



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

PLANIFICADOR DE MALLA CURRICULAR ALTERNATIVA PARA
ESTUDIANTES DE LA FACULTAD DE CIENCIAS FÍSICAS Y
MATEMÁTICAS DE LA UNIVERSIDAD DE CHILE

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERA CIVIL EN COMPUTACIÓN

RUPALI GABRIELA LÓPEZ ECHEVERRÍA

PROFESORA GUÍA:
JOCELYN SIMMONDS WAGEMANN

MIEMBROS DE LA COMISIÓN:
MARÍA CECILIA BASTARRICA PIÑEYRO
HUGO BELTRÁN ALEJOS

SANTIAGO DE CHILE
2023

Resumen

En la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, cada estudiante debe inscribir las asignaturas que cursará durante un semestre específico, basándose en las indicaciones dadas por la malla curricular de su carrera.

Por distintos motivos, es posible que su avance no coincida de manera exacta con lo que estipula la malla. Esta situación produce incertidumbre y obliga a tomar decisiones por cuenta propia.

Por lo tanto, se efectúa el presente trabajo con la finalidad de apoyar a los estudiantes de la facultad que experimentan dificultades como las recién descritas. Se da comienzo al mismo realizando un análisis exhaustivo, cuyo propósito consiste en determinar los requerimientos que mejor satisfacen sus necesidades.

Teniendo en cuenta los resultados del análisis, se opta por diseñar bocetos de interfaces de usuario que permitan estudiar la factibilidad técnica de los requisitos establecidos. De esta forma, se busca disminuir la incertidumbre de manera temprana y acrecentar así las probabilidades de éxito del proyecto.

Posteriormente, se diseña e implementa una aplicación web que posibilita la construcción de una malla curricular alternativa, según el progreso del alumno hasta el momento. Se entiende por malla curricular alternativa, una planificación del conjunto de ramos que se pretende cursar durante los semestres siguientes, considerando las asignaturas que se tienen pendientes. El sistema descrito simplifica el análisis de las opciones disponibles, mediante una visualización dinámica e interactiva. Así, facilita que el estudiante alcance una mayor certeza sobre cómo será su futuro académico.

Al terminar la implementación, se lleva a cabo una fase de validación con usuarios finales, en la cual se evalúa la funcionalidad y usabilidad de la herramienta construida. Por medio de la retroalimentación recibida en esta etapa, se adquiere una importante perspectiva sobre las ventajas y las falencias de la aplicación. Luego, se logra elaborar una serie de conclusiones, las cuales se espera constituyan un pilar fundamental para una posterior extensión del software desarrollado.

*We will laugh about it all
And we'll celebrate the things we've done
The years to come
The good, the bad
The sweet, the sad*

Agradecimientos

Quisiera expresar mis más profundos agradecimientos a un selecto grupo de personas. No lo habría logrado sin ustedes.

A mi queridísima familia, mi mamá, mi papá y mi hermana Camila, por su apoyo incondicional. Por creer en mí y por estar siempre ahí, cuando más les necesitaba.

A los gatitos en mi vida, Luisi, Fibi, Vicky y Robi, les agradezco su compañía. Aunque no lo puedan leer, se merecen este reconocimiento. Además de un suministro inagotable de cariño, y churus, por supuesto.

A mi querida amiga Coty, por tantas risas, por los chismes y los buenos momentos. Gracias por todos los años compartidos.

To my dear friend Candy, for always being there. For the endless chats and her undying support.

A los mentores de mi trayectoria profesional, Gonzalo, Camilo, Alexis, Carlos y Leo, les agradezco su infinita paciencia y toda la sabiduría compartida.

A mi profesora guía Jocelyn, por su ayuda, por sus consejos sensatos y por su buena disposición.

Tabla de Contenido

1. Introducción	1
1.1. Contexto	1
1.2. Definición del problema	2
1.3. Objetivos	3
1.4. Solución propuesta	3
1.4.1. Alcance de la solución propuesta	3
1.4.2. Funcionalidades propuestas	4
1.4.3. Características técnicas de la solución propuesta	5
1.5. Metodología	6
1.6. Estructura del documento	7
2. Estado del Arte	8
2.1. Recursos al interior de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile	8
2.1.1. Malla curricular de la carrera	8
2.1.2. Sección Mállame! de Ucampus	10
2.2. Estado del arte fuera de la facultad	10
2.2.1. Recursos disponibles en otras facultades de la Universidad de Chile	11
2.2.2. Recursos disponibles en otras universidades chilenas	11
3. Análisis y Diseño	15
3.1. Modelo de casos de uso	15

3.1.1.	Especificaciones de casos de uso de alto nivel	15
3.1.2.	Especificaciones de casos de uso expandidas	17
3.1.3.	Diagrama de casos de uso	19
3.2.	Arquitectura del sistema	19
3.3.	Modelo de datos	21
4.	Diseño de Interacción	23
4.1.	Diseño de interfaces de usuario: versión inicial	23
4.1.1.	Interfaces de módulo avance curricular	23
4.1.2.	Interfaces de módulo planificador de malla curricular alternativa	25
4.2.	Diseño de interfaces de usuario: versión final	26
4.2.1.	Interfaz de módulo inicio de sesión	26
4.2.2.	Interfaces de módulo avance curricular	27
4.2.3.	Interfaces de módulo planificador de malla curricular alternativa	29
4.3.	Validación inicial de interfaces diseñadas	34
5.	Implementación	36
5.1.	Especificaciones técnicas	36
5.2.	Automatización ejecución de la aplicación	36
5.2.1.	Tarea <code>default</code> de Gulp	37
5.2.2.	Tarea <code>serve</code> de Gulp	38
5.3.	Estructura y organización de código	39
5.4.	Interfaz planificador de malla curricular alternativa	41
5.4.1.	Archivo de tipo plantilla <code>planner.html</code>	41
5.4.2.	Archivo de tipo controlador <code>planner.controller.js</code>	45
5.5.	Estrategias para preservar las buenas prácticas de programación	53
6.	Validación	56
6.1.	Metodología	56

6.1.1. Evaluación de funcionalidad	56
6.1.2. Evaluación de usabilidad y utilidad	58
6.2. Resultados	59
6.2.1. Usuarios participantes	59
6.2.2. Resultados evaluación de funcionalidad	60
6.2.3. Resultados evaluación de usabilidad	61
6.2.4. Retroalimentación adicional	63
6.3. Interpretación de resultados	65
7. Conclusiones	66
Bibliografía	69

Índice de figuras

2.1.	Extracto de la malla curricular de la carrera de Ingeniería Civil en Computación de la Universidad de Chile.	9
2.2.	Extracto de la sección Mállame! Se encuentra en el apartado “Boletines FCFM” de la plataforma Ucampus, en la pestaña “Recuento Créditos”.	10
2.3.	Página visualizada en el portal Planner luego de acceder mediante la opción “Ingreso Anónimo”.	13
2.4.	Ventana pop-up del portal Planner que exhibe instrucciones de uso, visualizada luego de acceder mediante la opción “Ingreso Anónimo”.	13
3.1.	Diagrama de casos de uso para la solución propuesta.	19
3.2.	Diagrama de arquitectura lógica de la aplicación a implementar.	20
3.3.	Diagrama entidad-relación de modelo de datos para la aplicación.	21
4.1.	Boceto interfaz de selección de versión de la malla curricular correspondiente a la carrera cursada por el usuario (versión inicial). Elaboración propia. . . .	24
4.2.	Boceto interfaz para completar avance curricular del usuario (versión inicial). Elaboración propia.	24
4.3.	Boceto interfaz de avance curricular del usuario, segundo semestre seleccionado (versión inicial). Elaboración propia.	25
4.4.	Boceto interfaz de planificador de malla curricular alternativa. Vista completa, incluyendo sección de ramos no cursados y sección de asignación de ramos por semestre (versión inicial). Elaboración propia.	26
4.5.	Boceto interfaz de inicio de sesión (versión final). Elaboración propia.	27
4.6.	Boceto interfaz de selección de versión de la malla curricular correspondiente a la carrera cursada por el usuario (versión final). Elaboración propia.	27
4.7.	Boceto interfaz para completar avance curricular del usuario, primera parte (versión final). Elaboración propia.	28

4.8. Boceto interfaz para completar avance curricular del usuario, segunda parte (versión final). Elaboración propia.	28
4.9. Boceto interfaz para completar avance curricular del usuario, tercera parte (versión final). Elaboración propia.	29
4.10. Boceto interfaz para completar avance curricular del usuario, cuarta parte (versión final). Elaboración propia.	29
4.11. Boceto interfaz de planificador de malla curricular alternativa. Vista completa, incluyendo sección de ramos no cursados y sección de asignación de ramos por semestre. Incluye tooltip explicativo de ramo prioritario (versión final). Elaboración propia.	30
4.12. Boceto interfaz de planificador de malla curricular alternativa. Vista completa, incluyendo sección de ramos no cursados y sección de asignación de ramos por semestre. Incluye tooltip explicativo de ramo deshabilitado por requisito (versión final). Elaboración propia.	30
4.13. Boceto interfaz de planificador de malla curricular alternativa. Asignación de ramos a semestre en curso y semestre próximo (versión final). Elaboración propia.	31
4.14. Boceto interfaz de planificador de malla curricular alternativa. Asignación de ramos a semestre siguiente (versión final). Elaboración propia.	32
4.15. Boceto interfaz de planificador de malla curricular alternativa. Alerta de rechazo de asignación por incumplir relación de requisitos entre asignaturas (versión final). Elaboración propia.	32
4.16. Boceto interfaz de planificador de malla curricular alternativa. Asignación de ramos a nuevo semestre, avanzando hacia abajo en la sección de asignación de ramos por semestre (versión final). Elaboración propia.	33
4.17. Boceto ventana emergente de configuración de cantidad máxima de créditos por semestre (versión final). Elaboración propia.	33
4.18. Boceto interfaz de planificador de malla curricular alternativa. Alerta de incumplimiento de cantidad máxima de créditos por semestre (versión final). Elaboración propia.	34
5.1. Estructura del directorio temporal resultante de la ejecución de la tarea <code>default</code> de Gulp.	37
5.2. Diagrama de componentes principales del directorio base del proyecto.	40
5.3. Captura de pantalla de la interfaz planificador de malla curricular alternativa al acceder por primera vez a la aplicación (usando un conjunto de ramos pendientes de ejemplo).	41

5.4. Captura de pantalla de un elemento <i>placeholder</i> para un semestre vacío, perteneciente a la sección de ramos pendientes.	43
5.5. Captura de pantalla de un elemento <i>placeholder</i> para un semestre vacío, perteneciente a la sección de asignación de ramos por semestre.	44
5.6. Captura de pantalla del aspecto visual de un componente de asignatura. . .	44
5.7. Captura de pantalla del aspecto visual de un <i>tooltip</i> de asignatura.	44
5.8. Captura de pantalla de un ejemplo del <i>placeholder</i> que se visualiza al arrastrar una asignatura y posarla sobre un semestre.	50
5.9. Ejemplo de alerta de rechazo de asignación de ramo, por incumplimiento de relación de prerrequisito.	53
5.10. Ejemplo de alerta de rechazo de asignación de ramo, por incumplimiento de relación de restricción.	53
5.11. Ejemplo de alerta de rechazo de asignación de ramo, por incumplimiento de la pertenencia a un semestre en la malla.	53
6.1. Gráfico distribución de respuestas a P1 y P2.	62
6.2. Gráfico distribución de respuestas a P3 y P4.	62
6.3. Gráfico distribución de respuestas a P5 y P6.	62
6.4. Gráfico distribución de respuestas a P7 y P8.	63
6.5. Captura de pantalla de la sección de asignación de ramos por semestre, inicialmente vacía.	64

Índice de Tablas

3.1. Especificación de alto nivel para caso de uso número 1.	15
3.2. Especificación de alto nivel para caso de uso número 2.	16
3.3. Especificación de alto nivel para caso de uso número 3.	16
3.4. Especificación de alto nivel para caso de uso número 4.	16
3.5. Especificación de alto nivel para caso de uso número 5.	16
3.6. Especificación de alto nivel para caso de uso número 6.	17
3.7. Especificación de alto nivel para caso de uso número 7.	17
3.8. Especificación expandida para caso de uso número 3.	17
3.9. Especificación expandida para caso de uso número 4.	18
3.10. Especificación expandida para caso de uso número 5.	19

Índice de códigos

5.1. Extracto del programa referente a la definición de la tarea <code>default</code> de Gulp	37
5.2. Extracto del programa referente a la definición de la tarea <code>serve</code> de Gulp . .	39
5.3. Extracto del archivo <code>planner.html</code> que corresponde a la plantilla de la interfaz planificador de malla curricular alternativa	42
5.4. Estructura de un elemento de la respuesta retornada por la función <code>getCurriculum</code> de <code>curriculumService</code>	46
5.5. Extracto de la función <code>getPendingSubjectsPerSemester</code> del servicio de ramos pendientes	47
5.6. Extracto de estructura de la respuesta de <code>getPendingSubjectsPerSemester()</code> del servicio de ramos pendientes	48
5.7. Función <code>getInitialUpcomingSemester</code> del servicio de semestres futuros . .	48
5.8. Ejemplo de respuesta retornada por la función <code>getInitialUpcomingSemester</code> de <code>upcomingSemestersService</code>	49
5.9. Conjunto de métodos <code>listener</code> de la funcionalidad <i>drag and drop</i>	49
5.10. Función <code>dragMoveListener</code> de <code>vm.dragControlListeners</code> perteneciente al controlador del módulo de planificación	50
5.11. Función <code>droppedListener</code> de <code>vm.dragControlListeners</code> perteneciente al controlador del módulo de planificación	50
5.12. Extracto de la función <code>_isSubjectAllowedToDropInSemester</code> del controlador del planificador	52
5.13. Extracto del archivo <code>index.constants.js</code> que ejemplifica la definición de una constante de tipo mensaje	54
5.14. Extracto del archivo <code>index.constants.js</code> que ilustra la definición de una constante de tipo <code>enum</code>	54
5.15. Ejemplo de uso de documentación extraído del archivo <code>planner.controller.js</code>	54

5.16. Extracto del archivo <code>semester.model.js</code> donde se encuentra la definición de la entidad de tipo semestre	55
6.1. Caso de prueba 1	57
6.2. Caso de prueba 2	57
6.3. Caso de prueba 3	57
6.4. Caso de prueba 4	57

Capítulo 1

Introducción

1.1. Contexto

Considere a un estudiante perteneciente a la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile. Al inicio de cada semestre, él o ella debe registrar las asignaturas que cursará durante el período en la plataforma Ucampus. Dicho proceso se conoce como Inscripción Académica.

La malla curricular de la carrera cursada indica cuáles asignaturas corresponde inscribir en un momento dado. Este instrumento sirve como guía para que el estudiante pueda completar su Inscripción Académica eficazmente.

Sin embargo, si él o ella ha eliminado y/o reprobado ramos en el pasado, es pertinente asumir que su avance no coincide de forma exacta con lo estipulado por la malla. En estas circunstancias, la utilidad de la herramienta se ve reducida y el alumno se ve obligado a tomar decisiones por su cuenta.

Usualmente, ser capaz de minimizar el tiempo que se va a atrasar en la carrera es de vital importancia para un estudiante que se encuentre en esa condición. Poder distribuir las materias por semestre de la manera más equitativa posible – en cuánto a cantidad de créditos y carga académica – es también un criterio importante.

En la actualidad, la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile no cuenta con un mecanismo automatizado que simplifique la búsqueda del conjunto de ramos a inscribir óptimo, es decir, el conjunto de ramos que mejor se ajusta a los criterios descritos. Esta falta de guía suele ser causante de estrés e inseguridad.

El alumno debe limitarse a evaluar sus posibilidades manualmente. Pero, seguir este procedimiento presenta una serie de consecuencias potencialmente negativas. El estudiante no tiene la certeza de estar escogiendo la mejor opción, y es posible que cometa errores que le perjudiquen. Puede que inscriba una combinación de materias que le ocasione un retraso adicional en la carrera. Quizás escoja una asignación de ramos que constituye una carga académica excesiva. Afrontar esta situación es sumamente relevante para lograr visibilizar la

experiencia de estudiantes que han reprobado asignaturas.

1.2. Definición del problema

En el contexto particular de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile, se observa que las mallas curriculares de las carreras representan planificaciones notoriamente flexibles. En éstas, se suelen hallar asignaturas asociadas a un semestre específico, pero que no constituyen un prerrequisito para ninguna de las materias del período inmediatamente posterior, o incluso, para las materias de múltiples períodos hacia adelante.

Esto quiere decir que, para un semestre dado, los instrumentos nombrados ofrecen más de una alternativa en cuanto al conjunto de ramos que es factible inscribir. Por consiguiente, se advierte que una desviación del plan original puede generar imprevisibles efectos en cadena.

Con esto en mente, se establece el problema a abordar como el caso de los alumnos y alumnas de la facultad que se encuentran atrasados con respecto a su avance académico proyectado. Al estar siguiendo una malla flexible, estos estudiantes sienten incertidumbre acerca del porvenir. Cada vez que completan su Inscripción Académica, se enfrentan a una gran cantidad de posibilidades, y no consiguen visualizar con claridad el camino que seguirá cada una de ellas a mediano y largo plazo.

Existen diversas causas por las cuales se produce un retraso en la carrera, sin embargo, es frecuente que se deba a la reprobación de uno o más ramos. Asimismo, suspender una materia ocurre por distintos motivos, que no se limitan a experimentar dificultades desde el punto de vista académico. Puede tratarse de motivos de carácter personal, familiares, económicos, de salud mental, entre otros. A pesar de ello, es probable que una reprobación sea vista como un fracaso por los mismos estudiantes, y quizá también por su entorno. En estas circunstancias, es esencial contar con una solución, para alivianar preocupaciones, apoyar a los alumnos y hacerles sentir que lograrán volver a encauzar su trayectoria universitaria.

En el presente informe se plantea el diseño e implementación de una aplicación web que le permita a un alumno o alumna definir una malla curricular alternativa, según su avance hasta el momento. Se entiende por malla curricular alternativa, un plan sobre el conjunto de materias que se pretende cursar durante los semestres siguientes, considerando las asignaturas que se tienen pendientes.

El sistema propuesto simplifica el análisis de las opciones disponibles, mediante una visualización dinámica e interactiva. Así, facilita que el estudiante alcance una mayor certeza sobre cómo será su futuro académico.

Desde el punto de vista de experiencia de usuario, la aplicación debe ser, por sobre todo, usable e intuitiva. En cuanto a calidad de código, se espera que se caracterice por ser un software de alta mantenibilidad y extensibilidad.

Dado que se trata de una primera versión, la solución se considerará aceptable si proporciona las funcionalidades mínimas para construir una planificación de ramos a inscribir en los próximos períodos.

1.3. Objetivos

Objetivo general

El presente proyecto busca crear una aplicación web que simplifique y agilice para un estudiante la tarea de escoger un conjunto de asignaturas a cursar durante un semestre dado. Asimismo, se pretende otorgarle la posibilidad de considerar minuciosamente cada una de sus alternativas.

Objetivos específicos

1. Explicitar el conjunto de casos de uso que la solución abordará. Esto es, detallar las funcionalidades que se proveerán mediante la aplicación.
2. Diseñar los componentes de la herramienta: una arquitectura apta en manejar el flujo de datos eficientemente, una interfaz que satisfaga las funcionalidades establecidas previamente y un modelo de datos capaz de simular una malla curricular de una carrera universitaria.
3. Implementar los componentes diseñados, aplicando técnicas de programación que favorezcan la robustez, mantenibilidad, extensibilidad y usabilidad de la aplicación.
4. Aplicar un método de validación que permita determinar cuán bien se ha logrado cumplir con los objetivos acordados.

1.4. Solución propuesta

Con el propósito de abordar el problema planteado, se propone la creación de una aplicación web que le permita al alumno definir una malla curricular alternativa, según su avance hasta el momento. Esta aplicación se enfocará en simplificar y agilizar para un estudiante la tarea de escoger un conjunto de asignaturas a cursar durante un semestre dado, ofreciendo una herramienta de visualización dinámica e interactiva.

1.4.1. Alcance de la solución propuesta

La solución propuesta está dirigida particularmente a alumnos y alumnas cuyo avance curricular no coincide exactamente con lo que indica la malla de su carrera. Las dificultades enfrentadas por estudiantes que se encuentren en estas circunstancias son transversales a todas las carreras que imparte la facultad.

Sin embargo, no es factible abarcar de manera satisfactoria la totalidad de las carreras, considerando la cantidad de tiempo disponible para el desarrollo del presente proyecto. En consecuencia, se hace indispensable limitar el alcance a un sistema diseñado únicamente para

estudiantes de la carrera de Ingeniería Civil en Computación, dictada por el Departamento de Ciencias de la Computación de la Universidad de Chile.

Es oportuno mencionar que, por la misma razón indicada previamente, no es viable abordar el caso de alumnos que se encuentren cursando dos carreras de forma simultánea.

1.4.2. Funcionalidades propuestas

Para lograr cumplir los objetivos establecidos, se requiere considerar la implementación de una serie de funcionalidades que se pretende ofrecer al usuario, tales como: otorgar herramienta que posibilite ingresar el avance curricular hasta el momento, proporcionar visualización de asignaturas no aprobadas, permitir asignarlas a semestres específicos, respetar relaciones de requisitos entre ellas, y mostrar la suma de créditos asociada a cada semestre. El comportamiento esperado de cada una de estas funcionalidades se detallará a continuación:

- **Avance curricular:** El sistema debe conocer el avance del alumno hasta el momento actual, con respecto a la malla curricular de su carrera, para así ser capaz de identificar cuáles son las asignaturas que él o ella aún no ha aprobado. Dicha información es almacenada por la plataforma Ucampus.

Esta plataforma proporciona una variedad de opciones para que el usuario pueda exportar su avance curricular a un documento. Así, podría implementarse una interfaz que reciba este documento, pero esto implica que la aplicación se volvería dependiente de un sistema externo. Si en el futuro Ucampus realiza modificaciones de formato o contenido, la solución propuesta dejaría de funcionar correctamente.

Por lo tanto, se decide optar por una funcionalidad proporcionada directamente por la aplicación, que posibilite que el alumno complete su avance curricular, seleccionando los ramos de su malla que ha aprobado hasta el momento.

- **Asignaturas no aprobadas:** Considerando lo expuesto previamente, es pertinente afirmar que el sistema puede identificar el conjunto de materias que el estudiante no ha aprobado todavía. Estas asignaturas deben incluirse en la visualización que se muestra al usuario, en forma de rectángulos trasladables mediante *drag and drop*.

Cabe destacar que se planea considerar los ramos electivos de la malla como entidades abstractas. Por lo tanto, sería posible incluirlos en la planificación, incluso si aún no se sabe cuáles serán exactamente.

- **Asignación ramos a semestres específicos:** La interfaz de la aplicación debe contener una visualización de semestres venideros. De esta forma, el usuario puede arrastrar cada ramo – de la categoría de asignaturas no aprobadas – y asignarlo a un período específico.

Este debe ser un proceso iterativo, es decir, se debe posibilitar la asignación de cierto ramo a cierto semestre, seguida del cambio a un semestre diferente. Así, es factible ir modificando cada elección de manera sucesiva, hasta que el usuario se encuentre satisfecho con el resultado.

- **Respetar relación de requisitos entre asignaturas:** En cuanto al proceso descrito previamente, es prioritario que el sistema conozca la relación de requisitos entre asignaturas. De esta modo, podrá rechazar cualquier asignación que transgreda dicha relación.
- **Sumar créditos asociados a cada asignatura:** Durante la asignación de ramos, la aplicación debe ir sumando la cantidad total de créditos correspondiente a cada semestre y debe exhibirla en tiempo real.
También, debe existir una opción para escoger una cantidad máxima de créditos, mediante la cuál el sistema pueda alertar al usuario cuando alguna de sus asignaciones sobrepase este límite.
- **Exportar planificación:** Finalmente, se debe proporcionar un método para exportar un documento resumen de la planificación construida por el estudiante. Se ha determinado, por simplicidad, que el formato de este documento sea `csv`.

1.4.3. Características técnicas de la solución propuesta

Obtención y almacenamiento de datos

Se pretende que la aplicación a implementar exista como una herramienta completamente independiente, en lugar de vincularse con la plataforma Ucampus, para así acrecentar su capacidad de perdurar en el tiempo sin mayores inconvenientes. En consecuencia, los datos relevantes al proceso serán adquiridos directamente del usuario, a excepción de la malla curricular.

Es razonable suponer que la malla curricular de la carrera de Ingeniería Civil en Computación será modificada en innumerables ocasiones a lo largo del tiempo, ya sea debido a actualizaciones de requisitos, cursos, o incluso la aparición de nuevas versiones. Además, la herramienta presenta potencial de ser extendida a otras carreras de la facultad, lo que requeriría la adición de nuevas mallas curriculares.

Teniendo en cuenta estos antecedentes, es de suma importancia garantizar que la obtención y almacenamiento de la malla curricular sea un proceso encapsulado y aislado del resto de la implementación. Por lo tanto, se planea almacenar este documento en forma de archivo de configuración, lo que simplificará la tarea de actualización en caso de eventuales alteraciones.

Tecnologías a utilizar

Las tecnologías que se planea usar para construir la aplicación son: Python y Django para la implementación del *back-end*, junto con AngularJS (framework de JavaScript) para la implementación del *front-end*.

Esta decisión se fundamenta en el hecho de que se cuenta con más de 2.5 años de experiencia desarrollando software de forma profesional, utilizando el *stack* tecnológico mencionado.

Se estima que el conocimiento y las habilidades adquiridas durante ese período serán esenciales para concretar el proyecto exitosamente.

Mantenibilidad y extensibilidad de la solución propuesta

Se aspira a que la solución planteada se caracterice por ser un software de alta mantenibilidad y extensibilidad. Por ello, el *front-end* de la aplicación se constituirá de módulos encapsulados, y se priorizará alcanzar altos niveles de cohesión y bajos niveles de acoplamiento entre los mismos.

Estos módulos serán implementados siguiendo un estilo de arquitectura Modelo-vista-controlador, lo que facilitará evitar duplicaciones de código y reducir bugs. Entonces, se producirá una herramienta con código que, además de ser mantenible y extensible, será legible y entendible por otros desarrolladores.

Adicionalmente, se decide implementar un proceso de ejecución automatizado de la plataforma, con el propósito de generar fases de desarrollo más eficientes y producir código de mayor calidad.

1.5. Metodología

La metodología escogida para utilizarse durante el transcurso del proyecto corresponde a desarrollo iterativo e incremental. Tiene la finalidad de disminuir la incertidumbre de manera temprana, para así acrecentar las probabilidades de éxito. Las etapas que la conforman se describen a continuación:

- **Definición de requerimientos:** Se opta por elaborar un modelo de casos de uso, cuyo contenido se detalla en la Sección 3.1. Éste permite establecer los requerimientos del software de forma explícita y, por consiguiente, especifica las características *core* y las restricciones que debe cumplir la herramienta a implementar.
- **Definición de diseño de interacción:** Posteriormente, se diseña una serie de prototipos que representan a las principales interfaces de la aplicación. Dichos prototipos se exhiben más adelante, en la Sección 4.2. Su propósito es clarificar potenciales requisitos ambiguos y analizar la factibilidad técnica de los mismos.
- **Proceso de validación:** Se decide realizar dos fases de validación: la primera, después de crear los bocetos mencionados anteriormente, y la segunda, luego de finalizar la implementación de la solución.

La validación inicial – expuesta en la Sección 4.3 – pretende determinar si las ideas propias se corresponden con las expectativas de los usuarios y sus modelos mentales. Por otra parte, la validación final – detallada en el Capítulo 6 – tiene el objetivo de evaluar la usabilidad y utilidad de la aplicación, percibida por los usuarios finales.

- **Proceso de implementación:** Para finalizar, se procede con la implementación de la herramienta, descrita en profundidad en el Capítulo 5. Esta etapa consiste fundamentalmente en desarrollar código, y para ello, se emplea el programa Visual Studio Code. El versionamiento es manejado mediante la creación de un repositorio en GitLab, hacia el cual se efectúan *commits* recurrentemente.

1.6. Estructura del documento

Los siguientes capítulos del documento se encargan de describir cada una de las etapas que componen el presente trabajo. En el Capítulo 2, se especifica el resultado de la investigación sobre el estado del arte referente al problema abordado. Particularmente, se identifican los recursos existentes para manejar dicho problema en las demás facultades de la Universidad de Chile y el resto de las universidades chilenas.

El Capítulo 3 se refiere al análisis de requerimientos, efectuado por medio de la elaboración de un modelo de casos de uso. Además, se exhibe el diseño de ciertos componentes de la aplicación, tales como la arquitectura del sistema y el modelo de datos.

En el Capítulo 4 se detalla el proceso de confeccionar las interfaces de usuario para la herramienta, indicando una versión inicial y una versión final, con el propósito de justificar algunas decisiones de diseño. También, se describe el proceso de validación inicial que se lleva a cabo sobre las interfaces mencionadas.

El Capítulo 5 abarca aspectos fundamentales que se relacionan a la implementación misma de la herramienta. Detalla las especificaciones técnicas, explica cómo se logra automatizar la ejecución de la aplicación, describe la estructura y organización del código, destaca los programas más relevantes para el funcionamiento de la interfaz principal del planificador de malla curricular alternativa, y se refiere brevemente a estrategias empleadas con el propósito de preservar las buenas prácticas de programación.

El Capítulo 6 se trata sobre el proceso de validación final realizado. Se especifica la metodología utilizada, la cual consta de una evaluación de funcionalidad y una evaluación de usabilidad y utilidad. Además, se mencionan los resultados conseguidos y se proporciona una interpretación de los mismos.

Para finalizar, el Capítulo 7 contiene las conclusiones del trabajo, en varios ámbitos: cumplimiento de objetivos, resultados obtenidos, relevancia del trabajo realizado, lecciones aprendidas y potenciales trabajos futuros.

Capítulo 2

Estado del Arte

En el presente capítulo se describe el estado del arte pertinente al problema planteado, es decir, los recursos o metodologías que son usados en distintos contextos, similares al propio, para abordar el mismo problema.

En particular, la Sección 2.1 se refiere a las herramientas disponibles para orientar a un estudiante de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile durante su Inscripción Académica. Por otra parte, en la Sección 2.2 se detallan los instrumentos utilizados fuera de la facultad, ya sea en otras facultades de la Universidad de Chile, o en el resto de las universidades chilenas.

2.1. Recursos al interior de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile

Hoy en día, un estudiante que requiere de orientación al momento de efectuar su proceso de Inscripción Académica cuenta con dos recursos a su disposición: la malla curricular de la carrera y la sección Mállame! de la plataforma Ucampus. Ambos recursos se discuten en esta sección.

2.1.1. Malla curricular de la carrera

La malla curricular de la carrera es un documento que señala la asignación de asignaturas que un alumno debe cursar cada período, durante la totalidad de su estancia en la universidad. Tal como se observa en la Figura 2.1, dicha asignación es representada mediante un diagrama secuencial. Éste decreta las materias que se deben inscribir en cada semestre, incluyendo materias obligatorias y electivas.

Adicionalmente, se establecen relaciones de requisitos entre asignaturas. El hecho de que una asignatura sea requisito de otra quiere decir que no está permitido cursar la segunda sin haber aprobado antes la primera, en un semestre anterior al actual.

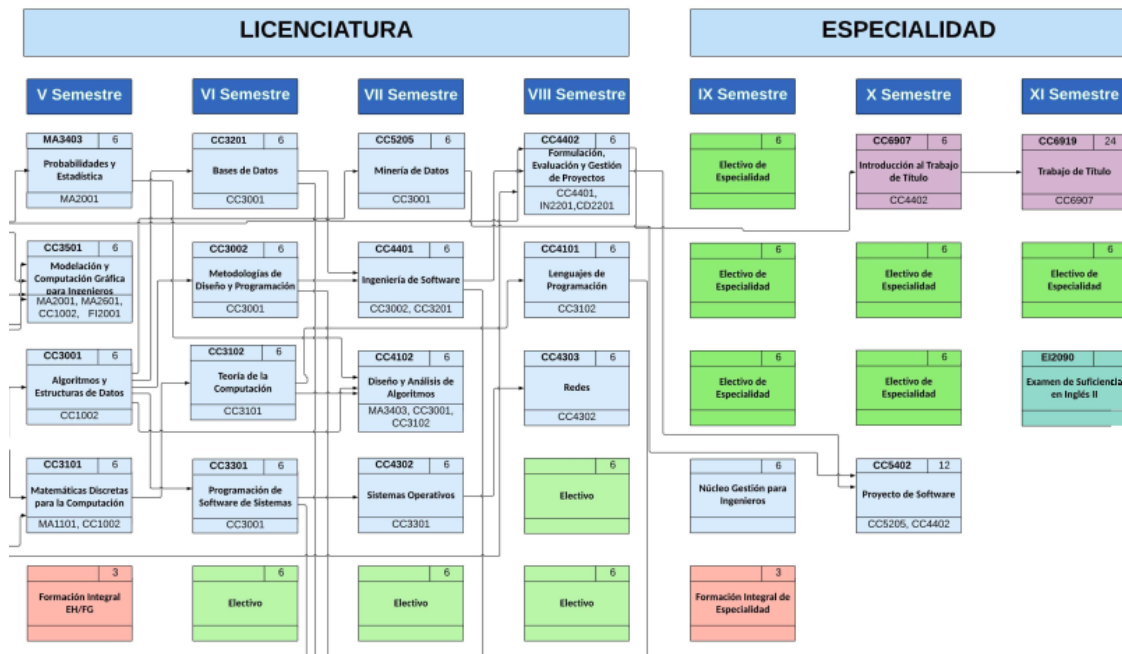


Figura 2.1: Extracto de la malla curricular de la carrera de Ingeniería Civil en Computación de la Universidad de Chile.

En el caso de un estudiante que no ha reprobado ninguna asignatura y ha cumplido exactamente con lo que indica la malla curricular, continuar utilizándola como material de apoyo no presenta inconvenientes.

Por el contrario, ocupar el mismo medio se vuelve trabajoso para un alumno que ha reprobado al menos un ramo en el pasado. Reprobar un ramo implica la obligación de volver a cursarlo en el semestre próximo. Si ha reprobado una asignatura requisito de otra que pertenece al semestre siguiente, no podrá inscribir la segunda hasta que apruebe la primera. Esto causa un efecto dominó en la planificación curricular del estudiante. En consecuencia, lo que expone la malla deja de ser válido para él.

La misma condición podría darse incluso si el ramo reprobado no es requisito de ninguna materia, puesto que también se debe tener en cuenta la carga académica asociada. Habitualmente, registrar un ramo adicional a los señalados en la malla resulta en una carga excesiva.

Por lo tanto, el alumno debe navegar esos obstáculos, e intentar acomodar su asignación de materias de la mejor manera posible. Llevar a cabo esta tarea de forma manual es notablemente complejo. Es esencial respetar las relaciones de requisitos, y no es factible lograr evaluar todas las posibilidades existentes, analizando sus efectos de manera efectiva y completa.

2.1.2. Sección Mállame! de Ucampus

En el apartado “Boletines FCFM” de la plataforma Ucampus se encuentra la sección Mállame! Un caso de ejemplo de lo expuesto en dicha sección se encuentra en la Figura 2.2.

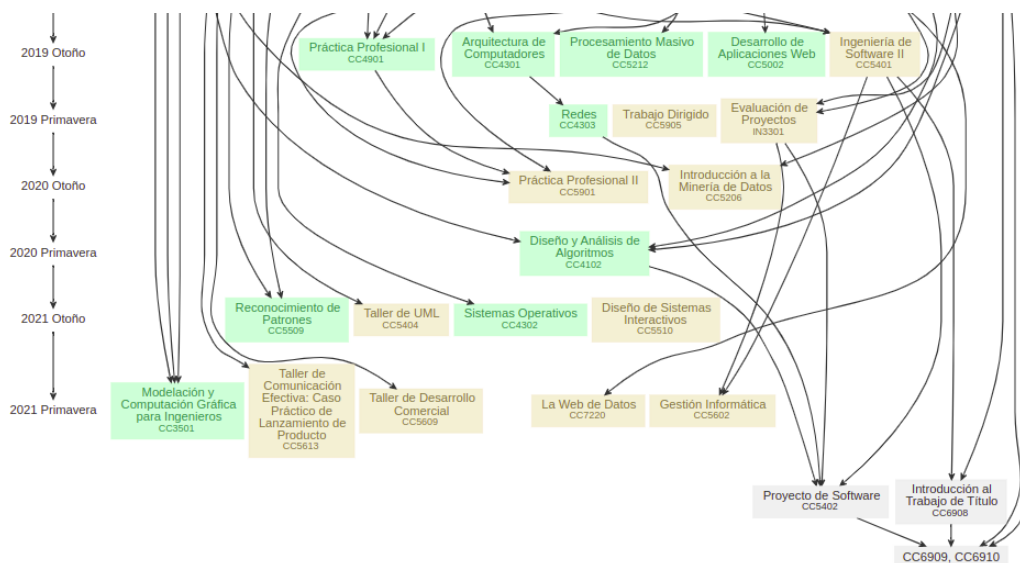


Figura 2.2: Extracto de la sección Mállame! Se encuentra en el apartado “Boletines FCFM” de la plataforma Ucampus, en la pestaña “Recuento Créditos”.

Esta herramienta consiste en una visualización, la cual exhibe el conjunto de asignaturas obligatorias que el alumno tiene permitido inscribir, considerando su avance curricular hasta el momento.

Para el caso de un estudiante que ha reprobado al menos un ramo en el pasado, la sección Mállame es ligeramente más útil que la malla curricular para guiar la Inscripción Académica. Identifica claramente cuáles son las materias que se le permiten cursar al alumno, según su avance en la carrera.

Sin embargo, no permite barajar opciones. Entrega únicamente la asignación de ramos para el semestre dado y no permite contemplar efectos a futuro. Por consiguiente, este recurso – al igual que el anterior – no es idóneo para un estudiante que haya reprobado al menos una asignatura en el pasado. El alumno debe enfrentarse a los mismos obstáculos mencionados en la sección previa.

2.2. Estado del arte fuera de la facultad

La necesidad de orientación al momento de efectuar la toma de ramos es un hecho que trasciende a los estudiantes de la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile. De igual forma, los estudiantes de otras facultades y otras universidades cuentan con algunos recursos de apoyo para el proceso, los cuáles se exhiben más adelante.

2.2.1. Recursos disponibles en otras facultades de la Universidad de Chile

El procedimiento de inscripción de ramos es diferente para cada una de las distintas facultades de la Universidad de Chile.

En algunas, tales como la Facultad de Economía y Negocios y la Facultad de Arquitectura y Urbanismo, se establecen restricciones de cantidad mínima y máxima de créditos que se permite inscribir en un semestre dado [9] [12]. Además, la FEN estipula un mínimo de asignaturas nuevas que se debe inscribir (asignaturas no cursadas previamente), y al reprobar un curso, exige que éste sea realizado por segunda vez en el semestre siguiente [9]. Es razonable asumir que la implantación de dichas limitaciones sobre la toma de ramos genera efectos similares a los efectos de una malla curricular rígida.

Por otra parte, cada facultad ha confeccionado una especie de instructivo – con el propósito de orientar a sus estudiantes durante la inscripción de asignaturas – el cual es accesible mediante el portal web. En la Facultad de Economía y Negocios, éste se materializa en forma de vídeos cortos [9], mientras que en la Facultad de Arquitectura y Urbanismo éste consta de una compilación de preguntas frecuentes [12]. La Facultad de Ciencias Sociales proporciona un documento con instrucciones paso a paso [7]. Si bien se proveen recursos de apoyo, éstos suelen enfocarse en la resolución de dudas concernientes al procedimiento mismo de toma de ramos, y no suelen aludir a decisiones de elección de asignaturas.

Con respecto a sistemas web, se sabe que la FEN utiliza su propio portal de inscripción [8], mientras que otras facultades, como la FAU [12] y la FACSIO [7] ocupan la plataforma Ucampus. Durante la presente investigación no se hallaron referencias a otras herramientas tecnológicas relacionadas al proceso.

2.2.2. Recursos disponibles en otras universidades chilenas

Recurso disponible en todas las universidades acreditadas del país

En la Sección 2.1.1 se recalca la importancia de la malla curricular de la carrera como instrumento guía de la Inscripción Académica, al igual que sus limitaciones. Es pertinente mencionar que este documento está incluido en el plan de estudios de una carrera universitaria [2] y como tal, forma parte de los criterios de evaluación para la acreditación de carreras profesionales de la Comisión Nacional de Acreditación [1].

Por consiguiente, la malla curricular es una herramienta que se encuentra a disposición de todos los alumnos y alumnas de universidades acreditadas a lo largo del país.

La utilidad del instrumento durante la inscripción de asignaturas depende en gran medida de la flexibilidad del programa. Una malla curricular rígida implica que un estudiante que ha reprobado ramos no se enfrenta a un rango de opciones en cuanto a su asignación de ramos futura. En ese caso, acudir a la malla en busca de orientación suele ser satisfactorio.

Por el contrario, mientras más flexible sea la malla, el alumno posee más alternativas de asignación de ramos futura. Así, él o ella experimenta el mismo tipo de dificultades descritas en la Sección 2.1.1.

Instructivos y portales de toma de ramos

Al igual que las facultades nombradas en la Sección 2.2.1, varias universidades chilenas ofrecen instructivos que buscan apoyar a sus estudiantes mientras completan su inscripción de asignaturas para un semestre dado.

En la Universidad Técnica Federico Santa María se provee de un manual referente al proceso de matrícula e inscripción de asignaturas [15], el cual contiene una serie de instrucciones minuciosas paso a paso. La Universidad Andrés Bello proporciona un tutorial de toma de ramos [11], y adicionalmente cuenta con una sección de preguntas frecuentes alusivas a la inscripción de ramos en su portal web [10]. La Universidad del Desarrollo presenta una selección de vídeos bajo la categoría “Ayuda con tu toma de ramos”, los cuales tratan temas tales como “¿Qué es la prioridad académica?” y “¿Cómo postular a la inscripción de asignaturas de manera correcta?”, entre otros [14].

Asimismo, las universidades emplean diversas plataformas web que facilitan la inscripción de materias a cursar durante cada período. Por ejemplo, en la Pontificia Universidad Católica de Chile se utiliza un sistema llamado Banner UC. Éste establece horarios de inscripción, los cuales se asignan a los alumnos en orden de rendimiento académico. Si se cumplen los requisitos del curso, se conceden vacantes de manera instantánea y se otorgan resultados inmediatos [6].

La Universidad Técnica Federico Santa María dispone de una plataforma que se conoce como SIGA. Ésta entrega un sinnúmero de servicios que involucran a la comunidad universitaria, entre los cuales se incluye la preinscripción e inscripción de asignaturas, horario personal, resumen académico y la ficha personal [16].

La Universidad de los Andes cuenta con una aplicación llamada miUANDES, mediante la cual los estudiantes deben inscribir los ramos que van a cursar en cada semestre. Además, en su portal web se puede acceder a un vídeo que ilustra el uso correcto de la aplicación [13].

Herramientas tecnológicas de apoyo

Del mismo modo que ocurre con las facultades mencionadas en la Sección 2.2.1, es usual que las universidades chilenas suministren a sus estudiantes con recursos de apoyo para la toma de ramos, pero éstos suelen enfocarse en la resolución de dudas concernientes al procedimiento mismo, sin hacer alusión a decisiones de elección de asignaturas.

Sin embargo, la presente indagación ha descubierto en particular una universidad que posee una herramienta tecnológica orientada a simplificar el proceso de escoger las asignaturas que se cursará: la Pontificia Universidad Católica de Chile.

**Planificador Currículo 2013 (Vs. 2018)
Para estudiantes de admisiones 2013 a 2019**

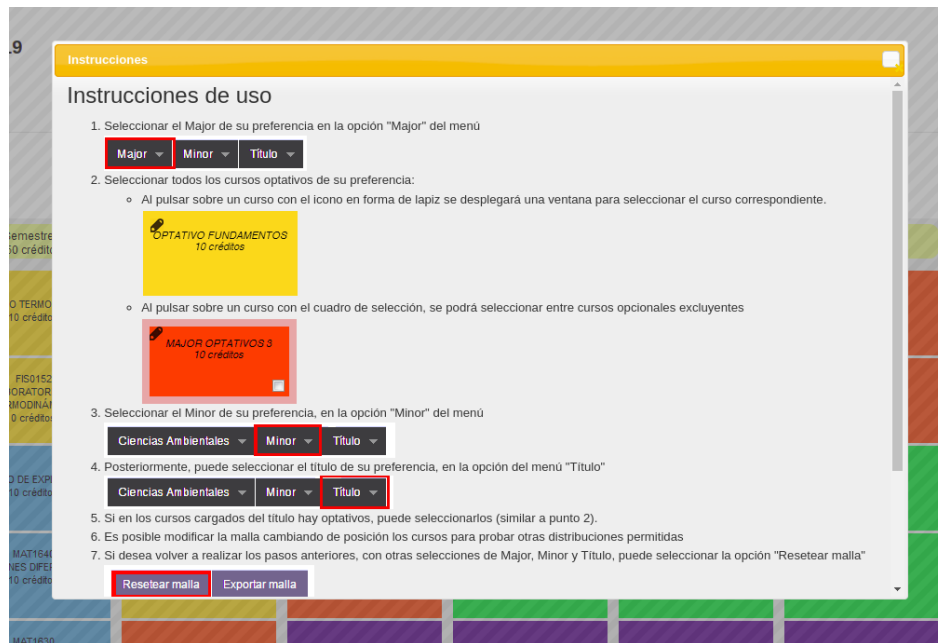
Licenciatura en Ciencias de la Ingeniería. Títulos Ingeniería.

Semestre 1 50 créditos	Semestre 2 50 créditos	Semestre 3 50 créditos	Semestre 4 50 créditos	Semestre 5 50 créditos	Semestre 6 50 créditos	Semestre 7 50 créditos	Semestre 8 50 créditos
QM100A QUÍMICA GENERAL I 10 créditos	IS1513 INTRODUCCIÓN A LA ECONOMÍA 10 créditos	OPTATIVO TERMODINÁMICA 10 créditos	OPTATIVO ELECTRICIDAD Y MAGNETISMO 10 créditos	IN2030 INVESTIGACIÓN, INNOVACIÓN Y EMPRENDIMIENTO 10 créditos	CURSO DE MAJOR 10 créditos	CURSO DE MAJOR 10 créditos	CAPSTONE CURSO DE MAJOR 10 créditos
MAT1810 CÁLCULO I 10 créditos	IC1103 INTRODUCCIÓN A LA PROGRAMACIÓN 10 créditos	FIS1152 LABORATORIO DE TERMODINÁMICA 8 créditos	FIS0153 LABORATORIO DE ELECTRICIDAD Y MAGNETISMO 8 créditos	OPTATIVO FUNDAMENTOS 10 créditos	CURSO DE MAJOR 10 créditos	CURSO DE MAJOR 10 créditos	CURSO DE MAJOR 10 créditos
IN1004 DESAFÍOS DE LA INGENIERÍA 10 créditos	MAT1620 CÁLCULO II 10 créditos	OPTATIVO DE EXPLORACIÓN 10 créditos	OPTATIVO BIOLÓGICO 10 créditos	CURSO DE MAJOR 10 créditos	CURSO DE MINOR 10 créditos	CURSO DE MAJOR 10 créditos	CURSO DE MINOR 10 créditos
MAT200 ÁLGEBRA LINEAL 10 créditos	OPTATIVO ESTÁTICA Y DINÁMICA 10 créditos	EQUACIONES DIFERENCIALES 10 créditos	E110115 PROBABILIDADES Y ESTADÍSTICA 10 créditos	CURSO DE MAJOR 10 créditos	CURSO DE MINOR 10 créditos	CURSO DE MINOR 10 créditos	CURSO DE MINOR 10 créditos
EL189 ÉTICA PARA INGENIEROS 10 créditos	FIS1151 LABORATORIO DE ESTÁTICA Y DINÁMICA 8 créditos	MAT1620 CÁLCULO II 10 créditos	CURSO DE MAJOR 10 créditos	OPTATIVO FORMACIÓN GENERAL 10 créditos	OPTATIVO FORMACIÓN GENERAL 10 créditos	OPTATIVO FORMACIÓN GENERAL 10 créditos	OPTATIVO FORMACIÓN GENERAL 10 créditos
	LET0003 DESARROLLO DE HABILIDADES COMUNICATIVAS ORALES	OPTATIVO FORMACIÓN GENERAL	OPTATIVO FORMACIÓN GENERAL				

Figura 2.3: Página visualizada en el portal Planner luego de acceder mediante la opción “Ingreso Anónimo”.

En la Facultad de Ingeniería de dicha universidad, los alumnos disponen de un portal web llamado Planner. Tal como se observa en la Figura 2.3, Planner es una herramienta de visualización dinámica e interactiva que permite que el usuario analice distintas alternativas de asignación de ramos a semestres, con el propósito de obtener su planificación curricular.

La plataforma incluye información complementaria, simbología e instrucciones de uso. Las últimas se muestran en la Figura 2.4, a continuación.



Instrucciones de uso

1. Seleccionar el Major de su preferencia en la opción "Major" del menú
 Major Minor Título
2. Seleccionar todos los cursos optativos de su preferencia:
 - Al pulsar sobre un curso con el icono en forma de lápiz se desplegará una ventana para seleccionar el curso correspondiente.
 - Al pulsar sobre un curso con el cuadro de selección, se podrá seleccionar entre cursos opcionales excluyentes
3. Seleccionar el Minor de su preferencia, en la opción "Minor" del menú
 Ciencias Ambientales Minor Título
4. Posteriormente, puede seleccionar el título de su preferencia, en la opción del menú "Título"
 Ciencias Ambientales Minor Título
5. Si en los cursos cargados del título hay optativos, puede seleccionarlos (similar a punto 2).
6. Es posible modificar la malla cambiando de posición los cursos para probar otras distribuciones permitidas
7. Si desea volver a realizar los pasos anteriores, con otras selecciones de Major, Minor y Título, puede seleccionar la opción "Resetear malla"

Resetear malla Exportar malla

Figura 2.4: Ventana pop-up del portal Planner que exhibe instrucciones de uso, visualizada luego de acceder mediante la opción “Ingreso Anónimo”.

A pesar de ser un instrumento valioso para los alumnos que requieren evaluar distintas alternativas de planificación curricular, se advierte que Planner presenta algunas deficiencias en términos de usabilidad. La información presentada es de carácter complejo, y se encuentra organizada de forma abrumadora. El solo hecho de necesitar instrucciones de uso sugiere que la interfaz no es particularmente intuitiva.

Capítulo 3

Análisis y Diseño

En el presente capítulo se exhiben las etapas que conforman el proceso de análisis y diseño para el proyecto, las cuales son: modelo de casos de uso (Sección 3.1), arquitectura del sistema (Sección 3.2) y modelo de datos para la aplicación (Sección 3.3).

3.1. Modelo de casos de uso

Las funcionalidades propuestas para la plataforma a implementar se han descrito en la Sección 1.4. Éstas se formalizan seguidamente, mediante el desarrollo de un modelo de casos de uso.

Un modelo de casos de uso es un modelo de las funciones previstas del sistema y su entorno, y está compuesto por varios documentos narrativos que describen la secuencia de eventos que ocurren al completar un proceso [3].

3.1.1. Especificaciones de casos de uso de alto nivel

Luego, las Tablas 3.1 a 3.7 presentan las especificaciones de casos de uso de alto nivel para la solución planteada. Esto es, un conjunto de descripciones conceptuales de los casos de uso identificados.

Caso de uso	CU1: Crear cuenta
Actores	Estudiante
Tipo	Primario
Descripción	Cuando el estudiante ingresa a la aplicación por primera vez, él o ella crea una cuenta, obteniendo así sus credenciales de autenticación.

Tabla 3.1: Especificación de alto nivel para caso de uso número 1.

Caso de uso	CU2: Iniciar sesión
Actores	Estudiante
Tipo	Primario
Descripción	Cada vez que el estudiante accede a la plataforma, él o ella inicia sesión utilizando las credenciales adquiridas luego de crear su cuenta.

Tabla 3.2: Especificación de alto nivel para caso de uso número 2.

Es pertinente mencionar que, en el caso de uso número 2 – expuesto en la Tabla 3.2 – se incluye un aspecto de seguridad primordial para el funcionamiento de la herramienta: la autenticación del usuario. Se estima que, potencialmente, esta acción podría ser facilitada mediante el enlace de la aplicación diseñada al sistema de autenticación de la plataforma Ucampus.

Caso de uso	CU3: Completar avance curricular
Actores	Estudiante
Tipo	Primario
Descripción	Posterior a iniciar sesión por primera vez, el estudiante completa su avance curricular, seleccionando las asignaturas de la malla de su carrera que ha aprobado hasta el momento.

Tabla 3.3: Especificación de alto nivel para caso de uso número 3.

Caso de uso	CU4: Crear planificación de malla curricular alternativa
Actores	Estudiante
Tipo	Primario
Descripción	Luego de completar su avance curricular, el estudiante visualiza el conjunto de asignaturas que aún no ha aprobado. Cada una de ellas es asignada a un semestre venidero específico. Este es un proceso iterativo, es decir, se posibilita la asignación de cierto ramo a cierto semestre, seguida del cambio a un semestre diferente, para todas las asignaturas no aprobadas. Así, es factible modificar cada elección de manera sucesiva, hasta que el estudiante se encuentra satisfecho con el resultado.

Tabla 3.4: Especificación de alto nivel para caso de uso número 4.

Caso de uso	CU5: Asignar un ramo a un semestre específico
Actores	Estudiante
Tipo	Primario
Descripción	Durante la creación de una planificación, se le muestran al estudiante los ramos que aún no ha aprobado. Él o ella selecciona un ramo específico para asignarlo a un semestre dado. Esta acción causa una actualización del conteo de créditos del semestre en cuestión.

Tabla 3.5: Especificación de alto nivel para caso de uso número 5.

Caso de uso	CU6: Configurar máximo de créditos
Actores	Estudiante
Tipo	Secundario
Descripción	Cuando el estudiante se encuentra creando una planificación, visualiza una sección que le permite configurar una cantidad máxima de créditos para cada semestre.

Tabla 3.6: Especificación de alto nivel para caso de uso número 6.

Caso de uso	CU7: Exportar planificación
Actores	Estudiante
Tipo	Secundario
Descripción	Una vez que el estudiante se encuentra satisfecho con el resultado de su asignación de ramos, descarga un documento resumen de la planificación construida.

Tabla 3.7: Especificación de alto nivel para caso de uso número 7.

3.1.2. Especificaciones de casos de uso expandidas

Una especificación de caso de uso expandida es una descripción en detalle del flujo de eventos del caso de uso [3].

Posteriormente se muestran las especificaciones de casos de uso expandidas para los casos de uso número 3, 4 y 5 (en las Tablas 3.8, 3.9 y 3.10, respectivamente). Se desarrollan únicamente los casos de uso mencionados puesto que son los más complejos y más relevantes para el funcionamiento del sistema completo.

Caso de uso	CU3: Completar avance curricular	
Actores	Estudiante	
Propósito	Completar avance curricular hasta el momento	
Resumen	Posterior a iniciar sesión por primera vez, el estudiante completa su avance curricular, seleccionando las asignaturas de la malla de su carrera que ha aprobado hasta el momento.	
Tipo	Primario y esencial	
Ref. Cruzadas	CU1, CU2, CU4	
Curso normal		
	Acciones de los actores	Respuestas del sistema
		1. Sistema muestra lista de carreras impartidas por la facultad.
	2. Estudiante elige carrera que se encuentra cursando.	3. Sistema exhibe lista de versiones de malla curricular asociadas a la carrera escogida.
	4. Estudiante selecciona versión de malla que le corresponde.	5. Sistema muestra ramos pertenecientes a la malla escogida.
	6. Estudiante selecciona ramos que ha aprobado hasta el momento.	7. Sistema almacena avance curricular del estudiante.

Tabla 3.8: Especificación expandida para caso de uso número 3.

Caso de uso	CU4: Crear planificación de malla curricular alternativa	
Actores	Estudiante	
Propósito		
Resumen	<p>Luego de completar su avance curricular, el estudiante visualiza el conjunto de asignaturas que aún no ha aprobado. Cada una de ellas es asignada a un semestre venidero específico.</p> <p>Este es un proceso iterativo, es decir, se posibilita la asignación de cierto ramo a cierto semestre, seguida del cambio a un semestre diferente, para todas las asignaturas no aprobadas. Así, es factible modificar cada elección de manera sucesiva, hasta que el estudiante se encuentra satisfecho con el resultado.</p>	
Tipo	Primario y esencial	
Ref. Cruzadas	CU3, CU5, CU6	
Curso normal		
Acciones de los actores		Respuestas del sistema
		1. Sistema muestra ramos que el estudiante aún debe aprobar y una visualización de semestres futuros junto a su conteo de créditos (inicialmente 0).
2. Estudiante selecciona un ramo específico y lo asigna a un semestre específico.		3. Sistema actualiza el conteo de créditos del semestre específico, sumándole la cantidad de créditos correspondiente a la asignatura asignada.
4. Estudiante repite el paso 2 hasta que ha asignado todos los ramos.		
Cursos alternos		
(2a) Estudiante intenta seleccionar una asignatura para la cual aún no cumple los requisitos.		
(3a) Sistema no le permite realizar la acción.		
(4a) Estudiante repite el paso 2a con un ramo que si cumple los requisitos y el CU continúa normalmente.		
(3b) Además de actualizar el conteo de créditos, la asignación del ramo desbloquea a otros de los cuáles era requisito, y el CU continúa normalmente.		
(3c) Además de actualizar el conteo de créditos, éste se destaca, debido a que supera la máxima cantidad de créditos a inscribir, configurada por el estudiante, y el CU continúa normalmente.		

Tabla 3.9: Especificación expandida para caso de uso número 4.

Caso de uso	CU5: Asignar un ramo a un semestre específico	
Actores	Estudiante	
Propósito	Asignar un ramo específico a un semestre futuro	
Resumen	<p>Durante la creación de una planificación, se le muestran al estudiante los ramos que aún no ha aprobado. Él o ella selecciona un ramo específico para asignarlo a un semestre dado. Esta acción causa una actualización del conteo de créditos del semestre en cuestión.</p>	
Tipo	Primario y esencial	
Ref. Cruzadas	CU4	
Curso normal		
Acciones de los actores		Respuestas del sistema
1. Estudiante selecciona un ramo específico para asignarlo a cierto semestre.		2. La acción se realiza exitosamente y el sistema actualiza el conteo de créditos de dicho semestre, sumándole la cantidad de créditos correspondiente al ramo recién asignado.

Cursos alternos
(1a) Estudiante intenta seleccionar una asignatura para la cual aún no cumple los requisitos.
(2a) Sistema no le permite realizar la acción y termina el CU.
(2b) Además de actualizar el conteo de créditos, la asignación del ramo desbloquea a otros de los cuáles era requisito.
(2c) Además de actualizar el conteo de créditos, éste se destaca, debido a que supera la máxima cantidad de créditos a inscribir, configurada por el estudiante, y el CU continúa normalmente.

Tabla 3.10: Especificación expandida para caso de uso número 5.

3.1.3. Diagrama de casos de uso

El diagrama de casos de uso observado en la Figura 3.1 permite visualizar cómo los actores satisfacen sus necesidades a través de los diferentes casos de uso. Dicho diagrama facilita contemplar el panorama funcional del sistema completo.

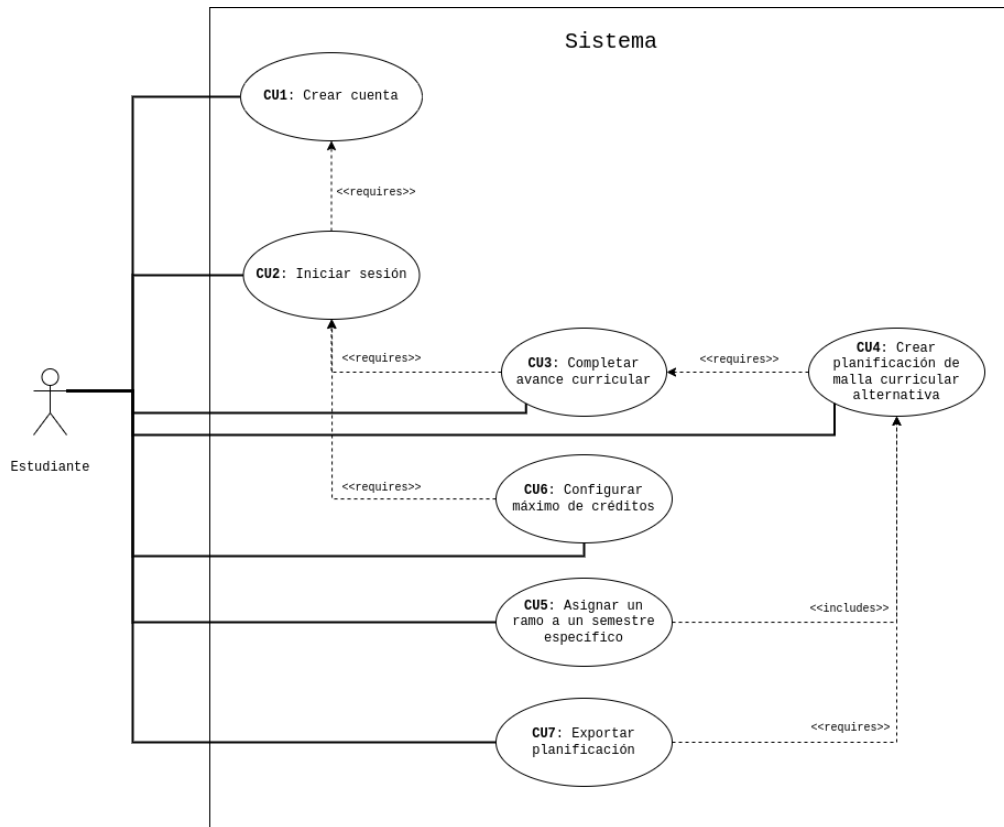


Figura 3.1: Diagrama de casos de uso para la solución propuesta.

3.2. Arquitectura del sistema

Más adelante se expone el diseño arquitectural de alto nivel para la aplicación a construir.

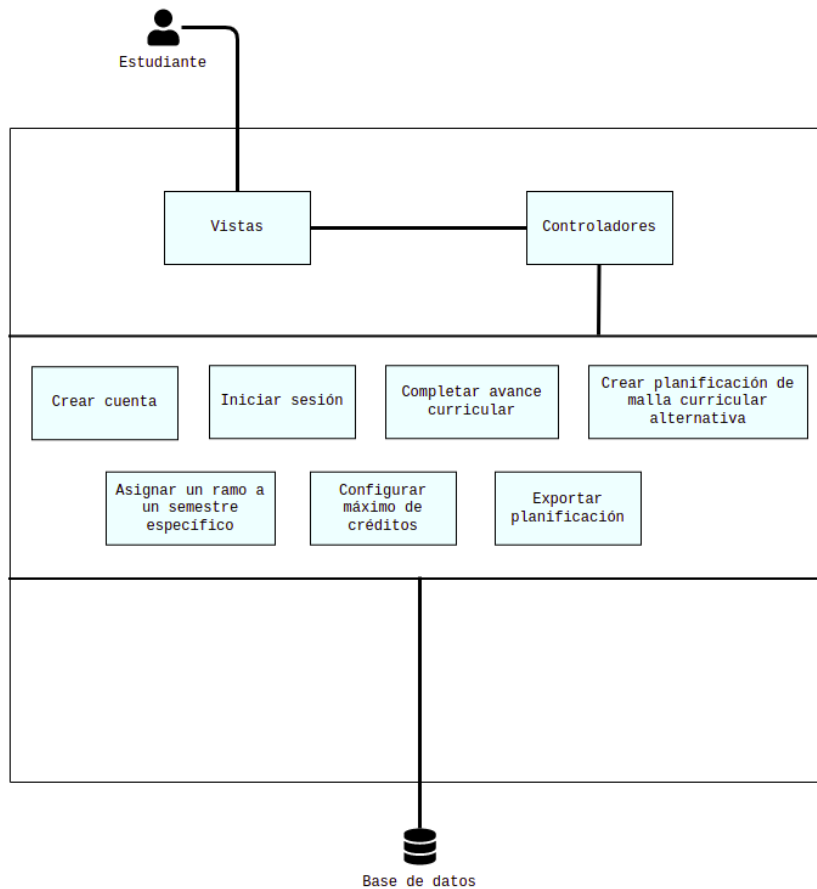


Figura 3.2: Diagrama de arquitectura lógica de la aplicación a implementar.

Tal como se ilustra en la Figura 3.2, la arquitectura lógica del sistema propuesto sigue un patrón de arquitectura de tres capas, separando al software en tres partes:

- **Capa de presentación:** contiene la interfaz de usuario y se implementa a través del framework AngularJS. En particular, está conformada por archivos vista y archivos controlador.

Los archivos vista se utilizan para establecer características visuales de ciertos elementos de la interfaz, mientras que el comportamiento esperado de dichos elementos y la interacción entre los mismos se define en los archivos controladores correspondientes.

- **Capa de lógica de negocio:** contiene los programas que implementan las reglas del negocio de la aplicación. En este caso se compone de módulos desarrollados mediante AngularJS.

Estos módulos constituyen un puente de comunicación entre la capa de presentación y la capa de datos, y se encargan de satisfacer los casos de uso establecidos en la Sección 3.1.

- **Capa de datos:** interactúa con la base de datos y se implementa usando Python y Django.

Su función principal consiste en obtener y almacenar información relevante para soportar los distintos casos de uso, especialmente: el avance curricular del estudiante y las planificaciones de malla curricular alternativa creadas por él.

3.3. Modelo de datos

En la Figura 3.3 se presenta el modelo de datos que se ha establecido con el propósito de manejar el flujo de información de la plataforma a implementar. Éste corresponde a un modelo de datos relacional.

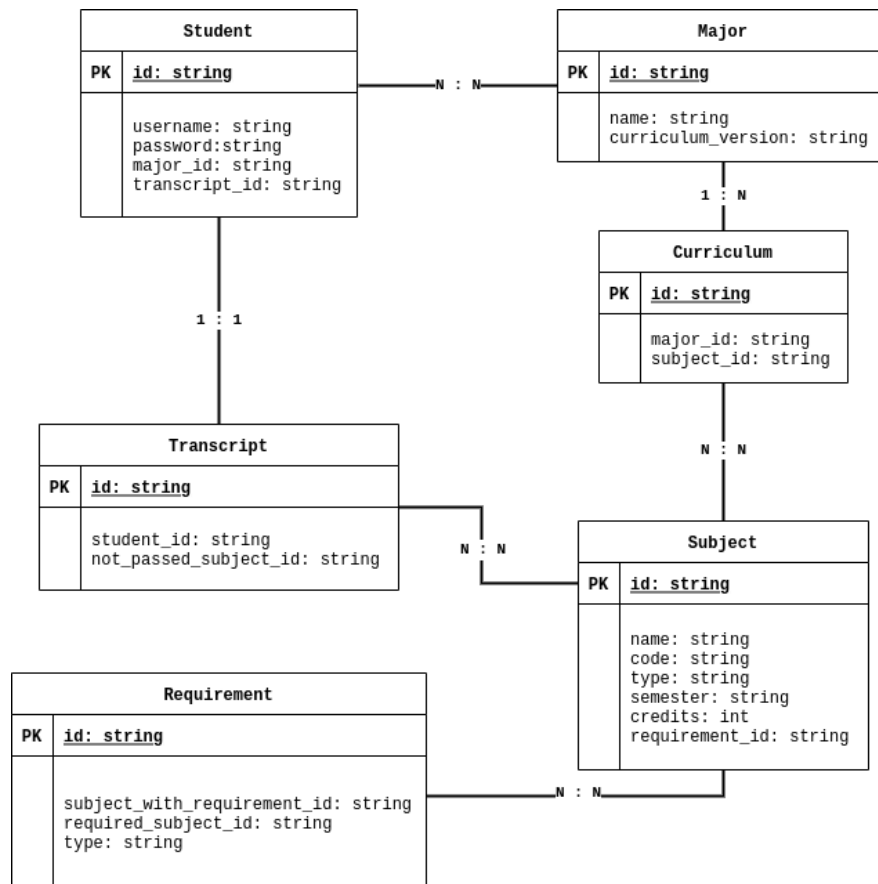


Figura 3.3: Diagrama entidad-relación de modelo de datos para la aplicación.

A partir del conjunto de datos a almacenar se han identificado las siguientes entidades: estudiante, carrera, malla curricular, avance curricular, asignatura y requisito.

La información del estudiante se almacena en la tabla **Student**. Dicha tabla se relaciona con las entidades **Major** y **Transcript**, las cuales guardan datos de carrera y avance curricular, respectivamente.

La tabla **Major** se vincula con la tabla **Curriculum**, siendo esta última representante de la entidad de malla curricular. **Curriculum** se relaciona con la tabla **Subject**, la cual contiene

información referente a asignaturas. La tabla **Transcript** vincula a un estudiante con una asignatura no aprobada, representando así el avance curricular de dicho estudiante.

La entidad **Subject** está relacionada con la entidad **Requirement**. Ésta simboliza las relaciones de requisito entre asignaturas, donde `subject_with_requirement_id` corresponde al identificador de la asignatura que posee el requisito y `required_subject_id` es el identificador del ramo que es requerido.

Capítulo 4

Diseño de Interacción

En el presente capítulo se detalla el proceso llevado a cabo para confeccionar las interfaces de usuario de la herramienta, elaboradas a partir de los requerimientos establecidos en el Capítulo 3.

Se ha dividido la explicación concerniente a los bocetos de interfaces de usuario en versión inicial y versión final, con el propósito de evidenciar el proceso de refinamiento de ideas llevado a cabo. Así, se pretende justificar las decisiones de diseño tomadas, mostrando los razonamientos que producen el resultado final.

Las interfaces correspondientes a la versión inicial se describen en la Sección 4.1, mientras que las de la versión final se describen en la Sección 4.2. La Sección 4.3 se refiere el proceso de validación inicial que se lleva a cabo sobre las interfaces mencionadas.

4.1. Diseño de interfaces de usuario: versión inicial

4.1.1. Interfaces de módulo avance curricular

Tal como se explica en la Sección 1.4, se decide implementar una funcionalidad que le permita al estudiante completar su avance curricular hasta el momento, para que así el sistema sea capaz de identificar las asignaturas que él o ella aún no ha aprobado.

Para ello, lo principal que se debe tener en cuenta es la existencia de distintas versiones de una malla curricular que se encuentren activas al mismo tiempo. Por ejemplo, para Ingeniería Civil en Computación se tienen las versiones v3 y v5.

La Figura 4.1 exhibe un boceto de interfaz de selección de versión de la malla curricular correspondiente a la carrera cursada por el estudiante. En él, se advierte la existencia de un recuadro que contiene instrucciones para que el usuario encuentre la versión de su malla (en caso que no la conozca). Dicho elemento se conoce como *tooltip*.

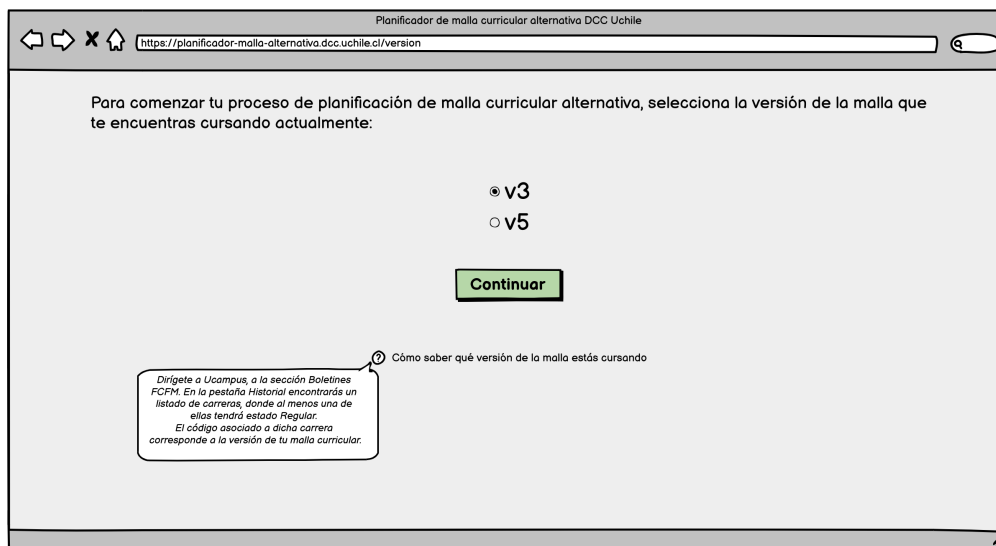


Figura 4.1: Boceto interfaz de selección de versión de la malla curricular correspondiente a la carrera cursada por el usuario (versión inicial). Elaboración propia.

Un *tooltip* es un componente común de la interfaz gráfica de usuario (GUI), constituido por un cuadro de texto que se hace visible al desplazar el *mouse* sobre un elemento o componente de la pantalla. Este cuadro de texto muestra información sobre el elemento, por ejemplo, una descripción de la función de un botón o lo que significa una abreviatura.

Luego de un análisis en busca de mejoras, se determina que el *tooltip* descrito debería transformarse en texto explícito. Así, se asegura que el estudiante tiene acceso a la información de manera inmediata y facilita su experiencia de usuario. Además, se identifica la necesidad de implementar una funcionalidad de selección de carrera cursada por el usuario, con la finalidad de privilegiar la extensibilidad.

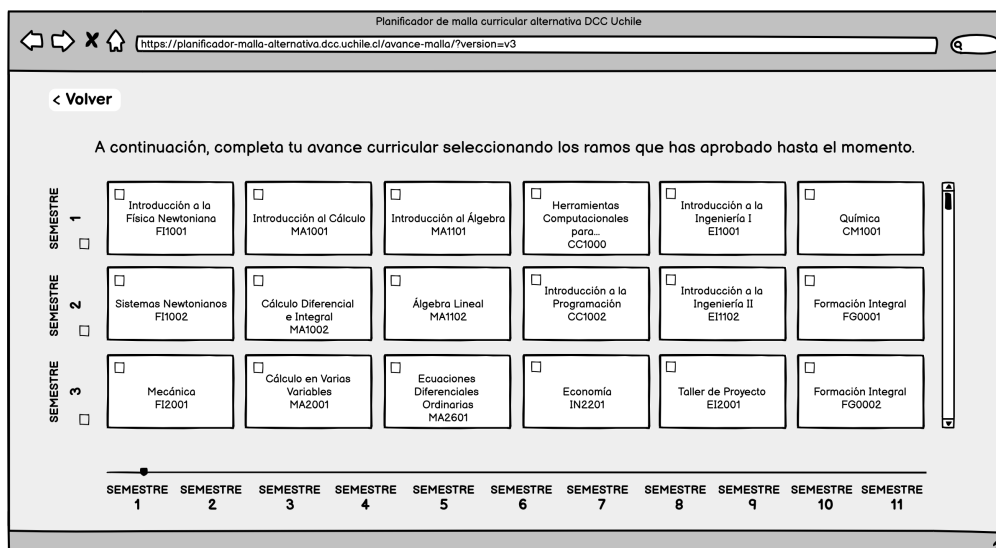


Figura 4.2: Boceto interfaz para completar avance curricular del usuario (versión inicial). Elaboración propia.

La Figura 4.2 muestra el boceto de interfaz para completar el avance curricular del estudiante. Al examinarla se advierte que resulta intuitivo organizar las asignaturas según el semestre al que pertenecen. También, representarlas mediante rectángulos tiene la intención de ser una ayuda visual. Ambas observaciones se consideran positivas en cuanto a usabilidad, por lo que se decide mantener dichas características en la versión final del boceto.

Por otra parte, se percibe que el propósito de la barra horizontal localizada en la parte inferior de la pantalla no queda claro, por lo que ésta se quita de la interfaz. Asimismo, se aprecia que la barra vertical para avanzar por la página no resulta especialmente amigable para el usuario. La manera en que se ha organizado la información resulta abrumadora, por lo tanto, se buscará mejorarla en la siguiente iteración del boceto.

Posteriormente se evidencia una alternativa de selección de ramos aprobados, en el boceto de la Figura 4.3. Ésta demuestra el funcionamiento del selector de semestre completo, mediante el cual se busca minimizar la cantidad de clics que se requieren para completar la tarea.

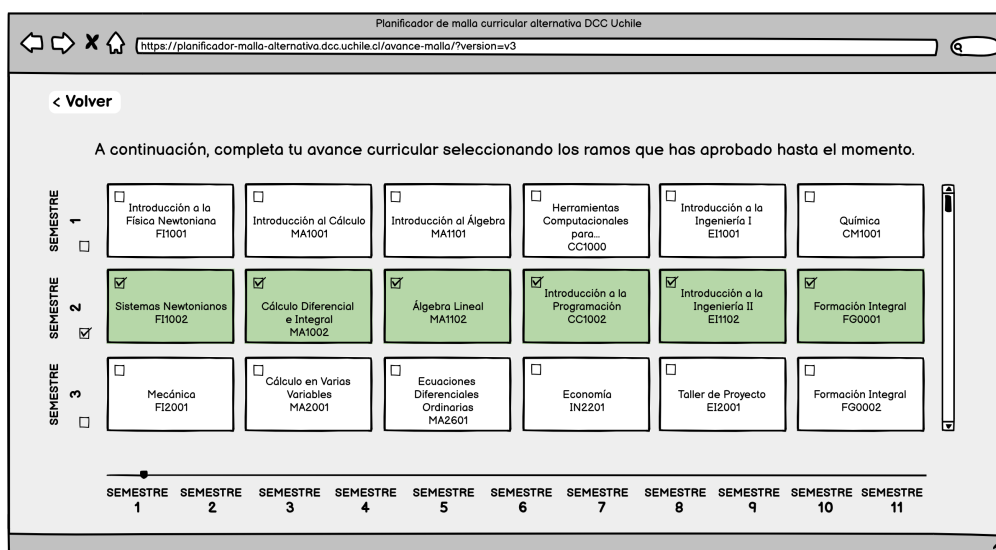


Figura 4.3: Boceto interfaz de avance curricular del usuario, segundo semestre seleccionado (versión inicial). Elaboración propia.

4.1.2. Interfaces de módulo planificador de malla curricular alternativa

Más adelante se expone el boceto de interfaz de usuario correspondiente al módulo planificador de malla curricular alternativa, al cual se accede inmediatamente después de completar el avance curricular.

La Figura 4.4 muestra cómo se vería esta interfaz, con una sección en la parte superior que exhibe los ramos que el estudiante aún no ha aprobado, seguida de una sección para asignar los ramos a semestres próximos (o al semestre en curso). Al igual que en las interfaces previas, se percibe que la organización de asignaturas por semestre resulta intuitiva, y también ayuda

al usuario a identificar cuáles son las más atrasadas.

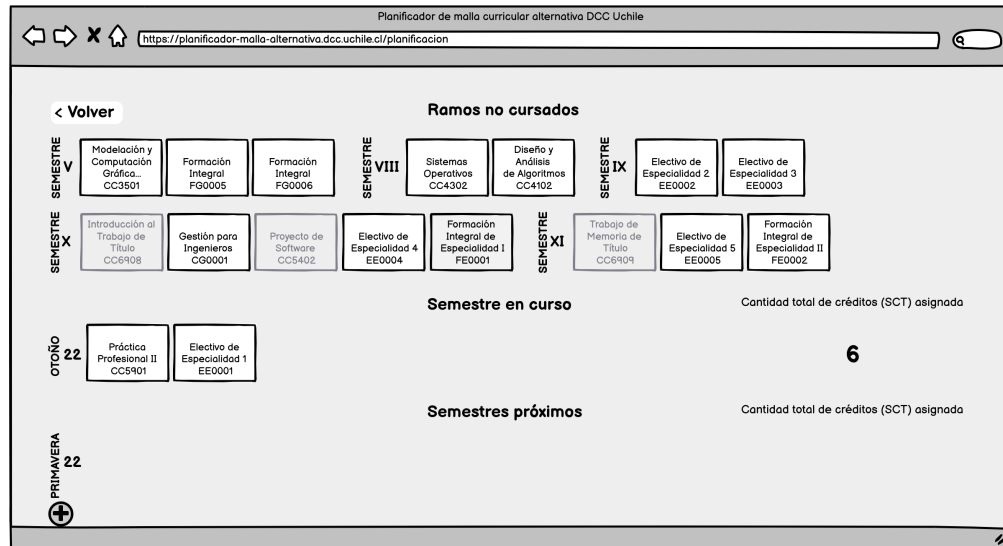


Figura 4.4: Boceto interfaz de planificador de malla curricular alternativa. Vista completa, incluyendo sección de ramos no cursados y sección de asignación de ramos por semestre (versión inicial). Elaboración propia.

La intención de distinguir entre semestre en curso y semestres próximos se justifica en el hecho de que es factible considerar el semestre en curso en la planificación. Particularmente, el alumno podría requerir simular la eliminación o reprobación de algunos de los ramos que se encuentra cursando en el momento presente.

Finalmente, se determina que la organización de la información en la página, siguiendo una orientación vertical, produce una vista compacta y abrumadora, por lo que este aspecto se revisará en la siguiente iteración.

4.2. Diseño de interfaces de usuario: versión final

A continuación se presentan los bocetos de interfaces de usuario finales, resultantes de un ciclo de iteración sobre los bocetos mostrados en la Sección 4.1.

4.2.1. Interfaz de módulo inicio de sesión

Se pretende que el módulo de inicio de sesión presente las características estándar esperadas en un módulo diseñado con este fin. Como se aprecia en la Figura 4.5, éste contiene un formulario para ingresar credenciales de autenticación, un botón para iniciar sesión, un enlace para manejar el olvido de la contraseña y un botón para crear una cuenta.

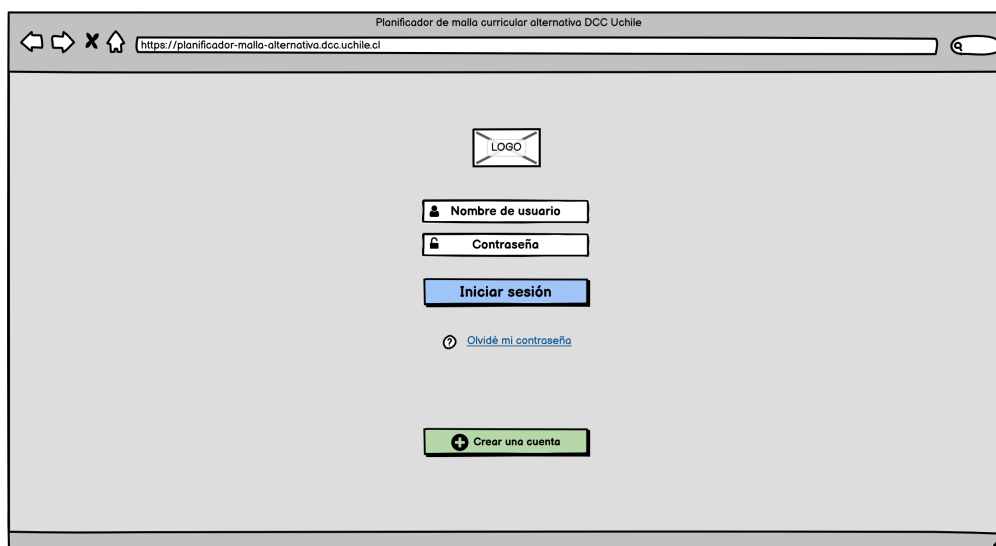


Figura 4.5: Boceto interfaz de inicio de sesión (versión final). Elaboración propia.

4.2.2. Interfaces de módulo avance curricular

La Figura 4.6 exhibe la versión final del boceto de interfaz de selección de versión de la malla curricular.

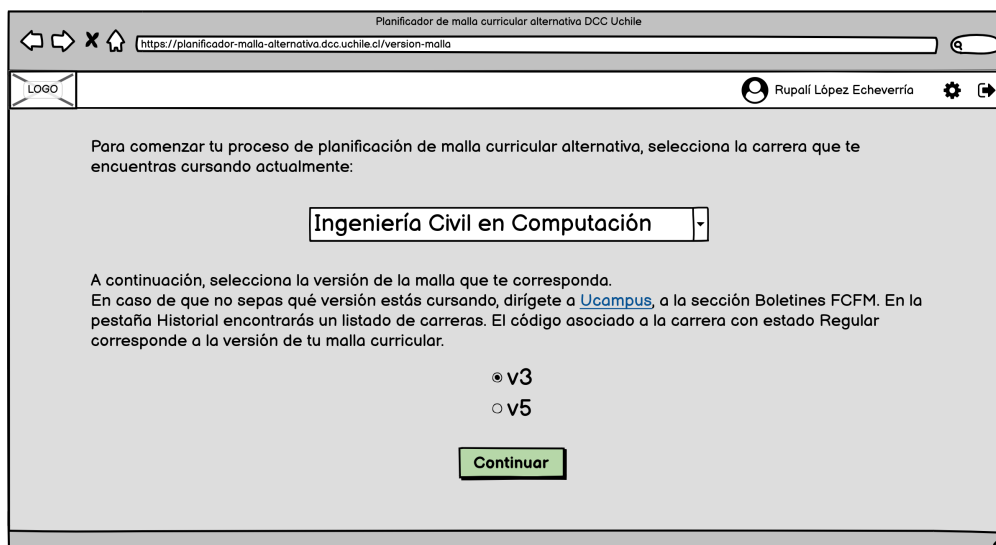


Figura 4.6: Boceto interfaz de selección de versión de la malla curricular correspondiente a la carrera cursada por el usuario (versión final). Elaboración propia.

Se agrega un selector de carrera (por el momento solo contiene la carrera de Ingeniería Civil en Computación) con el propósito de privilegiar la extensibilidad de la plataforma. Además, se añade un párrafo indicando al estudiante cómo puede obtener la versión de su malla curricular en el portal Ucampus. Así se asegura que el alumno pueda continuar usando la herramienta sin mayores inconvenientes.

En las Figuras 4.7, 4.8, 4.9 y 4.10 se ilustra el flujo de navegación que seguiría un usuario al completar su avance curricular para la versión v3 de la malla de la carrera de Ingeniería Civil en Computación.

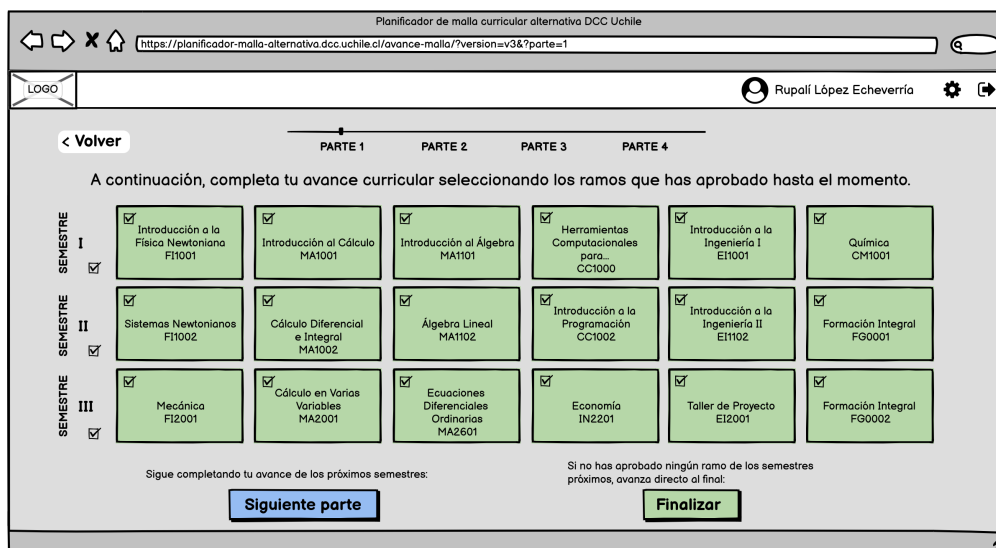


Figura 4.7: Boceto interfaz para completar avance curricular del usuario, primera parte (versión final). Elaboración propia.

Es pertinente advertir que se ha modificado la interfaz dividiéndola en cuatro secciones, con la finalidad de hacer el proceso menos abrumador para el usuario. Se ha agregado una herramienta de *breadcrumbs*, con el propósito de acrecentar la usabilidad. Ésta le indica al usuario con anticipación cuántas partes hay, y le permite saber en cuál de ellas se encuentra en todo momento.

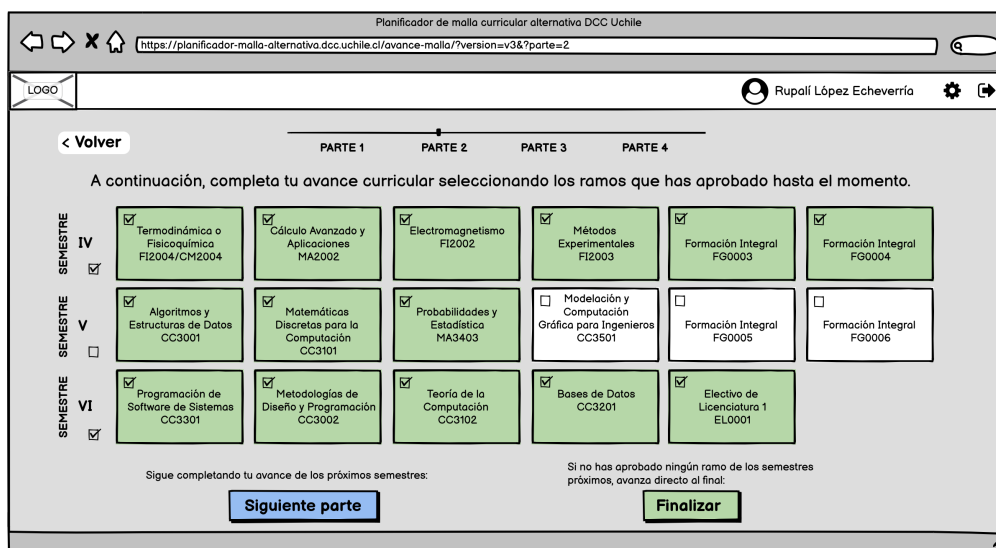


Figura 4.8: Boceto interfaz para completar avance curricular del usuario, segunda parte (versión final). Elaboración propia.

Los botones “Siguiete parte” y “Finalizar” pretenden ayudar a minimizar la cantidad de

clics requeridos para completar la acción, en caso de que el estudiante no haya aprobado ningún ramo en las partes siguientes.

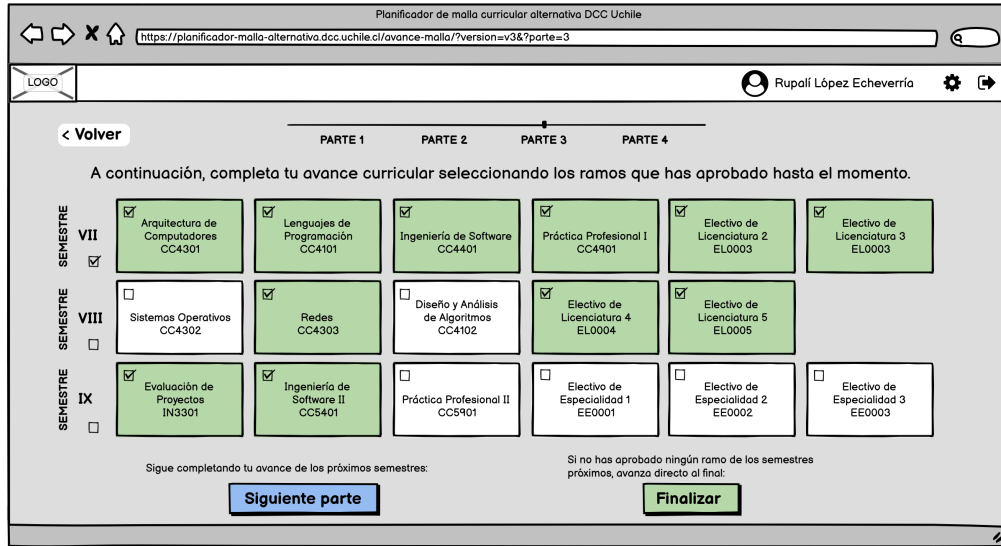


Figura 4.9: Boceto interfaz para completar avance curricular del usuario, tercera parte (versión final). Elaboración propia.

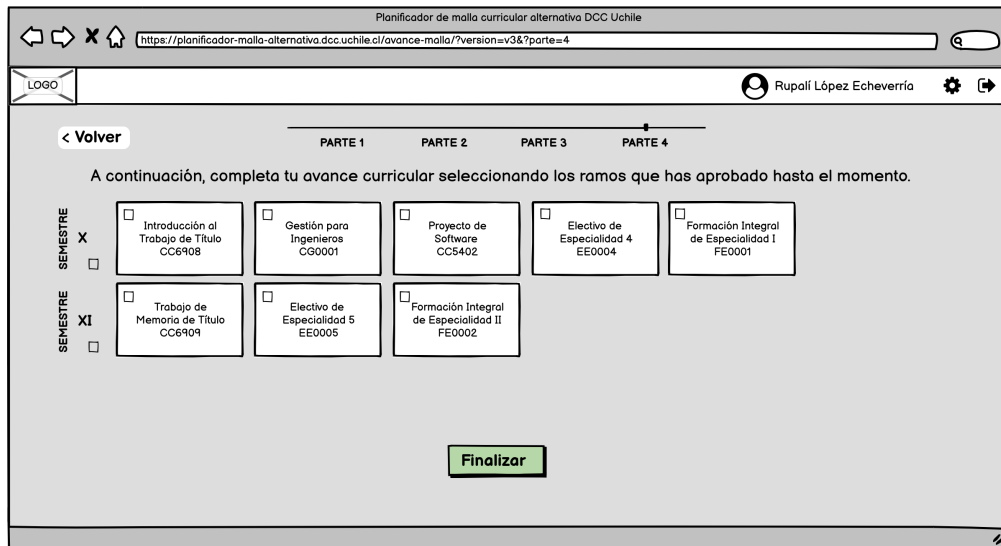


Figura 4.10: Boceto interfaz para completar avance curricular del usuario, cuarta parte (versión final). Elaboración propia.

4.2.3. Interfaces de módulo planificador de malla curricular alternativa

Más adelante se exponen las versiones finales de los bocetos de interfaces de usuario correspondientes al módulo planificador de malla curricular alternativa, al cual se accede inmediatamente después de completar el avance curricular.

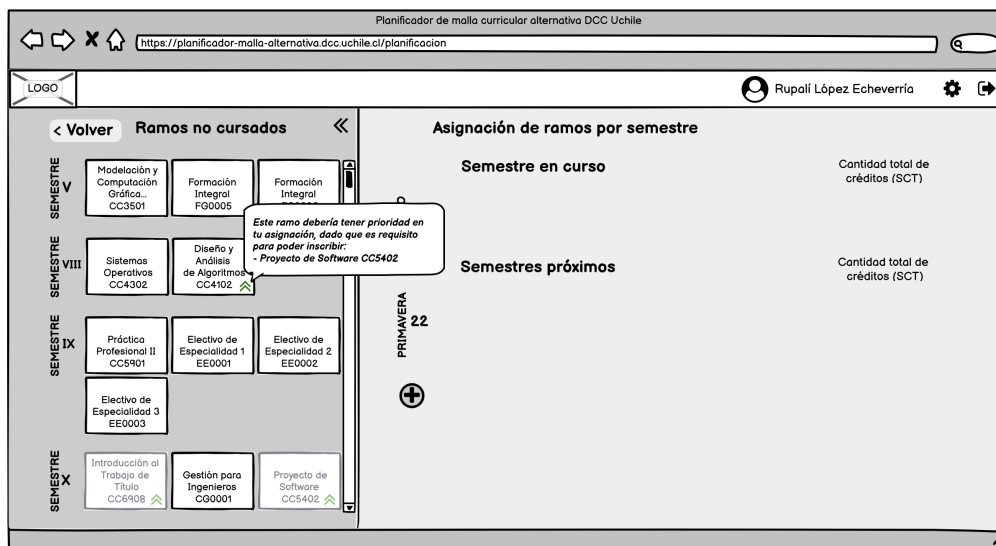


Figura 4.11: Boceto interfaz de planificador de malla curricular alternativa. Vista completa, incluyendo sección de ramos no cursados y sección de asignación de ramos por semestre. Incluye tooltip explicativo de ramo prioritario (versión final). Elaboración propia.

Las Figuras 4.11 y 4.12 muestran *tooltips* explicativos de la simbología implementada, lo que tiene la intención de incrementar la usabilidad de la interfaz. En particular, la Figura 4.11 ilustra el símbolo de asignatura prioritaria. Se determina la necesidad de identificar ramos bloqueantes, es decir, ramos requisito de uno o más ramos futuros, para indicar al estudiante que es recomendable otorgarles prioridad en la asignación. Se emplea una flecha hacia arriba para transmitir prioridad y se utiliza el color verde para intentar asociar un sentimiento positivo al símbolo.

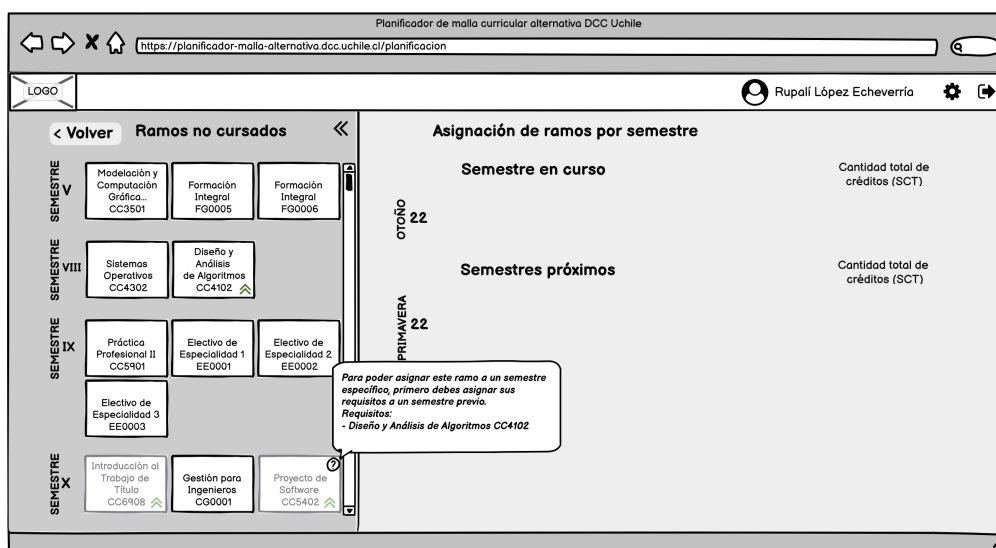


Figura 4.12: Boceto interfaz de planificador de malla curricular alternativa. Vista completa, incluyendo sección de ramos no cursados y sección de asignación de ramos por semestre. Incluye tooltip explicativo de ramo deshabilitado por requisito (versión final). Elaboración propia.

La Figura 4.12 muestra el *tooltip* explicativo de un ramo deshabilitado por requisito. Las dos funcionalidades descritas tienen la finalidad de guiar al usuario en el proceso, para intentar que tome las mejores decisiones posibles en cada asignación de ramos a semestres.

La Figura 4.13 muestra el resultado de asignar ramos al semestre en curso y al semestre inmediatamente próximo. Se observa un botón con el símbolo “+” abajo del semestre próximo, el cual sirve para agregar más semestres. Este comportamiento tiene la intención de dejar abierta la posibilidad para que el usuario planifique a mediano o largo plazo.

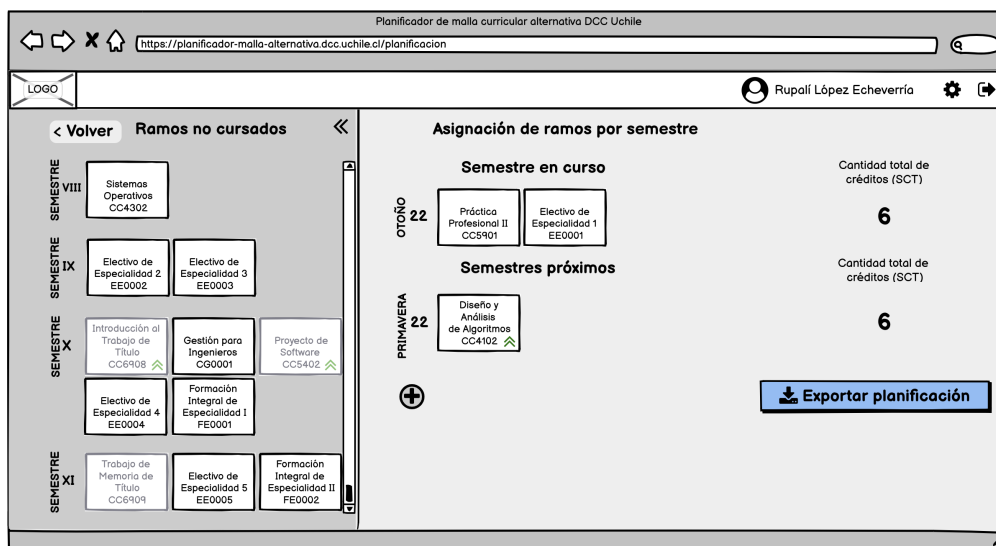


Figura 4.13: Boceto interfaz de planificador de malla curricular alternativa. Asignación de ramos a semestre en curso y semestre próximo (versión final). Elaboración propia.

Al costado derecho de cada semestre se exhibe la suma total de créditos por semestre. Esto tiene la finalidad de ayudar a orientar la elección del estudiante y apoyarlo en la búsqueda de un resultado balanceado en términos de carga académica.

Luego del análisis expuesto en la Sección 4.1, se modifica la disposición de los elementos en la interfaz, otorgando un tercio de la pantalla a la sección de ramos no cursados, y los dos tercios restantes a la sección de asignación de ramos por semestre. Esta alteración busca aprovechar de mejor manera el espacio disponible, organizar la información de una forma intuitiva para el usuario y evitar que la página se perciba saturada de información.

Las flechas laterales localizadas al costado del título “Ramos no cursados” corresponden a un botón que colapsa la sección, con la intención de otorgar mejor visibilidad al resto de la página.

Los ramos se asignan a los semestres mediante *drag and drop*, lo que tiene el propósito de ser una ayuda visual para el estudiante y permitirle realizar cambios de forma dinámica e interactiva.

El botón “Exportar planificación” posibilita la descarga de un documento resumen en formato *csv*. Este botón aparece cuando se ha completado al menos un semestre próximo, permitiendo así que el usuario planifique por cuantos semestres desee.

La Figura 4.14 exhibe el resultado de hacer aparecer un nuevo semestre apretando el botón “+”, y asignando un conjunto de ramos no cursados a éste.

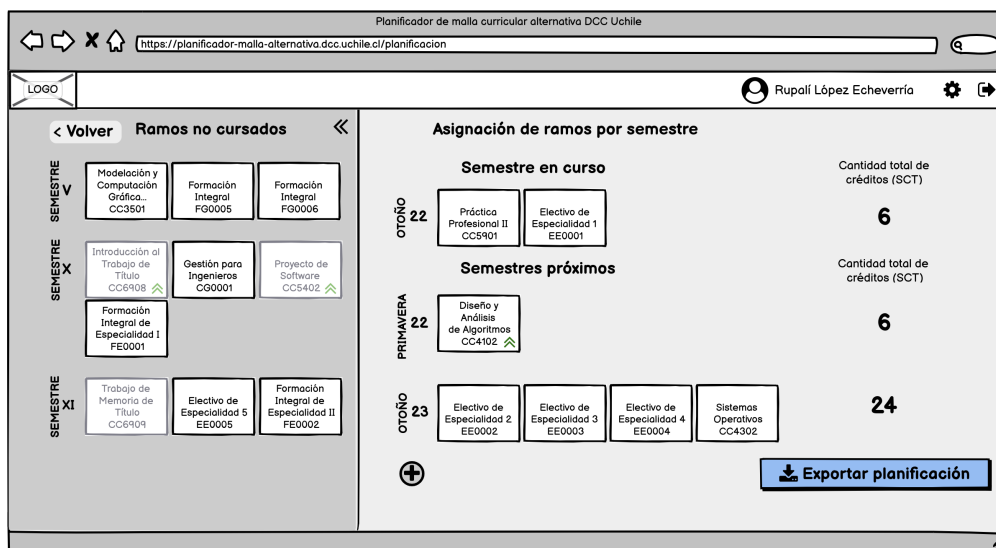


Figura 4.14: Boceto interfaz de planificador de malla curricular alternativa. Asignación de ramos a semestre siguiente (versión final). Elaboración propia.

La Figura 4.15 muestra la alerta que se dispara al intentar arrastrar el ramo “Proyecto de Software” al semestre en curso. Esto se debe a que en el momento de la asignación aún no se ha aprobado “Diseño y Análisis de Algoritmos”, asignatura requisito de “Proyecto de Software”. Esta funcionalidad pretende orientar al usuario en sus elecciones, evitando que realice una planificación incorrecta con respecto a lo que indica su malla curricular.

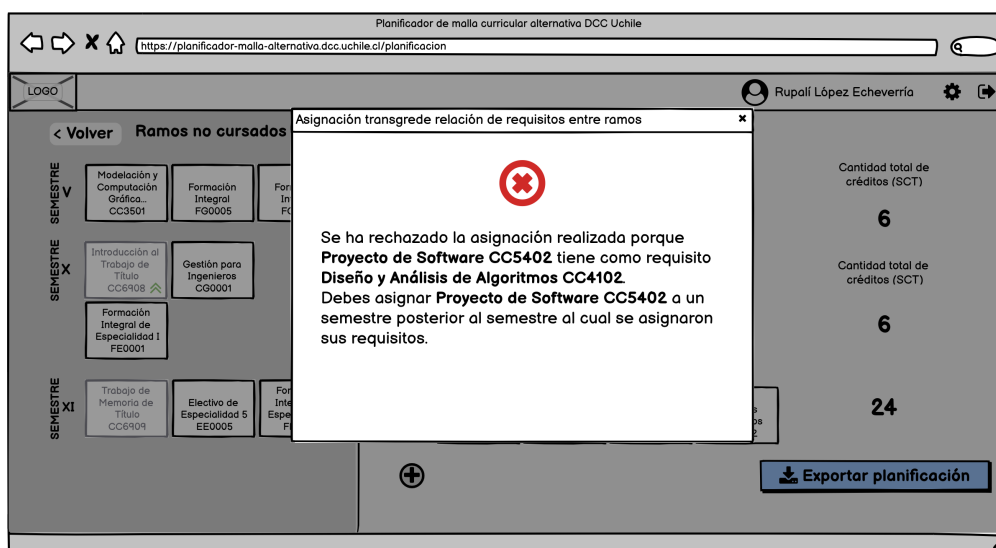


Figura 4.15: Boceto interfaz de planificador de malla curricular alternativa. Alerta de rechazo de asignación por incumplir relación de requisitos entre asignaturas (versión final). Elaboración propia.

La Figura 4.16 ilustra cómo se habilita la selección de un ramo no cursado con requisito, luego de la asignación del ramo requerido. Además, aparece una barra vertical para avanzar en la sección de asignación de ramos por semestre, dado que no se alcanza a mostrar toda la información en la pantalla.

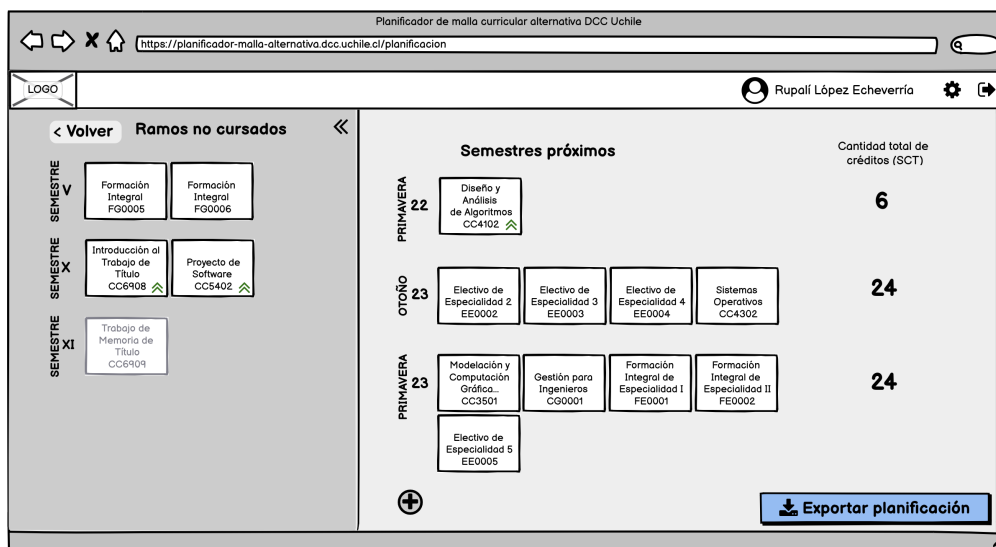


Figura 4.16: Boceto interfaz de planificador de malla curricular alternativa. Asignación de ramos a nuevo semestre, avanzando hacia abajo en la sección de asignación de ramos por semestre (versión final). Elaboración propia.

Cabe destacar que el proceso descrito es iterativo, es decir, se puede cambiar las asignaturas de semestre cuantas veces se quiera. Incluso se puede volver a colocarlas en la sección de ramos no cursados. El proceso solo termina una vez que se hace clic en “Exportar planificación”.

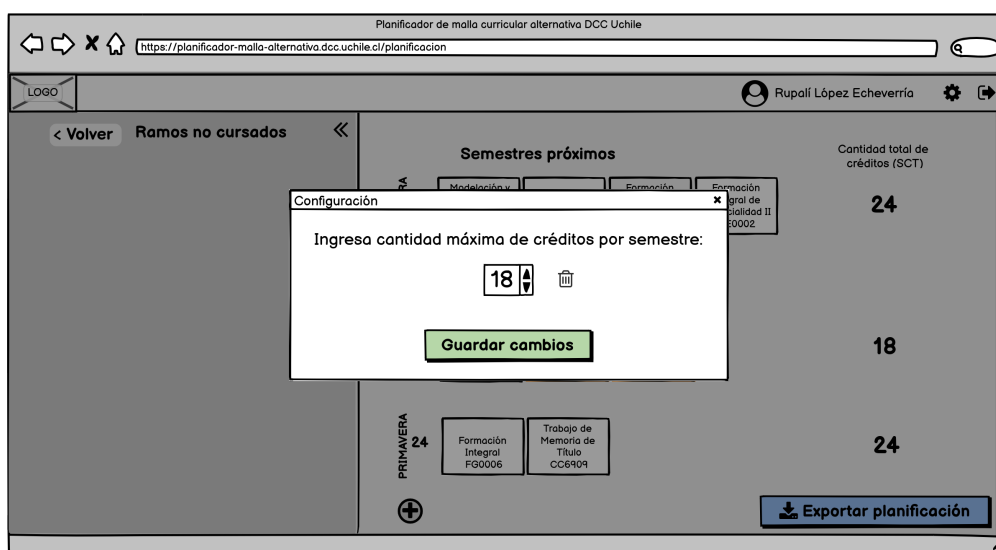


Figura 4.17: Boceto ventana emergente de configuración de cantidad máxima de créditos por semestre (versión final). Elaboración propia.

La Figura 4.17 expone el boceto de la ventana emergente de configuración de máximo de créditos por semestre. La restricción no existe por defecto. En el modal se permite actualizarla o eliminarla, según sea conveniente.

La Figura 4.18 muestra un boceto de la interfaz resultante luego de actualizar la restricción de cantidad máxima de créditos por semestre a 18, en la cual se resaltan en rojo los semestres que superan el número. Esto intenta ser una ayuda visual para el estudiante, para que pueda lograr un resultado balanceado en cuanto a carga académica.

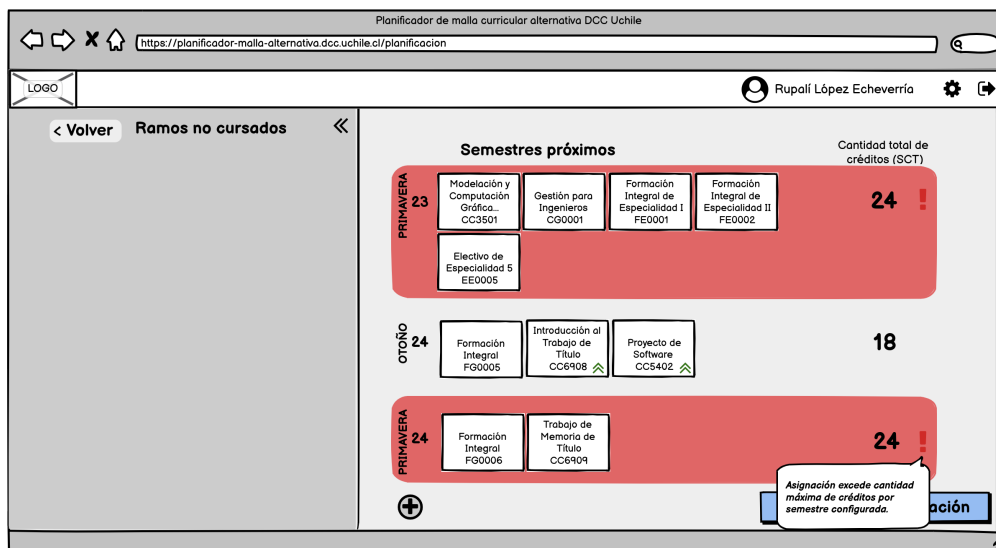


Figura 4.18: Boceto interfaz de planificador de malla curricular alternativa. Alerta de incumplimiento de cantidad máxima de créditos por semestre (versión final). Elaboración propia.

4.3. Validación inicial de interfaces diseñadas

Previo a comenzar la implementación de la aplicación, se lleva a cabo un breve proceso de validación inicial, con la finalidad de determinar si las nociones propias – plasmadas en los bocetos referidos en la Sección 4.2 – se corresponden con las expectativas de los usuarios y sus modelos mentales.

En esta instancia se prefiere realizar una evaluación rápida, considerando el tiempo limitado del que se dispone para completar el proyecto. Se opta por efectuar una segunda etapa de validación, más rigurosa y extensiva, al acabar la fase de implementación.

En el proceso de evaluación inicial participan 4 estudiantes del Departamento de Ciencias de la Computación: un memorista, uno que cursa el octavo semestre de la malla, y dos que cursan el décimo semestre.

La validación consiste en mostrarles y explicarles los prototipos diseñados, detallando el funcionamiento y las interacciones esperadas. Al mismo tiempo, se insta a los participantes a manifestar sus apreciaciones con respecto a lo observado. Especialmente, a señalar elementos que no comprendan, y a expresar sus ideas sobre potenciales mejoras. Todos los comentarios

recibidos son registrados de forma escrita.

La retroalimentación proporcionada por los estudiantes entrevistados es analizada y organizada según su prioridad. A partir de ella, se establecen los siguientes accionables, referentes al módulo planificador:

- En la sección “Ramos no cursados”, indicar la cantidad de créditos correspondiente a cada asignatura.
- En la sección de asignación de ramos a semestres, agregar bordes que permitan distinguir claramente los distintos semestres.
- Manejar el caso de un semestre vacío agregando un mensaje que indique cómo asignarle materias.
- Implementar el botón “Exportar planificación” de tal forma que sea visible en todo momento, pero que se encuentre deshabilitado cuando su uso no sea pertinente.

Por otra parte, se identifican ciertos aspectos que requieren ser analizados más profundamente, antes de poder alcanzar una resolución, tales como:

- Escoger un nuevo símbolo, con un significado más evidente, para indicar la prioridad que debe tener una asignatura en la planificación.
- Repensar el diseño de la alerta de transgresión de requisitos, considerando alternativas a la ventana emergente.
- Evaluar el modo de lidiar con los ramos cursados del semestre actual en la interfaz de planificación.
- Con respecto a la interfaz de avance curricular, examinar la posibilidad de dividir el avance en etapas referentes a plan común, licenciatura y especialidad. Además, repensar el diseño de los botones, tomando en cuenta el modelo mental del usuario.

Capítulo 5

Implementación

El presente capítulo abarca aspectos fundamentales que se relacionan al desarrollo mismo de la herramienta, tales como: especificaciones técnicas (Sección 5.1), automatización de la ejecución de la aplicación (Sección 5.2), estructura y organización del código (Sección 5.3), implementación de la interfaz planificador de malla curricular alternativa (Sección 5.4) y una serie de estrategias empleadas para preservar las buenas prácticas de programación (Sección 5.5).

5.1. Especificaciones técnicas

La aplicación se construye usando el framework AngularJS (versión 1.8.3). Para automatizar su ejecución, se emplea el ejecutor de tareas Gulp (versión 4.0.2).

Las dependencias del software son manejadas mediante Node (versión 18.0.0) y Bower (versión 1.8.14). Este último, en particular, se utiliza exclusivamente para paquetes de AngularJS, entre ellos, `angular-bootstrap` y `angular-ui-tree`.

5.2. Automatización ejecución de la aplicación

La integración, entrega e implementación continuas (CI/CD) son prácticas que permiten distribuir aplicaciones a clientes con frecuencia, mediante el uso de automatización en las etapas del desarrollo. Una ejecución automatizada genera fases de desarrollo más eficientes, y por sobre todo, produce código de mayor calidad.

A pesar de constituir una inversión de tiempo considerable, se opta por implementar un proceso de lanzamiento automatizado de la plataforma, puesto que contar con este atributo es notoriamente valioso para asegurar la mantenibilidad y extensibilidad del software.

Gulp es una herramienta que permite enriquecer el flujo de trabajo a través de la ejecución de tareas. Dichas tareas consisten de funciones asíncronas de JavaScript que llevan a cabo

operaciones involucradas en el desarrollo web, las cuales suelen ser repetitivas y consumir preciado tiempo y esfuerzo (tales como tests unitarios y *linting*).

5.2.1. Tarea default de Gulp

La tarea `default` se ejecuta por defecto al usar el comando `gulp` en la terminal de Linux. Su propósito es producir una carpeta de archivos temporales llamada `.tmp`, donde se reúne el código del proyecto.

Dicha carpeta se adhiere a la estructura expuesta en la Figura 5.1, en la cual se observan los archivos `index.css` y `index.html`. El primero está formado por el contenido de cada hoja de estilo de la aplicación (archivo con extensión `.scss`), mientras que el segundo incluye referencias a dependencias de `bower`, archivos con extensión `.js` y archivos de vistas (con extensión `.html`).

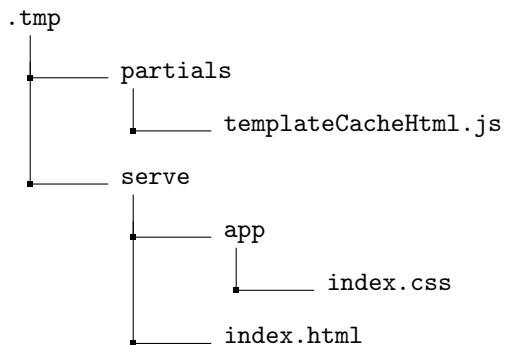


Figura 5.1: Estructura del directorio temporal resultante de la ejecución de la tarea `default` de Gulp.

El extracto del programa referente a la definición de la tarea `default` se aprecia más adelante, en el Código 5.1.

```
1 export async function clean() {
2   await Promise.resolve(deleteSync([
3     path.join(paths.dist, '/'), path.join(paths.tmp, '/')
4   ]));
5 };
6
7 export async function build() {
8   gulp.series('html')();
9   await Promise.resolve();
10 };
11
12 gulp.task('clean', clean);
13 gulp.task('build', build);
14
15 gulp.task('default', async function () {
16   gulp.series('clean', 'build')();
17   await Promise.resolve();
18 });
```

Código 5.1: Extracto del programa referente a la definición de la tarea `default` de Gulp

Se inicia invocando a la tarea `clean`, encargada de eliminar los archivos temporales creados durante la ejecución previa.

A continuación, la tarea `build` es llamada, y ésta a su vez invoca a la tarea `html`. Las etapas del flujo de ejecución de la tarea `html` son:

1. Se comienza ejecutando la tarea `inject`, la cual se comporta como sigue:
 - Empieza leyendo los archivos de tipo `script` (es decir, archivos con extensión `.js`), para luego aplicarles un análisis de código estático usando la herramienta ESLint. En caso de detectarse errores, éstos se imprimen en la terminal (aunque el programa no se detiene).
 - Posteriormente, se leen las hojas de estilo y se inyecta su contenido en `index.scss`. Este último archivo es copiado a la carpeta `.tmp`.
 - Después, se inyectan referencias a hojas de estilo y archivos `script` en `index.html`, siendo este último archivo copiado a la carpeta temporal.
2. Se procede a leer los archivos de tipo plantilla (es decir, archivos con extensión `.html`). En AngularJS, la primera vez que se usa una plantilla, ésta se debe cargar en un objeto de tipo caché, llamado `$templateCache`, para asegurar que se pueda recuperar rápidamente.
Entonces, por medio del uso del paquete `gulp-angular-templatecache`, se genera una serie de instrucciones que permiten que las plantillas sean concatenadas y registradas en dicho objeto. Estas instrucciones se almacenan en `templateCacheHtml.js` dentro de la carpeta `.tmp`.
3. Finalmente, el contenido de las plantillas mencionadas en la etapa anterior se adiciona a `index.html`, dentro de la carpeta temporal.

5.2.2. Tarea `serve` de Gulp

La tarea `serve` se ejecuta al utilizar el comando `gulp serve` en la terminal de Linux. Su misión principal consiste en lanzar la aplicación en un navegador. Además, se ocupa de vigilar cambios en el código, con el objetivo de facilitar una recarga automática de la página cuando corresponda.

Contar con la funcionalidad de recarga automática de la página es extremadamente significativo, dado que al tratarse de la implementación de un prototipo interactivo, esta característica permite reducir tiempos de desarrollo y lograr que el flujo de trabajo sea más eficiente.

El extracto del programa referente a la definición de la tarea `serve` se exhibe más adelante, en el Código 5.2. En él, se advierte que se realizan dos llamadas secuenciales: a la tarea `watch`, y al método `init` del paquete `browser-sync`.

Las fases de la ejecución de la tarea `watch` son:

1. Se invoca la tarea `inject`, cuyo comportamiento se describió en la Sección 5.2.1.
2. Se utiliza el comando `watch` de `gulp` para observar modificaciones en archivos, y ejecutar el trozo de código asociado cuando un cambio ocurra. En concreto, se busca detectar cambios en archivos con extensión `.html`, `.scss` y `.js`. Cada uno de los eventos mencionados finalmente resulta en la recarga de la página en el navegador.

Por otra parte, el método `init` de `browser-sync` se ocupa de inicializar un servicio que levanta un servidor, haciendo uso de la plantilla y el archivo de estilo de la carpeta temporal. Dicho servidor se lanza sobre el navegador entregado como parámetro a la función (en este caso, el navegador por defecto de la máquina que se esté utilizando).

```
1 export async function serve() {  
2   gulp.series('watch', function () {  
3     browserSyncInit([path.join(paths.tmp, '/serve'), paths.src]);  
4   })();  
5   await Promise.resolve();  
6 };  
7  
8 gulp.task('serve', serve);
```

Código 5.2: Extracto del programa referente a la definición de la tarea `serve` de Gulp

Entonces, se logra una implementación completa de un proceso de ejecución automatizada de la aplicación.

5.3. Estructura y organización de código

Se decide organizar el código del proyecto de manera que cada sección de la interfaz corresponda a una de las carpetas del directorio base, con la intención de acrecentar la legibilidad, y por lo tanto, la mantenibilidad y extensibilidad de la herramienta.

Como se manifiesta en la Figura 5.2, cada carpeta que representa una sección contiene al menos un archivo de tipo controlador, un archivo con extensión `.html` y una hoja de estilo.

El archivo de tipo controlador tiene como función manejar comportamientos, respondiendo a acciones o eventos que se produzcan dentro de la interfaz. En el archivo tipo plantilla se describe la estructura de los elementos mostrados en pantalla. La hoja de estilo abarca el listado de instrucciones que determinan la apariencia visual de la sección en cuestión.

Tanto el controlador como la plantilla poseen acceso a la misma información, gracias al *data binding* que provee AngularJS mediante la variable `vm` (*view model*). *Data binding* en AngularJS es la sincronización entre el modelo y la vista. Cuando la información cambia en el modelo, la vista refleja el cambio, y cuando la información cambia en la vista, el modelo también se actualiza.

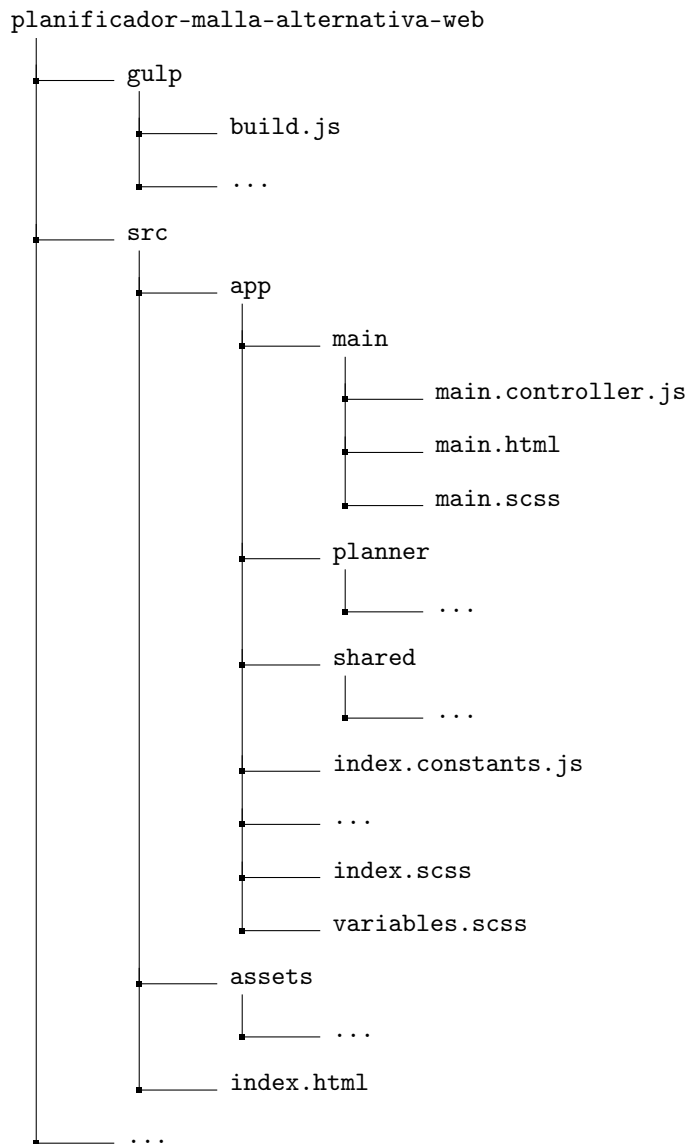


Figura 5.2: Diagrama de componentes principales del directorio base del proyecto.

Según lo observado en la Figura 5.2, se advierte que la aplicación se conforma de tres carpetas principales: la carpeta **main**, la carpeta **planner** y la carpeta **shared**.

La carpeta **main** representa a la base de la interfaz, sobre la cual se posicionan las demás secciones. Contiene elementos que serán visibles y accesibles en todo momento.

El contenido de la carpeta **planner** simboliza a la parte principal de la aplicación, esto es, el planificador de malla curricular alternativa descrito en la Sección 4.2. Éste proporciona una visualización de ramos pendientes del estudiante y permite asignarlos de manera interactiva a los semestres próximos.

La carpeta **shared** no se refiere a ninguna sección en particular, si no que incluye una serie de componentes, modelos y servicios que son de utilidad para el resto de la plataforma.

5.4. Interfaz planificador de malla curricular alternativa

Considerando los bocetos expuestos en la Sección 4.2, se podría suponer que la estrategia más coherente es implementar comenzando por el principio, ya sea con el módulo de inicio de sesión o el módulo de avance curricular. Sin embargo, teniendo en cuenta la limitante de tiempo, se estima que lo más provechoso es enfocarse a construir el módulo planificador de malla curricular alternativa, debido a que se trata de la funcionalidad *core* de la herramienta.

Las características de este módulo son determinadas fundamentalmente por el contenido de los archivos `planner.html` (de tipo plantilla) y `planner.controller.js` (de tipo controlador), el cual se detallará posteriormente.

5.4.1. Archivo de tipo plantilla `planner.html`

Tal como se exhibe en la Figura 5.3, la interfaz se divide en dos grandes subsecciones: “Ramos pendientes” y “Asignación de ramos por semestre”. La sección de ramos pendientes está constituida por el conjunto de asignaturas pendientes del estudiante, es decir, asignaturas pertenecientes a la malla curricular de su carrera que él o ella aún no ha aprobado. Las materias se representan mediante rectángulos, agrupados según el semestre de la malla al que corresponden.

Planificador malla curricular alternativa

Ramos pendientes
Aquí se muestran las asignaturas de tu malla curricular que aún no has aprobado.

SEMESTRE	Asignatura	Código	Créditos
VII	Diseño y Análisis de Algoritmos	CC4102	6 créditos
	Práctica Profesional I	CC4901	7 créditos
	Electivo de Licenciatura 2	EL0002	6 créditos
VIII	Lenguajes de Programación	CC4101	6 créditos
	Redes	CC4303	6 créditos
	Formulación, Evaluación y Gestión de Proyectos	CC4402	6 créditos
	Taller de Práctica Profesional II	CC5900	0 créditos
	Electivo de Licenciatura 3	EL0003	6 créditos
	Electivo de Licenciatura 4	EL0004	6 créditos
IX	Práctica Profesional II	CC5901	7 créditos
	Núcleo Gestión para Ingenieros	CG0001	6 créditos
	Electivo de Especialidad 1	EE0001	6 créditos
	Electivo de Especialidad 2	EE0002	6 créditos
	Electivo de Especialidad 3	EE0003	6 créditos
Formación Integral de Especialidad I	FE0001	3 créditos	
Formación Integral de Especialidad II	FE0002	3 créditos	

Asignación de ramos por semestre
Para obtener tu malla curricular alternativa, asigna tus ramos pendientes a semestres venideros.

Total créditos (SCT): 0

PRIMAVERA 2022

Arrastra una asignatura desde la sección de ramos pendientes y suéltala aquí para asignarla a este semestre

+

Figura 5.3: Captura de pantalla de la interfaz planificador de malla curricular alternativa al acceder por primera vez a la aplicación (usando un conjunto de ramos pendientes de ejemplo).

La sección de asignación de ramos por semestre tiene como objetivo permitir visualizar diversas alternativas de planificación para períodos futuros. Se compone inicialmente de un único semestre y un botón con el símbolo “+”, el cual posibilita la aparición de elementos consecutivos.

El período presentado es el más próximo, considerando la fecha actual (en este caso, Primavera 2022). En un principio se encuentra vacío, lo que es señalado por medio de un contenedor rectangular de tipo *placeholder*, cuyo texto indica al usuario cómo poblar el semestre en cuestión.

Además, se dispone de un contador que muestra la cantidad total de créditos asociada a cada período. Al comienzo ésta es 0, y se va actualizando cada vez que se realiza una asignación de una materia a un semestre específico.

Los atributos visuales del módulo planificador previamente descritos se establecen en el archivo `planner.html`. En él se determina la estructura de los elementos a mostrar, y la disposición que éstos tendrán en pantalla.

A continuación, el Código 5.3 expone un extracto de dicho archivo, evidenciando ciertas características relevantes que se detallarán más adelante.

```

1 <div class="planner-content">
2   <div class="... pending-subjects-per-semester-content"> ...
3     <div ui-tree="..." data-nodrop-enabled="..." ... ng-repeat="(
4       semesterNumber, semester) in vm.pendingSubjectsPerSemester">
5       <ul ui-tree-nodes ... ng-model="semester.subjects"> ...
6         <div ng-if="semester.subjects.length === 0" class="...">
7           <li ui-tree-node data-nodrag class="...">
8             {{ vm.emptyPendingSubjectsPlaceholder }} ... </li> </div>
9         <div ng-if="semester.subjects.length > 0" class="...">
10          <li ng-repeat="pendingSubject in semester.subjects"
11            ui-tree-node>
12            <div ng-if="vm.getSubjectTooltipMessage(...)" ...>
13              <subject subject="pendingSubject"></subject>
14              <span class="...">
15                {{ vm.getSubjectTooltipMessage(...) }} </span>
16            </div>
17            <div ng-if="!vm.getSubjectTooltipMessage(...)" ...>
18              <subject subject="pendingSubject"></subject> </div>
19          </li> </div> </ul>
20        </div>
21      </div>
22      <div class="... upcoming-semesters-content"> ...
23        <div ui-tree="..." data-nodrop-enabled="..." ...
24          ng-repeat="(semesterNumber, semester) in vm.upcomingSemesters">
25          <ul ui-tree-nodes ... ng-model="semester.subjects"> ...
26            <div ng-if="semester.subjects.length === 0" class="...">
27              <li ui-tree-node data-nodrag class="...">
28                ... {{ vm.emptyUpcomingSemesterPlaceholder }} </li> </div>
29            <div ng-if="semester.subjects.length > 0" class="...">
30              <li ng-repeat="assignedSubject in semester.subjects"
31                ui-tree-node>
32                <div ng-if="vm.getSubjectTooltipMessage(...)" ...>
33                  <subject subject="assignedSubject"></subject>
34                  <span class="...">
35                    {{ vm.getSubjectTooltipMessage(...) }} </span>
36                </div>
37                <div ng-if="!vm.getSubjectTooltipMessage(...)" ...>
38                  <subject subject="assignedSubject"></subject> </div>
39              </li> </div>
40            <div class="..."> {{ semester.assignedCredits }} </div> </ul>
41          </div>
42          <button class="..." ng-click="vm.addUpcomingSemester()"></button>
43        </div>
44      </div>

```

Código 5.3: Extracto del archivo `planner.html` que corresponde a la plantilla de la interfaz planificador de malla curricular alternativa

AngularJS UI Tree

Tree es un componente de AngularJS UI que proporciona soporte para implementar una funcionalidad de *drag and drop* (arrastrar y soltar). Se utiliza en la interfaz de planificación con la finalidad de facilitar la acción de asociar una asignatura a un semestre, arrastrándola desde la sección de ramos pendientes y soltándola en la sección de asignación de ramos por semestre.

Según lo que se observa en el Código 5.3, cada semestre – ya sea de malla o futuro – se simboliza mediante un elemento `ui-tree`. Así, los nodos de un árbol dado (`ui-tree-nodes`) corresponden a las materias asignadas a dicho período.

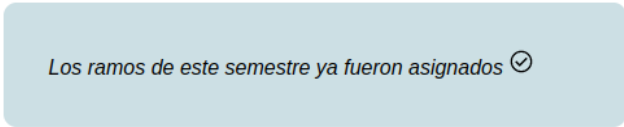
También, el paquete `angular-ui-tree` ofrece varias propiedades de gran utilidad para el desarrollo. Por ejemplo, el atributo `data-nodrop-enabled` sirve para evitar que se suelten nodos al interior del árbol. Entonces, en este caso se ocupa para impedir que se efectúen asignaciones que infringen relaciones de pertenencia a semestre en la malla (cuando se trata de la sección de ramos pendientes) o relaciones de prerequisite/ser prerequisite de (cuando se trata de la sección de asignación de ramos por semestre).

Las relaciones de prerequisite/ser prerequisite de se explican en mayor profundidad en la Sección 5.4.2.

Placeholder de semestre vacío

Un *placeholder* es un artefacto incluido temporalmente en la interfaz, con el propósito de reservar el lugar de un valor que no se conoce y que vendrá después.

Para el módulo planificador, se crean dos tipos de *placeholder*: el de semestre de malla vacío, y el de semestre futuro vacío. El *placeholder* de semestre de malla vacío se ubica en la sección de ramos pendientes. Como se muestra en la Figura 5.4, su mensaje señala “Los ramos de este semestre ya fueron asignados”, y por tanto, pretende notificar al usuario de la situación en la cual ha completado la asignación de las materias del período en cuestión.



Los ramos de este semestre ya fueron asignados ☑

Figura 5.4: Captura de pantalla de un elemento *placeholder* para un semestre vacío, perteneciente a la sección de ramos pendientes.

El *placeholder* de semestre futuro vacío se localiza en la sección de asignación de ramos por semestre. Como se exhibe en la Figura 5.5, su texto indica “Arrastra una asignatura desde la sección de ramos pendientes y suéltala aquí para asignarla a este semestre”, y por consiguiente, pretende comunicar al usuario de cómo poblar un semestre que se encuentra vacío.

① Arrastra una asignatura desde la sección de ramos pendientes y suéltala aquí para asignarla a este semestre

Figura 5.5: Captura de pantalla de un elemento *placeholder* para un semestre vacío, perteneciente a la sección de asignación de ramos por semestre.

Componente de asignatura

En la interfaz, cada asignatura se representa mediante un componente llamado **subject**. Los atributos de **subject** son el nombre, el código y la cantidad de créditos asociada a la asignatura dada. La apariencia visual del elemento descrito se manifiesta más adelante, en la Figura 5.6.

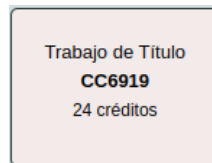


Figura 5.6: Captura de pantalla del aspecto visual de un componente de asignatura.

Teniendo en cuenta que la construcción del planificador involucra una gran cantidad de ramos, se elige usar componentes para implementarlos, dado que son herramientas esenciales en evitar la duplicación de código, y así se reduce la posibilidad de errores.

Tooltip para asignatura

Un *tooltip* es un elemento común de la interfaz gráfica de usuario (GUI), constituido por un cuadro de texto que se hace visible al desplazar el *mouse* sobre un elemento o componente de la pantalla. Este cuadro de texto muestra información sobre el elemento, por ejemplo, una descripción de la función de un botón o lo que significa una abreviatura.

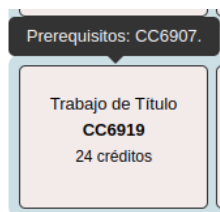


Figura 5.7: Captura de pantalla del aspecto visual de un *tooltip* de asignatura.

A cada componente de ramo se le asocia un *tooltip*, exceptuando el caso de asignaturas optativas o sin relaciones de prerequisite/ser prerequisite de. Dicho *tooltip* contiene un mensaje indicando las materias prerequisite de la asignatura relacionada y/o los ramos de los

que la asignatura relacionada es prerequisite (las relaciones de prerequisite/ser prerequisite de se esclarecen en la Sección 5.4.2). El aspecto visual del objeto descrito se presenta en la Figura 5.7.

5.4.2. Archivo de tipo controlador `planner.controller.js`

El archivo `planner.controller.js` se encarga de comunicarse con el archivo de tipo plantilla, a fin de manejar comportamientos y responder a acciones o eventos que se produzcan dentro de la interfaz.

Al activarse, lo primero que el controlador hace es obtener datos referentes a ramos pendientes y semestres próximos. Para ello, consulta a los servicios `pendingSubjectsService` y `upcomingSemestersService`, respectivamente. El funcionamiento de éstos se detalla a continuación.

Servicio de ramos pendientes `pendingSubjectsService`

En el servicio de ramos pendientes se localizan los métodos relacionados a la obtención del conjunto de asignaturas pendientes de un estudiante. Para conseguir ese cometido se requiere adquirir la lista completa de materias pertenecientes a la malla curricular de la carrera del estudiante, además de la lista de sus asignaturas pendientes.

Cabe mencionar que por el momento, y puesto que se trata de una versión inicial, la aplicación consta exclusivamente de código de *front-end*. En consecuencia, la creación de `pendingSubjectsService` se entiende como una primera aproximación a eventualmente lograr conectarse a un *back-end*, para recibir el conjunto de materias pendientes del estudiante y la malla curricular de su carrera de forma dinámica.

Mientras tanto, se opta por entregar la lista de identificadores de ramos pendientes mediante *hard coding*, es decir, escribiéndola directamente en el código. Esto se aísla en la función `getPendingSubjectsCodes`, con el objetivo de simplificar una posterior extensión del programa.

Por otra parte, la lista completa de materias propias de la malla curricular se consigue por medio de la invocación a la función `getCurriculum`, de `curriculumService`. Esta llamada retorna las asignaturas correspondientes a la carrera de Ingeniería Civil en Computación.

En `curriculumService` se encuentra disponible la versión `v3` y la versión `v5` de la malla. El parámetro de entrada del método ejecutado dictamina cuál de ellas se utiliza para obtener la respuesta.

Más adelante, el Código 5.4 enseña la estructura que tiene un elemento del parámetro de salida de `getCurriculum()`. Dicho método retorna un objeto tipo diccionario, el cual vincula un código identificador de ramo (por ejemplo: `CC6919`) a su correspondiente conjunto de atributos.

```

1 <codigo-identificador>: {
2   name: <nombre>,
3   type: <tipo>,
4   credits: <cantidad-creditos>,
5   semester: <numero-semester-malla>,
6   prerequisiteSubjects: [
7     { code: <codigo-prerrequisito>, type: <tipo-prerrequisito> },
8     ...
9   ],
10  restrictedSubjects: [
11    { code: <codigo-restriccion>, type: <tipo-restriccion> },
12    ...
13  ]
14 }

```

Código 5.4: Estructura de un elemento de la respuesta retornada por la función `getCurriculum` de `curriculumService`

Las propiedades incluidas en la respuesta son:

- **name**: Se refiere al nombre completo de la asignatura. Por ejemplo, para CC6919 el nombre completo es Trabajo de Título.
- **type**: Indica el tipo de ramo, es decir, si es obligatorio para la carrera estudiada o es de carácter electivo.
- **credits**: Indica la cantidad de créditos asociada a la asignatura.
- **semester**: Señala el número de semestre al que pertenece el ramo en la malla utilizada. Su valor es un número romano, entre I y XI.
- **prerequisiteSubjects**: Es una lista de objetos que representan materias prerrequisito de la asignatura en cuestión. Esto es, materias que deben ser aprobadas previo a inscribir la asignatura dada.

El tipo de prerrequisito puede ser consecutivo (debe ser aprobado el semestre anterior, a más tardar) o simultáneo (puede cursarse en paralelo, durante el mismo semestre), lo que se muestra mediante el atributo `type`.

En caso de no existir prerrequisitos, la lista se encuentra vacía.

- **restrictedSubjects**: Corresponde a una lista de objetos que representan ramos de las cuáles la materia en cuestión es prerrequisito. Esto es, ramos que no pueden ser inscritos hasta aprobar la materia dada.

También se conoce a esta relación como restricción, y sus posibles tipos son los mismos que los tipos de prerrequisito definidos anteriormente.

Si no existen relaciones de restricción, la lista está vacía.

Entonces, como se mencionó al inicio de la presente sección, el controlador del módulo planificador – al activarse – consulta por datos referentes a ramos pendientes, usando la siguiente instrucción: `pendingSubjectsService.getPendingSubjectsPerSemester()`. Un extracto de dicha función se presenta posteriormente, en el Código 5.5.

```

1 function getPendingSubjectsPerSemester() {
2   ... let curriculumSubjects = curriculumService.getCurriculum(
3     curriculumVersion);
4   let pendingSubjectsCodes = getPendingSubjectsCodes(EXAMPLE_ID_NUMBER);
5   pendingSubjectsCodes.forEach(function (subjectCode) {
6     let curriculumSubject = curriculumSubjects[subjectCode]; ...
7     let subjectObject = new SubjectModel(
8       curriculumSubject.name,
9       subjectCode, ...
10    );
11
12    if (pendingSubjectsPerSemester[semesterNumber]) {
13      pendingSubjectsPerSemester[semesterNumber].subjects.push(
14        subjectObject);
15    } else {
16      pendingSubjectsPerSemester[semesterNumber] = new SemesterModel(
17        semesterType, semesterNumber, [subjectObject]
18      );
19    }
20  });
21
22  ... sortingUtilsService.sortSubjectsByAttribute(
23    pendingSubjectsPerSemester[semesterId].subjects, 'code');
24  });
25  ... sortingUtilsService.sortCurriculumSemestersById(
26    pendingSubjectsPerSemester);
27  return pendingSubjectsPerSemester;
28 }

```

Código 5.5: Extracto de la función `getPendingSubjectsPerSemester` del servicio de ramos pendientes

Se advierte que se invoca a `curriculumService.getCurriculum(curriculumVersion)` y `getPendingSubjectsCodes(EXAMPLE_ID_NUMBER)`, para conseguir la lista completa de materias pertenecientes a la malla curricular y la lista de asignaturas pendientes, respectivamente. Notar que, provisionalmente, los parámetros `curriculumVersion` y `EXAMPLE_ID_NUMBER` se han escrito directamente en el programa. Se han dispuesto de esta manera con la intención de simplificar una posterior extensión de la aplicación.

Luego, `getPendingSubjectsPerSemester()` itera sobre la lista de códigos de ramos pendientes, usando cada elemento `subjectCode` para acceder a la lista de materias de la malla, y así obtener el conjunto de atributos de dicho ramo, `curriculumSubject` (mostrado en el Código 5.4, visto anteriormente). Después, esos atributos se almacenan en un objeto de tipo `SubjectModel`.

Para continuar, se verifica si el semestre al que `curriculumSubject` corresponde ya está incluido en la respuesta, en cuyo caso el objeto recién creado se agrega a la lista de asignaturas de dicho período. En caso contrario, se adiciona el ramo creado a una nueva instancia de un objeto de tipo `SemesterModel`, y esta última se incorpora a la respuesta.

Finalmente, se ordenan las listas de asignaturas según su código identificador, la lista de semestres según su número y se retorna el resultado. Un ejemplo que ilustra la estructura de la respuesta de la función `getPendingSubjectsPerSemester` se encuentra en el Código 5.6, a continuación.

```

1 {
2   "VII": {
3     "type": "CURRICULUM",
4     "number": "VII",
5     "subjects": [
6       {
7         "name": "Diseno y Analisis de Algoritmos",
8         "code": "CC4102",
9         "type": "mandatory",
10        "credits": 6,
11        "semesterInCurriculum": "VII",
12        "semesterAssignedInPlan": null,
13        "status": "PENDING",
14        "prerequisiteSubjects": [
15          {
16            "code": "CC3001",
17            "type": "consecutivo"
18          },
19          ...
20        ],
21        "restrictedSubjects": [],
22      },
23      ...
24    ],
25    "noDropEnabled": false,
26    "assignedCredits": null
27  }
28 }

```

Código 5.6: Extracto de estructura de la respuesta de `getPendingSubjectsPerSemester()` del servicio de ramos pendientes

Servicio de semestres futuros `upcomingSemestersService`

En el servicio de semestres futuros se ubica la función `getInitialUpcomingSemester`, la cual es llamada por el controlador del módulo planificador – al activarse – para obtener información concerniente al semestre más próximo. Se observa el contenido de dicha función en el Código 5.7, seguidamente.

```

1 function getInitialUpcomingSemester() {
2   let initialUpcomingSemester = {};
3   let semesterType = _getInitialUpcomingSemesterType();
4   let semesterNumber = _getInitialUpcomingSemesterNumber();
5   let semesterId = semesterNumber + ' ' + semesterType;
6
7   initialUpcomingSemester[semesterId] = new SemesterModel(
8     semesterType, semesterNumber, []
9   );
10  return initialUpcomingSemester;
11 }

```

Código 5.7: Función `getInitialUpcomingSemester` del servicio de semestres futuros

Se aprecia que el método calcula el tipo y número del período más próximo, a través de métodos auxiliares. El tipo de semestre puede ser `OTOÑO` o `PRIMAVERA`, y se determina a partir de la fecha actual, considerando la definición de dichas estaciones en el hemisferio sur. Por otra parte, el número del semestre se refiere al año más cercano, y se deduce igualmente a

partir de la fecha actual.

Luego de conseguir ambos valores, se crea un objeto de tipo `SemesterModel` y se retorna. Un ejemplo de ello se muestra más adelante, en el Código 5.8.

```
1 {
2   "2022 PRIMAVERA": {
3     "type": "PRIMAVERA",
4     "number": 2022,
5     "subjects": [],
6     "noDropEnabled": false,
7     "assignedCredits": 0
8   }
9 }
```

Código 5.8: Ejemplo de respuesta retornada por la función `getInitialUpcomingSemester` de `upcomingSemestersService`

En suma, al activarse, `planner.controller.js` logra obtener datos concernientes a ramos pendientes y semestres próximos, mediante la invocación al método `getPendingSubjectsPerSemester` de `pendingSubjectsService` y al método `getInitialUpcomingSemester` de `upcomingSemestersService`, respectivamente.

Funciones que manejan los movimientos de *drag and drop*

Para dirigir los movimientos de *drag and drop* efectuados sobre los asignaturas – simbolizadas por elementos `ui-tree-node` – el controlador de la interfaz de planificación posee una variable llamada `vm.dragControlListeners`, la cual se exhibe en el Código 5.9.

```
1 vm.dragControlListeners = {
2   accept: acceptListener,
3   dropped: droppedListener,
4   dragStart: dragStartListener,
5   dragStop: dragStopListener,
6   dragMove: dragMoveListener
7 };
```

Código 5.9: Conjunto de métodos *listener* de la funcionalidad *drag and drop*

Esta variable se compone de 5 funciones de tipo `listener`, cuyo propósito consiste en mantenerse escuchando los eventos `accept`, `dropped`, `dragStart`, `dragStop` y `dragMove`, que pueden producirse en cada uno de los elementos `ui-tree`. Dichos eventos surgen durante el lapso que transcurre entre arrastrar un contenedor de ramo desde un punto específico de la interfaz hasta soltarlo en algún otro punto.

Los métodos *listener* se ejecutan inmediatamente después de que se origina un evento dado, y contienen instrucciones que determinan las consecuencias de dicho evento en el módulo. Particularmente, los métodos más significativos son `dragMoveListener` y `droppedListener`.

El contenido de `dragMoveListener()` se muestra más adelante, en el Código 5.10:

```

1 function dragMoveListener(event) {
2   _resetNoDropRestrictions();
3   let subjectToBeMoved = event.source.nodeScope.$modelValue;
4   let semesterIdDest = event.dest.nodesScope.$element[0].id;
5
6   if (_isSubjectAllowedToDropInSemester(subjectToBeMoved, semesterIdDest)) {
7     if (_isSubjectMoveToAssignedSection(semesterIdDest)) {
8       handleSubjectMoveToAssignedSection(semesterIdDest);
9     } else {
10      handleSubjectMoveToPendingSection(semesterIdDest);
11    }
12  } else {
13    _hideSubjectToBeMovedPlaceholder();
14  }
15 }

```

Código 5.10: Función `dragMoveListener` de `vm.dragControlListeners` perteneciente al controlador del módulo de planificación

La función presentada se encarga de manejar el evento referente a un contenedor de asignatura siendo trasladado a lo largo de la pantalla. Es invocada cada vez que el elemento se posa sobre un punto dado (sin ser soltado), y cuando este punto corresponde a un semestre, verifica que esté permitido soltar el contenedor sobre él.

Si es así, se identifica hacia qué sección se está realizando el movimiento y se procede a desactivar la restricción `noDropEnabled` del período de destino (lo que posibilita que el traslado sea efectivo, si se suelta el ramo).

Usualmente, cuando una materia se arrastra hasta posicionarse sobre un semestre, en él se materializa una especie de *placeholder*, el cual indica la ubicación que tendrá la asignatura en caso de ser soltada (un ejemplo de lo recién descrito se aprecia en la Figura 5.8). Si no se acepta soltar el elemento, el *placeholder* es ocultado, con la finalidad de comunicar la restricción al usuario.

Asignación de ramos por semestre

Para obtener tu malla curricular alternativa, asigna tus ramos pendientes a se

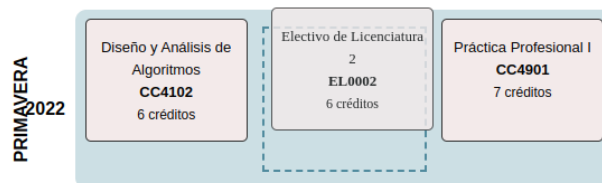


Figura 5.8: Captura de pantalla de un ejemplo del *placeholder* que se visualiza al arrastrar una asignatura y posarla sobre un semestre.

Los pasos a seguir para establecer si una asignación de ramo es legítima se explicarán en mayor profundidad posteriormente.

En el Código 5.11 se ubica el código referente a `droppedListener()`:

```

1 function droppedListener(event) {
2   let subjectToBeMoved = event.source.nodeScope.$modelValue;
3   let semesterIdSource = event.source.nodesScope.$element[0].id;
4   let semesterIdDest = event.dest.nodesScope.$element[0].id;
5
6   if (!_isSubjectAllowedToDropInSemester(
7     subjectToBeMoved, semesterIdDest, true)) {
8     if (!_isSubjectMoveToAssignedSection(semesterIdDest)) {
9       moveSubjectToAssignedSection(
10        subjectToBeMoved, semesterIdSource, semesterIdDest);
11     } else {
12       moveSubjectToPendingSection(subjectToBeMoved, semesterIdSource);
13     }
14   } else {
15     toastsService.push(
16       TOAST_ALERT_TYPE.error, vm.toastMessageForRejectedSubjectMove);
17   }
18   _resetNoDropRestrictions();
19 }

```

Código 5.11: Función `droppedListener` de `vm.dragControllListeners` perteneciente al controlador del módulo de planificación

El método mostrado responde al evento concerniente a un contenedor de asignatura siendo soltado en algún lugar de la pantalla. Nuevamente, comprueba que el movimiento sea legítimo, para determinar si efectivamente se puede proceder con él, o se requiere lanzar una alerta (es necesario realizar esta comprobación una segunda vez, dado que en la función `dragMoveListener` el único propósito de la verificación es decidir si se desactiva la restricción `noDropEnabled` o se oculta el *placeholder* de la materia).

Si el traslado de la asignatura está permitido, se realizan las siguientes operaciones:

- Se actualiza el estado del ramo, ya sea de `PENDING` a `ASSIGNED` o viceversa.
- Se actualiza el conteo de créditos para los semestres que corresponda. Esto es, se decrementa el contador del período de origen y se incrementa el contador del período de destino (para los períodos que se encuentren en la sección de asignación de ramos por semestre).
- Se actualiza el atributo del ramo `semesterAssignedInPlan`. Éste se convierte en `null` si la materia es depositada en la sección de ramos pendientes, o en el identificador del semestre de destino, si la materia se vincula a un semestre de la sección de asignación de ramos por semestre.
- Se agrega o elimina el ramo de la variable `vm.assignedSubjectsById`, según corresponda.

En caso de que el traslado de la asignatura sea ilegítimo, se genera una alerta usando un componente llamado `toast`. El contenido de la alerta se describirá en detalle más adelante.

Relaciones de prerrequisito y restricción entre asignaturas

La sección anterior hace referencia a comprobar que el traslado de una materia sea legítimo. Según lo observado en los Códigos 5.10 y 5.11, esto se lleva a cabo mediante la invocación a la función `_isSubjectAllowedToDropInSemester`. El extracto del programa referente a ella se encuentra en el Código 5.12, a continuación:

```
1 function _isSubjectAllowedToDropInSemester (  
2   subjectToBeMoved, semesterIdDest, isDroppedListenerCalling  
3 ) {  
4   if (_isSubjectMoveToAssignedSection(semesterIdDest)) {  
5     let arePrereqsFullfilled = arePrerequisiteSubjectRelationsFullfilled(  
6       subjectToBeMoved, semesterIdDest, isDroppedListenerCalling);  
7     if (!arePrereqsFullfilled) {  
8       return false;  
9     }  
10    return _areRestrictedSubjectRelationsFullfilled(  
11      subjectToBeMoved, semesterIdDest, isDroppedListenerCalling);  
12  } else {  
13    let isDestSemesterEqualToSemesterInCurriculum = (  
14      subjectToBeMoved.semesterInCurriculum === semesterIdDest);  
15    ...  
16    return isDestSemesterEqualToSemesterInCurriculum;  
17  }  
18 }
```

Código 5.12: Extracto de la función `_isSubjectAllowedToDropInSemester` del controlador del planificador

Considerando la implementación vista, se percibe que el hecho de que un traslado de asignatura sea permitido o no depende de la sección de destino de ésta. Si el ramo en movimiento se pretende asignar a un período de la sección de asignación de ramos por semestre, la verificación involucra las relaciones de prerrequisito y restricción entre asignaturas.

Antes de soltar una asignatura `subjectToBeMoved` en un período específico, se itera sobre sus ramos prerrequisito. Por cada uno de ellos, se verifica que se encuentren asignados, y si es así, se compara el semestre al cual están asignados, `prerequisiteSubject.semesterAssignedInPlan`, con el semestre al que se quiere asociar la asignatura `subjectToBeMoved`, es decir, `semesterIdDest`. Si el tipo de prerrequisito es consecutivo, debe cumplirse que la fecha de `prerequisiteSubject.semesterAssignedInPlan` sea estrictamente menor que la fecha de `semesterIdDest`. Para un tipo de prerrequisito simultáneo, la comparación se convierte en un menor o igual, es decir, ambos ramos deben al menos cursarse al mismo tiempo.

Si existe al menos un caso en el que no se cumple la condición descrita, la función `_isSubjectAllowedToDropInSemester` retorna `false`, indicando que la asignación que se ha intentado realizar no satisface las relaciones de prerrequisito, y por lo tanto se rechaza.

Por el contrario, si todos los prerrequisitos están cubiertos, se procede a revisar las relaciones de restricción entre asignaturas. Esto es, se revisan los ramos de los cuales la materia `subjectToBeMoved` es prerrequisito. Entonces, la metodología es similar a lo especificado previamente, exceptuando el hecho de que la comparación es estrictamente mayor o mayor o igual, según sea el caso.

Por otra parte, si el ramo en movimiento se pretende asignar a un semestre de la sección de ramos pendientes, la comprobación involucra contrastar el identificador del semestre de destino con el identificador del período al cual la asignatura `subjectToBeMoved` pertenece en la malla. Si éstos son idénticos, el traslado es legítimo y `_isSubjectAllowedToDropInSemester()` retorna `true`. Si no, el traslado es rechazado.

Alertas de rechazo de asignación de ramo

Si el método `_isSubjectAllowedToDropInSemester` retorna `false` cuando ha sido invocado por `droppedListener()`, se infiere que el movimiento deseado no puede ocurrir. Por consiguiente, se implementa una alerta que presenta un mensaje indicando la razón por la cual el traslado se ha rechazado. Ésta puede referirse al incumplimiento de las relaciones de prerrequisito, las relaciones de restricción o la pertenencia a un semestre en la malla. Un ejemplo de cada una se muestra en las Figuras 5.9, 5.10 y 5.11, respectivamente.

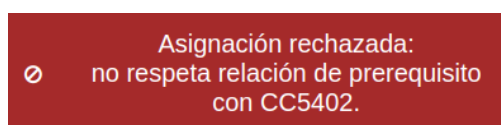


Figura 5.9: Ejemplo de alerta de rechazo de asignación de ramo, por incumplimiento de relación de prerrequisito.

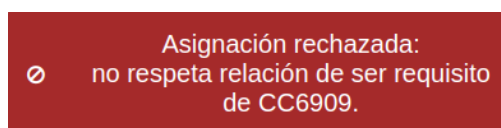


Figura 5.10: Ejemplo de alerta de rechazo de asignación de ramo, por incumplimiento de relación de restricción.

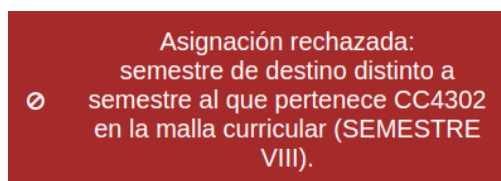


Figura 5.11: Ejemplo de alerta de rechazo de asignación de ramo, por incumplimiento de la pertenencia a un semestre en la malla.

5.5. Estrategias para preservar las buenas prácticas de programación

A continuación se mencionan brevemente algunas de las estrategias empleadas a lo largo del desarrollo, con el objetivo de favorecer las buenas prácticas de programación:

- **Archivos de constantes:** Se crean archivos de constantes (mediante la extensión `.constants.js`), dentro de los cuales se definen mensajes a mostrar en la interfaz, tal como lo ejemplifica el Código 5.13.

```
1 constant('INTERFACE_SENTENCE', {
2   'semester_title': 'SEMESTRE',
3   'credit_units': 'creditos',
4   'prerequisite_subjects_tooltip_message': 'Prerequisitos:
5     ${prerequisiteSubjects}.',
6   'restricted_subjects_tooltip_message': 'Es requisito de:
7     ${restrictedSubjects}.',
8   'accept_text': 'Aceptar'
9 })
10
```

Código 5.13: Extracto del archivo `index.constants.js` que ejemplifica la definición de una constante de tipo mensaje

Centralizar la información de esta forma tiene la ventaja de permitir efectuar actualizaciones fácil y rápidamente.

También, se definen constantes que representan el rango de una variable, tal como el tipo de un prerrequisito, el cual solo puede ser consecutivo o simultáneo, lo que se ilustra en el Código 5.14.

```
1 constant('REQUISITE_RELATION_TYPE', {
2   'consecutive': 'consecutivo',
3   'simultaneous': 'simultaneo'
4 })
5
```

Código 5.14: Extracto del archivo `index.constants.js` que ilustra la definición de una constante de tipo `enum`

Esto simula objetos de tipo `enum`, los que son útiles para restringir los posibles valores de una categoría. Así, se evitan inconsistencias generadas por errores de tipeo.

- **Documentación:** Se implementa el uso de documentación en inglés, ejemplificada en el Código 5.15. Se ubica al inicio de cada función, con el objetivo de explicar el propósito del método, y describir los parámetros de entrada y salida.

```
1 /**
2  * Check if subject to be moved is allowed to drop in given semester,
3  * considering different rules depending on if it's being moved to
4  * assigned section or pending subjects section.
5  * @param {SubjectModel} subjectToBeMoved - Subject being moved.
6  * @param {String} semesterIdDest - Identifier for semester that's
7  * potential destination for subject being moved.
8  * @returns {Boolean} True if subject is allowed to drop in semester,
9  * false otherwise.
10 */
11 function _isSubjectAllowedToDropInSemester(
12   subjectToBeMoved, semesterIdDest, isDroppedListenerCalling
13 ) {
14
```

Código 5.15: Ejemplo de uso de documentación extraído del archivo `planner.controller.js`

Además, se agregan comentarios esclarecedores en medio de la ejecución, cuando se considera necesario.

Se estima que esta metodología contribuye notoriamente a la legibilidad del código, y por lo tanto, a la mantenibilidad y extensibilidad de la herramienta.

- **Distinción entre funciones privadas y públicas:** Se adopta la convención de distinguir entre funciones privada y públicas, por medio de adicionar el prefijo `_` a métodos privados. De esta manera, se espera resguardar el encapsulamiento de cada módulo.
- **Definición de entidades:** Se opta por definir las clases `SubjectModel` y `SemesterModel` (expuesta en el Código 5.16), para representar las entidades referentes a asignaturas y semestres, respectivamente. Así, se consigue delimitar claramente cuáles son sus atributos, al igual que los tipos de éstos.

Asimismo, considerando que se va a reutilizar muchas veces la misma información, es provechoso para prevenir duplicación de código y consecuentemente, reducir errores.

```
1 function SemesterModel(SEMESTER_TYPE) {
2   class Semester {
3     constructor(type, number, subjects) {
4       this.type = type;
5       this.number = number;
6       this.subjects = subjects;
7       this.noDropEnabled = false;
8       this.assignedCredits = (type === SEMESTER_TYPE.curriculum)
9         ? null : 0;
10    }
11  }
12  return Semester;
13 }
14
```

Código 5.16: Extracto del archivo `semester.model.js` donde se encuentra la definición de la entidad de tipo semestre

- **Archivo para variables de estilo:** Se crea un archivo central llamado `variables.scss`, localizado en la raíz del proyecto. Éste se utiliza para almacenar definiciones de ciertas variables de estilo, las cuales se pretende reusar a lo largo de múltiples secciones. De este modo, se refuerza la consistencia en el aspecto visual de la interfaz.

Capítulo 6

Validación

El presente capítulo se trata sobre el proceso de validación final de la aplicación que se llevó a cabo. Se especifica la metodología utilizada (Sección 6.1), la cual consta de una evaluación de funcionalidad y una evaluación de usabilidad y utilidad. Además, se mencionan los resultados conseguidos (Sección 6.2) y se proporciona una interpretación de los mismos (Sección 6.3).

6.1. Metodología

Se estima que las características más significativas que podría tener la herramienta desarrollada son: que funcione correctamente, que sea fácil de usar y que sea útil. Contar con la primera implica que se cumple con los principales requerimientos establecidos en el Capítulo 3.

Por otra parte, lograr que la aplicación sea usable y útil se relaciona directamente con lo que declara el Modelo de Aceptación de Tecnología (TAM) [5]. La percepción de utilidad y la percepción de facilidad de uso son los factores más críticos en el proceso de adopción de tecnología. Por consiguiente, para un estudiante dado, mientras más útil considere que es la herramienta, más probable es que la utilice.

Luego, se opta por llevar a cabo una validación de la aplicación en tres ámbitos: funcionalidad, usabilidad y utilidad, detallados posteriormente.

6.1.1. Evaluación de funcionalidad

Con la finalidad de validar que el comportamiento del módulo planificador sea acertado, se efectúa una evaluación de funcionalidad con usuarios finales, es decir, estudiantes de la carrera de Ingeniería Civil en Computación.

Se conciben dos casos de prueba representativos, específicamente de conjuntos de ramos

pendientes, y se implementan de forma equivalente para las versiones v3 y v5 de la malla curricular. Dichos casos se exhiben a continuación, en los Códigos 6.1, 6.2, 6.3 y 6.4.

Los casos de prueba 1 y 2 – referentes a las versiones v3 y v5 de la malla, respectivamente – corresponden a una situación hipotética en la cual un estudiante, durante su quinto semestre, reprueba, elimina o no inscribe las siguientes asignaturas: “Algoritmos y Estructuras de Datos”, “Probabilidades y Estadística”, y un curso de formación integral. Entonces, el conjunto de ramos pendientes incluye las materias mencionadas, junto a las materias pertenecientes a los semestres siguientes.

```
1 ['CC3001', 'MA3403', 'FG0006', 'CC3301', 'CC3002', 'CC3102', 'CC3201',
2 'EL0001', 'CC4301', 'CC4101', 'CC4401', 'CC4901', 'EL0002', 'EL0003',
3 'CC4302', 'CC4303', 'CC4102', 'EL0004', 'EL0005', 'IN3301', 'CC5401',
4 'CC5901', 'EE0001', 'EE0002', 'EE0003', 'CC6908', 'CG0001', 'CC5402',
5 'EE0004', 'FE0001', 'CC6909', 'EE0005', 'FE0002']
```

Código 6.1: Caso de prueba 1

```
1 ['CC3001', 'MA3403', 'FG0004', 'CC3301', 'CC3002', 'CC3102', 'CC3201',
2 'EL0001', 'CC4900', 'CC5205', 'CC4101', 'CC4401', 'CC4901', 'EL0002',
3 'EL0003', 'CC4302', 'CC4303', 'CC4102', 'CC4402', 'EL0004', 'CC5900',
4 'CC5901', 'EE0001', 'EE0002', 'EE0003', 'CC6907', 'CG0001', 'CC5402',
5 'EE0004', 'FE0001', 'CC6919', 'EE0006', 'EE0005', 'FE0002']
```

Código 6.2: Caso de prueba 2

Los casos de prueba 3 y 4 – referentes a las versiones v3 y v5 de la malla, respectivamente – se relacionan a una situación hipotética en la cual un estudiante, durante su séptimo semestre, reprueba, elimina o no inscribe las siguientes materias:

- “Lenguajes de Programación”, “Práctica Profesional I”, y dos cursos electivos de licenciatura, considerando la versión v3 de la malla.
- “Diseño y Análisis de Algoritmos”, “Práctica Profesional I”, y un curso electivo de licenciatura, si se trata de la versión v5 de la malla.

```
1 ['CC4101', 'CC4901', 'EL0002', 'EL0003', 'CC4302', 'CC4303', 'CC4102',
2 'EL0004', 'EL0005', 'IN3301', 'CC5401', 'CC5901', 'EE0001', 'EE0002',
3 'EE0003', 'CC6908', 'CG0001', 'CC5402', 'EE0004', 'FE0001', 'CC6909',
4 'EE0005', 'FE0002']
```

Código 6.3: Caso de prueba 3

```
1 ['CC4101', 'CC4901', 'EL0002', 'EL0003', 'CC4303', 'CC4102', 'CC4402',
2 'EL0004', 'CC5900', 'CC5901', 'EE0001', 'EE0002', 'EE0003', 'CC6907',
3 'CG0001', 'CC5402', 'EE0004', 'FE0001', 'CC6919', 'EE0006', 'EE0005',
4 'FE0002']
```

Código 6.4: Caso de prueba 4

Entonces, el conjunto de ramos pendientes incluye las asignaturas mencionadas, junto a las asignaturas pertenecientes a los semestres siguientes.

Según lo presentado, se determinan tres actividades que los estudiantes entrevistados deben completar. Cada una de ellas indica un criterio particular que debe guiar su resolución. Las tareas a realizar se definen como sigue.

Actividad A

La actividad A consiste en crear una asignación de ramos pendientes a semestres, intentando alcanzar un equilibrio en cuanto a carga académica por período. Se aplica sobre el caso de prueba 1 o 2 (dependiendo de la versión de la malla cursada por el entrevistado o la entrevistada).

La tarea descrita tiene como propósito explorar la factibilidad de generar una planificación balanceada, cuando el estudiante no ha avanzado mayormente en la carrera, es decir, cuando se tiene una gran cantidad de ramos pendientes por organizar.

Actividad B

La actividad B involucra crear una asignación de ramos pendientes a semestres, evitando asociar asignaturas de gran carga académica al mismo semestre. Para esto, se emplea el caso de prueba 3 o 4 (dependiendo de la versión de la malla cursada por el entrevistado o la entrevistada).

Esta actividad pretende analizar la incidencia que tiene la carga académica percibida de las asignaturas en la acción de asociarlas a semestres futuros.

Actividad C

La actividad C se trata de crear una asignación de ramos pendientes a semestres, intentando minimizar la cantidad de semestres que tendrá la planificación. Para esto, se emplea el caso de prueba 3 o 4 (dependiendo de la versión de la malla cursada por el entrevistado o la entrevistada).

El objetivo de la labor mencionada es establecer la factibilidad de minimizar de manera efectiva la cantidad de semestres que tendrá una planificación. Esto es, establecer si es que la aplicación ofrece las funcionalidades requeridas para llegar al resultado correcto.

6.1.2. Evaluación de usabilidad y utilidad

Para evaluar la facilidad de uso y la utilidad percibida de la herramienta, se decide elaborar un cuestionario en escala de Likert, donde cada respuesta del o la estudiante revela

su grado de acuerdo o desacuerdo con la aseveración dada. Los valores posibles de la respuesta son: **totalmente en desacuerdo**, **en desacuerdo**, **neutral**, **de acuerdo**, **totalmente de acuerdo**.

El listado de afirmaciones que conforman la encuesta se confecciona basándose en distintos recursos disponibles, tales como la Escala de Usabilidad de un Sistema [4], y ajustándolos para ser más pertinentes a la situación propia.

Las aseveraciones que constituyen el cuestionario de evaluación de usabilidad y utilidad son las siguientes:

- P1. La organización de la información en la interfaz es clara.
- P2. Es fácil entender qué representa cada sección de la interfaz.
- P3. Es fácil entender para qué sirve cada elemento de la interfaz.
- P4. Las funcionalidades existentes me permiten planificar los ramos que tomaré los próximos semestres.
- P5. Los pasos a seguir para lograr planificar los ramos que tomaré los próximos semestres son intuitivos.
- P6. Me gustaría usar esta herramienta para planificar los ramos que tomaré los próximos semestres.
- P7. Creo que los y las estudiantes de la facultad aprenderían rápidamente como usar esta herramienta.
- P8. Esta herramienta es más simple de usar que mi metodología actual para planificar los ramos que tomaré los próximos semestres.

Cabe mencionar que, de la lista expuesta previamente, las afirmaciones P4 y P6 buscan medir utilidad percibida, mientras que el resto apunta a evaluar facilidad de uso.

De este modo, mientras mayor sea el grado de acuerdo del entrevistado con cada una de las declaraciones vistas, más positiva se concluye que fue su experiencia de uso.

Igualmente, se anima a los participantes a expresar sus apreciaciones en todo momento, durante la resolución de las actividades o para complementar sus respuestas al cuestionario. Así, se pretende conseguir una retroalimentación más detallada, lo que permite sacar conclusiones con mayor certeza.

6.2. Resultados

6.2.1. Usuarios participantes

En el proceso de validación del módulo planificador participan 4 estudiantes del Departamento de Ciencias de la Computación. Uno de ellos se encuentra cursando el sexto semestre

de la malla, otra ha inscrito asignaturas pertenecientes a los semestres séptimo y octavo, y las dos restantes están prontas a titularse. De los 4 participantes, tres utilizan la versión v3 de la malla, y uno utiliza la versión v5.

Los estudiantes se ofrecen voluntariamente a colaborar en esta instancia, respondiendo a un mensaje publicado en el grupo de Telegram del departamento. Las entrevistas se realizan de manera virtual, por conveniencia de todos los involucrados. Este hecho debe tomarse en consideración, puesto que potencialmente puede afectar los resultados.

La evaluación misma es antecedida por un par de preguntas de contextualización. De ellas, se deriva que la totalidad de los participantes ha experimentado dificultades al momento de escoger los ramos a inscribir en un semestre específico. Las razones varían: reprobado y/o eliminar asignaturas, irse de intercambio, o cursar dos especialidades a la vez.

También, se indaga sobre la estrategia o metodología usada para planear las materias que se tomarán los próximos semestres. De forma unánime, los estudiantes señalan que resuelven el problema planteado mediante análisis manual, en su mayor parte. Se apoyan de la malla curricular y la sección Mállame de Ucampus (referida en la Sección 2.1.2) para barajar alternativas y tomar decisiones, y una de ellos declara organizar su proyección en un archivo Excel, de creación propia.

6.2.2. Resultados evaluación de funcionalidad

Durante el desarrollo de las actividades de validación, la aplicación funciona correctamente y permite que los usuarios concluyan sus tareas sin problemas.

Actividad A

Al decir “equilibrio en cuanto a carga académica”, la instrucción de la actividad A es deliberadamente ambigua, con la intención de no sesgar la toma de decisiones del participante. Aun así, los resultados indican que todos infieren lo mismo: se refiere a asignar la misma cantidad de créditos a cada semestre, aproximadamente.

El total de los entrevistados consigue finalizar la tarea con éxito. En el peor de los casos, se presenta una diferencia de 10 créditos entre el semestre con mayor asignación y el semestre con menor asignación. Sin embargo, se advierte que dicha discrepancia está vinculada a alguna de las asignaturas de práctica profesional. La causalidad de esto se explica con mayor detalle en la Sección 6.2.4.

Es pertinente señalar que, mientras todos logran asignar de manera balanceada, no todos optan por asignar la misma cantidad de créditos. Una de las estudiantes escoge una cantidad menor, y por lo tanto, su planificación contiene un semestre adicional a la del resto.

Actividad B

La actividad B es realizada por 3 de los 4 participantes. El estudiante de sexto semestre manifiesta no tener muy claro cuáles son los ramos que implican mayor carga académica, por lo que se le indica que no la complete.

De los resultados obtenidos, se identifican reiteradamente ciertas materias como asignaturas de carga académica considerable: “Lenguajes de Programación”, “Diseño y Análisis de Algoritmos”, y “Proyecto de Software”. Esto sugiere que existe cierto consenso entre las opiniones de las entrevistadas.

No obstante, existen otras materias que son señaladas por una única persona. Por ejemplo, dicha persona considera que “Ingeniería de Software I” es un ramo demandante, y también, decide asignar “Proyecto de Software” e “Introducción al Trabajo de Título” a semestres distintos, debido a la alta carga académica percibida del segundo.

Actividad C

La totalidad de los estudiantes logra completar la Actividad C, pero solo dos de ellos encuentran la cantidad mínima de semestres requeridos para la planificación del caso en cuestión, es decir, 4 semestres.

El resto de los participantes construyen una proyección constituida por 5 períodos. Se percibe que eso se debe a haber priorizado el criterio de carga académica balanceada. Entonces, se revela cierta resistencia a sobrecargar los semestres, incluso como parte de un ejercicio hipotético.

6.2.3. Resultados evaluación de usabilidad

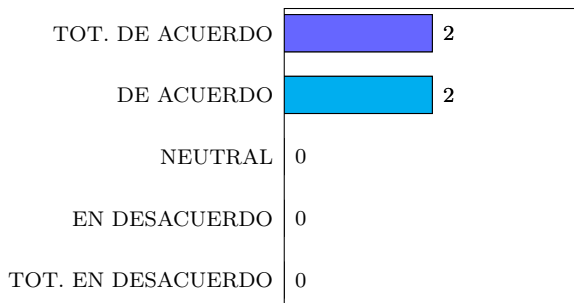
A continuación se reiteran las afirmaciones que conforman el cuestionario de evaluación de usabilidad:

- P1. La organización de la información en la interfaz es clara.
- P2. Es fácil entender qué representa cada sección de la interfaz.
- P3. Es fácil entender para qué sirve cada elemento de la interfaz.
- P4. Las funcionalidades existentes me permiten planificar los ramos que tomaré los próximos semestres.
- P5. Los pasos a seguir para lograr planificar los ramos que tomaré los próximos semestres son intuitivos.
- P6. Me gustaría usar esta herramienta para planificar los ramos que tomaré los próximos semestres.

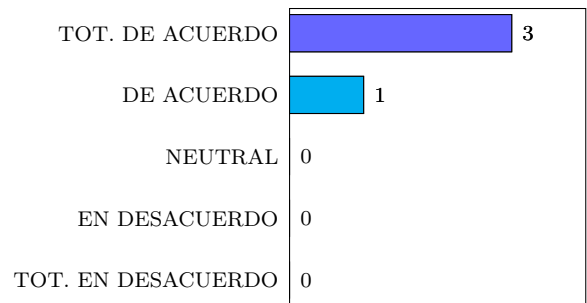
P7. Creo que los y las estudiantes de la facultad aprenderían rápidamente como usar esta herramienta.

P8. Esta herramienta es más simple de usar que mi metodología actual para planificar los ramos que tomaré los próximos semestres.

Más adelante, las Figuras 6.1, 6.2, 6.3 y 6.4 exhiben gráficos que ilustran la distribución de las respuestas para cada una de las aseveraciones listadas:

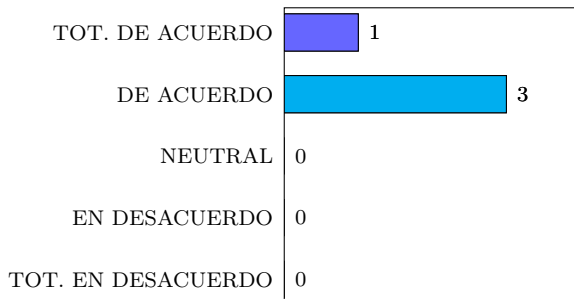


(a) Respuestas P1.

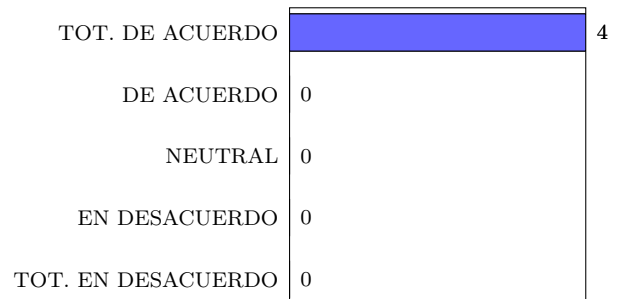


(b) Respuestas P2.

Figura 6.1: Gráfico distribución de respuestas a P1 y P2.

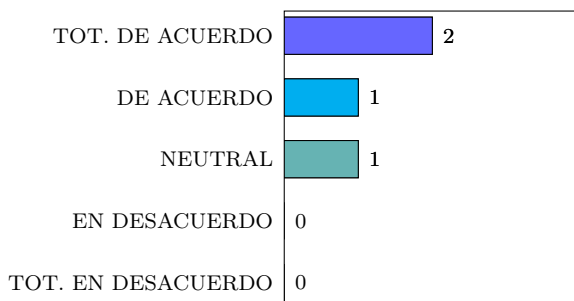


(a) Respuestas P3.

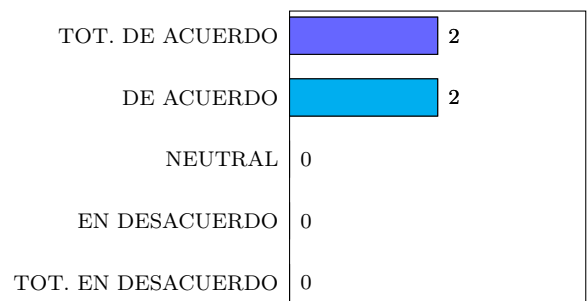


(b) Respuestas P4.

Figura 6.2: Gráfico distribución de respuestas a P3 y P4.



(a) Respuestas P5.



(b) Respuestas P6.

Figura 6.3: Gráfico distribución de respuestas a P5 y P6.

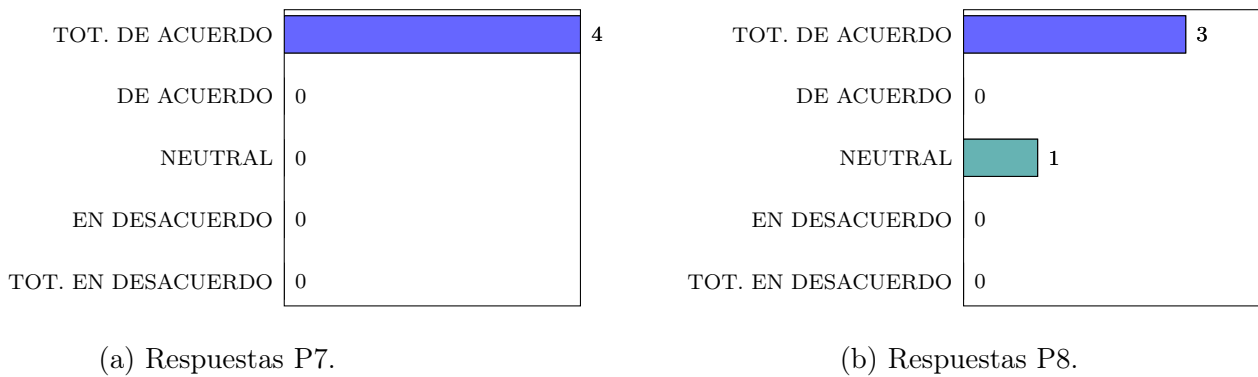


Figura 6.4: Gráfico distribución de respuestas a P7 y P8.

6.2.4. Retroalimentación adicional

A partir de las entrevistas de validación realizadas, surgen múltiples comentarios de retroalimentación sobre ciertos temas específicos, por lo que se considera conveniente prestarles especial atención. Estos comentarios se explican en mayor profundidad posteriormente.

Mejora de indicadores de prerequisites

Considerando un usuario que se encuentra resolviendo su planificación, se sugiere que sería provechoso idear una manera para comunicarle más claramente cuáles son los ramos que están habilitados para ser trasladados (porque se cumplen sus relaciones de prerequisite). Actualmente, no es posible determinarlo a simple vista, ya que se debe revisar el contenido del *tooltip* asociado a cada asignatura.

Además, se manifiesta reiteradamente que al visualizar una restricción de prerequisite (como las que se enseñan en las Figuras 5.7 y 5.10), se vuelve engorroso identificar a qué materia se hace referencia, dado que no es frecuente conocer todos los códigos de las materias de memoria. Se recomienda mostrar el nombre del ramo, junto a o en lugar del código.

Definición incompleta de tipos de prerequisites

Los entrevistados consultan sobre distintas particularidades concernientes a los tipos de relaciones de prerequisite, las cuales no se tenían contempladas previamente. Por ejemplo, se expresa la duda de si se permite inscribir más de 30 créditos al cursar el ramo “Trabajo de Título”. En caso de no ser así, se sugiere incorporar dicha restricción a la herramienta, puesto que se trata de un dato muy relevante para el o la estudiante. Además, se recomienda incluir la restricción de cantidad de créditos aprobados asociada a la asignatura “Introducción al Trabajo de Título”.

Se pregunta por los prerequisites de los ramos de formación integral de especialidad. A pesar de tratarse de entidades abstractas, se aconseja analizarlos con mayor detalle, ya que

suelen tener prerequisites que impiden inscribirlos en una etapa temprana de la carrera.

Finalmente, se alude a ciertos tipos de prerequisites que no han sido abarcados en la versión actual de la aplicación, tales como requisitos AUTOR. También, la posibilidad de realizar solicitudes al CAD que autoricen al alumno a inscribir asignaturas para las cuáles no cumple los requisitos, en circunstancias especiales. Se sugiere tener en cuenta dichas instancias.

Cantidad de créditos asociada a las asignaturas de práctica profesional

Durante el desarrollo de las actividades para evaluar funcionalidad, se descubre el siguiente dilema: las asignaturas de práctica profesional tienen asociadas 7 créditos, cada una. Los estudiantes están al tanto de que dichas prácticas suelen llevarse a cabo durante el verano, y que el trabajo a realizar durante el semestre se limita a escribir un informe, el cual usualmente se entrega a fin del primer mes.

Entonces, estas asignaturas causan algo de confusión en la asignación. Por ejemplo, puede que el estudiante planifique un semestre con 37 créditos (7 más de lo recomendado), considerando que la práctica “no cuenta”. Pero, al examinar su planificación, debe hacer un esfuerzo consciente para recordar que tener un semestre con alta carga académica asociada se debe a la asignación de la práctica. No es algo que pueda apreciar a simple vista. Por consiguiente, se vuelve imperioso analizar cómo se podría mejorar la situación descrita.

Botón para agregar semestre

Se señala que la finalidad del botón para agregar semestre – visualizado en la esquina inferior izquierda de la Figura 6.5 – no es tan evidente. Se estima que añadir una indicación lo haría más patente. También, se considera que su posicionamiento en la interfaz causa que tienda a pasar desapercibido.

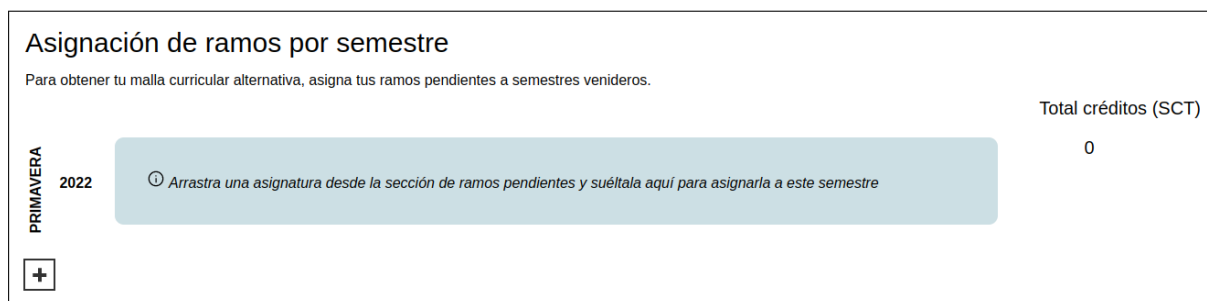


Figura 6.5: Captura de pantalla de la sección de asignación de ramos por semestre, inicialmente vacía.

Indicador total de créditos por semestre

En el lado derecho de la Figura 6.5 se observa el indicador de total de créditos por semestre. Se cree que su aspecto visual tiene potencial de mejora. En particular, se sugiere demarcar los valores mediante cuadros de texto, y acercarlos a los contenedores de sus semestres asociados.

6.3. Interpretación de resultados

De los resultados de la evaluación de funcionalidad se concluye que los usuarios entrevistados logran utilizar la herramienta sin mayor dificultad, y obtienen una planificación con la cual se sienten satisfechos.

Por otra parte, para interpretar las respuestas conseguidas en la evaluación de usabilidad se analizan los gráficos de distribución de respuestas exhibidos en las Figuras 6.1, 6.2, 6.3 y 6.4. Así, se advierte que las preguntas con mejores resultados son P4 y P7 (Figuras 6.2b y 6.4a), las cuales estipulan: “Las funcionalidades existentes me permiten planificar los ramos que tomaré los próximos semestres” y “Creo que los y las estudiantes de la facultad aprenderían rápidamente como usar esta herramienta”, respectivamente.

Por el contrario, los peores resultados se identifican en las preguntas P3 y P5 (Figuras 6.2a y 6.3a), que declaran: “Es fácil entender para qué sirve cada elemento de la interfaz” y “Los pasos a seguir para lograr planificar los ramos que tomaré los próximos semestres son intuitivos”, respectivamente.

Luego, se infiere que, desde el punto de vista de los participantes, la aplicación presenta un funcionamiento aceptable, aunque su usabilidad demuestra un cierto nivel de deficiencia.

Aun así, se estima que se logró cumplir el objetivo de la aplicación de manera aceptable, teniendo en cuenta que las apreciaciones recibidas son mayormente favorables. Con respecto a la retroalimentación adicional, expuesta en la Sección 6.2.4, se considera que la totalidad de los comentarios son extremadamente acertados y constituyen una perspectiva valiosa que permitirá continuar progresando en la usabilidad de la herramienta en el futuro.

Capítulo 7

Conclusiones

El trabajo realizado consistió en diseñar e implementar una solución al problema referente a los alumnos y alumnas de la facultad que se encuentran atrasados con respecto a su avance académico proyectado, quienes experimentan incertidumbre sobre sus procesos de Inscripción Académica futuros. La solución desarrollada es una aplicación web que ofrece una herramienta de visualización dinámica e interactiva, la cual posibilita definir una malla curricular alternativa que se ajuste a la situación particular de cada alumno.

Se estima que haber conseguido construir dicha aplicación implica que el objetivo general del proyecto (establecido en la Sección 1.3) se ha cumplido de forma aceptable, puesto que el software posee las funcionalidades mínimas para que un estudiante considere sus opciones y escoja un conjunto de asignaturas a cursar durante un semestre dado.

En cuanto a los objetivos específicos (declarados en la Sección 1.3), se cree que la mayoría de ellos se satisface en razonable medida. Sin embargo, es pertinente mencionar que los objetivos concernientes al diseño e implementación de los componentes de la herramienta no han sido cubiertos en su totalidad. Teniendo en cuenta la limitante de tiempo, no fue factible alcanzar a abarcar todos los elementos diseñados en un inicio. De todos modos, se priorizaron los más relevantes, y se desarrollaron de tal manera de facilitar su posterior extensión, por lo que la aplicación se considera igualmente un logro notable.

Según lo que se ha visto en el Capítulo 6, los resultados obtenidos sugieren que el software es usable, desde el punto de vista de usuarios finales. Pero, al mismo tiempo, se manifiestan ciertos puntos cruciales, especialmente los que conciernen a la implementación de las relaciones de prerequisites. Se piensa que tal vez habría sido conveniente efectuar un proceso de validación adicional, durante el desarrollo mismo, para así tomar consciencia de dicho problema en una etapa más temprano. No obstante, se estima que una decisión como esa supone un *trade-off*. Añadir una nueva tarea implica que el tiempo disponible para las demás disminuye, entonces, cómo afrontar la situación descrita se reduce a determinaciones subjetivas basadas en criterios personales.

Con respecto al impacto y la relevancia del proyecto desarrollado, se cree que ésta se encuentra fundamentalmente en el trabajo previo. Contar con un producto tangible es muy provechoso, pero lo más significativo que se puede destacar es el extensivo análisis del proble-

ma y su contexto, al igual que los procesos iterativos de diseño y validación efectuados. Así, fue posible perfeccionar los requerimientos y concretizar las ideas, para producir una primera aproximación a la solución. Igualmente, se preve que los resultados de esta etapa previa se convertirán en un pilar fundamental al momento de extender la herramienta, para eventualmente conseguir una aplicación robusta que tenga una incidencia real en la trayectoria universitaria de los alumnos y alumnas de la facultad.

Además de los conocimientos técnicos adquiridos, una de las principales lecciones aprendidas fue ser estratégica en la planificación de las etapas del proyecto, y también, saber priorizar los distintos aspectos del desarrollo, de acuerdo a cuáles conllevan la mayor ganancia en un análisis de costo versus beneficio (es decir, saber que si la implementación de un atributo va a tomar una gran cantidad de tiempo, pero su presencia va a pasar casi desapercibida, a la larga no vale demasiado la pena).

Para finalizar, en lo que concierne a un posible trabajo futuro, se sugiere considerar dos aspectos que se cree aportarán valor. El primero es enlazar la obtención de ramos pendientes a un sistema de *back-end*, para habilitar la funcionalidad de cuentas de usuarios, para contar con un sistema de autenticación de los mismos, y para obtener la información de forma dinámica. Según lo que se considere más conveniente, se debería tomar una decisión en cuanto a constituir este sistema de manera completamente independiente, o conectarlo con la plataforma Ucampus. Se estima que este es el paso siguiente más lógico, dado que inmediatamente permitiría que la aplicación se convierta en productiva, es decir, sería factible que se utilice en circunstancias reales, no simuladas.

El segundo aspecto que sería una gran contribución en un trabajo futuro se refiere a mejorar la usabilidad de la interfaz. Tomando en cuenta la retroalimentación recibida en la fase de validación final (detallada en el Capítulo 6), se considera una labor esencial. Se sugiere enfocarse a repensar la implementación de las relaciones de prerrequisitos y cómo éstas se comunican al estudiante, con el objetivo de entregar una experiencia de usuario notoriamente más intuitiva.

Bibliografía

- [1] Comisión Nacional de Acreditación CNA-Chile. Aprueba criterios de evaluación para la acreditación de carreras profesionales, carreras profesionales con licenciatura y programas de licenciatura. <https://www.cnachile.cl/Criterios%20y%20Procedimientos/DJ%20009-4%20Criterios.pdf>, (visitada el 25/04/2022).
- [2] Comisión Nacional de Acreditación CNA-Chile. Glosario de términos complementarios criterios de acreditación de pregrado. <https://www.cnachile.cl/documentos%20de%20paginas/glosario%20pregrado.pdf>, (visitada el 25/04/2022).
- [3] Craig Larman. *UML y Patrones, Introducción al análisis y diseño orientado a objetos*. Prentice Hall, 1999.
- [4] John Brooke. *SUS: A quick and dirty usability scale*, 1995.
- [5] Pedro Palos-Sanchez, Ana Reyes-Menendez & Jose Ramon Saura. Modelos de adopción de tecnologías de la información y cloud computing en las organizaciones. In *Información tecnológica*, volume 30, pages 3–12, 2019.
- [6] Pontificia Universidad Católica de Chile. Banner UC: Infórmate y planifica tu toma de ramos. <https://www.ing.uc.cl/boletines/banner-uc-informate-y-planifica-tu-toma-de-ramos/>, (visitada el 25/04/2022).
- [7] Secretaría de Estudios, Facultad de Ciencias Sociales, Universidad de Chile. Instructivo inscripción de asignaturas pregrado primer semestre, año 2022. http://facso.uchile.cl/dam/jcr:d2ed1a42-91d5-4771-88d3-a7f6929a5118/Instructivo_Inscripcion_Asignaturas_1%C2%B0_semestre_2022_PREGRADO.pdf, (visitada el 25/04/2022).
- [8] Secretaría de Estudios, Facultad de Economía y Negocios, Universidad de Chile. Inscribe tus ramos. <https://secretariaestudios.fen.uchile.cl/inscribe-tus-ramos>, (visitada el 25/04/2022).
- [9] Secretaría de Estudios, Facultad de Economía y Negocios, Universidad de Chile. Toma de ramos. <https://secretariaestudios.fen.uchile.cl/procesos-academicos/proceso-toma-de-ramos>, (visitada el 25/04/2022).
- [10] Universidad Andrés Bello. Preguntas frecuentes. <https://www.unab.cl/inscripcion-de-ramos/preguntas-frecuentes/>, (visitada el 25/04/2022).

- [11] Universidad Andrés Bello. Tutorial de toma de ramos. <https://www.unab.cl/wp-content/uploads/2017/09/tutorial-inscripcion-de-asignaturas-pregrado-diurno-y-vespertino-alumnos-antiguos-unab.pdf>, (visitada el 25/04/2022).
- [12] Universidad de Chile, Facultad de Arquitectura y Urbanismo,. Inscripción de asignaturas. <http://www.fau.uchile.cl/estudiantes/preguntas-frecuentes/inscripcion-de-asignaturas>, (visitada el 25/04/2022).
- [13] Universidad de Los Andes. Toma de ramos. <https://www.uandes.cl/toma-de-ramos/>, (visitada el 25/04/2022).
- [14] Universidad del Desarrollo. Calendario académico. <https://alumnos-scl.udd.cl/calendario-academico/>, (visitada el 25/04/2022).
- [15] Universidad Técnica Federico Santa María. Proceso de matrícula e inscripción de asignaturas, primer semestre 2022. https://autoservicio.usm.cl/docs/usmtransparente/informaciones/Instructivo_Proceso_de_Matricula_e_Inscripcion_de_Asignaturas_2022_1.pdf, (visitada el 25/04/2022).
- [16] Universidad Técnica Federico Santa María. ¿Qué es SIGA? https://sig.usm.cl/pag/portada_que_es.html, (visitada el 25/04/2022).