un curso. De este modo, se evidencia que HU_05 se ha cumplido correctamente.

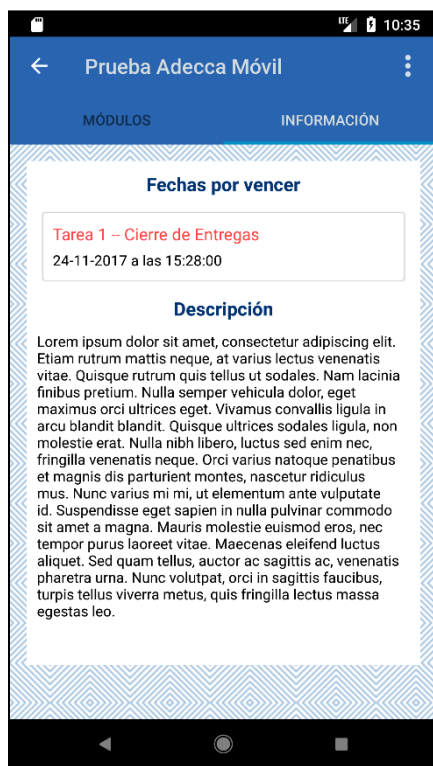


Figura 54. Captura de pantalla HU_05

HU_06

La Figura 54, corresponde a la pantalla que muestra la descripción de un recurso didáctico. De este modo, se evidencia que HU_06 se ha cumplido correctamente.

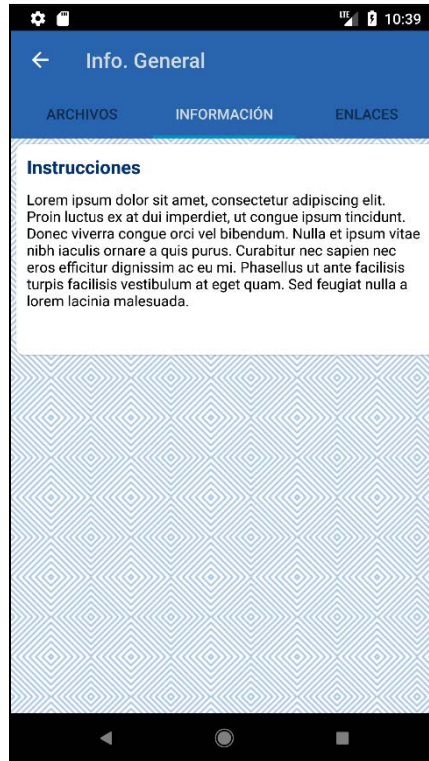


Figura 55. Captura de pantalla HU_06

HU_07

La Figura 56, corresponde a la pantalla que muestra los archivos de un recurso didáctico. Si se selecciona uno de los archivos, el dispositivo solicita permisos para descargarlo al almacenamiento, o si ya tiene permisos, lo descarga directamente. De este modo, se evidencia que HU_07 se ha cumplido correctamente.

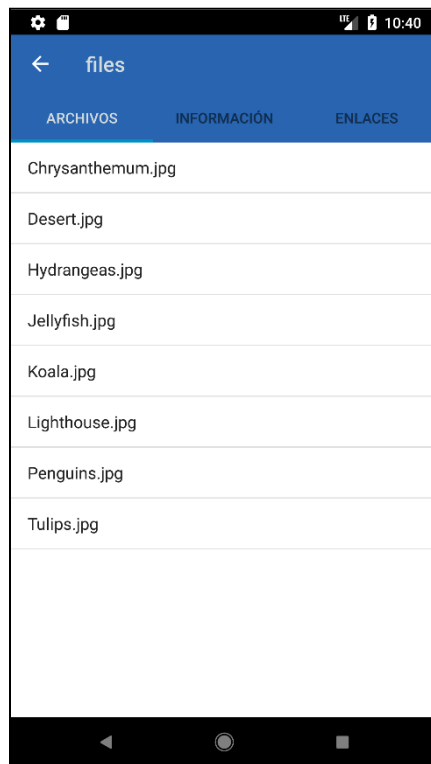


Figura 56. Captura de pantalla HU_07

HU_08

La Figura 57, corresponde a la pantalla que muestra la fecha de apertura y cierre de un recurso didáctico. De este modo, se evidencia que HU_08 se han cumplido correctamente.

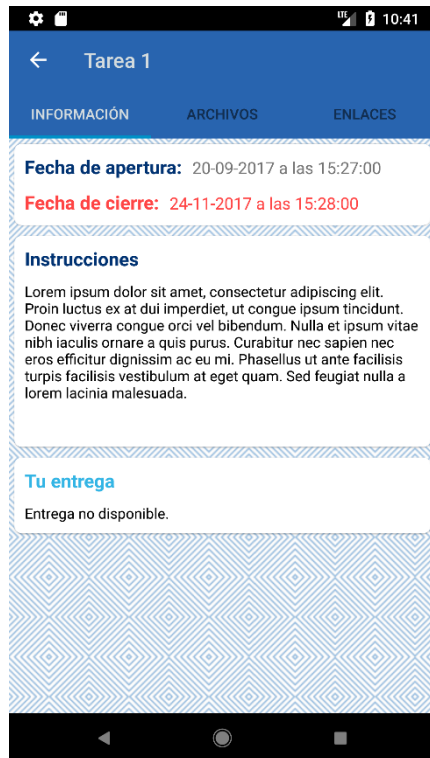


Figura 57. Captura de pantalla HU_08

HU_09

La Figura 58, corresponde a la pantalla que muestra enlaces de un recurso didáctico. Si se selecciona uno de los enlaces, el dispositivo muestra un panel con los navegadores que pueden abrir el enlace, si hay más de uno. Escogido un navegador, se muestra la página del enlace. De este modo, se evidencia que HU_09 se ha cumplido correctamente.

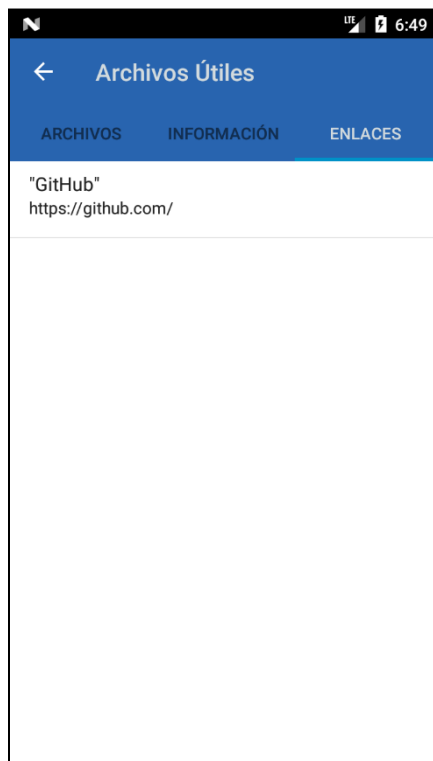


Figura 58. Captura de pantalla HU_09

HU_10

La Figura 59, corresponde a la pantalla que muestra la opción para descargar la guía didáctica del curso. Si se selecciona y el dispositivo no tiene permisos de almacenamiento, se muestra un cuadro solicitando los permisos. Si el dispositivo tiene permiso de almacenamiento, el archivo se descarga directamente. De este modo, se evidencia que HU_10 se ha cumplido correctamente.

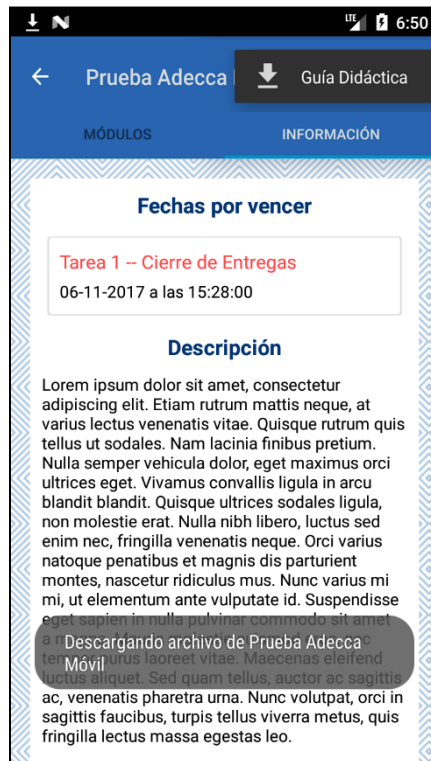


Figura 59. Captura de pantalla HU_10

HU_11

La Figura 60, corresponde a la pantalla que muestra la opción para descargar el programa del curso. Si se selecciona y el dispositivo no tiene permisos de almacenamiento, se muestra un cuadro solicitando los permisos. Si el dispositivo tiene permiso de almacenamiento, el archivo se descarga directamente. De este modo, se evidencia que HU_11 se ha cumplido correctamente.

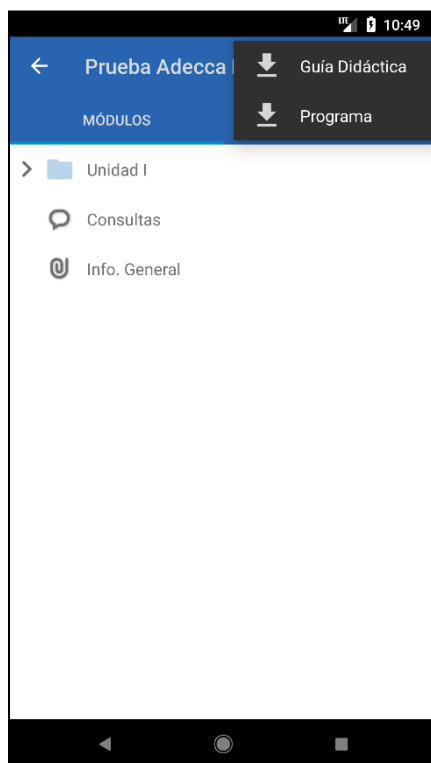


Figura 60. Captura de pantalla HU_11

HU_12

La Figura 61, muestra la opción del menú secundario que permite cerrar la sesión del usuario. Si se selecciona esta opción, la aplicación vuelve a la pantalla que muestra las Noticias UBB, junto con el menú de usuario sin autenticar. De este modo, se evidencia que HU_12 se ha cumplido correctamente.

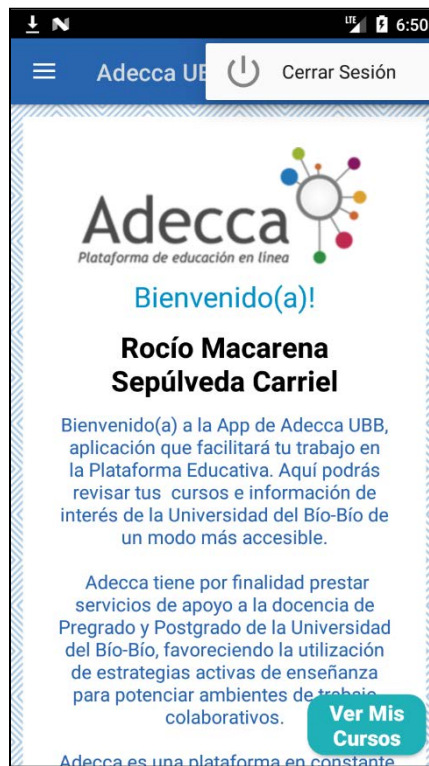


Figura 61. Captura de pantalla HU_12

4.2.3 Primer periodo de pruebas

La documentación de las primeras pruebas de sistema, aplicadas a cada historia de usuario en este Sprint, se encuentra a continuación.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_05	1	Determinar si un usuario puede ver la descripción de un curso sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de información de un curso	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver la descripción de un curso	El usuario participa en un curso que contiene descripción Sesión iniciada	Ninguno	Descripción del curso	Al presionar la pestaña de información de un curso	Se muestra la descripción del curso	Éxito
	3	Determinar si un usuario puede saber que no hay descripción en un curso	El usuario participa en un curso que no contiene descripción Sesión iniciada	Ninguno	Mensaje indicando que no hay descripción	Al presionar la pestaña de información de un curso	Se muestra mensaje indicando que el curso no tiene descripción	Éxito
	4	Determinar si un usuario puede ver la descripción de un curso en un dispositivo con sistema Android Ice Cream Sandwich, con la pantalla horizontal	El usuario participa en un curso con descripción Sistema Android 4.0 La pantalla está horizontal	Ninguno	Descripción del curso	Al presionar la pestaña de información de un curso	Se muestra la descripción del curso	Éxito
	5	Determinar si un usuario puede ver la descripción de un curso en un dispositivo con sistema Android Nougat, con la pantalla horizontal	El usuario participa en un curso con descripción Sistema Android 7.0 La pantalla está horizontal	Ninguno	Descripción del curso	Al presionar la pestaña de información de un curso	Se muestra la descripción del curso	Éxito

Tabla 42. Pruebas de Sistema en HU_05

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_06	1	Determinar si un usuario puede ver la descripción de un recurso didáctico sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de información de un recurso didáctico	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver la descripción de un recurso didáctico	El usuario participa en un recurso didáctico que contiene descripción Sesión iniciada	Ninguno	Descripción del recurso didáctico	Al presionar la pestaña de información de un recurso didáctico	Se muestra la descripción del recurso didáctico	Éxito
	3	Determinar si un usuario puede saber que no hay descripción en un recurso didáctico	El usuario participa en un recurso didáctico que no contiene descripción Sesión iniciada	Ninguno	Mensaje indicando que no hay descripción	Al presionar la pestaña de información de un recurso didáctico	Se muestra un mensaje indicando que no hay descripción	Éxito
	4	Determinar si un usuario puede ver la descripción de un recurso didáctico en un dispositivo con sistema Android Ice Cream Sandwich, con la pantalla horizontal	El usuario participa en un recurso didáctico que contiene descripción Sistema Android 4.0 La pantalla está horizontal	Ninguno	Descripción del recurso didáctico	Al presionar la pestaña de información de un recurso didáctico	Se muestra la descripción del recurso didáctico	Éxito
	5	Determinar si un usuario puede ver la descripción de un recurso didáctico en un dispositivo con sistema Android Nougat, con la pantalla horizontal	El usuario participa en un recurso didáctico que contiene descripción Sistema Android 7.0 La pantalla está horizontal	Ninguno	Descripción del recurso didáctico	Al presionar la pestaña de información de un recurso didáctico	Se muestra la descripción del recurso didáctico	Éxito

Tabla 43. Pruebas de Sistema en HU_06

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_07	1	Determinar si un usuario puede ver los archivos que hay dentro de un recurso didáctico sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de archivos de un recurso didáctico	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver los archivos que hay dentro de un recurso didáctico	El usuario participa en un recurso didáctico que contiene archivos Sesión iniciada	Ninguno	Lista de archivos	Al presionar la pestaña de archivos de un recurso didáctico	Se muestra la lista de recursos didácticos	Éxito
	3	Determinar si un usuario puede saber que no hay archivos en un recurso didáctico	El usuario participa en un recurso didáctico que no contiene archivos Sesión iniciada	Ninguno	Mensaje indicando que no hay archivos	Al presionar la pestaña de archivos de un recurso didáctico	Se muestra mensaje indicando que no hay archivos	Éxito
	4	Determinar si un usuario puede descargar los archivos que hay dentro de un recurso didáctico sin permisos de almacenamiento en un dispositivo con sistema Nougat y con pantalla horizontal	El usuario participa en un recurso didáctico que contiene archivos Sesión iniciada Sist. Android 7.0 La pantalla está horizontal No hay permiso de almacenamiento	Ninguno	Solicitud para usar el almacenamiento del dispositivo	Al presionar un archivo de la lista de archivos de un recurso didáctico	Se muestra la solicitud para acceder al almacenamiento del dispositivo	Éxito
	5	Determinar si un usuario puede descargar los archivos que hay dentro de un recurso didáctico con permisos de almacenamiento en un dispositivo con sistema Nougat y con pantalla horizontal	El usuario participa en un recurso didáctico que contiene archivos Sesión iniciada Sist. Android 7.0 La pantalla está horizontal Sí hay permiso de almacenamiento	Ninguno	Archivo descargado	Al presionar un archivo de la lista de archivos de un recurso didáctico	Se descarga el archivo	Éxito

6	Determinar si un usuario puede descargar los archivos que hay dentro de un recurso didáctico en un dispositivo con sistema Ice Cream Sandwich y con pantalla horizontal	El usuario participa en un recurso didáctico que contiene archivos Sesión iniciada Sistema Android 4.0 La pantalla está horizontal	Ninguno	Archivo descargado	Al presionar un archivo de la lista de archivos de un recurso didáctico	No se descarga el archivo ni aparecen mensajes de error	FRACASO
---	---	---	---------	--------------------	---	---	---------

Tabla 44. Pruebas de Sistema en HU_07

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_08	1	Determinar si un usuario puede ver la fecha de apertura y cierre de un recurso didáctico sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de información de un recurso didáctico	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver la fecha de apertura y cierre de un recurso didáctico	El usuario participa en un recurso didáctico que contiene fecha de cierre Sesión iniciada	Ninguno	Fecha de apertura y fecha de cierre	Al presionar la pestaña de información de un recurso didáctico	Se muestra la fecha de apertura y cierre	Éxito
	3	Determinar si un usuario puede ver la fecha de apertura y cierre de un recurso didáctico en un dispositivo con sistema Android Ice Cream Sandwich, con la pantalla horizontal	El usuario participa en un recurso didáctico que contiene fecha de cierre Sist. Android 4.0 La pantalla está horizontal	Ninguno	Fecha de apertura y fecha de cierre	Al presionar la pestaña de información de un recurso didáctico	Se muestra la fecha de apertura y cierre	Éxito
	4	Determinar si un usuario puede ver la fecha de apertura y cierre de un recurso didáctico en un dispositivo con sistema Android Nougat, con la pantalla horizontal	El usuario participa en un recurso didáctico que contiene fecha de cierre Sist. Android 7.0 La pantalla está horizontal	Ninguno	Fecha de apertura y fecha de cierre	Al presionar la pestaña de información de un recurso didáctico	Se muestra la fecha de apertura y cierre	Éxito

Tabla 45. Pruebas de Sistema en HU_08

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_09	1	Determinar si un usuario puede ver los enlaces que hay dentro de un recurso didáctico sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de enlaces de un recurso didáctico	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver los enlaces que hay dentro de un recurso didáctico	El usuario participa en un recurso didáctico que contiene enlaces Sesión iniciada	Ninguno	Lista de enlaces	Al presionar la pestaña de enlaces de un recurso didáctico	Se muestra la lista de enlaces	Éxito
	3	Determinar si un usuario puede saber que no hay enlaces en un recurso didáctico	El usuario participa en un recurso didáctico que no contiene enlaces Sesión iniciada	Ninguno	Mensaje indicando que no hay enlaces	Al presionar la pestaña de enlaces de un recurso didáctico	Se muestra mensaje indicando que no hay enlaces	Éxito
	4	Determinar si un usuario puede abrir los enlaces que hay dentro de un recurso didáctico	El usuario participa en un recurso didáctico que contiene enlaces Sesión iniciada	Ninguno	Panel solicitando abrir el enlace con un navegador o enlace abierto en navegador	Al presionar un enlace de la lista de enlaces de un recurso didáctico	Se abre el enlace en un navegador	Éxito
	5	Determinar si un usuario puede ver los enlaces que hay dentro de un recurso didáctico en un dispositivo con sistema Nougat y con pantalla horizontal	El usuario participa en un recurso didáctico que contiene enlaces Sesión iniciada Sist. Android 7.0 La pantalla está horizontal	Ninguno	Lista de enlaces	Al presionar la pestaña de enlaces de un recurso didáctico	Se abre el enlace en un navegador	Éxito
	6	Determinar si un usuario puede ver los enlaces que hay dentro de un recurso didáctico en un dispositivo con sistema Ice Cream Sándwich y con pantalla horizontal	El usuario participa en un recurso didáctico que contiene enlaces Sesión iniciada Sist. Android 4.0 La pantalla está horizontal	Ninguno	Lista de enlaces	Al presionar la pestaña de enlaces de un recurso didáctico	Se abre el enlace en un navegador	Éxito

Tabla 46. Pruebas de Sistema en HU_09

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_10	1	Determinar si un usuario puede descargar la guía didáctica de un curso, sin Internet.	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la opción de guía en el menú de un curso	No se muestra el mensaje	FRACASO
	3	Determinar si un usuario puede saber que no hay guía didáctica en un curso	El usuario participa en un curso que no contiene guía didáctica Sesión iniciada	Ninguno	Mensaje indicando que no hay guía	Al presionar la opción de guía en el menú de un curso	Se muestra mensaje indicando que no hay archivo	Éxito
	4	Determinar si un usuario puede descargar la guía didáctica de un curso sin permisos de almacenamiento en un dispositivo con sistema Nougat y con pantalla horizontal	El usuario participa en curso que contiene guía didáctica Sesión iniciada Sist. Android 7.0 La pantalla está horizontal No hay permiso de almacenamiento	Ninguno	Solicitud para usar el almacenamiento del dispositivo	Al presionar la opción de guía en el menú de un curso	Se muestra la solicitud para acceder al almacenamiento del dispositivo	Éxito
	5	Determinar si un usuario puede descargar la guía didáctica de un curso con permisos de almacenamiento en un dispositivo con sistema Nougat y con pantalla horizontal	El usuario participa en un curso que contiene guía didáctica Sesión iniciada Sist. Android 7.0 La pantalla está horizontal Sí hay permiso de almacenamiento	Ninguno	Archivo descargado	Al presionar la opción de guía en el menú de un curso	Se descarga el archivo	Éxito
	6	Determinar si un usuario puede descargar la guía didáctica de un curso en un dispositivo con sistema Ice Cream Sandwich y con pantalla horizontal	El usuario participa en un curso que contiene guía didáctica Sesión iniciada Sist. Android 4.0 La pantalla está horizontal	Ninguno	Archivo descargado	Al presionar la opción de guía en el menú de un curso	No se descarga el archivo ni aparecen mensajes de error	FRACASO

Tabla 47. Pruebas de Sistema en HU_10

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_11	1	Determinar si un usuario puede descargar el programa de un curso, sin Internet.	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la opción de programa en el menú de un curso	No se muestra el mensaje	FRACASO
	3	Determinar si un usuario puede saber que no hay programa en un curso	El usuario participa en un curso que no contiene programa Sesión iniciada	Ninguno	Mensaje indicando que no hay programa	Al presionar la opción de programa en el menú de un curso	Se muestra mensaje indicando que no hay archivo	Éxito
	4	Determinar si un usuario puede descargar el programa de un curso sin permisos de almacenamiento en un dispositivo con sistema Nougat y con pantalla horizontal	El usuario participa en curso que contiene programa Sesión iniciada Sist. Android 7.0 La pantalla está horizontal No hay permiso de almacenamiento	Ninguno	Solicitud para usar el almacenamiento del dispositivo	Al presionar la opción de programa en el menú de un curso	Se muestra la solicitud para acceder al almacenamiento del dispositivo	Éxito
	5	Determinar si un usuario puede descargar el programa de un curso con permisos de almacenamiento en un dispositivo con sistema Nougat y con pantalla horizontal	El usuario participa en un curso que contiene programa Sesión iniciada Sist. Android 7.0 La pantalla está horizontal Sí hay permiso de almacenamiento	Ninguno	Archivo descargado	Al presionar la opción de programa en el menú de un curso	Se descarga el archivo	Éxito
	6	Determinar si un usuario puede descargar el programa de un curso en un dispositivo con sistema Ice Cream Sandwich y con pantalla horizontal	El usuario participa en un curso que contiene programa Sesión iniciada Sist. Android 4.0 La pantalla está horizontal	Ninguno	Archivo descargado	Al presionar la opción de programa en el menú de un curso	No se descarga el archivo ni aparecen mensajes de error	FRACASO

Tabla 48. Pruebas de Sistema en HU_11

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_12	1	Determinar si un usuario puede cerrar sesión	Sesión iniciada	Ninguno	Pantalla de noticias UBB con el menú de un usuario sin autenticar	Al presionar el botón "Cerrar sesión" del menú secundario de inicio	Se muestra la web de noticias UBB y el menú de deslogeado	Éxito

Tabla 49. Pruebas de Sistema en HU_12

4.2.4 Segundo periodo de pruebas

En este punto se aplica la misma prueba que fracasó en el primer periodo de pruebas, de acuerdo a la historia de usuario y el escenario, con el objetivo de verificar que el problema se ha resuelto. A continuación se presenta la documentación del segundo periodo de pruebas.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_07	6	Determinar si un usuario puede descargar los archivos que hay dentro de un recurso didáctico en un dispositivo con sistema Ice Cream Sandwich y con pantalla horizontal	El usuario participa en un recurso didáctico que contiene archivos Sesión iniciada Sistema Android 4.0 La pantalla está horizontal	Ninguno	Archivo descargado	Al presionar un archivo de la lista de archivos de un recurso didáctico	Se descarga el archivo	Éxito

Tabla 50. Pruebas de Sistema en HU_07

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_10	1	Determinar si un usuario puede descargar la guía didáctica de un curso, sin Internet.	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la opción de guía en el menú de un curso	Se muestra mensaje indicando que no hay Internet	Éxito
	6	Determinar si un usuario puede descargar la guía didáctica de un curso en un dispositivo con sistema Ice Cream Sandwich y con pantalla horizontal	El usuario participa en un curso que contiene guía didáctica Sesión iniciada Sist. Android 4.0 La pantalla está horizontal	Ninguno	Archivo descargado	Al presionar la opción de guía en el menú de un curso	Se descarga el archivo	Éxito

Tabla 51. Pruebas de Sistema en HU_10

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_11	1	Determinar si un usuario puede descargar el programa de un curso, sin Internet.	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la opción de programa en el menú de un curso	Se muestra mensaje indicando que no hay Internet	Éxito
	6	Determinar si un usuario puede descargar el programa de un curso en un dispositivo con sistema Ice Cream Sandwich y con pantalla horizontal	El usuario participa en un curso que contiene programa Sesión iniciada Sist. Android 4.0 La pantalla está horizontal	Ninguno	Archivo descargado	Al presionar la opción de programa en el menú de un curso	Se descarga el archivo	Éxito

Tabla 52. Pruebas de Sistema en HU_11

4.3 Sprint 3

Durante este Sprint se logró implementar las cuatro historias de usuario especificadas en el Product Backlog. El registro de lo que se ha realizado por cada historia se muestra a continuación.

4.3.1 Scrum Task Board

Las tareas realizadas en cada historia de usuario se muestran a continuación, en la Figura 62.

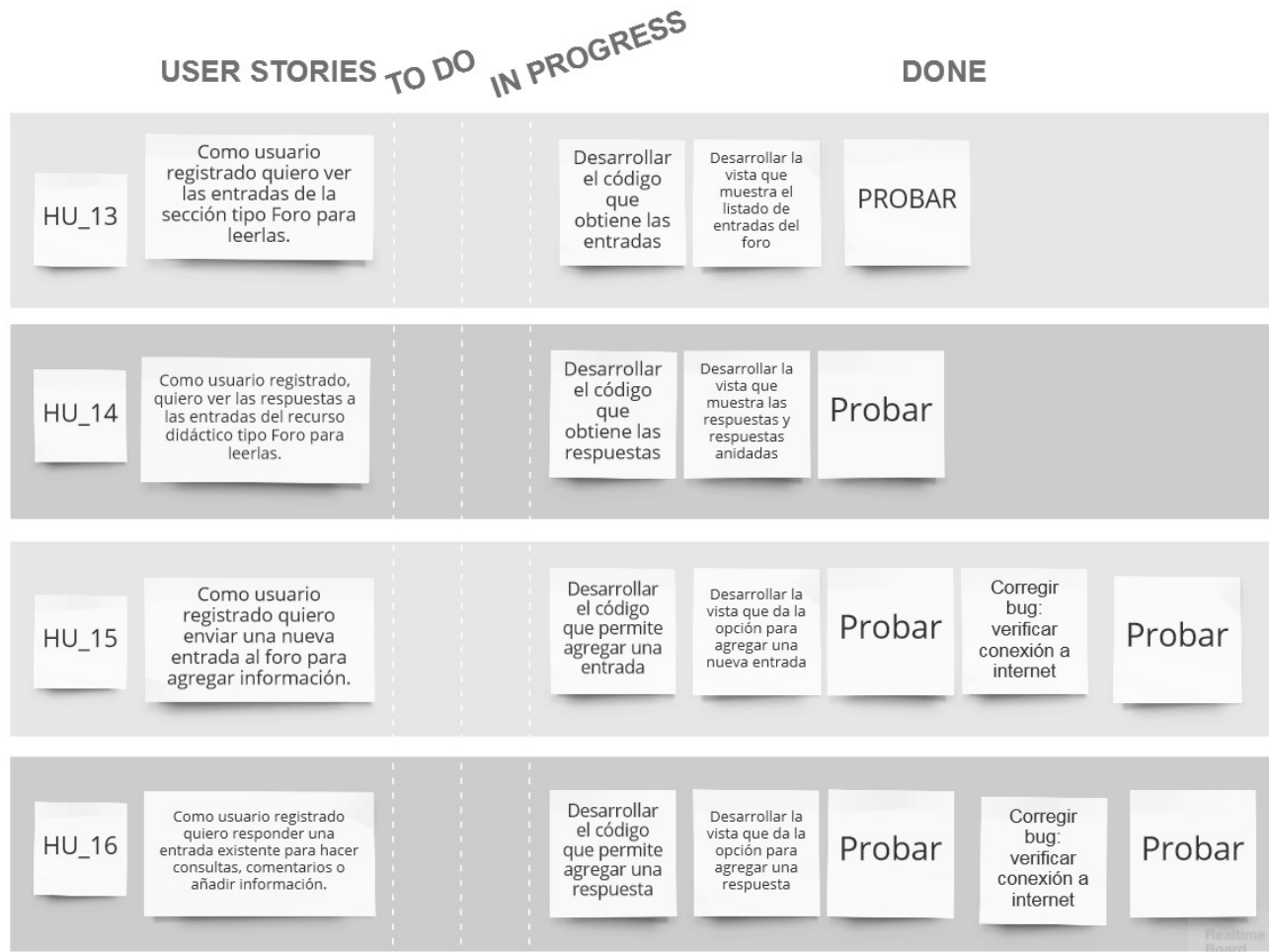


Figura 62. Scrum Task Board, Sprint 3

4.3.2 Cumplimiento de las historias de usuario

A continuación, se evidencia el cumplimiento de cada historia de usuario, mediante capturas de pantalla de la aplicación, además de una breve explicación de la misma.

HU_13

La Figura 63, muestra las lista de entradas cuando se selecciona un recurso didáctico de tipo foro. De este modo, se evidencia que HU_13 se ha cumplido correctamente.



Figura 63. Captura de pantalla HU_13

HU_14

La Figura 64, muestra la lista de respuestas de una entrada seleccionada. De este modo, se evidencia que HU_14 se ha cumplido correctamente.

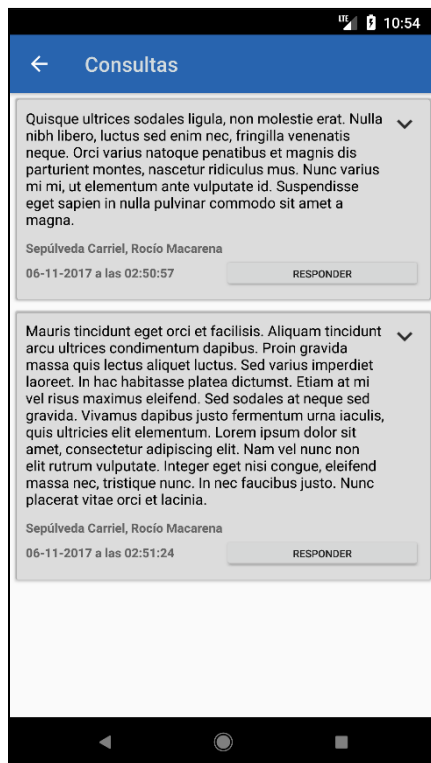


Figura 64. Captura de pantalla HU_14

HU_15

La Figura 65, muestra el cuadro de diálogo que se despliega al presionar el botón “+”, que permite agregar una entrada en un foro, como se aprecia en la figura. Si al presionar “publicar” la entrada se registra correctamente, se despliega un mensaje indicando que se ha publicado, o se despliega un mensaje de error si no se ha registrado. De este modo, se evidencia que HU_15 se ha cumplido correctamente.

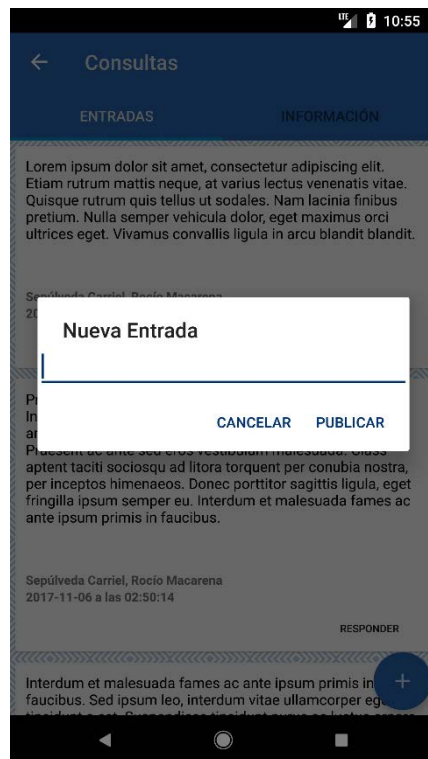


Figura 65. Captura de pantalla HU_15

HU_16

La Figura 66, muestra el cuadro de diálogo que se despliega al presionar el botón “responder” de una respuesta a una entrada, como se aprecia en la figura. Si al presionar “publicar” la respuesta se registra correctamente, se despliega un mensaje indicando que se ha publicado, o se despliega un mensaje de error si no se ha registrado. De este modo, se evidencia que HU_16 se ha cumplido correctamente.

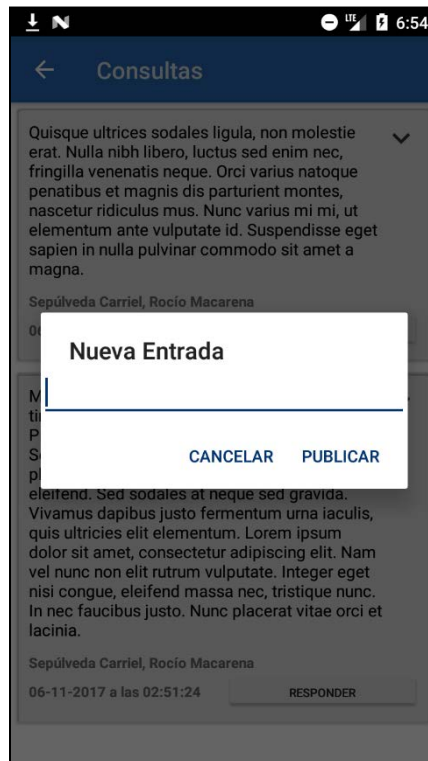


Figura 66. Captura de pantalla HU_16

4.3.3 Primer periodo de pruebas

La documentación de las primeras pruebas de sistema, aplicadas a cada historia de usuario en este Sprint, se encuentra a continuación.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_13	1	Determinar si un usuario puede ver las entradas de un foro sin Internet	Sesión iniciada Sin conexión a Internet	Ninguno	Mensaje indicando que no hay conexión a Internet	Al presionar un recurso didáctico de tipo Foro	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver las entradas de un foro	Sesión iniciada Usuario con participación en un foro con entradas	Ninguno	Lista con las entradas del foro	Al presionar un recurso didáctico de tipo Foro	Se muestra la lista de entradas del foro	Éxito
	3	Determinar si un usuario puede saber que un foro no tiene entradas	Sesión iniciada Conexión a Internet Usuario con participación en un foro sin entradas	Ninguno	Mensaje indicando que aún no hay entradas en el foro	Al presionar un recurso didáctico de tipo Foro	Se muestra mensaje indicando que no hay entradas	Éxito
	4	Determinar si un usuario puede ver las entradas de un foro en Android Nougat con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas Sist. Android 7.0 Pantalla horizontal	Ninguno	Lista con las entradas del foro	Al presionar un recurso didáctico de tipo Foro	Se muestra la lista de entradas del foro	Éxito
	5	Determinar si un usuario puede ver las entradas de un foro en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas Sist. Android 4.0 Pantalla horizontal	Ninguno	Lista con las entradas del foro	Al presionar un recurso didáctico de tipo Foro	Se muestra la lista de entradas del foro	Éxito

Tabla 53. Pruebas de Sistema en HU_13

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_14	1	Determinar si un usuario puede ver las respuestas a las entradas de un foro sin Internet	Sesión iniciada Sin conexión a Internet	Ninguno	Mensaje indicando que no hay conexión a Internet	Al presionar una entrada de un Foro	Se muestra mensaje indicando que no hay Internet	Éxito
	2	Determinar si un usuario puede ver las respuestas a las entradas de un foro	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas y respuestas	Ninguno	Lista con las respuestas a la entrada de un foro	Al presionar una entrada de un Foro	Se muestra la lista con las respuestas a la entrada	Éxito
	3	Determinar si un usuario puede saber que una entrada de foro no tiene respuestas	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas sin respuesta	Ninguno	Mensaje indicando que aún no hay respuestas a tal entrada	Al presionar una entrada de un Foro	Se muestra mensaje indicando que no hay respuestas	Éxito
	4	Determinar si un usuario puede ver las respuestas a las entradas de un foro en Android Nougat con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas y respuestas Sist. Android 7.0 Pantalla horizontal	Ninguno	Lista con las respuestas a la entrada de un foro	Al presionar una entrada de un Foro	Se muestra la lista con las respuestas a la entrada	Éxito
	5	Determinar si un usuario puede ver las respuestas a las entradas de un foro en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas y respuestas Sist. Android 4.0 Pantalla horizontal	Ninguno	Lista con las respuestas a la entrada de un foro	Al presionar una entrada de un Foro	Se muestra la lista con las respuestas a la entrada	Éxito

Tabla 54. Pruebas de Sistema en HU_14

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_15	1	Determinar si un usuario puede agregar una entrada a un foro sin Internet	Sesión iniciada Sin conexión a Internet	Mensaje a publicar	Mensaje indicando que no hay conexión a Internet	Al presionar el botón que permite añadir entrada	No se muestra ningún mensaje ni error	FRACASO
	2	Determinar si un usuario puede agregar una entrada a un foro	Sesión iniciada Usuario con participación en un foro	Mensaje a publicar	Mensaje indicando que la entrada fue publicada	Al presionar el botón que permite añadir entrada	Se muestra mensaje indicando que la entrada fue publicada	Éxito
	3	Determinar si un usuario puede agregar una entrada a un foro en Android Nougat con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro Sist. Android 7.0 Pantalla horizontal	Mensaje a publicar	Mensaje indicando que la entrada fue publicada	Al presionar el botón que permite añadir entrada	Se muestra mensaje indicando que la entrada fue publicada	Éxito
	4	Determinar si un usuario puede agregar una entrada a un foro en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro Sist. Android 4.0 Pantalla horizontal	Mensaje a publicar	Mensaje indicando que la entrada fue publicada	Al presionar el botón que permite añadir entrada	Se muestra mensaje indicando que la entrada fue publicada	Éxito

Tabla 55. Pruebas de Sistema en HU_15

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_16	1	Determinar si un usuario puede agregar una respuesta a una entrada de foro sin Internet	Sesión iniciada Sin conexión a Internet	Mensaje a publicar	Mensaje indicando que no hay conexión a Internet	Al presionar el botón que permite añadir respuesta	No se muestra ningún mensaje ni error	FRACASO
	2	Determinar si un usuario puede agregar una respuesta a una entrada de foro	Sesión iniciada Usuario con participación en un foro con entradas	Mensaje a publicar	Mensaje indicando que la respuesta fue publicada	Al presionar el botón que permite añadir respuesta	Se muestra mensaje indicando que la respuesta fue publicada	Éxito
	3	Determinar si un usuario puede agregar una respuesta a una entrada de foro en Android Nougat con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas Sist. Android 7.0 Pantalla horizontal	Mensaje a publicar	Mensaje indicando que la respuesta fue publicada	Al presionar el botón que permite añadir respuesta	Se muestra mensaje indicando que la respuesta fue publicada	Éxito
	4	Determinar si un usuario puede agregar una respuesta a una entrada de foro en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Conexión a Internet Usuario con participación en un foro con entradas Sist. Android 4.0 Pantalla horizontal	Mensaje a publicar	Mensaje indicando que la respuesta fue publicada	Al presionar el botón que permite añadir respuesta	Se muestra mensaje indicando que la respuesta fue publicada	Éxito

Tabla 56. Pruebas de Sistema en HU_16

4.3.4 Segundo periodo de pruebas

En este punto se aplica la misma prueba que fracasó en el primer periodo de pruebas, de acuerdo a la historia de usuario y el escenario, con el objetivo de verificar que el problema se ha resuelto. A continuación se presenta la documentación del segundo periodo de pruebas.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_15	1	Determinar si un usuario puede agregar una entrada a un foro sin Internet	Sesión iniciada Sin conexión a Internet	Mensaje a publicar	Mensaje indicando que no hay conexión a Internet	Al presionar el botón que permite añadir entrada	Se muestra mensaje indicando que no hay Internet	Éxito

Tabla 57. Pruebas de Sistema en HU_15

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_16	1	Determinar si un usuario puede agregar una respuesta a una entrada de foro sin Internet	Sesión iniciada Sin conexión a Internet	Mensaje a publicar	Mensaje indicando que no hay conexión a Internet	Al presionar el botón que permite añadir respuesta	Se muestra mensaje indicando que no hay Internet	Éxito

Tabla 58. Pruebas de Sistema en HU_16

4.4 Sprint 4

Durante este Sprint se logró implementar las ocho historias de usuario especificadas en el Product Backlog. El registro de lo que se ha realizado por cada historia se muestra a continuación.

4.4.1 Scrum Task Board

Las tareas realizadas en cada historia de usuario se muestran a continuación, en la Figura 67.

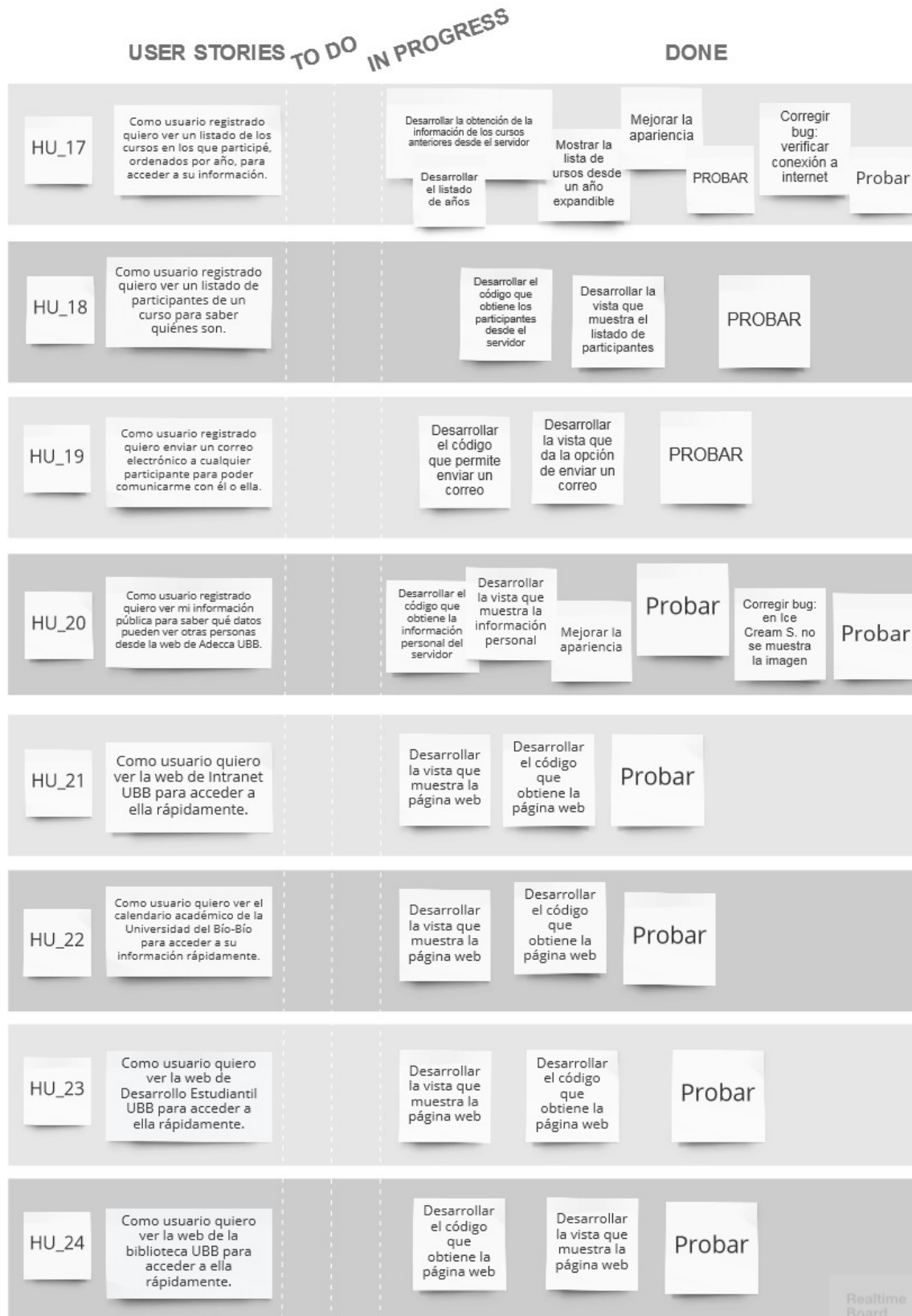


Figura 67. Scrum Task Board, Sprint 4

4.4.2 Cumplimiento de las historias de usuario

A continuación, se evidencia el cumplimiento de cada historia de usuario, mediante capturas de pantalla de la aplicación, además de una breve explicación de la misma.

HU_17

La Figura 68, muestra los cursos anteriores de un usuario, ordenados por año mediante una lista desplegable. Al seleccionar cualquier curso también se muestran los contenidos de este, como en HU_03. De este modo, se evidencia que HU_17 se ha cumplido correctamente.



Figura 68. Captura de pantalla HU_17

HU_18

La Figura 69, muestra el nombre de los participantes de un curso. El color del nombre de los participantes varía según el tipo de participante, es decir, quienes sean docentes aparecen con su nombre en color azul claro, los ayudantes aparecen con color azul medio, mientras que los estudiantes aparecen con color azul oscuro. En este caso, sólo hay docentes y estudiantes. De este modo, se evidencia que HU_18 se ha cumplido correctamente.

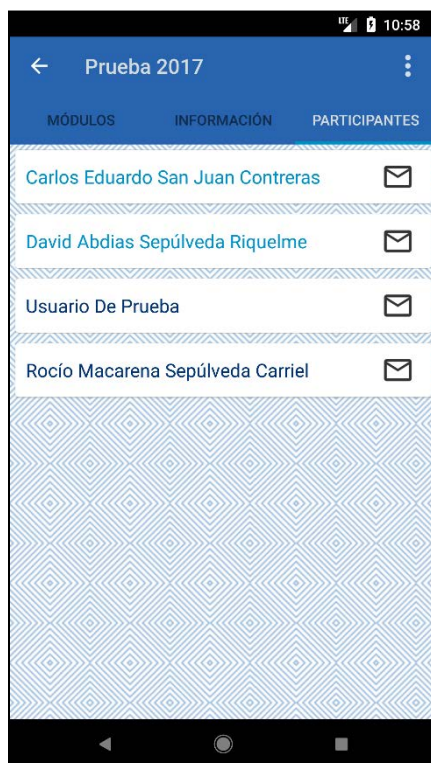


Figura 69. Captura de pantalla HU_18

HU_19

La Figura 70, muestra lo que ocurre al seleccionar el ícono de correo junto al nombre de un participante. Dependiendo del dispositivo, se pueden mostrar varias opciones por donde enviar el correo. Al seleccionar un servidor de correos válido, como Gmail, se abre la pantalla que permite redactar un correo en la aplicación de Gmail, con la dirección de correo del participante en la casilla de destinatario. De este modo, se evidencia que HU_19 se ha cumplido correctamente.

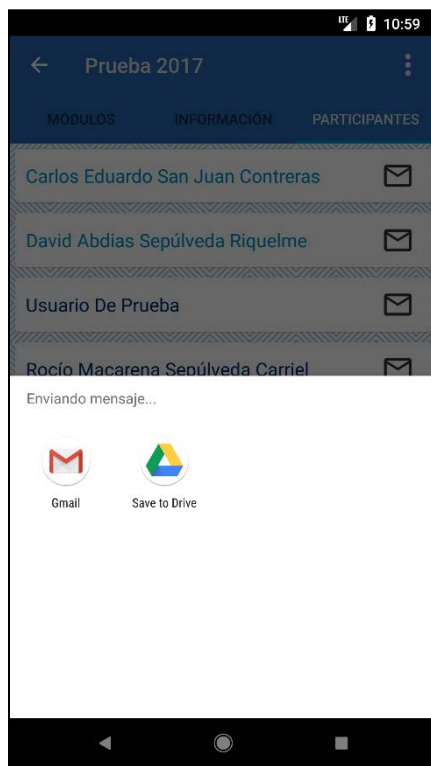


Figura 70. Captura de pantalla HU_19

HU_20

La Figura 71, muestra el perfil que aparece al presionar el nombre de usuario en el menú principal. Los únicos datos de usuario que siempre están presentes corresponden al nombre y correo, ya que no varían. El resto de los datos puede no aparecer, si es que no están registrados en la web de Adecca UBB. De este modo, se evidencia que HU_20 se ha cumplido correctamente.



Figura 71. Captura de pantalla HU_20

HU_21

La Figura 72, muestra lo que ocurre al presionar la opción de “Intranet” de los menús. Debido a la seguridad de la web de Intranet, no permite ser accedida a través de la aplicación, por lo que se debe pasar esta tarea al navegador, de lo contrario, la Universidad podría bloquear las direcciones IP y MAC del usuario. Una vez iniciado el navegador, se muestra la web de Intranet UBB. De este modo, se evidencia que HU_21 se ha cumplido correctamente.

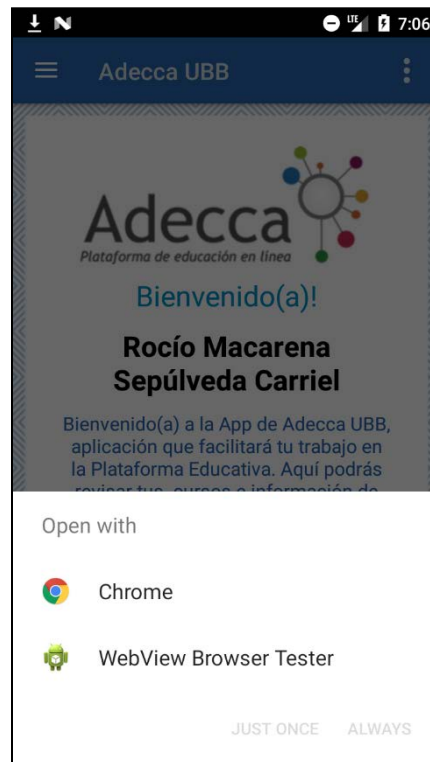


Figura 72. Captura de pantalla HU_21

HU_22

La Figura 73, muestra lo que ocurre al presionar la opción de “Calendario Académico” de los menús. A diferencia de lo que ocurre en HU_21, la web que muestra el calendario académico no tiene mayores restricciones de seguridad, por lo que el contenido de la página se muestra en la aplicación. De este modo, se evidencia que HU_22 se ha cumplido correctamente.



Figura 73. Captura de pantalla HU_22

HU_23

La Figura 74, muestra lo que ocurre al presionar la opción de “DDE Chillán” o “DDE Concepción” en los menús. El contenido de la página se muestra en la aplicación. De este modo, se evidencia que HU_23 se ha cumplido correctamente.



Figura 74. Captura de pantalla HU_23

HU_24

La Figura 75, muestra lo que ocurre al presionar la opción de los menús “Werken” (red de bibliotecas de la Universidad del Bío-Bío). El contenido de la página al no ser responsivo, es apenas visible, entonces se optó por abrir la página en un navegador externo, al igual que HU_21. De este modo, se evidencia que HU_24 se ha cumplido correctamente.

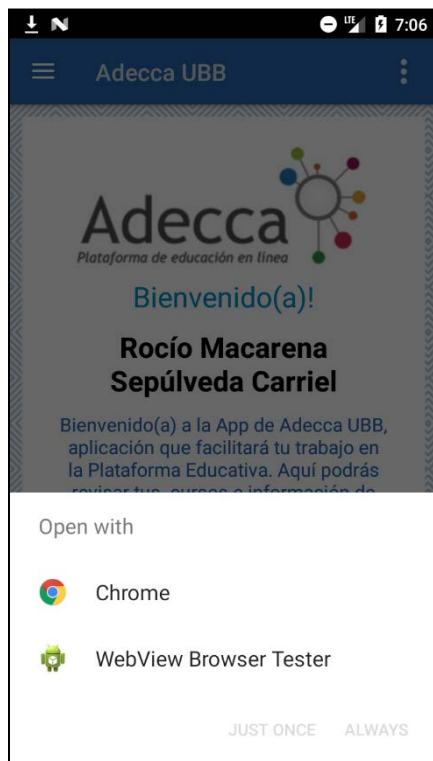


Figura 75. Captura de pantalla HU_24

4.4.3 Primer periodo de pruebas

La documentación de las primeras pruebas de sistema, aplicadas a cada historia de usuario en este Sprint, se encuentra a continuación.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_17	1	Determinar si un usuario puede ver sus cursos anteriores sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar el botón cursos anteriores de la pantalla que muestra los cursos	No se muestra ningún mensaje ni error	FRACASO
	2	Determinar si un usuario puede saber que no está participando en cursos anteriores	El usuario no participa en cursos anteriores al semestre actual Sesión iniciada	Ninguno	Mensaje indicando que no tiene cursos anteriores	Al presionar el botón cursos anteriores de la pantalla que muestra los cursos	Se muestra mensaje indicando que no hay cursos anteriores	Éxito
	3	Determinar si un usuario puede ver sus cursos anteriores	Sesión iniciada El usuario no participa en cursos anteriores al semestre actual	Ninguno	Lista de cursos en los que participó el usuario, ordenados por año	Al presionar el botón cursos anteriores de la pantalla que muestra los cursos	Se muestra la estructura que presenta los cursos anteriores por año	Éxito
	4	Determinar si un usuario puede ver sus cursos anteriores, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada El usuario no participa en cursos anteriores al semestre actual Sistema Android 4.0 La pantalla está horizontal	Ninguno	Lista de cursos en los que participó el usuario, ordenados por año	Al presionar el botón cursos anteriores de la pantalla que muestra los cursos	Se muestra la estructura que presenta los cursos anteriores por año	Éxito
	5	Determinar si un usuario puede ver sus cursos anteriores, en Android Nougat con pantalla horizontal	Sesión iniciada El usuario no participa en cursos anteriores al semestre actual Sistema Android 7.0 La pantalla está horizontal	Ninguno	Lista de cursos en los que participó el usuario, ordenados por año	Al presionar el botón cursos anteriores de la pantalla que muestra los cursos	Se muestra la estructura que presenta los cursos anteriores por año	Éxito

Tabla 59. Pruebas de Sistema en HU_17

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_18	1	Determinar si un usuario puede ver los participantes de un curso sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de participantes en la pantalla de un curso	Se muestra mensaje indicando que no hay conexión a Internet	Éxito
	2	Determinar si un usuario puede ver los participantes de un curso	Sesión iniciada Usuario participa en un curso	Ninguno	Lista de los participantes del curso	Al presionar la pestaña de participantes en la pantalla de un curso	Se muestra la lista de participantes	Éxito
	3	Determinar si un usuario puede ver los participantes de un curso, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Lista de los participantes del curso	Al presionar la pestaña de participantes en la pantalla de un curso	Se muestra la lista de participantes	Éxito
	4	Determinar si un usuario puede ver los participantes de un curso, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Lista de los participantes del curso	Al presionar la pestaña de participantes en la pantalla de un curso	Se muestra la lista de participantes	Éxito

Tabla 60. Pruebas de Sistema en HU_18

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_19	1	Determinar si un usuario puede enviar un correo al participante de un curso	Sesión iniciada Usuario participa en un curso	Ninguno	Opciones para escoger un servidor de correo electrónico o servidor de correo electrónico	Al presionar el icono de mensaje junto al nombre de un participante de un curso	Se muestran opciones de servidores de correo electrónicos	Éxito
	2	Determinar si un usuario puede enviar un correo al participante de un curso, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Cuadro de diálogo con opciones para escoger un servidor de correo electrónico	Al presionar el icono de mensaje junto al nombre de un participante de un curso	Se muestran opciones de servidores de correo electrónicos	Éxito
	3	Determinar si un usuario puede enviar un correo al participante de un curso, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Cuadro de diálogo con opciones para escoger un servidor de correo electrónico	Al presionar el icono de mensaje junto al nombre de un participante de un curso	Se muestran opciones de servidores de correo electrónicos	Éxito

Tabla 61. Pruebas de Sistema en HU_19

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_20	1	Determinar si un usuario puede ver su perfil sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar el nombre de usuario en el menú principal	Se muestra mensaje indicando que no hay conexión a Internet	Éxito
	2	Determinar si un usuario puede ver su perfil	Sesión iniciada Usuario participa en un curso	Ninguno	Datos del usuario	Al presionar el nombre de usuario en el menú principal	Se muestra el perfil del usuario	Éxito
	3	Determinar si un usuario puede ver su perfil, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Sistema Android 4.0 La pantalla está horizontal	Ninguno	Datos del usuario	Al presionar el nombre de usuario en el menú principal	Se muestra el perfil del usuario sin la fotografía de perfil	LOGRO PARCIAL
	4	Determinar si un usuario puede ver su perfil, en Android Nougat con pantalla horizontal	Sesión iniciada Sistema Android 7.0 La pantalla está horizontal	Ninguno	Datos del usuario	Al presionar el nombre de usuario en el menú principal	Se muestra el perfil del usuario	Éxito

Tabla 62. Pruebas de Sistema en HU_20

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_21	1	Determinar si un usuario puede abrir la web de Intranet UBB	Conexión a Internet	Ninguno	Navegador con la web de Intranet	Al presionar "Intranet" en menú principal	Se abre la web de Intranet en un navegador externo a la app	Éxito

Tabla 63. Pruebas de Sistema en HU_21

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_22	1	Determinar si un usuario puede ver el calendario académico UBB	Conexión a Internet	Ninguno	Pantalla con el calendario académico de la Universidad del Bío-Bío	Al presionar "Calendario Académico" en menú principal	Se abre la web del calendario académico en un navegador externo a la app	Éxito

Tabla 64. Pruebas de Sistema en HU_22

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_23	1	Determinar si un usuario puede ver la web de Desarrollo Estudiantil en Chillán	Conexión a Internet	Ninguno	Pantalla mostrando la web de DDE Chillán	Al presionar "DDE Chillán" en menú principal	Se abre la web de desarrollo estudiantil en la aplicación	Éxito
	2	Determinar si un usuario puede ver la web de Desarrollo Estudiantil en Concepción	Conexión a Internet	Ninguno	Pantalla mostrando la web de DDE Concepción	Al presionar "DDE Concepción" en menú principal	Se abre la web de desarrollo estudiantil en la aplicación	Éxito

Tabla 65. Pruebas de Sistema en HU_23

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_24	1	Determinar si un usuario puede ver la web de la biblioteca UBB (Werken)	Conexión a Internet	Ninguno	Pantalla mostrando la web de Werken UBB	Al presionar "Werken" en menú principal	Se abre la web de Werken en el navegador del dispositivo	Éxito

Tabla 66. Pruebas de Sistema en HU_24

4.4.4 Segundo periodo de pruebas

En este punto se aplica la misma prueba que fracasó en el primer periodo de pruebas, de acuerdo a la historia de usuario y el escenario, con el objetivo de verificar que el problema se ha resuelto. A continuación se presenta la documentación del segundo periodo de pruebas.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_17	1	Determinar si un usuario puede ver sus cursos anteriores sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar el botón cursos anteriores de la pantalla que muestra los cursos	Se muestra mensaje indicando que no hay Internet	Éxito

Tabla 67. Pruebas de Sistema en HU_17

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_20	3	Determinar si un usuario puede ver su perfil, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Sistema Android 4.0 La pantalla está horizontal	Ninguno	Datos del usuario	Al presionar el nombre de usuario en el menú principal	Se muestra el perfil del usuario completo	Éxito

Tabla 68. Pruebas de Sistema en HU_20

4.5 Sprint 5

Durante este Sprint se logró implementar las cuatro historias de usuario especificadas en el Product Backlog. El registro de lo que se ha realizado por cada historia se muestra a continuación.

4.5.1 Scrum Task Board

Las tareas realizadas en cada historia de usuario se muestran a continuación, en la Figura 76.

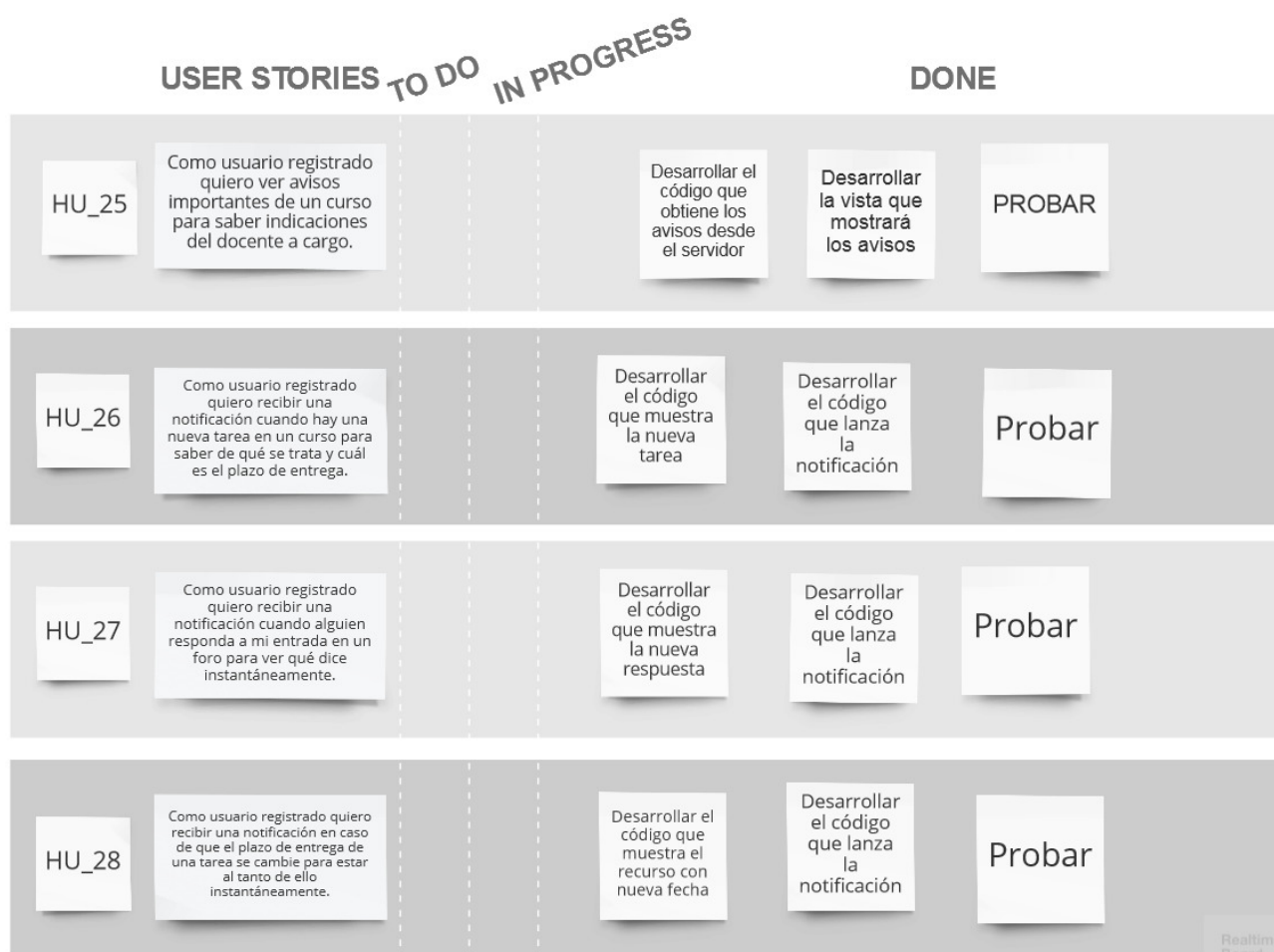


Figura 76. Scrum Task Board, Sprint 5

4.5.2 Cumplimiento de las historias de usuario

A continuación, se evidencia el cumplimiento de cada historia de usuario, mediante capturas de pantalla de la aplicación, además de una breve explicación de la misma.

HU_25

La Figura 77, muestra los avisos que han sido publicados en un curso. De este modo, se evidencia que HU_25 se ha cumplido correctamente.

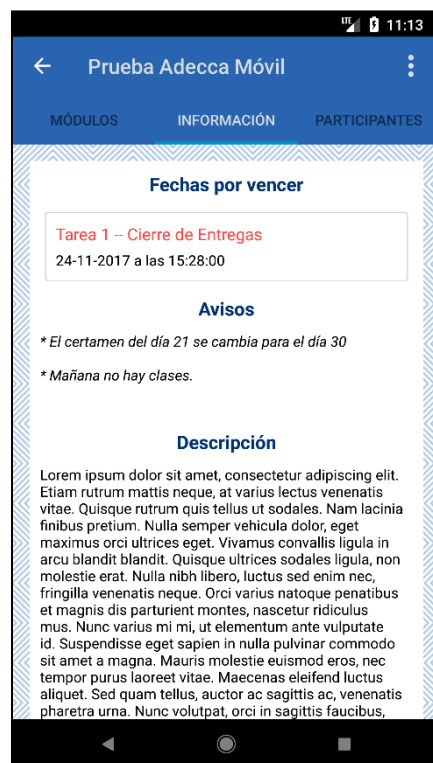


Figura 77. Captura de pantalla HU_25

HU_26

La Figura 78, muestra la notificación que recibe el dispositivo, cuando se publica una nueva tarea en un curso, incluyendo el nombre de este. Al presionarla, muestra la nueva tarea del curso. Así, se evidencia que HU_26 se ha cumplido correctamente.

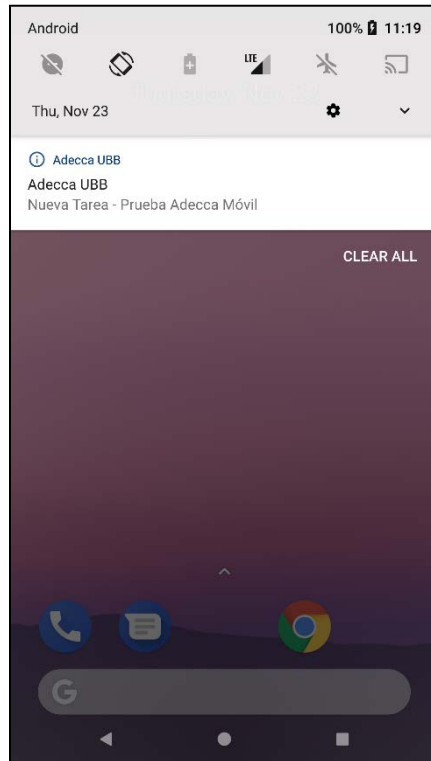


Figura 78. Captura de pantalla HU_26

HU_27

La Figura 79, muestra la notificación que recibe el dispositivo, cuando al usuario le responden una entrada en el foro. Incluye el nombre del foro, nombre del curso y, al presionarla, muestra el foro donde se produjo la actividad. Así, se evidencia que HU_27 se ha cumplido correctamente.

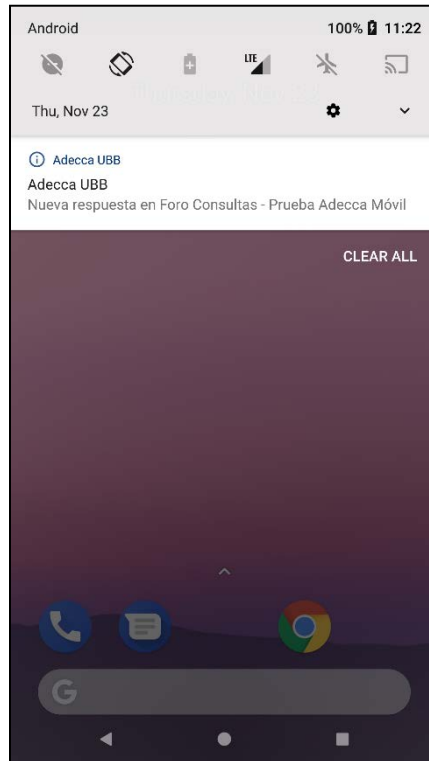


Figura 79. Captura de pantalla HU_27

HU_28

La Figura 80, muestra la notificación que recibe el dispositivo, cuando se cambia el plazo de una tarea en un curso. Incluye el nombre del recurso tipo tarea, nombre del curso y, al presionarla, muestra la tarea donde se produjo la actividad. Así, se evidencia que HU_28 se ha cumplido correctamente.

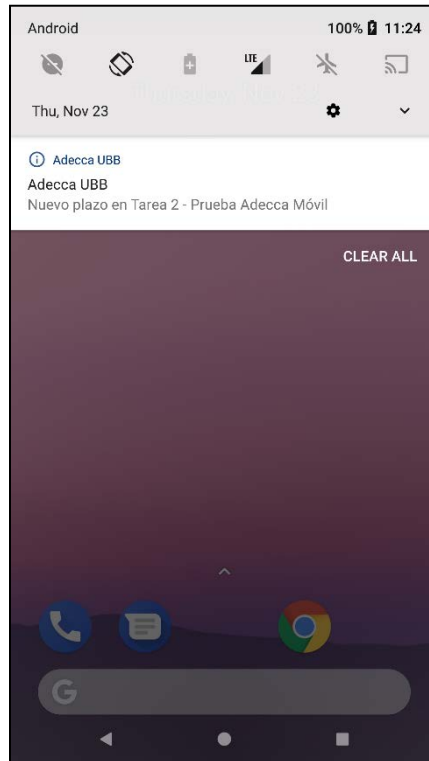


Figura 80. Captura de pantalla HU_28

4.5.3 Primer periodo de pruebas

La documentación de las primeras pruebas de sistema, aplicadas a cada historia de usuario en este Sprint, se encuentra a continuación.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_25	1	Determinar si un usuario puede ver los avisos de un curso sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de Información en la pantalla de un curso (Figura 65)	No se muestra mensaje indicando que no hay conexión a Internet	Fracaso
	2	Determinar si un usuario puede ver los avisos de un curso	Sesión iniciada Usuario participa en un curso	Ninguno	Lista de los avisos del curso	Al presionar la pestaña de Información en la pantalla de un curso	Se muestra la lista de avisos del curso	Éxito
	3	Determinar si un usuario puede ver los avisos de un curso, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Lista de los avisos del curso	Al presionar la pestaña de Información en la pantalla de un curso	Se muestra la lista de avisos del curso	Éxito
	4	Determinar si un usuario puede ver los avisos de un curso, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Lista de los avisos del curso	Al presionar la pestaña de Información en la pantalla de un curso	Se muestra la lista de avisos del curso	Éxito

Tabla 69. Pruebas de Sistema en HU_25

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_26	1	Determinar si un usuario puede recibir una notificación cuando hay una nueva tarea en un curso	Sesión iniciada Usuario participa en un curso	Ninguno	Notificación	-	Se muestra la notificación en la parte superior de la pantalla	Éxito
	2	Determinar si un usuario puede ver la nueva tarea a partir de una notificación recibida, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Pantalla mostrando el nuevo recurso didáctico de tipo tarea	Al presionar la notificación	Se muestra el nuevo recurso didáctico de tipo tarea	Éxito
	3	Determinar si un usuario puede ver la nueva tarea a partir de una notificación recibida, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Pantalla mostrando el nuevo recurso didáctico de tipo tarea	Al presionar la notificación	Se muestra el nuevo recurso didáctico de tipo tarea	Éxito

Tabla 70. Pruebas de Sistema en HU_26

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_27	1	Determinar si un usuario puede recibir una notificación cuando responde a una entrada suya en un foro	Sesión iniciada Usuario participa en un curso y en un foro	Ninguno	Notificación	-	Se muestra la notificación en la parte superior de la pantalla	Éxito
	2	Determinar si un usuario puede ver la respuesta a su entrada a partir de una notificación recibida, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso y en un foro Sistema Android 4.0 La pantalla está horizontal	Ninguno	Pantalla mostrando las respuestas a su entrada	Al presionar la notificación	Se muestran las respuestas a su entrada	Éxito
	3	Determinar si un usuario puede ver la respuesta a su entrada a partir de una notificación recibida, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso y en un foro Sistema Android 7.0 La pantalla está horizontal	Ninguno	Pantalla mostrando las respuestas a su entrada	Al presionar la notificación	Se muestran las respuestas a su entrada	Éxito

Tabla 71. Pruebas de Sistema en HU_27

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_28	1	Determinar si un usuario puede recibir una notificación cuando se cambie el plazo de entrega de una tarea	Sesión iniciada Usuario participa en un curso	Ninguno	Notificación	-	Se muestra la notificación en la parte superior de la pantalla	Éxito
	2	Determinar si un usuario puede ver la tarea a partir de una notificación recibida, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Pantalla mostrando las tarea	Al presionar la notificación	Se muestra la tarea con la fecha actualizada	Éxito
	3	Determinar si un usuario puede ver la tarea a partir de una notificación recibida, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Pantalla mostrando las tarea	Al presionar la notificación	Se muestra la tarea con la fecha actualizada	Éxito

Tabla 72. Pruebas de Sistema en HU_28

4.5.4 Segundo periodo de pruebas

En este punto se aplica la misma prueba que fracasó en el primer periodo de pruebas, de acuerdo a la historia de usuario y el escenario, con el objetivo de verificar que el problema se ha resuelto. A continuación se presenta la documentación del segundo periodo de pruebas.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_25	1	Determinar si un usuario puede ver los avisos de un curso sin Internet	No hay conexión a Internet Sesión iniciada	Ninguno	Mensaje indicando que no hay Internet	Al presionar la pestaña de Información en la pantalla de un curso (Figura 65)	No se muestra mensaje indicando que no hay conexión a Internet	Éxito

Tabla 73. Pruebas de Sistema en HU_25

4.6 Sprint 6

Durante este Sprint se logró implementar las cinco historias de usuario especificadas en el Product Backlog. El registro de lo que se ha realizado por cada historia se muestra a continuación.

4.6.1 Scrum Task Board

Las tareas realizadas en cada historia de usuario se muestran a continuación, en la Figura 81.

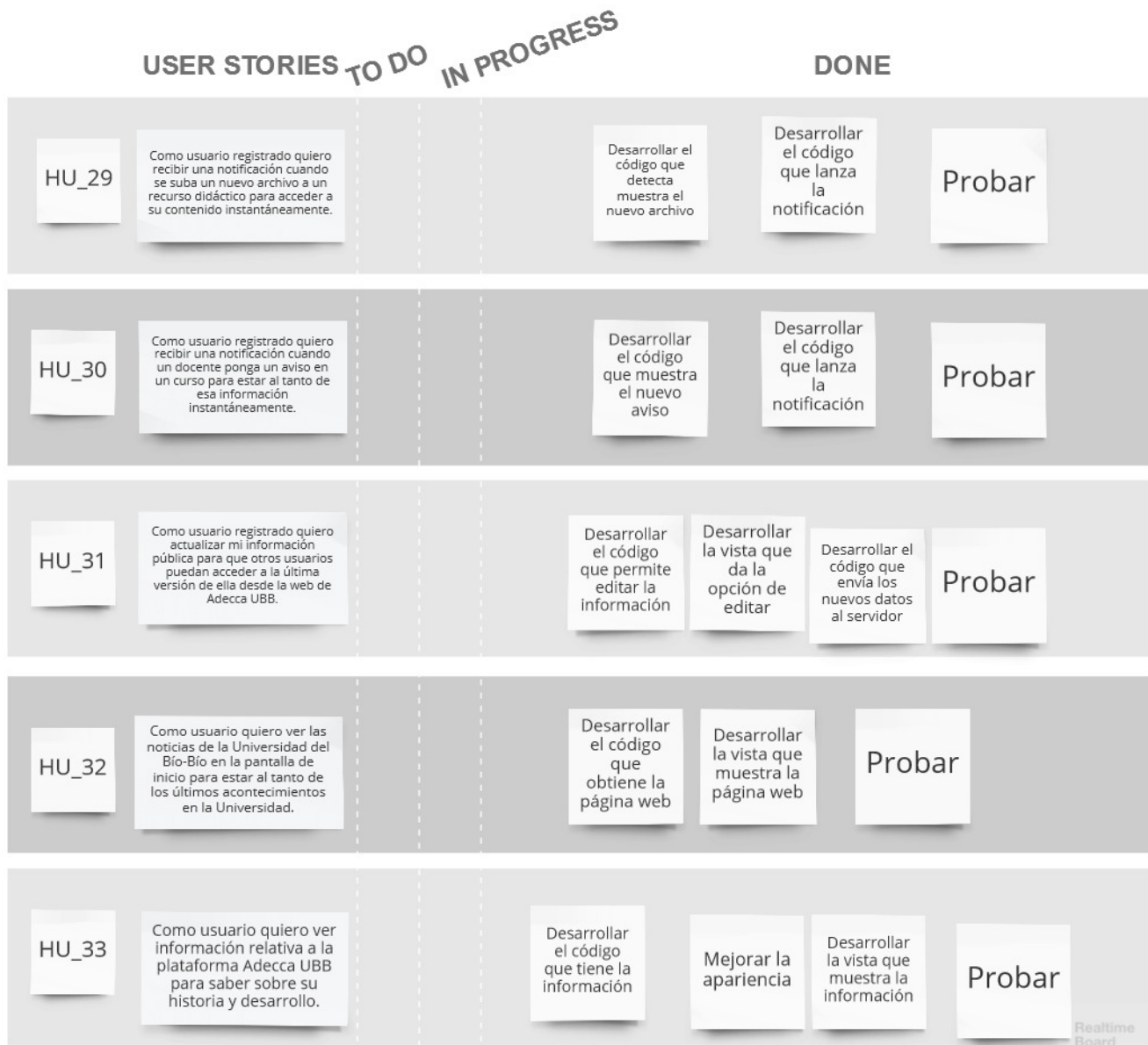


Figura 81. Scrum Task Board, Sprint 6

4.6.2 Cumplimiento de las historias de usuario

A continuación, se evidencia el cumplimiento de cada historia de usuario, mediante capturas de pantalla de la aplicación, además de una breve explicación de la misma.

HU_29

La Figura 82, muestra la notificación que recibe el dispositivo, cuando se sube un nuevo archivo al recurso didáctico de un curso. Al presionarla, muestra el recurso didáctico actualizado. Así, se evidencia que HU_29 se ha cumplido correctamente.

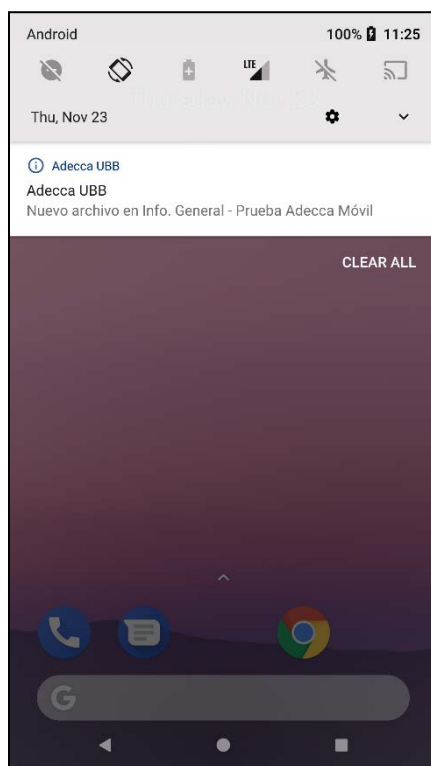


Figura 82. Captura de pantalla HU_29

HU_30

La Figura 83, muestra la notificación que recibe el dispositivo, cuando un docente publica un aviso en un curso. Al presionarla, muestra el curso donde se publicó el aviso. Así, se evidencia que HU_30 se ha cumplido correctamente.

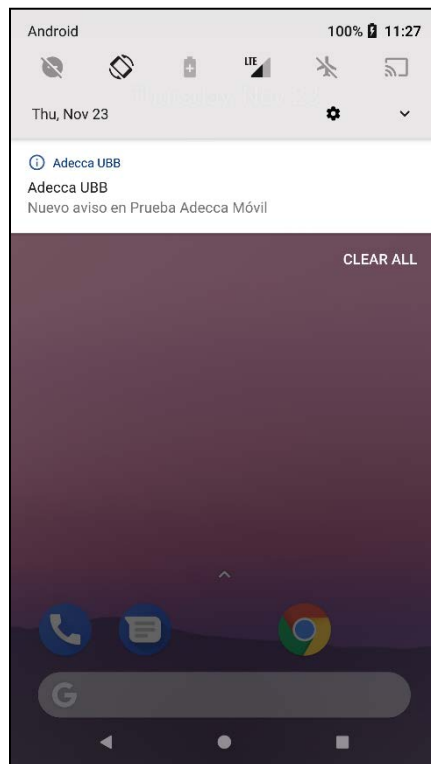


Figura 83. Captura de pantalla HU_30

HU_31

La Figura 84, muestra la pantalla que se abre al presionar un botón de “editar” en el perfil de un usuario. Permite cambiar los atributos de alias, sitio web, dirección y teléfono. Este último debe anotarse sin el código de área y país, ya que se agrega en el servidor. Al presionar actualizar, el sistema vuelve a la pantalla de perfil de usuario con los datos actualizados. Así, se evidencia que HU_31 se ha cumplido correctamente.



Figura 84. Captura de pantalla HU_31

HU_32

La Figura 85, muestra lo que ocurre al presionar la opción de “Noticias UBB” en el menú de usuario autenticado. En el caso de un usuario sin autenticar, las noticias se muestran como pantalla principal. De este modo, se evidencia que HU_32 se ha cumplido correctamente.



Figura 85. Captura de pantalla HU_32

HU_33

La Figura 86, muestra la información de Adecca UBB, al presionar la opción de los menús “Acerca de Adecca UBB”. Así, se evidencia que HU_33 se ha cumplido correctamente.

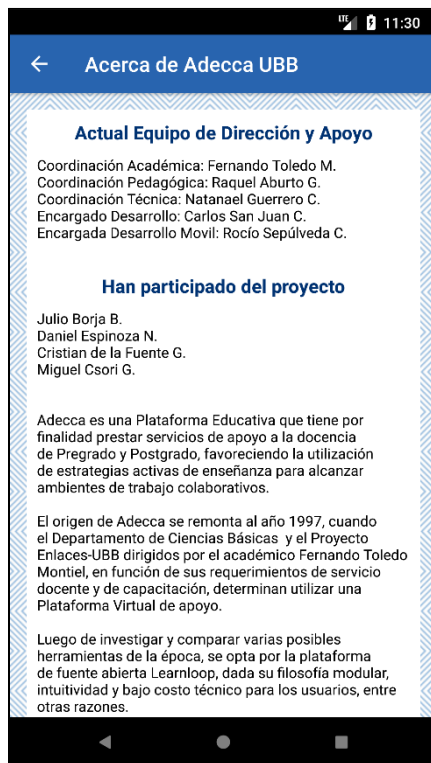


Figura 86. Captura de pantalla HU_33

4.6.3 Primer periodo de pruebas

La documentación de las primeras pruebas de sistema, aplicadas a cada historia de usuario en este Sprint, se encuentra a continuación.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_29	1	Determinar si un usuario puede recibir una notificación cuando se suba un archivo a un curso	Sesión iniciada Usuario participa en un curso	Ninguno	Notificación	-	Se muestra la notificación en la parte superior de la pantalla	Éxito
	2	Determinar si un usuario puede ver el recurso didáctico con el nuevo archivo, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Pantalla mostrando el recurso didáctico	Al presionar la notificación	Se muestra el recurso didáctico con el nuevo archivo	Éxito
	3	Determinar si un usuario puede ver el recurso didáctico con el nuevo archivo, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Pantalla mostrando el recurso didáctico	Al presionar la notificación	Se muestra el recurso didáctico con el nuevo archivo	Éxito

Tabla 74. Pruebas de Sistema en HU_29

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_30	1	Determinar si un usuario puede recibir una notificación cuando se suba un aviso a un curso	Sesión iniciada Usuario participa en un curso	Ninguno	Notificación	-	Se muestra la notificación en la parte superior de la pantalla	Éxito
	2	Determinar si un usuario puede ver el aviso en el curso, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 4.0 La pantalla está horizontal	Ninguno	Pantalla mostrando el aviso	Al presionar la notificación	Se muestra el aviso en el curso	Éxito
	3	Determinar si un usuario puede ver el aviso en el curso, en Android Nougat con pantalla horizontal	Sesión iniciada Usuario participa en un curso Sistema Android 7.0 La pantalla está horizontal	Ninguno	Pantalla mostrando el aviso	Al presionar la notificación	Se muestra el aviso en el curso	Éxito

Tabla 75. Pruebas de Sistema en HU_30

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_31	1	Determinar si un usuario puede editar su perfil sin Internet	Sesión iniciada Sin conexión a Internet	Datos a actualizar (foto, web, alias, etc).	Mensaje indicando que no hay conexión a Internet	Al presionar el botón que permite actualizar el perfil	No se muestra mensaje indicando que no hay Internet	Fracaso
	2	Determinar si un usuario puede editar su perfil	Sesión iniciada	Datos a actualizar (foto, web, alias, etc).	Pantalla de perfil con datos actualizados	Al presionar el botón que permite actualizar el perfil	Se muestra la pantalla de perfil con los datos actualizados	Éxito
	3	Determinar si un usuario puede editar su perfil, en Android Nougat con pantalla horizontal	Sesión iniciada Conexión a Internet Sist. Android 7.0 Pantalla horizontal	Datos a actualizar (foto, web, alias, etc).	Pantalla de perfil con datos actualizados	Al presionar el botón que permite actualizar el perfil	Se muestra la pantalla de perfil con los datos actualizados	Éxito
	4	Determinar si un usuario puede editar su perfil, en Android Ice Cream Sandwich con pantalla horizontal	Sesión iniciada Conexión a Internet Sist. Android 4.0 Pantalla horizontal	Datos a actualizar (foto, web, alias, etc).	Pantalla de perfil con datos actualizados	Al presionar el botón que permite actualizar el perfil	Se muestra la pantalla de perfil con los datos actualizados	Éxito

Tabla 76. Pruebas de Sistema en HU_31

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_32	1	Determinar si un usuario puede ver la web de Noticias UBB	Conexión a Internet	Ninguno	Pantalla mostrando la web de Noticias UBB	Al presionar "Noticias UBB" en menú principal	Se abre la web de Noticias UBB en la aplicación	Éxito

Tabla 77. Pruebas de Sistema en HU_32

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_33	1	Determinar si un usuario puede ver la web de Noticias UBB	Aplicación abierta con o sin Internet	Ninguno	Pantalla mostrando la información de Adecca UBB	Al presionar "Acerca de Adecca UBB" en menú principal	Se muestra la información de Adecca UBB	Éxito

Tabla 78. Pruebas de Sistema en HU_33

4.6.4 Segundo periodo de pruebas

En este punto se aplica la misma prueba que fracasó en el primer periodo de pruebas, de acuerdo a la historia de usuario y el escenario, con el objetivo de verificar que el problema se ha resuelto. A continuación se presenta la documentación del segundo periodo de pruebas.

ID HU	Escenario	Objetivo	Contexto	Inputs	Outputs	Evento	Resultado	Evaluación
HU_31	1	Determinar si un usuario puede editar su perfil sin Internet	Sesión iniciada Sin conexión a Internet	Datos a actualizar (foto, web, alias, etc).	Mensaje indicando que no hay conexión a Internet	Al presionar el botón que permite actualizar el perfil	Se muestra mensaje indicando que no hay Internet	Éxito

Tabla 79. Pruebas de Sistema en HU_31

Anexo 5: Pruebas de eficiencia

Las tablas Tabla 80 y Tabla 81, muestran las pruebas según el servicio probado, horario en que han realizado, tipo de red y resultado.

Servicio	Horario	Tipo de red	Resultado (segundos)
Q_Info	8:30	Pública	3
	8:33	Privada	3
	11:30	Pública	10
	11:32	Privada	5
	16:00	Pública	5
	16:05	Privada	3
	18:00	Pública	5
	18:03	Privada	5

Tabla 80. Pruebas de tiempo de respuesta para Q_Info

Servicio	Horario	Tipo de red	Resultado (segundos)
A_Perfil	8:35	Pública	2
	8:36	Privada	3
	11:35	Pública	8
	11:38	Privada	5
	16:08	Pública	4
	16:11	Privada	3
	18:06	Pública	3
	18:07	Privada	3

Tabla 81. Pruebas de tiempo de respuesta para A_Perfil