



UNIVERSIDAD DE TALCA
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA CIVIL EN COMPUTACIÓN

**Desarrollo de aplicación prototipo que permite la
elaboración de un Plan de Clases para la
Universidad de Talca**

DANIELA ROSSANA PAREDES RODRÍGUEZ

Profesor Guía: DANIEL MORENO CÓRDOVA

Memoria para optar al título de
Ingeniero Civil en Computación

Curicó – Chile
Julio, 2019

CONSTANCIA

La Dirección del Sistema de Bibliotecas a través de su encargado Biblioteca Campus Curicó certifica que el autor del siguiente trabajo de titulación ha firmado su autorización para la reproducción en forma total o parcial e ilimitada del mismo.



Two circular official stamps and handwritten signatures in blue ink. The left stamp is from the 'DIRECCIÓN SISTEMA DE BIBLIOTECAS' of the 'UNIVERSIDAD DE TALCA'. The right stamp is from the 'SISTEMA DE BIBLIOTECAS CAMPUS CURICO' of the 'UNIVERSIDAD DE TALCA'.

Curicó, 2022

A mi eterno acompañante. Porque esas noches de memoria no habrían sido las mismas sin tu apoyo emocional. Gracias y hasta siempre Spock.

AGRADECIMIENTOS

Esta es probablemente una de las últimas páginas que escriba. Y debe ser la más difícil. Porque a lo largo de esta etapa, a veces buena, a veces maldita, hubo una gran cantidad de gente que pasó y dejó una huella. A veces imborrable.

Pero acá vamos.

Dedicado a mi madre, cuyo esfuerzo y perseverancia se traducen en este documento, pero que no refleja ni la más mínima parte de lo grandiosa que es por ella misma. Por el apoyo a pesar de las adversidades, por la comprensión, por los reclamos y la insistencia en algunas ocasiones, por inculcarme el amor al estudio y enseñarme que, con un poco de esfuerzo, puedes ser bueno y lograr cualquier cosa. Por ser el ejemplo de persona, madre y profesional que aspiro a ser algún día.

A Yorch, mi mejor amigo. Mi mayor soporte durante estos años. Nunca dejaré de agradecer el día en que nos presentaron y pasaste a formar parte de mi vida. Todos estos años de juegos, de estudio, de trabajo intenso, de decepciones, de triunfos y comidas no habrían sido lo mismo para mi sin ti a mi lado. Sin tu compañía, tus regaños, tus ideas y tu humor negro. Eres un ejemplo de persona, de líder, de estudiante y de profesional que tuve el agrado de conocer y con quien pude trabajar y compartir. Gracias por todo tu esfuerzo y por todos los momentos lindos que me brindaste durante todos estos años. Gracias por la contención, por las risas, por enseñarme sobre la perseverancia y esfuerzo, algo que sólo había visto en mi madre. Gracias y espero seguir contando contigo por el resto de mi vida.

A Andrea y Gerardo, mis otros mejores amigos. Porque sin ellos no habría logrado terminar esta memoria a tiempo. Por su comprensión, su aliento y su ayuda, por apoyarme en los momentos más difíciles, por dejarme formar parte de sus vidas. Gracias por escucharme cada vez que fue necesario y por apoyarme en este momento tan complicado.

A Rapa, mi padre putativo y profesor de la Universidad. Gracias por estar ahí para mi cuando lo necesité y creer que era capaz de lograrlo. Por todas las veces que me enseñó y me dio un consejo que siempre es y será bien recibido. Por todas las veces que nos (no sólo a mí) ayudó en nuestras causas como estudiantes o problemas personales y nos brindó la mano cuando nadie más quería hacerlo, sin recibir nada a cambio. Por ser el ingeniero al que más admiro y un profesor como pocos. Por ser

la excelente persona que tuve la fortuna de conocer.

A Fabian, por brindarme tu amistad y compañía durante estos últimos años y hacerme olvidar a veces lo difícil que era este momento. A Alan, por toda la ayuda brindada, siempre dispuesto a ofrecer una mano y enseñarte con la mayor de las paciencias. A mis amigos, a mis conocidos, a los integrantes del CAA, a las personas que tuve el agrado de conocer y con la que pude trabajar. A todos aquellos con los que tuve un momento amable y divertido y que hicieron de mi día algo bueno e interesante, o los que estuvieron ahí en un momento malo, e hicieron de ese momento algo más llevadero.

A los profesores del módulo, Matt Bardeen y Ricardo Pérez, quienes hicieron de esta última experiencia algo agradable y tranquilo. A mi profesor guía, Daniel Moreno, por su disposición, tiempo y ayuda durante todo este proceso. Espero que las futuras generaciones de ingenieros sigan contando con sus habilidades de enseñanza y su buen humor. A todos los profesores que creyeron en mí y me contrataron como ayudante. A la profesora Ruth Garrido y a Marcela Pacheco, por toda la ayuda brindada durante todos estos años. Especialmente a Marcela, por todo su apoyo, ayuda, profesionalismo, conceso y buenas palabras. Durante todos mis años en centro de alumnos, no pude tener mejor profesora que ella. Siempre extrañaré el pasarme al tercer piso a saludarla y hacer hora en la oficina de secretaría.

Y no menos importante, gracias a aquellos que me brindaron apoyo emocional y contención durante la corrección de esta memoria. Me demoré, fue duro y fastidioso, pero sin su apoyo y ayuda no habría podido lograrlo.

Finalmente a ti, a quien no pude despedir ni llorar como correspondía. A quien más me dio apoyo moral, contención y soporte durante esas noches de trabajo. A quien nunca olvidare y que siempre llevaré tatuado en mi brazo. A mi grandioso Spock, que siempre vivirá en mi corazón y en cada una de las hojas de esta memoria.

TABLA DE CONTENIDOS

	página
Dedicatoria	I
Agradecimientos	II
Tabla de Contenidos	IV
Índice de Figuras	VIII
Índice de Tablas	XIV
Resumen	XVIII
1. Introducción	19
1.1. Descripción del Contexto	19
1.2. Definición del Problema	20
1.3. Objetivos	21
1.3.1. Objetivo General	21
1.3.2. Objetivos Específicos	21
1.4. Alcances	21
1.5. Propuesta de Solución	22
1.6. Descripción de contenidos	23
2. Marco Teórico	24
2.1. Metodología Curricular Basada en Competencias	24
2.1.1. Fase A	25
2.1.2. Fase B	26
2.1.3. Fase C	26
2.1.4. Fase D	28
2.1.5. Fase E	28
2.2. Tecnologías Utilizadas	29
2.2.1. Estación de Trabajo	30
2.2.2. Frameworks	30

2.2.3.	Lenguajes	31
2.2.4.	JQuery	33
2.2.5.	AWS	34
2.3.	Trabajos Relacionados	35
2.3.1.	Software de Asistencia Universidad de Talca	35
2.3.2.	Excel de plan de clases	35
2.4.	Metodologías	36
2.4.1.	Metodología de Desarrollo: Tradicional Iterativo Incremental	36
2.4.2.	Metodología de evaluación en Ingeniería de Software: Pruebas de Caja Negra	37
2.4.3.	Metodología de evaluación en Ingeniería de Software: Pruebas de Usabilidad	38
2.5.	Resumen	40
3.	Metodología utilizada	41
3.1.	Entorno	41
3.1.1.	Personas	41
3.1.2.	La aplicación	42
3.1.3.	Herramientas	42
3.2.	Metodología	43
3.2.1.	Reuniones	44
3.2.2.	Cómo se trabajó	44
3.3.	Resumen	45
4.	Especificación de Requisitos	46
4.1.	Características de los Usuarios	47
4.2.	Requisitos de Usuario	47
4.2.1.	Requisitos de Usuario: Generales	48
4.2.2.	Requisitos de Usuario: Módulos	48
4.2.3.	Requisitos de Usuario: Competencias, Aprendizajes y Saberes	49
4.3.	Requisitos de Sistema	50
4.3.1.	Requisitos de Sistema: Generales	50
4.3.2.	Requisitos de Sistema: Módulos	51
4.3.3.	Requisitos de Sistema: Unidades	53

4.3.4.	Requisitos de Sistema: Actividades	54
4.3.5.	Requisitos de Sistema: Competencias	55
4.3.6.	Requisitos de Sistema: Aprendizajes	55
4.3.7.	Requisitos de Sistema: Saberes	56
4.4.	Restricciones	57
4.4.1.	Restricciones de Software	57
4.4.2.	Restricciones de Interfaz	57
4.5.	Matriz de Dependencias	59
4.6.	Casos de Uso	61
4.6.1.	Iniciar Sesión	64
4.6.2.	Gestionar Periodos y Años	65
4.6.3.	Imprimir Plan de Clases	66
4.7.	Resumen	67
5.	Diseño del Sistema	68
5.1.	Arquitectura Física	68
5.2.	Arquitectura Lógica	70
5.3.	Diagrama de Clases	72
5.3.1.	Vista	73
5.3.2.	Controlador	73
5.3.3.	Data	75
5.3.4.	Modelo	77
5.4.	Diagrama Entidad-Relación	80
5.4.1.	Relaciones del Modelo	80
5.5.	Resumen	83
6.	Desarrollo	84
6.1.	Iteraciones	84
6.2.	Funcionamiento del Prototipo	106
6.3.	Resumen	115
7.	Pruebas y Análisis de Resultados	116
7.1.	Pruebas de Caja Negra	116
7.2.	Pruebas de Usabilidad	118
7.3.	Resultados	119

7.4. Análisis de resultados	123
8. Conclusiones y Trabajos Futuros	127
Glosario	129
Bibliografía	132
Anexos	
A: Marco Teórico	136
A.1. Diagramas de Metodología para el Diseño Curricular	136
A.2. Plan Curricular Ingeniería Civil en Computación	139
A.3. Software	141
A.3.1. Software de Asistencia Universidad de Talca	141
A.3.2. Excel Plan de Clases Construcción de Software	147
B: Especificación de Requisitos	152
B.1. Matriz de Dependencias	153
B.2. Casos de Uso nivel 0	157
C: Desarrollo	165
C.1. Iteraciones: Primer Prototipo	166
C.2. Iteraciones: Segundo Prototipo	183
D: Pruebas y Resultados	197
D.1. Pruebas de caja negra, Tablas	197
D.2. Pruebas de caja negra, Imágenes	217
D.3. Pruebas de usabilidad: Encuesta	238
D.4. Pruebas de usabilidad: Resultado de encuestas	252
D.5. Pruebas de usabilidad: gráficos	263

ÍNDICE DE FIGURAS

	página
2.1. Diagrama de Metodología para el Diseño Curricular, Fase A.	26
2.2. Diagrama de Metodología para el Diseño Curricular, Fase B.	27
2.3. Diagrama de Metodología para el Diseño Curricular, Fase C.	27
2.4. Diagrama de Metodología para el Diseño Curricular, Fase D.	28
2.5. Diagrama de Metodología para el Diseño Curricular, Fase E.	29
2.6. Ejemplo de plantilla demo de Gentelella.	32
2.7. Ejemplo del Modelo Incremental.	37
2.8. Diagrama de representación de una prueba de caja negra.	38
3.1. Documento de registro de iteraciones	43
3.2. Documento de registro de iteraciones, tareas desplegadas	43
4.1. Ejemplo de Organización de la Matriz de Dependencias.	60
4.2. Matriz de Dependencia con Requisitos Prioritarios Destacados	60
4.3. Diagrama de Casos de Uso Nivel 1: Iniciar Sesión	64
4.4. Diagrama de Casos de Uso Nivel 1: Gestionar Periodos y Años	65
4.5. Diagrama de Casos de Uso Nivel 1: Imprimir Plan de Clases	66
5.1. Diagrama de representación de la Arquitectura Física del sistema.	69
5.2. Diagrama general de representación de la Arquitectura Lógica	70
5.3. Diagrama general de representación de la Vista de la Arquitectura Lógica	72
5.4. Diagrama general de representación del Controlador de la Arquitectu- tura Lógica	72
5.5. Diagrama general de representación del Modelo de la Arquitectura Lógica	73
5.6. Diagrama de Clases: Vista	74
5.7. Diagrama de Clases: Controlador	75
5.8. Diagrama de Clases: Data	77
5.9. Diagrama de Clases: Modelo	78
5.10. Diagrama Entidad-Relación	81

6.1.	Esquema general del funcionamiento del prototipo	106
6.2.	Vista inicio	109
6.3.	Vista competencias	109
6.4.	Vista aprendizajes	110
6.5.	Vista saberes	110
6.6.	Vista usuarios	111
6.7.	Vista cursos	111
6.8.	Vista unidades	112
6.9.	Vista actividades, versión calendario	112
6.10.	Vista actividades	113
6.11.	Vista guardar plan de clases	113
6.12.	Ejemplo plan de clases	114
7.1.	Gráfico de dominio en el uso de un computador	120
7.2.	Gráfico de expectativas del usuario con respecto a la aplicación . . .	121
7.3.	Gráfico de satisfacción del usuario con respecto a la aplicación (Parte 1)	124
7.4.	Gráfico de satisfacción del usuario con respecto a la aplicación (Parte 2)	125
7.5.	Gráfico de tiempo total de creación de un plan de clases	126
A.1.	Diagrama de Metodología para el Diseño Curricular Parte 1	137
A.2.	Diagrama de Metodología para el Diseño Curricular Parte 2	138
A.3.	Plan Curricular 16 de Ingeniería Civil en Computación	140
A.4.	Software de Asistencia Universidad de Talca: Lista de cursos asociados a un profesor	141
A.5.	Software de Asistencia Universidad de Talca: Lista de clases de un curso vacío	142
A.6.	Software de Asistencia Universidad de Talca: Modal agregar clases a un curso	142
A.7.	Software de Asistencia Universidad de Talca: Modal agregar clases a un curso con una clase agregada	143
A.8.	Software de Asistencia Universidad de Talca: Lista de clases de un curso	144
A.9.	Software de Asistencia Universidad de Talca: Modificar una Clase . .	145
A.10.	Software de Asistencia Universidad de Talca: Ver una clase	146

A.11. Excel Plan de Clases Construcción de Software: Unidad 1, semanas 3, 4 y 5	147
A.12. Excel Plan de Clases Construcción de Software: Unidad 1, semanas 4, 5 y 6	148
A.13. Excel Plan de Clases Construcción de Software: Unidad 2, semanas 7, 8 y 9	149
A.14. Excel Plan de Clases Construcción de Software: Unidad 2, semanas 10 y 11 y Unidad 3, semanas 12, 13, 14, 15, 16 y 17	150
A.15. Excel Plan de Clases Construcción de Software: Unidad 3, semana 17	151
B.1. Matriz de Dependencias, Requisitos de Usuario del 1 al 8. Requisitos de Sistema del 1 al 35	153
B.2. Matriz de Dependencias, Requisitos de Usuario del 9 al 17. Requisitos de Sistema del 1 al 35	154
B.3. Matriz de Dependencias, Requisitos de Usuario del 1 al 8. Requisitos de Sistema del 36 al 57.	155
B.4. Matriz de Dependencias, Requisitos de Usuario del 9 al 17. Requisitos de Sistema del 36 al 57.	156
B.5. Diagrama de Casos de Uso Nivel 0: Competencia	157
B.6. Diagrama de Casos de Uso Nivel 0: Aprendizaje	158
B.7. Diagrama de Casos de Uso Nivel 0: Saber	159
B.8. Diagrama de Casos de Uso Nivel 0: Curso	160
B.9. Diagrama de Casos de Uso Nivel 0: Unidad	161
B.10. Diagrama de Casos de Uso Nivel 0: Actividades	162
B.11. Diagrama de Casos de Uso Nivel 0: Usuario	163
B.12. Diagrama de Casos de Uso Nivel 0: Otros Procesos	164
C.1. Primer prototipo: iniciar sesión	166
C.2. Primer prototipo: mis cursos	167
C.3. Primer prototipo: modificar cursos	168
C.4. Primer prototipo: lista de cursos	169
C.5. Primer prototipo: asignar competencias a cursos	170
C.6. Primer prototipo: modificar cursos, administrador	171
C.7. Primer prototipo: nuevo curso	172
C.8. Primer prototipo: unidades	173

C.9. Primer prototipo: modificar unidad	174
C.10. Primer prototipo: crear unidad	175
C.11. Primer prototipo: actividades	176
C.12. Primer prototipo: crear actividad	177
C.13. Primer prototipo: modificar actividad	178
C.14. Primer prototipo: lista de competencias	179
C.15. Primer prototipo: lista de aprendizajes	180
C.16. Primer prototipo: asignar aprendizajes a competencias	181
C.17. Primer prototipo: lista de saberes	182
C.18. Segundo prototipo: iniciar sesión	183
C.19. Segundo prototipo: ver cursos	184
C.20. Segundo prototipo: ver usuarios	185
C.21. Segundo prototipo: crear un usuario	186
C.22. Segundo prototipo: ver competencias	187
C.23. Segundo prototipo: ver una competencia al detalle	188
C.24. Segundo prototipo: crear un curso	189
C.25. Segundo prototipo: ver aprendizajes	190
C.26. Segundo prototipo: crear un aprendizaje	191
C.27. Segundo prototipo: asignar competencias a cursos	192
C.28. Segundo prototipo: asignar aprendizaje a competencias	193
C.29. Segundo prototipo: ver saberes	194
C.30. Segundo prototipo: crear un saber	195
C.31. Segundo prototipo: asignar saberes a aprendizajes	196
D.1. CN01 Iniciar sesión, Entrada - Salida	217
D.2. CN02 Crear un curso, Entrada - Salida	218
D.3. CN03 Modificar curso, Entrada - Salida	218
D.4. CN04 Deshabilitar curso, Entrada - Salida	219
D.5. CN05 Habilitar curso, Entrada - Salida	219
D.6. CN06 Crear curso por periodo, Entrada - Salida	220
D.7. CN07 Modificar curso por periodo, Entrada - Salida	220
D.8. CN08 Deshabilitar curso por periodo, Entrada - Salida	221
D.9. CN09 Habilitar curso por periodo, Entrada - Salida	221
D.10. CN10 Crear unidad, Entrada - Salida	222

D.11.CN11 Modificar unidad, Entrada - Salida	222
D.12.CN12 Deshabilitar unidad, Entrada - Salida	223
D.13.CN13 Habilitar unidad, Entrada - Salida	223
D.14.CN14 Crear actividad, Entrada - Salida	224
D.15.CN15 Modificar actividad, Entrada - Salida	224
D.16.CN16 Deshabilitar actividad, Entrada - Salida	225
D.17.CN17 Habilitar actividad, Entrada - Salida	225
D.18.CN18 Crear competencia, Entrada - Salida	226
D.19.CN19 Modificar competencia, Entrada - Salida	226
D.20.CN20 Deshabilitar competencia, Entrada - Salida	227
D.21.CN21 Habilitar competencia, Entrada - Salida	227
D.22.CN22 Crear aprendizaje, Entrada - Salida	228
D.23.CN23 Modificar aprendizaje, Entrada - Salida	228
D.24.CN24 Deshabilitar aprendizaje, Entrada - Salida	229
D.25.CN25 Habilitar aprendizaje, Entrada - Salida	229
D.26.CN26 Crear saber, Entrada - Salida	230
D.27.CN27 Modificar saber, Entrada - Salida	230
D.28.CN28 Deshabilitar saber, Entrada - Salida	231
D.29.CN29 Habilitar saber, Entrada - Salida	231
D.30.CN30 Asignar profesor a curso, Entrada - Salida	232
D.31.CN31 Desasignar profesor a curso, Entrada - Salida	232
D.32.CN32 Asignar alumno a curso, Entrada - Salida	233
D.33.CN33 Desasignar alumno a curso, Entrada - Salida	233
D.34.CN34 Crear usuario, Entrada - Salida	234
D.35.CN35 Modificar usuario, Entrada - Salida	234
D.36.CN36 Deshabilitar usuario, Entrada - Salida	235
D.37.CN37 Habilitar usuario, Entrada - Salida	235
D.38.CN38 Recuperar contraseña, Entrada - Salida	236
D.39.CN39 Cambiar Contraseña, Entrada - Salida	236
D.40.CN40 Guardar plan de clases, Entrada - Salida	237
D.41.Pruebas de usabilidad: Instrucciones Generales	238
D.42.Pruebas de usabilidad: Cuestionario pre-evaluación Parte 1	239
D.43.Pruebas de usabilidad: Cuestionario pre-evaluación Parte 2	240
D.44.Pruebas de usabilidad: Cuestionario evaluación Parte 1	241

D.45.Pruebas de usabilidad: Cuestionario evaluación Parte 2	242
D.46.Pruebas de usabilidad: Cuestionario evaluación Parte 3	243
D.47.Pruebas de usabilidad: Cuestionario evaluación Parte 4	244
D.48.Pruebas de usabilidad: Cuestionario evaluación Parte 5	245
D.49.Pruebas de usabilidad: Cuestionario evaluación Parte 6	246
D.50.Pruebas de usabilidad: Cuestionario evaluación Parte 7	247
D.51.Pruebas de usabilidad: Cuestionario evaluación Parte 8	248
D.52.Pruebas de usabilidad: Cuestionario evaluación Parte 9	249
D.53.Pruebas de usabilidad: Cuestionario post-evaluación Parte 1	250
D.54.Pruebas de usabilidad: Cuestionario post-evaluación Parte 2	251
D.55.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 1)	252
D.56.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 2)	253
D.57.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 3)	254
D.58.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 4)	255
D.59.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 5)	256
D.60.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 6)	257
D.61.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 7)	258
D.62.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 8)	259
D.63.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 9)	260
D.64.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 10)	261
D.65.Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 11)	262
D.66.Test de evaluación: Gráfico de likert ver los cursos	263
D.67.Test de evaluación: Gráfico de likert cambiar contraseña	264
D.68.Test de evaluación: Gráfico de likert crear una competencia, un aprendi- zaje y dos saberes	265
D.69.Test de evaluación: Gráfico de likert agregar un aprendizaje a una competencia y los saberes a un aprendizaje	266
D.70.Test de evaluación: Gráfico de likert crar un curso, una unidad y algunas actividades	267
D.71.Test de evaluación: Gráfico de likert asociar una competencia a un curso	268
D.72.Test de evaluación: Gráfico de likert asociar el saber de un curso a una unidad y a la o las actividades	269
D.73.Test de evaluación: Gráfico de likert configurar periodo y año	270
D.74.Test de evaluación: Gráfico de likert guardar plan de clases	271

ÍNDICE DE TABLAS

	página
4.1. Requisitos de Usuario: Generales	48
4.2. Requisitos de Usuario: Módulos, Unidades y Actividades	48
4.3. Requisitos de Usuario: Módulos, Unidades y Actividades	49
4.4. Requisitos de Usuario: Competencias, Aprendizajes y Saberes	49
4.5. Requisitos de Usuario: Competencias, Aprendizajes y Saberes	50
4.6. Requisitos de Sistema: Generales	50
4.7. Requisitos de Sistema: Generales	51
4.8. Requisitos de Sistema: Generales	52
4.9. Requisitos de Sistema: Módulos	52
4.10. Requisitos de Sistema: Módulos	53
4.11. Requisitos de Sistema: Unidades	53
4.12. Requisitos de Sistema: Unidades	54
4.13. Requisitos de Sistema: Actividades	54
4.14. Requisitos de Sistema: Actividades	55
4.15. Requisitos de Sistema: Competencias	55
4.16. Requisitos de Sistema: Aprendizajes	56
4.17. Requisitos de Sistema: Saberes	56
4.18. Requisitos de Sistema: Saberes	57
4.19. Restricciones de Software	58
4.20. Restricciones de Software	59
4.21. Restricciones de Interfaz	59
6.1. Iteración 1	85
6.2. Iteración 2	85
6.3. Iteración 3	86
6.4. Iteración 4	87
6.5. Iteración 5	87
6.6. Iteración 5	88
6.7. Iteración 6	88
6.8. Iteración 6	89
6.9. Iteración 7	89

6.10. Iteración 7	90
6.11. Iteración 8	90
6.12. Iteración 8	91
6.13. Iteración 9	91
6.14. Iteración 9	92
6.15. Iteración 10	92
6.16. Iteración 10	93
6.17. Iteración 11	93
6.18. Iteración 11	94
6.19. Iteración 12	95
6.20. Iteración 13	96
6.21. Iteración 14	97
6.22. Iteración 15	98
6.23. Iteración 16	98
6.24. Iteración 17	98
6.25. Iteración 17	99
6.26. Iteración 18	99
6.27. Iteración 19	100
6.28. Iteración 20	100
6.29. Iteración 21	101
6.30. Iteración 22	101
6.31. Iteración 23	101
6.32. Iteración 24	102
6.33. Iteración 25	102
6.34. Iteración 26	102
6.35. Iteración 27	103
6.36. Iteración 28	103
6.37. Iteración 29	104
6.38. Iteración 30	104
6.39. Iteración 31	105
6.40. Iteración 32	105
6.41. Iteración 33	106
7.1. Caja Negra - Iniciar sesión	117

7.2. Caja Negra - Guardar plan de clases	118
D.1. Caja Negra - Iniciar sesión	197
D.2. Caja Negra - Crear un curso	198
D.3. Caja Negra - Modificar curso	198
D.4. Caja Negra - Deshabilitar curso	199
D.5. Caja Negra - Habilitar curso	199
D.6. Caja Negra - Crear curso por periodo	200
D.7. Caja Negra - Modificar curso por periodo	200
D.8. Caja Negra - Deshabilitar curso por periodo	201
D.9. Caja Negra - Habilitar curso por periodo	201
D.10. Caja Negra - Crear unidad	202
D.11. Caja Negra - Modificar unidad	202
D.12. Caja Negra - Deshabilitar unidad	203
D.13. Caja Negra - Habilitar unidad	203
D.14. Caja Negra - Crear actividad	204
D.15. Caja Negra - Modificar actividad	204
D.16. Caja Negra - Deshabilitar actividad	205
D.17. Caja Negra - Habilitar actividad	205
D.18. Caja Negra - Crear competencia	206
D.19. Caja Negra - Modificar competencia	206
D.20. Caja Negra - Deshabilitar competencia	207
D.21. Caja Negra - Habilitar competencia	207
D.22. Caja Negra - Crear aprendizaje	208
D.23. Caja Negra - Modificar aprendizaje	208
D.24. Caja Negra - Deshabilitar aprendizaje	209
D.25. Caja Negra - Habilitar aprendizaje	209
D.26. Caja Negra - Crear saber	210
D.27. Caja Negra - Modificar saber	210
D.28. Caja Negra - Deshabilitar saber	211
D.29. Caja Negra - Habilitar saber	211
D.30. Caja Negra - Asignar profesor a curso	212
D.31. Caja Negra - Desasignar profesor a curso	212
D.32. Caja Negra - Asignar alumno a curso	213

D.33.Caja Negra - Desasignar alumno a curso	213
D.34.Caja Negra - Crear usuario	214
D.35.Caja Negra - Modificar usuario	214
D.36.Caja Negra - Deshabilitar usuario	215
D.37.Caja Negra - Habilitar usuario	215
D.38.Caja Negra - Recuperar contraseña	216
D.39.Caja Negra - Cambiar Contraseña	216
D.40.Caja Negra - Guardar plan de clases	217

RESUMEN

La metodología curricular basada en competencias de la Universidad de Talca se basa en un curriculum de formación académica y profesional compuesto de en promedio 300 ECTS. Gracias a un análisis entre las empresas y la institución, se puede construir un currículum fundamentado en el trabajo y las necesidades del mercado.

Actualmente, la forma de elaborar un plan de clases es usando una planilla de Excel, y a pesar de que existe un formato estándar, éste no se encuentra validado. Según el comité curricular, el tiempo promedio de creación de un plan de clases es aproximadamente de cuatro horas, en el que cada profesor debe elegir con base en el syllabus del módulo una lista de saberes que debe asignar a cada actividad de cada unidad.

En consecuencia, nace la oportunidad de crear un prototipo que se encargue de gestionar el proceso de creación de un plan de clases basado en una plataforma web; en el que se puede gestionar las competencias, saberes y aprendizajes de un curso de forma mucho más simple. Se podrá crear y mantener un plan de clases, sus unidades y actividades, y podrá asociarse los saberes pertinentes para guardar el plan de clases ya creado como pdf.

Con respecto al desarrollo del prototipo, se usa la metodología de desarrollo tradicional iterativa; buscando obtener un mayor control sobre los requerimientos del software, además de proporcionar un producto que el cliente pueda juzgar a lo largo de las iteraciones.

Se especifica el diseño lógico y físico de la plataforma, con el objetivo de entender el entorno en el que el prototipo se verá inmerso e identificar los módulos que forman parte de éste, a través de diagramas de casos de uso y entidad relación. Estos diagramas impactan en el desarrollo, debido a que proporcionan una mejor perspectiva de cómo el prototipo es y facilita una construcción de software sencilla.

Finalmente, se evalúa que el prototipo cumpla con los requisitos mínimos solicitados por el cliente mediante el uso de pruebas de caja negra. Además, para comprobar que los objetivos planteados de esta memoria se cumplan y para poder tener una perspectiva real del usuario, se realizaron pruebas de usabilidad.

1. Introducción

Este capítulo tiene como propósito el orientar al lector sobre el contexto en el que el problema se desarrolla, los objetivos a cumplir, el alcance y la propuesta de solución.

1.1. Descripción del Contexto

Desde el año 2004, la Universidad de Talca inició un proceso de una transformación de carreras de pregrado en el llamado **Proyecto de Rediseño Curricular Basado en Competencias**, orientado a generar un valor distintivo en sus egresados a través del desarrollo de competencias, valores y una ciudadanía activa. Se creó una metodología basada en un cierto número de competencias en base a DACUM [27, 22]. DACUM entrega una propuesta metodológica de diseño curricular basada en un perfil profesional dictado por las competencias y capacidades asociadas que pretenden disminuir la brecha entre la universidad y las empresas [27].

La Universidad de Talca pretende visualizar el proceso de enseñanza a partir de la aplicación de conocimientos, habilidades y actitudes en base a una profesión específica. Para ello, el currículum de formación académica y profesional fue dividido en tres líneas de formación, que se organizaron en base a las competencias comprometidas en el perfil de egreso y los procedimientos y actitudes de los futuros profesionales: saber, saber hacer y saber ser.

Los currículos de formación académica y profesional están contenidos en líneas de módulos en trayectorias; módulos que son impartidos durante el año en diferentes rangos temporales (anuales, semestrales, etc). Cada uno de los módulos contiene un conjunto de competencias, aprendizajes y saberes en los cuales se espera que el

estudiante se habilite durante el tiempo que éste se imparte. La planificación del módulo es expresada en un syllabus y en un plan de clases. El plan de clases debe establecer las actividades asociadas a unidades de aprendizaje entorno a un rango de tiempo. Según el Modelo Educativo Universidad de Talca [3], *"dan cuenta de la contribución del módulo al perfil de egreso, es decir, lo que específicamente se espera del módulo, así como los métodos y sistemas de evaluación que serán implementados"*. Poder saber si los módulos impartidos realmente son una contribución al perfil de egreso es una cuestión de importancia para la Metodología Curricular Basada en Competencias, así como lo es también el poder validar que las competencias asignadas a los módulos y a los currículos de formación académica y profesional realmente sean aprendidas por los alumnos y reflejadas en el plan de clases.

1.2. Definición del Problema

Según la cultura de evaluación de la Universidad de Talca, es necesario realizar un seguimiento y monitoreo constante del modelo educativo con el fin de asegurar su calidad como establecimiento. Esto implica una evaluación docente, el desarrollo del aprendizaje y competencias de los estudiantes, la gestión curricular y la satisfacción del perfil de egreso del estudiante. Con respecto al desarrollo del aprendizaje y las competencias, son los profesores quienes realizan un diagnóstico y seguimiento de éstas [3].

Si bien la elaboración de un plan de clases es esencial para establecer actividades y tareas relacionadas a unidades de aprendizaje que reflejan la propuesta formativa de cada carrera [3], no existe ningún sistema que facilite la creación de éste, además de una tablilla en Excel.

Si bien la Universidad establece una instancia de evaluación del modelo educativo, esta está enfocada sólo en la evaluación docente (no de los módulos, ni de los conocimientos que se reciben en general). Cabe decir que la evaluación docente es genérica y no posee un enfoque específico sobre cada carrera o sus módulos. La única posibilidad para emitir una opinión por parte de los estudiantes sobre un módulo es en los comentarios, y éstos rara vez son tomados en cuenta.

Eventualmente, esto podría traducirse en que la calidad de un curso pudiese bajar, y en consecuencia, los alumnos podrían no estar habilitados en las competencias establecido por el perfil de egreso, ya que podrían no haber adquirido los aprendizajes

y saberes necesarios para el curso. A largo plazo, esto redundaría en bajar la calidad de la universidad en términos de docencia y calidad de aprendizaje al no cumplir con el perfil de egresado.

Para poder evaluar el modelo educativo, es necesario elaborar un sistema web que permita facilitar la elaboración de un plan de clases. Un sistema de esta naturaleza ofrece múltiples prestaciones, entre ellas: poder mantener un registro de las competencias, actividades y saberes asociadas a un curso y el poder modificarlos de forma simple cuando los currículos sean actualizados; además de permitir la creación un plan de clases que pueda ser consultado durante el tiempo; y a su vez, esto último otorga al alumnado y al cuerpo docente de la Universidad mayor claridad sobre si los saberes están realmente siendo aplicados en las actividades del curso en un periodo o no.

1.3. Objetivos

1.3.1. Objetivo General

Reducir el tiempo de creación de un plan de clases de la carrera Ingeniería Civil en Computación.

1.3.2. Objetivos Específicos

- Automatizar la creación y actualización de un plan de clases de la carrera Ingeniería Civil en Computación de manera sencilla para el usuario.
- Disminuir el tiempo de creación de un plan de clases de la carrera Ingeniería Civil en Computación.
- Permitir de manera sencilla realizar cambios en el plan de clases de un curso de la carrera Ingeniería Civil en Computación.
- Diseñar un conjunto de pruebas preestablecidas en conjunto con los usuarios.

1.4. Alcances

- Se considera una plataforma web.

- El sistema tiene contemplado los roles de administrador, profesor y alumno. No se pueden crear nuevos roles.
- Sólo el administrador puede crear cuentas de usuario.
- La creación del plan de clases se basa en el modelo educativo basado en competencias adoptado por la Universidad de Talca.
- El formato del plan de clases puede ser fácilmente modificado por un mantenedor. Sin embargo, es dependiente de la **Metodología Curricular Basada en Competencias** de la que se hablará más adelante.

1.5. Propuesta de Solución

Para solucionar el problema, se ha propuesto la elaboración de un sistema web que facilite al cuerpo docente la elaboración y evaluación de un plan de clases. El plan de clases debe contemplar:

- Gestión de Competencias, Saberes y Aprendizajes: Funcionalidad que permite crear y mantener un registro de las competencias que provienen del currículo de formación académica y profesional. El currículo se traduce en una serie de módulos (o asignaturas en un modelo tradicional). Cada módulo cuenta con un syllabus, en que se especifica formalmente las competencias, saberes y aprendizajes que le competen. Naturalmente, la secuencia de módulos se orquesta de forma tal de dar cuenta completa del perfil de egreso. El plan de clases depende de los propósitos expresados en el syllabus y el syllabus representa la propuesta formativa de la carrera, es decir, aquellas competencias, aprendizajes y saberes que se esperan del módulo. No se puede crear un plan de clases sin el contenido del syllabus de un módulo.
- Gestión de Módulos: Funcionalidad que permite crear y mantener un registro de un plan de clases, tanto de los módulos y sus unidades, como las actividades relacionadas, los encargados (profesores) de un periodo establecido, y alumnos inscritos en el módulo. Se pretende suplantarse la creación manual de un plan de clases por una semi-automática.

1.6. Descripción de contenidos

El resto de este documento de memoria se organizó de la siguiente manera:

- El capítulo dos detalla el marco teórico, información necesaria para que pueda entenderse el contexto en el que esta memoria se desenvuelve.
- El capítulo tres, se hace referencia a la metodología de trabajo utilizada para realizar el proyecto, además de la forma en la que se organizó la planificación del prototipo y del documento en sí.
- En cuanto al capítulo cuatro, se hace alusión a los requisitos del proyecto, necesarios para poder entender las necesidades del cliente y así poder comenzar a construir una solución en respuesta.
- En el capítulo cinco se detalla el diseño de la aplicación. Comenzando con la arquitectura física, la arquitectura lógica y siguiendo con el diagrama de clases y el Entidad-Relación; indispensables para sintetizar y representar los requerimientos y así poder dividir las secuencias de procesamiento y la forma en la cual la información será representada.
- El capítulo seis se enfoca en el desarrollo e implementación del prototipo. Aquí es donde se habla de las principales decisiones tomadas para poder facilitar la construcción de éste, así como la descripción del prototipo final.
- El capítulo seis menciona las pruebas utilizadas sobre el prototipo finalmente construido, así como los resultados obtenidos tras el uso de la aplicación por los usuarios.
- Y por último, el capítulo ocho corresponde a las conclusiones con base en el trabajo y las pruebas realizadas. Además, se menciona el trabajo futuro a desarrollar.

2. Marco Teórico

En este capítulo se detalla el marco teórico que da soporte a esta memoria. Su objetivo es brindar al lector la información suficiente como para que éste pueda entender el contexto en el cual se desenvuelve el problema. Esto incluye al modelo educativo de la Universidad de Talca, que detalla las características del currículo de formación académica y profesional, los roles del mercado laboral, del cuerpo docente y del alumnado en él, las metodologías de enseñanza y aprendizaje y el proceso de conformación de un plan de clases.

También se presentan las tecnologías elegidas para la construcción del prototipo y las metodologías de trabajo y de pruebas.

2.1. Metodología Curricular Basada en Competencias

El modelo de la Metodología para el diseño de un currículo orientado a las competencias propuesto por la Universidad de Talca se basa en un currículo de formación académica y profesional, compuesto de en promedio 300 ECTS¹ [3]. Para lograr identificar los perfiles profesionales a formar se utiliza la metodología DACUM², que analiza ocupaciones y procesos de trabajo. Según en trabajo titulado: Una metodología para el diseño de un currículo orientado a las competencias [27] “*DACUM genera insumos para conducir procesos de análisis funcional y poner en práctica una relación más estrecha entre la institución formadora y las empresas*”. Es decir, gracias al análisis de DACUM sobre los procesos productivos empresariales y las tareas derivadas, se puede construir un currículo basado en el trabajo y las necesidades reales del mercado.

¹Sistema Europeo de Transferencia y Acumulación de Créditos

²Developing A CurriculUM

En la metodología propuesta para el diseño curricular, centrada en la conformación de la malla curricular, ilustrado en las Figuras A.1, A.2, 2.1, 2.2, 2.3, 2.4 y 2.5, se representa el perfil profesional dado por las competencias y aprendizajes o capacidades resultantes de la metodología DACUM. La Universidad de Talca optó por una metodología que se enfoca no sólo en las capacidades cognitivas (saber conocer), sino que también en las procedimentales (saber hacer) y actitudes personales e interpersonales (saber ser y saber estar) [27].

2.1.1. Fase A

Como se puede observar en la Figura 2.1, a partir de las competencias identificadas en el mercado laboral que la Universidad de Talca espera satisfacer, se encuentran asociados uno o más módulos responsables de la competencia correspondiente, módulos que se encuentran divididos en tres líneas de formación curricular: la formación fundamental, la formación básica y la formación disciplinar [27]. Donde:

- La **formación básica** provee el conocimiento fundamental en relación a las ciencias en las que la profesión o disciplina se basa.
- La **formación fundamental** provee el desarrollo de competencias de formación ciudadana, de desarrollo personal y para el trabajo académico.
- Finalmente, la **formación disciplinar** provee conocimiento de la profesión o disciplina, buscando que el estudiante desarrolle habilidades y actitudes en relación al perfil profesional.

Estos módulos pueden modificarse con el propósito de poder facilitar su reconocimiento y la comunicación con el entorno académico-profesional. La formación básica y la fundamental conducen a la obtención del grado de licenciado, mientras que la formación disciplinar conduce a la obtención del título profesional [3].

La importancia de definir competencias asociadas a un módulo es que, una vez finalizado éste, son las competencias definidas por el sector del mercado laboral donde se desempeñará el profesional quienes especifican las exigencias para la aprobación de un módulo [27]. Cabe recalcar que es el mercado laboral quien da el producto esperado y los saberes y aprendizajes requeridos para el logro de la competencia; mientras que los contenidos de los módulos se encuentran a cargo de los docentes.

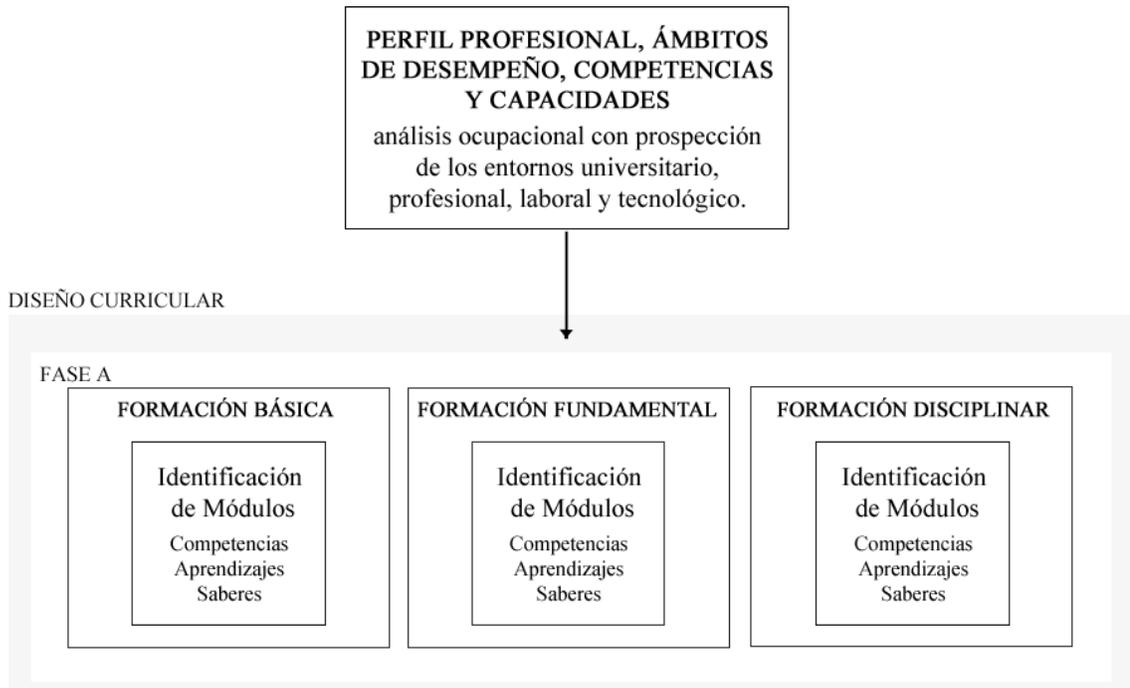


Figura 2.1: Diagrama de Metodología para el Diseño Curricular, Fase A.

2.1.2. Fase B

Siendo identificadas las competencias y como se explicó en la Sección 2.1.1, se necesita establecer el orden por el cual los módulos serán cursados [27]. Es necesario asumir que los módulos pueden tener de cero a más de un prerrequisito y/o correquisito como se puede vislumbrar en el Anexo A.3 (el plan curricular de Ingeniería Civil en Computación) y a grandes rasgos en la Figura 2.2. Este ordenamiento permite asignar los saberes y aprendizajes requeridos para cumplir el perfil de egreso en cada uno de los módulos de la malla curricular del programa específico.

2.1.3. Fase C

Una vez identificados los saberes y aprendizajes que son requeridos para lograr una competencia asociada a un módulo, como se explicó en la Sección 2.1.2, se debe estructurar cada uno de éstos. En la Fase C, ilustrada en la Figura 2.3, es donde por cada módulo se deben establecer los contenidos que son abordados, una estimación de tiempo y los recursos que puedan ser necesarios (bibliografía básica, infraestructura, etc.). Además, debe revisarse cada módulo o competencia, en caso de que en ambos

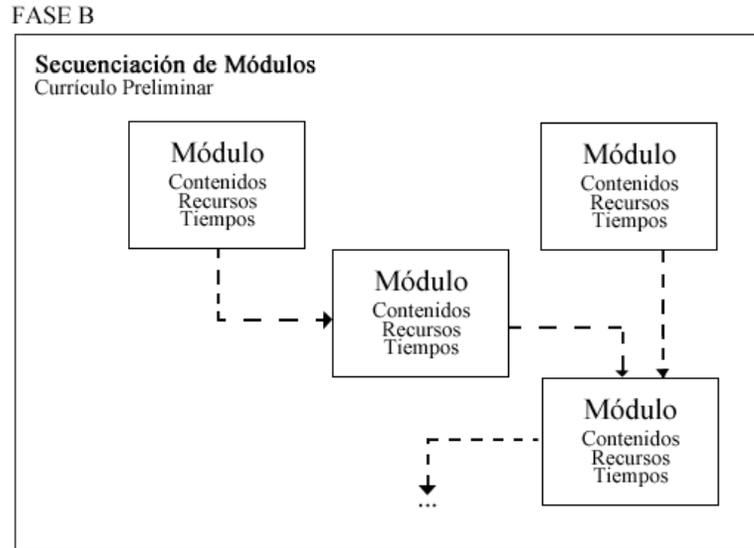


Figura 2.2: Diagrama de Metodología para el Diseño Curricular, Fase B.

se abarque mucho o muy poco y sea necesario el redefinirlos. Debe intentarse de que la relación entre módulo y competencia sea uno a uno, no obstante, la metodología curricular basada en competencias contempla el hecho de que un módulo contenga a múltiples competencias [27] y que la misma competencia será abordada en múltiples módulos, pero con un distinto grado de avance.

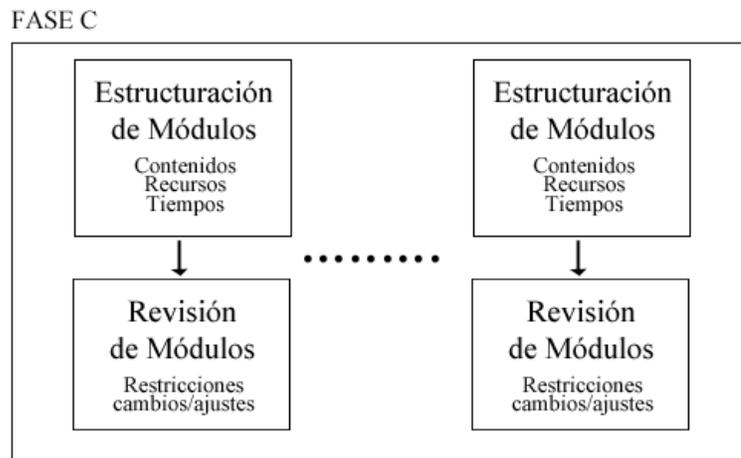


Figura 2.3: Diagrama de Metodología para el Diseño Curricular, Fase C.

Estableciéndose las competencias y los módulos correspondientes con sus tiempos y su bibliografía, es necesario verificar que los tiempos establecidos estén dentro de

las restricciones vigentes impuestas por la Universidad de Talca. El tiempo total de cada módulo debe ser ajustado al régimen curricular de la Universidad (semestral) y la suma de los tiempos que los alumnos deben destinar a los módulos no debe exceder al tiempo total asignado. Estos tiempos, inicialmente expresados en horas semanales, son transformados en créditos ECTS³ [27]. Actualmente, año 2019, un ECTS equivale a una hora y media por semana.

2.1.4. Fase D

Si bien en la Fase C, detallada en la Sección 2.1.3, se debe verificar que los tiempos de cada módulo estén dentro de las restricciones definidas, en la Fase D, ilustrado en la Figura 2.4, se realiza una revisión exhaustiva del tiempo empleado en el plan académico. Es necesario verificar que exista tanto un balance en los tiempos de cada periodo académico como la viabilidad del logro de las competencias que se consignaron en el perfil profesional definido. En tal caso, es necesario volver a generar competencias, volver a desplazar módulos, cambiar los pre y correquisitos, volviendo a la Fase A [27].

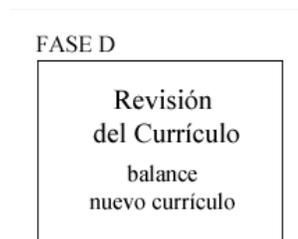


Figura 2.4: Diagrama de Metodología para el Diseño Curricular, Fase D.

2.1.5. Fase E

Una vez que el plan curricular fue aceptado, a partir de los contenidos y las competencias que se aspiran a lograr por cada módulo se debe comenzar a elaborar el Syllabus para cada uno de ellos, como se ilustra en la Figura 2.5. Es decir, por cada módulo es necesario que exista un syllabus [27]. En la Universidad de Talca, el syllabus corresponde a un documento oficial que expresa el consenso de la escuela

³En español, Sistema Europeo de Transferencia y Acumulación de Créditos

correspondiente al perfil profesional establecido, que contiene: la organización de las unidades de aprendizaje, las actividades y los contenidos asociados a estas unidades, los recursos que son involucrados para cada actividad y los tiempos relacionados. En plan de clases es creado a partir de la organización explícita de las actividades en ciertos espacios de tiempo requeridos para cada una de las unidades de aprendizaje [3].

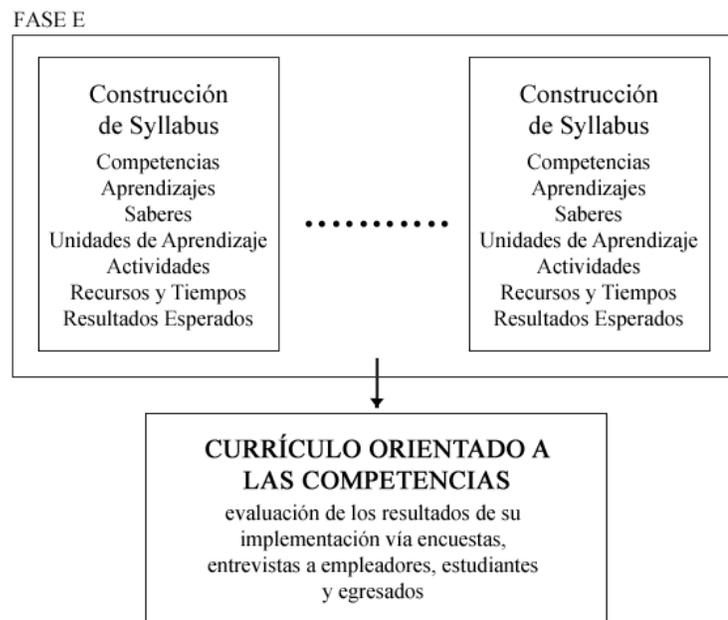


Figura 2.5: Diagrama de Metodología para el Diseño Curricular, Fase E.

Finalmente, una vez creado el currículo, es necesario mantener una evaluación constante para poder ajustar el currículo a nuevas restricciones (como la incorporación de nuevas competencias o la eliminación de alguna de ellas). En tal caso, en cualquiera de las fases del modelo está la posibilidad de “volver atrás”, sobre todo si las modificaciones requeridas son esenciales [27].

2.2. Tecnologías Utilizadas

En esta sección se presentan las tecnologías utilizadas, con la finalidad de poder introducir al lector sobre los principales motivos por los cuales éstas fueron elegidas para el desarrollo del prototipo.

2.2.1. Estación de Trabajo

La estación de trabajo utilizada para la elaboración del proyecto corresponde a un computador con Intel Core i7 de 2.80GHz, 12Gb de RAM, con conexión a internet y sistema operativo Windows 10 Home. Se desconoce los computadores que se utilizan para realizar las pruebas de usabilidad, sin embargo las pruebas de caja negra se realizan en esta estación. En relación con los navegadores utilizados se consideran Mozilla Firefox e Microsoft Edge.

La estación de trabajo utilizada para el montaje y puesta en marcha del proyecto corresponde a un servidor en Amazon Web Services que cuenta con 1 vCPU⁴, 1GB de RAM, 8GB de disco SSD⁵ y sistema operativo Amazon Linux 2.

2.2.2. Frameworks

A continuación, se describen los frameworks utilizados en esta memoria.

ASP.NET Core

ASP.NET es un framework que permite construir aplicaciones Web. Las aplicaciones pueden ser escritas en cualquier lenguaje compatible con Common Language Runtime, en este caso C#. Las páginas web ASP.NET pueden ser solicitadas en cualquier explorador o dispositivo del cliente, generando de forma dinámica páginas compatibles con dispositivos preparados para trabajar en Web como celulares, computadores entre otros [14]. Si bien existen otras tecnologías que podían resolver el problema (como Ruby on Rails), se eligió ASP.NET Core porque es multiplataforma, además de poseer una infraestructura ligera capaz de poder ejecutarse en cualquier entorno. También por la experiencia en el uso de este framework. A diferencia de ASP.NET Framework, ASP.NET Core es un Framework de código abierto que tiene las siguientes características:

- Es compatible con Windows, macOS y Linux.
- Usa páginas Razor, que permite que la codificación de interfaces de usuario web sea más sencilla y productiva.

⁴Procesador virtual

⁵Unidad de estado sólido

- Utiliza el patrón Modelo-Vista-Controlador, que permite pruebas en las aplicaciones web.
- Tiene la capacidad de hospedarse en IIS, Nginx, Apache, Docker o de auto hospedarse en su propio proceso[16].

Bootstrap

Bootstrap es un conjunto de herramientas de código abierto que permite desarrollar sitios y aplicaciones web adaptables y móviles gracias al uso de HTML, CSS y Javascript. Actualmente existen variadas plantillas de diseño con formularios, botones, menús de navegación, entre otros elementos, siendo compatible con la mayoría de los navegadores [2]. Se utiliza esta tecnología, debido a que es empleada en múltiples aplicaciones y para el usuario final iba a resultar mucho más agradable el poder manejar un prototipo con una interfaz conocida.

- **Gentelella:** Gentelella es una plantilla de Bootstrap orientada a la administración, cuyo uso es gratuito. A partir de elementos de Bootstrap 3 en conjunto con una variedad de complementos y herramientas de JQuery se creó un framework para crear paneles de control o administración. Los scripts incluidos en esta plantilla además de Bootstrap y que forman parte de esta memoria son: Font Awesome, FullCalendar, dataTables y Parsley [6]. Un ejemplo de esta plantilla se puede observar en la Figura 2.6. Se opta por utilizar una plantilla administrativa como esta, dado que resultaba mucho más fácil el trabajar con una interfaz ya hecha.

2.2.3. Lenguajes

A continuación, se describen los lenguajes de programación y de marcado utilizados en esta memoria.

C#

C# es un lenguaje de programación orientado a objetos desarrollado por Microsoft. Deriva de C/C++ y utiliza un modelo de orientación a objetos similar al de Java, con mejoras de otros lenguajes. Si bien forma parte de la plataforma .NET,

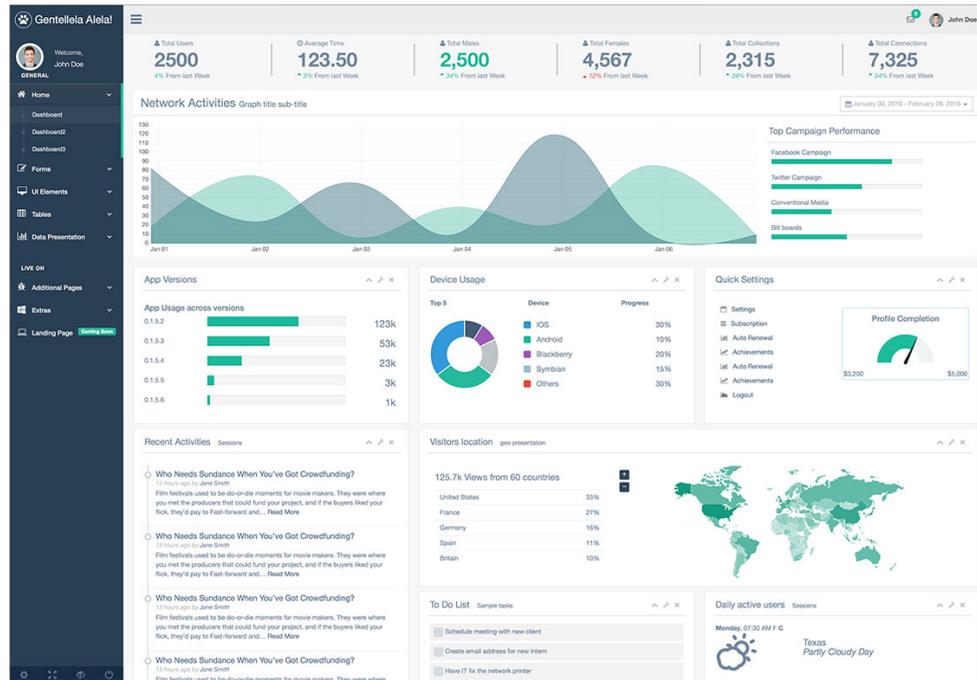


Figura 2.6: Ejemplo de plantilla demo de Gentelella.

C# es un lenguaje de programación independiente, diseñado para generar programas sobre aquella plataforma [9]. Se utiliza para esta memoria debido a que el lenguaje de programación para ASP.NET Core es C#.

HTML-Razor

HTML es un lenguaje de marcado utilizado para la creación de páginas web. Originalmente fue diseñado como un lenguaje que describía semánticamente documentos científicos, pero con el transcurso del tiempo fue adaptado para ser utilizado por otro tipo de documentos o aplicaciones. En vez de incrustarse directamente en el código, se realiza una referencia a la ubicación de éste mediante texto y es tarea del navegador web el interpretar estos elementos y visualizar la página final. HTML pretende que cualquier página web se interprete de la misma forma y por cualquier navegador actualizado [26]. Dado que el prototipo es un sistema web, se utiliza para esta memoria debido a que ASP.NET permite el utilizar HTML junto a otros scripts y componentes. Además, la autora de esta memoria tiene experiencia previa en el uso de este lenguaje de marcado.

Razor es un lenguaje de marcado utilizado para incrustar código en páginas

web. Utiliza el marcado de Razor, C# y HTML. Los archivos cuyo contenido sean del lenguaje poseen una extensión .cshtml. El estándar predeterminado de Razor es HTML, no habiendo grandes diferencias con un archivo HTML propiamente tal, diferenciándose en que Razor admite C# y utiliza el símbolo @ para pasar de HTML a C#. Razor toma aquellas expresiones de C# y las transforma a HTML [17].

CSS

CSS es un lenguaje de diseño gráfico que permite definir y crear la vista de un documento escrito en lenguaje de marcado. Si bien HTML define la estructura y la semántica del contenido, CSS le da un estilo visual a los documentos web e interfaces escritas en HTML o XHTML. El CSS usualmente es utilizado para modificar la fuente, el color, el tamaño y espaciado del contenido, entre otras cosas [19]. En esta memoria fue necesario, a pesar de disponer de una plantilla de Bootstrap robusta, el poder realizar modificaciones para que la interfaz fuese lo más intuitiva para el usuario en lo posible. Según los alcances del proyecto, el prototipo debía ser compatible al menos con un computador al ser una plataforma web. Debido a esto, para poder asegurar la compatibilidad con los diferentes navegadores, fue necesario el utilizar CSS.

JavaScript

JavaScript es un lenguaje de programación interpretado, considerado dinámico. Permite que la interacción entre el cliente y las páginas web sean mucho más dinámicas. También puede utilizarse para interactuar con el servidor (Node.js es un ejemplo de ello) [23]. Dado que, para ciertas funcionalidades es necesario acceder a ellas sin que la página sea actualizada, algo imposible de lograr sólo utilizando HTML-Razor, se utilizó este lenguaje. Además de la experiencia en el uso de ésta.

2.2.4. JQuery

jQuery es una biblioteca multiplataforma de código abierto perteneciente a JavaScript. Permite simplificar la forma de interactuar con los documentos HTML, el manejo de eventos, el desarrollo de animaciones y la interacción con las páginas web mediante el uso de AJAX con una API fácil de usar que funciona en múltiples navegadores web. Se define como versátil y extenso [8]. JQuery fue usado en esta

memoria dado que era necesario que la interacción entre el usuario y algunas librerías fuesen dinámicas.

- **FullCalendar:** Calendario de eventos de código abierto de jQuery. Es completamente personalizable, sin embargo, no es compatible con dispositivos móviles [10]. Se optó por utilizar esta tecnología, debido a lo cómodo que es para el usuario el poder visualizar los eventos en un calendario. Además de ser compatible con la plantilla Gentelella.
- **DataTables:** Complemento de código abierto de jQuery que agrega características avanzadas a cualquier tabla HTML como paginación, búsqueda instantánea y clasificación, entre otros. Para el usuario final es intuitivo, fácil de usar y rápido [11]. Se optó por utilizar esta tecnología, debido a que posee un filtro propio para poder buscar y poder visualizar una tabla de forma sencilla que es una funcionalidad indispensable para el sistema web desarrollado en esta memoria.
- **Parsley:** Biblioteca de código abierto de validación de formularios de Javascript. Proporciona a sus usuarios comentarios sobre el envío de sus formularios antes de procesarlos en el servidor. Si bien no reemplaza a la validación por parte del servidor, proporciona un ahorro de tiempo para el usuario y para el desarrollador, conteniendo todos los tipos de validaciones básicos utilizados [31]. Se opta por utilizar esta tecnología, a diferencia del clásico validador de JQuery, debido a que Parsley dispone de muchas más funciones intuitivas y modificables para la restricción de campos de formulario.

2.2.5. AWS

Amazon Web Services es un conjunto de servicios web que ofrece Amazon.com a través de la web. El servicio que se utiliza en esta memoria, es Amazon EC2⁶ que permitió el usar un servidor virtual (llamado instancia) por horas, proveyendo control sobre localizaciones geográficas; ventaja que permite optimizar la latencia. Para esta memoria y ya que no se poseía un servidor donde ser instalada, se opta por elegir esta tecnología debido a la facilidad que otorga en crear un servidor en minutos y el poder aumentar o disminuir la capacidad dependiendo de los requisitos. Además de

⁶Amazon Elastic Compute Cloud

la experiencia de uso en ella y el hecho de que los usuarios que pudiesen probar la aplicación no se encontrasen en el mismo entorno físico [1].

2.3. Trabajos Relacionados

En esta sección se presentan propuestas o trabajos similares, en el que se lograron identificar dos propuestas relacionadas al prototipo propuesto, las cuales se detallan a continuación.

2.3.1. Software de Asistencia Universidad de Talca

Software creado por la universidad que se encuentra almacenado en el sitio de Ingeniería Utalca⁷ ilustrado en los anexos A.4, A.5, A.6, A.7, A.8, A.9 y A.10. Este software se enfoca principalmente en la gestión de asistencia del alumnado. Permite agregar clases a una instancia de curso dada fechas de inicio y fin, un día de la semana en específico, un bloque de inicio y término y una sala asociada. Lamentablemente, no se encuentra habilitada en el periodo de tiempo en el que se necesita realizar un plan de clases, no captura todas las competencias (más bien ninguna) y no permite imprimir una lista de clases.

2.3.2. Excel de plan de clases

Método tradicional por el cual se crea un plan de clases, ilustrado en las Figuras A.11, A.12, A.13, A.14 y A.15. Consiste en un documento en Excel, que contiene un plan de clases dividido en unidades y semanas, un rango de fecha, actividades y saberes asociados (en este caso, del curso de construcción de Software). Lamentablemente, el proceso de creación de un plan de clases como aquel, según el comité curricular de Ingeniería Civil en Computación, es aproximadamente de una a dos horas si el plan de clases se encuentra completamente armado (debido a la necesidad de ajustar clases, fechas, evaluaciones, horarios de atención y saberes) y de cuatro a seis horas si el plan de clases necesita armarse de cero, sin tomar en cuenta el tiempo de ocio. Además de que la estructura del documento varía según la clase y el profesor.

⁷<http://www.ingenieria.otalca.cl/>

2.4. Metodologías

Una metodología es definida como un conjunto de principios organizativos alrededor de los cuales se pueden responder a las necesidades de un proyecto. Es posible aplicar una variedad de métodos a cualquier problema, ya sea de investigación o no, y a menudo es necesario usar una combinación de métodos para lograr comprender y resolver el problema. La elección de un método, ya sea de desarrollo o de evaluación, depende de los recursos disponibles (participantes como sujetos de prueba), de qué tan cerca esté el método a responder a las preguntas u objetivos planteados y del problema por sí mismo. A continuación, se detallan las dos metodologías elegidas, tanto de desarrollo como de evaluación.

2.4.1. Metodología de Desarrollo: Tradicional Iterativo Incremental

La **metodología de desarrollo tradicional iterativo incremental** es una combinación de partes del modelo en cascada, con una aplicación iterativa. El modelo en cascada, también conocido como el *ciclo de vida clásico*, estipula un enfoque secuencial sobre el desarrollo. Comenzando con la recopilación de requisitos, siguiendo con la planificación, el modelado, la construcción y el despliegue del software. Difícilmente puede esperarse un entregable funcional de manera primordial, lo que puede no contemplar la incertidumbre natural presente en muchos proyectos.

El modelo incremental, intenta aplicar las secuencias lineales del modelo de cascada de manera escalonada con respecto al tiempo. Cada una de las secuencias planificadas debe producir entregables del software, tal y como se puede observar en la Figura 2.7. Se enfoca principalmente en proporcionar un conjunto limitado de funcionalidad que debe refinarse y expandirse en entregas posteriores del software, siguiendo las secuencias del modelo en cascada.

Usualmente, el primer incremento debe ser un *producto esencial*. Esto implica que deben agregarse requisitos iniciales, pero las características extra que pueda tener el software quedan a juicio de una evaluación detallada, o del cliente. Durante cada incremento, se debe cuestionar la modificación del producto principal, de tal forma que las necesidades del cliente puedan quedar satisfechas [24].

Las ventajas que proporciona la metodología de desarrollo tradicional iterativo incremental para poder ser implementada en el sistema son:

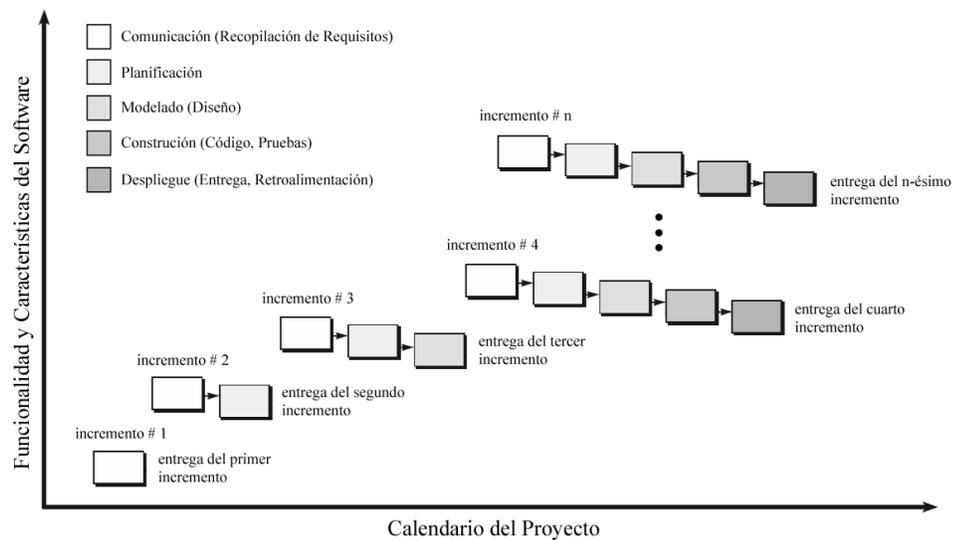


Figura 2.7: Ejemplo del Modelo Incremental.

- Se pueden definir entregables de forma iterativa.
- Se tiene mucho más control sobre los requerimientos del software que el modelo ágil y al mismo tiempo, proporciona una elasticidad ideal para establecer más o corregir éstos.
- El plan de incrementos puede variar dependiendo de las funcionalidades logradas. Sin embargo, a diferencia de la metodología ágil ésta es mucho más rígida, al no existir grandes cambios en torno a las especificaciones de requisitos, sólo en torno al avance.

2.4.2. Metodología de evaluación en Ingeniería de Software: Pruebas de Caja Negra

También denominadas pruebas funcionales, en las pruebas de caja negra se diseñan casos de prueba basándose en información netamente de especificación. El enfoque son las salidas generadas en respuesta a variables de entradas específicas y condiciones de ejecución, tal y como se muestra en la Figura 2.8, buscando validar la funcionalidad general del sistema. Se realiza en función de los requisitos recopilados, por lo que cualquier requisito incompleto o impredecible se vuelve fácilmente detectable. La prueba de caja negra es realizada según la perspectiva del usuario

final, con la implicancia de que el sujeto de pruebas maneja tanto las variables de entrada válidas, como las no válidas. En este entorno no es requerido elegir un sujeto de pruebas de una población definida, sin embargo si es sustancial tener claro el objetivo a alcanzar, así como las variables dependientes e independientes que afectan a la prueba [30].

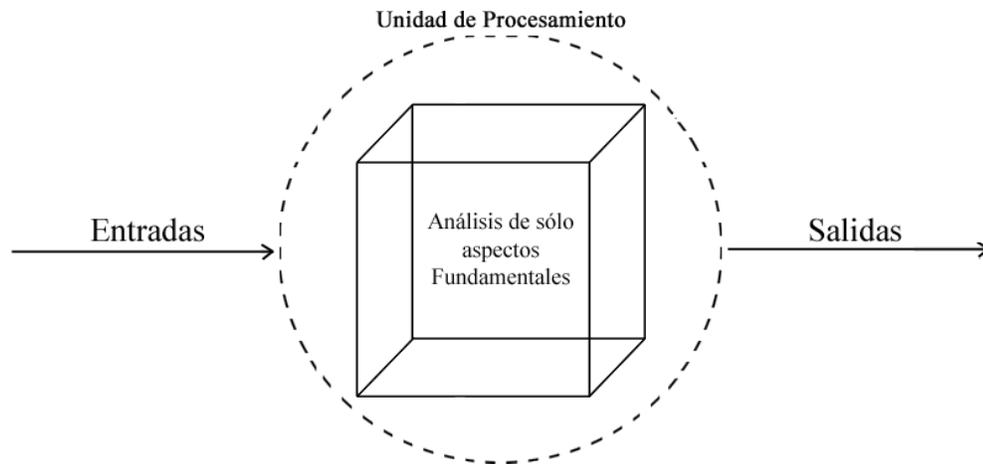


Figura 2.8: Diagrama de representación de una prueba de caja negra.

En las pruebas de caja negra, el sujeto de pruebas no puede ni debe tener acceso al código fuente, donde sólo éste tiene la certeza de que puede ingresar información en cierta instancia y de que obtendrá una respuesta. Se suele realizar desde el inicio de vida del proyecto de software, donde todos los miembros del equipo de pruebas deben participar desde la fase de recopilación y análisis de requisitos del cliente [30]. En la Sección 7.1 se detallan las pruebas de caja negra realizadas al prototipo.

2.4.3. Metodología de evaluación en Ingeniería de Software: Pruebas de Usabilidad

Las pruebas de usabilidad miden el grado en el que los usuarios son capaces de interactuar con la aplicación de forma efectiva y la capacidad del usuario para cumplir con las actividades que éste haya tenido al momento interactuar con la aplicación. Según **la ingeniería del software, un enfoque práctico** [24] “*proporciona retroalimentación relevante y refuerza un enfoque de interacción consistente*”. Una prueba de usabilidad puede realizarse en diferentes niveles; ya sea, evaluar la usabilidad de una funcionalidad específica o de una aplicación completa. La siguiente secuencia de

pasos puede considerarse como un resumen de cómo realizar el diseño de un test de usabilidad según la ingeniería del software, un enfoque específico [24]:

1. Definir un conjunto de categorías de prueba de usabilidad e identificar las metas de cada una.
2. Diseñar pruebas que permiten la evaluación de cada meta.
3. Seleccionar a los participantes que realicen las pruebas.
4. Instrumentar la interacción de los participantes con la aplicación mientras se lleva a cabo la prueba.
5. Desarrollar un mecanismo para valorar la usabilidad de la aplicación.

El primer paso de la prueba es constituir objetivos de prueba para cada categoría a evaluar, como lo son navegación, contenido, utilidad, etc. Para esta memoria, los objetivos de prueba fueron **contenido**, **navegación** y **utilidad**. A partir de cada una de estas categorías se debe diseñar una serie de pruebas, ya sea una revisión visual o una actividad de funcionalidad.

Según **usability 101: introduction to usability** [21] en esta etapa es necesario utilizar usuarios representativos para poder identificar los problemas de usabilidad más importantes. Conforme cada funcionalidad es probada, es necesario que los usuarios la evalúen con una escala cualitativa para poder evaluar la usabilidad de la aplicación. En esta memoria se utilizará una escala de likert.

Escala de Likert

La escala de Likert es una escala psicométrica⁸, en la cual las respuestas en una investigación son calificadas en un rango. Usualmente se consulta a un usuario su acuerdo o desacuerdo para cierto tipo de declaraciones, donde el rango responde a los sentimientos o percepciones del usuario para un artículo determinado. Busca darle un valor cuantitativo a una percepción subjetiva [28].

Normalmente, se utilizan cinco niveles de respuesta, donde se intenta medir una respuesta positiva o negativa para una afirmación. Por lo mismo, es necesario realizar una escala equilibrada, donde el número de respuestas positivas y negativas sea la

⁸Escala que mide atributos mentales

misma. Un ejemplo simple de respuesta en una escala likert sería de la siguiente forma:

1. Muy en desacuerdo
2. En desacuerdo
3. Ni de acuerdo ni en desacuerdo
4. De acuerdo
5. Muy de acuerdo.

En la Sección 7.2 se detallan las pruebas de usabilidad realizadas en esta memoria.

2.5. Resumen

En este capítulo se ha detallado el marco teórico que da soporte a esta memoria. Se escribe sobre la metodología curricular orientada a las competencias, que se basa en un currículo de formación académica y profesional asociado a uno o más módulos divididos en tres líneas de formación curricular: la formación fundamental, la formación básica y la formación disciplinar. También se detallan las tecnologías utilizadas; para así poder introducir al lector sobre los principales motivos por los que éstas fueron elegidas para el desarrollo del prototipo. Adicionalmente se mencionan los trabajos relacionados; tanto modelos curriculares como aplicaciones similares al prototipo. Finalmente, se detallan tanto la metodología de desarrollo, tradicional iterativo incremental, como las metodologías de evaluación, pruebas de caja negra y pruebas de usabilidad.

3. Metodología utilizada

Como se ha mencionado anteriormente, una metodología es definida como un conjunto de principios organizativos alrededor de los cuales se pueden responder a las necesidades de un proyecto. Por este mismo motivo, es necesario elegir una metodología para poder organizar, administrar y asegurarse que el proceso de desarrollo sea eficiente.

3.1. Entorno

Es importante para poder aplicar una metodología, el estar consiente del ambiente en el cual el proyecto se desenvuelve. Esto contempla el análisis de personas, las herramientas utilizadas y la aplicación en sí misma. A continuación, se detallarán estos tres aspectos.

3.1.1. Personas

Los tres roles que se desempeñan en esta memoria son el cliente, la desarrolladora y los interesados. Por lo tanto, al ser un trabajo de una sola persona y al existir actividades adicionales, es necesario planificar las etapas del proyecto con cuidado.

Además, si bien la curva de aprendizaje es baja debido al conocimiento de las tecnologías a aplicar, si existe poca experiencia en la metodología de trabajo lo que podría retrasar el proyecto. Aún así, se cuenta con compromiso, autoaprendizaje y organización para lograr que el proyecto funcione.

3.1.2. La aplicación

Como se mencionó en la Sección 1.2, si bien la elaboración de un plan de clases es esencial, no existe ningún sistema que facilite la creación de éste además de una tablilla en Microsoft Excel, asimismo no se tiene un estándar de plan de clases. Es por ese motivo que los requerimientos del sistema son cambiantes en cuanto a la facilidad de uso y al estándar final, por lo que es altamente probable que éstos cambien a medida que el proyecto avanza. Sin embargo, el objetivo principal se conserva y los nuevos requisitos son evaluados y aceptados siempre y cuando el impacto no sea negativo ni en el tiempo, ni en la visión del cliente.

El prototipo otorga un archivo PDF que contiene un plan de clases, con las actividades y saberes ordenados según las unidades de un curso o módulo determinado. Cabe destacar, que no puede guardarse un plan de clases a menos que existan competencias, aprendizajes y saberes asociados al curso, unidades y actividades correspondientes.

3.1.3. Herramientas

Para poder tener un control de las iteraciones se utiliza un documento en Microsoft Excel, con la finalidad tener un registro de cada una de éstas. Como se muestra en la Figura 3.1, se registran las iteraciones con su número, tarea, fecha inicial, fecha final, duración en días y cumplimiento. Para poder facilitar el término exitoso de la iteración y tener una noción de lo que se debe hacer, ésta se subdividió en tareas como se muestra en la Figura 3.2.

Para el diseño de la arquitectura física y lógica, el diagrama entidad relación, de clases y casos de uso se utiliza una aplicación llamada StarUML, que permite la creación de diagramas de forma gráfica y en el caso de los diagramas entidad relación y de clases, exportar a código.

Con respecto al desarrollo de la aplicación, durante las primeras iteraciones se utiliza el IDE Visual Studio para el proceso de creación del proyecto. Sin embargo, se decidió el cambiar de IDE a VSCode, debido a que Visual Estudio no es compatible con algunas librerías de creación de archivos PDF, lo que es imperativo para el resultado exitoso del proyecto.

Gantt Plan de Clases					
Num	Tarea	Fecha Inicial	Fecha Final	Tiempo Duración	Cumplimiento
Primer Semestre					
0	Establecer proyecto de formulación	22-mar	12-abr	21	
1	Generar documento de requisitos	12-abr	02-may	20	
2	Generar documento de diseño	02-may	14-may	12	
3	Generar CRUD	14-may	28-may	14	
4	CRUD Competencias	28-may	03-jun	6	
5	CRUD Aprendizajes	04-jun	10-jun	6	
6	CRUD Usuarios	11-jun	17-jun	6	
7	CRUD Saberes	18-jun	24-jun	6	
8	CRUD Cursos	25-jun	01-jul	6	
9	CRUD Cursos por Periodo	02-jul	08-jul	6	
10	CRUD Cursos por Periodo 2	09-jul	15-jul	6	
11	CRUD Unidades	16-jul	22-jul	6	
Segundo Semestre					
12	Correcciones faltantes	04-ago	10-ago	6	
11	CRUD Actividades	11-ago	17-ago	6	
12	Interfaz 1	25-ago	31-ago	6	
13	Guardar plan de clases 1	01-sept	07-sept	6	
14	Guardar plan de clases 2	08-sept	14-sept	6	
15	Autenticación	15-sept	21-sept	6	
16	Perfil-Inicio	22-sept	28-sept	6	
17	Autenticación 2	29-sept	05-oct	6	
18	Interfaz 2	06-oct	12-oct	6	
19	Competencias - Aprendizajes	13-oct	19-oct	6	
20	Aprendizaje - Saber	20-oct	26-oct	6	
21	Competencia - Curso	27-oct	02-nov	6	
22	Saber - Unidad	03-nov	09-nov	6	
23	Saber - Actividad	10-nov	16-nov	6	
24	Perfil y configuración	17-nov	23-nov	6	

Figura 3.1: Documento de registro de iteraciones

Num	Tarea	Fecha Inicial	Fecha Final	Tiempo Duración	Cumplimiento
Primer Semestre					
0	Establecer proyecto de formulación	22-mar	12-abr	21	
1	Generar documento de requisitos	12-abr	02-may	20	
1.1	Realizar levantamiento de requisitos	12-abr	02-may	20	
1.2	Redactar propósito, alcance y contexto	12-abr	16-abr	4	
1.3	Formalizar requisitos de usuario y de sistema	16-abr	23-abr	7	
1.4	Generar diagrama de caso de uso nivel 0	23-abr	01-may	8	
1.5	Actualizar requisitos de usuario y sistema	23-abr	01-may	8	
1.6	Redactar Restricciones generales	23-abr	01-may	8	

Figura 3.2: Documento de registro de iteraciones, tareas desplegadas

3.2. Metodología

Para lograr comprender el problema, facilitar la administración del proyecto y otorgar la flexibilidad justa en su desarrollo es que se eligió como metodología de desarrollo de software la metodología Tradicional Iterativo Incremental. Los motivos

por los cuales se utiliza esta metodología por sobre otras con una amplia documentación y como se explica en detalle en la Sección 2.4.1 son:

- Se definen entregables en iteraciones.
- Se tiene mucho más control sobre los requerimientos de software a diferencia de un modelo ágil, sobre todo al contar con una cantidad de tiempo acotado.
- El plan de incremento varía respecto al avance, no a los cambios en sí.

3.2.1. Reuniones

Para cada una de las iteraciones se realizan reuniones con el cliente, donde según la metodología Tradicional Iterativo Incremental detallada en la Sección 2.4.1, cada una de las secuencias planificadas debe producirse un entregable de software.

La importancia de estas reuniones es que se obtienen requisitos o modificaciones de estos últimos, requisitos que son documentados durante cada paso y se traducen en iteraciones para asegurar el éxito del proyecto. Al ser el cliente también el profesor guía, se mide el progreso del proyecto y se dan soluciones a problemas que se presenten.

3.2.2. Cómo se trabajó

Después de la primera reunión con el cliente donde se detecta la problemática a desarrollar, lo primero que se realiza es investigar sobre la metodología curricular de la Universidad en sus documentos oficiales. También se consultó al cuerpo docente del departamento de Ciencias de la Computación sobre los estándares de planes de clase que ellos utilizaban. Con toda la información reunida se procede a realizar el documento de especificación de requisitos, tratando de conseguir la mayor cantidad de información sobre lo que el cliente quería y necesitaba para el proyecto.

Luego de obtener los requisitos, que se van a pulir durante todas las iteraciones con el cliente, se crea el documento de diseño. Este documento contempla las arquitecturas físicas y lógicas, el diagrama de clases, entidad relación y casos de uso, la línea de navegación y el diseño de la interfaz.

Posterior a la creación de ambos documentos, se procede a crear la planificación del proyecto, a través de las iteraciones requeridas por la metodología elegida. Éstas

fueron planificadas en secuencia según las necesidades del cliente y las prioridades de tiempo y aprendizaje.

Las iteraciones pueden ir modificándose a lo largo del proyecto, dependiendo de la evolución y los cambios de los requisitos. Es por esto, que en las reuniones con el cliente se requiere el confirmar la iteración que viene, para así darle la oportunidad de adaptar esta según sus intereses. Confirmándose ésta, se procede a dividir la iteración en tareas a completar y es ahí en donde la metodología comienza a ser aplicada. Si todo resulta en el plazo estipulado, la funcionalidad es entregada. En caso contrario, queda como retraso y debe terminarse en la siguiente iteración, junto a las nuevas tareas que se asignan en esta última.

Finalmente, una vez que los requisitos se encuentren completamente aplicados, se realizan las pruebas de caja negra. En caso de existir una prueba rechazada es imperativo el dedicar una interacción completa para corregir fallos y así, realizar las pruebas de usabilidad con los distintos usuarios.

3.3. Resumen

En este capítulo se mostró la metodología utilizada para realizar el proyecto. Es decir, cómo se tomaron los elementos del marco teórico para producir el prototipo, desde la evaluación del entorno: las personas, la aplicación y las herramientas, hasta la aplicación de la metodología: las reuniones realizadas y como se trabajó para lograr el éxito del proyecto.

4. Especificación de Requisitos

La extracción de requisitos se realiza con el propósito de encontrar una solución razonable, evaluar las necesidades del proyecto y del cliente antes de comenzar a construir la solución. La importancia de realizar requisitos de software es establecer una forma de minimizar el riesgo de que los requisitos fueran alterándose con el tiempo.

Los requisitos de software se fueron obteniendo mediante reuniones con el cliente, efectuadas una vez a la semana, desde el 12 de abril hasta el 2 de mayo de 2018, donde cada semana se fueron evaluando y puliendo los requerimientos del cliente hasta generar un documento de especificación de requisitos de software (ERS) donde se detalla los aspectos del software mucho antes de comenzar el desarrollo.

Los requerimientos para esta memoria son divididos en tres grupos:

- Requisitos de Usuario: Objetivos y metas que fueron capturados durante las reuniones con el cliente, de alto nivel. Según **Ingeniería de Software** de Ian Sommerville[25] *“Son declaraciones, en lenguaje natural, de los servicios que se espera el sistema o producto proporcione. Además de las restricciones bajo las cuales se espera que éste funcione”*. Sin embargo, en el contexto de esta memoria, las restricciones se detallan en una sección propia sin importar si son Requisitos de Usuario o Requisitos de Sistema.
- Requisitos de Sistema: Objetivos y metas que si bien también fueron capturados durante las reuniones con el cliente, detallan las funcionalidades del sistema. Sin embargo, a diferencia de un Requisito de Usuario, éste es mucho más preciso y define con exactitud qué es lo que se espera implementar.

- Restricciones: Limitaciones que el sistema posee. En el contexto de esta memoria, éstos son detallados por Hardware, Software e Interfaz.

Los requerimientos o restricciones son presentados en cuadros, donde la primera columna corresponde al código y a la característica o nombre del requerimiento o restricción, y la segunda columna corresponde a la descripción de la característica. La notación para el código del requerimiento es **RUXX**, **RSXX** y **RGXX** (donde RU corresponde a requisitos de usuario, RS corresponde a requisitos de sistema y RG corresponde a restricciones generales. Posteriormente, XX es el número identificador del requerimiento). En caso de tratarse de Requisitos de Sistema, se añadió una columna extra que corresponde a la dependencia de éstos con los Requisitos de Usuario; cuya notación es **RUXX**.

4.1. Características de los Usuarios

A continuación, se detallan los grupos de usuarios, que son beneficiarios (usuarios) del producto o sistema en desarrollo de forma directa o indirecta.

- Administrador: Posee una cuenta debidamente registrada en el sistema. Gestiona todos los aspectos del sistema.
- Profesor: Posee una cuenta debidamente registrada en el sistema. Gestiona tanto unidades como actividades de un módulo, además de asociar y desasociar saberes a éste. Es quien puede agregar o quitar alumnos de un módulo.
- Alumno: Posee una cuenta debidamente registrada en el sistema. No posee mayores interacciones además de poder ver los cursos que le son asociados.

4.2. Requisitos de Usuario

En esta sección se describen los requerimientos de usuario, los cuales son clasificados en tres categorías, a saber, **Generales**, **Módulos** y **Competencias, Aprendizajes y Saberes**.

4.2.1. Requisitos de Usuario: Generales

El Cuadro 4.1 muestra los cuatro requisitos de usuario generales, que definen funciones básicas para la operación de la aplicación web. Estas funciones incluyen la administración de las cuentas de usuario, la definición del tipo de archivo del documento que contiene el plan de clases y la necesidad de usar una base de datos.

Cuadro 4.1: Requisitos de Usuario: Generales

Código y Característica	Descripción
RU01: Un administrador puede gestionar cuentas de usuario.	Permitir al administrador crear una cuenta de usuario, leer, modificarla, deshabilitar y habilitar.
RU02: Un usuario puede modificar su cuenta.	El sistema debe permitir a un administrador, profesor o alumno modificar su cuenta.
RU03: Un administrador y un profesor pueden guardar un plan de clases.	El sistema debe permitir a un administrador o a un profesor la generación de plan de clases en formato PDF, el cual puede ser almacenado en su dispositivo.
RU04: Asegurar la persistencia de los datos.	El sistema debe almacenar los datos ingresados al sistema de manera persistente.

4.2.2. Requisitos de Usuario: Módulos

Los Cuadros 4.2 y 4.3 muestran los siete requisitos de usuario, que definen funciones básicas para los módulos, las unidades y actividades de la aplicación web. Estas funciones incluyen la gestión de un módulo con sus respectivas características y la asignación de profesores y alumnos a un módulo.

Cuadro 4.2: Requisitos de Usuario: Módulos, Unidades y Actividades

Código y Característica	Descripción
RU05: Un administrador puede gestionar módulos.	El sistema debería permitir a un administrador leer, crear, modificar, deshabilitar, habilitar y asignar profesores a un curso.
RU06: Un profesor puede gestionar el contenido de un módulo.	El sistema debería ser capaz de permitir a un profesor el modificar la descripción de un módulo.

Cuadro 4.3: Requisitos de Usuario: Módulos, Unidades y Actividades

Código y Característica	Descripción
RU07: Un administrador y un profesor pueden gestionar las unidades de un módulo.	El sistema debería permitir a un administrador o a un profesor el crear, leer, modificar, habilitar, deshabilitar y asignar clases a una unidad de un curso.
RU08: Un administrador y un profesor pueden gestionar las actividades de una unidad.	El sistema debería permitir a un administrador o a un profesor el crear, leer, modificar, habilitar, deshabilitar y asignar clases a una actividad de una unidad.
RU09: Un administrador o un profesor pueden asignar y desasignar alumnos a un curso.	El sistema debería permitir a un administrador o a un profesor el asignar o desasignar alumnos a un módulo.
RU10: Un administrador puede asignar y desasignar profesores a un curso.	El sistema debe permitir a un administrador asignar o desasignar profesores a un curso.
RU11: Un usuario puede visualizar una lista con los cursos anteriores de un profesor y un alumno.	El sistema debe permitir a un profesor, un administrador o un alumno visualizar los cursos anteriores de un profesor y un alumno.

4.2.3. Requisitos de Usuario: Competencias, Aprendizajes y Saberes

Los Cuadros 4.4 y 4.5 detallan los seis requisitos de usuario, que definen funciones para las Competencias, Aprendizajes y Saberes de la aplicación web. Estas funciones incluyen la gestión de Competencias, Aprendizajes y Saberes y la asignación de Competencias a módulos y de Saberes a unidades y actividades.

Cuadro 4.4: Requisitos de Usuario: Competencias, Aprendizajes y Saberes

Código y Característica	Descripción
RU12: Un administrador puede gestionar Competencias.	El sistema debe permitir a un administrador el crear, leer, modificar, deshabilitar y habilitar Competencias.
RU13: Un administrador puede gestionar Aprendizajes.	El sistema debe permitir a un administrador el crear, leer, modificar, deshabilitar y habilitar Aprendizajes. Además, debe asignar y desasignar Aprendizajes a Competencias

Cuadro 4.5: Requisitos de Usuario: Competencias, Aprendizajes y Saberes

Código y Característica	Descripción
RU14: Un administrador puede gestionar Saberes.	El sistema debe permitir a un administrador el crear, leer, modificar, habilitar y deshabilitar Saberes. Además, debería asignar y desasignar Saberes a Aprendizajes.
RU15: Un administrador puede asignar y desasignar Competencias a un curso.	El sistema debe permitir a un administrador el asignar y desasignar competencias a un módulo.
RU16: Un administrador o un profesor pueden asignar y desasignar Saberes de un módulo a sus unidades.	El sistema debe permitir a un administrador o a un profesor el asignar y desasignar Saberes de una competencia perteneciente a un curso a sus unidades.
RU17: Un administrador o un profesor pueden asignar y desasignar Saberes de una unidad a sus actividades.	El sistema debe permitir a un administrador o a un profesor asignar o desasignar Saberes asignados anteriormente a una unidad, a sus actividades correspondientes.

4.3. Requisitos de Sistema

En esta sección se describen los requerimientos de sistema, los cuales serán clasificados en **Generales**, **Módulos**, **Unidades**, **Actividades**, **Competencias**, **Aprendizajes** y **Saberes**, Para ofrecer una mayor claridad de los requerimientos.

4.3.1. Requisitos de Sistema: Generales

Los Cuadros 4.6, 4.7 y 4.8 muestran los once requisitos de usuario, que explyan funciones para la administración de cuentas de usuario y la definición de plan de clases que fueron previamente definidos en el Cuadro 4.1.

Cuadro 4.6: Requisitos de Sistema: Generales

Código y Característica	Descripción	RU
RS01: Autenticar Usuario	El sistema debe permitir al administrador, profesor y alumno ingresar a la aplicación usando sus credenciales de acceso.	RU01 a RU17

Cuadro 4.7: Requisitos de Sistema: Generales

Código y Característica	Descripción	RU
RS02: Crear cuenta de Usuario.	El sistema debe permitir al administrador crear una cuenta de usuario, usando su Nombre, Rut, Correo Electrónico y Contraseña.	RU01
RS03: Modificar cuenta de Usuario	El sistema debe permitir al administrador modificar una cuenta de usuario. Sin embargo, el administrador no puede modificar el id del usuario.	RU01
RS04: Deshabilitar cuenta de usuario.	El sistema debe permitir al administrador deshabilitar una cuenta de usuario que no sea la suya.	RU01
RS05: Habilitar cuenta de usuario.	El sistema debe permitir al administrador el habilitar una cuenta de usuario.	RU01
RS06: Modificar contraseña de cuenta.	El sistema debe permitir cualquier usuario el modificar la contraseña de su propia cuenta.	RU02
RS07: Modificar correo electrónico.	El sistema debe permitir cualquier usuario el modificar el correo electrónico de su propia cuenta.	RU02
RS08: Recuperar Contraseña	El sistema debe permitir cualquier usuario recuperar su cuenta utilizando su correo electrónico.	RU02
RS09: Guardar Plan de Clases.	El sistema debe permitir a un administrador guardar un plan de clases de cualquier curso, y a un profesor, guardar un plan de clases de los cursos asociados a éste.	RU01, RU03, RU05, RU07, RU08, RU12, RU13, RU14
RS10: Leer Usuarios.	El sistema debe permitir a un administrador visualizar la lista de Usuarios registrados en el sistema.	RU01

4.3.2. Requisitos de Sistema: Módulos

Los siguientes Cuadros 4.9 y 4.10 detallan los trece requisitos que explayan funciones para la administración de módulos y la asignación de Competencias, profesores

Cuadro 4.8: Requisitos de Sistema: Generales

RS11: Definir Periodos y Años.	El sistema debe permitir a un administrador definir tanto cantidad totales de periodos, como el periodo y el año actual en el que el periodo escolar se encuentra.	RU03, RU06, RU09, RU11
---------------------------------------	--	---------------------------------

y alumnos que fueron previamente definidos en los Cuadros 4.2 y 4.3.

Cuadro 4.9: Requisitos de Sistema: Módulos

Código y Característica	Descripción	RU
RS12: Crear Módulo.	El sistema debe permitir al administrador crear un módulo y un módulo por periodo.	RU05
RS13: Modificar Módulo.	El sistema debe permitir al administrador modificar un módulo. Sin embargo, el administrador no puede modificar el id del módulo.	RU05
RS14: Deshabilitar Módulo.	El sistema debe permitir al administrador deshabilitar un módulo.	RU05
RS15: Habilitar Módulo.	El sistema debe permitir al administrador habilitar un módulo que haya sido previamente deshabilitado.	RU05
RS16: Asignar Competencia a Módulo.	El sistema debe permitir al administrador asignar una competencia previamente creada a un módulo.	RU05, RU15
RS17: Desasignar Competencia a Módulo.	El sistema debe permitir al administrador desasignar una competencia a un módulo previamente asignada.	RU05, RU15
RS18: Asignar Profesor a Módulo.	El sistema debe permitir al administrador asignar un profesor a un módulo.	RU10
RS19: Desasignar Profesor a Módulo.	El sistema debe permitir al administrador desasignar un profesor de un módulo.	RU10
RS20: Asignar Alumno a Módulo.	El sistema debe permitir al administrador o un profesor asignar un alumno a un módulo.	RU09

Cuadro 4.10: Requisitos de Sistema: Módulos

Código y Característica	Descripción	RU
RS21: Desasignar Alumno a Módulo.	El sistema debe permitir al administrador o a un profesor desasignar un alumno de un módulo.	RU09
RS22: Modificar descripción del módulo.	El sistema debe permitir a un profesor modificar la descripción de un módulo.	RU06
RS23: Leer Módulos Período Actual.	El sistema debe permitir a un profesor visualizar los Módulos en el periodo y el año actual en el que se encuentra inscrito.	RU11
RS24: Leer Cursos.	El sistema debería permitir a un administrador el poder visualizar la lista de cursos totales existentes. Y al profesor, la lista de cursos totales en el que se encuentre inscrito.	RU05, RU11

4.3.3. Requisitos de Sistema: Unidades

Los Cuadros 4.11 y 4.12 especifican los siete requisitos que explayan funciones para la administración de unidades y la asignación de Saberes que fueron previamente definidos en los Cuadros 4.2 y 4.3.

Cuadro 4.11: Requisitos de Sistema: Unidades

Código y Característica	Descripción	RU
RS25: Crear Unidad en Módulo.	El sistema debe permitir al administrador o a un profesor crear una Unidad en un Módulo previamente creado.	RU07
RS26: Modificar Unidad.	El sistema debe permitir al administrador o a un profesor modificar el nombre o la descripción de una Unidad.	RU07
RS27: Habilitar Unidad.	El sistema debe permitir al administrador o a un profesor habilitar una unidad que ha sido deshabilitada.	RU07
RS28: Deshabilitar Unidad.	El sistema debe permitir al administrador o a un profesor deshabilitar una unidad habilitada.	RU07
RS29: Asignar Saber de un Módulo a una Unidad.	El sistema debe permitir al administrador o a un profesor asignar el Saber de un Módulo a una Unidad.	RU07, RU16

Cuadro 4.12: Requisitos de Sistema: Unidades

Código y Característica	Descripción	RU
RS30: Desasignar Saber de Módulo a una Unidad.	El sistema debe permitir al administrador o a un profesor desasignar un Saber de una Unidad.	RU07, RU16
RS31: Leer Unidades.	El sistema debe permitir a un administrador visualizar la lista de Unidades de un Módulo.	RU07

4.3.4. Requisitos de Sistema: Actividades

Los Cuadros 4.13 y 4.14 muestran los siete requisitos que explayan funciones para la administración de actividades y la asignación de Saberes en el sistema web, que fueron previamente definidos en los Cuadros 4.2 y 4.3.

Cuadro 4.13: Requisitos de Sistema: Actividades

Código y Característica	Descripción	RU
RS32: Crear Actividad en Unidad.	El sistema debe permitir al administrador o a un profesor crear una Actividad, con su Nombre, Descripción, Tipo, Fecha de Inicio y Fecha de Término.	RU08
RS33: Modificar Actividad.	El sistema debe permitir al administrador o a un profesor modificar el Nombre, la Descripción, el Tipo, la Fecha de Inicio y la Fecha de Término de una Actividad.	RU08
RS34: Habilitar Actividad.	El sistema debe permitir al administrador o a un profesor habilitar una Actividad previamente deshabilitada.	RU08
RS35: Deshabilitar Actividad.	El sistema debe permitir al administrador o a un profesor deshabilitar una Actividad.	RU08
RS36: Asignar Saber de una Unidad a una Actividad.	El sistema debe permitir al administrador o a un profesor asignar el Saber de una Unidad a una Actividad.	RU08, RU17
RS37: Desasignar Saber de una Unidad a una Actividad.	El sistema debe permitir al administrador o a un profesor desasignar un Saber de una Actividad.	RU08, RU17

Cuadro 4.14: Requisitos de Sistema: Actividades

Código y Característica	Descripción	RU
RS38: Leer Actividades.	El sistema debe permitir a un administrador visualizar la lista de Actividades de una Unidad.	RU08

4.3.5. Requisitos de Sistema: Competencias

El siguiente Cuadro 4.15 detalla los cinco requisitos que explayan funciones para la administración de Competencias que fueron previamente definidos en los Cuadros 4.4 y 4.5.

Cuadro 4.15: Requisitos de Sistema: Competencias

Código y Característica	Descripción	RU
RS39: Crear Competencia.	El sistema debe permitir al administrador crear una competencia.	RU12
RS40: Modificar Competencia.	El sistema debe permitir al administrador modificar el nombre, la descripción, el nivel de dominio (básico, intermedio o avanzado) y el tiempo de desarrollo de una competencia.	RU12
RS41: Habilitar Competencia.	El sistema debe permitir al administrador habilitar una competencia previamente deshabilitada.	RU12
RS42: Deshabilitar Competencia.	El sistema debe permitir al administrador deshabilitar una Competencia.	RU12
RS43: Leer Competencias.	El sistema debe permitir a un administrador visualizar la lista de Competencias existentes.	RU12

4.3.6. Requisitos de Sistema: Aprendizajes

En el Cuadro 4.16 se muestran siete requisitos que explayan funciones para la administración de Aprendizajes que fueron previamente definidos en los Cuadros 4.4 y 4.5.

Cuadro 4.16: Requisitos de Sistema: Aprendizajes

Código y Característica	Descripción	RU
RS44: Crear Aprendizaje.	El sistema debería permitir al administrador el poder crear un Aprendizaje con su nombre, categoría, subcategoría y estado.	RU13
RS45: Modificar Aprendizaje.	El sistema debería permitir al administrador el poder modificar el nombre, la categoría, la subcategoría y el estado de un Aprendizaje.	RU13
RS46: Habilitar Aprendizaje.	El sistema debería permitir al administrador el poder habilitar un Aprendizaje previamente deshabilitado.	RU13
RS47: Deshabilitar Aprendizaje.	El sistema debería permitir al Administrador el poder deshabilitar un Aprendizaje.	RU13
RS48: Asignar Aprendizaje a Competencia.	El sistema debería permitir al administrador el poder asignar un Aprendizaje a una Competencia.	RU12, RU13
RS49: Desasignar Aprendizaje a Competencia.	El sistema debería permitir al administrador el poder desasignar un Aprendizaje de una competencia.	RU12, RU13
RS50: Leer Aprendizajes.	El sistema debería permitir a un administrador el poder visualizar la lista de Aprendizajes existentes.	RU13

4.3.7. Requisitos de Sistema: Saberes

Los siguientes Cuadros 4.17 y 4.18 muestran requisitos que expliquen funciones para la administración de Saberes para el sistema web, que fueron previamente definidos en los Cuadros 4.4 y 4.5

Cuadro 4.17: Requisitos de Sistema: Saberes

Código y Característica	Descripción	RU
RS51: Crear Saber.	El sistema debe permitir al administrador o a un profesor crear un Saber con su código, nombre, tipo y estado.	RU14

Cuadro 4.18: Requisitos de Sistema: Saberes

Código y Característica	Descripción	RU
RS52: Modificar Saber.	El sistema debe permitir al administrador o a un profesor modificar el código, nombre, tipo y estado.	RU13
RS53: Habilitar Saber.	El sistema debe permitir al administrador o a un profesor habilitar un Saber previamente deshabilitado.	RU13
RS54: Deshabilitar Saber.	El sistema debe permitir al administrador o a un profesor deshabilitar un Saber.	RU13
RS55: Asignar Saber a Aprendizaje.	El sistema debe permitir al administrador o a un profesor asignar un Saber a una competencia.	RU13, RU14
RS56: Desasignar Saber a Aprendizaje.	El sistema debe permitir al administrador o a un profesor desasignar un Saber de una competencia.	RU13, RU14
RS57: Leer Saberes	El sistema debe permitir a un administrador visualizar la lista de Saberes existentes.	RU13

4.4. Restricciones

En esta sección se describirán las Restricciones, las cuales se dividen en **Software** e **Interfaz**.

4.4.1. Restricciones de Software

Los siguientes Cuadros 4.19 y 4.20 muestran restricciones de software, que definen características de interacción entre el usuario y el prototipo. Estas funciones incluyen las restricciones de modificación y eliminación de datos y las limitaciones de la administración de cuentas de usuario.

4.4.2. Restricciones de Interfaz

El Cuadro 4.21 detalla dos restricciones de interfaz que definen características visuales, como la plataforma donde el prototipo es compatible y la visualización del logotipo de la universidad.

Cuadro 4.19: Restricciones de Software

Código y Característica	Descripción
RG01: El sistema debe estar en español	Debido al alcance de la solución propuesta (el cual se realiza con el propósito de que sea utilizado por el cuerpo docente de la Universidad de Talca), el idioma del Software debe ser español.
RG02: No se pueden modificar el código de un dato	Debido que es fundamental mantener la integridad entre datos y disminuir la posibilidad de redundancia e inconsistencia, es necesario que un dato no pueda ser modificado.
RG03: No se pueden eliminar datos, sólo deshabilitarlos	Debido a que el RU04 indica que el sistema debe persistir los datos que se ingresen, para asegurar el correcto funcionamiento del producto es que se decide que el sistema debe poder deshabilitar y habilitar datos. No eliminarlos.
RG04: Un profesor no puede gestionar ni Competencias, ni Aprendizajes, Ni Saberes de un curso.	Debido a que el administrador es quien debe realizar esta operación, un profesor no puede asignar ni quitar Competencias a un Módulo. Menos gestionarlas.
RG05: Un profesor no puede gestionar Unidades y Actividades en un Módulo al que no fue asignado.	Debido a los limitados permisos que un Profesor posee, es imperativo que éste no pueda realizar operaciones sobre Unidades y Actividades en un Módulo al que no fue asociado.
RG06: Un profesor no puede asignar Saberes a una Unidad o una Actividad de un Módulo al que no fue asignado.	Debido a los limitados permisos que un Profesor posee, es imperativo que éste no pueda asociar o quitar saberes a una Unidad o una Actividad de un Módulo al que no fue asociado.
RG07: Un profesor no puede asignar Competencias a Aprendizajes ni Saberes a los Aprendizajes.	Debido a que el administrador es quien debe realizar esta operación, un profesor no puede asignar o desasignar competencias a aprendizajes, ni saberes a los aprendizajes.
RG08: Un Alumno no puede gestionar Usuarios, ni Saberes, ni Competencias, ni Aprendizajes, ni Módulos, ni Unidades, ni Actividades.	Debido a que el administrador es quien debe realizar esta operación, un alumno no puede gestionar ningún módulo del prototipo. Sólo cambiar o recuperar su contraseña.

Cuadro 4.20: Restricciones de Software

Código y Característica	Descripción
RG09: El Correo Electrónico y el Rut del Usuario, deben ser únicos	Debido que es fundamental evitar la redundancia de datos, es necesario que el correo electrónico y el Rut del usuario sean únicos.
RG10: Un usuario no puede eliminarse a sí mismo.	Debido que es fundamental mantener la integridad entre datos, un usuario no puede eliminarse a sí mismo.
RG11: Un usuario no puede deshabilitarse a sí mismo.	Debido que es fundamental mantener la integridad entre datos, un usuario no puede deshabilitarse a sí mismo.
RG12: Un usuario no puede modificar su código de usuario.	Debido que es fundamental mantener la integridad entre datos y disminuir la posibilidad de redundancia e inconsistencia, es necesario que el código de un usuario no pueda ser modificado.
RG13: Un usuario no puede modificar su id de usuario.	Debido que es fundamental mantener la integridad entre datos y disminuir la posibilidad de redundancia e inconsistencia, es necesario que el código de un usuario no pueda ser modificado.

Cuadro 4.21: Restricciones de Interfaz

Código y Característica	Descripción
RG24: El sistema debe poder visualizarse en un computador.	Debido a que el prototipo es un sistema web y dado que la elaboración de un plan de clases se suele realizar por el cuerpo docente en un computador, es imperativo que el prototipo funcione para uno.
RG25: El logo de la Universidad debe estar visible en el sistema.	Debido a que el software se pensó para dar solución a una problemática de la Universidad de Talca, es imperativo que éste sea visible en alguna parte del producto.

4.5. Matriz de Dependencias

En esta sección se presenta la matriz de dependencias. La matriz de dependencias es una forma en la cual se puede evaluar de qué manera se relacionan los requisitos, para así poder analizar la importancia de un requisito por sobre otros para el cliente, o el impacto de los cambios en el proyecto en sí.

Como se puede observar en la Figura 4.1, en la Matriz de Responsabilidades los requisitos de usuario son el origen de la dependencia y los requisitos de sistema, son los requisitos que se crearon en base a los requisitos de sistema.

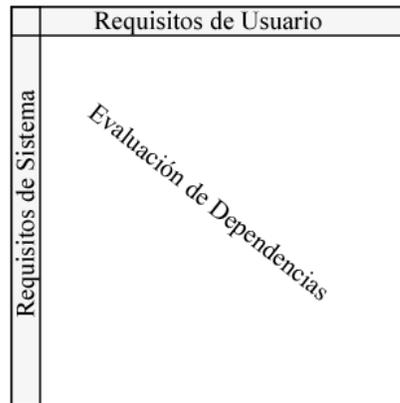


Figura 4.1: Ejemplo de Organización de la Matriz de Dependencias.

Se puede observar, que los requisitos de sistema más dependientes son el RS01 (Autenticar Usuario), RS09 (Guardar Plan de Clases) y RS11 (Definir Periodos y Años) destacados en la Figura 4.2. Despreciando la Autenticación de Usuario, dado que es un requisito primordial para garantizar la seguridad del prototipo, el requisito más importante para el cliente es **Guardar Plan de Clases**. En el Anexo B, Figuras B.1, B.2, B.3 y B.4 se puede visualizar la matriz completa de dependencias, la cual consta de diez y siete requisitos de usuario y cincuenta y siete de sistema.

RS/RU	RU01	RU02	RU03	RU04	RU05	RU06	RU07	RU08	RU09	RU10	RU11	RU12	RU13	RU14	RU15	RU16	RU17
RS01	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
RS02	X																
RS03	X																
RS04	X																
RS05	X																
RS06		X															
RS07		X															
RS08		X															
RS09	X		X		X		X	X				X	X	X			
RS10	X																
RS11			X			X			X								

Figura 4.2: Matriz de Dependencia con Requisitos Prioritarios Destacados

Si bien no fueron destacados, los requisitos de sistema que generan una dependencia, aunque mínima, son: el RS16 (Asignar Competencia a Curso), RS17 (Desasignar

Competencia a Curso), RS24 (Leer Cursos), RS29 (Asignar Saber de un Curso a una Unidad), RS30 (Desasignar Saber de Curso a una Unidad), RS36 (Asignar Saber de una Unidad a una Actividad), RS37 (Desasignar Saber de una Unidad a una Actividad), RS48 (Asignar Aprendizaje a Competencia), RS49 (Desasignar Aprendizaje a Competencia), RS55 (Asignar Saber a Aprendizaje) y RS56 (Desasignar Saber a Aprendizaje).

4.6. Casos de Uso

En esta sección se presentan los principales Casos de Uso, con la finalidad de describir el comportamiento del sistema en las diferentes condiciones donde éste realiza una interacción con alguno de los Actores o Usuarios. Según [24], “*En esencia, un caso de uso narra una historia estilizada sobre cómo interactúa un usuario final con el sistema en circunstancias específicas*”.

En las Figuras B.5, B.6, B.7, B.8, B.9, B.10, B.11 y B.12 del Anexo B se pueden vislumbrar el diagrama de casos de uso nivel 0 del sistema, separados por Competencia, Aprendizaje, Saber, Curso, Unidad, Actividad, Usuarios y Otros procesos, que proporciona una visión general de las principales funcionalidades del software. Donde:

- En la Figura B.5 del Anexo B, el usuario ejecuta la aplicación e ingresa al prototipo utilizando su nombre de usuario (RUT sin dígito verificador) y contraseña. Una vez validado en el sistema, si el usuario es Administrador puede crear una competencia o revisar una lista de competencias ya creadas previamente, donde debe ser capaz de modificar o deshabilitar cualquiera de ellas. También puede habilitar una competencia que ha sido deshabilitada previamente.
- Como se puede observar en la Figura B.6 del Anexo B, una vez el usuario ingresa sus credenciales y es validado, si éste es Administrador puede crear un aprendizaje o revisar una lista de aprendizajes ya creados previamente, donde debe ser capaz de modificar o deshabilitar cualquiera de ellos. También puede habilitar un aprendizaje que ha sido deshabilitada previamente, agregar un aprendizaje a una competencia existente o eliminar el aprendizaje de la competencia.

- En la Figura B.7 del Anexo B, si el usuario es de tipo Administrador, puede crear un saber o revisar una lista de saberes ya creados previamente, donde debe ser capaz de modificar o deshabilitar cualquiera de ellos. También puede habilitar un saber que ha sido deshabilitado previamente, agregar un saber a un aprendizaje existente o eliminar el saber del aprendizaje.
- En la Figura B.8 del Anexo B, si el usuario es Administrador y ya fue validado en el sistema puede crear un curso o revisar una lista de cursos ya creados previamente, donde debe ser capaz de modificar o deshabilitar cualquiera de ellos. También puede habilitar un curso que ha sido deshabilitado previamente, gestionar profesores (proceso en el cual un administrador añade o elimina profesores de un curso en un periodo y año determinado) o competencias (proceso en el cual un administrador añade o eliminar una competencia de un curso). En el caso de que el usuario sea un administrador o un profesor, ambos pueden gestionar alumnos de un curso en un periodo y año determinado (proceso en el cual un administrador o un profesor añade o elimina alumnos de un curso).
- En la Figura B.9 del Anexo B, el usuario Administrador puede elegir uno de los cursos para poder crear una unidad o revisar una lista de unidades ya creadas previamente, donde puede modificar o deshabilitar cualquiera de ellos, habilitar una unidad que ha sido deshabilitado previamente, agregar un saber de una lista de saberes que son parte de las competencias de un curso a una unidad previamente creada, o eliminar un saber de una unidad.
- Se puede observar en la Figura B.10 del Anexo B, el usuario Administrador puede elegir una de las unidades para poder crear una actividad o revisar una lista de actividades ya creadas previamente, donde puede modificar o deshabilitar cualquiera de ellos, habilitar una actividad que ha sido deshabilitado previamente, agregar un saber de una lista de saberes que son parte de las competencias de un curso a una actividad previamente creada, o eliminar un saber de una actividad.
- En la Figura B.11, del Anexo B, se puede observar que el Administrador puede Crear una cuenta de Usuario, o puede acceder a la Lista de Usuarios creados en el Sistema donde puede Habilitar, Deshabilitar o Modificar sus cuentas. Todos

los Usuarios del Sistema pueden ver su propio perfil, modificar la contraseña de su cuenta o recuperar ésta, mediante el uso de su correo electrónico.

- En la Figura B.12, del Anexo B, el Administrador puede imprimir el plan de clases o gestionar los periodos y años una vez inició sesión en el prototipo.

Con la intención de proporcionar una forma más simple de poder visualizar las funcionalidades del sistema que presentan una mayor dependencia, se crearon tres casos de uso nivel uno.

4.6.1. Iniciar Sesión

El usuario ejecuta la aplicación e ingresa al prototipo utilizando su nombre de usuario (RUT sin dígito verificador) y contraseña. Luego, el sistema valida las credenciales del usuario. Si el RUT o la contraseña del usuario no son correctas, el sistema devuelve una advertencia y no le permite entrar. En caso contrario, permite al usuario utilizar el prototipo. En caso de olvidar la contraseña, el usuario puede recuperar su contraseña utilizando su correo electrónico.

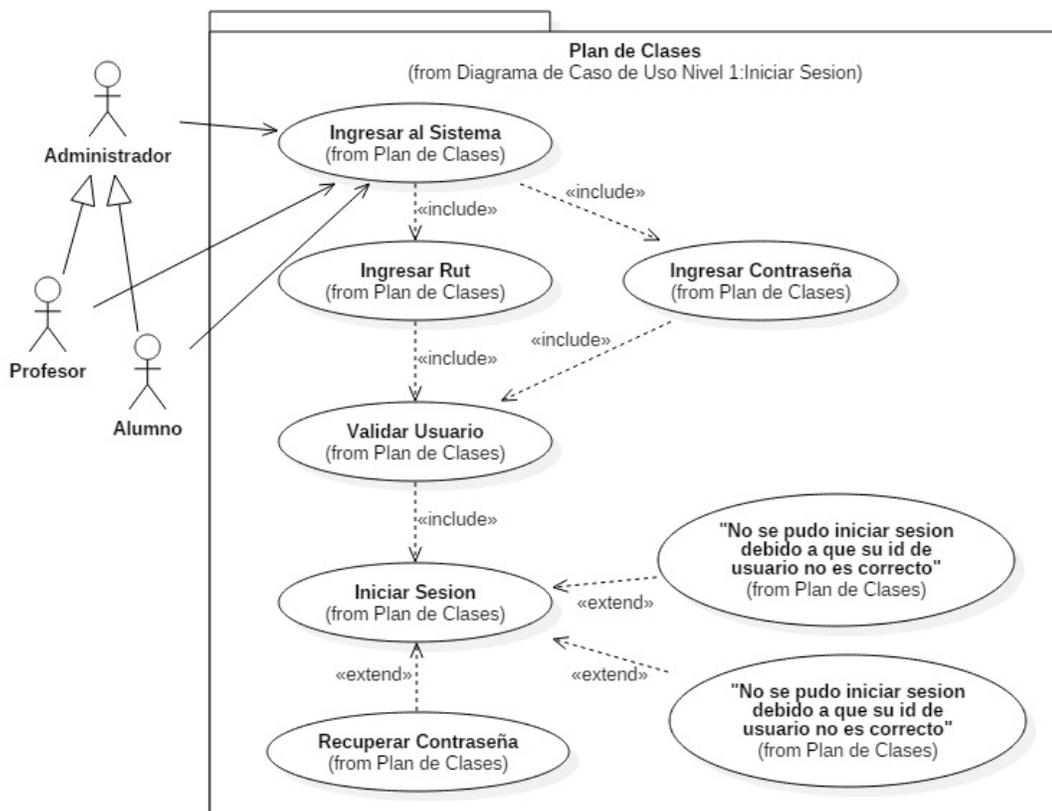


Figura 4.3: Diagrama de Casos de Uso Nivel 1: Iniciar Sesión

4.6.2. Gestionar Periodos y Años

El administrador ejecuta la aplicación e ingresa al prototipo utilizando sus credenciales de acceso. El administrador puede ingresar una cantidad de periodos que pueden ser usados en la creación de cursos, el año actual y el periodo en el cual se encuentra. Luego, el sistema valida los datos ingresados. En caso de no ingresar un número entero positivo mayor a cero en el caso de la cantidad de periodos, o un número entre uno y la cantidad de periodos ingresados el sistema devuelve una advertencia y no guarda los datos. Tampoco lo hace en caso de que el año ingresado sea incorrecto.

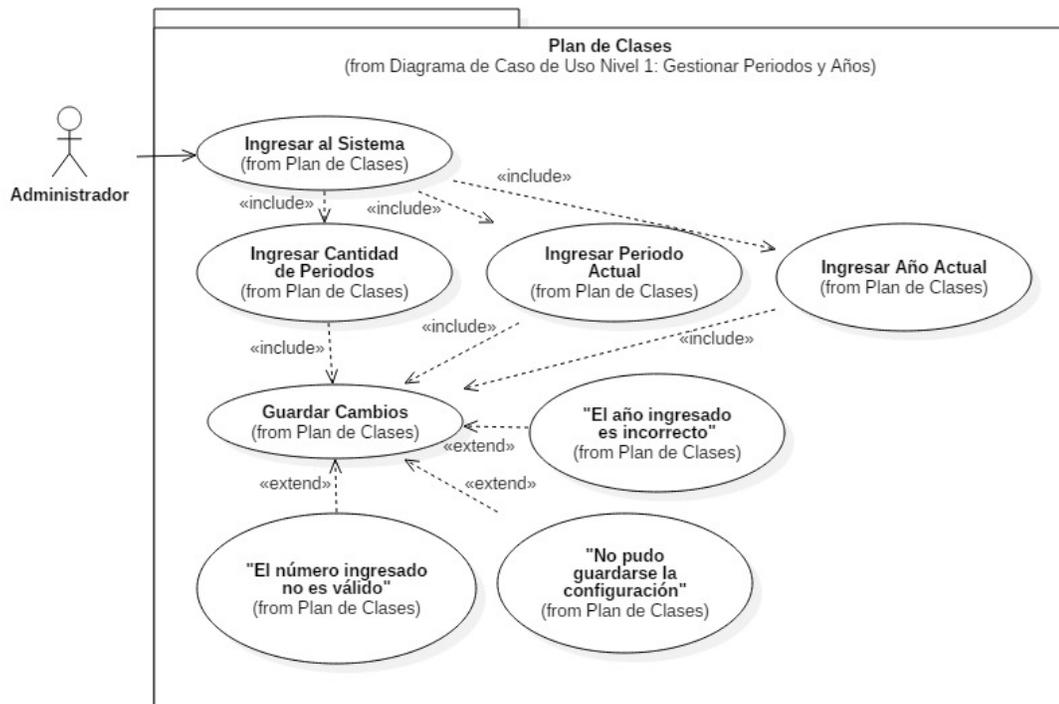


Figura 4.4: Diagrama de Casos de Uso Nivel 1: Gestionar Periodos y Años

4.6.3. Imprimir Plan de Clases

El administrador o profesor ejecuta la aplicación e ingresa al prototipo utilizando sus credenciales de acceso. El administrador o profesor puede seleccionar un curso. En caso de que no haya cursos creados previamente, o en caso de que el profesor no esté asignado a un curso el sistema devuelve una advertencia y no genera el plan de clases. Luego, el administrador o profesor debe seleccionar un año y un periodo. Si estos no han sido creados previamente, al igual que los cursos, el sistema devuelve una advertencia y no genera el plan de clases. Finalmente, el sistema genera un plan de clases en formato PDF que el administrador o profesor pueden guardar, abrir e imprimir.

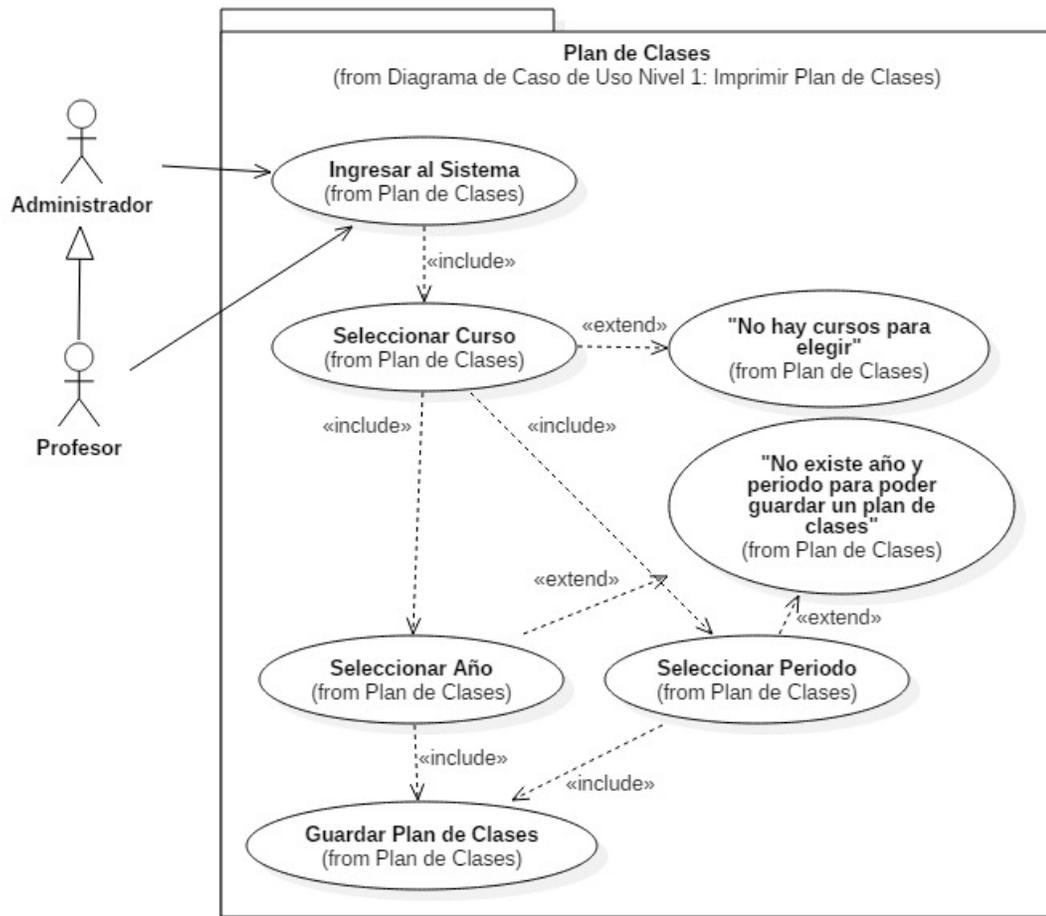


Figura 4.5: Diagrama de Casos de Uso Nivel 1: Imprimir Plan de Clases

4.7. Resumen

En este capítulo se presentan los requisitos captados al cliente durante la primera iteración, donde en cada semana se evalúan y pulen hasta generar un documento de requisitos. Los requisitos de sistema nacen como consecuencia, detallando las funcionalidades del prototipo que el cliente espera. Esto origina la matriz de dependencias, donde se pueden identificar aquellos requisitos críticos.

También se establecen las restricciones del prototipo y los diferentes casos de uso, que describen el comportamiento del sistema en las diferentes condiciones donde se realiza una interacción.

5. Diseño del Sistema

Según [24] el diseño es definido como: “*Un proceso de múltiples etapas que a partir de los requerimientos de información sintetiza las representaciones de los datos y la estructura del programa, las características de la interfaz y los detalles del procedimiento*”. El diseño tiene relación directa con la toma de decisiones cuyo objetivo es dividir las secuencias de procesamiento y la representación de la información, elaborando representaciones coherentes y planeadas de sistemas [24].

En este capítulo se presentan los métodos por los cuales se crea el plano preliminar por el cual el software es elaborado. Esto incluye la arquitectura física, lógica, los diagramas de clase, relacional, de actividades, y el diseño de la interfaz del sistema.

5.1. Arquitectura Física

La arquitectura física utilizada para dar solución al problema es la **arquitectura cliente-servidor**. En la arquitectura cliente-servidor, la aplicación es modelada como un conjunto de servicios que un servidor proporciona y a su vez, un conjunto de clientes que utilizan estos servicios. En este caso, es necesario que los clientes conozcan qué servidores se encuentran disponibles, a pesar de desconocer la existencia de otros clientes [25].

El principal motivo por el cual esta arquitectura fue elegida para dar respuesta al problema es porque permite evitar que el cliente sea sobrecargado, dividiendo los procesos entre el cliente y el servidor. Debido a que el rango de usuarios en una primera etapa se centra en el cuerpo docente del Departamento de Ciencias de la Computación de la Universidad de Talca, se descartaron desventajas como la generación de un tráfico significativo (el servidor es quien se lleva toda la carga de

procesamiento). Sin embargo, es recomendable utilizar un modelo de cliente rico o un modelo con más capas si la cantidad de usuarios objetivo aumenta.

Tal y como muestra la figura 5.1, la solución es denotada por una aplicación estructurada en tres capas, que refleja la estructura lógica de la aplicación a desarrollar. La **capa de presentación** es donde la información es presentada al cliente y donde se realiza la interacción con él, denotado por los dispositivos que son compatibles con la aplicación sin importar el sistema operativo que el usuario utilice al usar un diseño web responsivo. Tal cual se expuso en la RG17, el dispositivo más compatible con el prototipo es un computador o una tablet.

La **capa de procesamiento** de la aplicación tiene relación con la implementación de la lógica de la aplicación, sirviendo de enlace con la capa de datos para procesar los requerimientos desde los usuarios y sus respuestas. Por lo tanto, es ahí donde se alojan los controladores del prototipo.

Finalmente, la **capa de gestión de datos** contiene todas aquellas operaciones relacionadas a la base de datos. Estas dos últimas se encuentran juntas, dado que se utilizó una arquitectura cliente-servidor simple, donde la aplicación fue organizada como un servidor y donde el cliente sólo es responsable de parte de la capa de presentación del software [25].

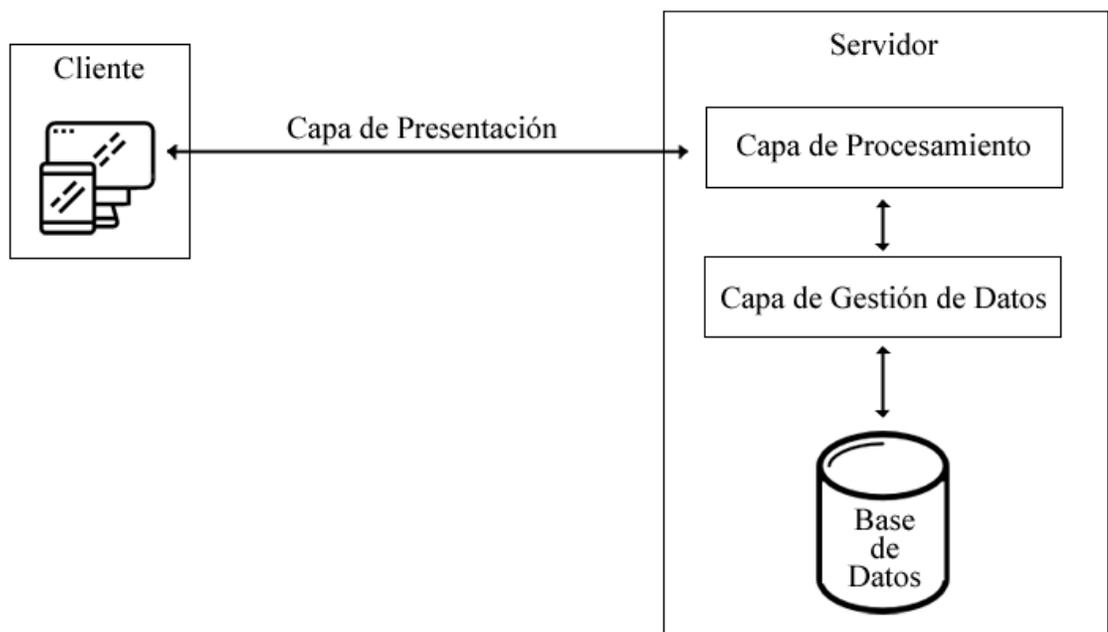


Figura 5.1: Diagrama de representación de la Arquitectura Física del sistema.

5.2. Arquitectura Lógica

La arquitectura lógica utilizada para dar solución al problema es el **modelo vista controlador**. Este modelo es usado para construir interfaces de usuario cuya finalidad es desacoplar las vistas de los modelos [25], con el objetivo de beneficiar la posterior modificación de los componentes sin la necesidad de conocer todos los módulos, y crear un conjunto especial de componentes necesarios para soportar un proceso de desarrollo de software altamente interactivo [32].

Como se puede observar en la Figura 5.2, el modelo vista controlador se encuentra estructurado en tres contenedores, donde los objetos de las diferentes clases asumen las operaciones asociadas al dominio de la aplicación, la visualización del estado de ésta y la interacción del usuario con el modelo y la vista [32]. Cada vez que los datos del modelo son alterados, el controlador se encarga de notificar a las vistas que dependen de él y estas a su vez pueden actualizarse a sí mismas. Este enfoque permite que varias vistas sean asignadas a un modelo y se puedan crear nuevas vistas sin la necesidad de escribir el modelo de forma repetitiva [25].

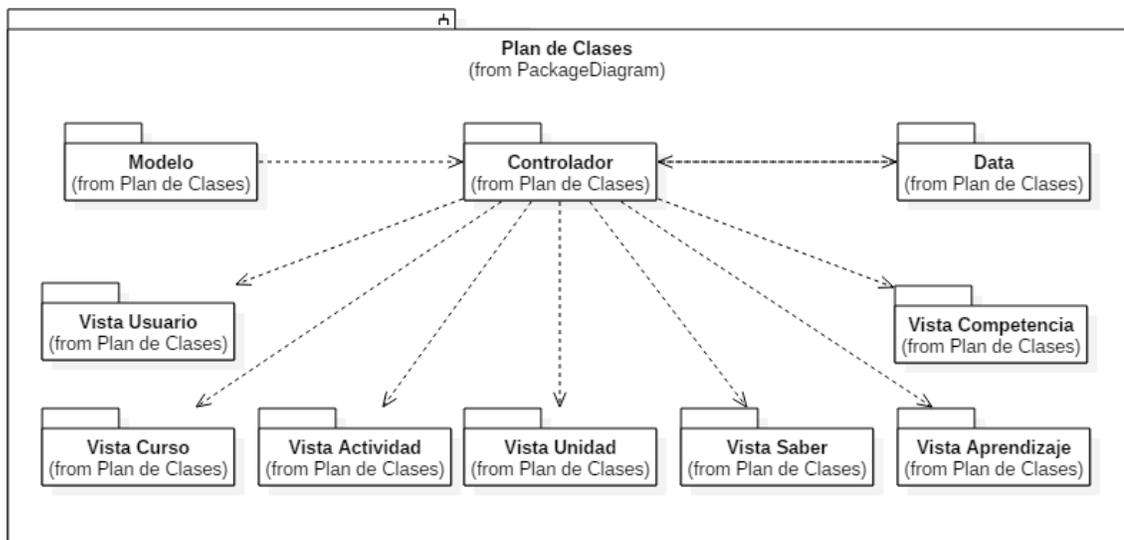


Figura 5.2: Diagrama general de representación de la Arquitectura Lógica

En concordancia con la arquitectura física del sistema, en la *capa de presentación* se pueden encontrar las vistas de los usuarios del sistema, como se puede observar en las Figuras 5.3 y 5.2. La **vista** debe asegurarse de que su apariencia refleje el estado del modelo [25], proporcionando la capacidad de realizar transformaciones gráficas

y ventanas [32]. Las vistas consideradas son las siguientes:

- **Vista Usuario:** Contiene todas las vistas que tienen relación con el usuario en sí. Esto incluye la vista de iniciar sesión, la vista de modificar contraseña, la vista de perfil y la vista principal. También los procesos administrativos del prototipo como la gestión de usuario y la vista de configuración de periodos y años.
- **Vista Curso:** Contiene todas las vistas que tienen relación con los procesos administrativos relacionados a la gestión de cursos. Además de la vista general de cursos para administradores, se encuentra la vista de cursos para profesores y alumnos, la vista de asignar y desasignar competencias a cursos, la vista de asignar y desasignar personal y la más importante: la vista de guardar plan de clases.
- **Vista Unidad:** Contiene todas las vistas que tienen relación con los procesos administrativos relacionados a la gestión de unidades. Además de la vista general de unidades para administradores y profesores, se encuentra la vista de asignar y desasignar saberes a unidades.
- **Vista Actividad:** Contiene todas las vistas que tienen relación con los procesos administrativos relacionados a la gestión de actividades. Además de la vista general de actividades para administradores y profesores, se encuentra la vista de asignar y desasignar saberes a actividades.
- **Vista Competencia:** Contiene todas las vistas que tienen relación con los procesos administrativos relacionados a la gestión de competencias.
- **Vista Aprendizaje:** Contiene todas las vistas que tienen relación con los procesos administrativos relacionados a la gestión de aprendizajes. Además de la vista general de aprendizajes, se encuentra la vista de asignar y desasignar aprendizajes a competencias.
- **Vista Saber:** Contiene todas las vistas que tienen relación con los procesos administrativos relacionados a la gestión de saberes. Además de la vista general de saberes, se encuentra la vista de asignar y desasignar saberes a aprendizajes.

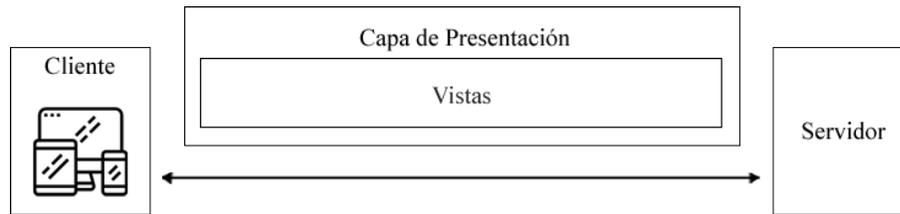


Figura 5.3: Diagrama general de representación de la Vista de la Arquitectura Lógica

En la *capa de procesamiento* se encuentran alojados los **controladores** del sistema dentro del módulo controlador, los cuales se encargan de definir la forma en la que la interfaz reacciona a las entradas del usuario [25], siendo el nexo entre la vista, el modelo y otros controladores de ser necesarios como se puede observar en la Figura 5.4. El detalle de cada módulo se explica en la siguiente sección.

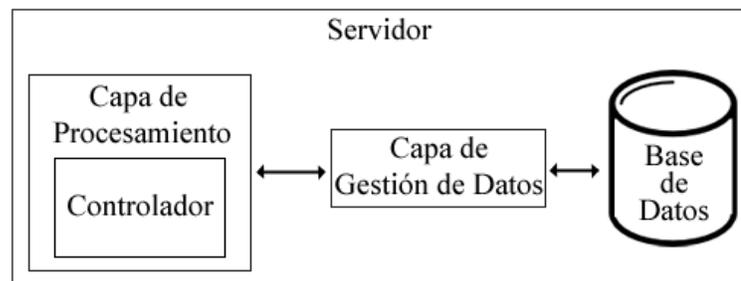


Figura 5.4: Diagrama general de representación del Controlador de la Arquitectura Lógica

Finalmente, en la *capa de gestión de datos* es donde se encuentra tanto el modelo como los controladores que tienen relación directa con la base de datos como se puede observar en la Figura 5.5 que fue denominado Data, que también se puede observar en la Figura 5.2. El **modelo** es representado como la estructura central de los datos y del sistema en sí [32]. En la siguiente sección se puede observar el modelo a nivel de diseño de componentes que forma parte de la base de la construcción del sistema.

5.3. Diagrama de Clases

En esta sección se presenta el diagrama de clases, con la finalidad de definir la estructura del sistema y modelar a un nivel más detallado las relaciones entre módulos definidos en las Secciones 5.1 y 5.2 (arquitectura física y lógica). Dado que

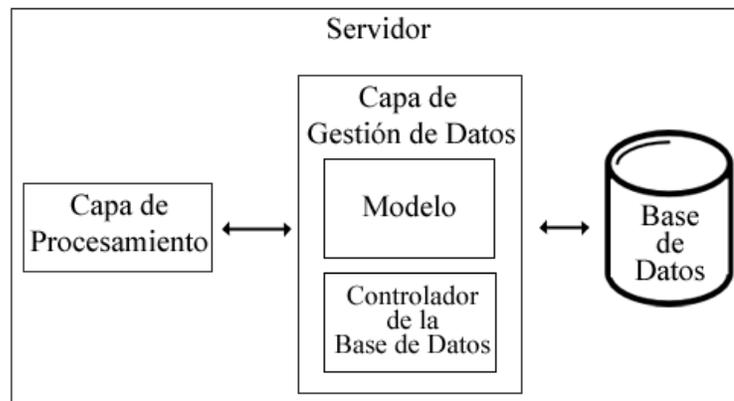


Figura 5.5: Diagrama general de representación del Modelo de la Arquitectura Lógica

en la Figura 5.2 se puede observar que según la arquitectura MVC¹ el sistema es estructurado en tres tipos de contenedores, el diagrama de clases sigue la misma organización, como se puede observar en las Figuras 5.6, 5.7, 5.8 y 5.9.

5.3.1. Vista

Éste módulo contiene las vistas del sistema. Se pueden observar las clases del sistema de forma más exhaustiva, divididos por los módulos descritos en la Sección 5.2.

5.3.2. Controlador

Éste módulo contiene las clases principales del sistema. Son quienes se encargan de implementar las interfaces del software, conectarlas con el modelo y con la base de datos y realizar las operaciones primordiales del prototipo.

Clases del Módulo

- **Controlador Cursos:** Clase que se encarga de controlar y visualizar todas aquellas funcionalidades dedicadas a un curso propiamente tal (gestionar cursos, asignar cursos a usuarios, asociar cursos a competencias y guardar un plan de clases). Se encuentra conectada con la vista unidad (index).

¹Modelo Vista Controlador

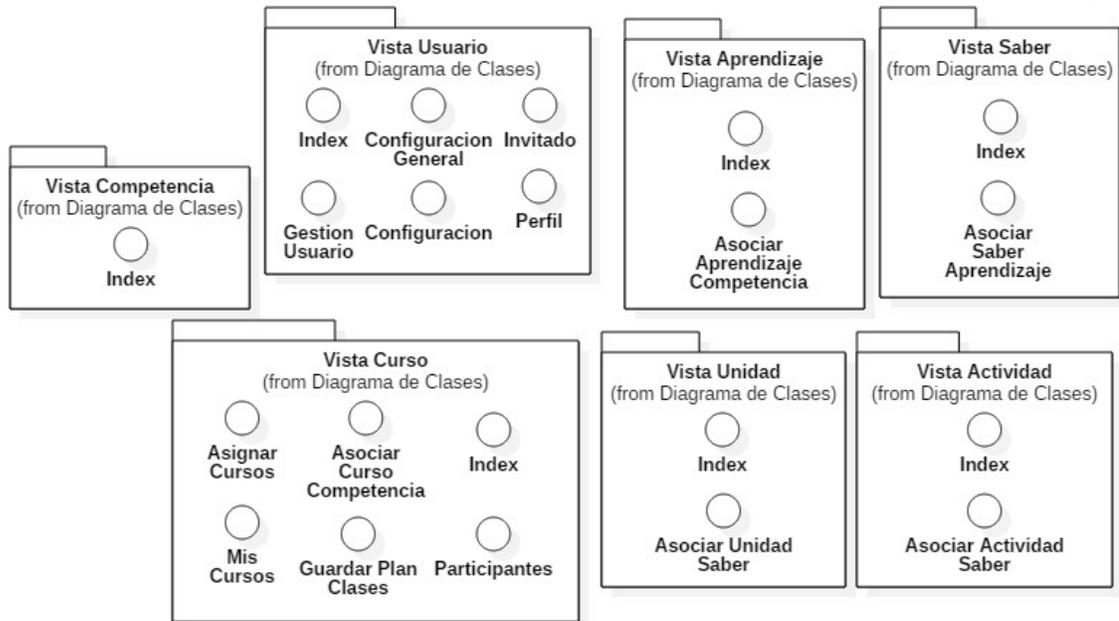


Figura 5.6: Diagrama de Clases: Vista

- **Controlador Unidades:** Clase que se encarga de controlar y visualizar todas aquellas funcionalidades dedicadas a unidades (gestionar unidades y asociar saberes de cursos a unidades). Se encuentra conectada con la vista actividad (index) y con la vista curso (index).
- **Controlador Actividades:** Clase que se encarga de controlar y visualizar todas aquellas funcionalidades dedicadas a actividades (gestionar actividades y asociar saberes de unidades a actividades). Se encuentra conectada con la vista unidad (index) y con la vista curso (index)
- **Controlador Competencias:** Clase que se encarga de controlar y visualizar la funcionalidad dedicada a competencias (gestionar competencias).
- **Controlador Aprendizajes:** Clase que se encarga de controlar y visualizar todas aquellas funcionalidades dedicadas a aprendizajes (gestionar aprendizajes y asociar aprendizajes a competencias).
- **Controlador Saberes:** Clase que se encarga de controlar y visualizar todas aquellas funcionalidades dedicadas a saberes (gestionar saberes y asociar saberes a aprendizajes).

- **Controlador Usuario:** Clase que se encarga de controlar y visualizar todas aquellas funcionalidades dedicadas a usuarios (vista principal, gestionar usuarios, configuración general del prototipo, iniciar sesión, ver perfil y configuración de la cuenta). Se encuentra conectada con la vista curso (index).

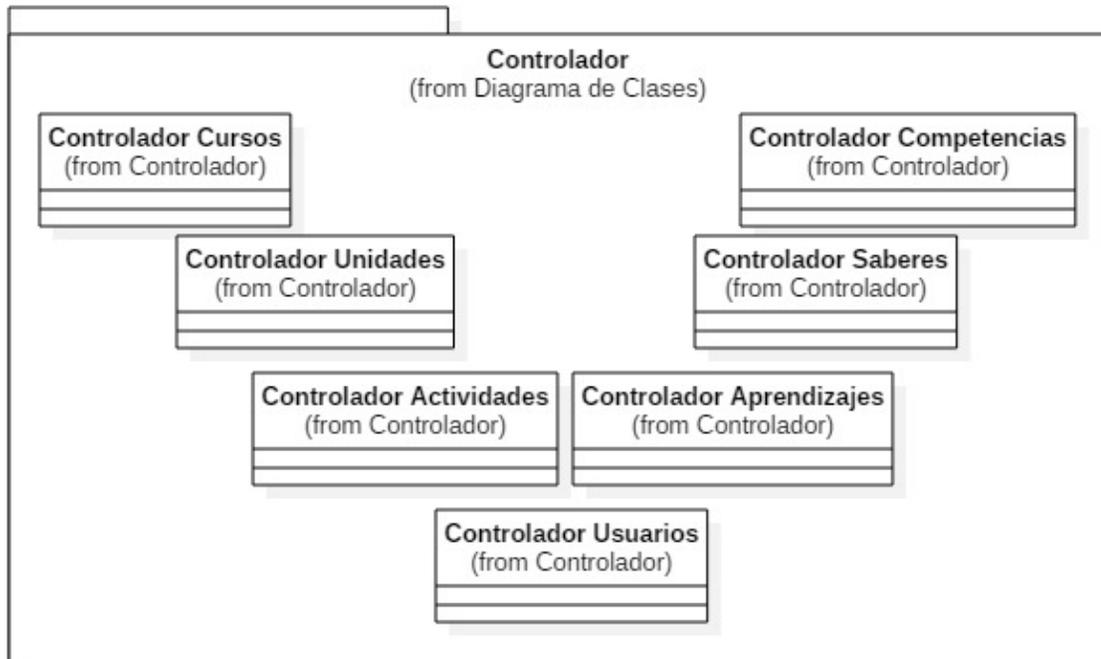


Figura 5.7: Diagrama de Clases: Controlador

5.3.3. Data

Al igual que el controlador, es una clase que se encarga de realizar operaciones primordiales del prototipo. Dada la arquitectura, se podría decir que es parte del controlador. Sin embargo, el módulo fue separado debido a que la funcionalidad de data se encarga de realizar operaciones sólo relacionadas con la base de datos.

Clases del Módulo

- **DataSource:** Clase principal del sistema, que es el nexo entre las clases dedicadas a cada una de las entidades principales del modelo, el modelo y todos los controladores. Si bien es una de las clases críticas del modelo al poseer tanta

dependencia, resulta mucho más fácil de modificar y reduce el acoplamiento entre las operaciones de la base de datos y los controladores, al necesitar utilizar más de una clase del módulo data.

- **DataCurso:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad Curso. A través de la clase DataSource, se encuentra conectada con el controlador cursos, el controlador unidades, el controlador usuarios y el modelo.
- **DataUnidad:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad unidad. A través de la clase DataSource, se encuentra conectada con el controlador cursos, el controlador actividades, el controlador usuarios y el modelo.
- **DataActividad:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad actividad. A través de la clase DataSource, se encuentra conectada con el controlador cursos, el controlador unidades, el controlador actividades, el controlador usuarios y el modelo.
- **DataCompetencia:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad competencia. A través de la clase DataSource, se encuentra conectada con el controlador competencias, el controlador cursos, el controlador usuarios y el modelo.
- **DataAprendizaje:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad aprendizaje. A través de la clase DataSource, se encuentra conectada con el controlador aprendizaje, el controlador unidad, el controlador usuarios y el modelo.
- **DataSaber:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad saber. A través de la clase DataSource, se encuentra conectada con el controlador saber, el controlador unidad, el controlador actividad, el controlador usuarios y el modelo.
- **DataUsuario:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad usuario. A través de la clase DataSource, se encuentra conectada con todos los controladores y el modelo.

- **DataConfiguracion:** Clase que se encarga controlar todas aquellas operaciones dedicadas a la entidad configuracionBasica. A través de la clase DataSource, se encuentra conectada con todos los controladores y el modelo.

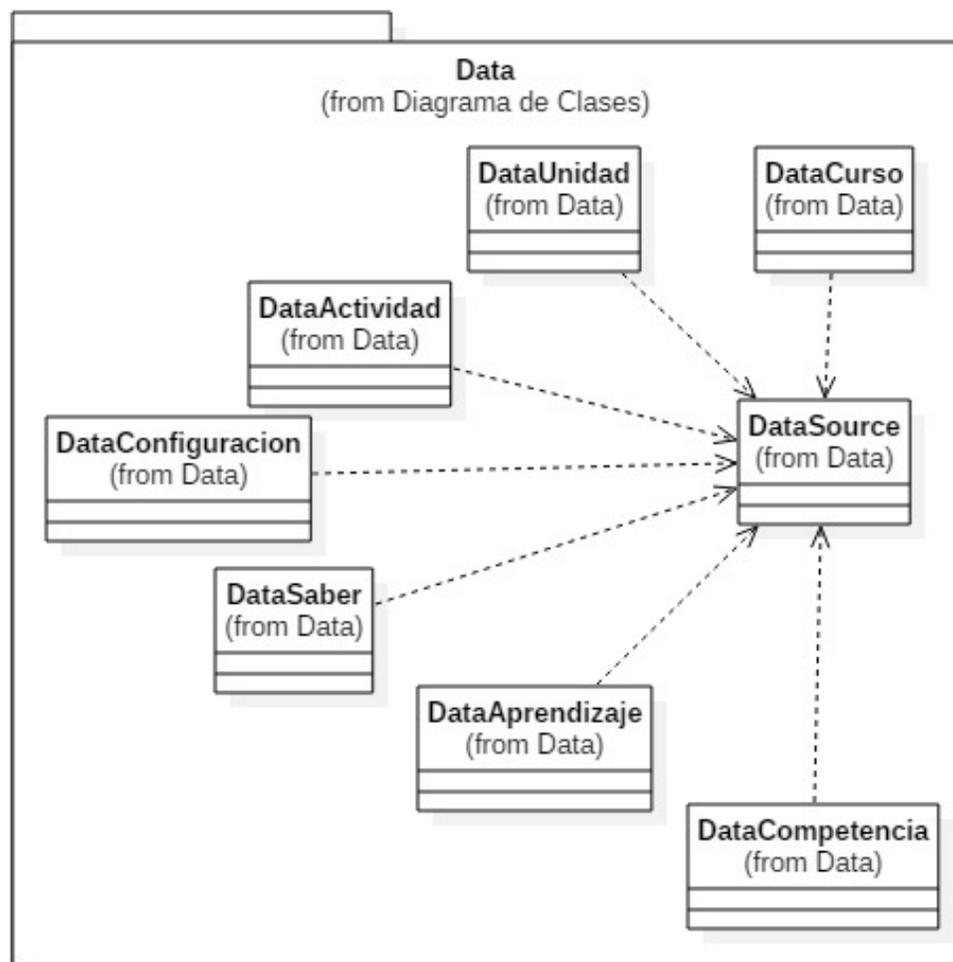


Figura 5.8: Diagrama de Clases: Data

5.3.4. Modelo

Según [24], un diagrama de clases UML² “*aporta una visión estática o de estructura del sistema, sin mostrar la naturaleza dinámica de las comunicaciones entre los objetos de las clases*”. Es decir, proporciona una visión de las clases principales que

²Lenguaje de Modelado Unificado

corresponden a la estructura del sistema y muestran como éste se comporta en relación a sus objetos y atributos. Según se puede observar en la Figura 5.2, el modelo interactúa principalmente con los módulos de controlador y data, formando parte de prácticamente todas sus clases.

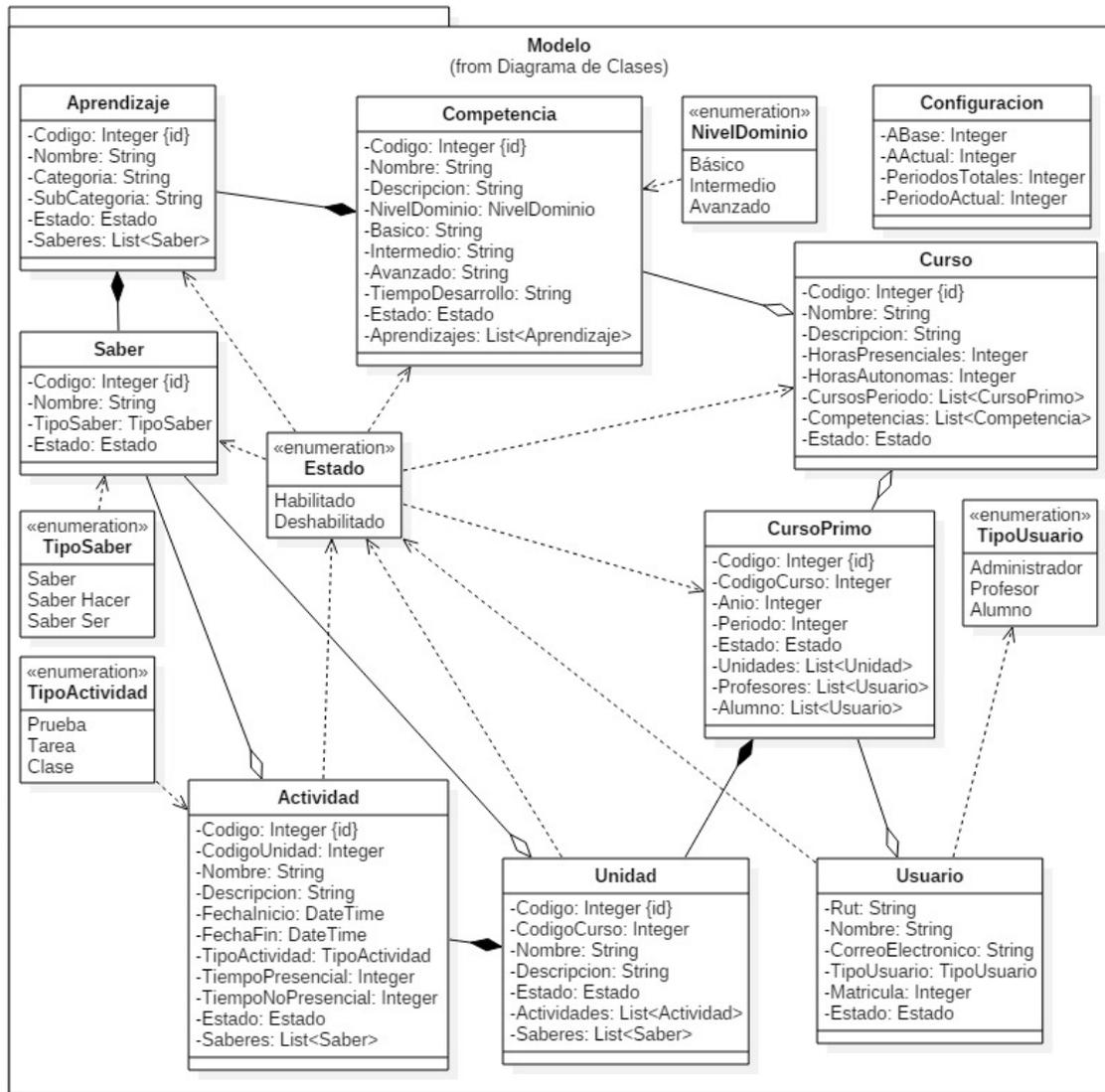


Figura 5.9: Diagrama de Clases: Modelo

Relaciones entre Objetos

Dado el diagrama de clases detallado en la figura 5.9, las relaciones entre clases se componen de la siguiente forma:

- **Competencia** → **Aprendizaje**: Una competencia puede contener muchos aprendizajes, que son dependientes de la competencia creada.
- **Aprendizaje** → **Saber**: Un aprendizaje puede contener muchos saberes, que son dependientes del aprendizaje creado.
- **Curso** → **Competencia**: Un curso puede contener muchas competencias, que son independientes del curso creado.
- **Curso** → **CursoPrimo**: Un curso puede contener muchos periodos en un año determinado, que son dependientes del curso creado.
- **CursoPrimo** → **Usuario**: Un periodo de un curso puede contener muchos usuarios, que son independientes del curso y del periodo creado.
- **CursoPrimo** → **Unidad**: Un periodo de un curso puede contener muchas unidades, que son dependientes del curso y del periodo creado.
- **Unidad** → **Actividad**: Una unidad puede contener muchas actividades, que son dependientes de la actividad creada.
- **Saber** → **Unidad**: A una unidad pueden asociárseles muchos saberes que forman parte de las competencias asociadas a un curso. Los saberes son independientes de la unidad creada, pero dependientes de la competencia asociada al curso.
- **Saber** → **Actividad**: A una actividad pueden asociárseles muchos saberes que fueron asociadas a una unidad, que forman parte de las competencias asociadas a un curso. Los saberes son independientes de la actividad creada, pero dependientes de la competencia asociada al curso.
- **Estado** → **Todo**: Todas las clases existentes en el modelo (con la excepción de la clase configuración) son dependientes de la clase estado, debido al RU04 descrito en el Cuadro 4.1.

- **TipoUsuario** → **Usuario**: La clase tipoUsuario es dependiente de la clase usuario.
- **NivelDominio** → **Competencia**: La clase nivelDominio es dependiente de la clase competencia.
- **TipoSaber** → **Saber**: La clase tipoSaber es dependiente de la clase saber.
- **TipoActividad** → **Actividad**: La clase tipoActividad es dependiente de la clase actividad.

5.4. Diagrama Entidad-Relación

En esta sección se presenta el modelo entidad relación, que simboliza los datos que son generados dentro del prototipo, siendo éste la representación de la base de datos. El modelo contiene trece entidades relacionales que describen las características de la información procesada.

5.4.1. Relaciones del Modelo

Dado el modelo entidad relación detallado en la Figura 5.10, las relaciones entre entidades se componen de la siguiente forma:

- **Competencia (Codigo)** → **Aprendizaje (CodigoCompetencia)**: Una competencia puede tener muchos aprendizajes. Pero un aprendizaje, sólo puede ser asociado a una competencia.
- **Aprendizaje (Codigo)** → **Saber (CodigoAprendizaje)**: Un aprendizaje puede tener muchos saberes. Pero un saber, sólo puede ser asociado a un aprendizaje.
- **Aprendizaje (Codigo)** → **Saber (CategoriaAprendizaje)**: Un aprendizaje puede tener muchos saberes. Pero un saber, sólo puede tener la categoría del aprendizaje asociado.
- **CompetenciaCurso (CodigoCompetencia)** → **Competencia (Codigo)**: Una competencia puede estar asociada a muchos cursos. Mientras que a un curso pueden serle asociadas muchas competencias.

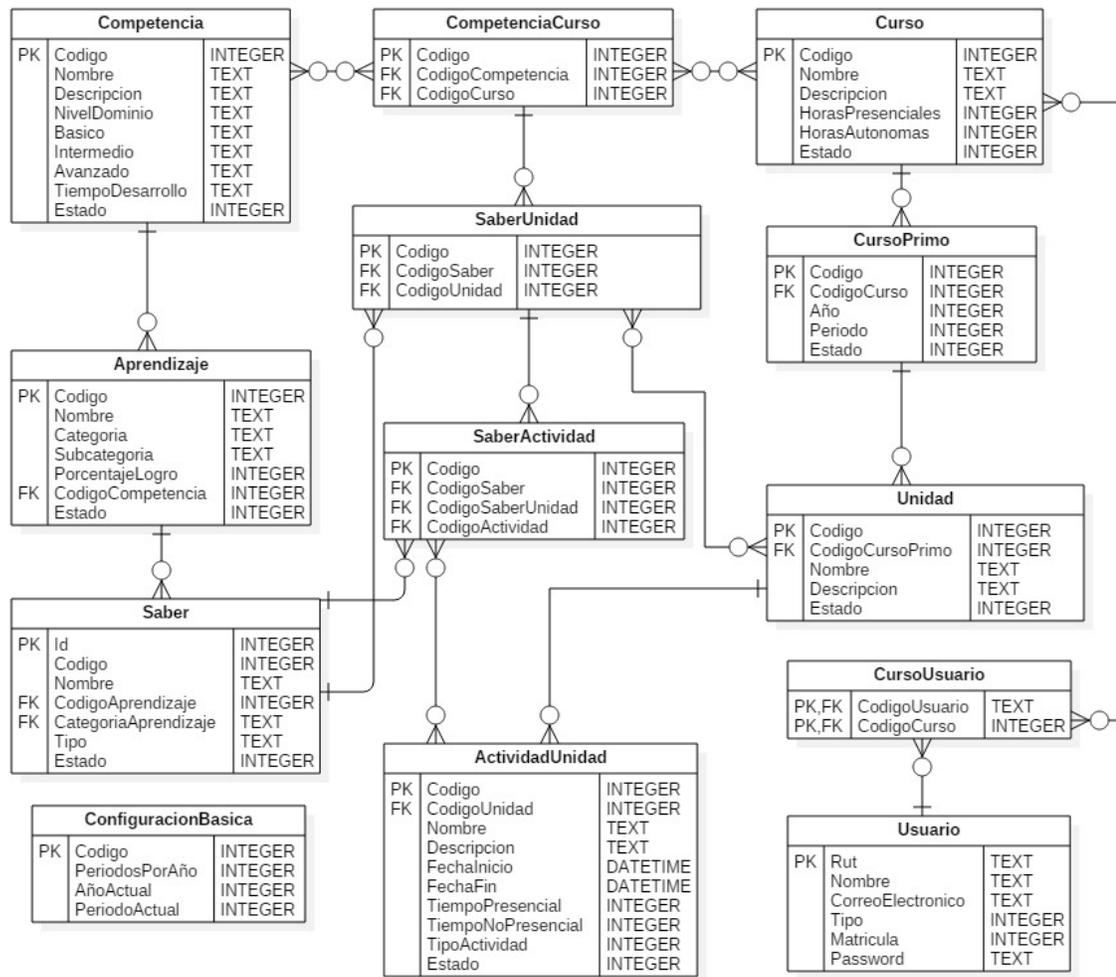


Figura 5.10: Diagrama Entidad-Relación

- **CompetenciaCurso (CodigoCurso) → Curso (Codigo):** Una competencia puede estar asociada a muchos cursos. Mientras que a un curso pueden serle asociadas muchas competencias.
- **Curso (Codigo) → CursoPrimo (CodigoCurso):** Un curso puede tener muchos periodos en ciertos años. Pero un periodo y un año en específico sólo pueden ser parte de un curso.
- **CursoPrimo (Codigo) → Unidad (CodigoCursoPrimo):** Un curso en un periodo y un año en específico puede tener muchas unidades. Pero una unidad está asociada sólo a un curso en un periodo y un año en específico.

- **Unidad (Codigo) → ActividadUnidad (CodigoUnidad):** Una unidad puede tener muchas actividades. Pero una actividad está asociada solamente a una unidad.
- **SaberUnidad (CodigoCompetenciaCurso) → CompetenciaCurso (Codigo):** La competencia asignada a un curso puede tener muchos saberes asignados, los cuales pueden ser asociados a muchas unidades. Mientras que una unidad puede tener muchos saberes asignados que pertenecen a una competencia que fue asignado a un curso.
- **SaberUnidad (CodigoSaber) → Saber (Id):** La competencia asignada a un curso puede tener muchos saberes asignados, los cuales pueden ser asociados a muchas unidades. Mientras que una unidad puede tener muchos saberes asignados que pertenecen a una competencia que fue asignado a un curso.
- **SaberUnidad (CodigoUnidad) → Unidad (Codigo):** La competencia asignada a un curso puede tener muchos saberes asignados, los cuales pueden ser asociados a muchas unidades. Mientras que una unidad puede tener muchos saberes asignados que pertenecen a una competencia que fue asignado a un curso.
- **SaberActividad (CodigoSaberUnidad) → SaberUnidad (Codigo):** Un saber asignado a una unidad, puede ser asociado a múltiples actividades. Asimismo, un saber asignado a una actividad puede estar asociado a múltiples unidades.
- **SaberActividad (CodigoSaber) → Saber(Id):** Un saber asignado a una unidad, puede ser asociado a múltiples actividades. Mientras que un saber asignado a una actividad puede estar asociado a múltiples unidades.
- **SaberActividad (CodigoActividad) → ActividadUnidad (Codigo):** Un saber asignado a una unidad, puede ser asociado a múltiples actividades. Asimismo, un saber asignado a una actividad puede estar asociado a múltiples unidades.
- **CursoUsuario (CodigoCurso) → Curso (Codigo):** Un usuario puede ser asignado a múltiples cursos. Asimismo, un curso puede ser asociado a múltiples usuarios.

- **CursoUsuario (CodigoUsuario) → Usuario (Rut):** Un usuario puede ser asignado a múltiples cursos. Asimismo, un curso puede ser asociado a múltiples usuarios.

5.5. Resumen

En este capítulo se presentan los diagramas que forman parte del diseño del prototipo. La arquitectura física se utiliza para identificar los componentes que participan dentro de la solución; siendo la elegida la arquitectura cliente servidor priorizando la sobrecarga del lado del servidor. En concordancia, la arquitectura lógica elegida fue el modelo vista controlador, principalmente para beneficiar la modificación de componentes sobre todo al disponer de una metodología de desarrollo donde es probable que el diseño cambie en cierta medida. Finalmente, el diagrama de clases define la estructura y el diagrama entidad relación, representa los datos generados dentro del prototipo.

6. Desarrollo

En este capítulo se detallan los procesos de desarrollo e implementación del prototipo, además de las principales decisiones tomadas para poder facilitar la construcción de éste. Dado que la metodología escogida es la metodología de desarrollo tradicional iterativo incremental, cada paso durante el desarrollo se realiza en secuencia para poder producir entregables como se explica en la Sección 2.4.1. Cada uno de esos entregables fueron probados, en base a los requisitos y características que se esperaban conseguir.

Con el motivo de poder ofrecer al lector una mayor comprensión sobre el desarrollo, en primer lugar se describen las herramientas utilizadas en el desarrollo (cuyas tecnologías se encuentran detalladas en la sección 2.2), luego cada una de las iteraciones donde se detallan el trabajo realizado en cada una de ellas y finalmente, se explicará a grandes rasgos el funcionamiento del prototipo.

6.1. Iteraciones

La duración de cada iteración viene determinada en su mayoría por la fecha de entrega. Sin embargo, cada semana se realizan reuniones de avance y entrega con el cliente, con la finalidad de hacerlo parte de la planificación y del desarrollo del software propiamente tal. A pesar de que la metodología dicta que los entregables son esencialmente software, se acordó que dos entregables se basarían en documentos.

Las iteraciones son presentadas en cuadros, donde la primera columna corresponde a la **tarea**. La segunda columna, llamada **F Inicial** y **F Final**, corresponden a las fechas de inicio y fin de la tarea. La última columna, llamada **RC** corresponden a los requisitos de sistema o restricciones de software comprometidos. En las filas

debajo de la tarea se describen las subtareas que se crean con el fin de llegar a la tarea completada.

En la Iteración 1 se realizan las primeras reuniones con el cliente para así poder desarrollar el Documento de Requisitos. Este documento incluye los requisitos de usuario y sistema, los casos de uso de nivel 0 y las restricciones generales. La Ficha de la Iteración 1 se puede ver en el Cuadro 6.1.

Cuadro 6.1: Iteración 1

Tarea	F Inicial - F Final	RC
Generar documento de requisitos	12/04/2018 - 02/04/2018	-
<i>SubTareas</i>		
Realizar levantamiento de requisitos		
Redactar propósito, alcance y contexto		
Formalizar requisitos de usuario y de sistema		
Generar diagrama de caso de uso nivel 0		
Actualizar requisitos de usuario y sistema		
Redactar restricciones generales		

A partir de la Iteración 2, descrita en el Cuadro 6.2 se pudo ofrecer al usuario una muestra de cómo lucirá la aplicación. En esta iteración se desarrollaron los diagramas de arquitectura física, lógica, de clases y entidad-relación, además de crear un prototipo simple que puede visualizarse en el Anexo C, Figuras C.1 a C.17.

Cuadro 6.2: Iteración 2

Tarea	F Inicial - F Final	RC
Generar documento de diseño	02/05/2018 - 14/05/2018	-
<i>SubTareas</i>		
Crear diagrama arquitectura física		
Crear diagrama arquitectura lógica		
Crear diagrama de clases		
Crear diagrama entidad-relación		
Investigar manual de marca universidad		
Crear prototipo		

Como se puede observar en el Cuadro 6.3, en la Iteración 3 se comenzó con el desarrollo del prototipo, abordando todos aquellos procesos relacionados a la base

de datos. Esto incluye traspasar el diagrama entidad-relación a código SQL, crear la base de datos y las sentencias SQL básicas, programar la conexión de la base de datos con el software y hacer un testeo básico.

Se debe destacar que, a partir de la Iteración 3 y hasta que comenzó a rehacerse la interfaz en la Iteración 14, la primera vista de la aplicación con base en el primer prototipo se puede observar en el Anexo C, Figuras C.18 a C.31. Asimismo, se decide junto con el cliente que no es necesario especificar el tipo de profesor cuando se crea uno. También que el tipo de actividades debe ser escalable, debido a que la cantidad de éstas puede variar en el futuro.

Cuadro 6.3: Iteración 3

Tarea	F Inicial - F Final	RC
Crear base de datos	14/05/2018 - 28/05/2018	RG02, RG03, RG04, RG05, RG09
<i>SubTareas</i>		
Traspasar entidad-relación a lenguaje SQL		
Crear base de datos		
Crear aplicación base		
Realizar nexos base de datos – aplicación base		
Crear sentencias SQL básicas para CRUD		
Generar código que soporte cada sentencia SQL generada		
Testear base de datos		

A partir de la Iteración 4 descrita en el Cuadro 6.4, los productos a entregar al cliente son tangibles para él, pero con una funcionalidad mínima que fue desarrollándose a lo largo del tiempo. En esta Iteración se tuvo que añadir el RU03, debido a que el cliente olvidó mencionar la necesidad de guardar un plan de clases, por lo que se modificaron las iteraciones y se reorganizaron en base a sus necesidades. Se desarrollan las primeras funcionalidades del sistema, como ver una competencia y crearla, además de una parte de las funciones modificar, habilitar y deshabilitar competencias.

La ejecución de pruebas de usabilidad se deja para el final, debido a correcciones sugeridas durante iteraciones anteriores por el cliente y diferentes usuarios de prueba.

Cuadro 6.4: Iteración 4

Tarea	F Inicial - F Final	RC
CRUD Competencias	28/05/2018 - 03/06/2018	RS39, RS40, RS41, RS42, RS43
<i>SubTareas</i>		
Crear vista		
Crear backend mostrar competencias		
Crear clase competencia		
Crear enum nivelDominio		
Crear enum estado		
Crear controlador mostrar competencias		
Crear repositorio mostrar competencias		
Enlazar vista-controlador-repositorio mostrar competencias		
Crear backend crear competencias		
Crear controlador crear competencias		
Crear repositorio crear competencias		
Enlazar vista-controlador-repositorio crear competencias		
Crear controlador modificar competencias		
Crear repositorio modificar competencias		
Enlazar vista-controlador-repositorio modificar competencias		
Crear controlador habilitar competencias		
Crear repositorio habilitar competencias		
Enlazar vista-controlador-repositorio habilitar competencias		
Crear controlador deshabilitar competencias		
Crear repositorio deshabilitar competencias		
Enlazar vista-controlador-repositorio deshabilitar competencias		

En la Iteración 5 descrita en los Cuadros 6.5 y 6.6 se ofrece al cliente las funcionalidades asociadas a los aprendizajes, como ver y crear un aprendizaje, además de una parte de las funciones modificar, habilitar y deshabilitar éstos.

Cuadro 6.5: Iteración 5

Tarea	F Inicial - F Final	RC
CRUD Aprendizajes	04/06/2018 - 10/06/2018	RS44, RS45, RS46, RS47, RS50
<i>SubTareas</i>		
Crear vista		

Cuadro 6.6: Iteración 5

Tarea	F Inicial - F Final	RC
CRUD Aprendizajes	04/06/2018 - 10/06/2018	RS44, RS45, RS46, RS47, RS50
SubTareas		
Crear backend mostrar aprendizajes		
Crear clase aprendizaje		
Crear controlador mostrar aprendizajes		
Crear repositorio mostrar aprendizajes		
Enlazar vista-controlador-repositorio mostrar aprendizajes		
Crear backend crear aprendizajes		
Crear controlador crear aprendizajes		
Crear repositorio crear aprendizajes		
Enlazar vista-controlador-repositorio crear aprendizajes		
Crear controlador modificar aprendizajes		
Crear repositorio modificar aprendizajes		
Enlazar vista-controlador-repositorio modificar aprendizajes		
Crear controlador habilitar aprendizajes		
Crear repositorio habilitar aprendizajes		
Enlazar vista-controlador-repositorio habilitar aprendizajes		
Crear controlador deshabilitar aprendizajes		
Crear repositorio deshabilitar aprendizajes		
Enlazar vista-controlador-repositorio deshabilitar aprendizajes		

Como se puede observar en los Cuadros 6.7 y 6.8, durante la Iteración 6 el cliente decidió priorizar la creación y mantenimiento de cuentas de usuario, por lo que se le ofrecen las funcionalidades asociadas a éstos.

Cuadro 6.7: Iteración 6

Tarea	F Inicial - F Final	RC
CRUD Usuarios	18/06/2018 - 24/06/2018	RS02, RS03, RS04, RS05, RS06, RS07
SubTareas		
Crear vista		
Crear backend mostrar usuarios		
Crear clase usuario		
Crear enum tipoUsuario		

Cuadro 6.8: Iteración 6

Tarea	F Inicial - F Final	RC
CRUD Usuarios	18/06/2018 - 24/06/2018	RS02, RS03, RS04, RS05, RS06, RS07
<i>SubTareas</i>		
Crear vista		
Crear backend mostrar usuarios		
Crear clase usuario		
Crear enum tipoUsuario		
Crear controlador mostrar usuarios		
Crear repositorio mostrar usuarios		
Enlazar vista-controlador-repositorio mostrar usuarios		
Crear backend crear usuarios		
Crear controlador crear usuarios		
Crear repositorio crear usuarios		
Enlazar vista-controlador-repositorio crear usuarios		
Crear controlador modificar usuarios		
Crear repositorio modificar usuarios		
Enlazar vista-controlador-repositorio modificar usuarios		
Crear controlador habilitar usuarios		
Crear repositorio habilitar usuarios		
Enlazar vista-controlador-repositorio habilitar usuarios		
Crear controlador deshabilitar usuarios		
Crear repositorio deshabilitar usuarios		
Enlazar vista-controlador-repositorio deshabilitar usuarios		
Cambiar Contraseña		

Durante la Iteración 7 detallada en los Cuadros 6.9 y 6.10, el enfoque está en los saberes, otorgando al cliente las funcionalidades de ver y crear un saber, además de parte de las funciones modificar, habilitar y deshabilitar éstos.

Cuadro 6.9: Iteración 7

Tarea	F Inicial - F Final	RC
CRUD Saberes	11/06/2018 - 17/06/2018	RS51, RS52, RS53, RS54, RS57
<i>SubTareas</i>		
Crear vista		

Cuadro 6.10: Iteración 7

Tarea	F Inicial - F Final	RC
CRUD Saberes	11/06/2018 - 17/06/2018	RS51, RS52, RS53, RS54, RS57
<i>SubTareas</i>		
Crear backend mostrar saberes		
Crear clase saber		
Crear enum tipoSaber		
Crear controlador mostrar saberes		
Crear repositorio mostrar saberes		
Enlazar vista-controlador-repositorio mostrar saberes		
Crear backend crear saberes		
Crear controlador crear saberes		
Crear repositorio crear saberes		
Enlazar vista-controlador-repositorio crear saberes		
Crear controlador modificar saberes		
Crear repositorio modificar saberes		
Enlazar vista-controlador-repositorio modificar saberes		
Crear controlador habilitar saberes		
Crear controlador habilitar saberes		
Crear repositorio habilitar saberes		
Enlazar vista-controlador-repositorio habilitar saberes		
Crear controlador deshabilitar saberes		
Crear repositorio deshabilitar saberes		
Enlazar vista-controlador-repositorio deshabilitar saberes		

La Iteración 8 detallada en los Cuadros 6.11 y 6.12, se ofrece al cliente las funcionalidades asociadas a los módulos o cursos, como ver y crear un curso, además de una parte de las funciones modificar, habilitar y deshabilitar éstos.

Cuadro 6.11: Iteración 8

Tarea	F Inicial - F Final	RC
CRUD Cursos	25/06/2018 - 01/07/2018	RS12, RS13, RS14, RS15, RS22, RS23, RS24
<i>SubTareas</i>		
Crear vista		

Cuadro 6.12: Iteración 8

Tarea	F Inicial - F Final	RC
CRUD Cursos	25/06/2018 - 01/07/2018	RS12, RS13, RS14, RS15, RS22, RS23, RS24
<i>SubTareas</i>		
Crear backend mostrar cursos		
Crear clase curso		
Crear controlador mostrar cursos		
Crear repositorio mostrar cursos		
Enlazar vista-controlador-repositorio mostrar cursos		
Crear backend crear cursos		
Crear controlador crear cursos		
Crear repositorio crear cursos		
Enlazar vista-controlador-repositorio crear cursos		
Crear controlador modificar cursos		
Crear repositorio modificar cursos		
Enlazar vista-controlador-repositorio modificar cursos		
Crear controlador habilitar cursos		
Crear repositorio habilitar cursos		
Enlazar vista-controlador-repositorio habilitar cursos		
Crear controlador deshabilitar cursos		
Crear repositorio deshabilitar cursos		
Enlazar vista-controlador-repositorio deshabilitar cursos		

Como se puede observar en la Iteración 9 descrita en los Cuadros 6.13 y 6.14, se ofrece al cliente las funcionalidades asociadas a los módulos o cursos por periodo, como ver y crear un curso, además de una parte de la función modificar.

Cuadro 6.13: Iteración 9

Tarea	F Inicial - F Final	RC
CRUD Cursos por periodo	02/07/2018 - 08/07/2018	RS12, RS13, RS14, RS15, RS22, RS23, RS24
<i>SubTareas</i>		
Crear vista		
Crear backend mostrar cursos por periodo		

Cuadro 6.14: Iteración 9

Tarea	F Inicial - F Final	RC
CRUD Cursos por periodo	02/07/2018 - 08/07/2018	RS12, RS13, RS14, RS15, RS22, RS23, RS24
SubTareas		
Crear clase cursoPrimo		
Crear controlador mostrar cursos por periodo		
Crear repositorio mostrar cursos por periodo		
Enlazar vista-controlador-repositorio mostrar cursos por periodo		
Crear backend crear cursos por periodo		
Crear controlador crear cursos por periodo		
Crear repositorio crear cursos por periodo		
Enlazar vista-controlador-repositorio crear cursos por periodo		
Crear controlador modificar cursos por periodo		
Crear repositorio modificar cursos por periodo		
Enlazar vista-controlador-repositorio modificar cursos por periodo		

Debido a dificultades durante la implementación y un cambio en la vista a petición del cliente, se decidió dividir la funcionalidad asociada a los módulos o cursos por periodo en dos. En esta iteración detallada en los Cuadros 6.15 y 6.16, se ofrece al cliente parte de las funcionalidades habilitar y deshabilitar, además de las asignaciones de profesores y alumnos a los cursos.

Se decide junto al cliente que cuando se crea un saber en un curso, éste debe ser duplicado. También se debe asignar al aprendizaje y a la competencia de forma automática. Sin embargo, aquello queda pendiente de evaluación, debido a que es la competencia la que se asocia al curso, no el saber propiamente tal.

Cuadro 6.15: Iteración 10

Tarea	F Inicial - F Final	RC
CRUD Cursos por periodo 2	09/07/2018 - 15/07/2018	RS12, RS13, RS14, RS15, RS22, RS23, RS24
SubTareas		
Modificar vista		
Crear controlador habilitar cursos por periodo		

Cuadro 6.16: Iteración 10

Tarea	F Inicial - F Final	RC
CRUD Cursos por periodo 2	09/07/2018 - 15/07/2018	RS12, RS13, RS14, RS15, RS22, RS23, RS24
<i>SubTareas</i>		
Crear repositorio habilitar cursos por periodo		
Enlazar vista-controlador-repositorio habilitar cursos por periodo		
Crear controlador deshabilitar cursos por periodo		
Crear repositorio deshabilitar cursos por periodo		
Enlazar vista-controlador-repositorio deshabilitar cursos por periodo		
Asignar profesor a curso		
Desasignar profesor a curso		
Asignar alumno a curso		
Desasignar alumno a curso		

En la Iteración 11 descrita en los Cuadros 6.17 y 6.18 se ofrece al cliente las funcionalidades asociadas a las unidades, tales como ver y crear una unidad, además de un sector de las funciones modificar, habilitar y deshabilitar éstos.

En profesor del módulo sugiere que la vista del profesor debía ser el enfoque del código. Tras una consulta con el cliente, se decide dejar la autenticación para más adelante.

Cuadro 6.17: Iteración 11

Tarea	F Inicial - F Final	RC
CRUD Unidades	16/07/2018 - 22/07/2018	RS25, RS26, RS27, RS28, RS31
<i>SubTareas</i>		
Crear vista		
Crear backend mostrar unidades		
Crear clase unidad		
Crear controlador mostrar unidades		
Crear repositorio mostrar unidades		
Enlazar vista-controlador-repositorio mostrar unidades		
Crear backend crear unidades		
Crear controlador crear unidades		

Cuadro 6.18: Iteración 11

Tarea	F Inicial - F Final	RC
CRUD Unidades	16/07/2018 - 22/07/2018	RS25, RS26, RS27, RS28, RS31
SubTareas		
Crear repositorio crear unidades		
Enlazar vista-controlador-repositorio crear unidades		
Crear controlador modificar unidades		
Crear repositorio modificar unidades		
Enlazar vista-controlador-repositorio modificar unidades		
Crear controlador habilitar unidades		
Crear repositorio habilitar unidades		
Enlazar vista-controlador-repositorio habilitar unidades		
Crear controlador deshabilitar unidades		
Crear repositorio deshabilitar unidades		
Enlazar vista-controlador-repositorio deshabilitar unidades		

La siguiente iteración detallada en el Cuadro 6.19, se comienza posterior al inicio del segundo semestre. Se abordan las funciones que permiten que las competencias, aprendizajes, saberes, cursos, cursos por periodos y unidades puedan modificar, habilitar y deshabilitar sus datos correspondientes. También se realiza un cambio de repositorio, con la finalidad de hacer del proyecto algo privado.

A partir de esta iteración y posterior a una reunión con el cliente se considera que, si una competencia es modificada y esta es asociada a un curso, debe crearse una competencia completamente nueva con las modificaciones. Además, si un curso es deshabilitado, los periodos, las unidades y las actividades serán deshabilitadas.

El profesor de un curso sugiere que se considere la trayectoria de un saber. Es decir, que un alumno sea capaz de ver los saberes asociados en una unidad. El cliente decide que a pesar de que la trayectoria de un saber no es prescindible para el prototipo, si estaba de acuerdo con que el alumno sea capaz de ver los saberes, pero de una actividad.

Cuadro 6.19: Iteración 12

Tarea	F Inicial - F Final	RC
Procedimientos de base de datos faltantes y cambio de repositorio	04/08/2018 - 10/08/2018	RS03, RS04, RS05, RS13, RS14, RS15, RS26, RS27, RS28, RS40, RS41, RS42, RS45, RS46, RS47, RS52, RS53, RS54
<i>SubTareas</i>		
Crear backend modificar competencias		
Crear backend habilitar competencias		
Crear backend deshabilitar competencias		
Crear backend modificar aprendizajes		
Crear backend habilitar aprendizajes		
Crear backend deshabilitar aprendizajes		
Crear backend modificar saberes		
Crear backend habilitar saberes		
Crear backend deshabilitar saberes		
Crear backend modificar usuarios		
Crear backend habilitar usuarios		
Crear backend deshabilitar usuarios		
Crear backend modificar cursos		
Crear backend habilitar cursos		
Crear backend deshabilitar cursos		
Crear backend modificar cursos por periodo		
Crear backend habilitar cursos por periodo		
Crear backend deshabilitar cursos por periodo		
Crear backend modificar unidades		
Crear backend habilitar unidades		
Crear backend deshabilitar unidades		
Cambiar de repositorio		

Como se puede observar en la Iteración 13 descrita en el Cuadro 6.20, se ofrece al cliente las funcionalidades asociadas a los a las actividades, como ver, crear modificar, habilitar y deshabilitar una actividad.

Cuadro 6.20: Iteración 13

Tarea	F Inicial - F Final	RC
CRUD Actividades	11/08/2018 - 17/08/2018	RS32, RS33, RS34, RS35, RS38
SubTareas		
Crear vista		
Crear backend mostrar actividades		
Crear clase actividad		
Crear controlador mostrar actividades		
Crear repositorio mostrar actividades		
Enlazar vista-controlador-repositorio mostrar actividades		
Crear backend crear actividades		
Crear controlador crear actividades		
Crear repositorio crear actividades		
Enlazar vista-controlador-repositorio crear actividades		
Crear backend modificar actividades		
Crear controlador modificar actividades		
Crear repositorio modificar actividades		
Enlazar vista-controlador-repositorio modificar actividades		
Crear backend habilitar actividades		
Crear controlador habilitar actividades		
Crear repositorio habilitar actividades		
Enlazar vista-controlador-repositorio habilitar actividades		
Crear backend deshabilitar actividades		
Crear controlador deshabilitar actividades		
Crear repositorio deshabilitar actividades		
Enlazar vista-controlador-repositorio deshabilitar actividades		

En esta iteración se cambió la interfaz por completo, optando por utilizar una plantilla de Bootstrap orientada a la administración. Tras la reunión con el cliente se eligió a Genterella, detallada en la Sección 2.2. Los colores, sin embargo, debían intentar ser acordes a los colores institucionales a pesar de no existir un manual de marca para aquello ni un estándar por el que otras aplicaciones institucionales se rijan. El siguiente Cuadro 6.20 muestra el desarrollo de las vistas requeridas por el cliente; tanto las vistas de funcionalidades ya hechas, como la vista competencias, como vistas de funcionalidades que serán abordadas en las próximas iteraciones, como la vista de autenticación.

Se decide junto al cliente que cuando una competencia es deshabilitada, sólo la competencia es lo que se deshabilita, ni debe desasociarse de un módulo o curso. Cuando una actividad es modificada, se debe revisar si está asociada o no. De estarlo, se debe crear un duplicado de libre asociación. Esto se abordará en futuras correcciones.

Cuadro 6.21: Iteración 14

Tarea	F Inicial - F Final	RC
Interfaz	18/08/2018 - 24/08/2018	-
<i>SubTareas</i>		
Crear mapa de navegación		
Crear mapa de navegación		
Diseñar bosquejo vista inicio		
Crear vista inicio html-razor		
Diseñar bosquejo vista competencia		
Crear vista competencias html-razor		
Diseñar bosquejo vista aprendizaje		
Crear vista aprendizajes html-razor		
Diseñar bosquejo vista saber		
Crear vista saber html-razor		
Diseñar bosquejo vista curso		
Crear vista curso html-razor		
Diseñar bosquejo vista curso por periodo		
Crear vista curso por periodo html-razor		
Diseñar bosquejo vista usuario		
Crear vista usuario html-razor		
Diseñar bosquejo vista perfil		
Crear vista perfil html-razor		
Diseñar bosquejo vista autenticación		
Crear vista autenticación html-razor		
Diseñar bosquejo vista guardar plan de clases		
Crear vista guardar plan de clases html-razor		

Como se puede observar en el Cuadro 6.22, durante la Iteración 15 se comienza a trabajar en la funcionalidad esencial del proyecto. Sin embargo, debido a la complejidad de la tarea al cliente sólo se le muestra cómo se verá el plan de clases y la vista que deberá ejecutar. Esta iteración y la siguiente han estado en discusión desde el comienzo del proyecto, debido a que un plan de clases no tiene un estándar que lo rijan. Sin embargo, al final se decide por un estándar final que puede verse en la

Sección 6.2, Figura 6.12.

Cuadro 6.22: Iteración 15

Tarea	F Inicial - F Final	RC
Guardar plan de clases	25/08/2018 - 31/08/2018	RS09
<i>SubTareas</i>		
Investigar formas de generar y guardar archivos		
Pruebas con librerías PDF		
Crear backend guardar plan de clases		
Crear repositorio guardar plan de clases		

Debido a dificultades durante la implementación al no encontrar una librería vigente y compatible, se decidió dividir la iteración en dos. En esta iteración detallada en el Cuadro 6.23, se sigue con la investigación y las pruebas con librerías hasta encontrar una y finalmente se desarrolla el controlador para poder mostrarle al cliente la funcionalidad lista.

Cuadro 6.23: Iteración 16

Tarea	F Inicial - F Final	RC
Guardar plan de clases 2	01/09/2018 - 07/09/2018	RS09
<i>SubTareas</i>		
Investigar formas de generar y guardar archivos		
Pruebas con librerías PDF		
Crear controlador guardar plan de clases		
Enlazar vista-controlador-repositorio guardar plan de clases		

La siguiente iteración detallada en el Cuadro 6.24, se muestra al cliente el comienzo de la vista de autenticación e ideas de cómo se recuperará la contraseña en el prototipo. Se opta por utilizar correo electrónico para ello. El cliente sugiere como trabajo futuro clonar las competencias del periodo anterior, decidiendo al final que las competencias son dependientes del curso o módulo, no del periodo.

Cuadro 6.24: Iteración 17

Tarea	F Inicial - F Final	RC
Autenticación	08/09/2018 - 14/09/2018	-
<i>SubTareas</i>		
Crear vista		

Cuadro 6.25: Iteración 17

Tarea	F Inicial - F Final	RC
Autenticación	08/09/2018 - 14/09/2018	-
<i>SubTareas</i>		
Crear vista		
Crear backend autenticación		
Crear repositorio autenticación		
Enlazar vista-controlador-repositorio autenticación		
Crear modal recuperar contraseña		
Crear repositorio recuperar contraseña		
Enlazar vista-controlador-repositorio recuperar contraseña		

Al estar en semana de receso por fiestas patrias, se optó por elegir una actividad simple y ligera, por lo que las tareas pendientes de la iteración anterior se dejaron para una subsiguiente. En esta iteración ilustrada en el Cuadro 6.26, se ofrece al cliente una vista de cómo se verá el inicio del prototipo y el perfil del usuario ya autenticado.

Cuadro 6.26: Iteración 18

Tarea	F Inicial - F Final	RC
Perfil, Inicio	15/09/2018 - 21/09/2018	-
Crear vista		
Crear backend perfil		
Crear controlador perfil		
Crear repositorio perfil		
Enlazar vista-controlador-repositorio perfil		
Crear vista		
Crear backend inicio		
Crear controlador inicio		
Crear repositorio inicio		
Enlazar vista-controlador-repositorio inicio		

En la Iteración 19 descrita en el Cuadro 6.27 se abordan las tareas pendientes de la iteración 17. Acá el cliente experimenta con la autenticación completada y la recuperación de la contraseña.

Cuadro 6.27: Iteración 19

Tarea	F Inicial - F Final	RC
Autenticación 2	10/09/2018 - 27/10/2018	-
SubTareas		
Crear controlador autenticación		
Crear repositorio recuperar contraseña		
Crear código para enviar correo electrónico		

En la siguiente iteración detallada en el Cuadro 6.28, se muestra al cliente la interfaz de las próximas funcionalidades a desarrollar, como los enlaces competencia-aprendizaje y curso-competencia, entre otros, además de mejorar la vista de recuperar contraseña.

Cuadro 6.28: Iteración 20

Tarea	F Inicial - F Final	RC
Interfaz 2	29/09/2018 - 05/10/2018	-
Diseñar bosquejo vista competencia-aprendizaje		
Crear vista competencia-aprendizaje html-razor		
Diseñar bosquejo vista aprendizaje-saber		
Crear vista aprendizaje-saber html-razor		
Diseñar bosquejo vista curso-competencia		
Crear vista curso-competencia html-razor		
Diseñar bosquejo vista unidad-saber		
Crear vista unidad-saber html-razor		
Diseñar bosquejo vista actividad-saber		
Crear vista actividad-saber html-razor		
Diseñar bosquejo vista modificar cuenta		
Crear vista modificar cuenta html-razor		
Diseñar bosquejo vista recuperar contraseña		
Crear vista recuperar contraseña html-razor		

En la Iteración 21 descrita en el Cuadro ?? se ofrece al cliente las funcionalidades asociadas a la asociación de competencias y aprendizajes existentes en la vista aprendizaje.

Cuadro 6.29: Iteración 21

Tarea	F Inicial - F Final	RC
Enlazar Competencias-Aprendizajes	06/10/2018 - 12/10/2018	RS48, RS49
SubTareas		
Corrección deshabilitar competencia		
Crear modal competencias-aprendizajes		
Crear backend competencias-aprendizajes		
Crear repositorio competencias-aprendizajes		
Enlazar modal-controlador-repositorio competencias-aprendizajes		

En la Iteración 22 descrita en el Cuadro 6.30 se ofrece al cliente las funcionalidades asociadas a la asociación de aprendizajes y saberes existentes en la vista saber.

Cuadro 6.30: Iteración 22

Tarea	F Inicial - F Final	RC
Enlazar Aprendizaje-Saber	13/10/2018 - 19/10/2018	RS55, RS56
SubTareas		
Crear modal competencias-aprendizajes		
Crear backend competencias-aprendizajes		
Crear repositorio competencias-aprendizajes		
Enlazar modal-controlador-repositorio competencias-aprendizajes		

En la Iteración 23 descrita en el Cuadro 6.31 se ofrece al cliente las funcionalidades asociadas a la asociación de competencias y cursos o módulos existentes en la vista curso.

Cuadro 6.31: Iteración 23

Tarea	F Inicial - F Final	RC
Enlazar Competencia-Curso	20/10/2018 - 26/10/2018	-
SubTareas		
Crear modal competencia-curso		
Crear backend competencia-curso		
Crear repositorio competencia-curso		
Enlazar modal-controlador-repositorio competencia-curso		

En la Iteración 24 descrita en el Cuadro 6.32 se ofrece al cliente las funcionalidades asociadas a la asociación de saberes y unidades existentes en la vista unidad.

Cuadro 6.32: Iteración 24

Tarea	F Inicial - F Final	RC
Enlazar Saber-Unidad	27/10/2018 - 02/11/2018	
SubTareas		
Crear modal saber-unidad		
Crear backend saber-unidad		
Crear repositorio saber-unidad		
Enlazar modal-controlador-repositorio saber-unidad		

En la Iteración 25 descrita en el Cuadro 6.33 se ofrece al cliente las funcionalidades asociadas a la asociación de saberes y actividades existentes en la vista actividad.

Cuadro 6.33: Iteración 25

Tarea	F Inicial - F Final	RC
Enlazar Saber-Actividad	03/11/2018 - 09/11/2018	-
SubTareas		
Crear modal saber-actividad		
Crear backend saber-actividad		
Crear repositorio saber-actividad		
Enlazar modal-controlador-repositorio saber-actividad		

La siguiente iteración detallada en el Cuadro 6.34 se agregó una funcionalidad a la vista Perfil, que permite al cliente ver los cursos asociados a él en el periodo actual, y el total de los cursos inscritos. Además, se le ofreció una nueva vista de configuración general, donde el cliente puede configurar la cantidad de periodos y el periodo actual en el que se encuentra.

Cuadro 6.34: Iteración 26

Tarea	F Inicial - F Final	RC
Perfil y configuraciones restantes	10/11/2018 - 16/11/2018	
SubTareas		
Agregar cursos actuales de un usuario a perfil		
Agregar cursos totales de un usuario a perfil		
Crear vista configuraciones generales		
Crear backend configuraciones generales		
Crear controlador configuraciones generales		
Crear repositorio configuraciones generales		
Enlazar vista-controlador-repositorio configuraciones generales		

Como se puede observar en la Iteración 27 descrita en el Cuadro 6.35, se ofrece al cliente las correcciones de errores y de funcionalidades, como la duplicación de saberes cuando se es asociada una competencia a un curso o el que cualquier usuario registrado pueda ver los saberes de una actividad.

Cuadro 6.35: Iteración 27

Tarea	F Inicial - F Final	RC
Correcciones	17/11/2018 - 23/11/2018	-
SubTareas		
Corregir deshabilitar competencia		
Duplicar saber de una competencia		
Duplicar competencia cuando se es modificada		
Mostrar saberes en actividades		
Corrección de vista unidad cuando un curso es deshabilitado		

En la Iteración 28 detallada en el Cuadro 6.36, se da inicio posterior al comienzo del primer semestre del año 2019. Se informa al cliente del comienzo de las pruebas de caja negra y se le muestran las tablas creadas para ese propósito.

Cuadro 6.36: Iteración 28

Tarea	F Inicial - F Final	RC
Testing 1	23/03/2019 - 29/03/2019	-
SubTareas		
Preparación de pruebas de caja negra competencias		
Ejecución de pruebas de caja negra competencias		
Preparación de pruebas de caja negra aprendizajes		
Ejecución de pruebas de caja negra aprendizajes		
Preparación de pruebas de caja negra saberes		
Ejecución de pruebas de caja negra saberes		
Preparación de pruebas de caja negra cursos		
Ejecución de pruebas de caja negra cursos		
Preparación de pruebas de caja negra cursos por periodo		
Ejecución de pruebas de caja negra cursos por periodo		
Preparación de pruebas de caja negra unidades		
Ejecución de pruebas de caja negra unidades		

En la Iteración 29 detallada en el Cuadro 6.37, se muestra al cliente las correcciones resultantes del testeo anterior, como modificar competencias y saberes.

Cuadro 6.37: Iteración 29

Tarea	F Inicial - F Final	RC
Correcciones 2	29/03/2019 - 04/04/2019	-
SubTareas		
Corrección deshabilitar competencia		
Corrección modificar saber		
Corrección deshabilitar curso por periodo		

Como se puede observar en la Iteración 30 descrita en el Cuadro 6.38, se muestra al cliente el inicio del segundo testeo de caja negra. El cliente insiste en probar la aplicación y detecta que había problemas con algunas operaciones. Tras algunas revisiones se detecta que era un problema del servidor y quedó de corregirse para la próxima iteración.

Cuadro 6.38: Iteración 30

Tarea	F Inicial - F Final	RC
Testing 2	06/04/2019 - 13/04/2019	-
SubTareas		
Preparación de pruebas de caja negra actividades		
Ejecución de pruebas de caja negra actividades		
Preparación de pruebas de caja negra plan de clases		
Ejecución de pruebas de caja negra plan de clases		
Preparación de pruebas de caja negra autenticación		
Ejecución de pruebas de caja negra autenticación		
Ejecución de pruebas de caja negra envío de correo		
Preparación de pruebas de caja negra perfil		
Ejecución de pruebas de caja negra perfil		
Preparación de pruebas de caja negra enlace competencia-aprendizaje		
Ejecución de pruebas de caja negra enlace competencia-aprendizaje		

En la Iteración 31 detallada en el Cuadro ??, se muestra al cliente las correcciones resultantes del testeo anterior, como deshabilitar actividades y ver todos los cursos.

Cuadro 6.39: Iteración 31

Tarea	F Inicial - F Final	RC
Correcciones 2	14/04/2019 - 20/04/2019	-
<i>SubTareas</i>		
Corrección deshabilitar actividades		
Corrección formato envío de correo electrónico		
Corrección ver todos los cursos asociados en el perfil de un usuario		

En la Iteración 32 detallada en el Cuadro 6.40, se muestra al cliente los resultados del último testeo de la aplicación.

Cuadro 6.40: Iteración 32

Tarea	F Inicial - F Final	RC
Testing final	21/04/2019 - 27/04/2019	-
<i>SubTareas</i>		
Preparación de pruebas de caja negra enlace aprendizaje-saber		
Ejecución de pruebas de caja negra enlace aprendizaje-saber		
Preparación de pruebas de caja negra enlace competencia-curso		
Ejecución de pruebas de caja negra enlace competencia-curso		
Preparación de pruebas de caja negra enlace saber-unidad		
Ejecución de pruebas de caja negra enlace saber-unidad		
Preparación de pruebas de caja negra enlace saber-actividad		
Ejecución de pruebas de caja negra enlace saber-actividad		
Preparación de pruebas de caja negra configuración general		
Ejecución de pruebas de caja negra configuración general		
Preparación de test de usabilidad con el cliente		
Ejecución de test de usabilidad con el cliente		

Finalmente, en la Iteración 33 descrita en el Cuadro 6.41 se ofrece al cliente las correcciones finales en base a las pruebas de usabilidad ejecutadas anteriormente, se corrige el test de usabilidad y comienza la ejecución de pruebas con usuarios.

Cuadro 6.41: Iteración 33

Tarea	F Inicial - F Final	RC
Correcciones Finales	28/04/2019 - 04/05/2019	-
SubTareas		
Correcciones finales		
Corrección montaje aplicación en servidor		
Preparación de test de usabilidad		
Corrección test de usabilidad		
Ejecución de test de usabilidad con el cliente		

6.2. Funcionamiento del Prototipo

Para poder lograr una comprensión a grandes rasgos sobre cómo funciona el prototipo, la Figura 6.1 detalla un esquema simple de cómo éste se encuentra conformado. Por simpleza, se omite la funcionalidad de Iniciar Sesión a sabiendas que éste es vital para que cualquier usuario tenga acceso a cualquier sector del prototipo.

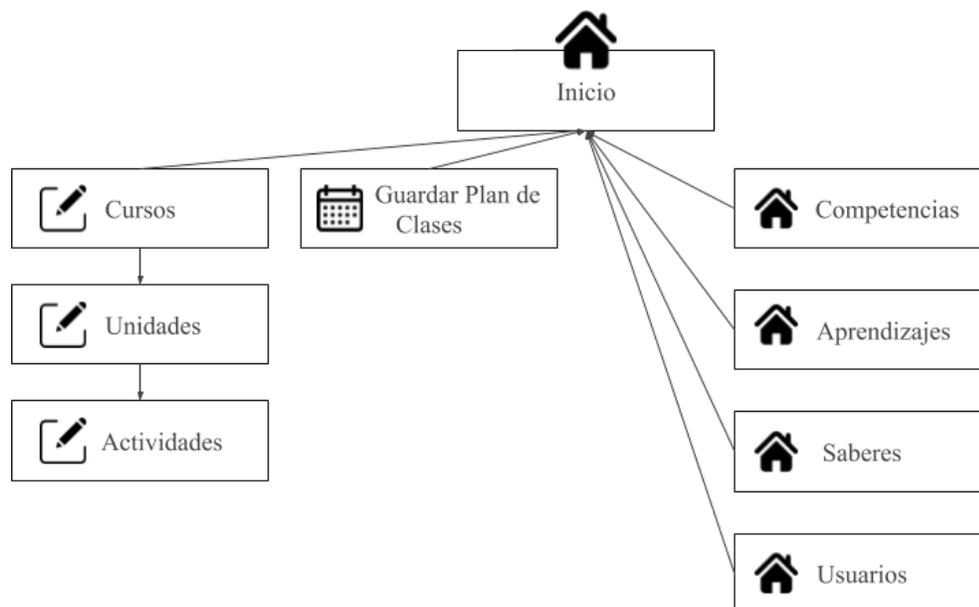


Figura 6.1: Esquema general del funcionamiento del prototipo

- Inicio:** La Figura 6.2 ilustra la vista inicio, que es la primera vista a la que el usuario accede luego de iniciar sesión. Muestra, de ser administrador, la cantidad en número y en gráfico de datos ingresados en el prototipo.

- **Competencias:** La Figura 6.3 ilustra la vista competencias que lista, crea, modifica, deshabilita y habilita competencias. En la imagen se visualiza una tabla cuyas columnas corresponden a nombre, descripción y detalle, además de darle la opción al usuario administrador de crear, modificar y deshabilitar/habilitar. No se encuentra ligada con ninguna otra funcionalidad, a pesar de que es necesaria para poder crear un plan de clases.
- **Aprendizajes:** La Figura 6.4 ilustra la vista aprendizajes que lista, crea, modifica, deshabilita y habilita aprendizajes. Aquí es posible el asociar el aprendizaje a alguna competencia. En la imagen se visualiza una tabla cuyas columnas corresponden a categoría, subcategoría, nombre, competencia asociada, además de darle la opción al usuario administrador de crear, modificar y deshabilitar/habilitar.
- **Saberes:** La Figura 6.5 ilustra la vista saberes que lista, crea, modifica, deshabilita y habilita saberes. Aquí es posible el asociar el saber a algún aprendizaje. En la imagen se visualiza una tabla cuyas columnas corresponden a aprendizaje asociado, saber, saber hacer y saber ser, además de darle la opción al usuario administrador de crear, modificar y deshabilitar/habilitar.
- **Usuarios:** La Figura 6.6 ilustra la vista usuarios que lista, crea, modifica, reinicia la contraseña, deshabilita y habilita usuarios. Es la única forma de poder crear un usuario en el prototipo. En la imagen se puede observar una tabla cuyas columnas corresponden a RUT, nombre, correo electrónico, tipo y matrícula, además de darle la opción al usuario administrador de crear, modificar y deshabilitar/habilitar.
- **Cursos:** La Figura 6.7 ilustra la vista cursos que lista, crea, modifica, deshabilita y habilita cursos y los periodos de un curso. Es una de las funcionalidades centrales, puesto que sin esta resulta imposible poder imprimir un plan de clases. También permite asociarle competencias y gestionar participantes (profesores y alumnos). En la imagen se visualiza una tabla cuyas columnas corresponden a nombre, descripción, horas presenciales, horas autónomas, año y periodo, además de darle la opción al usuario administrador de crear, modificar y deshabilitar/habilitar un curso y el periodo de un curso.

- **Unidades:** La Figura 6.8 ilustra la vista unidades que lista, crea, modifica, deshabilita y habilita unidades. Depende de los periodos de un curso para ser accedido. También permite asociarle saberes de las competencias asociadas a un curso previamente. En la imagen se visualiza una tabla cuyas columnas corresponden nombre y descripción, además de darle la opción al usuario administrador de crear, modificar y deshabilitar/habilitar. También se puede observar los datos del curso correspondiente.
- **Actividades:** Vista que lista, crea, modifica, deshabilita y habilita actividades. Depende de una unidad previamente creada para ser accedido. También permite asociarle saberes de las unidades asociadas previamente. Las Figuras 6.9 y 6.10 ilustran dos tipos de vista para las actividades:
 - En tabla, cuyas columnas corresponden a nombre, descripción, fecha de inicio, fecha de fin, tipo de actividad, tiempo presencial de la actividad y tiempo no presencial de la actividad.
 - En calendario, donde el usuario puede hacer clic y arrastrar el puntero sobre las fechas para crear una actividad de forma más dinámica.
- **Guardar plan de clases:** La Figura 6.11 ilustra la vista que permite al usuario guardar un plan de clases. Depende de todas las otras funcionalidades, especialmente de cursos, dado que si no existen en el prototipo, el prototipo no permite ejecutar esta funcionalidad.
- Finalmente, en la Figura 6.12 se puede ver el formato básico de la creación de un plan de clases, en este caso se utiliza el módulo de Modelos Discretos como ejemplo. Dada cada una de las unidades, se les asignó un rango de fechas, una actividad y los saberes asociados a estos, donde las actividades fueron ordenadas según la fecha de creación de la actividad.

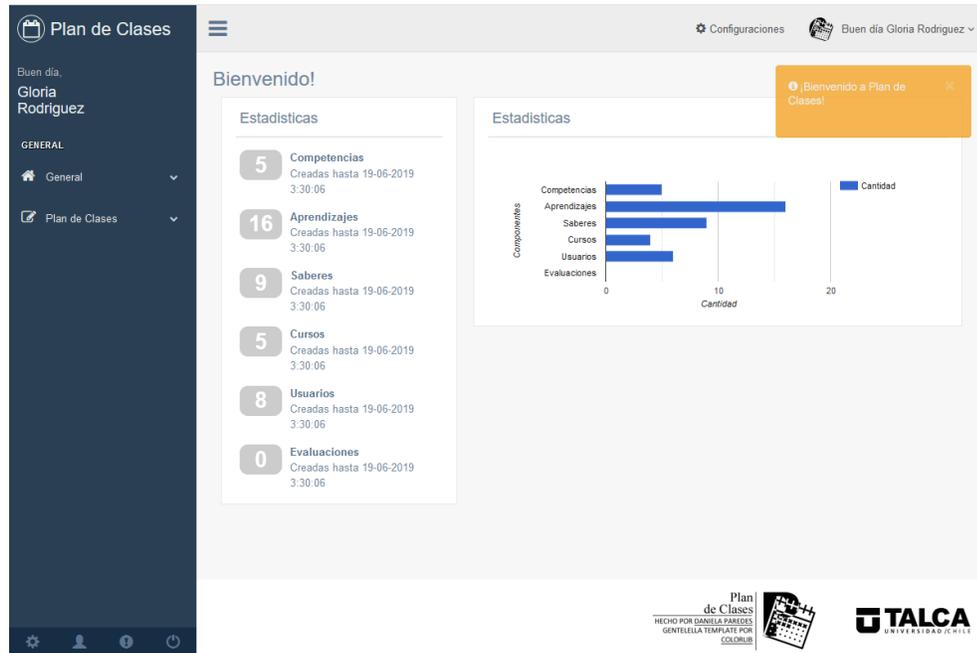


Figura 6.2: Vista inicio

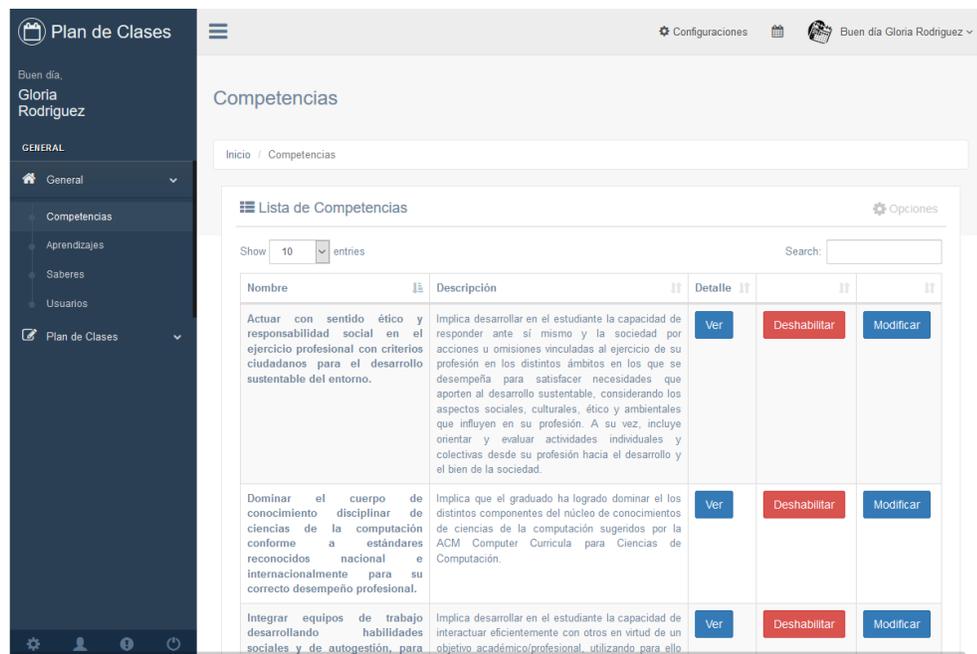


Figura 6.3: Vista competencias

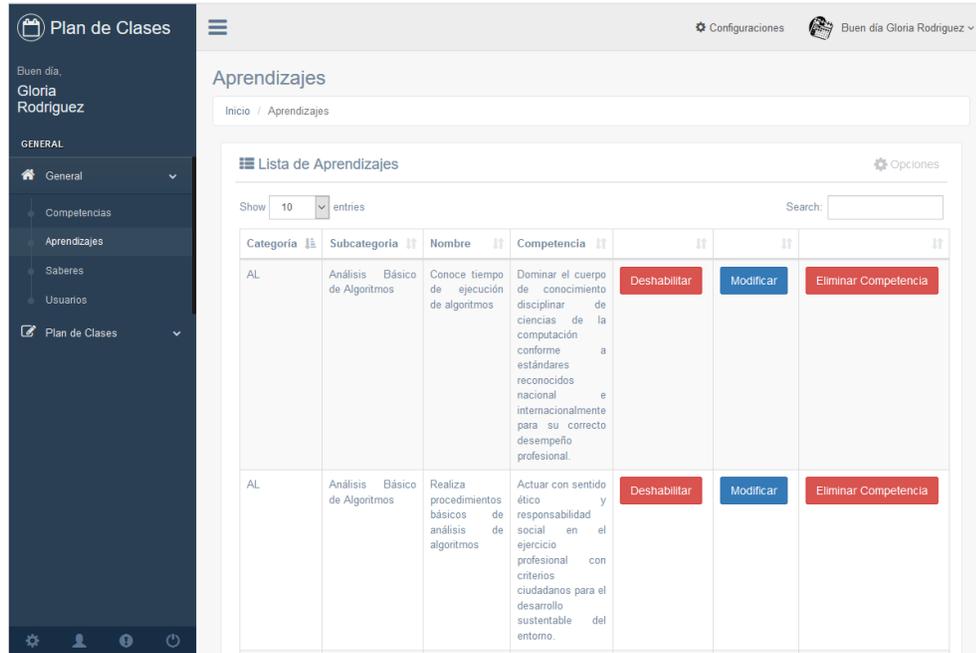


Figura 6.4: Vista aprendizajes

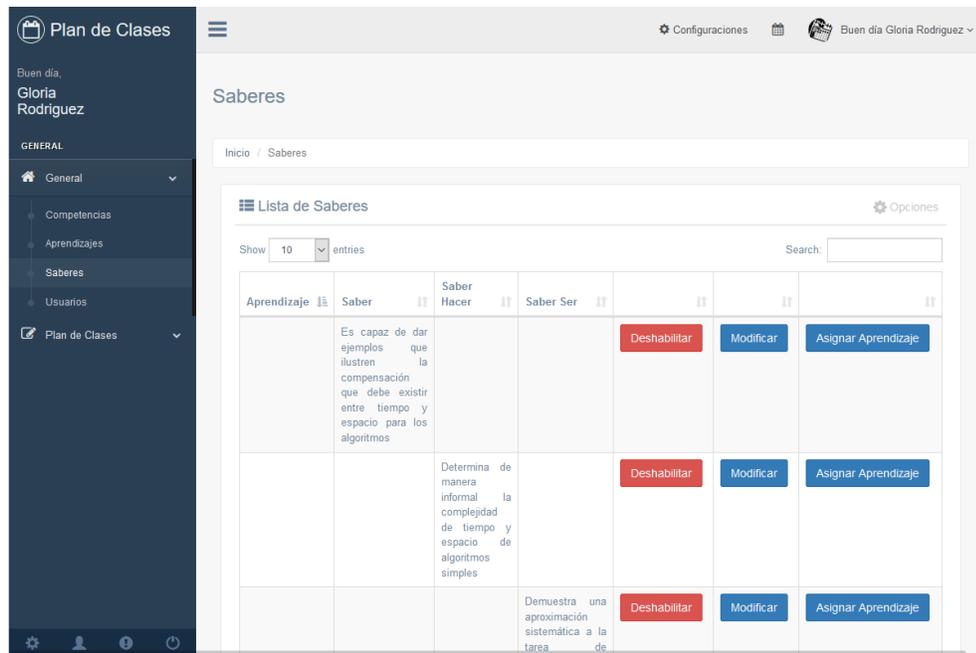


Figura 6.5: Vista saberes

Usuarios Registrados

Inicio / Usuarios

Lista de Usuarios Opciones

Show 10 entries Search:

	Rut	Nombre	Correo Electrónico	Tipo	Matricula
Opciones	13 288 528-6	Rodrigo Paredes	rparedes@ucv.cl	Profesor	No tiene
Opciones	17 214 876-3	Nicolás Rojas	nrojas14@alumnos.ucv.cl	Alumno	2014407535
Opciones	17 225 863-7	Daniela Paredes	dparedes12@alumnos.ucv.cl	Alumno	2009407826
Opciones	17 224 523-6	Yann Sepúlveda	yannsepulveda17@alumnos.ucv.cl	Profesor	No tiene
Opciones	18 343 138-98	Cristian Sepúlveda	csul191@ucv.cl	Alumno	2019407064
Opciones	18 254 373-8	Viviana Rojas	vrojas14@alumnos.ucv.cl	Profesor	No tiene
Opciones	9 284 126-2	Gloria Rodríguez	gloria-r@ucv.cl	Administrador	No tiene
Opciones	9 288 386-3	Administrador	admin1717@ucv.cl	Administrador	No tiene

Figura 6.6: Vista usuarios

Cursos

Inicio / Cursos

Lista de Cursos Opciones

Show 10 entries Search:

	Nombre	Descripción	Horas Presenciales	Horas Autónomas	Año	Período
Opciones	Programación Avanzada	El curso de programación avanzada algo hace	6	3	2018	1
Opciones	Programación Avanzada	El curso de programación avanzada algo hace	6	3	2017	1
Opciones	Programación Avanzada	El curso de programación avanzada algo hace	6	3	2017	2
Opciones	Algoritmos	El curso de algoritmos algo hace	6	3	2018	1
Opciones	Modelos Discretos	Formar Ingenieros	6	3	2018	1
Opciones	Modelos Discretos	Formar Ingenieros	6	3	2019	1
Opciones	Construcción de Software	Este es el curso de Construcción de Software	1	9	2019	1
Opciones	Taller de problemas de Ingeniería	Ya no hace nada.	1	7	2018	2

Figura 6.7: Vista cursos

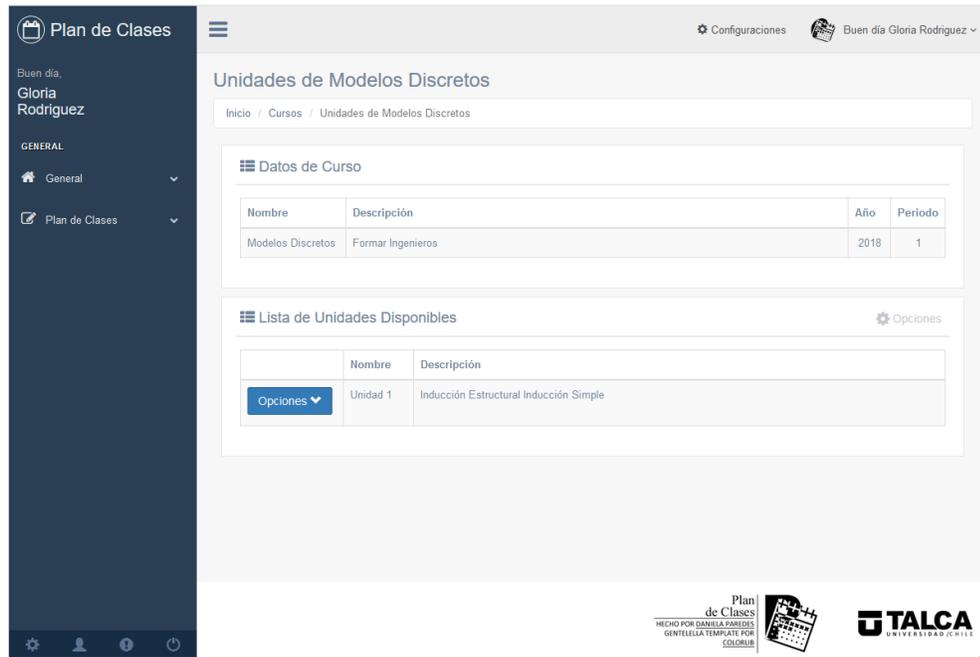


Figura 6.8: Vista unidades

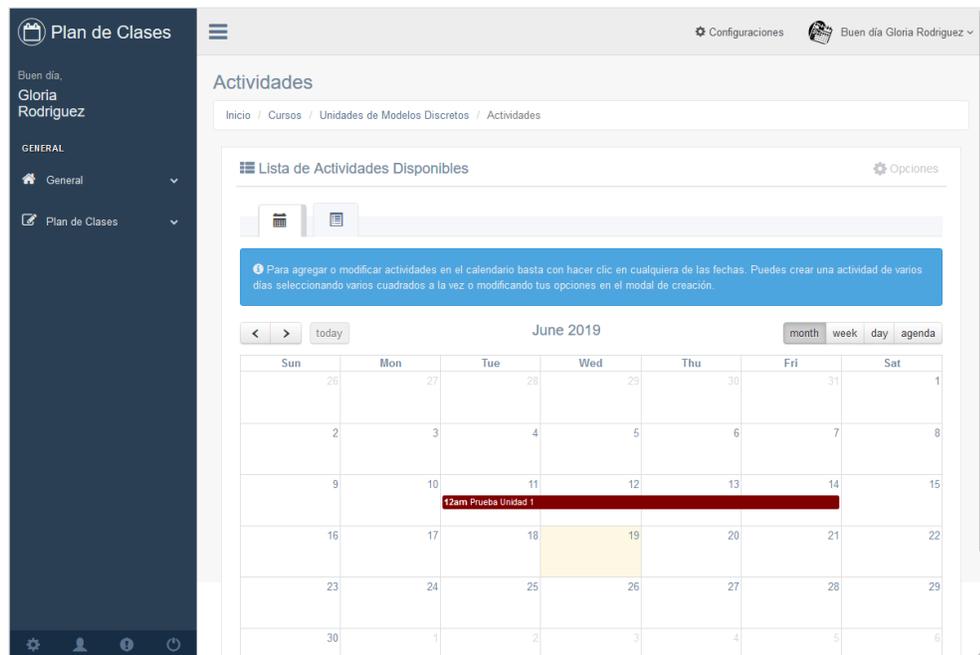


Figura 6.9: Vista actividades, versión calendario

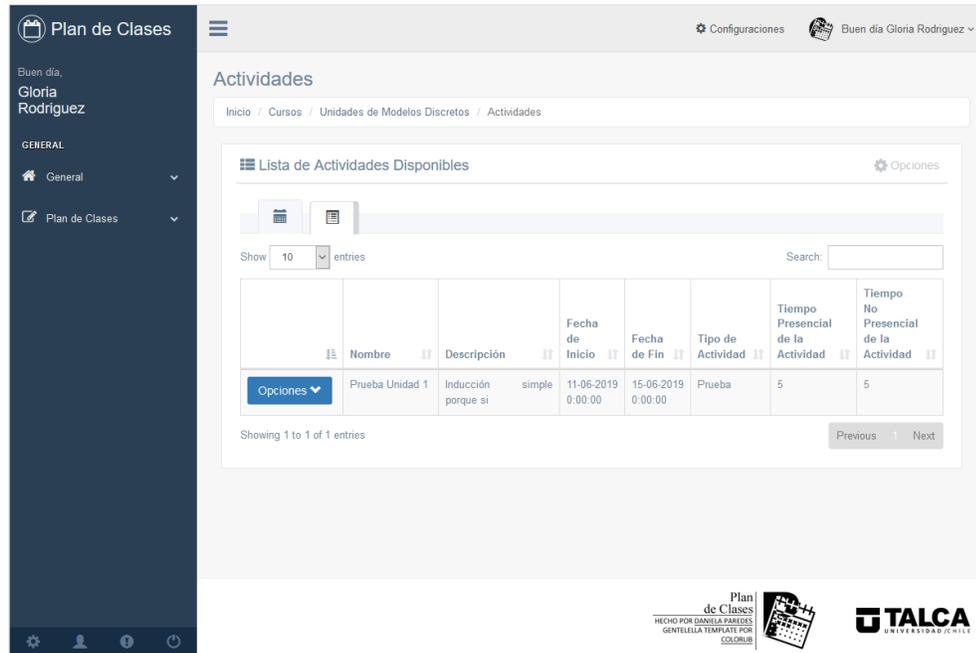


Figura 6.10: Vista actividades

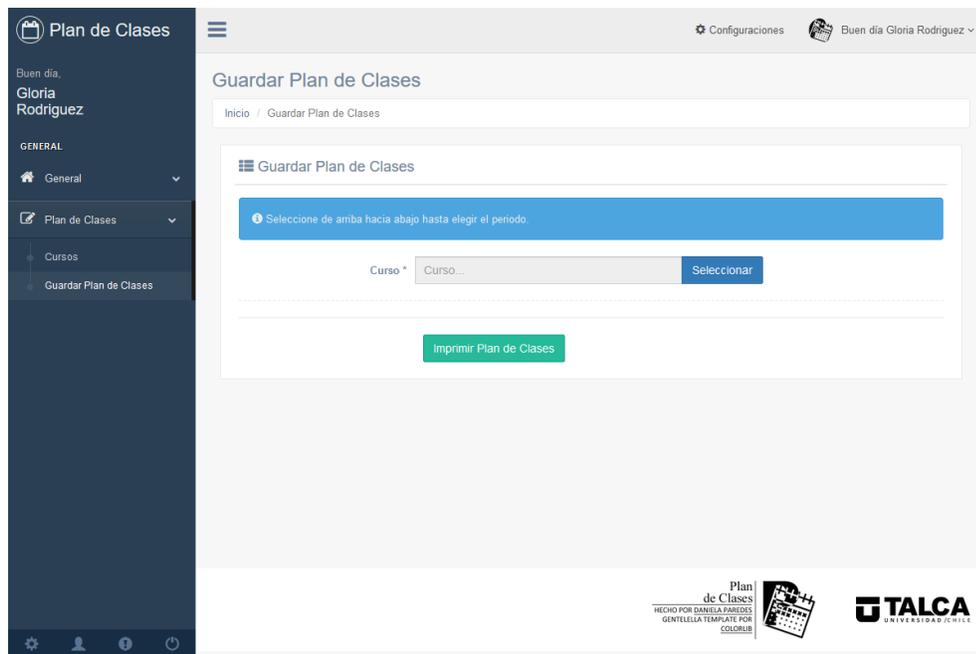


Figura 6.11: Vista guardar plan de clases



Plan de Clases

Modelos discretos

Rodrigo Paredes

Unidad 1		
Fechas	Actividades	Saberes
04-06-2018 - 06-06-2018	Tarea Unidad 1	AL - 2.22: Demuestra una aproximación sistemática a la tarea de identificación de peligros y riesgos en una situación particular. AL - 1.5: Lista y contrasta clases de complejidad estándar
11-06-2018 - 15-06-2018	Prueba Unidad uno	AL - 2.22: Demuestra una aproximación sistemática a la tarea de identificación de peligros y riesgos en una situación particular. AL - 1.1: Conoce la diferencia entre el peor, promedio y mejor tiempo de ejecución de un algoritmo
Unidad 2		
Fechas	Actividades	Saberes
05-08-2018 - 12-08-2018	Introducción a recurrencias	AL - 1.5: Lista y contrasta clases de complejidad estándar
31-08-2018 - 31-08-2018	Prueba Unidad 2	AR - 2.2: Demuestra una aproximación sistemática a la tarea de identificación de peligros y riesgos en una situación particular. AL - 1.1: Conoce la diferencia entre el peor, promedio y mejor tiempo de ejecución de un algoritmo

Figura 6.12: Ejemplo plan de clases

6.3. Resumen

En este capítulo se especifican los procesos de desarrollo e implementación del prototipo. Debido a que la metodología de desarrollo escogida es la metodología tradicional iterativa, se divide en 33 iteraciones. Además se describe el funcionamiento general del prototipo.

7. Pruebas y Análisis de Resultados

En este capítulo se presenta un resumen de las pruebas más relevantes. Se inicia con las pruebas de caja negra y de usabilidad, para terminar con los análisis de resultados.

Las pruebas de caja negra fueron realizadas en las últimas iteraciones con la finalidad de verificar que el sistema cumple con los requisitos que el cliente solicitó. Las pruebas de usabilidad fueron realizadas con usuarios analistas QA¹ y usuarios sin especialización en aseguramiento de calidad.

Para poder realizar las pruebas de usabilidad, se dispuso de un servidor cuyas especificaciones son detalladas en la Sección 2.2, cuya url² fue entregada a los usuarios destinados a realizar las pruebas.

7.1. Pruebas de Caja Negra

Las pruebas de caja negra se presentan en cuadros, donde la primera fila representa al código de la prueba, cuya notación es **CNXX** (Donde CN corresponde a ‘caja negra’ y XX es el número identificador del requerimiento). La segunda fila corresponde a los requisitos de sistema implicados, cuya nomenclatura se encuentran detalladas en la Sección 4.3. La tercera fila corresponde a precondiciones de la prueba, de existir alguna. La cuarta y quinta fila fueron divididas en tres partes; donde la primera columna corresponde a las entradas dadas por el usuario, la segunda columna corresponde a la salida esperada o respuesta del software y la tercera columna es

¹Quality assurance

²Uniform Resource Locator

la salida obtenida, que corresponde a la salida que el software dio en el momento de la prueba. Finalmente, la última fila corresponde a las observaciones que pudieran hacerse dado el caso.

Las pruebas más importantes y que se detallarán a continuación son las de guardar un plan de clases e iniciar sesión. En el Anexo D, Secciones D.1 y D.2 se presentan mayores detalles sobre las Pruebas de Caja negra, además de diagramas e imágenes que facilitan entender el resultado de cada prueba de caja negra mediante capturas de pantalla sobre el prototipo.

La prueba CN01 consiste en iniciar una sesión de trabajo en el prototipo. Para esto, el usuario debe indicar su RUT sin dígito verificador y contraseña. El detalle de esta prueba se muestra en el Cuadro D.1.

Cuadro 7.1: Caja Negra - Iniciar sesión

CN01: Iniciar sesión		
Requisitos de sistema	RS01	
Precondiciones	-	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Usuario ▪ Contraseña 	<ul style="list-style-type: none"> ▪ Sesión registrada en la aplicación. ▪ Mensaje de éxito. ▪ Excepción: Notificación de rechazo: “No pudo iniciar sesión” 	<ul style="list-style-type: none"> ▪ Sesión registrada con éxito. ▪ Mensaje de éxito.
Observaciones	-	

La prueba CN40 consiste en guardar un plan de clases. Para lograr esto, el usuario debe seleccionar un curso: si es un profesor, sólo podrá guardar un plan de clases de un curso al que haya sido inscrito y un periodo. El detalle de esta prueba se muestra en el cuadro D.40.

Cuadro 7.2: Caja Negra - Guardar plan de clases

CN40: Guardar plan de clases		
Requisitos de sistema	RS09	
Precondiciones	CN01	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del curso por periodo 	<ul style="list-style-type: none"> ▪ Documento generado para el usuario ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Documento generado exitosamente ▪ Mensaje de éxito
Observaciones	-	

7.2. Pruebas de Usabilidad

Para poder aplicar esta prueba, a cada uno de los usuarios se le entrega una url a la aplicación montado en un servidor de amazon (cuyos requerimientos mínimos fueron descritos en la Sección 2.2). Las pruebas de usabilidad fueron divididas en tres partes, a saber:

- El **Cuestionario de Pre-evaluación** tiene como objetivo el poder conocer el nivel de dominio del usuario y las expectativas que éste tiene con respecto al prototipo a evaluar.
- El **Cuestionario de evaluación** es la evaluación del prototipo propiamente tal, dividido en una serie de tareas donde el usuario debe interactuar con ciertas funcionalidades del sistema y dejar tanto sus impresiones, como el tiempo promedio que demora en realizarlo. También existe una instancia donde puede dar una recomendación sobre la funcionalidad abordada. Las tareas a ejecutar fueron las siguientes:
 - Ver los Cursos
 - Cambiar su contraseña
 - Crear una competencia, un aprendizaje y dos saberes

- Agregar un aprendizaje a una competencia y los saberes a un aprendizaje
 - Crear un curso, una unidad y algunas actividades
 - Asociar una competencia a un curso
 - Asociar el saber de un curso a una unidad y a la o las actividades
 - Configurar periodo y año
 - Crear un plan de clases
- Finalmente, el **Cuestionario Post-Evaluación** mide las impresiones finales y generales del usuario con respecto al prototipo. En esta instancia se espera que además de las tareas solicitadas en el cuestionario anterior, el usuario explore a gusto la aplicación.

Las imágenes del test de usabilidad se pueden encontrar en el Anexo D, Figuras D.41 a D.54 y en el siguiente enlace: <https://forms.gle/VZ6c3agSELLfYy859>.

7.3. Resultados

El test de usabilidad fue realizado por once personas, cuyos resultados se encuentran en el Anexo D, Figuras D.55 a D.65. Donde se puede destacar que:

Resultados pre-evaluación

La Figura 7.1 detalla el nivel de dominio en el uso de un computador o un tablet del usuario que se encarga de realizar el test. Se puede ver en el gráfico que el 36,4% (cuatro personas) de los usuarios se encuentra en un nivel promedio, el mismo porcentaje (cuatro personas) se encuentra en un nivel cercano a un dominio alto y el 27,3% (tres personas) considera que domina completamente el uso de un computador o un tablet.

Acerca de la pregunta **¿Cuáles son sus expectativas con respecto a la aplicación?**, según la Figura 7.2 se puede destacar que del 100% de los usuarios, el 36,4% espera que la interfaz y la funcionalidad no debería ser necesariamente similares a otras aplicaciones (de preferencia de la Universidad de Talca). Del 100% de los usuarios, el 36,4% posee una opinión neutra sobre la expectativa de una interfaz atractiva, sin embargo el 27,2% de los usuarios se encuentra más cerca

Dominio en el uso de un computador

1 = Muy bajo y 5 = Muy alto

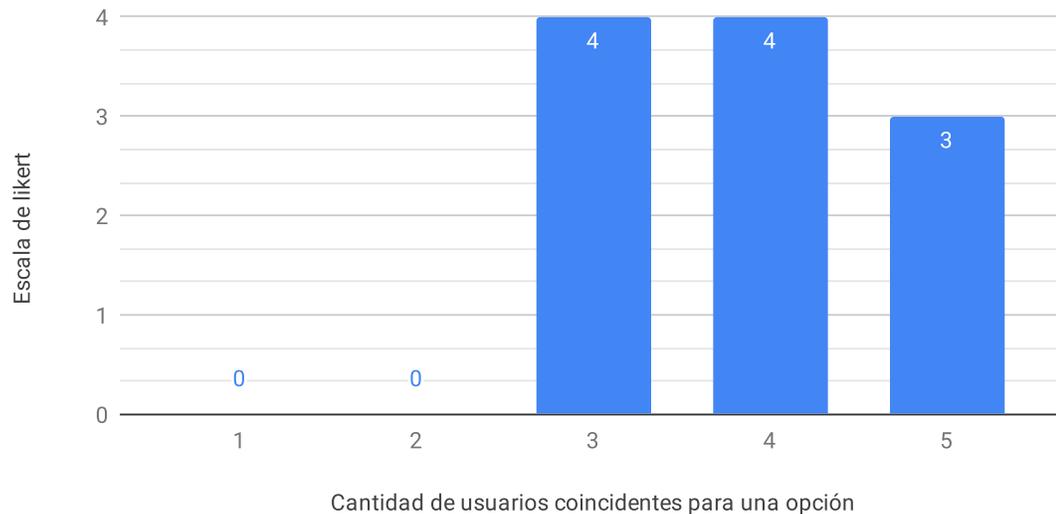


Figura 7.1: Gráfico de dominio en el uso de un computador

de tener una alta expectativa sobre esto. El 36,4% de los usuarios posee una alta expectativa sobre el buen rendimiento del prototipo, seguido por el 27,2% que se encuentran cercanos a esta opinión. Finalmente, el 72,7% de los usuarios espera que el prototipo posea las funcionalidades necesarias para poder crear un plan de clases, seguido del 27,2% que se encuentra cercano a esta opinión.

Resultados relevantes de la evaluación

Sobre las pruebas de likert realizadas en el test de evaluación para cada una de las tareas, se puede decir que:

- Para la tarea **ver los cursos** ilustrado en la Figura D.66, se puede ver que de once usuarios, el 72,7% opina que la funcionalidad de ver cursos fue muy fácil y el 54,5% opinó que es sencillo. Esta funcionalidad obtuvo evaluaciones positivas y ningún comentario.
- Para la tarea **cambiar su contraseña** ilustrada en la Figura D.67, se puede ver que el 90,9% de los usuarios opina que la funcionalidad fue fácil. Del 100%

Expectativas con respecto a la aplicación

1 = Muy bajas y 5 = Muy altas

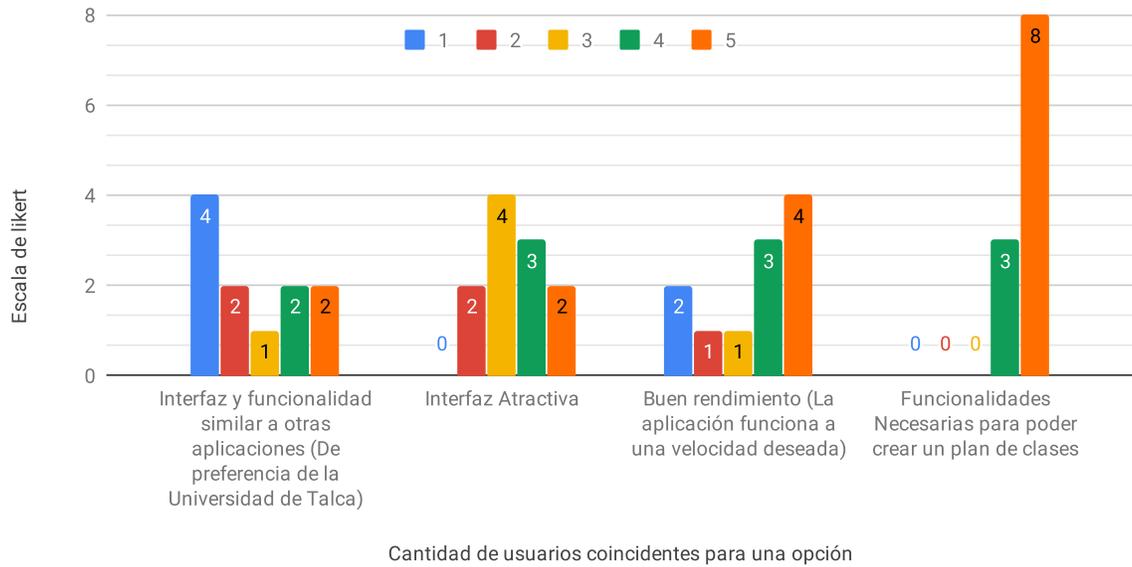


Figura 7.2: Gráfico de expectativas del usuario con respecto a la aplicación

de los usuarios, el 72,7 % opinó que la funcionalidad es sencilla de usar. En los comentarios, hubo discrepancias sobre la utilidad de los botones en la parte inferior del navegador.

- Para la tarea **crear una competencia, un aprendizaje y dos saberes** ilustrada en la Figura D.68, se puede ver que el 54,5 % de los usuarios opina que la funcionalidad fue fácil, recibiendo opiniones positivas en general. Del 100 % de los usuarios, el 54,5 % opinó que la funcionalidad fue sencilla de utilizar. Un usuario opinó que establecer el enlace al momento de la creación de la competencia, se disminuiría la confusión en los usuarios.
- Para la tarea **agregar un aprendizaje a una competencia y los saberes a un aprendizaje** ilustrada en La Figura D.69, se puede ver que el 36,4 % de los usuarios opina que la funcionalidad no fue fácil, sin embargo el 27,2 % opina justo lo contrario. Del 100 % de los usuarios, el 45,4 % opinó que la funcionalidad fue no sencilla de usar, recibiendo comentarios negativos en su mayoría. Los comentarios que hubieron, fueron nuevamente recomendaciones

sobre el enlace entre competencia, aprendizaje y saber. Hubo cuatro usuarios que opinaron algo similar al gráfico anterior.

- Para la tarea **crear un curso, una unidad y algunas actividades** ilustrada en la Figura D.70, se puede observar que en general el 81,8 % de los usuarios mantiene una opinión positiva sobre la facilidad de uso de la funcionalidad y hubieron dos usuarios, quienes mantuvieron una opinión neutral. Sin embargo, se puede destacar que el 45,4 % de los usuarios mantuvo una opinión neutral sobre la sencillez de la funcionalidad y el 27,2 % opinó que la funcionalidad no era sencilla. Dentro de los comentarios de los usuarios, hubieron dos que sostuvieron lo confuso que era el crear el periodo de un curso y uno que destacó el uso del calendario para crear una actividad.
- Para la tarea **asociar una competencia a un curso** ilustrada en la Figura D.71, se puede observar que en general el 100 % de los usuarios mantiene una opinión positiva sobre la facilidad de uso de la funcionalidad. Sin embargo, destaca que en general, el 81,8 % de los usuarios tiene opiniones positivas sobre la sencillez de la funcionalidad y el resto mantuvo una opinión neutra sobre ello.
- Para la tarea **asociar el saber de un curso a una unidad y a la o las actividades** ilustrado en el Anexo D.72, podemos observar que en general, el 100 % de los usuarios mantiene una opinión positiva sobre la facilidad de uso de la funcionalidad. Sin embargo, se puede destacar que en general, el 81,8 % de los usuarios tiene opiniones positivas sobre la sencillez de la funcionalidad y el resto mantuvo una opinión neutra sobre ello.
- Finalmente, para la tarea **configurar periodo y año** ilustrada en la Figura D.73, se puede observar que si bien el 100 % de los usuarios concuerda que era de fácil uso y el 72,7 % que era sencillo, hubo comentarios de usuarios que sugirieron la inutilidad de la funcionalidad.
- Para la tarea **crear un plan de clases** ilustrado en la Figura D.74, se puede observar que el 100 % de los usuarios concuerda que era de fácil uso y sencilla.

Resultados relevantes post-evaluación

Sobre los resultados del cuestionario de post-evaluación ilustrada en las Figuras 7.3 y 7.4, se puede decir que:

- El 36,4 % de los usuarios piensa que no necesita ayuda o un manual de usuario para volver a utilizar el prototipo. Sin embargo, existe un 27,2 % que posee una opinión neutral al respecto y un 27,2 % de los usuarios que si piensa que necesita ayuda o un manual.
- Sobre la funcionalidad, el 45,5 % de los usuarios está o de acuerdo con que las funcionalidades del prototipo se encuentran bien integradas, o posee una opinión neutra. Esto puede deberse en parte por los comentarios negativos recibidos en algunos tópicos como se pudo ver en las Figuras D.69 y D.73.
- El 90,9 % de los usuarios está de acuerdo con que el prototipo es fácil de usar. Sin embargo, el resto mantuvo una opinión neutral.
- El 72,7 % de los usuarios no está de acuerdo con que el prototipo sea sencillo de usar. Sin embargo, el resto mantuvo una opinión neutral.
- El 90,9 % de los usuarios está de acuerdo con que el prototipo cumple perfectamente las expectativas durante las pruebas.
- El 100 % de los usuarios está de acuerdo con que la interfaz del prototipo es agradable.
- El 100 % de los usuarios está de acuerdo con que no necesita de más funcionalidades para poder realizar un plan de clases.
- Finalmente, el 36 % de los usuarios se siente seguro de utilizar el prototipo. Sin embargo, el 54,5 % de los usuarios mantiene una opinión neutral y el resto posee una opinión negativa al respecto.

7.4. Análisis de resultados

Debido a que el objetivo general de esta memoria es reducir el tiempo de creación de un plan de clases de la carrera Ingeniería Civil en Computación, en las pruebas

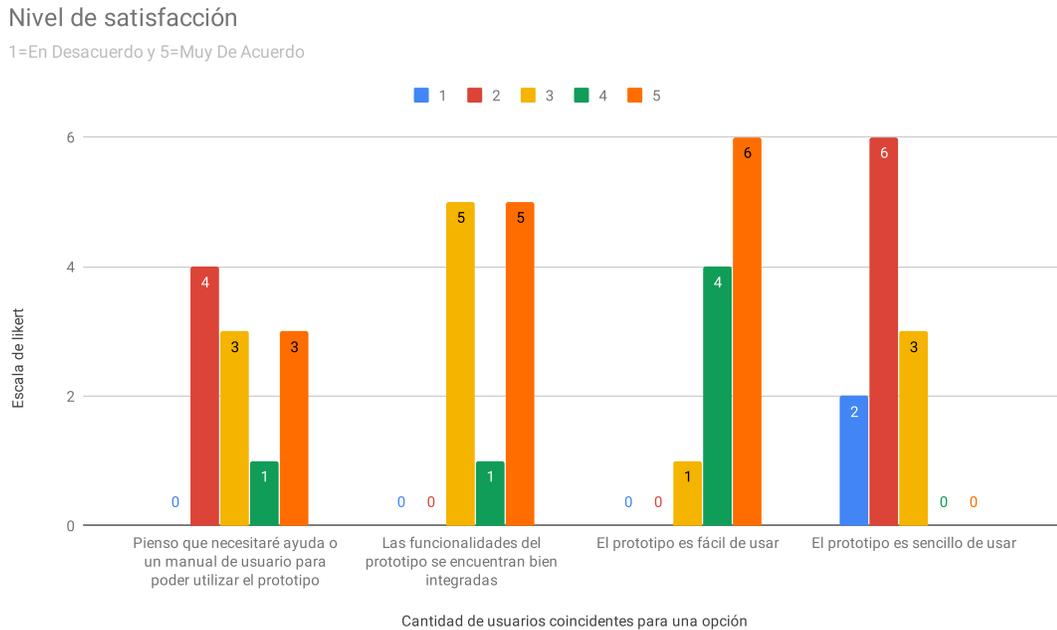


Figura 7.3: Gráfico de satisfacción del usuario con respecto a la aplicación (Parte 1)

de usabilidad detalladas en la Sección 7.2 se crearon una lista de nueve actividades que el usuario debía realizar, donde se asumió que éstas corresponden a los pasos mínimos para poder crear un plan de clases. Sabiendo que, el comité curricular indica que el tiempo promedio de creación de un plan de clases es de cuatro a seis horas (se tomó cinco horas como tiempo estimado) y el promedio de modificación de un plan de clases es de una a dos horas (se tomó una hora como tiempo estimado), la Figura 7.5 detalla el tiempo total promedio por usuario en la creación de un plan de clases.

Se aprecia que el tiempo promedio entre los once usuarios es de 49 minutos con 07 segundos para crear un plan de clases desde cero, aproximadamente un sexto del tiempo reportado por el comité curricular. Si bien se observa que los usuarios que les tomó una mayor cantidad de tiempo (de izquierda a derecha, 64 minutos, 88 minutos y 83 minutos) si sobrepasa el tiempo estimado para poder modificar un plan de clases, estos tiempos no sobrepasan el tiempo promedio de creación y por lo tanto, puede afirmar que el objetivo general de esta memoria fue cumplido.

Con respecto a los objetivos específicos, se tiene que:

- En la Figura 7.5 se puede inferir que el tiempo de creación de un plan de clases

Nivel de satisfacción

1=En Desacuerdo y 5=Muy De Acuerdo

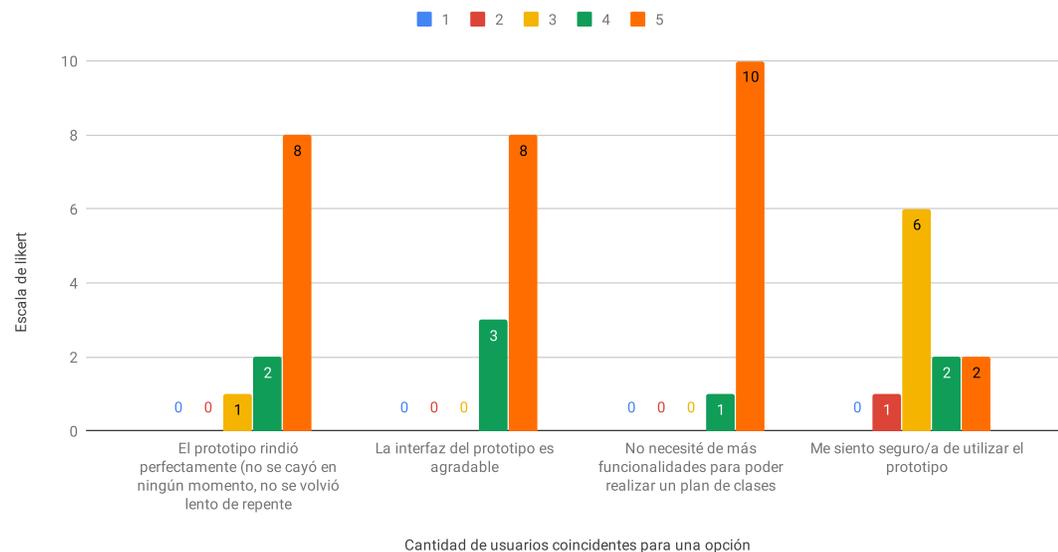


Figura 7.4: Gráfico de satisfacción del usuario con respecto a la aplicación (Parte 2)

se redujo de 300 minutos a 49 minutos aproximadamente, por lo que se puede decir que se aumentó la velocidad de la creación de un plan de clases. Inclusive, podría afirmarse que este tiempo puede reducirse aún más por medio del uso reiterado de la aplicación.

- Como se explica anteriormente, en las Figuras D.68 y D.69 se aprecia que el 100 % de los usuarios opina que el crear una competencia, un aprendizaje y dos saberes fue fácil. Lamentablemente, dado que sólo el 27,2 % de los usuarios opina que agregar un aprendizaje a una competencia y los saberes a un aprendizaje fue fácil se puede concluir que éste objetivo fue cumplido a medias. Si se toma en cuenta los comentarios de los usuarios al respecto, donde al menos cuatro opina que ésta funcionalidad no fue intuitiva, podría afirmarse que de corregir ésta falencia, el objetivo podría cumplirse en un futuro.
- Como se explicó anteriormente, en la Figura 7.3 se aprecia que el 90 % de los usuarios opina que la aplicación es fácil de usar. Con esto, se puede decir que el objetivo facilitar el proceso de creación de un plan de clases de la carrera Ingeniería Civil en Computación se cumplió. También permitir de manera sen-

Tiempo total de creación de un plan de clases

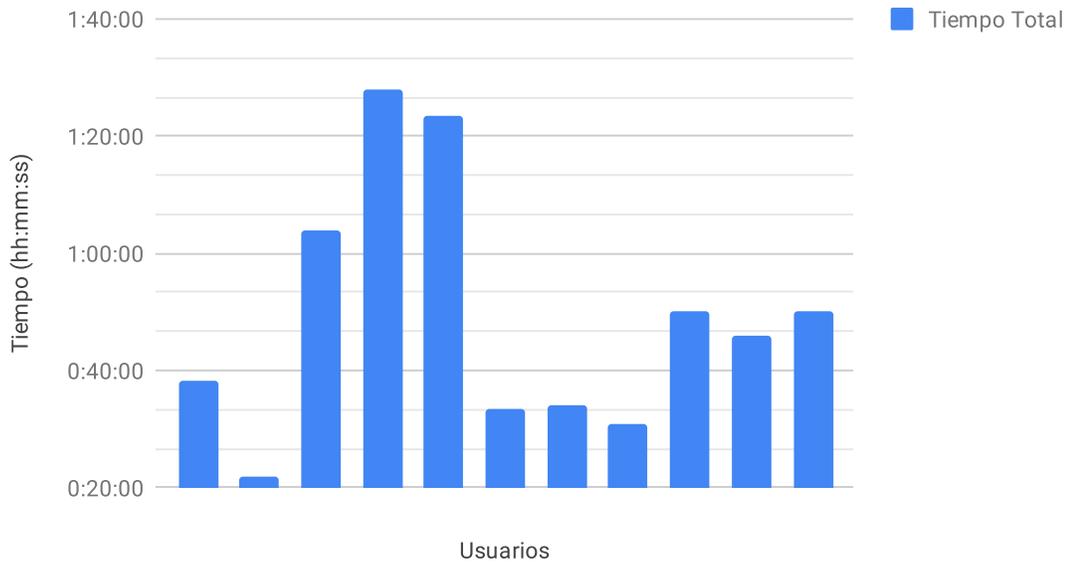


Figura 7.5: Gráfico de tiempo total de creación de un plan de clases

cilla realizar cambios en el plan de clases de un curso de la carrera Ingeniería Civil en Computación.

- Dado que diseñar un conjunto de pruebas para que los usuarios pudieran garantizar el uso de la aplicación fue parte de la metodología propuesta, si puede decirse que éste objetivo se cumplió.
- Finalmente, aunque automatizar la persistencia de la creación de un plan de clases no es algo tangible, por el simple hecho de que el usuario pueda modificar un plan de clases ya creado asegura que éste objetivo ha sido cumplido. Sin embargo, se considera como trabajo futuro solicitar un servidor de la universidad donde la aplicación pueda ser montada para que de alguna forma se asegure que éste objetivo sea cumplido. Con esto se puede decir que en general, se cumple el objetivo específico de facilitar el proceso de creación de un plan de clases.

8. Conclusiones y Trabajos Futuros

Desde el año 2004 la Universidad de Talca inició un proceso de transformación de carreras de pregrado, utilizando el proyecto de rediseño curricular basado en competencias y pretendiendo visualizar el proceso de enseñanza a partir de la aplicación de conocimientos, habilidades y actitudes en base a una profesión específica.

En esta memoria se detalla el desarrollo y construcción de un prototipo para poder facilitar la creación de un plan de clases, esencial para poder establecer actividades y tareas que reflejen la propuesta formativa de cada carrera. Los planes de clases se crean en base a competencias, aprendizajes y saberes aplicados a cursos, unidades y actividades y reflejan el contenido de las clases, laboratorios y evaluaciones. El prototipo propuesto permite crear y mantener un registro de competencias creadas a partir del currículo de formación académica y profesional, además de permitir crear y mantener un registro, tanto de los módulos y sus unidades como las actividades relacionadas.

El prototipo fue probado a través de pruebas de caja negra, verificando si éste cumple con los requisitos capturados en base a la metodología de desarrollo tradicional iterativo. También fue aplicado un test de usabilidad, con el motivo de probar los objetivos planteados en esta memoria y evaluar el grado en el que los usuarios son capaces de interactuar con la aplicación. Tal y como se mencionó en la Sección 7.3, el prototipo pasó todas las pruebas de caja negra, y en el test de usabilidad entregó tiempos menores a aquellos obtenidos usando el método tradicional de planillas de Microsoft Excel o tablas en Microsoft Word.

El objetivo principal del desarrollo de esta aplicación era reducir el tiempo de creación de un plan de clases. Se puede aseverar que esto fue cumplido, ya que

como se pudo observar en los resultados detallados en la Sección 7.3, el tiempo de creación del plan de clases se vio drásticamente reducido, de 300 minutos a 49 minutos aproximadamente. Esto se sustenta en gran parte en el hecho que cada etapa del desarrollo de este software fue acompañado de pruebas de caja negra, por cada funcionalidad nueva añadida. Cabe destacar que los usuarios mencionaron que el prototipo fue simple de usar y que no sería necesario un manual de uso, a pesar del riesgo de que se debía tener un dominio sobre el contenido de un plan de clases para poder comprender a cabalidad lo que se estaba realizando.

Como trabajo futuro, se tendrá en consideración el tratar de buscar un sitio en la universidad para poder publicar la plataforma y así poder asegurar de que el objetivo secundario de persistencia se cumpla. También podría considerarse el hecho de enlazar este prototipo con otras aplicaciones de la universidad, lo que haría de éste prototipo algo más visible hacia la comunidad.

Según los comentarios de los usuarios que probaron el sistema, la funcionalidad de configuración general está demás; por lo que sería bueno el modificarla. Reemplazar la configuración del año y el periodo por las fechas de inicio de los periodos sería una buena opción; así poder modificar el plan de clases y añadir las semanas en las cuales se realizan las actividades sería una buena forma de acercar el plan de clases a lo que el cliente espera y que no pudo tener debido al tiempo de desarrollo.

Encontrar una forma de simplificar aún más creación de competencias, aprendizajes y saberes, como cargarlos automáticamente de la plataforma de Syllabus. Lo que garantizaría que la aplicación fuese mucho más sencilla y hubiese menos confusiones a la hora de crear alguno.

Finalmente, se debe completar la funcionalidad de evaluación de módulos; funcionalidad que debería permitir tanto al cuerpo docente como al alumnado validar las actividades generadas a través del plan de clases. En ella, el profesor podría marcar aquellos saberes que se han impartido durante una unidad de aprendizaje y el alumnado podrá validar esta información, mediante el uso de un formulario de selección múltiple. Esta funcionalidad se encontraba en etapa de backend y controlador, antes de que junto al cliente se decidiera el eliminarlo.

Glosario

API: Interfaz de programación que proporciona acceso a funciones de un software. Suelen ser publicadas por sitios para permitir acceso a características de bajo nivel del mismo sitio, procurando el no dar información sobre cómo ésta tarea es desarrollada. Normalmente es utilizado para construir aplicaciones sin la necesidad de volver a programar funciones que ya fueron hechas, priorizando la reutilización de código que se sabe que funciona correctamente [20].

AJAX (*Asynchronous JavaScript And XML*): Técnica de desarrollo web para crear aplicaciones interactivas que son ejecutadas por el lado del cliente (el navegador). Mejora la interacción, la velocidad y la usabilidad de las aplicaciones [18].

Common Language Runtime: Entorno de ejecución orientado a códigos de programas que funcionan en .NET. El Common Language Runtime facilita el diseño de componentes y aplicaciones, donde los objetos interactúan entre los distintos idiomas y sus componentes pueden ser integrados. Esto se debe a los compiladores del lenguaje y las herramientas utilizan un sistema común definido por el tiempo de ejecución [15].

CRUD: Sigla en inglés que significa *Create, Read, Update, Delete* (Crear, Leer, Modificar, Eliminar o Habilitar y Deshabilitar) [12].

Componente: Bloque de construcción de software de cómputo. Según [24] un componente es “*una parte modular, desplegable y sustituible del sistema, que incluye la implantación y expone un conjunto de interfaces*”.

DACUM: Developing A CurriculUM. Es un método de trabajo y análisis que permite en base a criterios de desempeño, crear un currículum [22].

Fácil: Según la Real Academia Española, algo fácil es algo que no requiere gran esfuerzo, capacidad o habilidad. Por ejemplo: un problema resuelto sin problema.

Framework: Marco o entorno de trabajo, conjunto estándar de conceptos, criterios y buenas prácticas con el cual se puede enfrentar y resolver un problema [4]. Un Framework provee una arquitectura, estructura y una metodología de trabajo previamente definida cuya característica es su alta cohesión y bajo acoplamiento [29].

Lenguaje de Marcado: Forma de codificar un documento con base en etiquetas, donde se busca incorporar información adicional sobre la estructura de un texto. A diferencia de un lenguaje de programación, el lenguaje de marcado no posee funciones de aritmética o definiciones de variables [13, 7].

Gestionar: En este contexto, se refiere a las funciones básicas de persistencia de un software. Es decir: Crear, Modificar, Leer, Eliminar (o Habilitar y Deshabilitar en caso de que no se permita la eliminación). Consultar CRUD.

Módulo: También conocido como **Curso**. Un módulo es una unidad de trabajo-aprendizaje, que permiten que el estudiante pueda habilitarse en una competencia asignada a éste [3].

Plan de Clases: Plan que se encarga de organizar y mostrar detalladamente las sesiones de clases, en función de los propósitos expresados en el syllabus [3].

Sencillo: Según la Real Academia Española, algo sencillo es algo que no ofrece dificultad. Por ejemplo: un problema resuelto en pocos pasos.

Stakeholder: Interesado o parte interesada. Son aquellas personas que se ven afectadas por las decisiones o las actividades de un proyecto, empresa u organización [5].

Syllabus: Documento que traduce los requisitos de cada escuela sobre las competencias e itinerarios del aprendizaje que representan la propuesta pedagógica de una carrera [3].

Usabilidad: Medida de calidad de la experiencia de un usuario al momento de interactuar con una aplicación. Estudia la forma de diseñar aplicaciones para que las interacciones entre usuario y éstas sean de forma fácil, cómoda e intuitiva [21].

Bibliografía

- [1] Amazon. Amazon web services (aws) - cloud computing services. <https://aws.amazon.com/>. [Online; accedido 16-junio-2018].
- [2] Bootstrap. Bootstrap - the most popular html, css, and js library in the world. <https://getbootstrap.com/>. [Online; accedido 17-noviembre-2018].
- [3] Vicerrectoría de Docencia de Pregrado. Modelo educativo universidad de Talca. http://fen.otalca.cl/pdf/Modelo_Educativo_Universidad_de_Talca.pdf. [Online; accedido 16-agosto-2018].
- [4] Cambridge Dictionary. Significado de framework. <https://dictionary.cambridge.org/es/diccionario/ingles/framework>. [Online; accedido 16-octubre-2018].
- [5] R Edward Freeman and David L Reed. Stockholders and stakeholders: A new perspective on corporate governance. *California management review*, 25(3):88–106, 1983.
- [6] Gentelella GitHub. Gentelella. <https://github.com/puikinsh/gentelella>. [Online; accedido 17-noviembre-2018].
- [7] Allen Renear y Steven DeRose James Coombs' Markup systems and the future of scholarly text processing. <http://xml.coverpages.org/coombs.html>. [Online; accedido 17-noviembre-2018].
- [8] jQuery. jquery. <https://jquery.com/>. [Online; accedido 18-noviembre-2018].
- [9] Jesse Liberty. *Programming C: Building .NET Applications with C*. Center on Education And Training For Employment, 2005. [Online; accedido 18-noviembre-2018].

- [10] FullCalendar LLC. Fullcalendar - javascript event calendar. <https://fullcalendar.io/>. [Online; accedido 25-noviembre-2018].
- [11] SpryMedia Ltd. Datatables - table plug-in for jquery. <https://datatables.net/>. [Online; accedido 18-noviembre-2018].
- [12] James Martin. *Managing the Data Base Environment*. Pearson Education, 1983.
- [13] Merriam-Webster. Markup language | definition of markup lenguaje. <https://www.merriam-webster.com/dictionary/markup%20language>. [Online; accedido 17-noviembre-2018].
- [14] Microsoft. Asp.net overview. [https://docs.microsoft.com/en-us/previous-versions/aspnet/4w3ex9c2\(v=vs.100\)](https://docs.microsoft.com/en-us/previous-versions/aspnet/4w3ex9c2(v=vs.100)). [Online; accedido 27-octubre-2018].
- [15] Microsoft. Common language runtime overview. [https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-1.1/ddk909ch\(v=vs.71\)](https://docs.microsoft.com/en-us/previous-versions/dotnet/netframework-1.1/ddk909ch(v=vs.71)). [Online; accedido 16-octubre-2018].
- [16] Microsoft. Introducción a asp.net core. <https://docs.microsoft.com/es-es/aspnet/core/view̄aspnetcore-2.1>. [Online; accedido 27-octubre-2018].
- [17] Microsoft. Razor syntax reference for ASP.NET Core. <https://docs.microsoft.com/en-us/aspnet/core/mvc/views/razorview̄aspnetcore-2.1>. [Online; accedido 17-noviembre-2018].
- [18] MDN Web Docs Mozilla. Ajax. <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>. [Online; accedido 18-noviembre-2018].
- [19] MDN Web Docs Mozilla. Learn to style html using css. <https://developer.mozilla.org/en-US/docs/Learn/CSS>. [Online; accedido 17-noviembre-2018].
- [20] Brian Mulloy. *Web API design*. Academic Press, 2013.
- [21] Jakob Nielsen. *Usability 101: Introduction to usability*.
- [22] Robert Norton. *DACUM Handbook Second Edition. Leadership Training Series No. 67*. Center on Education And Training For Employment, 1997.

- [23] Pluralsight. Ready to try javascript? <https://www.javascript.com/>. [Online; accedido 18-noviembre-2018].
- [24] Roger Pressman. *Software engineering: a practitioner's approach*. Palgrave Macmillan, 2010.
- [25] Ian Sommerville. *Ingeniería del software*. Pearson educación, 2011.
- [26] World Wide Web Consortium (W3C). Html 5.2 w3c recommendation. <https://www.w3.org/TR/html/>. [Online; accedido 17-noviembre-2018].
- [27] Rodolfo Schmal y Andrés Ruiz-Tagle. Una metodología para el diseño de un currículo orientado a las competencias. *Ingeniare. Revista chilena de Ingeniería*, 16 n°1:1–12, 2008.
- [28] Harry Boone y Deborah Boone. Analyzing likert data. *Journal of extension*, 50(2):1–5, 2012.
- [29] Thomas Gross y Douglas Schmidt. Framework design, a role modeling approach. <https://riehle.org/computerscience/research/dissertation/dissa4.pdf>. [Online; accedido 16-octubre-2018].
- [30] Srinivas Nidhra1 y Jagruthi Dondeti. Black box and white box testing techniques - a literature review. *International Journal of Embedded Systems and Applications*, pages 1–22, 2012.
- [31] Guillaume Potier y Marc-André Lafortune. Parsley, the ultimate javascript form validation library. <http://parsleyjs.org/>. [Online; accedido 25-noviembre-2018].
- [32] Glenn Krasner y Stephen Pope. A description of the model-view-controller user interface paradigm in the smalltalk-80 system. *Journal of object oriented programming*, 1(3):26–49, 1988. [Online; accedido 15-octubre-2018].

ANEXOS

A. Marco Teórico

En el anexo se presentarán los diagramas e imágenes que facilitan el entender el marco teórico. El diagrama de metodología para el diseño curricular, que representa la metodología para el diseño de un currículo propuesto por la Universidad de Talca, el plan curricular 16 de ingeniería civil en computación, los modelos curriculares similares al diseño propuesto y los software y soluciones similares al prototipo propuesto.

A.1. Diagramas de Metodología para el Diseño Curricular

Metodología Propuesta para el diseño curricular que denota del perfil profesional dado por las competencias y aprendizajes o capacidades resultantes del taller DACUM.

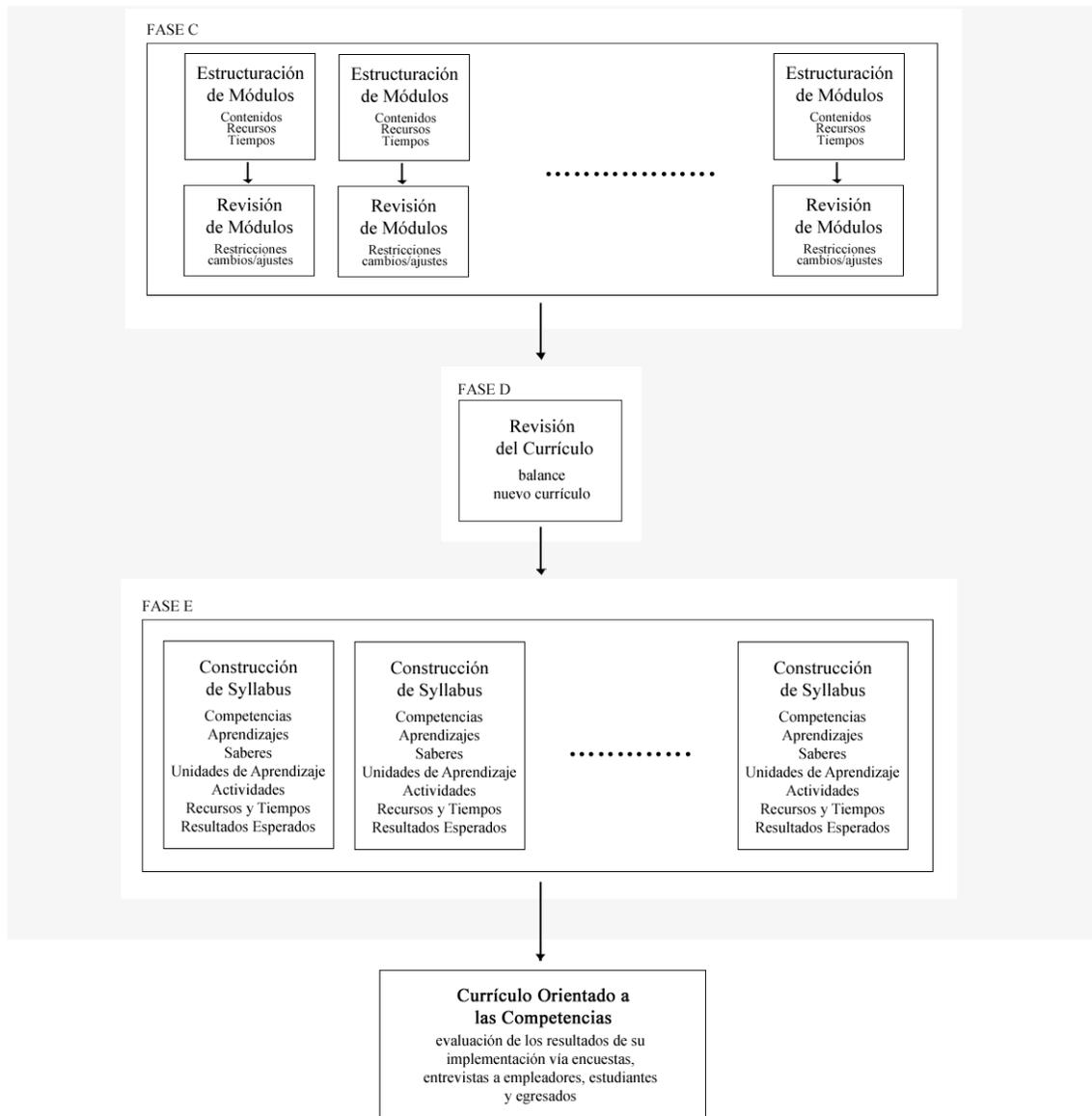


Figura A.2: Diagrama de Metodología para el Diseño Curricular Parte 2

A.2. Plan Curricular Ingeniería Civil en Computación

El plan curricular 16 de Ingeniería Civil en Computación, que fue aprobado el 16 de junio de 2015 cuya modificación fue realizada el 3 de mayo de 2017.

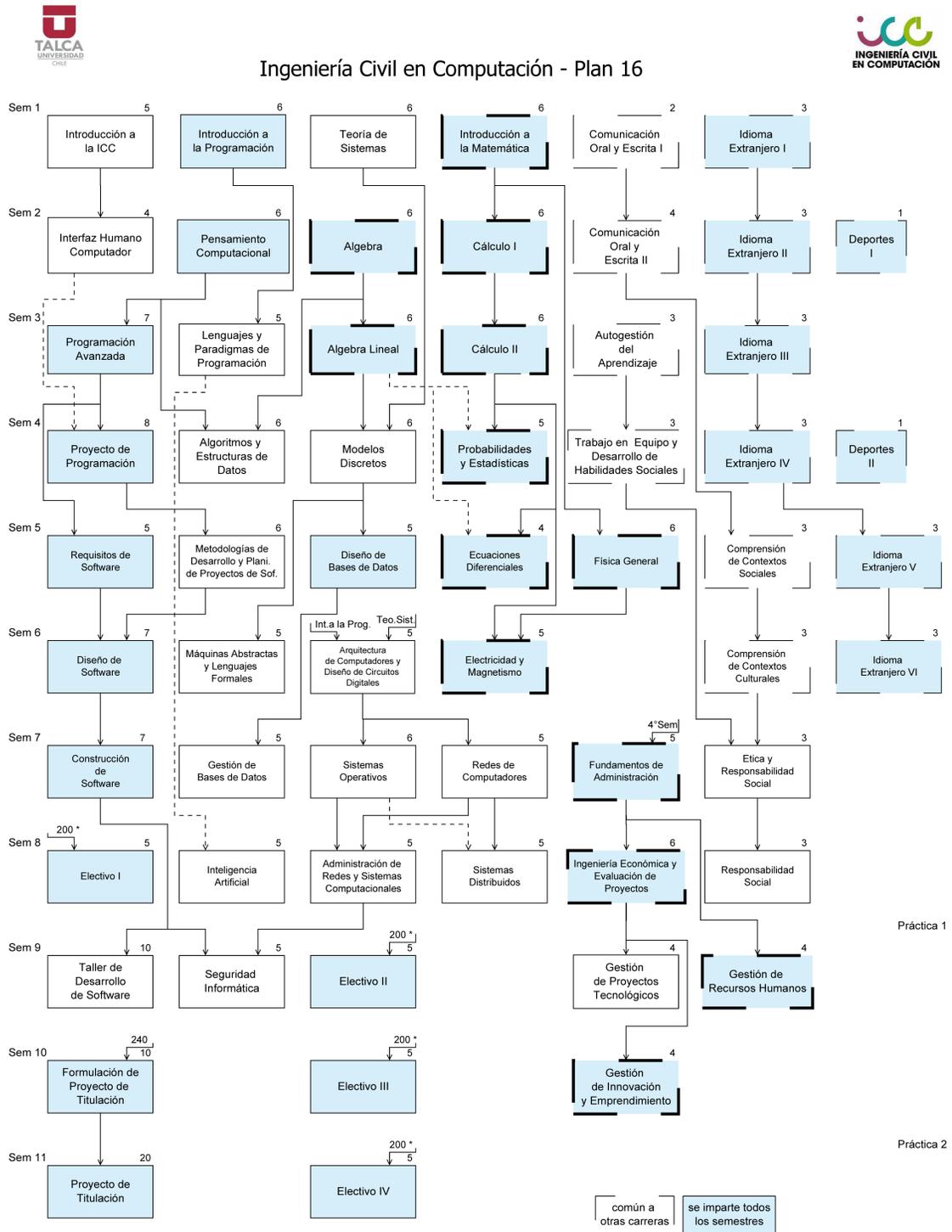


Figura A.3: Plan Curricular 16 de Ingeniería Civil en Computación

A.3. Software

A.3.1. Software de Asistencia Universidad de Talca

Mis Cursos

Cursos inscritos 2019/Otoño

CONSTRUCCION DE SOFTWARE
★ Sección: A + Créditos: 8

PENSAMIENTO COMPUTACIONAL
★ Sección: A + Créditos: 6

PROYECTO DE PROGRAMACION
★ Sección: A + Créditos: 6

REQUISITO DE SOFTWARE
★ Sección: A + Créditos: 6

The screenshot displays a user interface for a university's course management system. At the top, there is a header 'Mis Cursos'. Below it, a section titled 'Cursos inscritos' (Enrolled Courses) is shown, with a dropdown menu set to '2019/Otoño'. The list contains four courses, each with a blue title and a star icon followed by 'Sección: A' and a plus sign followed by the credit value. The courses are: 'CONSTRUCCION DE SOFTWARE' (8 credits), 'PENSAMIENTO COMPUTACIONAL' (6 credits), 'PROYECTO DE PROGRAMACION' (6 credits), and 'REQUISITO DE SOFTWARE' (6 credits).

Figura A.4: Software de Asistencia Universidad de Talca: Lista de cursos asociados a un profesor

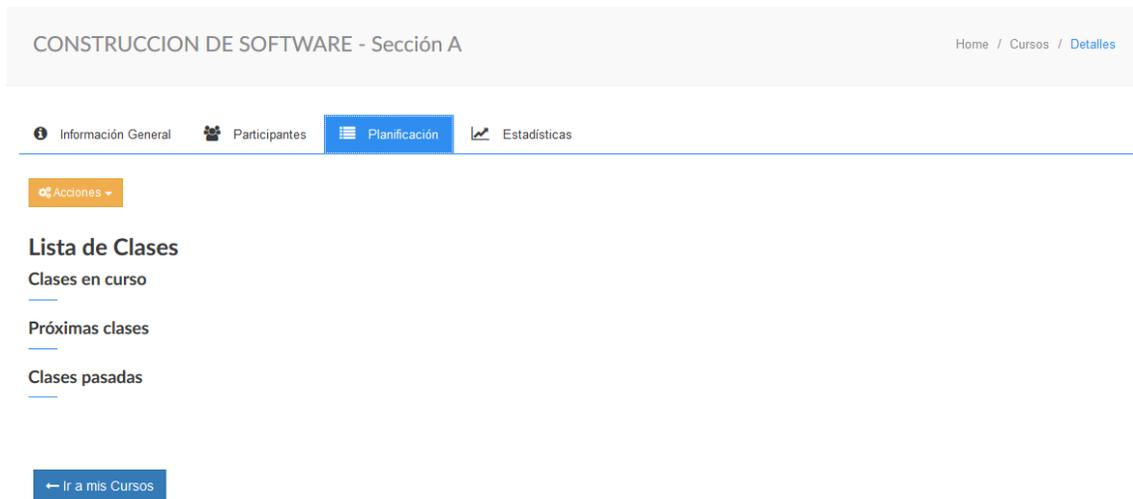


Figura A.5: Software de Asistencia Universidad de Talca: Lista de clases de un curso vacío

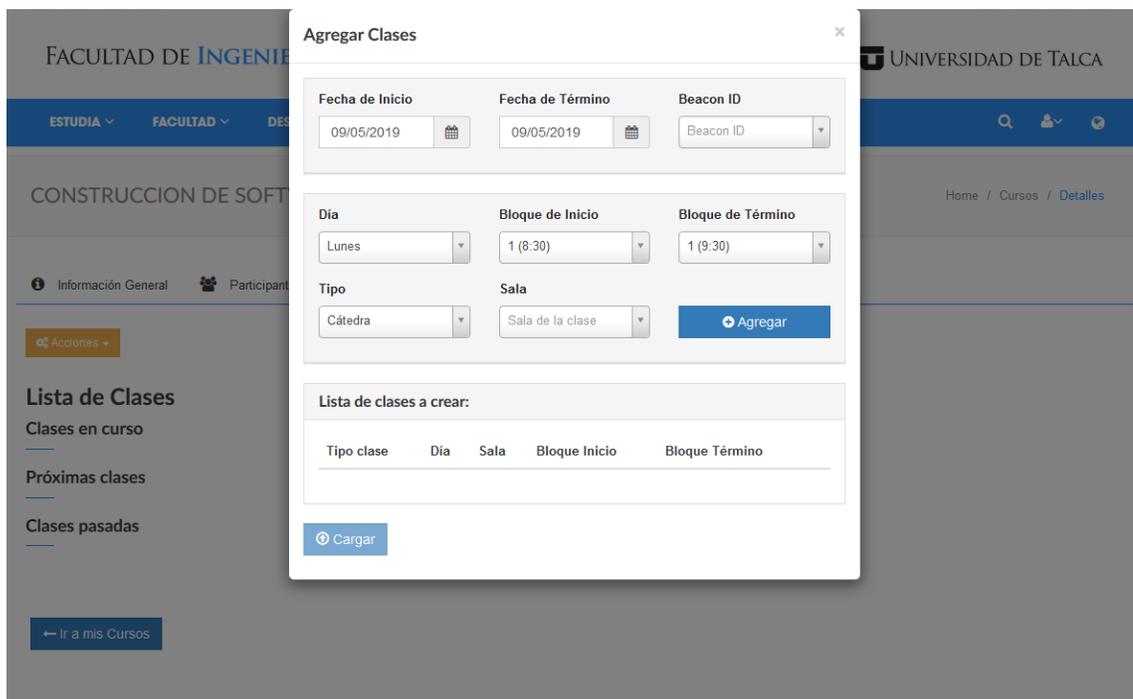


Figura A.6: Software de Asistencia Universidad de Talca: Modal agregar clases a un curso

Agregar Clases

Fecha de Inicio: 13/05/2019
Fecha de Término: 20/07/2019
Beacon ID: 3HRr

Día: Martes
Bloque de Inicio: 4 (12:00)
Bloque de Término: 5 (14:10)

Tipo: Laboratorio
Sala: Laboratorio de Co...

Agregar

Lista de clases a crear:

Tipo	clase	Día	Sala	Bloque Inicio	Bloque Término	
Clase	Lunes	Sala C1	9:40.00	11:50.00	Eliminar	

Cargar

Figura A.7: Software de Asistencia Universidad de Talca: Modal agregar clases a un curso con una clase agregada

CONSTRUCCION DE SOFTWARE - Sección A Home / Cursos / Detalles

Información General Participantes **Planificación** Estadísticas

Acciones

Lista de Clases

Clases en curso

Próximas clases

13-05-2019 9:40 - 11:50	Clase 1 CONSTRUCCION DE SOFTWARE	+
13-05-2019 12:00 - 14:10	Laboratorio 1 CONSTRUCCION DE SOFTWARE	+
20-05-2019 9:40 - 11:50	Clase 2 CONSTRUCCION DE SOFTWARE	+
20-05-2019 12:00 - 14:10	Laboratorio 2 CONSTRUCCION DE SOFTWARE	+
27-05-2019 9:40 - 11:50	Clase 3 CONSTRUCCION DE SOFTWARE	+
27-05-2019 12:00 - 14:10	Laboratorio 3 CONSTRUCCION DE SOFTWARE	+
03-06-2019 9:40 - 11:50	Clase 4 CONSTRUCCION DE SOFTWARE	+
03-06-2019 12:00 - 14:10	Laboratorio 4 CONSTRUCCION DE SOFTWARE	+
10-06-2019 9:40 - 11:50	Clase 5 CONSTRUCCION DE SOFTWARE	+
10-06-2019 12:00 - 14:10	Laboratorio 5 CONSTRUCCION DE SOFTWARE	+
17-06-2019 9:40 - 11:50	Clase 6 CONSTRUCCION DE SOFTWARE	+
17-06-2019 12:00 - 14:10	Laboratorio 6 CONSTRUCCION DE SOFTWARE	+
24-06-2019 9:40 - 11:50	Clase 7 CONSTRUCCION DE SOFTWARE	+

Figura A.8: Software de Asistencia Universidad de Talca: Lista de clases de un curso

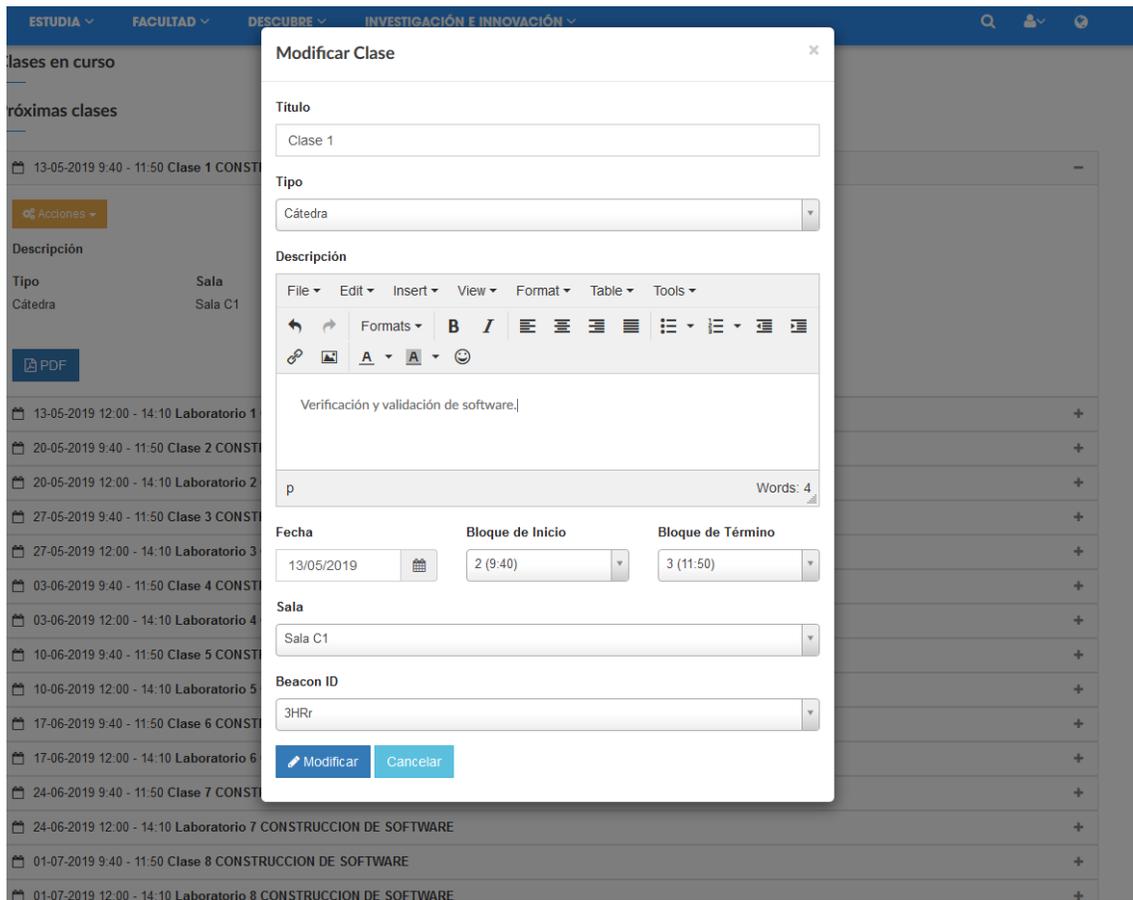


Figura A.9: Software de Asistencia Universidad de Talca: Modificar una Clase

13-05-2019 9:40 - 11:50 Clase 1 CONSTRUCCION DE SOFTWARE		-
Acciones		
Descripción		
Verificación y validación de software.		
Tipo	Sala	
Cátedra	Sala C1	
PDF		
13-05-2019 12:00 - 14:10 Laboratorio 1 CONSTRUCCION DE SOFTWARE		+
20-05-2019 9:40 - 11:50 Clase 2 CONSTRUCCION DE SOFTWARE		+
20-05-2019 12:00 - 14:10 Laboratorio 2 CONSTRUCCION DE SOFTWARE		+
27-05-2019 9:40 - 11:50 Clase 3 CONSTRUCCION DE SOFTWARE		+
27-05-2019 12:00 - 14:10 Laboratorio 3 CONSTRUCCION DE SOFTWARE		+
03-06-2019 9:40 - 11:50 Clase 4 CONSTRUCCION DE SOFTWARE		+

Figura A.10: Software de Asistencia Universidad de Talca: Ver una clase

A.3.2. Excel Plan de Clases Construcción de Software

Unidad I: Fundamentos de la Ingeniería de Software			
Semana	Rango Fechas	Actividades	Saberes
1	18/03/2019 19/03/2019	Presentación del Curso	N/A
		Conceptos Básicos	SDF-4.2: Explica por qué la creación de componentes de programación correctos es importante en la producción de software de alta calidad. [C] SE-3.2: Describe cómo el control de versiones puede ser utilizado para apoyar el manejo de release de software. [C] HCI-1.5a: Conoce el concepto de usabilidad en el contexto de aplicaciones de software. [C] SDF-4.9a: Conoce métodos de depugeo. [C] SE-2.3: Identifica y justifica los roles necesarios en un equipo de desarrollo de software. [U] SE-8.6: Discute las ventajas y desventajas de los diferentes tipos de reuso de software. [C]
2	25/03/2019 26/03/2019	Repaso de Ingeniería de Requisitos	SE-6.5: Compara y contrasta estrategias de integración, incluyendo top-down, bottom-up, e integración sandwich. [C] SE-3.5: Describe los aspectos relevantes a la hora de seleccionar las herramientas necesarias para el desarrollo de un sistema de software particular, incluyendo las herramientas necesarias para el manejo de requerimientos, modelado y diseño, implementación, construcción y pruebas. [C] SE-2.2: Crea y sigue una agenda de reuniones de equipo. [U]
		Repaso de Diseño de Software	HCI-2.2a: Conoce los principios de diseño de interfaces gráficas. [C] SDF4.7a: Conoce los aspectos y características básicas de la refactorización. [C] SE-6.1: Describe técnicas, expresiones de código (coding idioms) y mecanismos para la implementación de diseños logrando propiedades deseadas como confiabilidad, eficiencia y robustez. [C]
3	01/04/2019 02/04/2019	Planificación del Desarrollo de Software	SE-2.6a: Conoce métodos para la estimación de esfuerzo en el desarrollo de software. [C] SE-1.4: Diferencia las fases del proceso de desarrollo de software. [C] SE-8.2: Estima el impacto de una solicitud de cambio en un producto existente de tamaño medio. [U] SE-8.3: Usa refactorización en el proceso de modificación de un componente de software. [U] SE-2.6: Utiliza un método ad-hoc para estimar el esfuerzo de desarrollo de software (ej. tiempo) y lo compara con el esfuerzo efectivamente requerido. [U]
		Especificación Proyecto Unidad I (Informe)	N/A

Figura A.11: Excel Plan de Clases Construcción de Software: Unidad 1, semanas 3, 4 y 5

4	08/04/2019 09/04/2019	Fundamentos de la Construcción	IAS-3.3: Clasifica errores de validación comunes, y escribe código correcto de validación de entradas. [U]
			SE-3.3: Identifica ítems de configuración y utiliza una herramienta de control de código fuente en un pequeño proyecto en grupo. [U]
			SDF-4.7: Refactoriza un programa identificando oportunidades de aplicar abstracción procedural. [U]
			SE-7.3: Lleva a cabo, como parte de una actividad de equipo, la inspección de segmento de código de tamaño medio. [U]
			SDF-4.3: Identifica errores de codificación común que conducen a programas inseguros (ej. buffer overflow, memory leaks, código malicioso) y aplica estrategias para evitar dichos errores. [U]
			SDF-4.4: Conduce una revisión de código personal (enfocada en errores de programación típicos) sobre un componente de programación utilizando una klista de puntos provista. [U]
			SE-3.1: Describe la diferencia entre una configuración de manejo de software centralizada y distribuida. [C]
			SDF-4.8: Aplica una variedad de estrategias para el testeo y debugado de programas simples. [U]
5	15/04/2019 16/04/2019	Documentación de Software	SDF-4.11a: Conoce estándares de documentación para el proceso de desarrollo de software. [C]
			SP-6.1a: Conoce estándares para la redacción de informes profesionales. [C]
		Calidad de Software	SDF-4.11: Analiza el punto hasta el que código de otro programador concuerda con los estándares de documentación y de programación. [E]
			SE-6.4: Selecciona y usa un estándar de código definido en un pequeño proyecto de software. [U]
6	22/04/2019 23/04/2019	Roles del Equipo de Desarrollo	SE-6.3: Describe prácticas de codificación segura y defensiva. [C]
			SP-6.3: Desarrolla y entrega una presentación formal de calidad. [E]
		Prueba Unidad I	N/A
		Entrega Proyecto Unidad I	SP-6.1: Escribe documentos técnicos claros, concisos y precisos, siguiendo estándares bien definidos para el formato, la inclusión correcta de tablas, figuras y referencias. [U]
			SDF-4.12: Aplica estándares de documentación y codificación consistentes, que contribuyen a la legibilidad y mantención del software. [U]
		Especificación Proyecto Unidad II	N/A

Figura A.12: Excel Plan de Clases Construcción de Software: Unidad 1, semanas 4, 5 y 6

Unidad II: Verificación y Validación de Software			
Semana	Rango Fechas	Actividades	Saberes
7	29/04/2019 30/04/2019	Introducción a la Verificación y Validación de Software	SE-7.1: Distingue entre validación y verificación de un programa. [C] SE-7.2: Describe el rol que las herramientas juegan en la validación de software. [C] SE-7.5: Describe técnicas para identificar casos de prueba significativos (integración, regresión, de sistema, de aceptación, entre otros). [C] IAS-1.4: Explica el concepto de confianza y confiabilidad. [C]
		Revisión de Avance Proyecto Unidad II	N/A
8	06/05/2019 07/05/2019	Verificación y Validación de Software	IAS-3.4: Demuestra el uso de un lenguaje de programación de alto nivel para la prevención de condiciones de competencia y para el manejo de excepciones. [U] IAS-3.5: Demuestra la identificación y manejo adecuado de condiciones de error. [U] SE-3.6: Demuestra la capacidad de usar herramientas de software para apoyar el desarrollo de un proyecto de software de tamaño medio. [U] IAS-3.4a: Conoce los conceptos de condiciones de competencia y manejo de excepciones. [C] SE-9.3: Lista estrategias para minimizar fallos que pueden ser aplicadas en las distintas fases del ciclo de vida del software. [C] SE-7.8: Usa una herramienta de detección de defectos para detectarlos en un pequeño proyecto de software. [U]
		Revisión de Avance Proyecto Unidad II	N/A
9	13/05/2019 14/05/2019	Verificación y Validación de Software	SE-9.2: Describe cómo la confiabilidad de software contribuye a la confiabilidad de sistemas. [C] SE-7.4: Describe y distingue entre diferentes tipos y niveles de testing (unitario, de integración, de sistema, de aceptación, entre otros). [C] IAS-1.3: Explica los conceptos de autenticación, autorización y control de acceso. [C] SE-3.4: Describe cómo las herramientas de testeo estáticas y dinámicas disponibles pueden ser integradas en el ambiente de desarrollo de software. [C] SE-7.7: Describe cómo la seleccionar buenas pruebas de regresión y cómo automatizarlas. [C] SE-8.5: Esquematiza el proceso de testeo de regresión y su rol en el manejo de release. [C] SE-7.6: Crea y documenta un conjunto de pruebas para un segmento de código de tamaño medio. [U] HCI-1.5: Crea y conduce un test simple de usabilidad para una aplicación de software existente. [E]

Figura A.13: Excel Plan de Clases Construcción de Software: Unidad 2, semanas 7, 8 y 9

		Revisión de Avance Proyecto Unidad II	N/A
10	20/05/2019 21/05/2019	Suspensión lunes y martes por feriado de Glorias Navales.	N/A
11	27/05/2019 28/05/2019	Prueba Unidad II	N/A
		Entrega Proyecto Unidad II	GV-1.4: Construye una interfaz de usuario simple utilizando un API estándar. [U] IM-2.7: Usa un lenguaje de consulta declarativo para obtener información de una base de datos. [U] IM-1.6: Identifica aspectos relacionados a la persistencia de datos para una organización. [C] IM-1.13: Identifica vulnerabilidades y escenarios de fallo en tipos comunes de sistemas de información. [U] HCI-2.2: Crea una aplicación simple, en conjunto con ayuda y documentación, que apoye la interfaz gráfica del usuario. [U]
Unidad III: Caso Aplicado de Desarrollo Ágil			
Semana	Rango Fechas	Actividades	Saberes
12	03/06/2019 04/06/2019	Scrum	IM-1.12: Describe enfoques para escalar sistemas de información. [C] SP-6.2: Evalúa documentación técnica escrita para detectar problemas de varias indoles. [E]
		Revisión de Avance Proyecto Unidad III	N/A
		Especificación Proyecto Unidad III	N/A
13	10/06/2019 11/06/2019	Ejecución Simbólica	SE-9.1: Explica los problemas que existen para lograr altos niveles de confiabilidad. [C] SE-7.9: Discute las limitaciones de testeo en un dominio particular. [C]
		Particiones de Equivalencia	N/A
		Revisión de Avance Proyecto Unidad III	N/A
14	17/06/2019 18/06/2019	Pruebas de Caja Negra	N/A
		Revisión de Avance Proyecto Unidad III	N/A
15	24/06/2019 25/06/2019	Pruebas de Caja Blanca	N/A
		Revisión de Avance Proyecto Unidad III	N/A
16	01/07/2019 02/07/2019	Prueba Unidad III	N/A
		Revisión de Avance Proyecto Unidad III	N/A
17	08/07/2019 09/07/2019	Entrega Proyecto Unidad III	IM-1.7: Critica una aplicación de información con respecto a si satisface las necesidades de información de un usuario. [E] SDF-4.9: Construye, ejecuta y debugea programas utilizando una IDE moderna y herramientas asociadas, tales como herramientas de testeo unitario y debuggers visuales. [U] SDF-4.10: Construye y debugea programas usando las librerías estándar disponibles para un lenguaje de programación elegido. [U] SE-6.2: Construye código robusto utilizando mecanismos de manejo de

Figura A.14: Excel Plan de Clases Construcción de Software: Unidad 2, semanas 10 y 11 y Unidad 3, semanas 12, 13, 14, 15, 16 y 17

17	08/07/2019 09/07/2019	Entrega Proyecto Unidad III	IM-1.7: Critica una aplicación de información con respecto a si satisface las necesidades de información de un usuario. [E]
			SDF-4.9: Construye, ejecuta y debugea programas utilizando una IDE moderna y herramientas asociadas, tales como herramientas de testeo unitario y debuggers visuales. [U]
			SDF-4.10: Construye y debugea programas usando las librerías estándar disponibles para un lenguaje de programación elegido. [U]
			SE-6.2: Construye código robusto utilizando mecanismos de manejo de excepciones. [U]
			SE-6.7: Describe el proceso de analizar e implementar cambios a una extensa base de código existente. [C]
			SE-6.6: Describe el proceso de analizar e implementar cambios a una base de código desarrollada para un proyecto específico. [C]
			HCI-2.3: Conduce una evaluación cuantitativa y discute/reporta sus resultados. [U]
18	17-07-2019	Recuperación	N/A
		Reclamos (con fundamento)	N/A

Figura A.15: Excel Plan de Clases Construcción de Software: Unidad 3, semana 17

B. Especificación de Requisitos

En este anexo se presentan los diagramas que facilitan el entender la captura de requisitos. La matriz de dependencia dividida en dos imágenes, que facilita el análisis de prioridades de requisitos y el diagrama de caso de uso nivel 0, que corresponde a un esquema de las principales funcionalidades del prototipo.

B.1. Matriz de Dependencias

RS/RU	RU01	RU02	RU03	RU04	RU05	RU06	RU07	RU08
RS01	X	X	X	X	X	X	X	X
RS02	X							
RS03	X							
RS04	X							
RS05	X							
RS06		X						
RS07		X						
RS08		X						
RS09	X		X		X		X	X
RS10	X							
RS11			X			X		
RS12					X			
RS13					X			
RS14					X			
RS15					X			
RS16					X			
RS17					X			
RS18								
RS19								
RS20								
RS21								
RS22						X		
RS23								
RS24					X			
RS25							X	
RS26							X	
RS27							X	
RS28							X	
RS29							X	
RS30							X	
RS31							X	
RS32								X
RS33								X
RS34								X
RS35								X

Figura B.1: Matriz de Dependencias, Requisitos de Usuario del 1 al 8. Requisitos de Sistema del 1 al 35

RS/RU	RU09	RU10	RU11	RU12	RU13	RU14	RU15	RU16	RU17
RS01	X	X	X	X	X	X	X	X	X
RS02									
RS03									
RS04									
RS05									
RS06									
RS07									
RS08									
RS09				X	X	X			
RS10									
RS11	X								
RS12									
RS13									
RS14									
RS15									
RS16							X		
RS17							X		
RS18		X							
RS19		X							
RS20	X								
RS21	X								
RS22									
RS23			X						
RS24			X						
RS25									
RS26									
RS27									
RS28									
RS29								X	
RS30								X	
RS31									
RS32									
RS33									
RS34									
RS35									

Figura B.2: Matriz de Dependencias, Requisitos de Usuario del 9 al 17. Requisitos de Sistema del 1 al 35

RS/RU	RU01	RU02	RU03	RU04	RU05	RU06	RU07	RU08
RS36								X
RS37								X
RS38								X
RS39								
RS40								
RS41								
RS42								
RS43								
RS44								
RS45								
RS46								
RS47								
RS48								
RS49								
RS50								
RS51								
RS52								
RS53								
RS54								
RS55								
RS56								
RS57								

Figura B.3: Matriz de Dependencias, Requisitos de Usuario del 1 al 8. Requisitos de Sistema del 36 al 57.

RS/RU	RU09	RU10	RU11	RU12	RU13	RU14	RU15	RU16	RU17
RS36									X
RS37									X
RS38									
RS39				X					
RS40				X					
RS41				X					
RS42				X					
RS43				X					
RS44					X				
RS45					X				
RS46					X				
RS47					X				
RS48				X	X				
RS49				X	X				
RS50					X				
RS51						X			
RS52					X				
RS53					X				
RS54					X				
RS55					X	X			
RS56					X	X			
RS57					X				

Figura B.4: Matriz de Dependencias, Requisitos de Usuario del 9 al 17. Requisitos de Sistema del 36 al 57.

B.2. Casos de Uso nivel 0

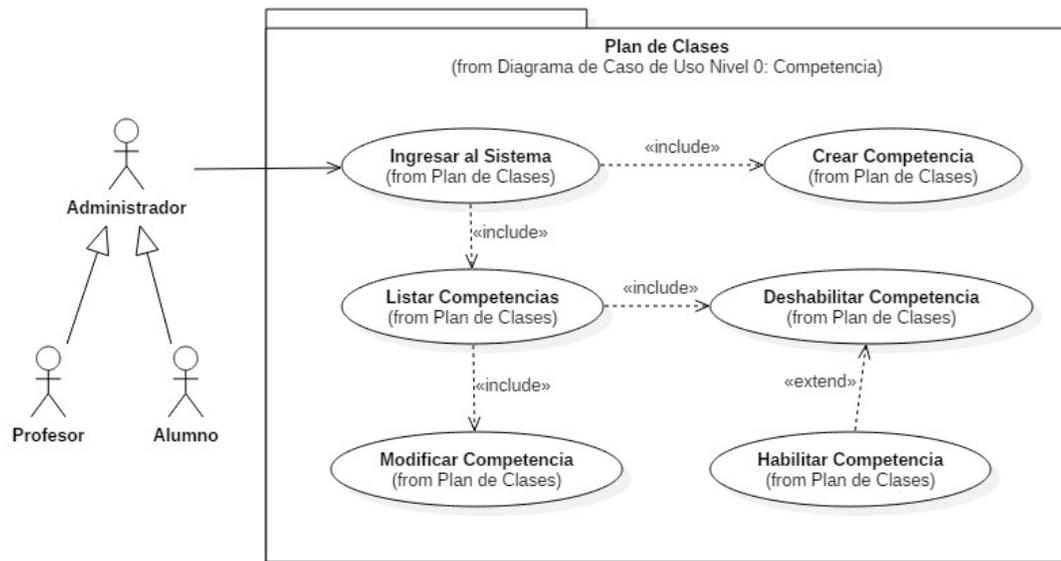


Figura B.5: Diagrama de Casos de Uso Nivel 0: Competencia

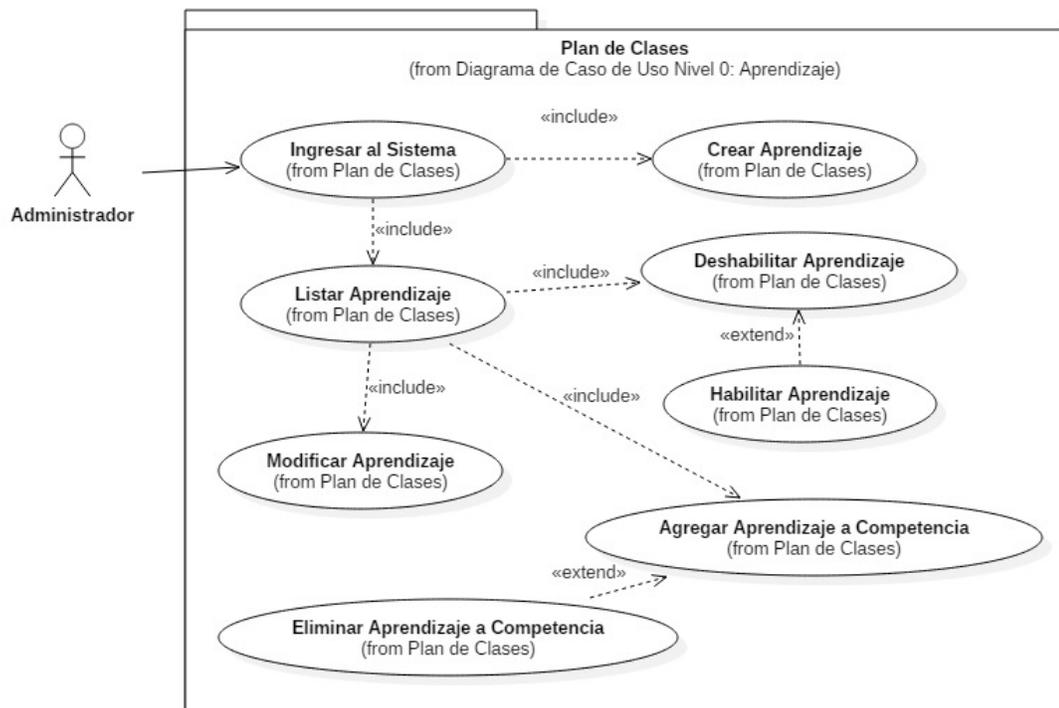


Figura B.6: Diagrama de Casos de Uso Nivel 0: Aprendizaje

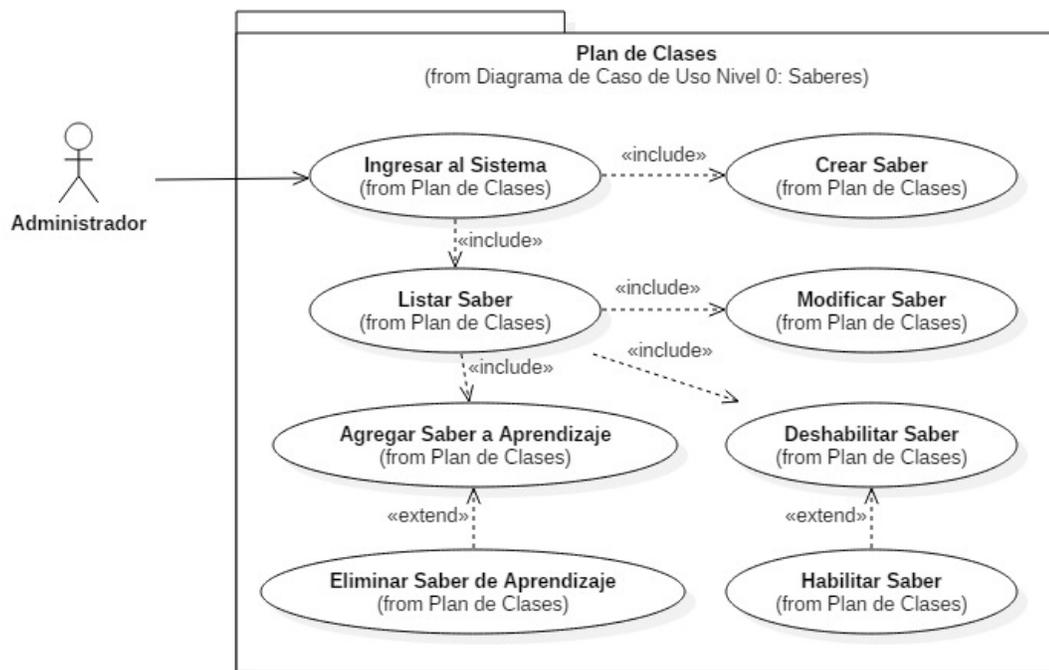


Figura B.7: Diagrama de Casos de Uso Nivel 0: Saber

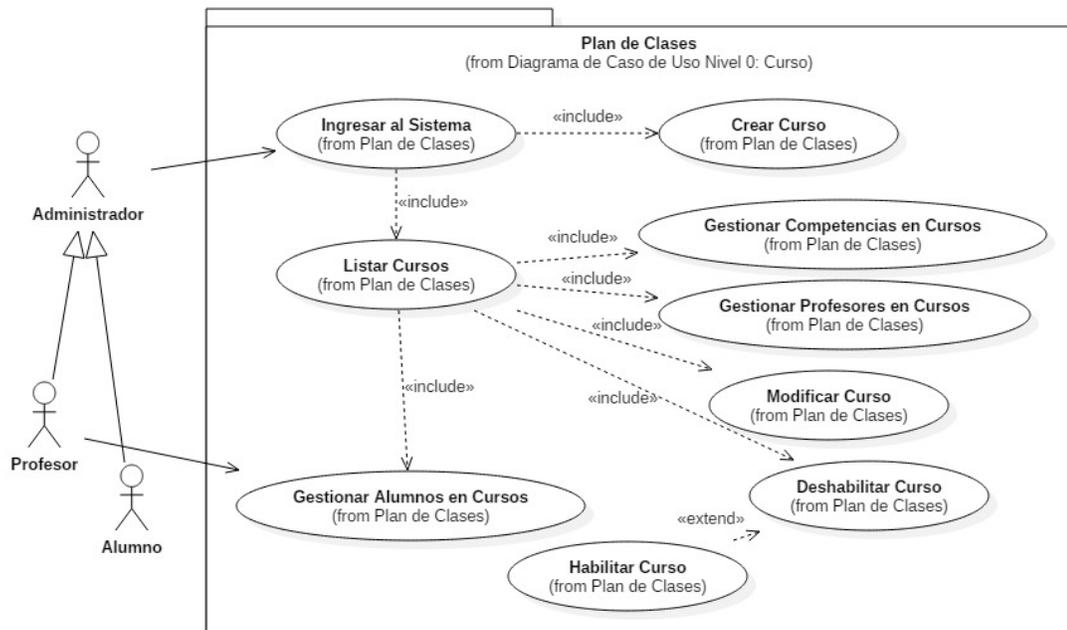


Figura B.8: Diagrama de Casos de Uso Nivel 0: Curso

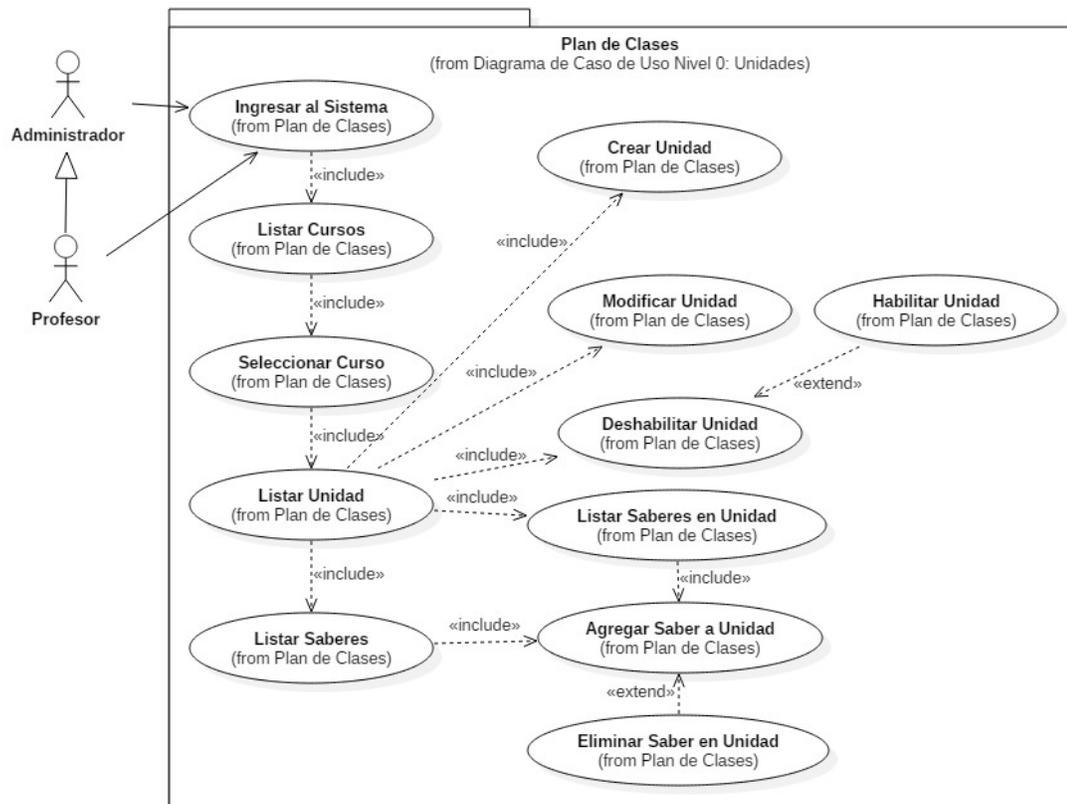


Figura B.9: Diagrama de Casos de Uso Nivel 0: Unidad

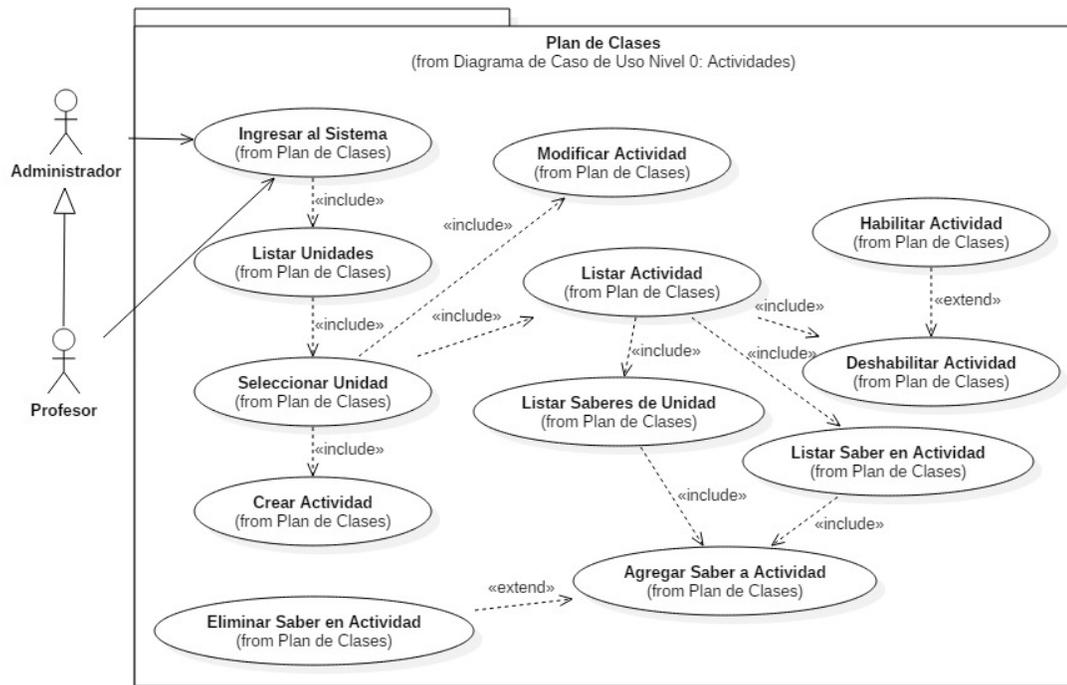


Figura B.10: Diagrama de Casos de Uso Nivel 0: Actividades

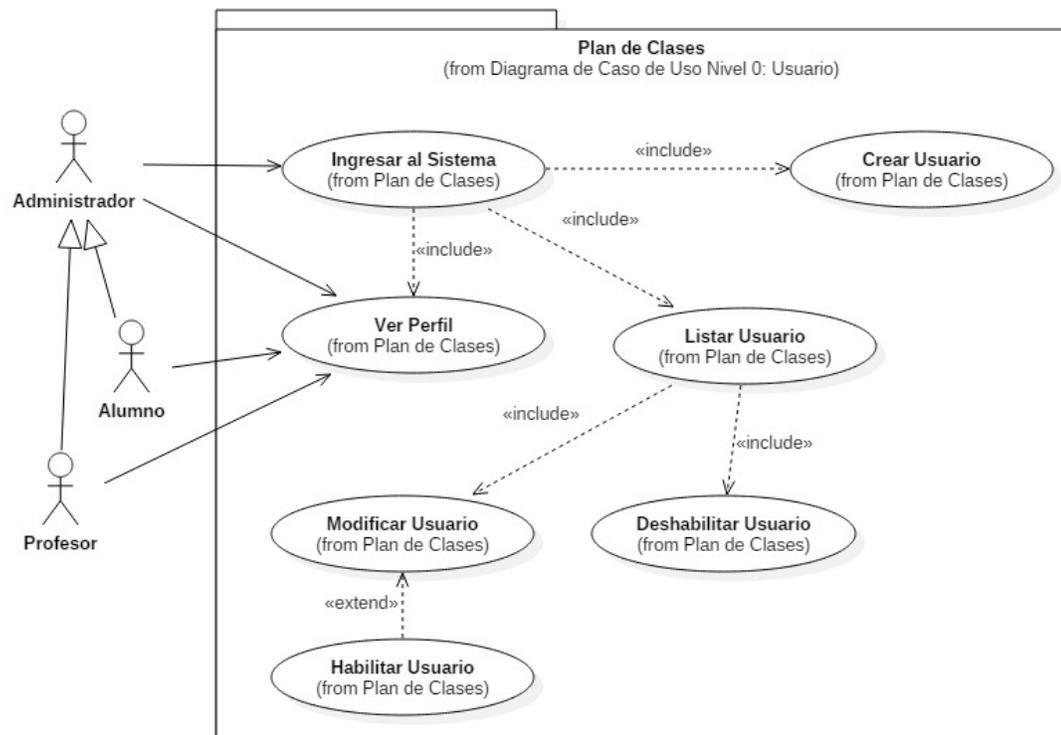


Figura B.11: Diagrama de Casos de Uso Nivel 0: Usuario

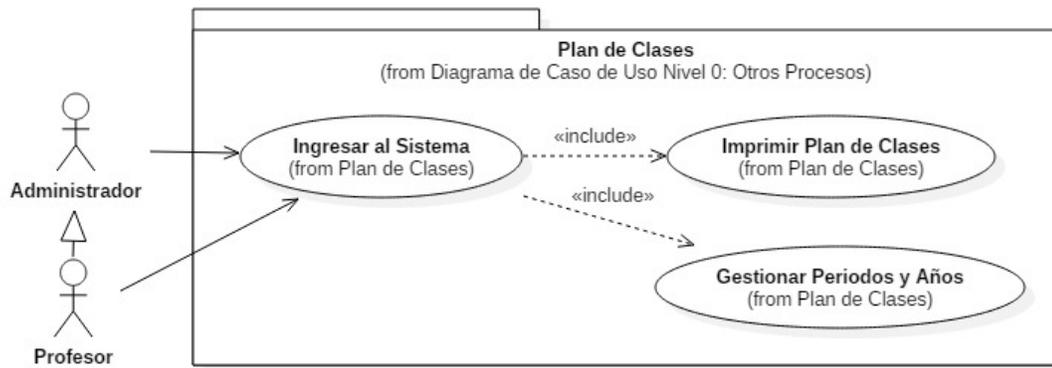


Figura B.12: Diagrama de Casos de Uso Nivel 0: Otros Procesos

C. Desarrollo

En el anexo se presentarán los diagramas e imágenes que facilitan el entender el desarrollo del software, mediante imágenes.

C.1. Iteraciones: Primer Prototipo

A Web Page

Plan de Clases Mis Cursos Competencias Aprendizajes Saberes Ayuda Iniciar Sesión

Iniciar Sesión

Nombre de Usuario

Contraseña

[¿Olvidó su Contraseña?](#)
[Registrarse](#)

@2018 - Universidad de Talca - 1 poniente 1141, Talca - Chile | [¿Cómo llegar?](#)   

The image shows a wireframe of a login page. At the top, there is a browser window header with navigation icons and a search bar. Below this is a horizontal menu with items: Plan de Clases, Mis Cursos, Competencias, Aprendizajes, Saberes, Ayuda, and Iniciar Sesión. The main content area features the title 'Iniciar Sesión' in a large font. Below the title are three vertically stacked input fields: 'Nombre de Usuario', 'Contraseña', and a button labeled 'Iniciar Sesión'. Underneath the button are two links: '¿Olvidó su Contraseña?' and 'Registrarse'. At the bottom, a red footer bar contains the text '@2018 - Universidad de Talca - 1 poniente 1141, Talca - Chile | ¿Cómo llegar?' followed by social media icons for Facebook, Twitter, and Instagram.

Figura C.1: Primer prototipo: iniciar sesión

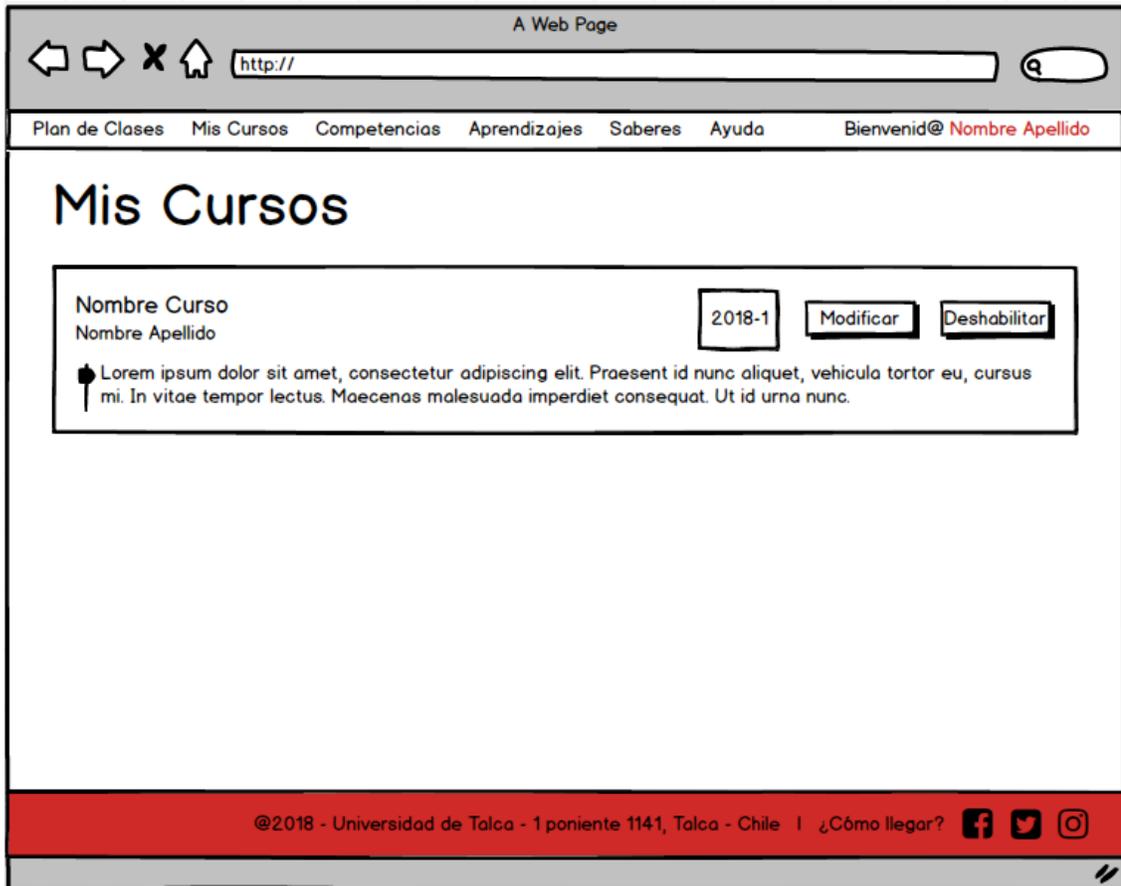


Figura C.2: Primer prototipo: mis cursos

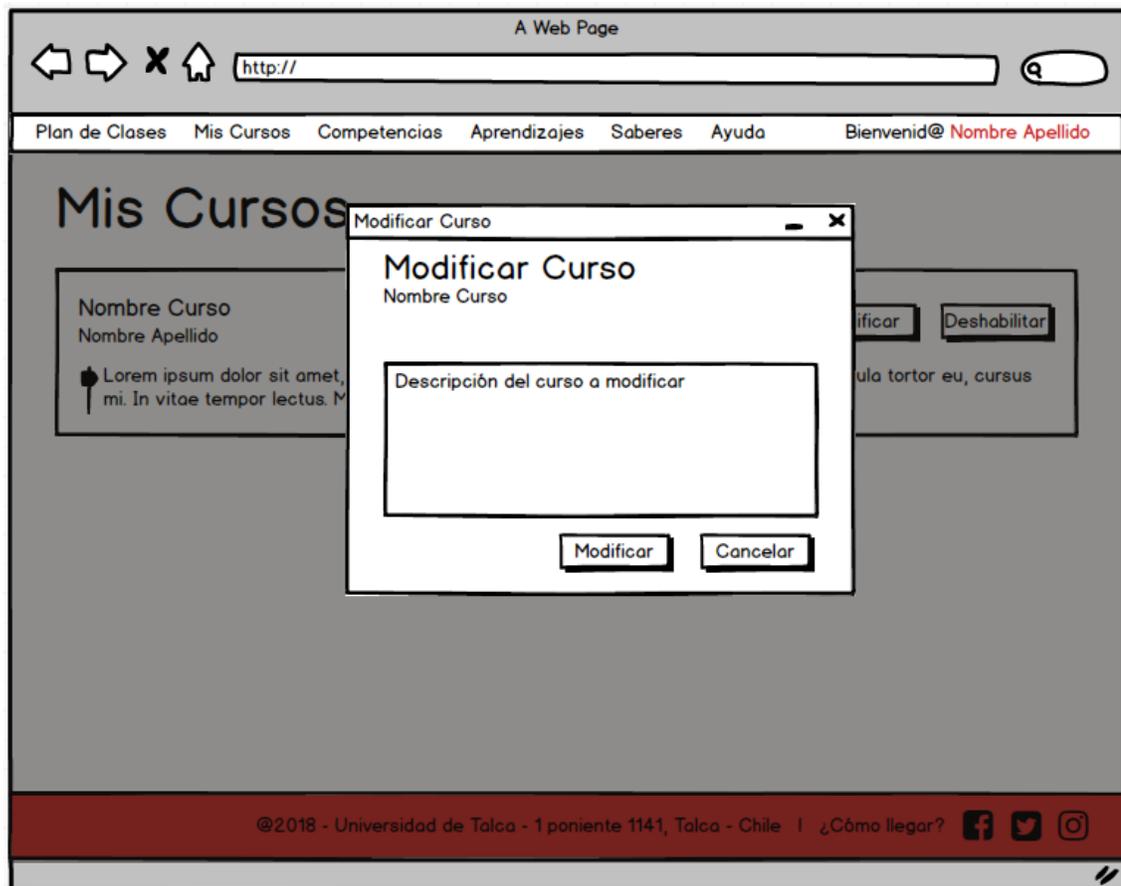


Figura C.3: Primer prototipo: modificar cursos

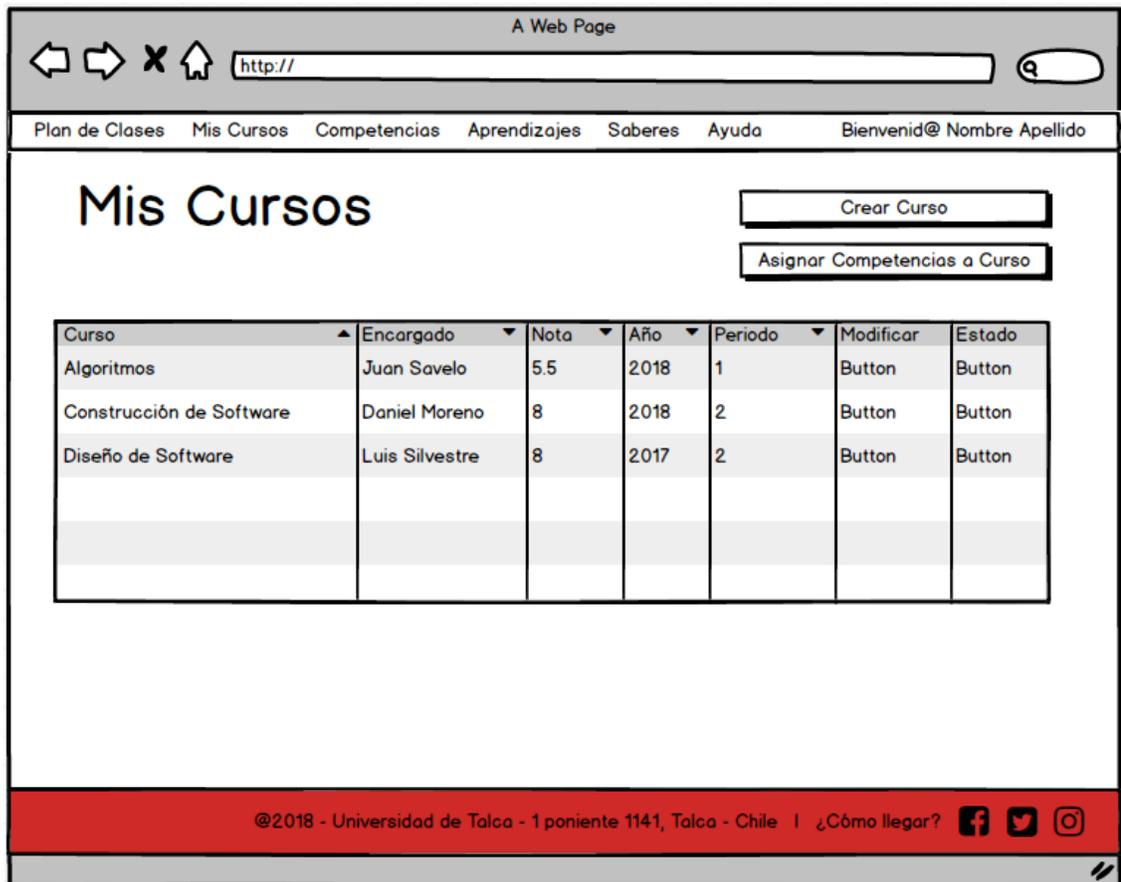


Figura C.4: Primer prototipo: lista de cursos

A Web Page

Plan de Clases Mis Cursos Competencias Aprendizajes Saberes Ayuda Bienvenid@ Nombre Apellido

Asignar Competencias a Cursos

Nombre Cursos

Nombre Curso

Seleccionar	Nombre	Descripción
<input type="checkbox"/>	Competencia 1	Descripción competencia 1
<input checked="" type="checkbox"/>	Competencia 2	Descripción competencia 2
<input type="checkbox"/>	Competencia 3	Descripción competencia 3
<input type="checkbox"/>	Competencia 4	Descripción competencia 4
<input type="checkbox"/>	Competencia 5	Descripción competencia 5
<input type="checkbox"/>	Competencia 6	Descripción competencia 6

Aceptar Volver

@2018 - Universidad de Talca - 1 poniente 1141, Talca - Chile | ¿Cómo llegar?   

Figura C.5: Primer prototipo: asignar competencias a cursos

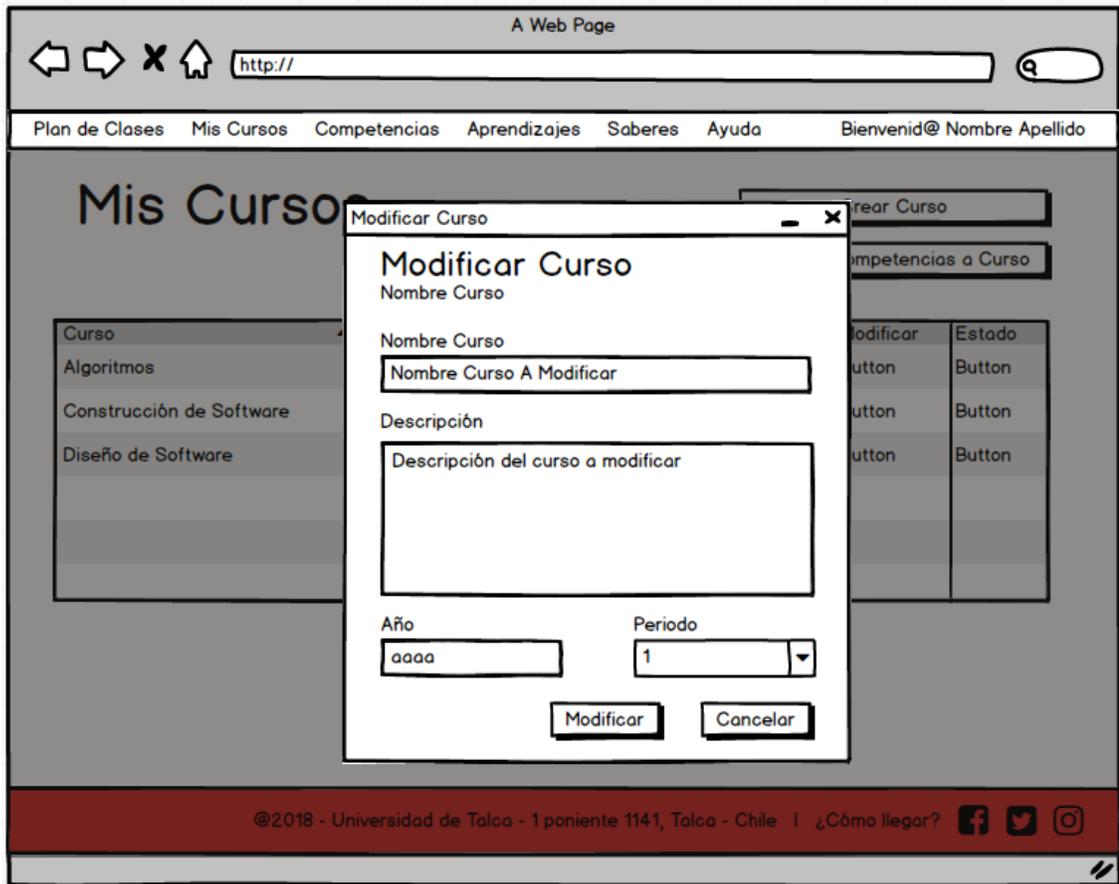


Figura C.6: Primer prototipo: modificar cursos, administrador

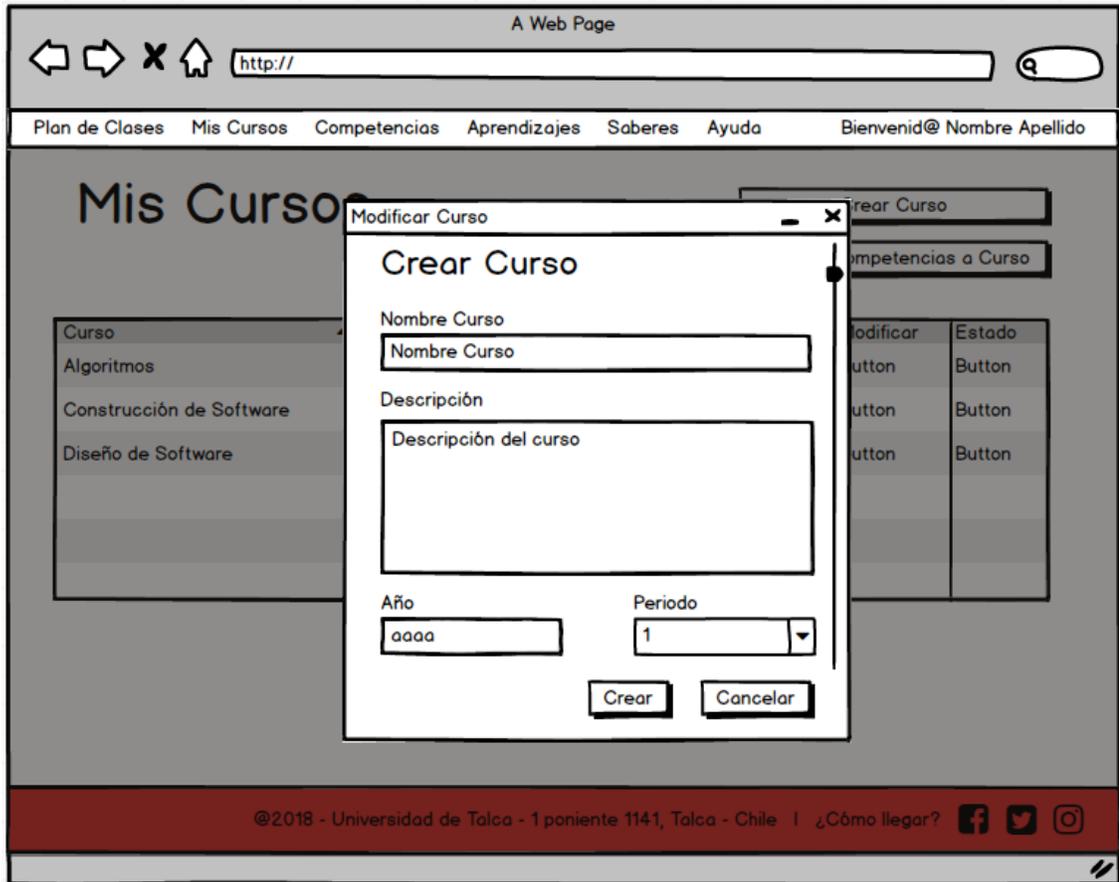


Figura C.7: Primer prototipo: nuevo curso

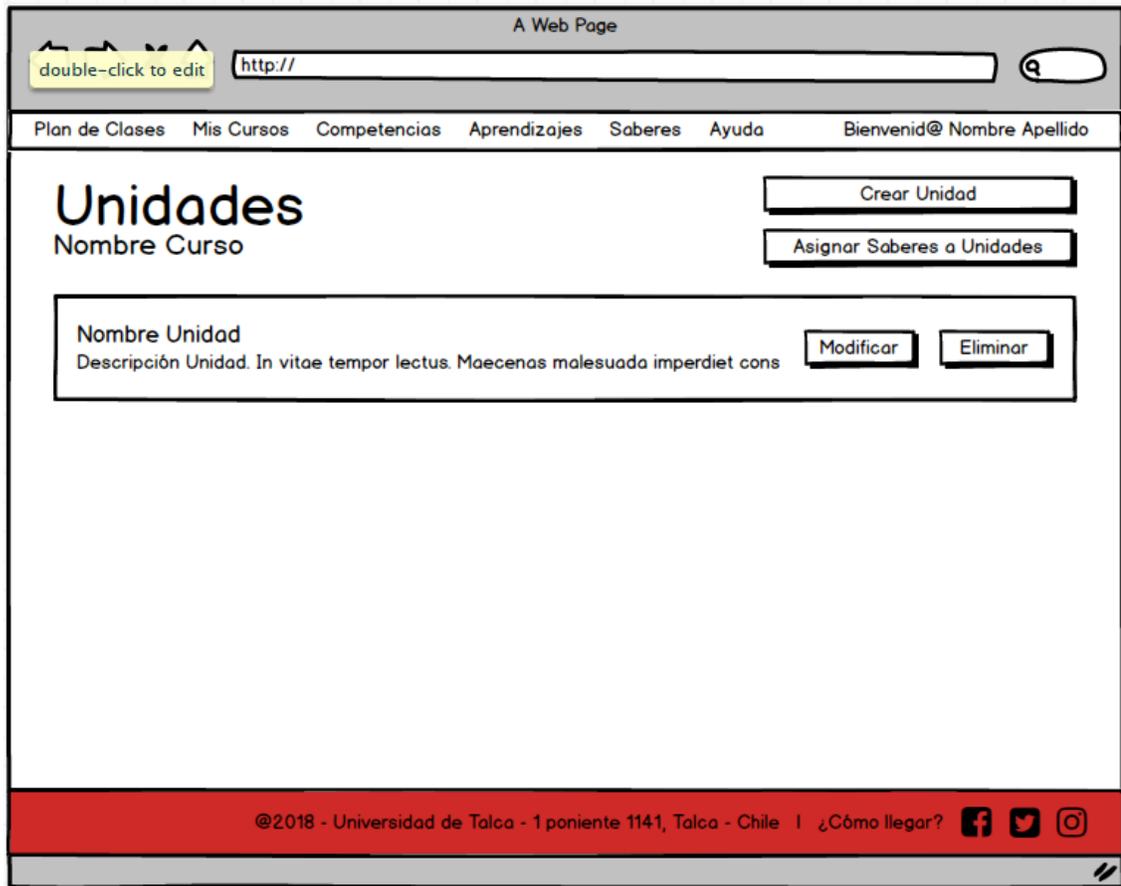


Figura C.8: Primer prototipo: unidades

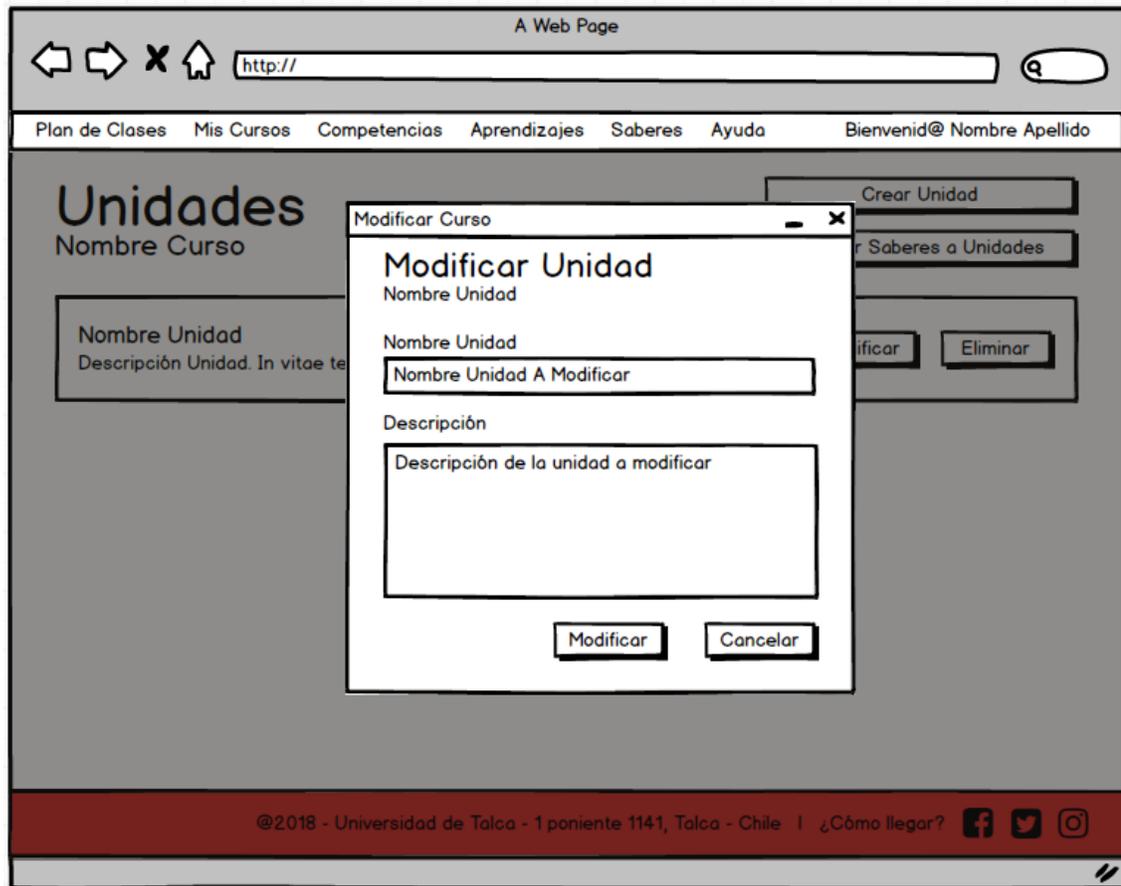


Figura C.9: Primer prototipo: modificar unidad

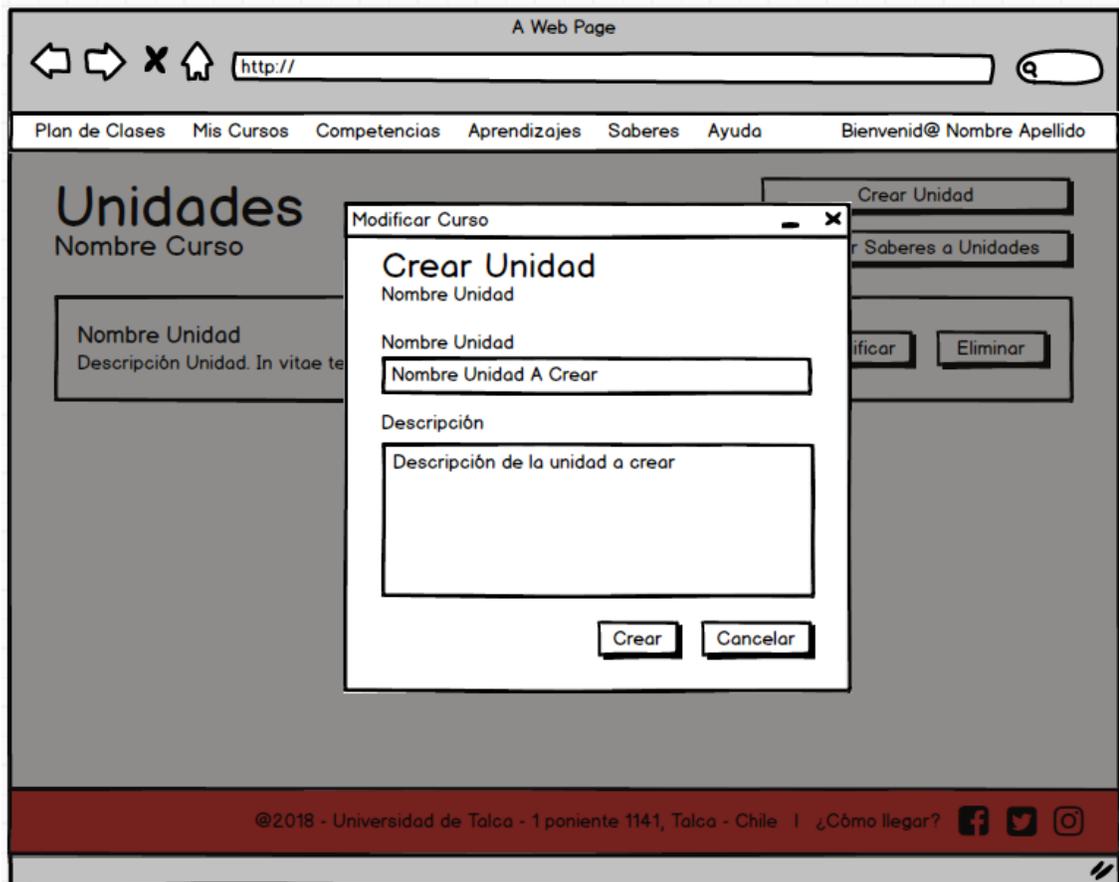


Figura C.10: Primer prototipo: crear unidad

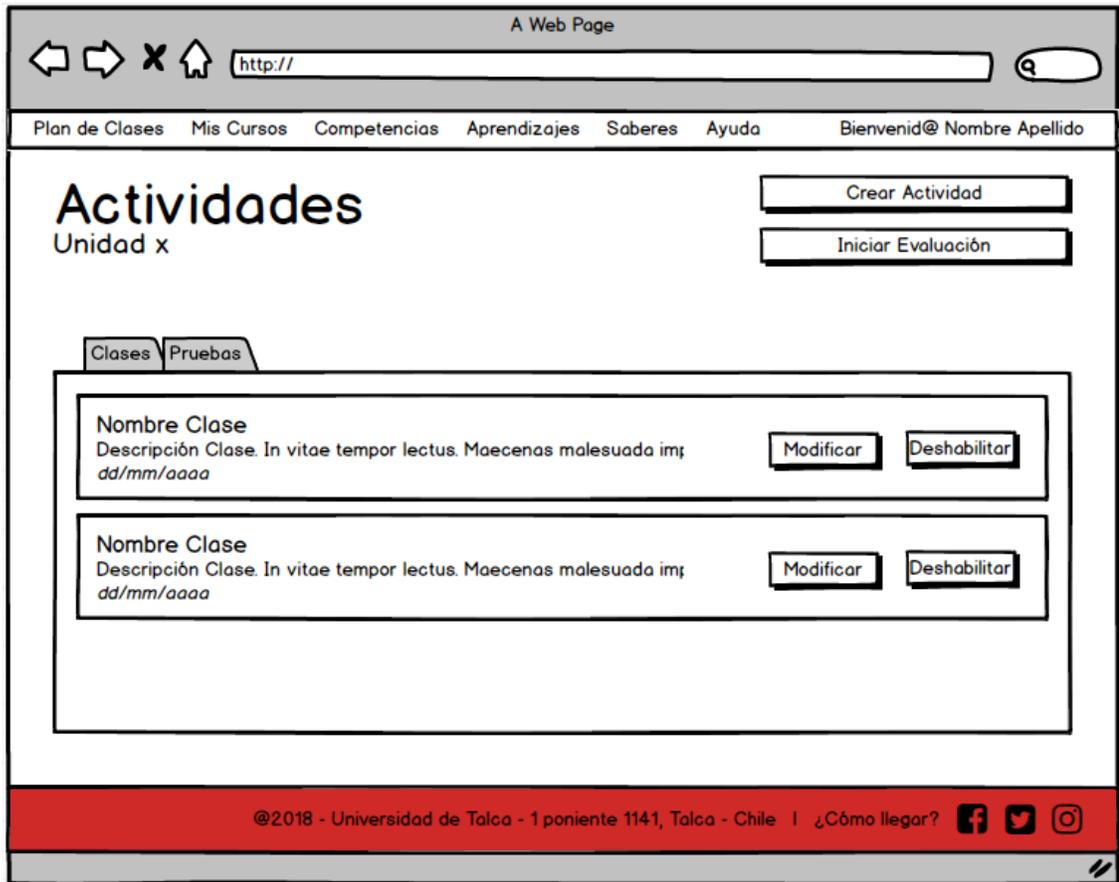


Figura C.11: Primer prototipo: actividades

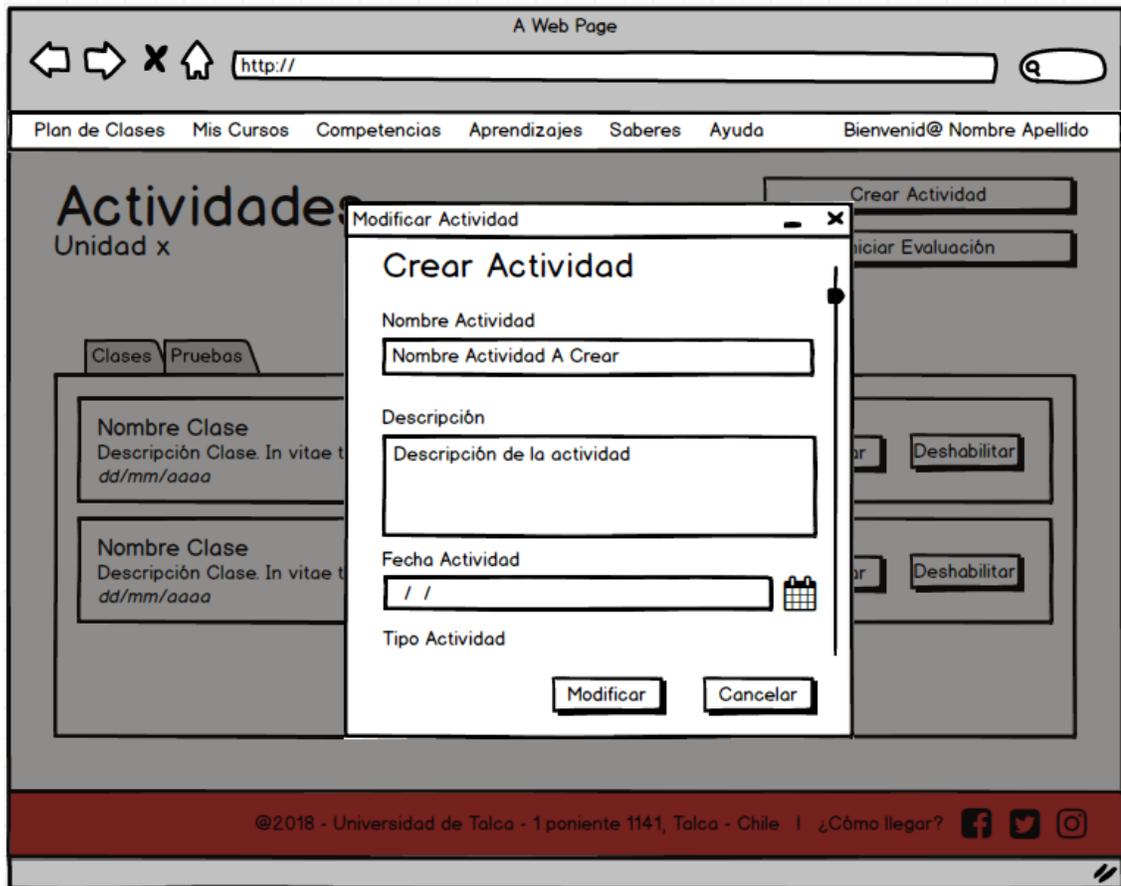


Figura C.12: Primer prototipo: crear actividad

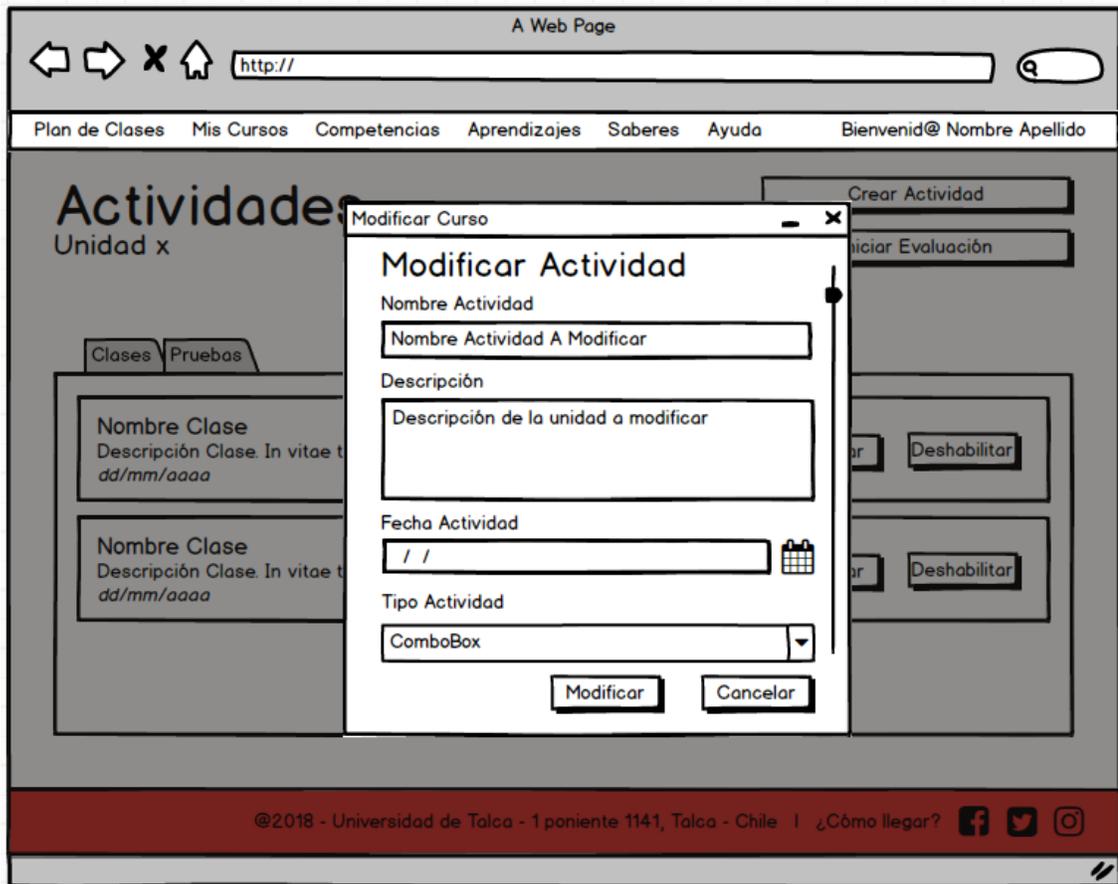


Figura C.13: Primer prototipo: modificar actividad



Figura C.14: Primer prototipo: lista de competencias

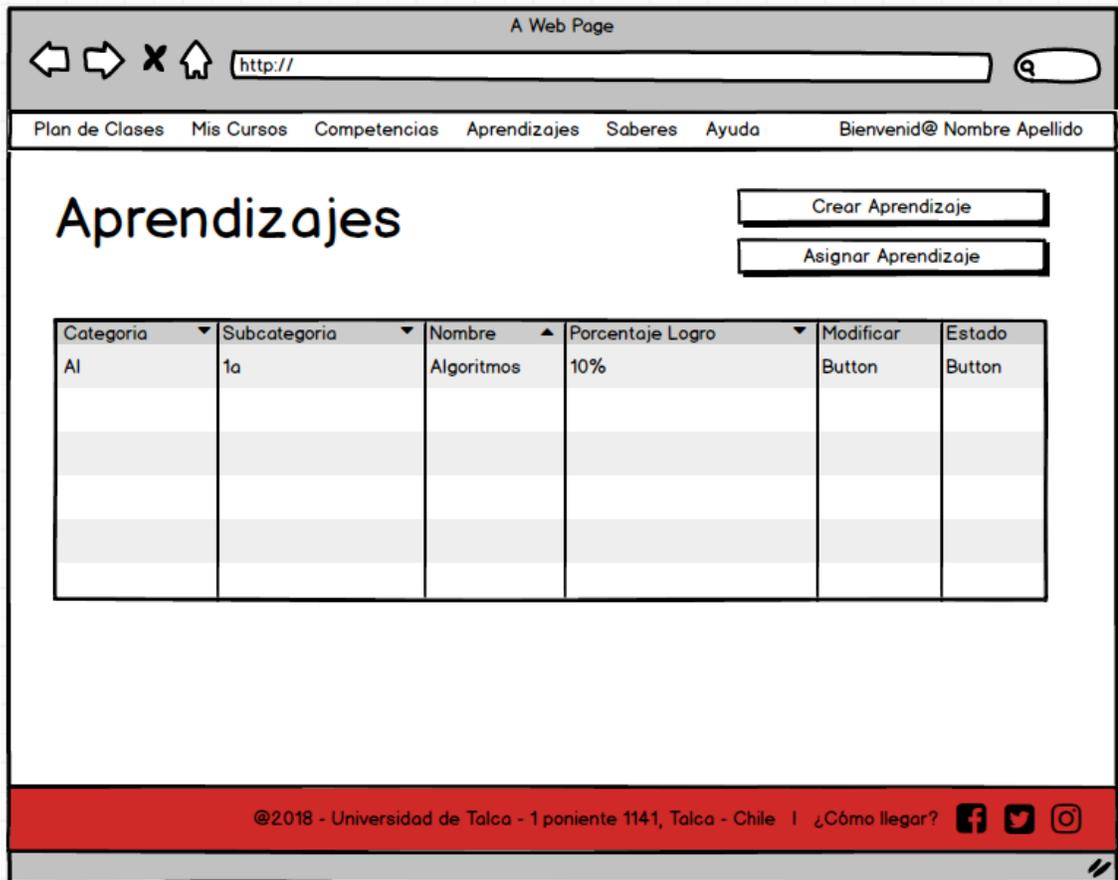


Figura C.15: Primer prototipo: lista de aprendizajes

The image shows a web browser window with the title 'A Web Page'. The address bar contains 'http://'. The navigation menu includes 'Plan de Clases', 'Mis Cursos', 'Competencias', 'Aprendizajes', 'Saberes', 'Ayuda', and 'Bienvenid@ Nombre Apellido'. The main heading is 'Asignar Aprendizajes a Competencias'. Below the heading is a dropdown menu for 'Nombre Competencia' with the text 'Nombre Competencia' and a downward arrow. A table with four columns is displayed: 'Seleccionar', 'Categoria', 'Subcategoria', and 'Nombre'. The table contains four rows of data, with the second and fourth rows having checked checkboxes. Below the table are two buttons: 'Aceptar' and 'Volver'. The footer contains the text '@2018 - Universidad de Talca - 1 poniente 1141, Talca - Chile | ¿Cómo llegar?' and social media icons for Facebook, Twitter, and Instagram.

Nombre Competencia

Seleccionar	Categoria	Subcategoria	Nombre
<input type="checkbox"/>	Categoria	Subcategoria	Aprendizaje 1
<input checked="" type="checkbox"/>	Categoria	Subcategoria	Aprendizaje 2
<input type="checkbox"/>	Categoria	Subcategoria	Aprendizaje 3
<input checked="" type="checkbox"/>	Categoria	Subcategoria	Aprendizaje 4

Aceptar Volver

@2018 - Universidad de Talca - 1 poniente 1141, Talca - Chile | ¿Cómo llegar?   

Figura C.16: Primer prototipo: asignar aprendizajes a competencias

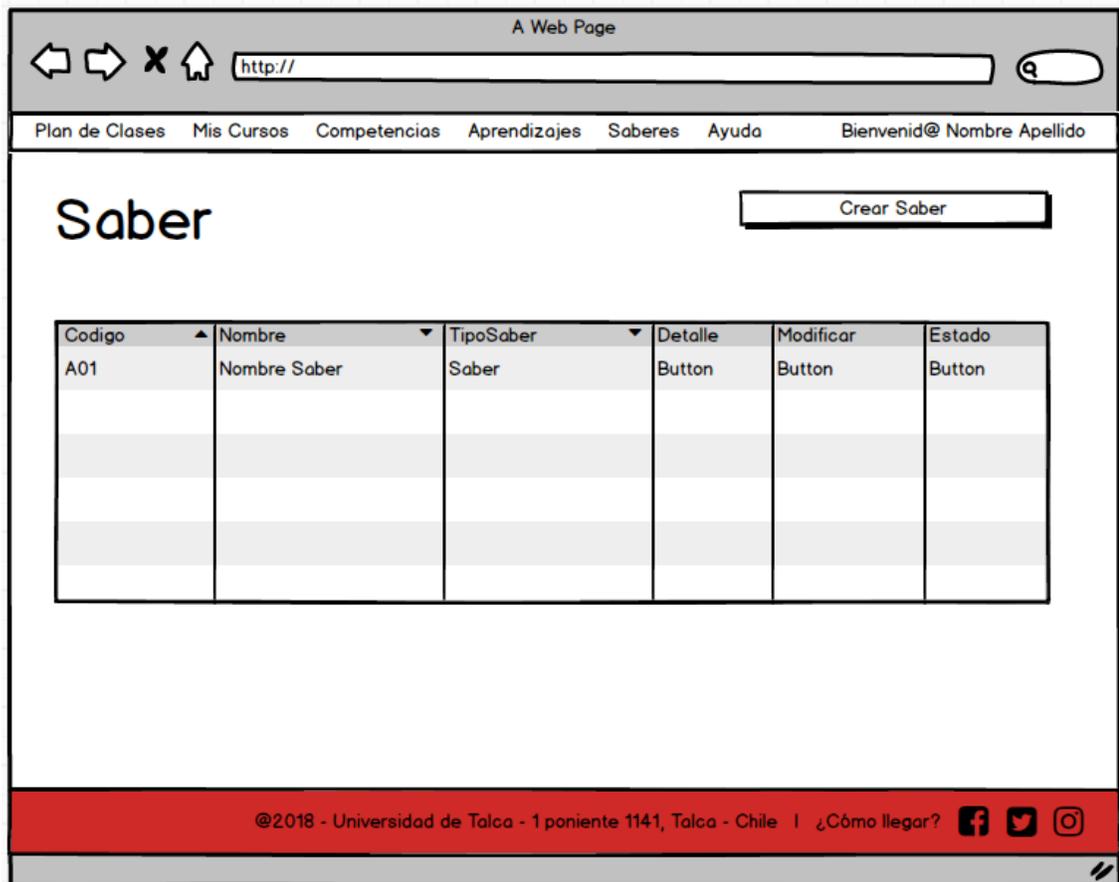


Figura C.17: Primer prototipo: lista de saberes

C.2. Iteraciones: Segundo Prototipo

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Iniciar Sesión

Nombre de Usuario

Ingrese Rut Sin digito Verificador ni puntos. Ej: 9999999 en vez de 9.999.999-9

Contraseña

Password

Submit

Figura C.18: Segundo prototipo: iniciar sesión

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Cursos Crear Curso

Nombre	Horas Presenciales	Horas Autónomas	Más	Estado
Programación Avanzada	6	3	Ver Al Detalle	Deshabilitar Editar Asignar Competencias
Algoritmos	6	3	Ver Al Detalle	Deshabilitar Editar Asignar Competencias
Modelos Discretos	6	3	Ver Al Detalle	Deshabilitar Editar Asignar Competencias
Construcción de Software	1	9	Ver Al Detalle	Deshabilitar Editar Asignar Competencias
Taller de problemas de Ingeniería	1	7	Ver Al Detalle	Deshabilitar Editar Asignar Competencias

Figura C.19: Segundo prototipo: ver cursos

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Usuarios Registrados [Crear Usuario](#)

Rut	Nombre	Correo Electrónico	Matrícula	Perfil	Estado	Modificar
13.089.529-8	Rodrigo Paredes	raparede@uchile.cl	0	Ver Perfil	Deshabilitar	Editar
17.014.870-3	Nicolas Rojas	nrojas14@alumnos.utalca.cl	2014407535	Ver Perfil	Deshabilitar	Editar
17.820.883-7	Daniela Paredes	dparedes09@alumnos.utalca.cl	2009407826	Ver Perfil	Deshabilitar	Editar
17.824.523-6	Yorch Sepulveda	yosepulveda11@alumnos.utalca.cl	0	Ver Perfil	Deshabilitar	Editar
19.043.138-10	Gregory Sepulveda	tze05103@cndps.com	2019407064	Ver Perfil	Deshabilitar	Editar
19.254.373-8	Vicente Rojas	vrojas16@alumnos.utalca.cl	0	Ver Perfil	Deshabilitar	Editar
9.244.120-2	Gloria Rodriguez	gloria.rv@hotmail.es	0	Ver Perfil	Deshabilitar	Editar
9.999.999-3	Administrador	amx53717@cndps.com	0	Ver Perfil	Deshabilitar	Editar

Figura C.20: Segundo prototipo: ver usuarios

The image shows a web application interface with a modal window titled "Crear un Usuario". The background is a "Usuarios Registrados" page with a table of users and a "Modificar" section with "Editar" buttons. The modal form contains the following fields:

- Rut del Usuario:
- Nombre de Usuario:
- Correo Electrónico:
- Tipo de Usuario:
- Matrícula:
- Contraseña:
- Contraseña:

At the bottom of the modal are two buttons: "Crear Usuario" (green) and "Cancelar" (grey).

Figura C.21: Segundo prototipo: crear un usuario

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!			
Competencias		Agregar Competencia	
Nombre	Descripción	Detalle	Estado
Dominar el cuerpo de conocimiento disciplinar de ciencias de la computación conforme a estándares reconocidos nacional e internacionalmente para su correcto desempeño profesional.	Implica que el graduado ha logrado dominar el los distintos componentes del núcleo de conocimientos de ciencias de la computación sugeridos por la ACM Computer Curricula para Ciencias de Computación.	Ver	Deshabilitar
Utilizar el cuerpo de conocimiento disciplinar en el quehacer profesional para la solución de problemas específicos de su ámbito laboral, considerando múltiples plataformas de desarrollo computacional	-	Ver	Deshabilitar
Utilizar las ciencias de la computación como herramienta tanto para el desarrollo de la misma como de otras ciencias y de actividades productivas en general, trabajando en equipos multidisciplinarios e integrando conocimiento específico de otras disciplinas.	-	Ver	Deshabilitar
Integrar equipos de trabajo desarrollando habilidades sociales y de autogestión, para potenciar la capacidad de crear valor desde	Implica desarrollar en el estudiante la capacidad de interactuar eficientemente con otros en virtud de un objetivo académico/profesional, utilizando para ello recursos, contribuyendo así	Ver	Deshabilitar

Figura C.22: Segundo prototipo: ver competencias

Ver Competencia

[Editar Competencia](#)[Competencia](#) / [Ver al Detalle](#)

Competencia Disciplinar

Dominar el cuerpo de conocimiento disciplinar de ciencias de la computación conforme a estándares reconocidos nacional e internacionalmente para su correcto desempeño profesional.

Implicancias (Descripción de la Competencia)

Implica que el graduado ha logrado dominar el los distintos componentes del núcleo de conocimientos de ciencias de la computación sugeridos por la ACM Computer Curricula para Ciencias de Computación.

Nivel de Dominio de la Competencia

Avanzado

Básico	Comprende los Fundamentos de Sistemas, Desarrollo de Software y Algoritmos conforme a estándares reconocidos para su correcto desempeño profesional.
Intermedio	Comprende además las Estructuras Discretas, Ingeniería de Software, Interfaces, Bases de datos, Lenguajes de programación y Arquitectura conforme a estándares reconocidos para su correcto desempeño profesional.
Avanzado	Comprender además la Computación Paralela, Visualización, Redes, Seguridad, Sistemas Inteligentes y Ciencias de la Computación en general, conforme a estándares reconocidos para su correcto desempeño profesional.

Tiempo de Desarrollo de la Competencia

El estudiantes inicia del desarrollo de esta competencia desde el primer semestre, y se desarrolla a lo largo de la carrera, fundamentalmente durante los siete primeros semestres.

Figura C.23: Segundo prototipo: ver una competencia al detalle

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Cursos

Nombre

- Programación Avanzada
- Algoritmos
- Modelos Discretos
- Construcción de Software

Taller de problemas de Ingeniería de 1 7

Ver Al Detalle Deshabilitar Editar Asignar Competencias

Crear un Curso

Nombre del Curso:

Descripción del Curso:

Horas Presenciales: Horas Autónomas:

Crear Curso Cancelar

Figura C.24: Segundo prototipo: crear un curso

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Aprendizajes

Agregar Aprendizaje

Asignar Aprendizajes

Categoría	Subcategoría	Nombre	Estado
AL	Análisis Básico de Algoritmos	Conoce tiempo de ejecución de algoritmos	Deshabilitar Editar
AL	Análisis Básico de Algoritmos	Realiza procedimientos básicos de análisis de algoritmos	Deshabilitar Editar
AL	Estrategias de Diseño de Algoritmos	Aplica estrategias básicas de diseño de Algoritmos	Deshabilitar Editar
AL	Estructuras de Datos y Algoritmos Fundamentales	Usa algoritmos fundamentales	Deshabilitar Editar
AL	Estructuras de Datos y Algoritmos Fundamentales	Entiende Estructuras de datos fundamentales	Deshabilitar Editar
AL	Computabilidad y Complejidad de Autómatas Básicos	Maneja autómatas básicos	Deshabilitar Editar
AL	Computabilidad y Complejidad de Autómatas Básicos	Entiende el concepto de "complejidad" asociado a la definición de problemas	Deshabilitar Editar
AR	Lógica Digital y Sistemas Digitales	Conoce los Fundamentos de sistemas digitales	Deshabilitar Editar

Figura C.25: Segundo prototipo: ver aprendizajes

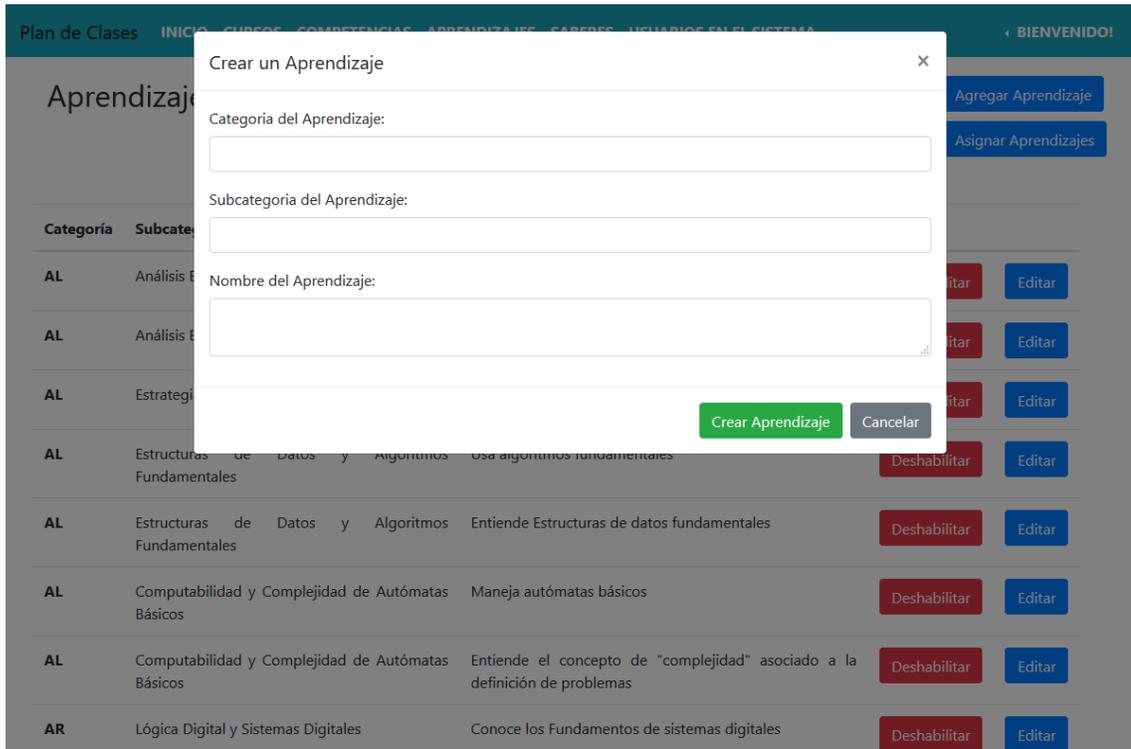


Figura C.26: Segundo prototipo: crear un aprendizaje

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Asignar Competencias a Programación Avanzada

[Cursos](#) / Programación Avanzada

Asignar Competencias

Competencias asociadas

#	Nombre	Descripción	
1	Dominar el cuerpo de conocimiento disciplinar de ciencias de la computación conforme a estándares reconocidos nacional e internacionalmente para su correcto desempeño profesional.	Implica que el graduado ha logrado dominar el los distintos componentes del núcleo de conocimientos de ciencias de la computación sugeridos por la ACM Computer Curricula para Ciencias de Computación.	Eliminar Competencia
2	Utilizar el cuerpo de conocimiento disciplinar en el quehacer profesional para la solución de problemas específicos de su ámbito laboral, considerando	-	Eliminar Competencia

Figura C.27: Segundo prototipo: asignar competencias a cursos

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Asignar Aprendizajes a Competencias

[Aprendizajes](#) / Asignar Aprendizajes a Competencias

Competencias asociadas

Categoría	Subcategoría	Nombre	Competencia Asignada	
AL	Análisis Básico de Algoritmos	Conoce tiempo de ejecución de algoritmos	Dominar el cuerpo de conocimiento disciplinar de ciencias de la computación conforme a estándares reconocidos nacional e internacionalmente para su correcto desempeño profesional.	Eliminar Competencia
AL	Análisis Básico de Algoritmos	Realiza procedimientos básicos de análisis de algoritmos	Actuar con sentido ético y responsabilidad social en el ejercicio profesional con criterios ciudadanos para el desarrollo sustentable del entorno.	Eliminar Competencia
AL	Estrategias de Diseño de Algoritmos	Aplica estrategias básicas de diseño de Algoritmos		Asignar Competencia

Figura C.28: Segundo prototipo: asignar aprendizaje a competencias

Aprendizaje	Saber	Saber Hacer	Saber Ser	Estado	Modificar
1.1	Conoce la diferencia entre el peor, promedio y mejor tiempo de ejecución de un algoritmo			Deshabilitar	Editar
1.7	Es capaz de dar ejemplos que ilustren la compensación que debe existir entre tiempo y espacio para los algoritmos			Deshabilitar	Editar
1.4	Reproduce la definición formal de la notación O-grande			Deshabilitar	Editar
1.5	Lista y contrasta clases de complejidad estándar			Deshabilitar	Editar
1.3		Determina de manera informal la complejidad de tiempo y espacio de algoritmos simples		Deshabilitar	Editar

Figura C.29: Segundo prototipo: ver saberes

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Saberes

Agregar Saber Asignar Saberes

Aprendizaje	Saber	Modificar
1.1	Conocer y promover la ejecución de algoritmos	Editar
1.7	Es capaz de ilustrar la existencia de tiempo y espacio para los algoritmos	Editar
1.4	Reproduce la definición formal de la notación O-grande	Deshabilitar Editar
1.5	Lista y contrasta clases de complejidad estándar	Deshabilitar Editar
1.3	Determina de manera informal la complejidad de tiempo y espacio de algoritmos simples	Deshabilitar Editar

Crear un Saber

Código del Saber:

Nombre del Saber:

Tipo de Saber:

Crear Saber Cancelar

Figura C.30: Segundo prototipo: crear un saber

Plan de Clases INICIO CURSOS COMPETENCIAS APRENDIZAJES SABERES USUARIOS EN EL SISTEMA BIENVENIDO!

Asignar Saberes a Aprendizajes

[Saberes](#) / Asignar Saberes a Aprendizajes

Saberes asociados

Codigo	Nombre	Aprendizaje Asignado	
1.1	Conoce la diferencia entre el peor, promedio y mejor tiempo de ejecución de un algoritmo	AL	Eliminar Aprendizaje
1.7	Es capaz de dar ejemplos que ilustren la compensación que debe existir entre tiempo y espacio para los algoritmos	AR	Eliminar Aprendizaje
1.4	Reproduce la definición formal de la notación O-grande	AR	Eliminar Aprendizaje
1.5	Lista y contrasta clases de complejidad estándar	AL	Eliminar Aprendizaje

Figura C.31: Segundo prototipo: asignar saberes a aprendizajes

D. Pruebas y Resultados

En el anexo se presentarán los diagramas e imágenes que facilitan el entender el resultado de cada prueba mediante tablas y capturas de pantalla sobre el sistema.

D.1. Pruebas de caja negra, Tablas

Cuadro D.1: Caja Negra - Iniciar sesión

CN01: Iniciar sesión		
Requisitos de sistema	RS01	
Precondiciones	-	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none">▪ Usuario▪ Contraseña	<ul style="list-style-type: none">▪ Sesión registrada en la aplicación.▪ Mensaje de éxito.▪ Excepción: Notificación de rechazo: “No pudo iniciar sesión”	<ul style="list-style-type: none">▪ Sesión registrada con éxito.▪ Mensaje de éxito.
Observaciones	-	

Cuadro D.2: Caja Negra - Crear un curso

CN02: Crear un curso		
Requisitos de sistema	RS12	
Precondiciones	CN01	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Nombre del curso ▪ Descripción del curso ▪ Horas presenciales ▪ Horas autónomas 	<ul style="list-style-type: none"> ▪ Curso guardado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso guardado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.3: Caja Negra - Modificar curso

CN03: Modificar curso		
Requisitos de sistema	RS24, RS13	
Precondiciones	CN01, CN02	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Nombre del curso ▪ Descripción del curso ▪ Horas presenciales ▪ Horas autónomas 	<ul style="list-style-type: none"> ▪ Curso actualizado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso actualizado exitosamente ▪ Mensaje de éxito
Observaciones	a	

Cuadro D.4: Caja Negra - Deshabilitar curso

CN04: Deshabilitar curso		
Requisitos de sistema	RS24, RS14	
Precondiciones	CN01, CN02	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Estado 	<ul style="list-style-type: none"> ▪ Curso deshabilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.5: Caja Negra - Habilitar curso

CN05: Habilitar curso		
Requisitos de sistema	RS24, RS15	
Precondiciones	CN01, CN04	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Estado 	<ul style="list-style-type: none"> ▪ Curso habilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.6: Caja Negra - Crear curso por periodo

CN06: Crear curso por periodo		
Requisitos de sistema	RS12	
Precondiciones	CN01, CN02	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Año ▪ Periodo ▪ Id del curso 	<ul style="list-style-type: none"> ▪ Periodo de un curso guardado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso por periodo guardado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.7: Caja Negra - Modificar curso por periodo

CN07: Modificar curso por periodo		
Requisitos de sistema	RS24, RS13	
Precondiciones	CN01, CN06	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Año ▪ Periodo ▪ Id del curso 	<ul style="list-style-type: none"> ▪ Periodo de un curso actualizado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso por periodo actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.8: Caja Negra - Deshabilitar curso por periodo

CN08: Deshabilitar curso por periodo.		
Requisitos de sistema	RS24, RS14	
Precondiciones	CN01, CN06, CN08	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del curso por periodo ▪ Estado 	<ul style="list-style-type: none"> ▪ Periodo de un curso deshabilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso por periodo actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.9: Caja Negra - Habilitar curso por periodo

CN09: Habilitar curso por periodo		
Requisitos de sistema	RS24, RS15	
Precondiciones	CN01, CN06	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del curso por periodo ▪ Estado 	<ul style="list-style-type: none"> ▪ Periodo de un curso habilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Curso por periodo actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.10: Caja Negra - Crear unidad

CN10: Crear unidad		
Requisitos de sistema	RS25	
Precondiciones	CN01, CN06	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso por periodo ▪ Nombre ▪ Descripción 	<ul style="list-style-type: none"> ▪ Unidad guardada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Unidad guardada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.11: Caja Negra - Modificar unidad

CN11: Modificar unidad		
Requisitos de sistema	RS31, RS26	
Precondiciones	CN01, CN06	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso por periodo ▪ Nombre ▪ Descripción 	<ul style="list-style-type: none"> ▪ Unidad actualizada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Unidad actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.12: Caja Negra - Deshabilitar unidad

CN12: Deshabilitar unidad		
Requisitos de sistema	RS31, RS28	
Precondiciones	CN01, CN06	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la unidad ▪ Estado 	<ul style="list-style-type: none"> ▪ Unidad deshabilitada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Unidad actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.13: Caja Negra - Habilitar unidad

CN13: Habilitar unidad		
Requisitos de sistema	RS31, RS27	
Precondiciones	CN01, CN06, CN12	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la unidad ▪ Estado 	<ul style="list-style-type: none"> ▪ Unidad habilitada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Unidad actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.14: Caja Negra - Crear actividad

CN14: Crear actividad		
Requisitos de sistema	RS32	
Precondiciones	CN01, CN10	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la unidad ▪ Nombre ▪ Descripción ▪ Fecha de inicio ▪ Fecha de fin ▪ Tipo de la actividad ▪ Tiempo presencial ▪ Tiempo no presencial 	<ul style="list-style-type: none"> ▪ Actividad guardada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Actividad guardada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.15: Caja Negra - Modificar actividad

CN15: Modificar actividad		
Requisitos de sistema	RS38, RS33	
Precondiciones	CN01, CN10	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la unidad ▪ Nombre ▪ Descripción ▪ Fecha de inicio ▪ Fecha de fin ▪ Tipo de la actividad ▪ Tiempo presencial ▪ Tiempo no presencial 	<ul style="list-style-type: none"> ▪ Actividad actualizada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Actividad actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.16: Caja Negra - Deshabilitar actividad

CN16: Deshabilitar actividad		
Requisitos de sistema	RS38, RS35	
Precondiciones	CN01, CN10	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la actividad ▪ Estado 	<ul style="list-style-type: none"> ▪ Actividad deshabilitada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Actividad actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.17: Caja Negra - Habilitar actividad

CN17: Habilitar actividad		
Requisitos de sistema	RS38, RS34	
Precondiciones	CN01, CN10, CN16	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la actividad ▪ Estado 	<ul style="list-style-type: none"> ▪ Actividad habilitada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Actividad actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.18: Caja Negra - Crear competencia

CN18: Crear competencia		
Requisitos de sistema	RS39	
Precondiciones	CN01	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Nivel de dominio ▪ Básico ▪ Intermedio ▪ Avanzado ▪ Tiempo de desarrollo 	<ul style="list-style-type: none"> ▪ Competencia guardada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Competencia guardada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.19: Caja Negra - Modificar competencia

CN19: Modificar competencia		
Requisitos de sistema	RS43, RS40	
Precondiciones	CN01, CN18	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Nombre ▪ Descripción ▪ Nivel de dominio ▪ Básico ▪ Intermedio ▪ Avanzado ▪ Tiempo de desarrollo 	<ul style="list-style-type: none"> ▪ Competencia actualizada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Competencia actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.20: Caja Negra - Deshabilitar competencia

CN20: Deshabilitar competencia		
Requisitos de sistema	RS43, RS42	
Precondiciones	CN01, CN18	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la competencia ▪ Estado 	<ul style="list-style-type: none"> ▪ Competencia deshabilitada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Competencia actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.21: Caja Negra - Habilitar competencia

CN21: Habilitar competencia		
Requisitos de sistema	RS43, RS41	
Precondiciones	CN01, CN18, CN20	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id de la competencia ▪ Estado 	<ul style="list-style-type: none"> ▪ Competencia habilitada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Competencia actualizada exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.22: Caja Negra - Crear aprendizaje

CN22: Crear aprendizaje		
Requisitos de sistema	RS44	
Precondiciones	CN01, CN18	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Nombre ▪ Categoría ▪ Sub categoría 	<ul style="list-style-type: none"> ▪ Aprendizaje guardado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Aprendizaje guardado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.23: Caja Negra - Modificar aprendizaje

CN23: Modificar aprendizaje		
Requisitos de sistema	RS50, RS45	
Precondiciones	CN01, CN18, CN22	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Nombre ▪ Categoría ▪ Sub categoría 	<ul style="list-style-type: none"> ▪ Aprendizaje actualizado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Aprendizaje actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.24: Caja Negra - Deshabilitar aprendizaje

CN24: Deshabilitar aprendizaje		
Requisitos de sistema	RS50, RS47	
Precondiciones	CN01, CN18, CN22	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del aprendizaje ▪ Estado 	<ul style="list-style-type: none"> ▪ Aprendizaje deshabilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Aprendizaje actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.25: Caja Negra - Habilitar aprendizaje

CN25: Habilitar aprendizaje		
Requisitos de sistema	RS50, RS46	
Precondiciones	CN01, CN18, CN24	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del aprendizaje ▪ Estado 	<ul style="list-style-type: none"> ▪ Aprendizaje habilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Aprendizaje actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.26: Caja Negra - Crear saber

CN26: Crear saber		
Requisitos de sistema	RS51	
Precondiciones	CN01, CN22	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Código ▪ Nombre ▪ Tipo del saber 	<ul style="list-style-type: none"> ▪ Saber guardado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Saber guardado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.27: Caja Negra - Modificar saber

CN27: Modificar saber		
Requisitos de sistema	RS57, RS52	
Precondiciones	CN01, CN26	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Código ▪ Nombre ▪ Tipo del saber 	<ul style="list-style-type: none"> ▪ Saber actualizado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Saber actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.28: Caja Negra - Deshabilitar saber

CN28: Deshabilitar saber		
Requisitos de sistema	RS57, RS54	
Precondiciones	CN01, CN26	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del saber ▪ Estado 	<ul style="list-style-type: none"> ▪ Saber deshabilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Saber actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.29: Caja Negra - Habilitar saber

CN29: Habilitar saber		
Requisitos de sistema	RS57, RS53	
Precondiciones	CN01, CN26, CN28	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del saber ▪ Estado 	<ul style="list-style-type: none"> ▪ Saber habilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Saber actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.30: Caja Negra - Asignar profesor a curso

CN30: Asignar profesor a curso		
Requisitos de sistema	RS24, RS16	
Precondiciones	CN01, CN06, CN35	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del profesor 	<ul style="list-style-type: none"> ▪ Asignación profesor-curso guardado en la base de datos ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Asignación profesor-curso actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.31: Caja Negra - Desasignar profesor a curso

CN31: Desasignar profesor a curso		
Requisitos de sistema	RS24, RS17	
Precondiciones	CN01, CN06, CN30	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del profesor 	<ul style="list-style-type: none"> ▪ Desvinculación profesor-curso actualizado en la base de datos ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Desvinculación profesor-curso actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.32: Caja Negra - Asignar alumno a curso

CN32: Asignar alumno a curso		
Requisitos de sistema	RS24, RS20	
Precondiciones	CN01, CN06, CN35	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del alumno 	<ul style="list-style-type: none"> ▪ Asignación alumno-curso guardado en la base de datos ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Asignación alumno-curso actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.33: Caja Negra - Desasignar alumno a curso

CN33: Desasignar alumno a curso		
Requisitos de sistema	RS24, RS21	
Precondiciones	CN01, CN06, CN32	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del alumno 	<ul style="list-style-type: none"> ▪ Desvinculación alumno-curso actualizado en la base de datos ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Desvinculación profesor-curso actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.34: Caja Negra - Crear usuario

CN34: Crear usuario		
Requisitos de sistema	RS02	
Precondiciones	CN01	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Rut ▪ Nombre ▪ Correo electrónico ▪ Tipo de usuario ▪ Matrícula 	<ul style="list-style-type: none"> ▪ Usuario guardado en la base de datos ▪ Correo electrónico enviado con contraseña al usuario ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Usuario guardado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.35: Caja Negra - Modificar usuario

CN35: Modificar usuario		
Requisitos de sistema	RS10, RS03	
Precondiciones	CN01, CN34	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Rut ▪ Nombre ▪ Correo electrónico ▪ Tipo de usuario ▪ Matrícula 	<ul style="list-style-type: none"> ▪ Usuario actualizado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Usuario actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.36: Caja Negra - Deshabilitar usuario

CN36: Deshabilitar usuario		
Requisitos de sistema	RS10, RS04	
Precondiciones	CN01, CN34	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Rut del usuario ▪ Estado 	<ul style="list-style-type: none"> ▪ Usuario deshabilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Usuario actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.37: Caja Negra - Habilitar usuario

CN37: Habilitar usuario		
Requisitos de sistema	RS10, RS05	
Precondiciones	CN01, CN34, CN36	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Rut del usuario ▪ Estado 	<ul style="list-style-type: none"> ▪ Usuario habilitado en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Usuario actualizado exitosamente ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.38: Caja Negra - Recuperar contraseña

CN38: Recuperar contraseña		
Requisitos de sistema	RS08	
Precondiciones	CN01	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Correo electrónico 	<ul style="list-style-type: none"> ▪ Contraseña actualizada en la base de datos ▪ Correo electrónico enviado con contraseña al usuario ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Contraseña actualizada en la base de datos ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.39: Caja Negra - Cambiar Contraseña

CN39: Cambiar Contraseña		
Requisitos de sistema	RS06	
Precondiciones	CN01	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Rut del usuario ▪ Contraseña 	<ul style="list-style-type: none"> ▪ Contraseña actualizada en la base de datos ▪ Mensaje de éxito ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Contraseña actualizada en la base de datos ▪ Mensaje de éxito
Observaciones	-	

Cuadro D.40: Caja Negra - Guardar plan de clases

CN40: Guardar plan de clases		
Requisitos de sistema	RS09	
Precondiciones	CN01	
Entrada	Salida esperada	Salida obtenida
<ul style="list-style-type: none"> ▪ Id del curso ▪ Id del curso por periodo 	<ul style="list-style-type: none"> ▪ Documento generado para el usuario ▪ Excepción: Mensaje de error 	<ul style="list-style-type: none"> ▪ Documento generado exitosamente ▪ Mensaje de éxito
Observaciones	-	

D.2. Pruebas de caja negra, Imágenes

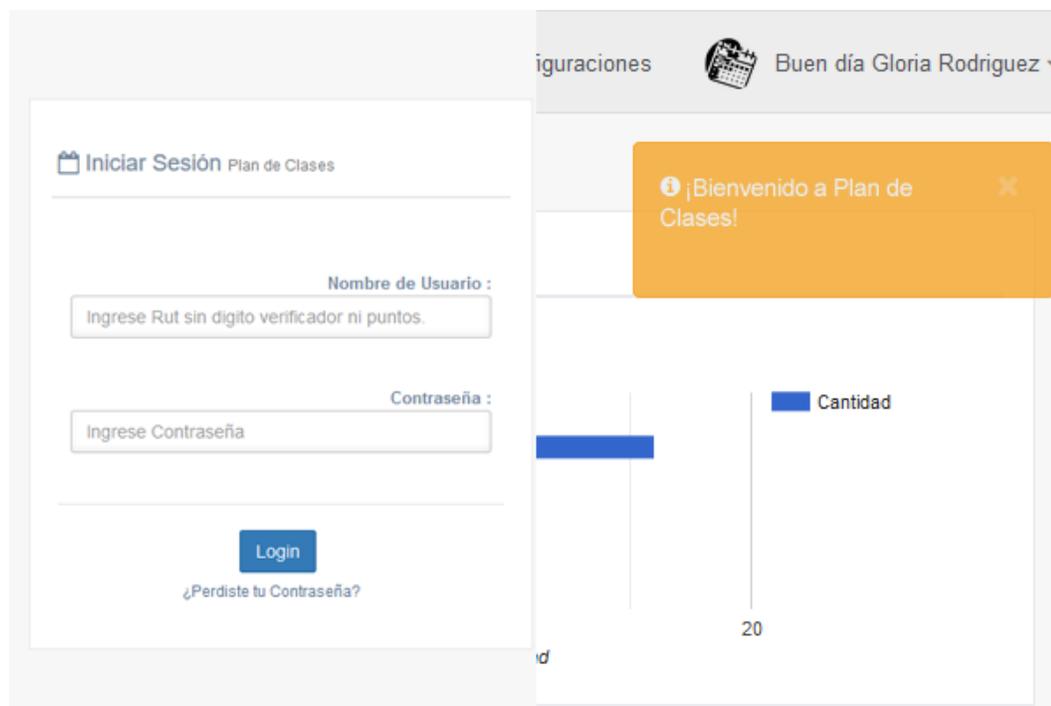


Figura D.1: CN01 Iniciar sesión, Entrada - Salida

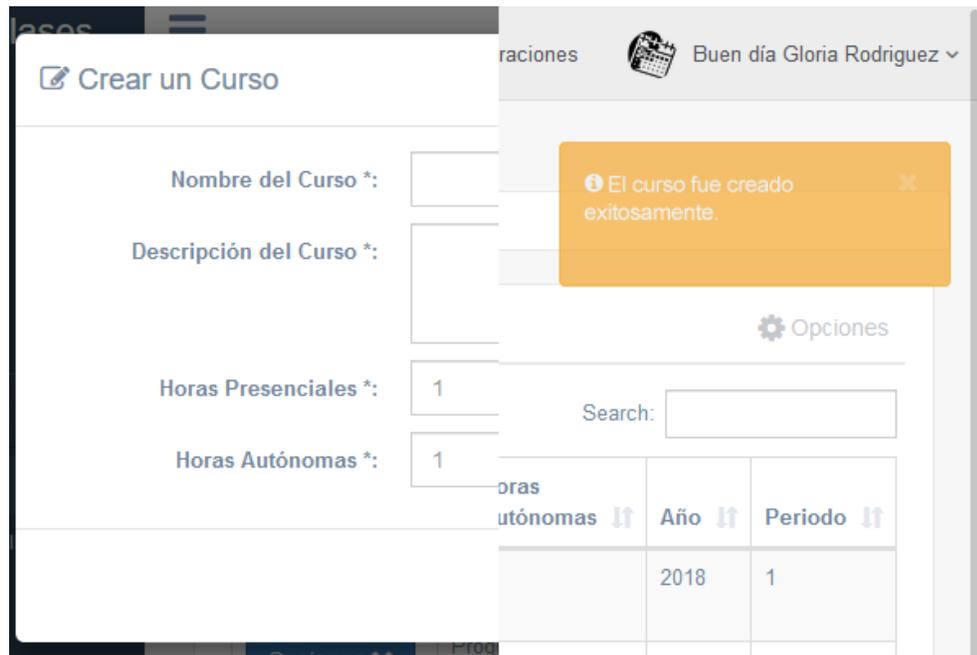


Figura D.2: CN02 Crear un curso, Entrada - Salida

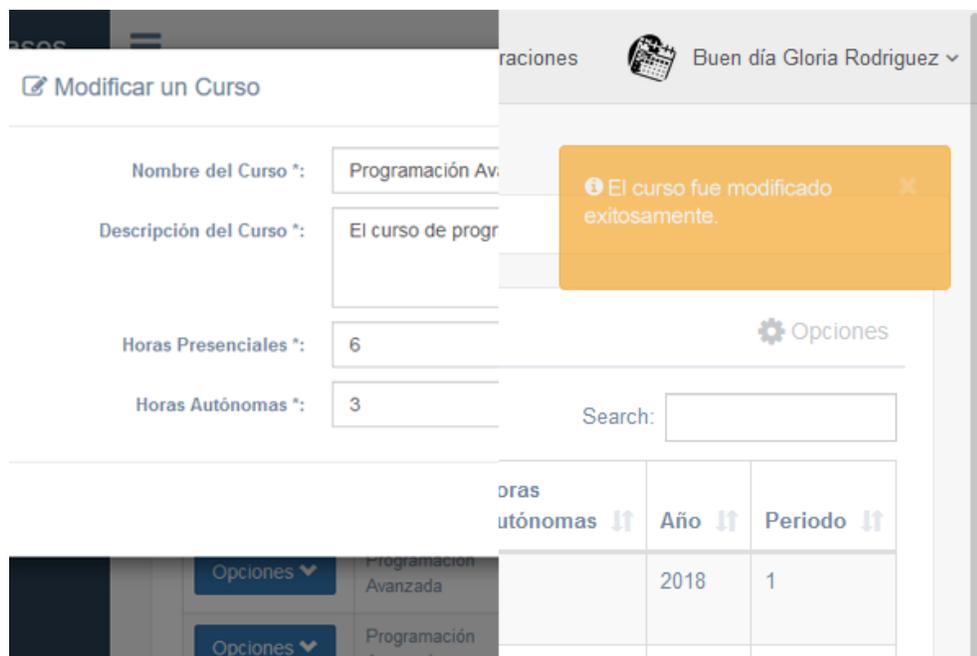


Figura D.3: CN03 Modificar curso, Entrada - Salida



Figura D.4: CN04 Deshabilitar curso, Entrada - Salida

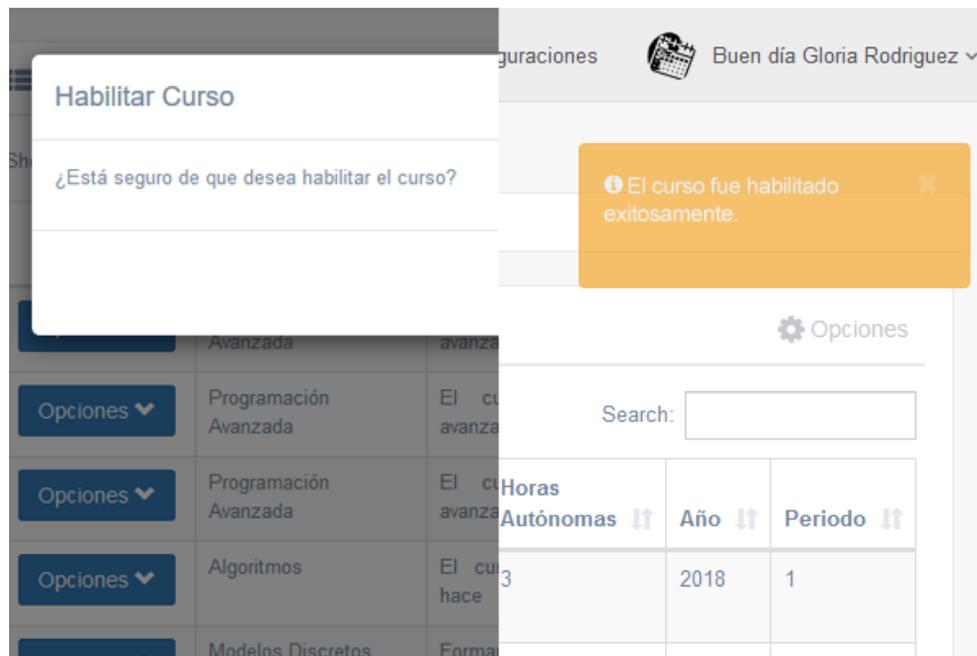


Figura D.5: CN05 Habilitar curso, Entrada - Salida



Figura D.6: CN06 Crear curso por periodo, Entrada - Salida

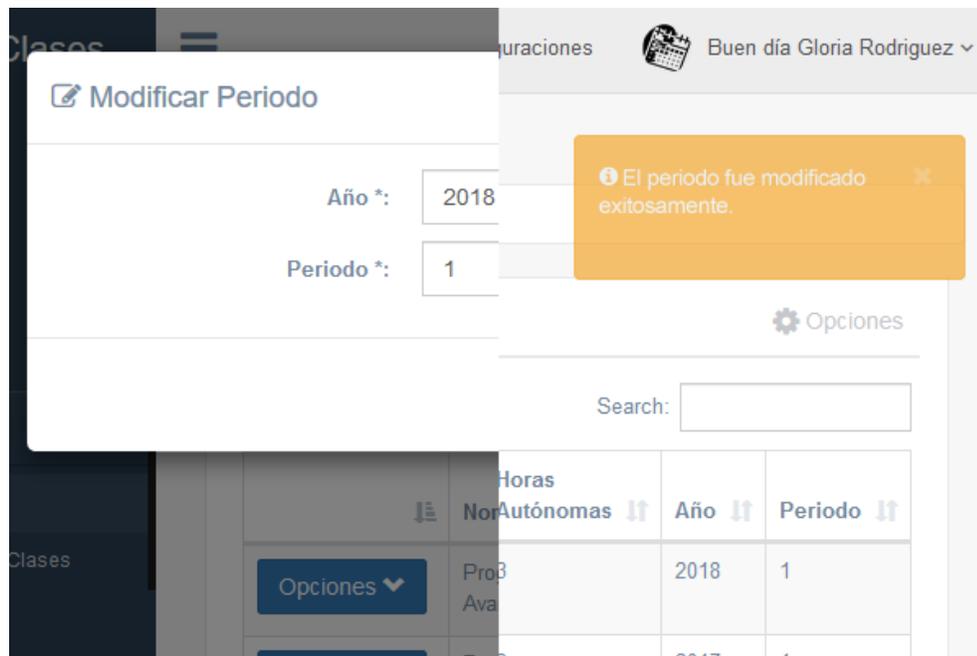


Figura D.7: CN07 Modificar curso por periodo, Entrada - Salida

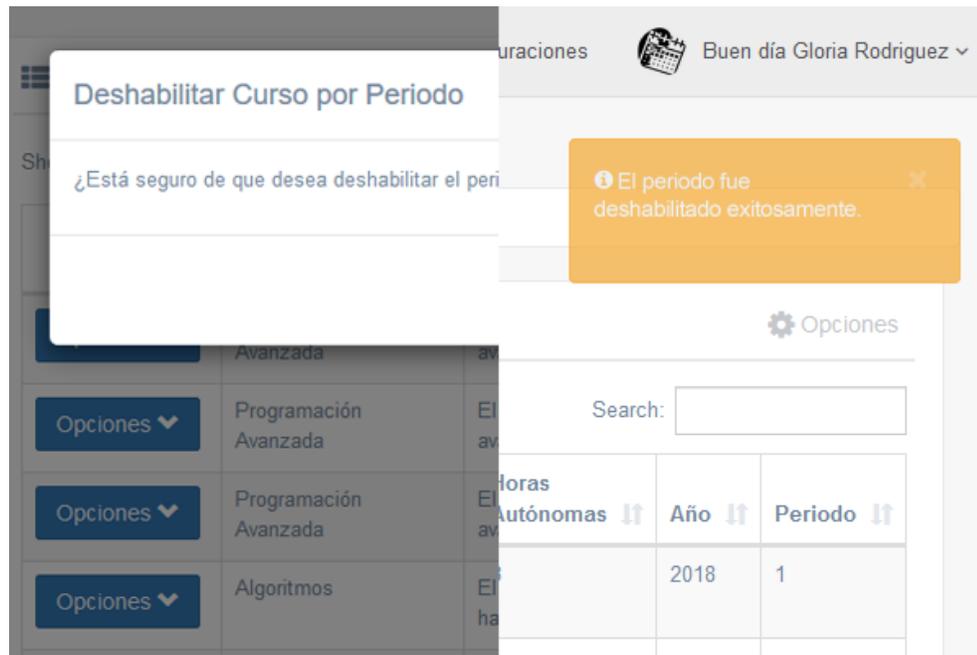


Figura D.8: CN08 Deshabilitar curso por periodo, Entrada - Salida



Figura D.9: CN09 Habilitar curso por periodo, Entrada - Salida

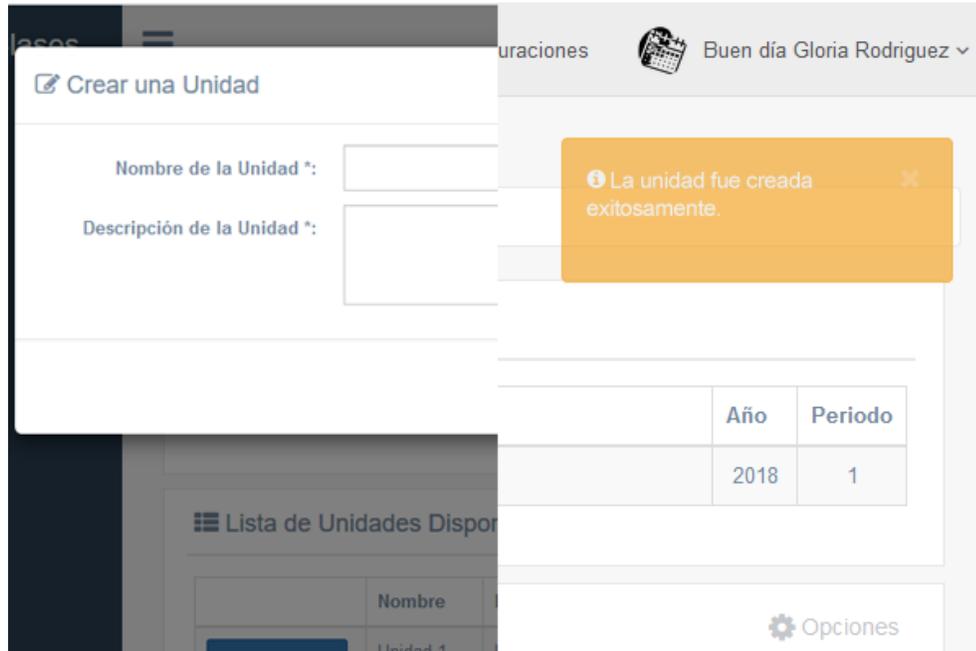


Figura D.10: CN10 Crear unidad, Entrada - Salida

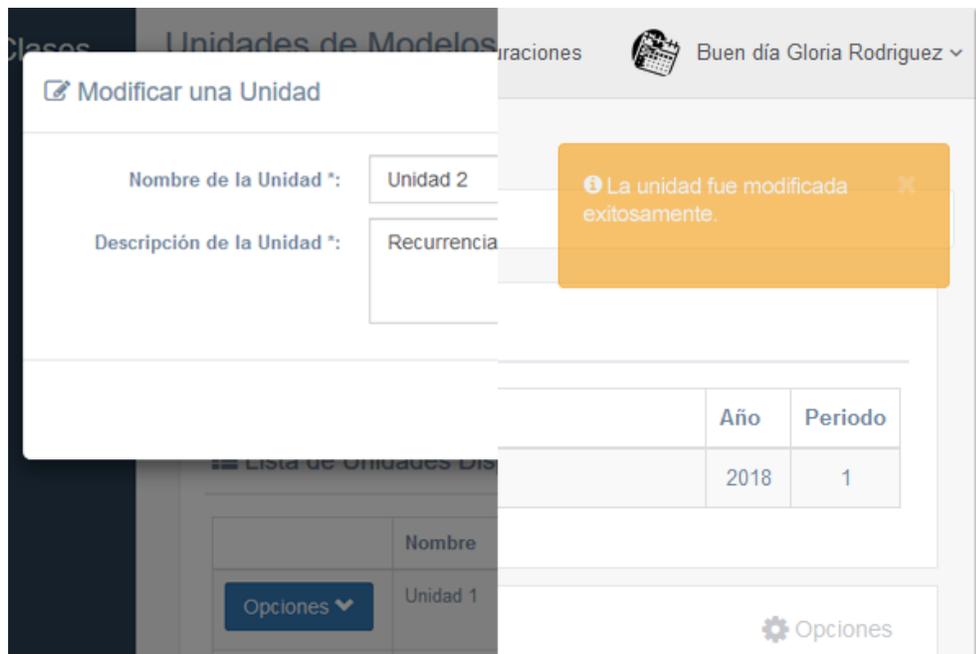


Figura D.11: CN11 Modificar unidad, Entrada - Salida

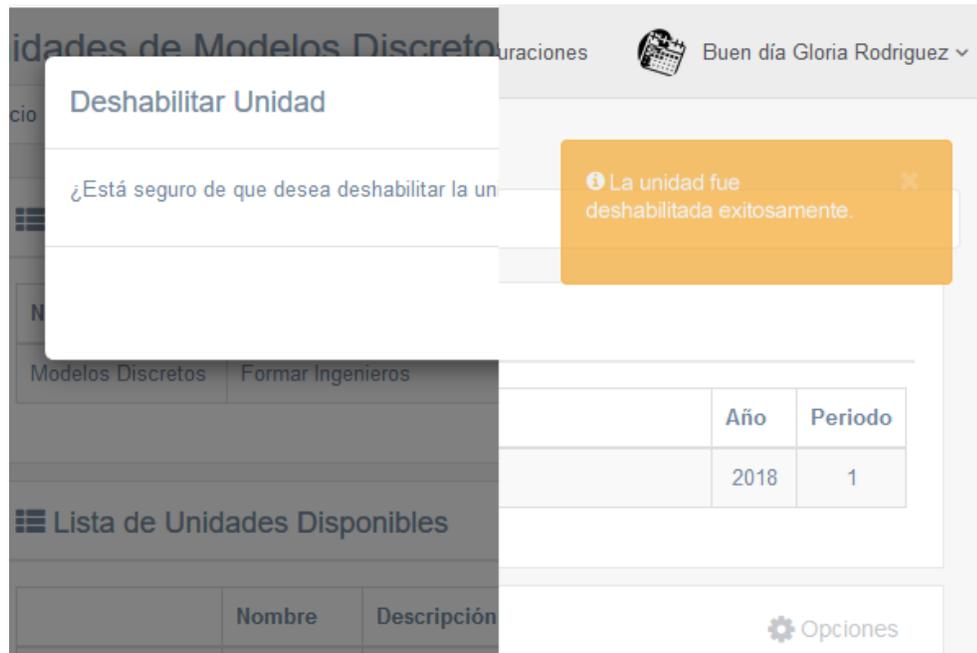


Figura D.12: CN12 Deshabilitar unidad, Entrada - Salida

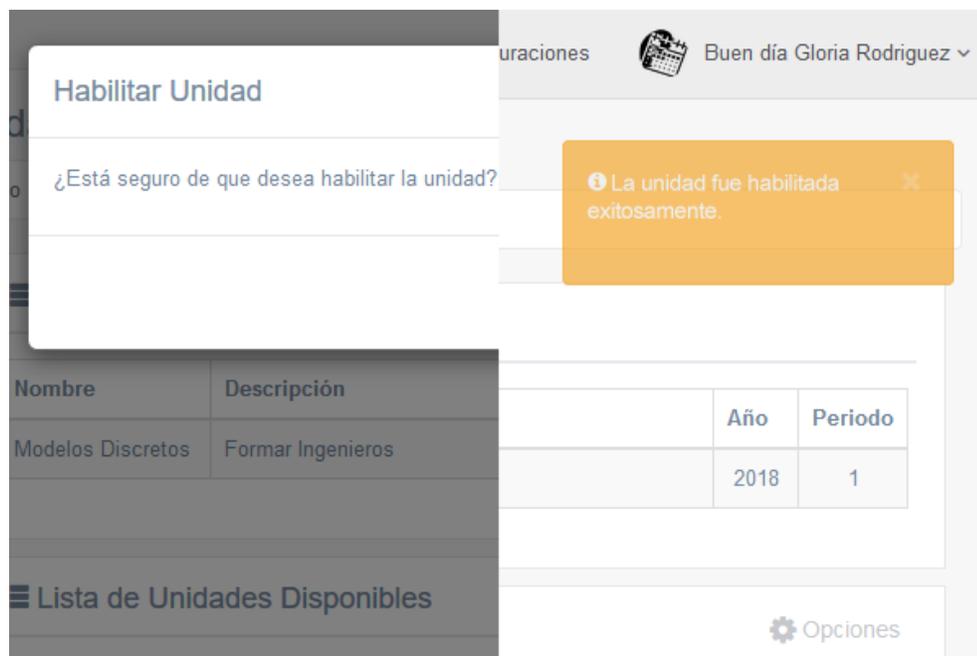


Figura D.13: CN13 Habilitar unidad, Entrada - Salida

Crear una Actividad

Nombre Actividad *: Tarea unidad 1

Descripción *: Tarea sobre recurrence

Fecha de Inicio *: 04/06/2019 00:00 A

Fecha de Término *: 06/06/2019 00:00 A

Tipo de Actividad *: Tarea

Tiempo Presencial *: 0

TiempoNo Presencial *: 8

La actividad fue creada exitosamente.

Opciones

month week day agenda

as. Puedes crear una actividad de varios

Figura D.14: CN14 Crear actividad, Entrada - Salida

Editar una Actividad

Nombre Actividad *: Prueba Unidad

Descripción *: Inducción simple

Fecha de Inicio *: 11/06/2019 00:00 A

Fecha de Término *: 15/06/2019 00:00 A

Tipo de Actividad *: Prueba

Tiempo Presencial *: 5

TiempoNo Presencial *: 5

La actividad fue modificada exitosamente.

Opciones

as. Puedes crear una actividad de varios

Figura D.15: CN15 Modificar actividad, Entrada - Salida

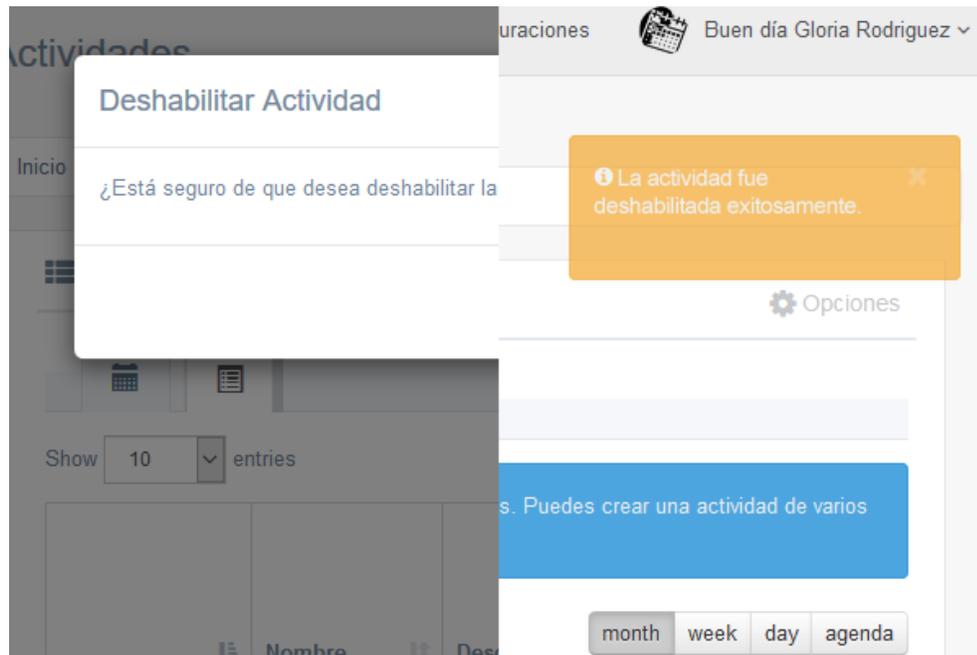


Figura D.16: CN16 Deshabilitar actividad, Entrada - Salida

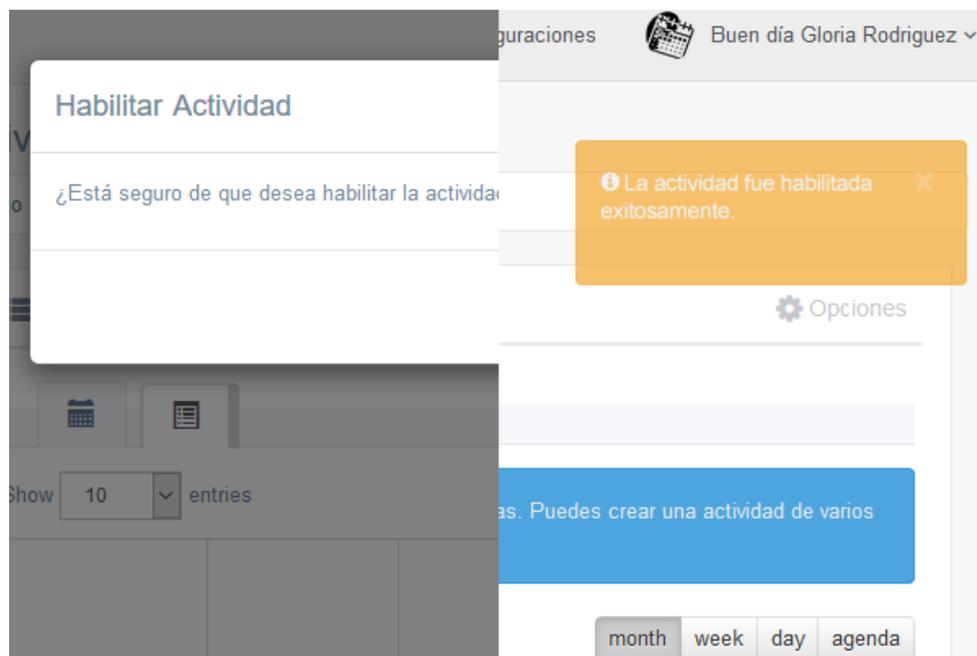


Figura D.17: CN17 Habilitar actividad, Entrada - Salida

de Clases

Buen día Gloria Rodriguez

Crear una Competencia

Nombre de la Competencia:

Descripción de la Competencia *:

Nivel de la Competencia *: Básico

Descripción Básica del Nivel *: Escriba aquí la descripción básica de

Descripción Intermedia del Nivel *: Escriba aquí la descripción intermedia

Descripción Avanzada del Nivel *: Escriba aquí la descripción avanzada

Tiempo de Desarrollo de la Competencia *:

La competencia fue creada exitosamente.

Opciones

Search:

Deshabilitar Modificar

Figura D.18: CN18 Crear competencia, Entrada - Salida

de Clases

Buen día Gloria Rodriguez

Editar Competencias

Nombre de la Competencia: Dominar el cuerpo de conocimiento de

Descripción de la Competencia *: Implica que el graduado ha logrado de conocimientos de ciencias de la compu Ciencias de Computación.

Nivel de la Competencia *: Avanzado

Descripción Básica del Nivel *: Comprende los Fundamentos de Siste estándares reconocidos para su corre

Descripción Intermedia del Nivel *: Comprende además las Estructuras D datos, Lenguajes de programación y A correcto desempeño profesional.

Descripción Avanzada del Nivel *: Comprender además la Computación I Inteligentes y Ciencias de la Computar su correcto desempeño profesional.

Tiempo de Desarrollo de la Competencia *: El estudiantes inicia del desarrollo de

La competencia ha sido modificada exitosamente.

Opciones

Search:

Deshabilitar Modificar

Figura D.19: CN19 Modificar competencia, Entrada - Salida

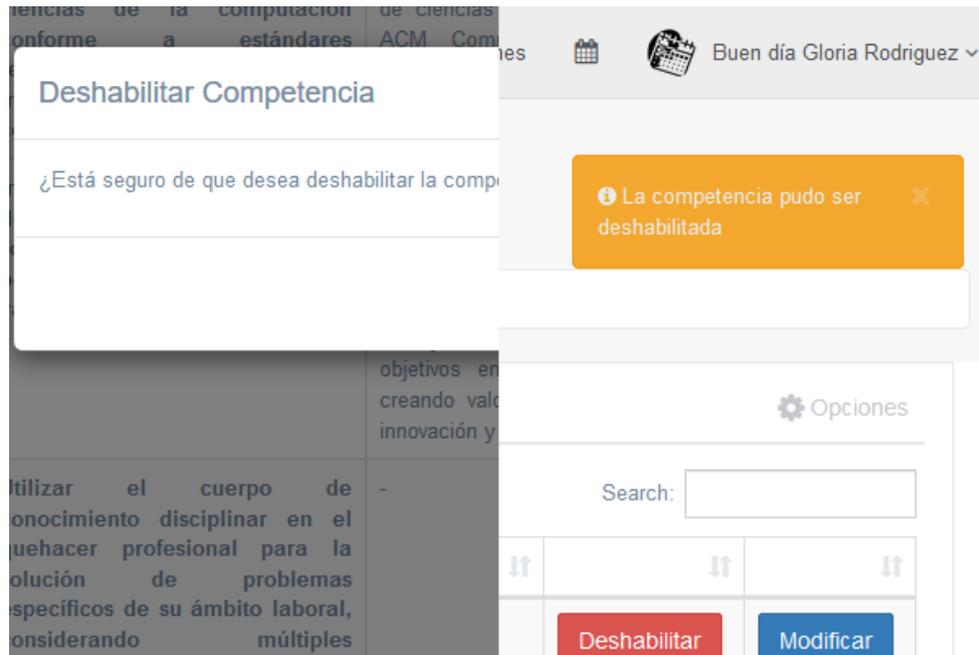


Figura D.20: CN20 Deshabilitar competencia, Entrada - Salida

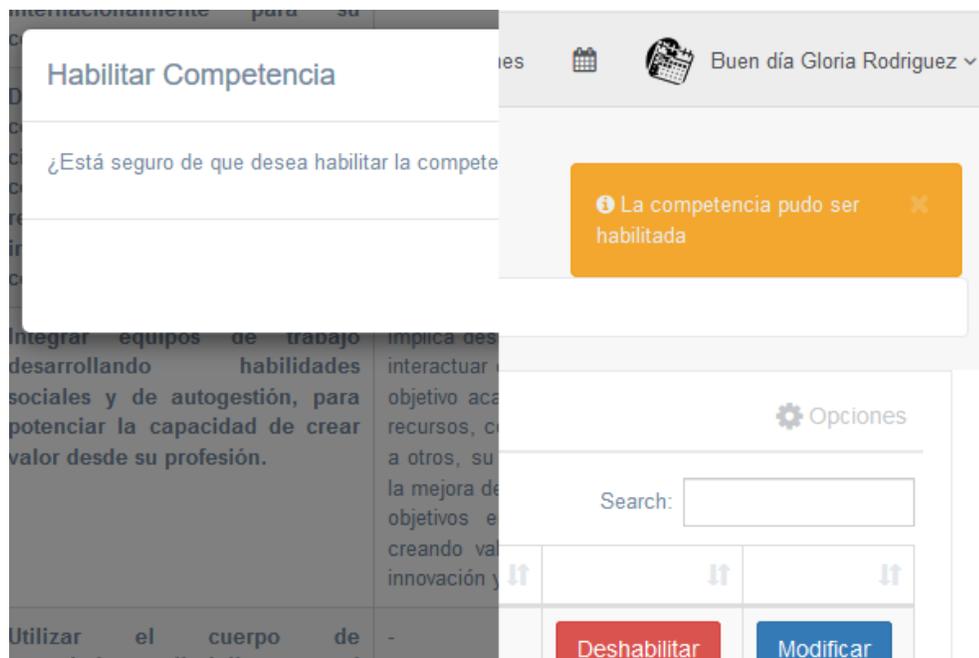


Figura D.21: CN21 Habilitar competencia, Entrada - Salida

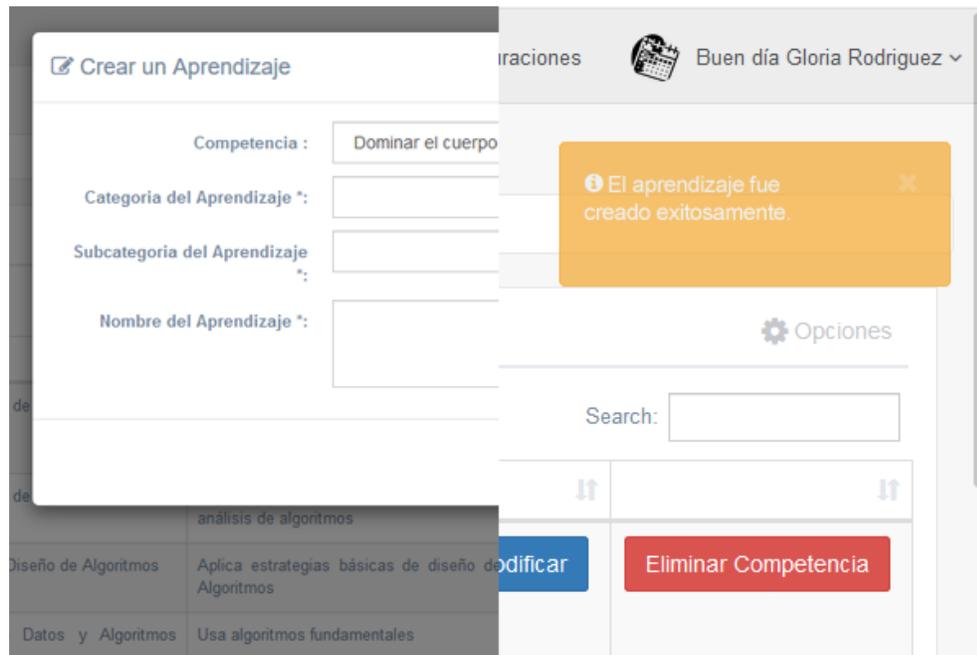


Figura D.22: CN22 Crear aprendizaje, Entrada - Salida

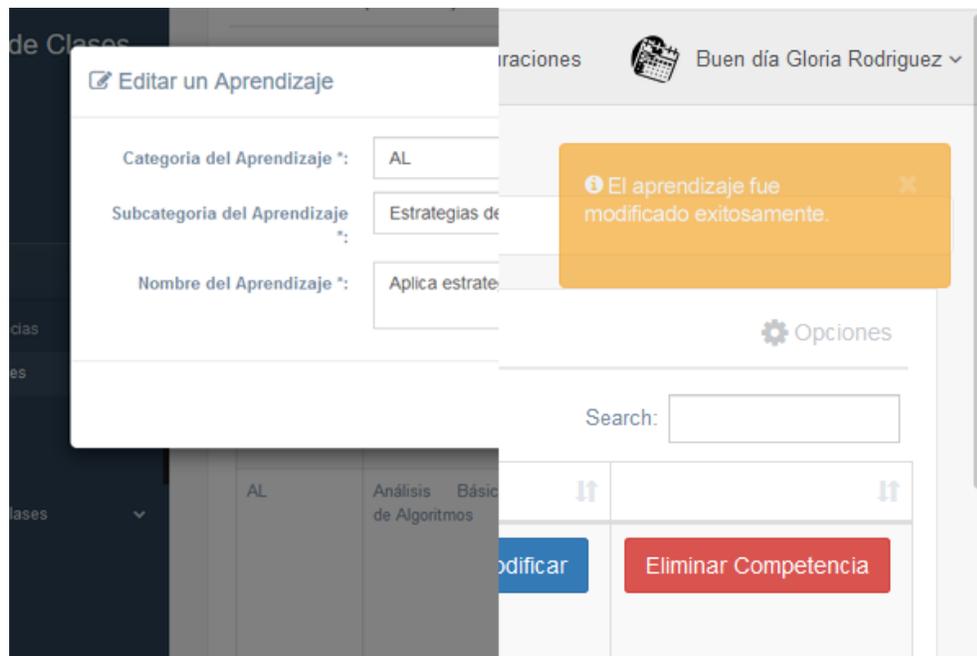


Figura D.23: CN23 Modificar aprendizaje, Entrada - Salida



Figura D.24: CN24 Deshabilitar aprendizaje, Entrada - Salida

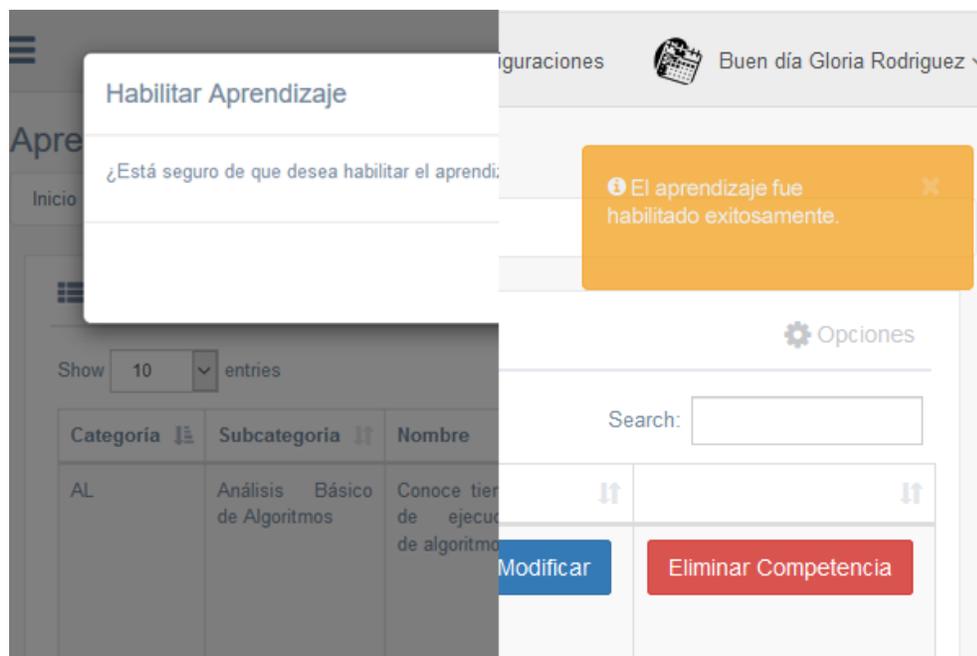


Figura D.25: CN25 Habilitar aprendizaje, Entrada - Salida

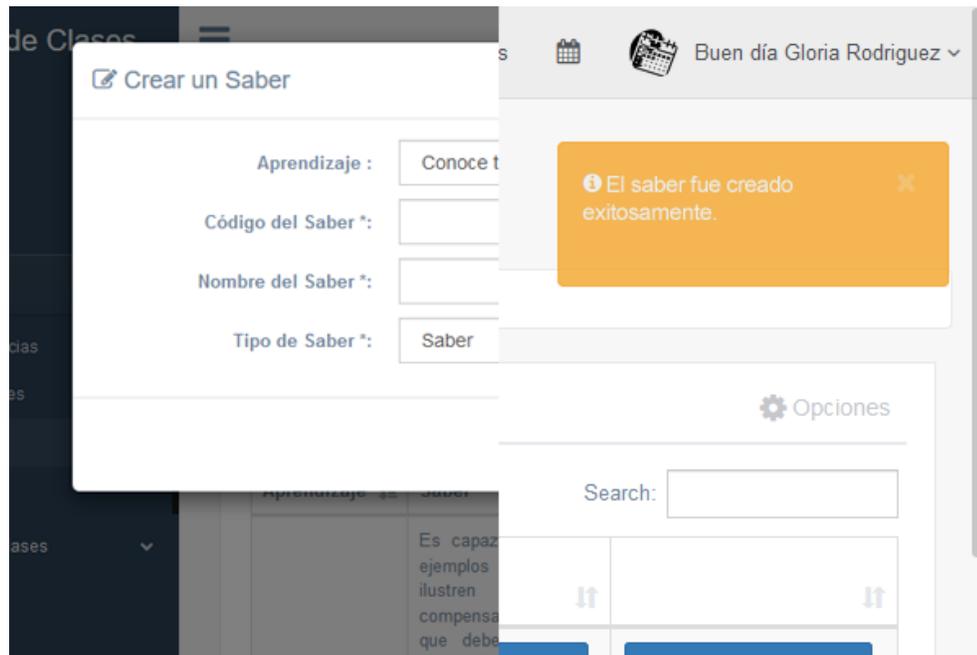


Figura D.26: CN26 Crear saber, Entrada - Salida

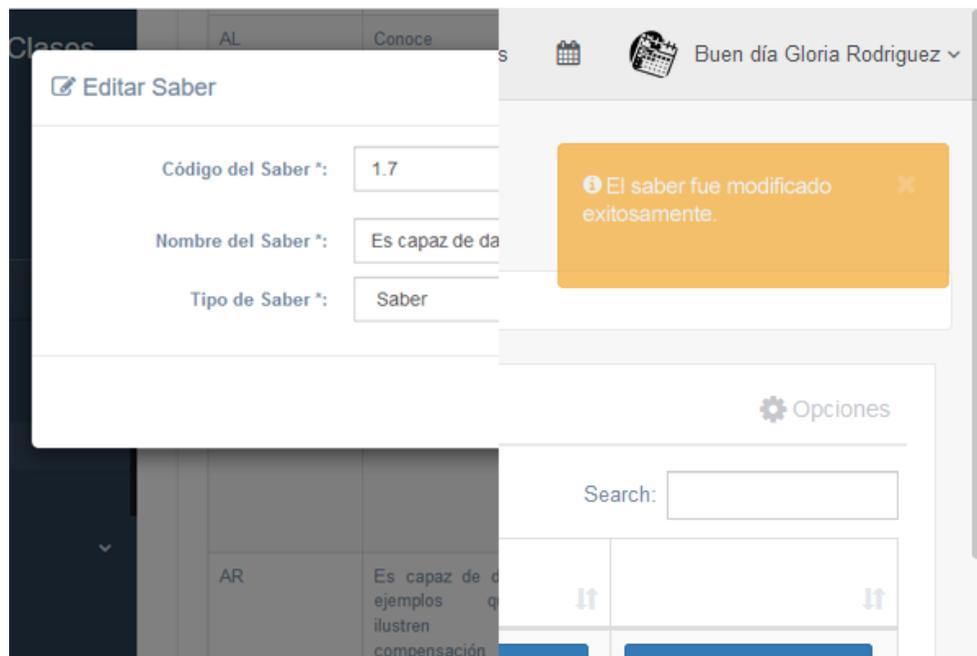


Figura D.27: CN27 Modificar saber, Entrada - Salida

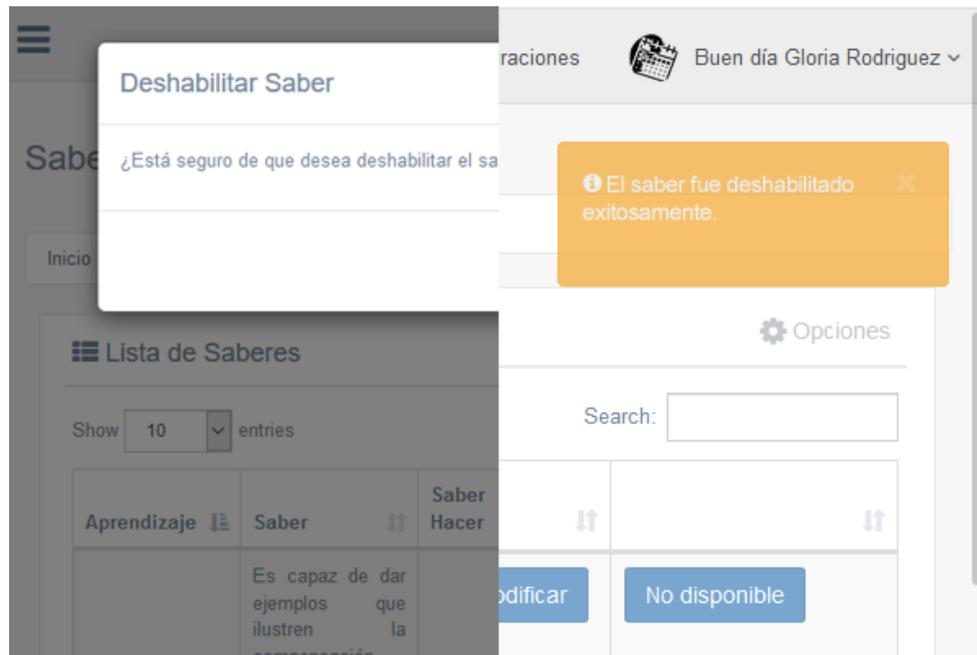


Figura D.28: CN28 Deshabilitar saber, Entrada - Salida

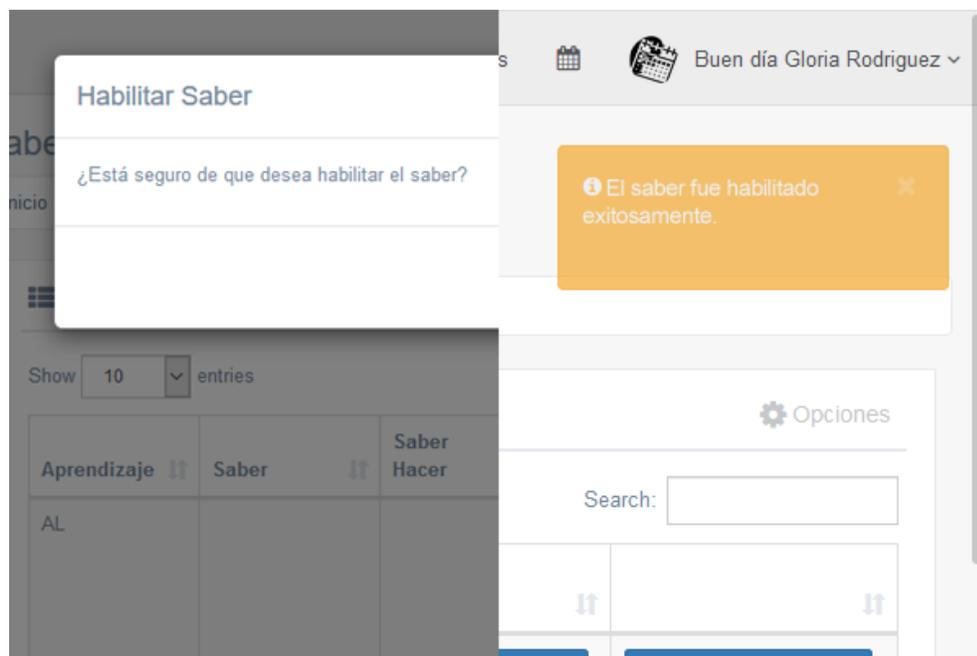


Figura D.29: CN29 Habilitar saber, Entrada - Salida

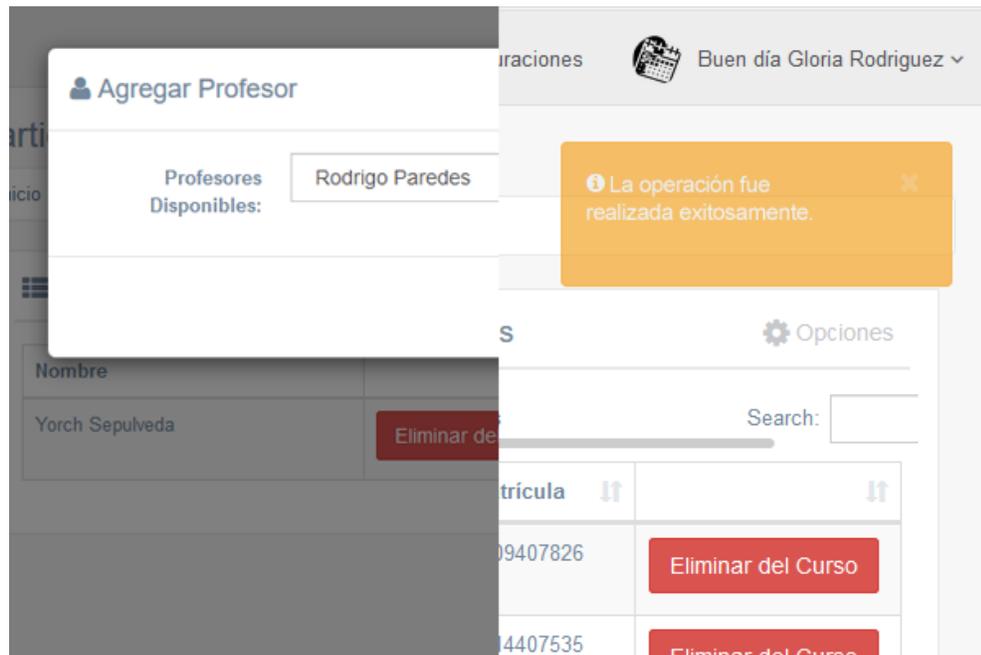


Figura D.30: CN30 Asignar profesor a curso, Entrada - Salida

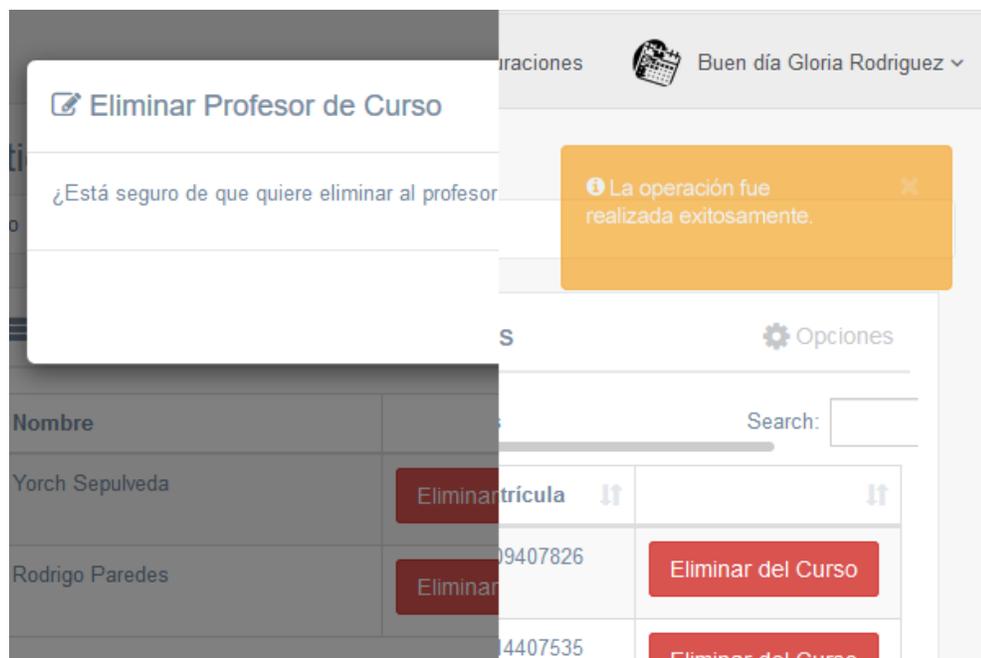


Figura D.31: CN31 Desasignar profesor a curso, Entrada - Salida

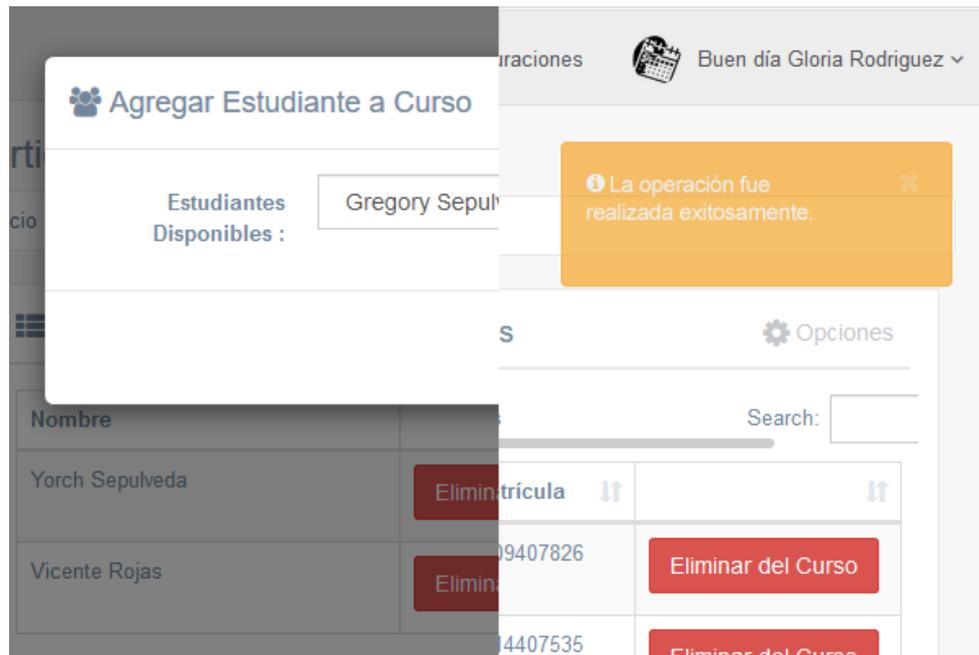


Figura D.32: CN32 Asignar alumno a curso, Entrada - Salida

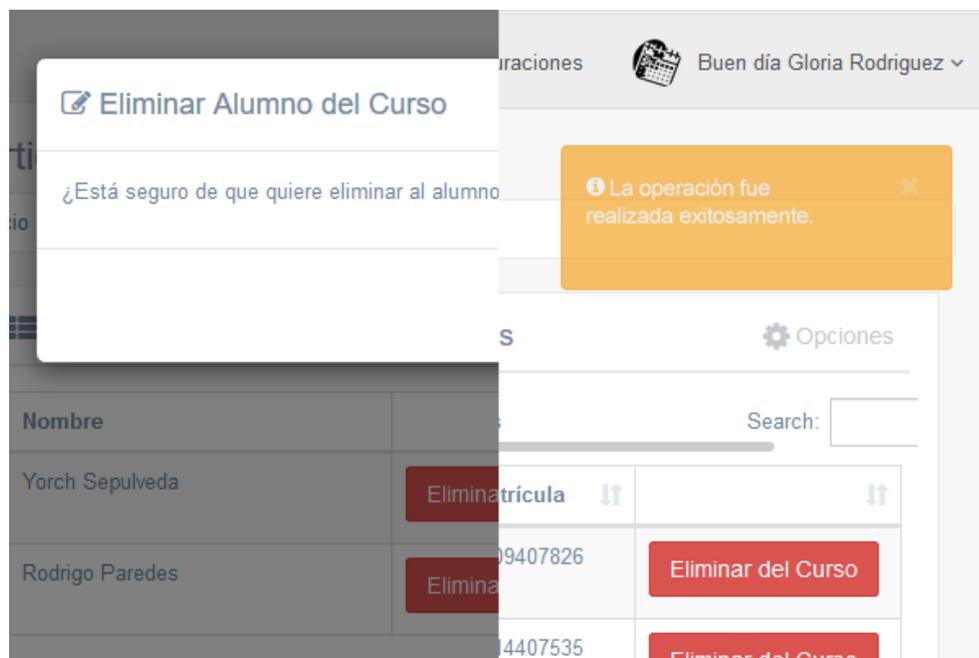


Figura D.33: CN33 Desasignar alumno a curso, Entrada - Salida

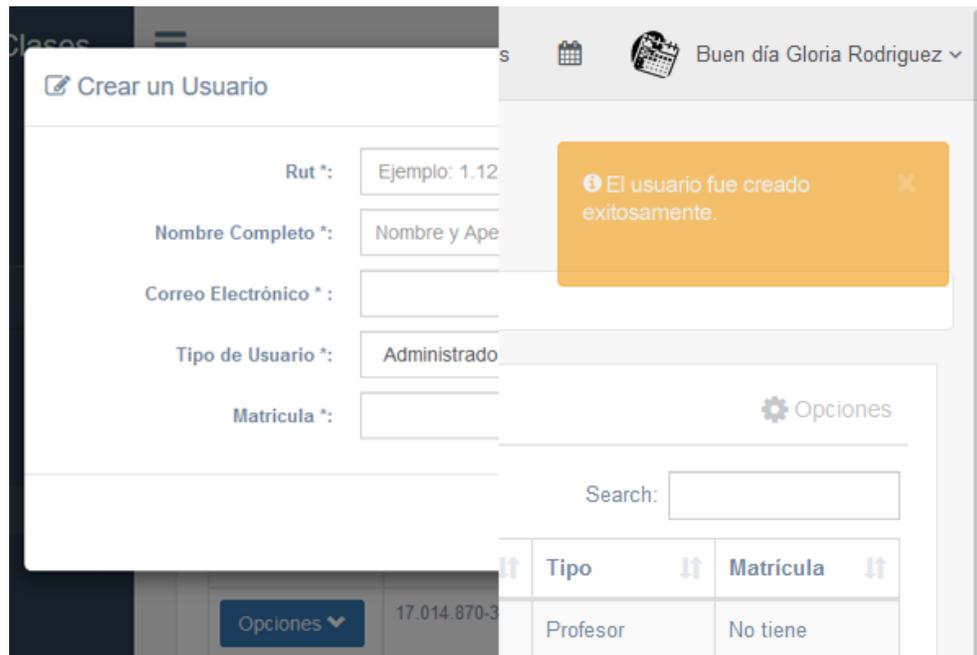


Figura D.34: CN34 Crear usuario, Entrada - Salida

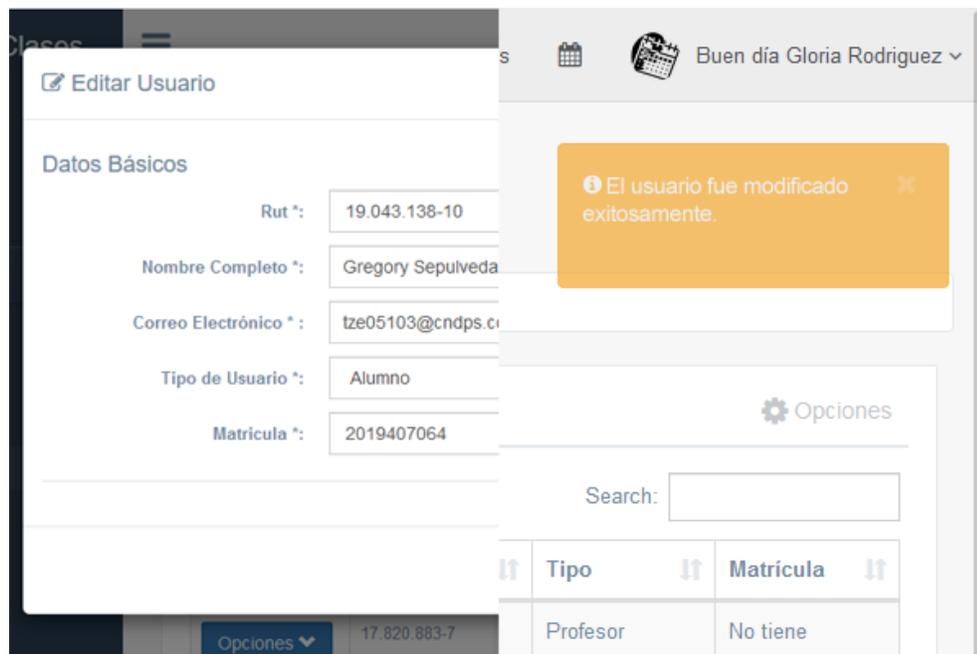


Figura D.35: CN35 Modificar usuario, Entrada - Salida

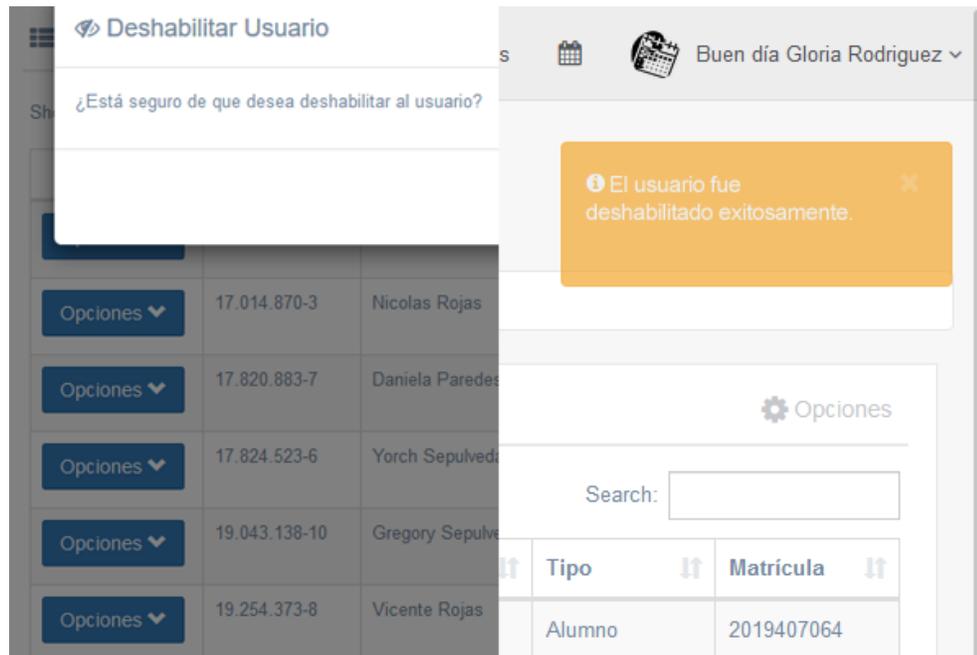


Figura D.36: CN36 Deshabilitar usuario, Entrada - Salida

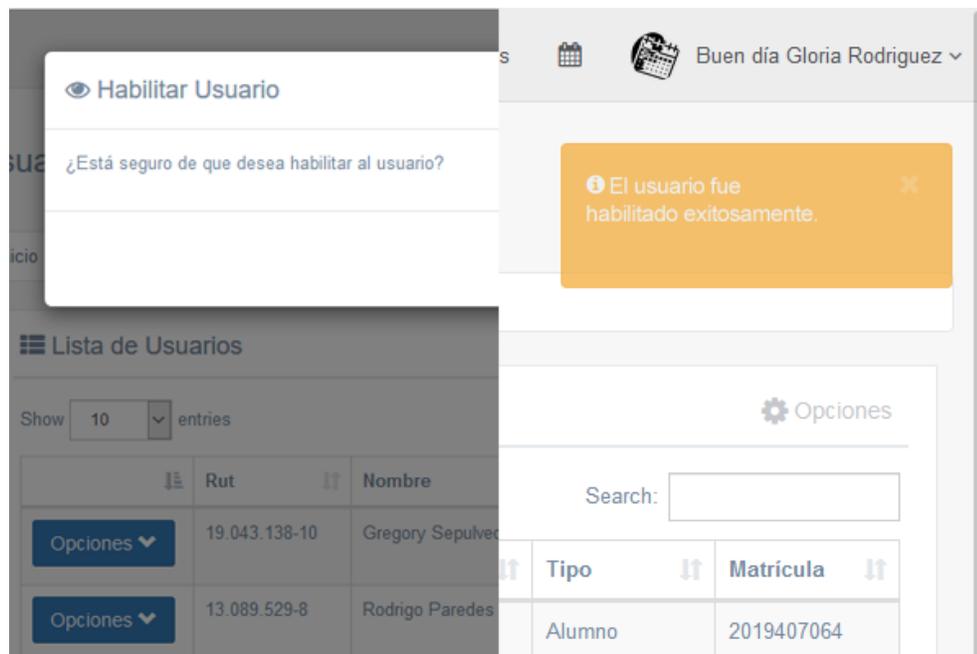


Figura D.37: CN37 Habilitar usuario, Entrada - Salida



Figura D.38: CN38 Recuperar contraseña, Entrada - Salida



Figura D.39: CN39 Cambiar Contraseña, Entrada - Salida

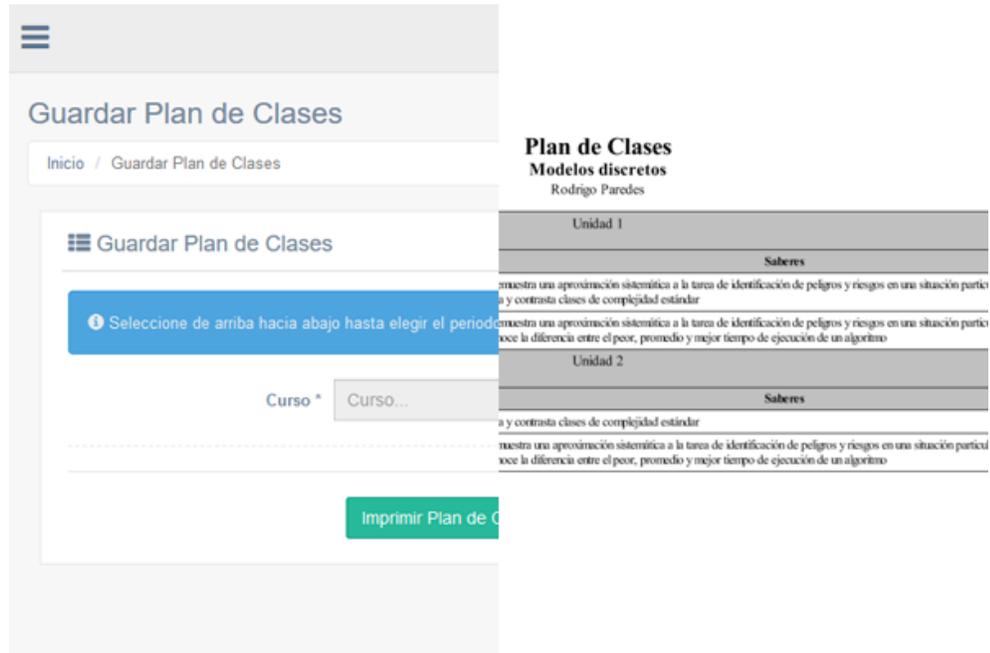


Figura D.40: CN40 Guardar plan de clases, Entrada - Salida

Test de Usabilidad: Plan de Clases

*Obligatorio

Cuestionario Pre-Evaluación

En las siguientes preguntas deberá elegir una respuesta en una escala del uno al cinco, con diferentes significados y que serán expuestos en cada pregunta. También se le proporcionará la instancia para expresar su opinión en caso de que ninguna de las respuestas se ajuste a su opinión.

Indique su nivel de dominio en el uso de un computador o un tablet *

	1	2	3	4	5	
Muy Bajo	<input type="radio"/>	Muy Alto				

¿Cuales son sus expectativas con respecto a la aplicación? *
1= Muy Bajas 5= Muy Altas

	1	2	3	4	5
Interfaz y funcionalidad similar a otras aplicaciones (De preferencia de la Universidad de Talca).	<input type="radio"/>				
Interfaz Atractiva	<input type="radio"/>				
Buen rendimiento (La aplicación funciona a una velocidad deseada).	<input type="radio"/>				

Figura D.42: Pruebas de usabilidad: Cuestionario pre-evaluación Parte 1

Buen rendimiento
(La aplicación funciona a una velocidad deseada).

Funcionalidades Necesarias para poder crear un plan de clases.

Otro ¿Cual?

Tu respuesta

ATRÁS SIGUIENTE

Página 2 de 4

Nunca envíes contraseñas a través de Formularios de Google.

The image shows a Google Form interface for a usability questionnaire. It features two rows of five radio button options each. The first row is titled 'Buen rendimiento (La aplicación funciona a una velocidad deseada)' and the second row is titled 'Funcionalidades Necesarias para poder crear un plan de clases'. Below these is a section for 'Otro ¿Cual?' with a text input field labeled 'Tu respuesta'. At the bottom, there are navigation buttons for 'ATRÁS' and 'SIGUIENTE', a progress indicator showing the current page is 2 of 4, and a security warning: 'Nunca envíes contraseñas a través de Formularios de Google.'

Figura D.43: Pruebas de usabilidad: Cuestionario pre-evaluación Parte 2

Test de Usabilidad: Plan de Clases

*Obligatorio

Cuestionario Evaluación

En las siguientes preguntas deberá realizar una lista de actividades generales, donde en cada una se le realizarán tres tipos de preguntas diferentes.

- Sobre la facilidad de uso: Deberá indicar en una escala del uno al cinco, donde uno significa que fue 'muy difícil' y cinco 'muy fácil' que tan fácil le pareció a usted realizar la actividad. Cabe señalar que la facilidad de uso, implica que tanto demoró usted el comprender el cómo hacerlo o que tanto ayudó en ello la interfaz de la aplicación.
- Sobre la sencillez de uso: Deberá indicar en una escala del uno al cinco que tan sencillo le pareció el proceso, donde uno significa que fue 'muy complejo' y cinco 'muy sencillo'.
- Sobre el tiempo promedio: Deberá indicar una duración promedio de horas, minutos y segundos de cuánto tiempo usted demoró en realizar la actividad.
- Recomendaciones: No es opcional, pero puede indicar usted si tiene una recomendación para mejorar ya sea en interfaz o en usabilidad la actividad realizada.

Ver los Cursos.

¿Qué tan fácil fue? *

	1	2	3	4	5	
Muy Difícil	<input type="radio"/>	Muy Fácil				

¿Qué tan sencillo fue? *

	1	2	3	4	5	
Muy Difícil	<input type="radio"/>	Muy Fácil				

Figura D.44: Pruebas de usabilidad: Cuestionario evaluación Parte 1

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s
: :

Recomendaciones
Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta

Cambiar su contraseña.

¿Qué tan fácil fue? *

1 2 3 4 5
Muy Difícil Muy Fácil

¿Qué tan sencillo fue? *

1 2 3 4 5
Muy Difícil Muy Fácil

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s
: :

Recomendaciones
Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta

Figura D.45: Pruebas de usabilidad: Cuestionario evaluación Parte 2

Crear una competencia, un aprendizaje y dos saberes

¿Qué tan fácil fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Qué tan sencillo fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s

__ : __ : __

Recomendaciones
Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta _____

Figura D.46: Pruebas de usabilidad: Cuestionario evaluación Parte 3

Agregar un aprendizaje a una competencia y los saberes a un aprendizaje.

¿Qué tan fácil fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Qué tan sencillo fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s

__ : __ : __

Recomendaciones
Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta _____

Figura D.47: Pruebas de usabilidad: Cuestionario evaluación Parte 4

Crear un curso, una unidad y algunas actividades.

¿Qué tan fácil fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Qué tan sencillo fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s

__ : __ : __

Recomendaciones
Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta _____

Figura D.48: Pruebas de usabilidad: Cuestionario evaluación Parte 5

Asociar una competencia a un curso.

¿Qué tan fácil fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Qué tan sencillo fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s

__ : __ : __

Recomendaciones
Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta _____

Figura D.49: Pruebas de usabilidad: Cuestionario evaluación Parte 6

Asociar el saber de un curso a una unidad y a la o las actividades

¿Qué tan fácil fue? *

	1	2	3	4	5	
Muy Difícil	<input type="radio"/>	Muy Fácil				

¿Qué tan sencillo fue? *

	1	2	3	4	5	
Muy Difícil	<input type="radio"/>	Muy Fácil				

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s

__ : __ : __

Recomendaciones
Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta _____

Figura D.50: Pruebas de usabilidad: Cuestionario evaluación Parte 7

Configurar periodo y año.

¿Qué tan fácil fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Qué tan sencillo fue? *

1 2 3 4 5

Muy Difícil Muy Fácil

¿Cuánto tiempo demoró en promedio en realizar la actividad? *

h min s

__ : __ : __

Recomendaciones

Indique usted si tiene una recomendación para mejorar esta funcionalidad.

Tu respuesta _____

Figura D.51: Pruebas de usabilidad: Cuestionario evaluación Parte 8

Test de Usabilidad: Plan de Clases

*Obligatorio

Cuestionario Post-Evaluación

En las siguientes preguntas deberá elegir una respuesta en una escala del uno al cinco, donde uno significa 'en desacuerdo' y cinco significa 'muy de acuerdo'. Cabe sugerir, que sería de mucha utilidad si posterior a realizar las actividades explorara la aplicación en general.

En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones *

1=En Desacuerdo 5=Muy De Acuerdo

	1	2	3	4	5
Pienso que necesitaré ayuda o un manual de usuario para poder utilizar el prototipo.	<input type="radio"/>				
Las funcionalidades del prototipo se encuentran bien integradas.	<input type="radio"/>				
El prototipo es fácil de usar.	<input type="radio"/>				
El prototipo es sencillo de usar.	<input type="radio"/>				
El prototipo rindió perfectamente (no se cayó en ningún momento, no se volvió lento de repente).	<input type="radio"/>				
La interfaz del prototipo es agradable.	<input type="radio"/>				

Figura D.53: Pruebas de usabilidad: Cuestionario post-evaluación Parte 1

No necesité de más funcionalidades para poder realizar un plan de clases.	<input type="radio"/>				
Me siento seguro/a de utilizar el prototipo.	<input type="radio"/>				

[ATRÁS](#) [ENVIAR](#) Página 4 de 4

Nunca envíes contraseñas a través de Formularios de Google.

Figura D.54: Pruebas de usabilidad: Cuestionario post-evaluación Parte 2

D.4. Pruebas de usabilidad: Resultado de encuestas

Indique su nivel de dominio en el uso de un computador o un tablet	¿Cuales son sus expectativas con respecto a la aplicación? [Interfaz y funcionalidad similar a otras aplicaciones (De preferencia de la Universidad de Talca).]	¿Cuales son sus expectativas con respecto a la aplicación? [Interfaz Atractiva]	¿Cuales son sus expectativas con respecto a la aplicación? [Buen rendimiento (La aplicación funciona a una velocidad deseada).]	¿Cuales son sus expectativas con respecto a la aplicación? [Funcionalidades Necesarias para poder crear un plan de clases.]	Otro ¿Cual?
4	1	3	5	5	
5	4	4	4	4	4
4	5	5	5	5	5
3	1	3	5	5	5
3	4	4	4	4	4
3	2	4	4	4	5
3	3	3	3	3	5
5	5	5	5	5	5
5	1	3	1	1	5
4	2	2	2	2	5
4	1	2	1	1	4

Figura D.55: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 1)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
5	5	0:00:30	Es bastante claro
5	4	0:01:00	
5	5	0:01:00	Estoy en la duda si será
5	5	0:07:00	
4	4	0:02:00	
5	5	0:01:30	
5	5	0:01:30	
5	5	0:00:30	
4	4	0:01:00	
4	4	0:03:00	
5	4	0:00:30	

Figura D.56: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 2)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
5	3	0:01:00	Me gusta que haya dos fo
5	5	0:03:00	
5	2	0:03:00	Siento que colocar los icc
5	5	0:02:00	
5	5	0:03:00	Dos formas de acceder a
5	5	0:02:00	
5	5	0:00:40	
5	5	0:01:00	
4	4	0:05:00	
5	5	0:03:00	
5	5	0:04:00	

Figura D.57: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 3)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
5	4	0:05:00	
5	4	0:02:00	
4	2	0:10:00	Cuando creas cualquiera
5	5	0:10:30	
4	4	0:09:30	
4	4	0:05:00	
5	5	0:05:00	
5	5	0:05:00	
4	4	0:10:00	
4	4	0:10:00	
5	5	0:08:00	

Figura D.58: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 4)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
4	2	0:10:00	Habría sido mas fácil si m
4	4	0:02:00	
2	1	0:15:00	Es la parte de la aplicació
1	1	0:30:00	
1	1	0:35:00	Se me fue muy complicac
2	2	0:06:00	
2	2	0:08:00	
2	2	0:10:00	Funcionalidad innecesaria
3	3	0:15:00	
4	4	0:08:00	
3	2	0:10:00	

Figura D.59: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 5)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
4	1	0:15:00	Es un poco confuso el cre
4	3	0:10:00	
5	3	0:30:00	Colocar un calendario est
5	3	0:15:00	
3	3	0:20:00	Me enredé con el periodo
4	4	0:10:00	
3	3	0:12:00	
4	2	0:08:00	
4	2	0:10:00	
5	5	0:16:00	
5	2	0:10:00	

Figura D.60: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 6)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
5	5	0:00:20	
5	5	0:01:00	
4	4	0:01:00	
4	4	0:08:00	
4	4	0:05:00	
5	3	0:03:00	
5	3	0:02:00	
5	4	0:02:00	
5	5	0:03:00	
4	4	0:02:00	
4	4	0:05:00	

Figura D.61: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 7)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
5	2	0:05:00	
4	5	0:01:00	
4	4	0:01:00	
4	5	0:10:00	
4	4	0:07:00	
5	4	0:03:03	
4	4	0:02:20	
4	4	0:03:00	
5	5	0:03:00	
4	4	0:02:00	
4	4	0:08:00	

Figura D.62: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 8)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
5	5	0:00:30	
5	5	0:01:00	
5	5	0:02:00	A mi parecer, una funcion
5	3	0:05:00	
5	5	0:01:00	
5	3	0:02:00	
5	5	0:01:00	
5	5	0:01:00	No me gustó esta funcion
5	5	0:02:00	
5	5	0:01:00	
5	4	0:04:00	

Figura D.63: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 9)

¿Qué tan fácil fue?	¿Qué tan sencillo fue?	¿Cuánto tiempo demoró en promedio en realizar la actividad?	Recomendaciones
5	5	0:01:00	Me gustó poder bajarlo
5	4	0:01:00	
5	5	0:01:00	Me hubiese gustado más
5	5	0:00:30	
5	5	0:01:00	
5	5	0:01:00	
5	5	0:01:30	
5	5	0:00:20	
5	5	0:01:00	
5	5	0:01:00	
5	5	0:00:30	

Figura D.64: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 10)

En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [Pienso que necesitaré ayuda o un manual de usuario para poder utilizar el prototipo.]	En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [Las funcionalidades del prototipo se encuentran bien integradas.]	En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [El prototipo es fácil de usar.]	En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [El prototipo es sencillo de usar.]	En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [El prototipo rindió perfectamente (no se cayó en ningún momento, no se volvió lento de repente).]	En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [La interfaz del prototipo es agradable.]	En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [No necesité de más funcionalidades para poder realizar un plan de clases.]	En general, indique el nivel de satisfacción con respecto a las siguientes afirmaciones [Me siento seguro/a de utilizar el prototipo.]
3	5	5	2	5	5	5	5
2	3	4	3	4	4	4	4
5	3	5	3	5	5	5	3
5	5	4	2	3	3	4	2
3	4	3	1	5	5	5	3
2	5	5	2	5	5	5	4
2	5	5	3	5	5	5	3
3	3	5	1	5	5	5	3
2	5	4	2	5	5	5	5
4	3	5	2	4	4	4	3
5	3	4	2	5	5	5	3

Figura D.65: Pruebas de usabilidad: Resultado de pruebas de usabilidad (Parte 11)

D.5. Pruebas de usabilidad: gráficos

Ver los Cursos

1=Muy difícil y 5=Muy fácil

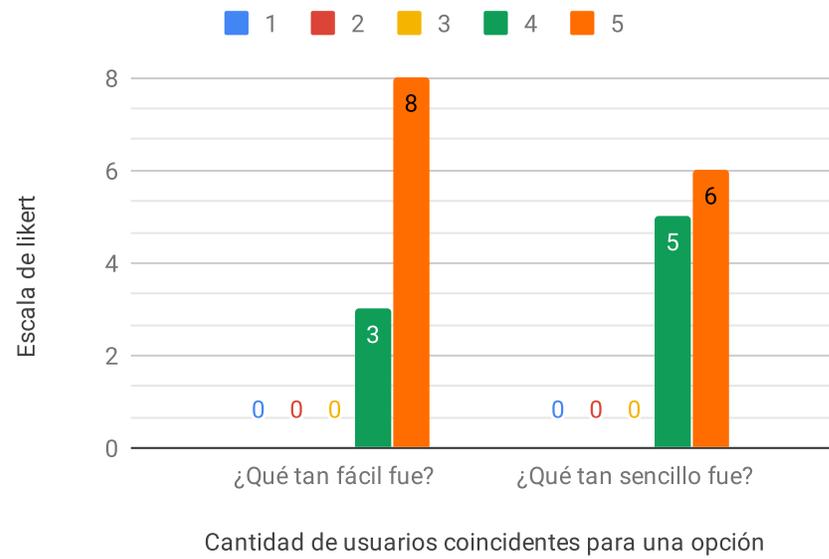


Figura D.66: Test de evaluación: Gráfico de likert ver los cursos

Cambiar su contraseña

1=Muy difícil y 5=Muy fácil

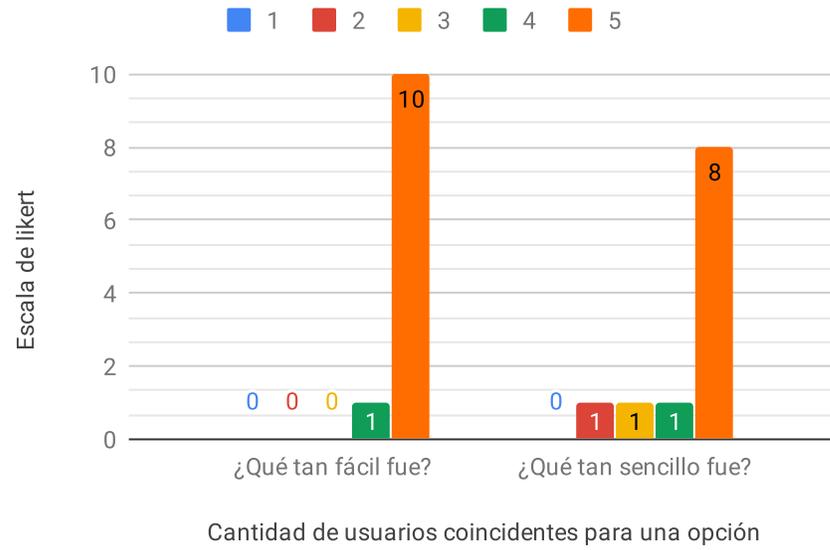


Figura D.67: Test de evaluación: Gráfico de likert cambiar contraseña

Crear una competencia, un aprendizaje y dos saberes

1=Muy difícil y 5=Muy fácil

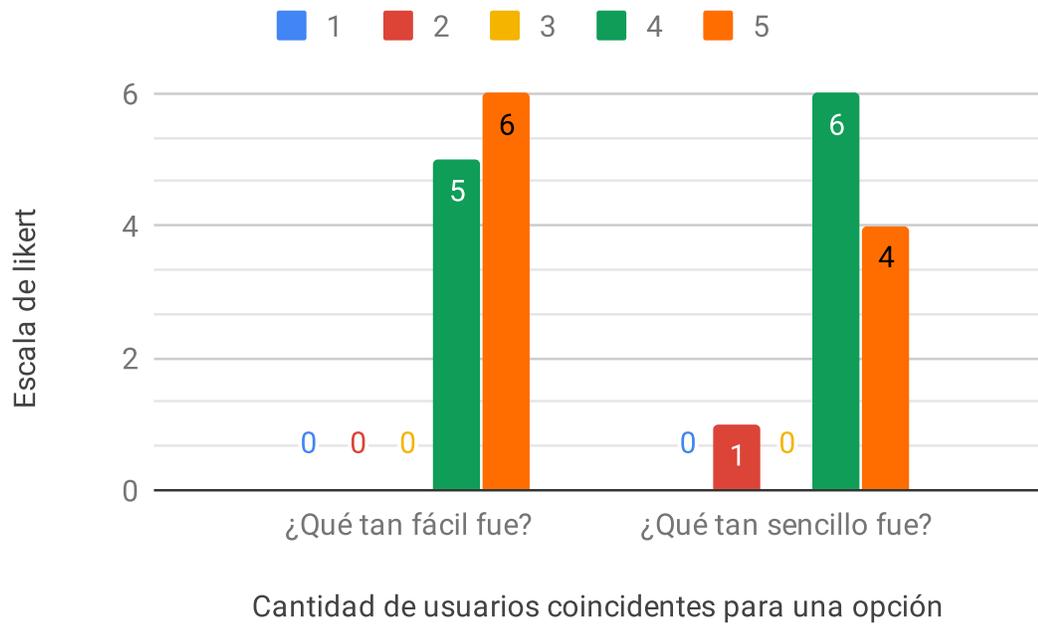


Figura D.68: Test de evaluación: Gráfico de likert crear una competencia, un aprendizaje y dos saberes

Agregar un aprendizaje a una competencia y los saberes a un aprendizaje

1=Muy difícil y 5=Muy fácil

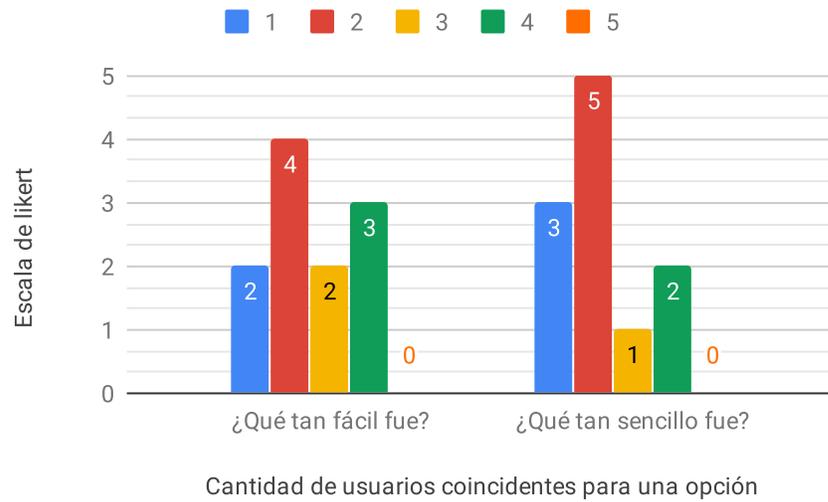


Figura D.69: Test de evaluación: Gráfico de likert agregar un aprendizaje a una competencia y los saberes a un aprendizaje

Crear un curso, una unidad y algunas actividades

1=Muy difícil y 5=Muy fácil

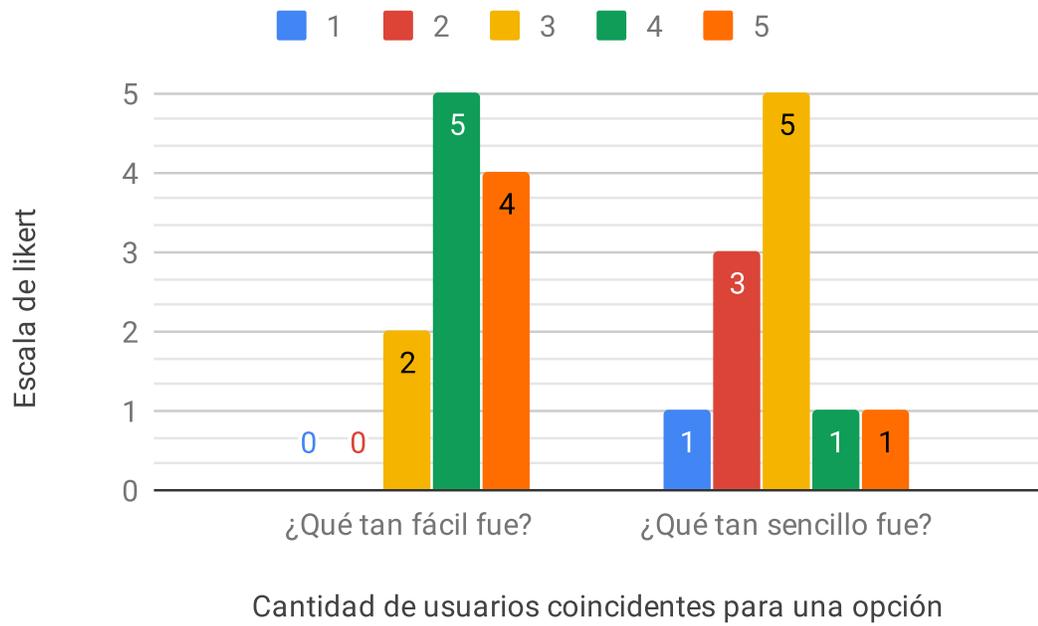


Figura D.70: Test de evaluación: Gráfico de likert crear un curso, una unidad y algunas actividades

Asociar una competencia a un curso

1=Muy difícil y 5=Muy fácil

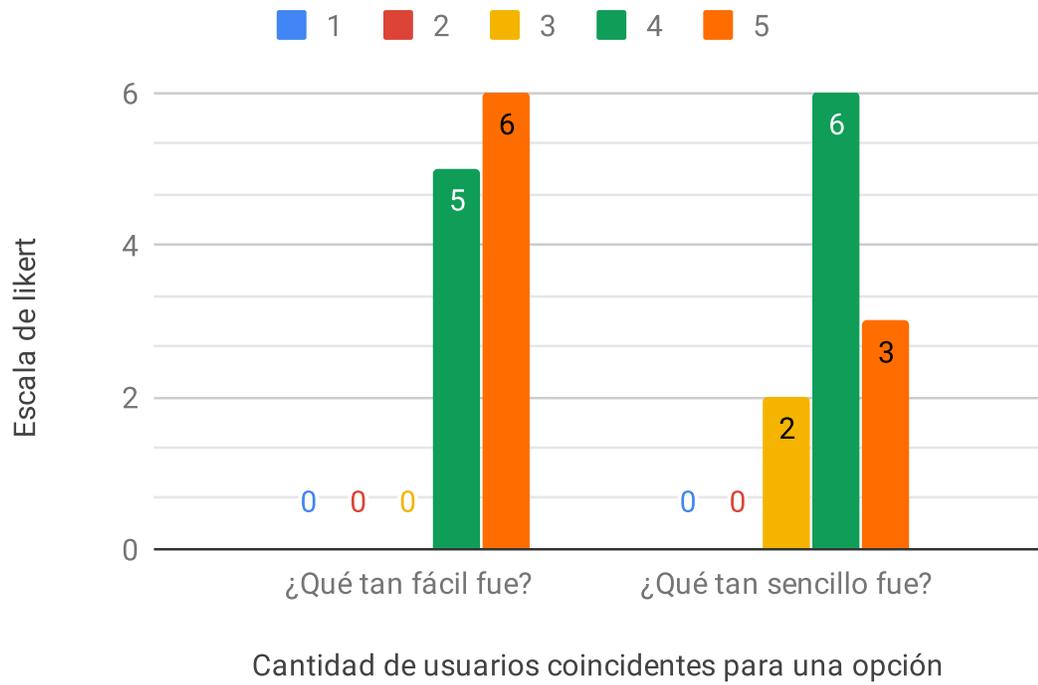


Figura D.71: Test de evaluación: Gráfico de likert asociar una competencia a un curso

Asociar el saber de un curso a una unidad y a la o las actividades

1=Muy difícil y 5=Muy fácil

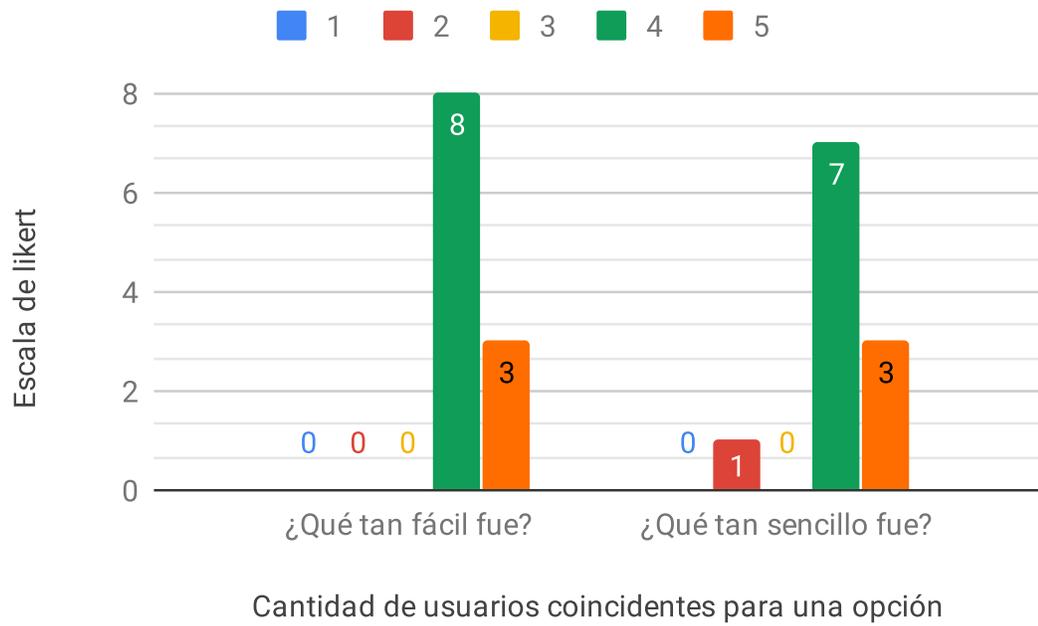


Figura D.72: Test de evaluación: Gráfico de likert asociar el saber de un curso a una unidad y a la o las actividades

Configurar periodo y año

1=Muy difícil y 5=Muy fácil

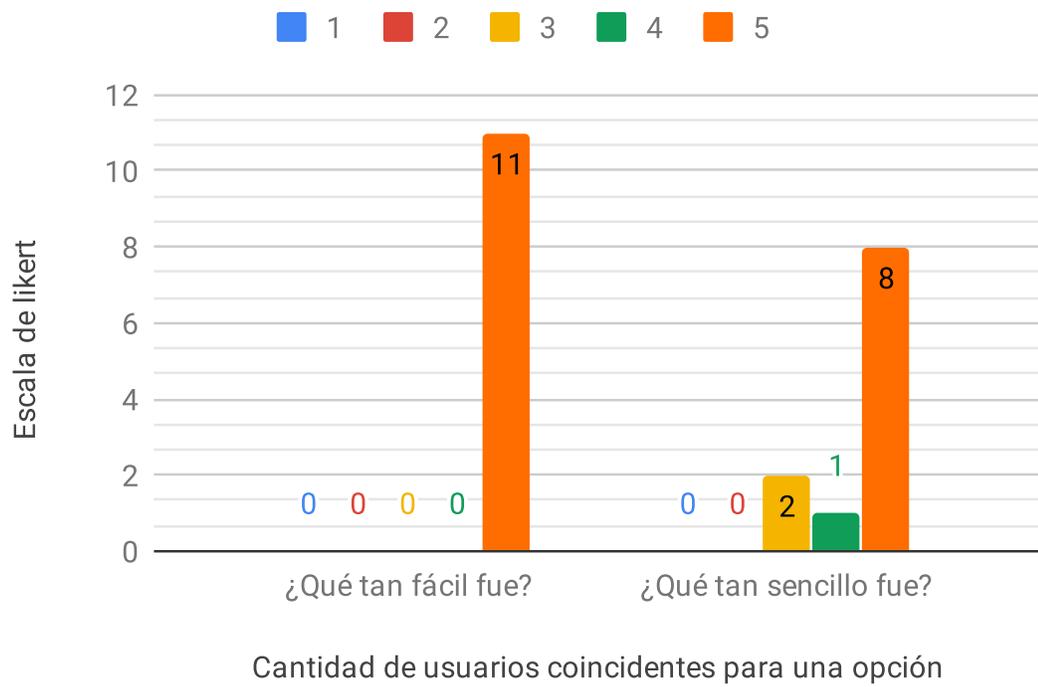


Figura D.73: Test de evaluación: Gráfico de likert configurar periodo y año

Crear un plan de clases

1=Muy difícil y 5=Muy fácil

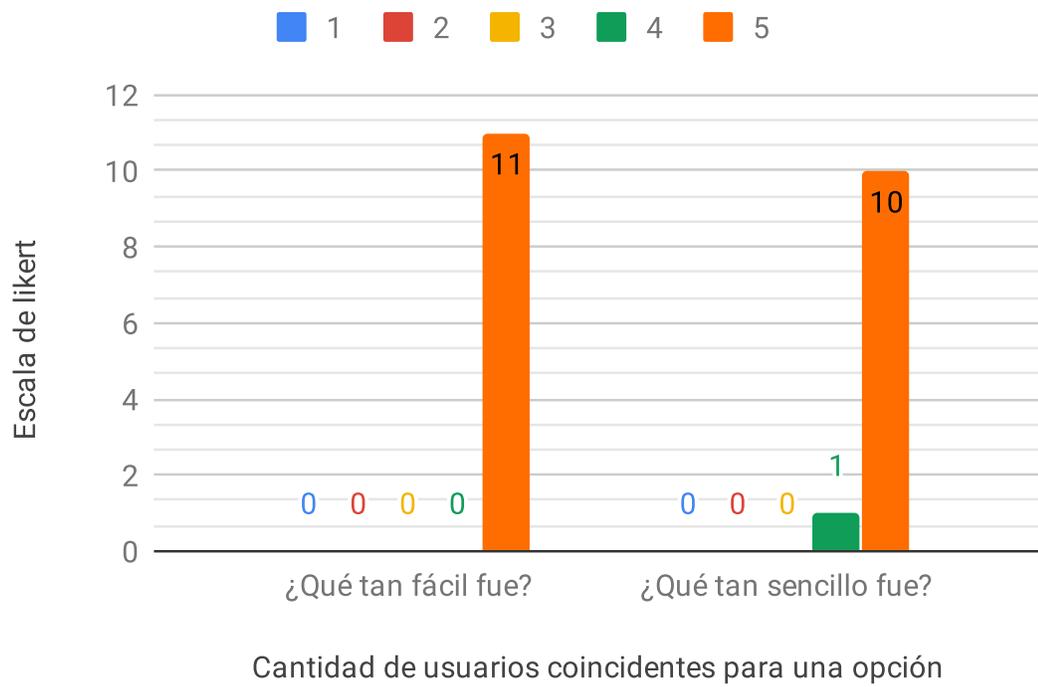


Figura D.74: Test de evaluación: Gráfico de likert guardar plan de clases