



UNIVERSIDAD
DE LA REPÚBLICA
URUGUAY



FACULTAD DE
INGENIERÍA

Elaboración de soporte tecnológico para el diseño de cursos

Informe de Proyecto de Grado presentado por

Etcheverry, Nestor
Hernandez, Pablo
Peláez, Sebastián

en cumplimiento parcial de los requerimientos para la graduación de la carrera
de Ingeniería en Computación de Facultad de Ingeniería de la Universidad de
la República

Supervisora: Regina Motz

Usuario experto: Mariana Porta

Montevideo, Viernes 31 de Marzo, 2023

Resumen

Se presenta el desarrollo de una herramienta tecnológica que brinda apoyo a los docentes en el diseño de cursos de enseñanza superior. La herramienta proporciona un espacio tecnológico con textos de ayuda y sugerencias que acompañan al docente en el proceso, permitiendo la personalización de los cursos según las necesidades de cada institución.

El desarrollo se realizó siguiendo un proceso de co-diseño trabajando junto con un usuario experto en el dominio de educación identificando problemas, necesidades y expectativas. Además, se aplicó la metodología ágil SCRUM para el desarrollo del software.

El resultado obtenido es un software desarrollado con Angular que se compila como una aplicación de escritorio para Windows, Linux y MacOS utilizando Electron. El diseño de la interfaz del software es flexible, ya que permite adaptarse a los requerimientos de cada curso e institución sin necesidad de desarrollar software adicional para ello, gracias a un esquema personalizable.

El resultado es una herramienta tecnológica que apoya al docente en el diseño de un curso a través de las 3 etapas de: elaboración del programa, implementación y dictado del curso.

Palabras clave: Diseño curricular, Co-diseño, Metodología ágil.

Índice

1. Introducción	4
1.1. Objetivo y actividades	5
1.2. Resultados Obtenidos	5
1.3. Organización del documento	6
2. Conceptos en el diseño de cursos	7
2.1 Diseño Curricular y Diseño Instruccional	7
2.2 Herramientas para los conceptos de diseño de cursos	8
2.2.1. ILDE	9
2.2.2. Edu-ÁREA	12
2.3 Conclusiones	13
3. Metodología de trabajo	14
3.1 Forma de trabajo en etapa de Diagnóstico y Análisis	15
3.2 Forma de trabajo en etapa de Diseño e Implementación	16
4. Resultados del Diagnóstico y Análisis	19
4.1. Resultados del Análisis	22
5. Resultados del Diseño e Implementación	24
5.1 Maquetación para el diseño de una solución	24
5.1.1 Pantalla de Inicio	24
5.1.2 Resultados de la primer Maqueta	26
5.1.3 Resultados de la segunda Maqueta	27
5.1.4 Resultados de la tercer Maqueta	29
5.1.5 Resultados de la maqueta final	32
5.1.6 Panel de control y menú	33
5.1.7 Área de trabajo	36
5.1.8 Estándar para atributos	38
5.1.9 Puntos críticos	40
5.2 Arquitectura para la implementación de la solución propuesta	41
5.3 Elección del Software de Desarrollo	43
5.4 Elección de la Base de Datos	45
5.4.1 Base de datos de documentos JSON	46
5.5 Servidor Express para una aplicación Angular	47
5.6 Frontend	47
5.7 Instalación de las Herramientas	48
5.8 Interfaz Gráfica	48
5.8.1 Desarrollo del Menú (Grafo)	49
5.8.2 Esquema de la Interfaz	49
5.8.3 Atributos con Herencia	52
5.8.4 Atributos con Orquestación	54
5.8.5 Datos fijos	57
5.9 Estructura de los datos	58
5.9.1 Sistema de guardado automático de información	60
5.9.2 Almacenamiento de Documentos	60

5.10 Versionado	62
5.11 Exportar e importar cursos	63
5.12 Comentarios	66
5.13 Sitio web de presentación	67
6. Evaluación de usabilidad	70
6.1 Resultados	71
6.2 Sugerencias	73
7. Conclusiones y Trabajos futuros	75
8. Referencias	79
9. Glosario	80
A. Anexo Formularios de Programa	83
A.1 Formulario de nuevo programa FCS 2022	83
A.2 Formulario Tercer semestre del Ciclo Inicial de la Facultad de Ciencias Sociales	86
A.3 Formulario Humanidades	86
B. Anexo Fase de análisis - Documento de conceptos	88
C. Anexo Sprints	94
D. Anexo Requerimientos	95
D.1 Requerimientos Funcionales	95
Autenticación y autorización de usuarios	95
Permisos sobre el diseño	96
Permisos anónimos	96
Permisos nominados	96
Etapas	96
Actor Administrador	96
Actor Docente	96
Etapa Elaboración del programa	96
Etapa Contextualización	96
Etapa Dictado	96
Almacenamiento y edición	97
Reutilización	97
Exportar	97
Compartir diseño de curso mediante repositorio	97
Compartir diseño de curso fuera de línea	97
Sugerencias	97
Enviar sugerencias	97
Recibir sugerencias	97
Recolección de datos	97
Nivel básico	98
Nivel básico con documento	98
Nivel avanzado	98
Analítica	98
Alertas	98
Versionado	98
D.2 Requerimientos no Funcionales	98

Requerimientos de interfaces externas	98
Interfaces de usuario	98
Interfaces de hardware	99
Interfaces de software	99
Tiempos de respuesta	99
Permisos	99
Seguridad	99
Derechos de autor	99
Manual de usuario	99
Mensajes de error	99
Comentarios privados (notas)	100
E. Anexo Encuesta de Usabilidad	100
E.1 Formulario de la encuesta	100
E.2 Respuestas recibidas	103

1. Introducción

El proceso de propuesta de cursos (obligatorios, opcionales, electivos o de educación permanente) en la Universidad de la República es una actividad que realizan los docentes para que un curso sea aprobado para su dictado dentro de un currículum. Este proceso implica la planificación y organización de los objetivos de aprendizaje, los contenidos, los métodos de enseñanza, las actividades, las evaluaciones y los recursos necesarios para el curso. En la elaboración de la propuesta del curso, el docente debe tener en cuenta los conocimientos previos exigidos a los estudiantes, los objetivos institucionales y los estándares de enseñanza y aprendizaje que se deben cumplir. El resultado de este proceso es el Programa del curso.

Para presentar el programa del curso, los servicios universitarios ofrecen un formulario que se completa con información sobre la carrera, edición, plan, número de créditos, carga horaria, equipo docente, conocimientos requeridos y modalidad de enseñanza, así como también los siguientes elementos para detallar el contenido del curso: fundamentación, objetivos, contenido, sistema de evaluación y bibliografía. Esta información se completa en Word u otro editor de texto y algunos ejemplos de estos formularios se pueden ver en el [Anexo A](#).

Sin embargo, se observa que a pesar de tener los programa de curso un formato digital no son utilizados en procesamientos automatizados. Por ejemplo, al ser formularios estáticos, no se realiza ningún tipo de control automático de consistencia en los datos ingresados, ni tampoco son parte de ningún sistema de control de versiones que puedan utilizar los docentes. Adicionalmente, ante la necesidad de rediseñar la modalidad de los cursos presenciales por motivos de la pandemia del COVID-19, fue necesario repensar los elementos especificados en estos programas. Aparece así el Proyecto de Mejoramiento de la Calidad de la Enseñanza Universitaria y la Universidad Digital (Proyecto UD) (Porta et al., 2021)) donde se establecen guías para la incorporación de tecnologías a los cursos desde la concepción de sus programas.

Según el estudio reportado en (Porta et al, 2021), se destaca la gran carga de estrés y desgaste experimentada por los docentes durante la transformación digital de sus cursos. Muchos de ellos carecían de conocimiento previo sobre las plataformas digitales y recursos educativos, así como de formación específica en el uso de tecnologías en el aula. Estas condiciones limitan fuertemente las actualizaciones a realizar en los Programas de cursos. Para lograr estos objetivos, entre otras acciones, se creó un curso denominado Rediseñando la Universidad Digital (RUD) (Porta et al. 2021).

El desarrollo de este proyecto de grado toma como antecedente directo el curso RUD, trabajando en modalidad de co-diseño con la docente Mariana Porta. Se partió de los resultados obtenidos en el dictado del curso RUD, el cual consta de las siguientes 3 etapas:

1. Recuperación de la participación docente e identificar las herramientas que se han utilizado.

2. Utilizar una rúbrica ya validada, para identificar áreas de mejora respecto a la dimensión digital de la propuesta formativa.
3. Orientar el proceso de rediseño, que se propone a través de cinco pasos.

Como producto del RUD, los docentes participantes generan un nuevo curso, rediseñado, mejorando, adaptado y transformado sus cursos con la incorporación de tecnología para enfrentar los desafíos de la pandemia.

El objetivo del presente Proyecto de Grado para Ingeniería en Computación es brindar soporte tecnológico a la etapa 3 del curso RUD, que consiste en el proceso de rediseño de un curso. Para lograrlo, se ha desarrollado un software que permite sistematizar el proceso de diseño de curso, teniendo en cuenta las etapas de definición de la propuesta del curso (programa), definir la contextualización del curso y su dictado. Este software guiará al docente en las tres etapas del proceso de diseño de curso, con un enfoque especial en el proceso de rediseño.

1.1. Objetivo y actividades

El objetivo principal de este proyecto es diseñar e implementar una herramienta tecnológica adaptable a las realidades de las instituciones de enseñanza superior para asistir a los docentes en el diseño de cursos permitiendo además dar trazabilidad entre los distintos momentos del ciclo de vida de un curso (elaboración del programa, contextualización y dictado).

Este proyecto, además de ofrecer una herramienta tecnológica, también lleva a cabo un análisis detallado sobre el actual proceso de diseño de un curso y las etapas que el docente debe seguir para alcanzar su objetivo.

Para alcanzar los objetivos planteados, el desarrollo de este proyecto realiza las siguientes actividades:

- Análisis detallado del proceso actual de diseño de cursos para identificar áreas de mejora y oportunidades de optimización.
- Identificar y documentar las etapas críticas del proceso de diseño del curso para ayudar a los docentes a diseñar sus cursos.
- Analizar los procesos que permitan a los docentes compartir y colaborar en el diseño de cursos.
- Desarrollar una interfaz de usuario intuitiva que permita a los docentes diseñar cursos sin la necesidad de un tutorial.
- Desarrollar un perfil de administrador que permite personalizar los requerimientos de los cursos de acuerdo a las normativas de las instituciones educativas.
- Ofrecer textos y guías de ayuda que acompañen al docente durante todo el proceso de diseño del curso.

1.2. Resultados Obtenidos

Los resultados obtenidos en el proyecto son:

- Diseño e implementación de una herramienta tecnológica para apoyar a los docentes en el diseño de cursos de enseñanza superior que presenta de forma integrada las tres etapas del curso: elaboración del programa, contextualización y dictado.
- Un sistema de ayuda en línea incorporado en la herramienta y adaptable a las necesidades de cada institución educativa para acompañar al docente en todo el proceso de diseño del curso.
- El software desarrollado: DTC (acrónimo de Diseña Tu Curso) está disponible para ser usado por docentes descargándolo para distintos sistemas operativos desde el sitio: <https://disenatucurso.netlify.app/>
- El código del software es código abierto y puede ser accedido en: <https://github.com/disenatucurso/disenatucursodocente>

1.3. Organización del documento

El resto de este documento se organiza de la siguiente forma:

- Capítulo 2: Presenta los conceptos que intervienen en el diseño de cursos.
- Capítulo 3: Presenta la descripción de la metodología de trabajo realizada durante el desarrollo de este proyecto.
- Capítulo 4: Presenta los resultados obtenidos durante la etapa de diagnóstico y análisis desarrolladas en este proyecto.
- Capítulo 5: Presenta los resultados obtenidos durante la etapa de diseño e implementación del proyecto. Se presenta la arquitectura de software para la solución propuesta. Justificación del software elegido para el desarrollo, así como también la estructura de los datos. También se explica el armado de la interfaz gráfica y el frontend. Por último, se presenta la descripción en la solución de ciertas particularidades para la solución de algunos requerimientos que necesitan un tratamiento especial, tales como el versionado y la exportación e importación de cursos.
- Capítulo 6: Evaluación de usabilidad de la solución propuesta.
- Capítulo 7: Conclusiones y trabajos futuros.
- Capítulo 8: Presenta las referencias usadas.
- Finalmente se encuentran el Glosario y los Anexos.

2. Conceptos en el diseño de cursos

El proceso de diseñar un curso comienza con la identificación de las necesidades y los objetivos de aprendizaje de los estudiantes y de la institución educativa, para luego determinar los recursos necesarios, los medios de enseñanza y los métodos de evaluación adecuados para alcanzar estos objetivos. Esto implica la selección de estrategias de enseñanza y aprendizaje apropiadas, el desarrollo de materiales didácticos y la creación de actividades de aprendizaje efectivas. Estos procesos se reconocen como Diseño Curricular y Diseño Instruccional.

2.1 Diseño Curricular y Diseño Instruccional

El Diseño Curricular se identifica en ocasiones con el concepto de planeamiento o con el currículo en su integridad (Arnaz, 1981). Es el momento importante de la determinación del perfil y la determinación de los contenidos y la metodología que se aplicará.

Dentro del diseño curricular se encuentran las etapas de:

- **Diagnóstico de problemas y necesidades** consiste en el proceso de determinación del marco teórico sobre los cuales se va a diseñar la unidad curricular.
- **Modelación del currículo** aquí se conceptualiza el modelo de la unidad curricular. Se busca cual es el criterio que adopta respecto a la propuesta de la institución educativa, el docente y el estudiante. Es el momento importante de la determinación del perfil y la determinación de los contenidos y la metodología.
- **Estructuración de la unidad curricular**, en este paso se determina el orden en que se va a desarrollar la unidad curricular, el lugar de cada componente, el tiempo necesario, las relaciones de precedencia e integración y todo lo que tiene que ver con un mapa curricular.
- **Organización para la puesta en práctica de la unidad curricular.** Esta tarea provee las medidas para garantizar la práctica de la unidad curricular.
- **Diseño de la evaluación de la unidad curricular.** Proceso donde se comprueba la validez del diseño, se determina en qué medida su implementación práctica y resultado satisfacen las necesidades de los estudiantes y de las instituciones educativas. La evaluación curricular ocurre inicialmente como una evaluación de diagnóstico, en la etapa pre activa del curso, donde el propósito principal es determinar si las condiciones para ejecutar la unidad curricular están dadas o si deben ser creadas. Durante la etapa activa de la unidad curricular ocurre una evaluación formativa o continua del proceso de enseñanza y aprendizaje. Esta evaluación tiene una función reguladora ya que estudia aspectos de la unidad curricular que no están funcionando y propone alternativas para mejorar los mismos. En la etapa pos activa de la unidad curricular se realiza una evaluación sumativa del proceso de enseñanza y aprendizaje que permite la toma de decisiones respecto a la unidad curricular para, cancelarlo, mejorarlo o rediseñar. El Diseño Instruccional por otra parte se identifica con el proceso de planificar, diseñar, desarrollar, implementar y evaluar materiales y actividades de aprendizaje para lograr objetivos educativos específicos (Barriga, 2005). En el trabajo de (Belloch, 2012) se analizan diferentes definiciones de diseño instruccional de diferentes autores. Algunos lo

definen como la disciplina que se ocupa de la planeación, la preparación y el diseño de los recursos y ambientes necesarios para que se lleve a cabo el aprendizaje. Otros lo definen como la ciencia de creación de especificaciones detalladas para el desarrollo, implementación, evaluación y mantenimiento de situaciones que facilitan el aprendizaje de pequeñas y grandes unidades de contenidos, en diferentes niveles de complejidad.

El informe *Technology-Based Instructional Design - Evolution and Major Trends* (Paquette, 2012) presenta una visión general del campo del Diseño Instruccional basado en tecnología, que se refiere al uso de ésta en la planificación y creación de entornos de aprendizaje. El autor describe la evolución del campo desde sus orígenes en la década de 1960 hasta el presente, y destaca las principales tendencias en la investigación y la práctica. Según este trabajo, el Diseño Instruccional se refiere a la creación de planes y modelos de instrucción que ayudan a los estudiantes a adquirir conocimientos y habilidades. El uso de la tecnología ha permitido a los diseñadores instruccionales crear entornos de aprendizaje más efectivos y personalizados para los estudiantes. El informe describe el ciclo de vida de un entorno de aprendizaje, que incluye la planificación, el diseño, la producción y la entrega del sistema, así como el mantenimiento y la revisión continua. El autor también describe varios paradigmas y enfoques en el Diseño Instruccional Basado en Tecnología, incluyendo herramientas de autor, sistemas expertos, tutoría inteligente, estándares de eLearning y entornos de aprendizaje basados en la web. Se presentan ejemplos de cada paradigma y se exploran las principales tendencias en la investigación y la práctica.

A pesar que la propuesta original del rediseño en 5 pasos desde donde partió este trabajo no diferenciaba claramente entre el diseño curricular y el diseño instruccional, podemos decir que luego de realizar las etapas de diagnóstico y análisis descritas en el Capítulo 4, la herramienta que se desarrolló en este Proyecto se encarga de asistir al docente en el Diseño Curricular. Sin embargo, intentando enlazar el diseño curricular con el diseño instruccional, se avanza en la identificación de los que se dió en llamar “punto críticos”, considerados como los elementos del diseño curricular en los cuales el docente puede necesitar una ayuda especial. Los puntos críticos abarcan, desde entender cómo se calculan los créditos asignados a los estudiantes por curso hasta tener referencias de lo que es el licenciamiento de un recurso educativo. El concepto de punto crítico definido en este trabajo se presenta en más detalle en el Capítulo 4 de Resultados de la etapa de diagnóstico y análisis.

2.2 Herramientas para los conceptos de diseño de cursos

Teniendo en cuenta el marco teórico expuesto, se realiza un relevamiento de herramientas tecnológicas existentes que apoyen el diseño de cursos.

Encontramos las herramientas EdApp¹, Adobe Captivate² e iSpring Suite³, todas con propuestas adecuadas a la especificación del ambiente de trabajo, orientadas al diseño instruccional pero sin permitir definir los objetivos del curso ni especificar el modelo pedagógico que se seguirá en el mismo. Estos aspectos son clave en la planificación de

¹ <https://www.edapp.com/>

² <https://www.adobe.com/es/products/captivate.html>

³ <https://www.ispring.es/ispring-suite>

cualquier diseño curricular y por lo tanto, limitan la efectividad de estos software en este sentido. Debido a estas razones, no se profundizó en la investigación de estos software y pasamos a analizar los entornos de trabajo ILDE y Edu-ÁREA que prometían herramientas para trabajar los aspectos de colaboración entre docentes, y de licenciamiento, respectivamente, ambos aspectos también de interés para este proyecto.

2.2.1. ILDE

El entorno de diseño de aprendizaje integrado, ILDE⁴ es un proyecto desarrollado por el grupo de investigación en tecnología educativa de la Universitat Pompeu Fabra de Barcelona (UPF). ILDE es el acrónimo de "Integrated Learning Design Environment", es decir, un entorno integrado para el diseño de aprendizaje.

Se trata de una plataforma en línea que permite a los usuarios crear y compartir materiales educativos interactivos y multimedia, como juegos, simulaciones, mapas conceptuales, entre otros. ILDE está diseñado para facilitar el proceso de creación de materiales educativos y para promover la colaboración entre docentes y estudiantes en el diseño y desarrollo de recursos educativos. Además, ILDE cuenta con una amplia variedad de recursos y herramientas que ayudan a los usuarios a crear materiales educativos atractivos e interactivos.

En la Figura 1 se puede observar la pantalla principal de ILDE. Allí se accede a las principales funcionalidades del software y a los diseños creados anteriormente.

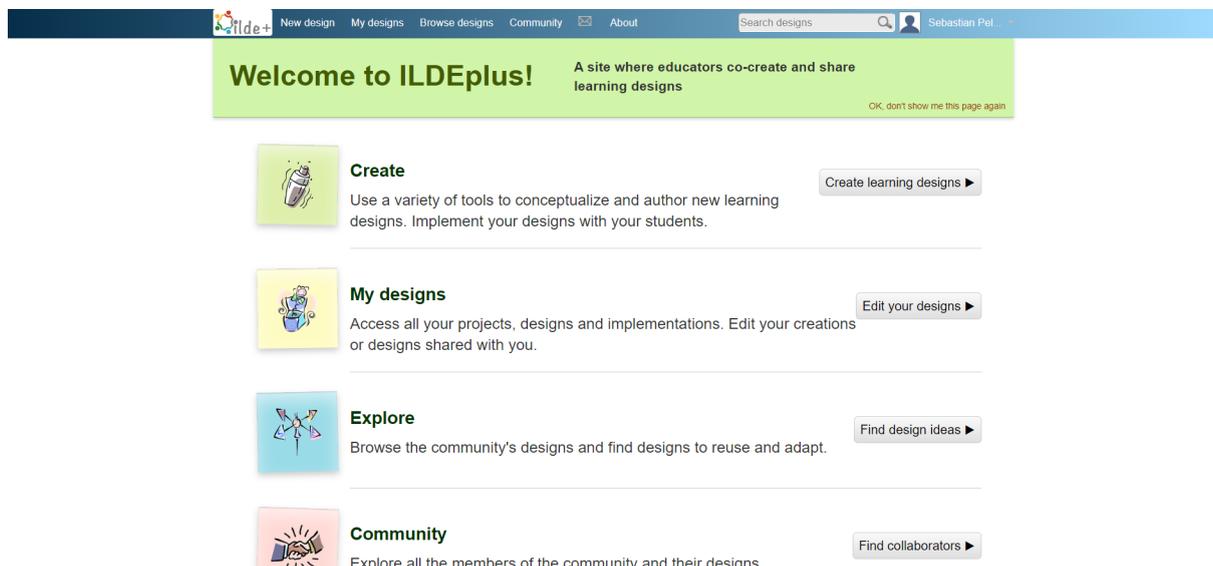


Figura 1.
Captura de pantalla de la pantalla principal de ILDE.

El entorno respalda la cooperación dentro de una comunidad de "diseño de aprendizaje" en la que sus miembros comparten y co-crean múltiples tipos de diseños de aprendizaje que cubren el ciclo de vida completo de un curso:

⁴ <https://ilde.upf.edu/>

1. Conceptualización: Implica trabajar con puntos de partida para los diseños. Este enfoque de diseño de aprendizaje puede ayudar a comprender mejor el contexto y los usuarios previstos, o a lidiar con abstracciones y borradores relacionados con los elementos del diseño y sus interconexiones.
2. Creación: Producir una definición detallada de un diseño de aprendizaje que está listo para usar con los alumnos.
3. Implementación: Aplicar un diseño de aprendizaje creado con un grupo particular de estudiantes usando un entorno de aprendizaje virtual específico y un conjunto de herramientas.

Las funcionalidades del software disponible en la plataforma ILDE incluyen:

1. Creación de materiales educativos interactivos y multimedia, como juegos, simulaciones, mapas conceptuales, presentaciones, entre otros.
2. Herramientas para el diseño y desarrollo de cursos completos, donde se pueden combinar varios tipos de materiales educativos.
3. La capacidad de incorporar imágenes, videos, audio y otros tipos de archivos multimedia en los materiales educativos.
4. La posibilidad de agregar preguntas, cuestionarios y exámenes para evaluar el aprendizaje del usuario.
5. La opción de personalizar el diseño y la apariencia de los materiales educativos.
6. La capacidad de compartir y colaborar en el proceso de diseño y desarrollo de los materiales educativos con otros docentes y estudiantes.
7. La posibilidad de exportar los materiales educativos en diferentes formatos para su uso en diferentes plataformas y dispositivos.
8. Herramientas de seguimiento y análisis para evaluar el rendimiento y la efectividad de los materiales educativos.

En la Figura 2 podemos observar el área de trabajo de ILDE, la cual tiene la apariencia de un editor de texto. La plataforma ofrece diversas plantillas de diseño de cursos, que incluyen un texto base con espacios en blanco que pueden ser completados por el usuario. En la parte superior del área de trabajo se encuentran las opciones de guardar el trabajo realizado y de compartirlo con otros usuarios de la plataforma. En la parte inferior del área de trabajo se encuentra la opción de agregar documentos y archivos adjuntos a los materiales educativos.

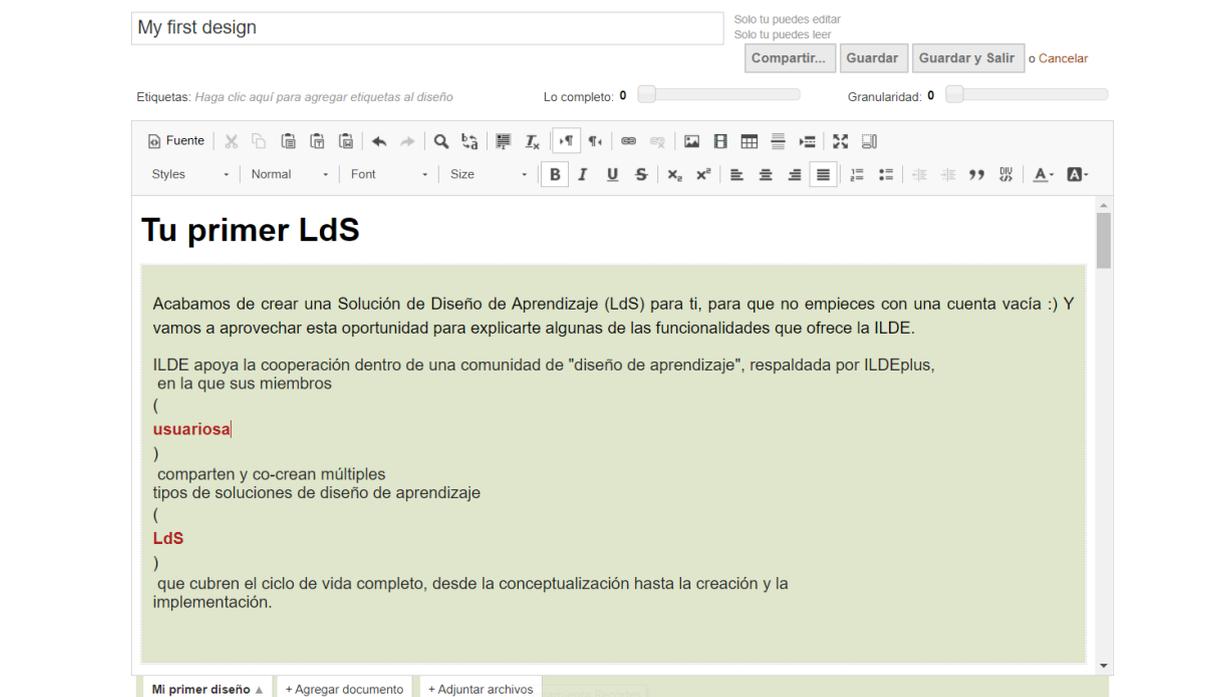


Figura 2.
Captura de pantalla del área de trabajo de ILDE.

El artículo Supporting awareness in communities of learning design practice (Michos y Hernández, 2018) se enfoca en la dimensión social del proceso de diseño del aprendizaje, que aún está poco explorada. El diseño de unidades de aprendizaje a menudo se considera una tarea individual del docente, pero no se ha investigado cómo los docentes trabajan e interactúan en pequeños grupos o dentro de comunidades educativas más grandes para informar mejor sus tareas de diseño. En este contexto, el artículo describe un estudio sobre el uso de una plataforma de diseño de aprendizaje social llamada ILDE. La investigación se centró en cuatro comunidades educativas diferentes: dos escuelas secundarias, un programa de maestría para futuros maestros y un curso en línea masivo abierto (MOOC) para profesores.

Los resultados muestran que la plataforma ILDE es útil para resumir la actividad de la comunidad y para identificar el contenido y las funciones de los miembros. Además, el uso de la plataforma aumentó las interacciones de los participantes, como las vistas de perfil, y los maestros mostraron su disposición a aprovechar las contribuciones de otros

En nuestro proyecto se ha tomado como referencia a ILDE, ya que compartimos la funcionalidad principal de poder diseñar cursos de manera colaborativa con otros docentes. Además, se comparten otras funcionalidades tales como la posibilidad de exportar diseños, adjuntar documentos o archivos al diseño, registrar un curso bajo licenciamiento Creative Commons (CC)⁵, guardar cursos y exportarlos.

⁵ <https://creativecommons.org/>

2.2.2. Edu-ÁREA

Edu-ÁREA (Caeiro-Rodríguez et al., 2015) es una plataforma educativa que tiene como objetivo principal fomentar la innovación en la enseñanza, principalmente mediante el uso de las TIC, nuevas pedagogías y recursos educativos abiertos. Se plantea una especificación de modelo de datos para Planes de Lección Abiertos (OLP) en el contexto de esta plataforma. El modelo de datos OLP puede desempeñar un papel clave en la importación y exportación de recursos educativos de los docentes en la plataforma (Caeiro y Mendoza, 2014).

La plataforma proporciona a los usuarios funcionalidades típicas de almacenamiento y compartición de Open Educational Resource (OER). Además incluye funcionalidades de edición en línea para algunos tipos de recursos educativos. En Edu-AREA, los OER se agrupan y gestionan de acuerdo a dos categorías: Recursos, sobre lo que se puede usar, y Planes, sobre lo que se puede hacer.

El énfasis principal de la plataforma radica en las características relacionadas con el soporte de los principios de apertura relacionados con los OER. Esto implica apoyar la creación, modificación y reutilización, la gestión de licencias, proporcionar atribución y reconocimiento.

La plataforma proporciona plantillas para apoyar la disposición de los diseños de cursos y las actividades. Cada plantilla tiene una disposición específica de campos de texto, áreas de texto, listas de elementos, etc. Todos estos campos se pueden editar en línea. Además, los planes de lección y las actividades pueden incluir otros recursos como documentos.

Los usuarios pueden proporcionar observaciones, evidencias y reflexiones sobre el uso de los OER en una experiencia educativa real, lo que puede ser muy valioso para otros usuarios.

Los planes de lección y actividades también pueden estar licenciados bajo los términos de Creative Commons (CC).

En la Figura 3 se puede observar la pantalla principal de la plataforma.

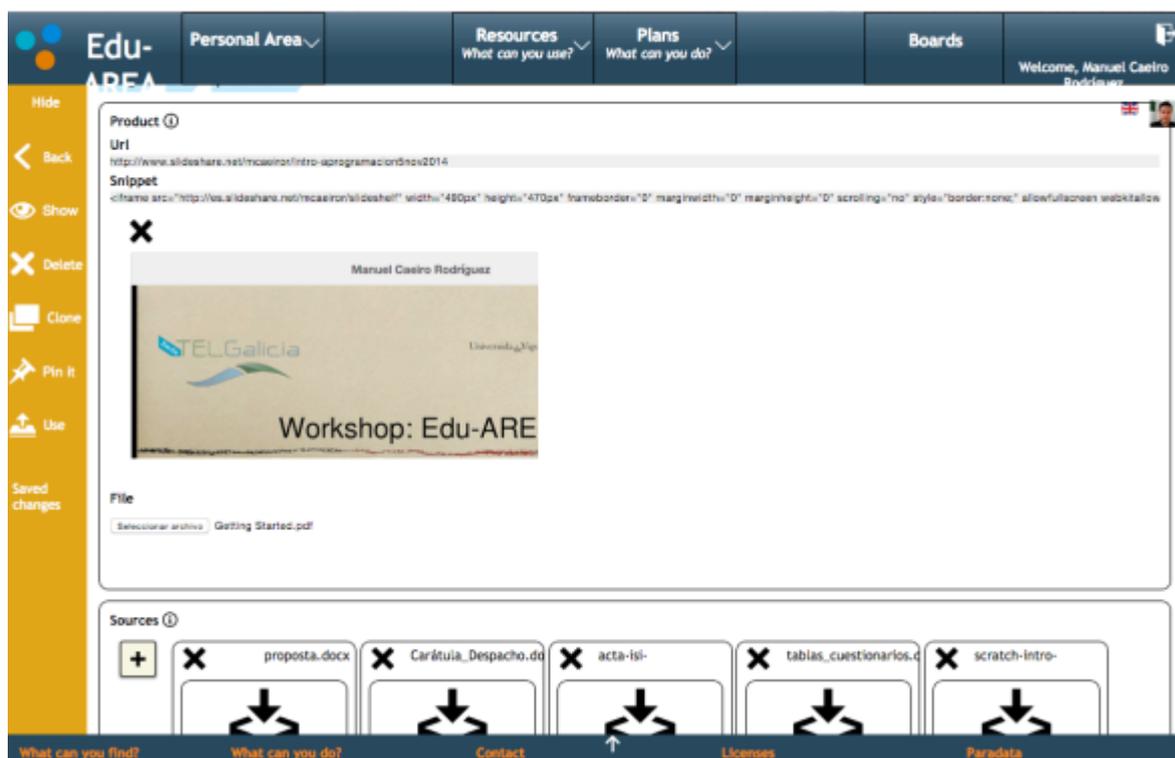


Figura 3.
Captura de pantalla de cómo se ve la plataforma Edu-Área.

Edu-Área ha sido una referencia valiosa para nuestro proyecto debido a varias características interesantes que compartimos. En primer lugar, su capacidad de gestionar recursos educativos y documentos de manera efectiva. También es muy interesante el hecho de que ofrezca licencias bajo la Licencia Creative Commons (CC) para compartir materiales sin infringir derechos de autor. Además, su sistema de plantillas y los campos de texto con disposición específica según una ubicación. Por último, su enfoque en la colaboración, permitiendo a los usuarios comentar y reflexionar sobre el uso de los recursos, es una característica que valoramos enormemente.

2.3 Conclusiones

En el ámbito del diseño de cursos, se observa una falta de claridad en la diferenciación y el tratamiento específico del uso de herramientas tecnológicas entre el diseño curricular y el diseño instruccional. Al realizar una revisión superficial de las herramientas disponibles, se constató que estas se centran principalmente en el diseño instruccional. Para abordar esta situación, se consultó a dos especialistas en educación con el fin de enfocar mejor el proyecto. Los resultados de estas consultas se presentan en el Capítulo 4 del diagnóstico y análisis.

3. Metodología de trabajo

El co-diseño es un enfoque de diseño colaborativo en el cual los desarrolladores trabajan de manera conjunta con expertos del dominio para identificar problemas y diseñar soluciones. En este proyecto específico, el co-diseño implica trabajar junto con la Socióloga, Máster en Educación Mariana Porta, actualmente doctoranda del Programa de Informática en Educación. También implica la creación de prototipos o maquetas que permita al usuario experto en el dominio entender cómo funcionará la herramienta antes de que se desarrolle completamente.

Para identificar los requerimientos del sistema se realizan reuniones de forma de establecer los pasos necesarios a cumplir en el proceso de diseño, analizar la situación actual, desarrollar un diseño conceptual para una solución tecnológica y crear las interfaces de usuario necesarias dentro del sistema.

Se han establecido dos etapas de trabajo. En la primera etapa se realiza el diagnóstico y análisis para determinar el alcance del problema a resolver e identificar los requerimientos. En la segunda etapa se diseña e implementa una solución, validando en forma continua con el experto en el dominio.

Como punto de partida para el diagnóstico y análisis, se toma como referencia el trabajo previo de Mariana Porta (Porta et al., 2021), que en su investigación sobre la Etapa 3 de RUD se enfoca en el rediseño de sistemas o procesos diseñada en los siguientes cinco pasos:

1. Definir modalidad.
2. Definir objetivos y contenidos.
3. Definir evaluación.
4. Definir estructura.
5. Diseñar en la plataforma.

Cada paso del rediseño es presentado en este modelo con 3 etapas: Pregunta central, concebir el diseño y desarrollar e implementar. La Figura 4, describe los 5 pasos del rediseño.



Figura 4.

Proceso de rediseño en 5 pasos - Recurso elaborado por Mag. Mariana Porta. Tomado de (Porta et al., 2022)

Partiendo de la propuesta del rediseño de un curso en 5 pasos, se trabaja en la etapa de Diagnóstico y Análisis del problema de planificación y/o rediseño de cursos, y posteriormente en la etapa de Diseño e Implementación.

3.1 Forma de trabajo en etapa de Diagnóstico y Análisis

La primera etapa de Diagnóstico se realizó a través de reuniones semanales de entre hora y media y dos horas de duración entre el equipo del proyecto, la usuaria experta y la tutora. Estas reuniones se realizaron por la plataforma Zoom debido a las diferentes localidades geográficas en que se encontraban los participantes. Las reuniones se extendieron con el objetivo del diagnóstico durante los primeros dos meses del proyecto. Además del diagnóstico (identificación del foco del problema) se trabajó también en el análisis, que es la identificación de los requisitos ordenados por prioridades para su implementación. Para realizar este análisis de requisitos se siguió la metodología del curso de Ingeniería de Software de la Facultad de Ingeniería⁶ basado en el Capítulo 4 del Libro Sommerville (Sommerville, 2004). La estructuración de los requisitos se realizó siguiendo el formato de La oficina de proyectos de informática⁷.

La interpretación de los requisitos, se registró en documentos de Google Docs con versiones que se mantenían actualizadas durante el transcurso del proyecto y así crear trazabilidad de las necesidades para el proyecto. Los documentos sirven como guía al momento de realizar avances, tener un histórico de los requerimientos solicitados y posteriormente consultarlos al momento de desarrollar la herramienta.

⁶ <https://eva.fing.edu.uy/course/view.php?id=613§ion=4#tabs-tree-start>

⁷

<https://sites.google.com/site/oficinaproyectosinformatica/home/archivo/PMOInformatica%20Documentos%20de%20requerimientos%20de%20software%20plantilla.doc?attredirects=0&d=1>

Para el seguimiento de esta etapa se crea un documento de conceptos para el diseño que se puede ver en detalle en el [Anexo B](#). Es completado por el usuario experto en el dominio y comentado por todo el equipo en las reuniones. Este documento recoge los atributos que deben ser solicitados al docente en el proceso de diseño. Como resultado se identifica la colección de atributos y la forma de agruparlos.

En base a los documentos y las reuniones de discusión, se establece un punto a partir del cual existe convergencia entre lo deseado por el usuario experto en el dominio y lo planteado por el equipo de desarrollo. Se acuerda con el usuario experto en el dominio un conjunto mínimo de requerimientos necesarios que hacen que el software pueda ser funcional y aporte valor a los docentes. Se establecen órdenes de prioridad entre los requerimientos, utilizando como criterio aquellos que aportan valor pero no son esenciales para la tarea del docente, los cuales se consideran de alta prioridad y se pasa a la etapa de Diseño e Implementación.

En el documento de conceptos para el diseño también se registran los atributos del sistema que son críticos y requieren de un tratamiento especial. Estos atributos serán definidos como puntos críticos. Los puntos críticos son momentos en que la herramienta tiene que brindar la información adecuada al docente para una mejor definición de los resultados. Aquí se le pueden sugerir herramientas externas al sistema. El administrador tiene que poder definir los puntos críticos para brindar un aporte conceptual, una información ampliatoria y/o una normativa al docente.

3.2 Forma de trabajo en etapa de Diseño e Implementación

Una vez que se han identificado los requerimientos, se procede a la presentación de soluciones iniciales, etapa de Diseño. Esto incluye una maqueta, que es una representación visual del proyecto que permite al usuario experto en el dominio tener una idea clara y concreta de cómo sería el producto final.

La plataforma Moqups⁸ fue la elegida para maquetar la aplicación y contar con una visión más realista al producto final con la cual esclarecer, quitar o agregar requisitos a lo interpretado. Se usó la plataforma online Moqups ya que cuenta con una versión gratuita, que permite representar la interacción entre pantallas. Además, se trata de una plataforma colaborativa en tiempo real. Estas maquetas se separan en los identificados como principales componentes del software a desarrollar.

Durante el proceso de desarrollo del software se le solicita al usuario experto en el dominio parámetros del sistema. Estos parámetros corresponden a la información general del software, como por ejemplo: los textos de ayuda, las descripciones de los puntos críticos, los datos fijos como carreras, herramientas, etc y demás datos que dependen de la institución. Para esto se le envía al usuario experto en el dominio una planilla con el parámetro requerido y un espacio para ser completado.

Durante la etapa de implementación se aplica la metodología ágil de desarrollo de software. El desarrollo ágil de software se refiere a métodos de ingeniería del software basados en el

⁸ <https://moqups.com/es/>

desarrollo iterativo e incremental donde los requisitos y soluciones evolucionan con el tiempo según la necesidad del proyecto. El trabajo es realizado mediante la colaboración de equipos auto-organizados y multidisciplinarios, inmersos en un proceso de toma de decisiones a corto plazo compartido. Existen muchos métodos de desarrollo ágil; la mayoría minimiza riesgos desarrollando software en lapsos cortos. El software desarrollado en una unidad de tiempo es llamado una “iteración”, la cual debe durar de una a cuatro semanas.

Cada iteración del ciclo de vida incluye: planificación, análisis de requisitos, diseño, codificación, revisión y documentación. Una iteración no debe agregar demasiada funcionalidad para justificar el lanzamiento del producto al mercado, sino que la meta es tener una versión de demostración al final de cada iteración. Al final de cada iteración el equipo vuelve a evaluar las prioridades del proyecto.

Los métodos ágiles enfatizan las comunicaciones cara a cara en vez de la documentación y que el software funcional es la primera medida del progreso.

Se tienen en cuenta 4 valores:

1. Las personas y las interacciones entre ellas por sobre los procesos y las herramientas.
2. El software funcionando por sobre la documentación detallada.
3. La colaboración con el cliente por sobre la negociación de contratos.
4. La respuesta a los cambios por sobre el seguimiento estricto de los planes.

Para esto utilizamos Scrum⁹, que en lugar de proporcionar descripciones detalladas y completas de cómo se llevará a cabo el proyecto, el equipo de desarrollo toma sus propias decisiones, ya que son ellos los que mejor conocen cómo resolver el problema. Por lo tanto, en el desarrollo con Scrum, en cada reunión de planificación o sprint se describen los objetivos y características a desarrollar en el próximo sprint, en lugar de presentar los criterios de entrada y salida como en otras metodologías.

Esto permite al equipo de desarrollo tener una mayor libertad para decidir la mejor manera de abordar el proyecto y resolver los desafíos que surjan, y fomenta la creatividad y la innovación. Sin embargo, también requiere una mayor colaboración y comunicación entre los miembros del equipo para asegurar el éxito del proyecto. En nuestro caso, utilizamos los mecanismos de comunicación que incluyen las Daily Scrums los martes y viernes. Estas reuniones tienen una duración de no más de 15 minutos y en ellas cada miembro del equipo de trabajo informa brevemente sobre lo que hizo en los días anteriores, lo que planea hacer en los próximos días y los obstáculos que ha enfrentado.

Siguiendo la metodología Scrum el periodo para hacer los sprints es de dos semanas, y se realiza una reunión los días miércoles en la que se analiza el progreso de la tarea.

Cada desarrollador toma una tarea y la prueba. Una vez finalizada la tarea queda lista para que otro desarrollador la vuelva a probar y sea validada.

El control de las tareas y versiones se realiza mediante GitLab, permitiendo así registrar las actividades y seguir el desarrollo de las mismas mediante comentarios. Así, cada semana

⁹ <https://www.scrum.org/>

se registran las tareas que se realizan con los respectivos comentarios de los progresos, finalizando por último con el incremento que se sube en un sistema de almacenamiento web (GitLab).

Gracias al uso de esta herramienta se puede seguir el proceso de desarrollo. Para esto se definen en GitLab dos ramas de trabajo: desarrollo y consolidado. En la rama de desarrollo se almacena el trabajo diario y cuando una tarea queda finalizada y probada se sube a la rama de consolidado. Esto permite la solución de cualquier error con mayor facilidad y también realizar el Product Backlog llevando con éxito el registro de las tareas realizadas a lo largo del proyecto.

En el [Anexo C](#) se desarrolla la distribución de las tareas y los sprints. Además, se establecen los objetivos y plazos de cada sprint. El proceso de desarrollo de software implica la validación del producto por parte de un usuario experto en el dominio. Este usuario será responsable de hacer pruebas y validar lo desarrollado hasta el momento. Para esto, se establecen tres momentos en que el equipo de desarrollo enviará una versión de prueba al usuario experto, al finalizar el desarrollo de software de la etapa que se esté implementando: al finalizar el sprint 2, 5 y 7. Estos plazos intermedios serán para validar progresivamente que el software cumpla con las expectativas y objetivos planteados. De esta manera, el equipo de desarrollo recibirá retroalimentación temprana y podrá ajustar el proyecto según sea necesario, garantizando que el software final cumpla con las necesidades del usuario experto en el dominio.

4. Resultados del Diagnóstico y Análisis

La etapa de diagnóstico y análisis en la metodología de co-diseño es la etapa donde se identifican los problemas (se hace el diagnóstico) y se definen los requerimientos y necesidades del proyecto a través de procesos de análisis, tormentas de ideas y reflexión conjunta. En particular, el análisis de los requerimientos implica la identificación y documentación detallada de las necesidades y expectativas del usuario experto en el dominio.

Durante la etapa de diagnóstico se obtuvo como principal resultado la identificación de que la principal funcionalidad del software es la de brindar un entorno de trabajo para los docentes donde se permite registrar el proceso de diseño de un curso, obteniendo sugerencias en puntos críticos, definidos como aquellas características del curso que aún no son ampliamente manejadas por los docentes. Además, debe permitir compartir con colegas el diseño del curso sin perder el control único sobre el trabajo. Se identifica a la propuesta del curso en principio como una construcción propia de un docente, sin embargo del diagnóstico de diferentes escenarios de trabajo queda abierta la posibilidad de ser también un trabajo colaborativo entre docentes. El sistema deberá gestionar el versionado del diseño de curso, registrando los autores y los cambios realizados por ellos.

Otro resultado importante logrado en esta etapa fue que el diseño de cinco pasos planteado por el usuario experto en el dominio se modificó hacia un proceso con lógica de software. En otras palabras, se reorganizó el proceso de cinco pasos originalmente propuesto por el usuario experto en el dominio, con el fin de transformarlo en un proceso que pueda ser ejecutado por un software.

En la búsqueda por comprender la lógica del proceso de diseño, es necesario identificar las interacciones y dependencias entre los distintos actores que intervienen en el proceso. Como se muestra en la Figura 5, los objetivos a alcanzar a través de la propuesta del proyecto educativo pueden verse afectados por la modalidad seleccionada para su contextualización. Por ejemplo, si el objetivo es mejorar el contacto físico entre los estudiantes, una modalidad online no sería adecuada. Por otro lado, lograr los objetivos con una modalidad y metodología específicas sólo será viable si se cuenta con un conjunto de recursos determinado, como un equipo docente adecuado, infraestructura edilicia y/o tecnológica, y recursos educativos, entre otros. Finalmente, de acuerdo con el diseño curricular, cada curso debe ser evaluado aplicando criterios y utilizando ciertas métricas, las cuales variarán según la modalidad, metodología y recursos asignados a cada curso. Este conocimiento es fundamental para poder observar de manera integrada los elementos utilizados en el diseño de un curso, ya que permite identificar posibles problemas antes de su contextualización.

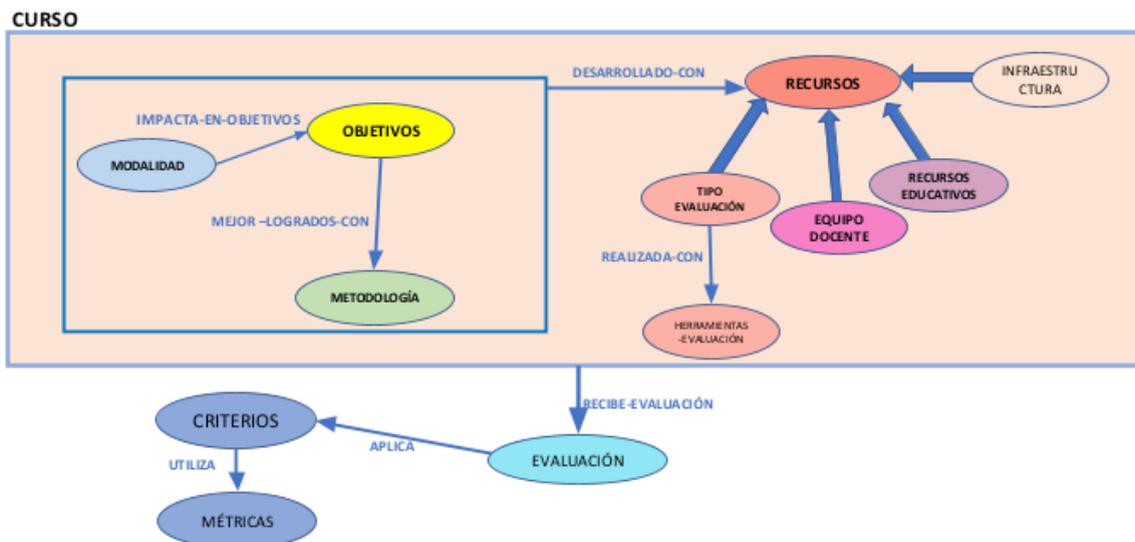


Figura 5.
Diagrama de interacciones en el proceso de cinco pasos.

Continuando con el análisis identificamos tres fases por las que pasa el docente en el proceso de creación de un curso:

1. Diseño de curso (pre-dictado): Se propone el curso.
2. Contextualización o producción de curso (pre-dictado): El curso es aprobado y ya hay estudiantes inscriptos. El docente tiene valores específicos sobre el curso que pueden determinar variaciones en el diseño.
3. Edición o dictado del curso: En este nivel hay estudiantes realizando el curso, el curso está activo.

De estas fases surgen interacciones entre ellas que son representadas en la Figura 6. El diagrama de la Figura 6 se interpreta de la siguiente manera: a partir de los datos generados en el diseño del curso, donde se redacta el Programa del Curso (nivel de Diseño de Curso en la Figura 6), se pueden crear varias instancias del mismo (nivel de Producción del Curso en la Figura 6). Por ejemplo, para el mismo programa de curso podría haber una instancia de producción en línea y otra presencial. Finalmente, en el nivel de Ediciones del Curso, se encuentran distintas ediciones para cada instancia de producción. Por ejemplo, para la instancia de producción presencial podría haber una edición matutina y otra vespertina que presenten diferencias entre ellas. En otras palabras, cada edición del curso depende de cada instancia específica.

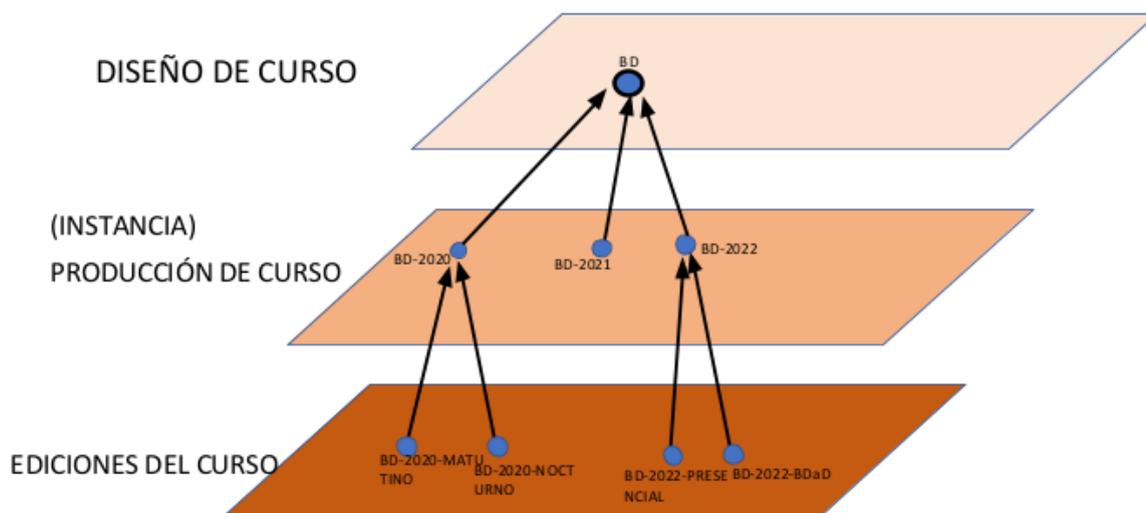


Figura 6
Fases del proceso de diseño de un curso.

Hay algunos datos que, una vez establecidos por el docente, no pueden ser modificados durante el curso, mientras que otros datos pueden ser editados en cualquier momento. Es importante identificar cuando la capacidad de modificar estos datos tendrá un impacto en el diseño final.

A partir de este análisis, sugerimos un diseño del ciclo de vida de un curso compuesto por tres etapas. En acuerdo con el usuario experto en el dominio, cambiamos el nombre de las etapas del diagrama de la Figura 6 de la siguiente manera: Diseño de Curso por Elaboración del Programa, Producción de Curso por Contextualización y Ediciones del curso por Dictado.

Entonces, las 3 etapas del ciclo de un curso se definen de la siguiente manera:

- **Elaboración del programa:** La elaboración del Programa educativo es el proceso sistemático, toma de decisiones acerca de los objetivos, contenidos, metodología de enseñanza y formas de evaluación del curso. El documento al cual generalmente se le llama "programa" es el resultado de dicho proceso, en formato escrito. Generalmente, ese documento debe ser presentado y evaluado para formar parte de una carrera de grado, posgrado, propuesta de educación permanente, etc.
- **Contextualización:** En esta etapa, se integran datos relativos al contexto en el cual el curso se desarrollará: por ej. número de docentes, número de estudiantes inscriptos, disponibilidad de equipamiento, disponibilidad de aulas, calendario académico, entre otros. A partir de esta información relativa a las condiciones de cursada, se tomarán decisiones sobre el desarrollo del curso, por ejemplo, cómo se ajustan los contenidos planificados al tiempo disponible o cómo se articulará el uso de los diversos entornos presenciales y digitales para asegurar la unidad y fluidez de la propuesta.

Recortado: En esta etapa, se integran datos relativos al contexto en el cual el curso se desarrollará. A partir de esta información relativa a las condiciones de cursada, se tomarán decisiones sobre el desarrollo del curso.

- **Dictado:** Una vez el curso se está implementando, se tomarán algunas decisiones acerca del diseño, sobre la marcha, basadas en el desarrollo del curso y en la respuesta de los estudiantes. También puede haber cambios contextuales que implican modificaciones de las condiciones de cursada, por ej. ausencias prolongadas inesperadas de docentes, medidas sanitarias de emergencia o cambios en la infraestructura. Estas decisiones requieren ajustes de tiempos, temas, recursos educativos, modalidad de cursada, etc. Registrar los cambios hechos y toda información que se determine relevante permite rastrear, explicar y comprender el motivo del cambio realizado.

Esta etapa se desarrolla desde el inicio del curso hasta el final de su dictado. En esta etapa se agregan nuevos datos y se pueden modificar algunos atributos de la etapa anterior de acuerdo a decisiones del docente. Además, en esta etapa es posible registrar los resultados de las evaluaciones del curso por parte de los docentes a partir de una rúbrica definida.

El sistema deberá permitir interactuar (consultar, modificar y crear copias) con el ciclo de vida de un curso.

4.1. Resultados del Análisis

El principal resultado logrado en la etapa de Análisis es la identificación de los requisitos ordenados por prioridades para su implementación. La especificación completa de todos los requisitos se presenta en el [Anexo D](#).

Como requisito no funcional se identifica que la herramienta debe ser de acceso público, es decir, cualquiera podrá descargar y usar el software en su computadora.

El sistema debe permitir compartir un curso con otros docentes en cualquiera de sus 3 etapas utilizando las licencias Creative Commons (CC) de derechos de autor. Además del curso, se debe compartir el historial de cambios así como también los autores de los mismos.

El sistema debe permitir interactuar con el ciclo de vida de un curso sin necesidad de contar con conexión a internet.

Una vez finalizada la etapa de recolección de requerimientos, se acordó con el usuario experto en el dominio un conjunto de ellos que serían esenciales y de prioridad alta, ya que son cruciales para el funcionamiento básico del sistema.

Los requerimientos de alta prioridad definidos son los que aseguran que el software cumpla con las siguientes características: funcionar sin conexión a internet (offline), ser instalable en dispositivos, que la interfaz sea flexible desacoplada del software y permitir la colaboración entre usuarios. Estos requerimientos son fundamentales para garantizar que el software cumpla con las necesidades y objetivos del usuario experto en el dominio.

El compromiso es de lograr la mayor cantidad posible de requerimientos, pero siempre teniendo en cuenta las limitaciones de tiempo y recursos, sin exceder el plazo del proyecto. Se hace un análisis detallado de los requerimientos para determinar cuáles son esenciales para el funcionamiento básico del sistema y cuáles son opcionales.

5. Resultados del Diseño e Implementación

5.1 Maquetación para el diseño de una solución

La maqueta es definida luego de un proceso de varias reuniones con el usuario experto en el dominio y escuchar sus requerimientos. En este proceso se vuelve a trabajar sobre el documento de conceptos para el diseño del [Anexo B](#). Como resultado del trabajo sobre la maqueta se identifica que la interfaz de usuario debe ser simple de usar para el docente, guiando el proceso de diseño del curso, sin tener necesidad de estudiar un manual de usuario para su uso. Debe ser flexible para su edición, es decir, en cualquier momento se puede acceder a cualquier opción para completar sus datos.

A continuación se presentan los resultados de diseño que se trabajaron a través de la maquetación.

5.1.1 Pantalla de Inicio

La pantalla de inicio es la primera interfaz que el usuario visualiza al abrir la aplicación. Es una de las pantallas más importantes ya que es la que da la primera impresión al usuario y su diseño y contenido son fundamentales para guiar al usuario en el uso de la aplicación. En la pantalla de inicio, se suele presentar información básica sobre la aplicación y sus funciones, un botón o enlace para iniciar sesión o registrarse, y accesos rápidos a las funciones principales de la aplicación. Es importante que la pantalla de inicio sea intuitiva y fácil de usar, con un diseño sencillo y atractivo para que el usuario se sienta cómodo y pueda encontrar rápidamente lo que está buscando. Además, la pantalla de inicio se actualiza periódicamente para ofrecer contenido renovado y relevante para el usuario.

En el encabezado se tiene un título y la opción de iniciar sesión. Esta opción estará visible solo cuando el sistema esté conectado a un servidor de institución que proporcione los esquemas de sus cursos y sus usuarios. Mientras el sistema funcione sin estar conectado a los servidores de las instituciones, esta opción permanecerá oculta. En la Figura 7 se puede ver el diseño propuesto para la pantalla de inicio de sesión.

Iniciar sesión



The image shows a login form with a header area containing a placeholder for a profile picture. Below the header, there are three input fields: 'Repositorio de la institución' (with 'UDELAR' and a dropdown arrow), 'Usuario', and 'Contraseña'. A blue 'Iniciar' button is positioned below the fields, and a blue link 'Continuar Offline' is at the bottom. A callout box points to the dropdown arrow in the 'Repositorio de la institución' field, containing the text: 'Sugerencia donde encontrar la institución. Puede ser una lista de instituciones o la url a un repositorio, en este caso explicar donde acceder a la url.'

Figura 7
Maqueta - inicio de sesión.

En la sección de “Mis cursos” se listan todos los cursos que tiene el docente. Desde allí podrá acceder a la modificación del curso en cualquier momento.

El botón "Importar" permite al usuario subir un archivo en formato Json, que contiene los datos de un curso. Esto permite al usuario importar cursos que ya hayan sido diseñados anteriormente, en lugar de tener que crearlos desde cero. Una vez completada la importación, el curso se verá reflejado en la lista de "Mis cursos" donde podrá ser editado o eliminado si es necesario.

Los cursos se muestran al usuario en forma de lista de Cards como se puede ver en la Figura 8.

Mis cursos

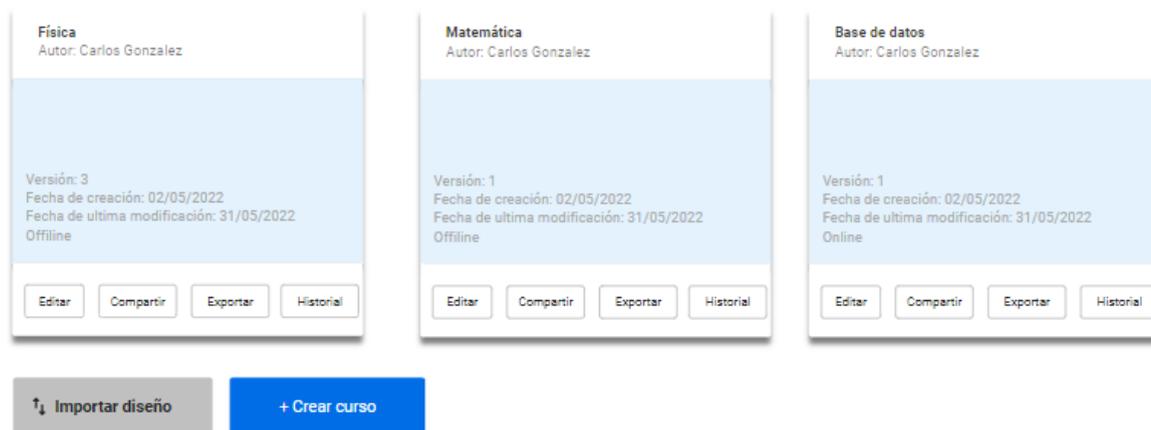


Figura 8. Maqueta - Pantalla de inicio

Es importante mencionar que en los ejemplos de las imágenes que se presentan a continuación, la etapa que se visualiza en color azul fue originalmente denominada "Implementación". Sin embargo, durante el proceso de desarrollo del software, el usuario experto en el dominio solicitó que se cambiara el nombre de esta etapa a "Contextualización". Esto se debe a que esta etapa se enfoca en la aplicación del conocimiento adquirido en un contexto específico y en la adaptación del mismo a las necesidades del usuario. Es importante mencionar este cambio para evitar confusiones y asegurar que el usuario comprenda correctamente el proceso y las etapas del software.

5.1.2 Resultados de la primer Maqueta

El primer diseño presentado al usuario experto en el dominio muestra las etapas del diseño claramente delimitadas. El formato utilizado es de un formulario donde el docente pasaría de pantalla en pantalla completando los campos solicitados.

Al ingresar a la plataforma, el docente verá una lista de todos sus cursos y al seleccionar uno para editar, será dirigido directamente a la pantalla de contenido del curso, que es el primer atributo disponible para editar. Como se puede observar en la Figura 9, en la parte inferior de la pantalla hay un botón "Siguiente" que lo lleva al próximo atributo a editar y así sucesivamente. Este diseño facilita la navegación y permite al docente completar la edición del curso de manera eficiente y organizada. Además, se busca una experiencia de usuario intuitiva y sencilla para que el docente pueda centrarse en la creación y edición del curso.

The screenshot shows a web interface for course design. At the top left, there is a header 'Contenido' with a graduation cap icon. At the top right, the course name 'Matemática' is displayed. The main content area is enclosed in a light blue border and contains several sections:

- Cantidad de créditos:** A text input field containing '10' and a summary box stating '12 créditos calculados'.
- Normativa:** A section titled 'Normativa' with a question '¿Cómo se cuentan los créditos de acuerdo a UDELAR?' and a link to a document: <https://dgjuridica.udelar.edu.uy/215-ordenanza-de-estudios-de-grado-y-otros->
- Temas y/o Competencias:** A table with three columns: 'Temas y/o Competencias', 'Descripcion', and 'Cantidad de horas'. It contains two rows of data:

Temas y/o Competencias	Descripcion	Cantidad de horas
tema: competencias... Nuevo tema	Contenido	15 horas
tema: competencias... Tema	Contenido	15 horas
- + Agregar más:** A blue button to add more topics.
- Total:** A summary line at the bottom right stating '= 30 horas (equivalente a 12 créditos)'.

Below the form, there is a blue button labeled 'SIGUIENTE'.

Figura 9. Primera maqueta.

La idea de presentar los datos al cliente en forma de formulario fue bien recibida, pero se descartó la posibilidad de que el flujo entre ellos sea lineal y secuencial, pasando de pantalla en pantalla mediante un botón “siguiente”. Se busca un flujo más flexible y adaptable a la realidad actual, en la que el proceso de completar los atributos del docente no es lineal y se hace a demanda.

5.1.3 Resultados de la segunda Maqueta

En el diseño presentado al usuario experto en el dominio, se presenta una primera pantalla en formato de dashboard, donde el docente tiene una vista general de todas las etapas y los atributos necesarios para completar el curso. La pantalla cuenta con una sección en la parte superior que muestra el progreso en porcentaje del diseño del curso.

La vista general se presenta en formato de tarjetas seleccionables, cada una de las cuales contiene un resumen de los atributos incluidos en esa etapa. Al presentar las opciones en forma de tarjetas, se permite al docente acceder a cualquier etapa en cualquier orden. El diseño propuesto se puede ver en la Figura 10.



Continúa editando tu curso



Figura 10. Segunda maqueta.

Luego de seleccionar una tarjeta en el dashboard, el usuario es dirigido a una página donde se presenta un formulario con los campos necesarios para completar esa etapa del diseño. Al final del formulario, en el pie de la página, se encuentran todas las tarjetas mostradas en el dashboard, lo que permite al docente saltar a cualquier etapa del diseño en cualquier momento. Esto rompe la linealidad del proceso y permite una mayor flexibilidad en la navegación. En la Figura 11 se puede ver dicha sección del diseño.

Cantidad de créditos

10 **1**

12 créditos calculados **2**

Normativa

¿Cómo se cuentan los créditos de acuerdo a UDELAR?
<https://djudica.udelar.edu.uy/215-ordenanza-de-estudios-de-grado-y-otros/>

Tema	Contenido	Cantidad de horas
Tema	Contenido	15 horas
Tema	Contenido	15 horas

+ Agregar más

= 30 horas (equivale a 12 credits)

Continúa editando tu curso

- Información general**
Unidad curricular, Área de conocimiento, cupo, y Forma de aprobación
- Recursos**
Bibliografía, Recursos educativos y licencia de derecho de autor
- Objetivos**
Modalidad de enseñanza, objetivos de aprendizaje, marco pedagógico

Figura 11
Segunda maqueta - Contenido interno y navegación.

Este diseño presenta varias ventajas, como la posibilidad de que el docente tenga una vista general de su trabajo en el dashboard y la libertad de acceder a cualquier etapa en cualquier momento. Además, se resalta la flexibilidad del diseño al romper el formato tradicional de formularios.

Sin embargo, el usuario experto en el dominio consideró que la herramienta podría mejorar si se incluyera una forma de conectar las etapas entre sí para crear una mayor sensación de sinergia. También se mencionó que la sección con los porcentajes de avance en la parte superior podría ser mejorada ya que no es la información central de la herramienta y podría ser mejor presentada.

5.1.4 Resultados de la tercer Maqueta

En esta nueva maqueta se tienen en cuenta las sugerencias del usuario experto en el dominio. En la sección del dashboard, las etapas se presentan colapsadas y al hacer clic en una de ellas, se expanden los grupos de atributos, indicando cuáles han sido completados y cuáles aún no. De esta forma, se tiene una visión general del avance en todo momento.

Además, cómo se puede observar en la Figura 12, se han diseñado las etapas de manera diferenciada, utilizando colores llamativos y distintos para cada una. De esta manera, el usuario puede identificar y asociar mejor la etapa en la que se encuentra a través del color.



Figura 12
Tercera maqueta

Este diseño fue muy bien recibido por el usuario experto en el dominio, especialmente por el uso de colores. Sin embargo, consideró que no sería adecuado utilizar números para identificar las etapas, ya que se busca que el diseño del curso no sea lineal. Por esta razón, la visualización en formato lineal hacia abajo también no es la mejor opción, ya que se busca mostrar una relación circular, donde el diseño de un curso es de insumo para futuros diseños.

Se evalúan diversas alternativas para mostrar el menú de forma visualmente atractiva y que cumpla con las expectativas del usuario experto en el dominio. Durante la investigación, se encontró Sheila Project¹⁰ que sirve de inspiración. La interfaz resultó muy interesante y cumple con todos nuestros requerimientos. En Sheila Project es posible visualizar un proceso completo de los 6 pasos del menú, y al hacer clic en cada uno de ellos, se despliega una definición detallada y las acciones correspondientes, tal como se puede apreciar en la Figura 13.

¹⁰ <https://sheilaproject.eu/>



Figura 13. Proceso de seis pasos del proyecto Sheila.

Dentro de ese contexto, el usuario puede crear su propio Framework con las 6 etapas. Allí se presentan los atributos que debe ir completando y un panel a la derecha que describe el contenido esperado como se observa en la Figura 14.

1 Map Political Context

Action

Think about the political context of your institution. Are there any internal or external drivers that surface a need for learning analytics?

Drag & Drop statements from the right.

Challenges

Policy

METHODOLOGY

Consider contextual elements (e.g., institutional size, structure) to identify opportunities for learning analytics.

Select Key Stakeholders

Identify opportunities to build learning analytics upon existing projects or practice.

Select Key Stakeholders

PURPOSE

CUSTOM STATEMENTS

Overall Progress: 0%

Number of statements per dimension

Help Review Share

Figura 14. Proceso de seis pasos del proyecto Sheila.

Además, en la Figura 14 también se puede ver en la parte inferior un menú de progreso que permite acceder en cualquier momento a cualquiera de las 6 etapas con un porcentaje de avance.

Sheila fue una fuente de inspiración para el diseño de la interfaz del proyecto debido a su originalidad en presentar todo el proceso en ciclos y la facilidad de navegación entre las seis etapas. Además, se utilizó un color diferente para cada etapa, lo que ayuda a diferenciarlas visualmente.

5.1.5 Resultados de la maqueta final

Finalmente se llega a un diseño que conforma, inspirado en el diseño de la interfaz de Sheila Project, con un menú principal en forma de grafo cíclico, como se puede ver en la Figura 15 y se explica con más detalles en la siguiente sección.

Curso 2022

Editá tu curso

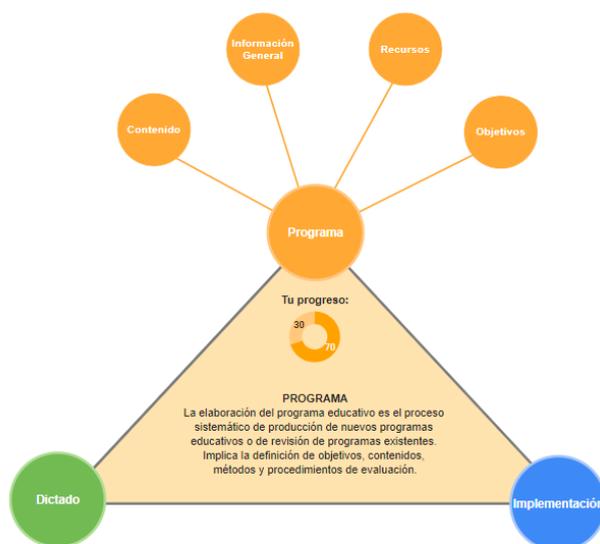


Figura 15. Menú - Maqueta final

Cada nodo del grafo colapsado representa una etapa, y todas las etapas están conectadas entre sí. Al expandir un nodo, se muestran los grupos de atributos relacionados a esa etapa. Al expandir estos grupos, se visualizan los atributos editables. Además, como un grupo de atributos puede estar presente en más de una etapa, al tratarse de un grafo, queda visualmente representado con una arista hacia cada etapa en la que se encuentra.

Cuando se selecciona un grupo de atributos, en una sección inferior se carga el formulario con los campos necesarios para completar ese grupo de atributos. Esta visualización permite que el docente tenga siempre visible el grafo con todo el diseño del curso y el formulario con los campos que está editando actualmente, brindando una mayor claridad y eficiencia en el proceso de creación del curso.

Dado que el sistema debe ser flexible y fácilmente editable por un administrador, es necesario definir un conjunto de atributos con el mismo comportamiento y estética. Los atributos de la interfaz no están fijos en una pantalla, es decir, el administrador tiene la

capacidad de ubicar un atributo en un lugar de la interfaz, dependiendo del esquema definido para el curso que se esté trabajando. Esto permite una mayor personalización y adaptabilidad del sistema a las necesidades de cada institución.

El administrador tiene la capacidad de definir cuántos atributos se mostrarán en cada línea o espacio de trabajo, lo que hace que la interfaz sea dinámica y que se pueda aprovechar mejor el espacio en pantalla. Esta tarea puede realizarse en tiempo real, agregando o quitando atributos y viendo su resultado en pantalla, permitiendo una mayor flexibilidad y adaptabilidad del sistema a las necesidades de cada institución.

A continuación se describe la estructura de las secciones definidas en la maqueta.

5.1.6 Panel de control y menú

Una vez seleccionado un curso para trabajar sobre él, se visualiza la pantalla de control y el menú para navegar sobre las etapas del diseño.

Para el diseño del menú se eligió un formato de grafo, esto permite navegar a cualquier opción en cualquier momento.

Se manejan distintos tamaños de los nodos según el nivel de detalle. Como se muestra en el grafo de ejemplo, las opciones de “Programa”, “Contextualización” y “Dictado” son las más grandes. Cuando se selecciona una opción, se abren sus sub-opciones en un tamaño más pequeño. Lo mismo sucede al elegir una de estas sub-opciones.

A cada etapa se le asigna un color. La paleta de colores manejada para los nodos son bien diferenciables. Se busca que el usuario pueda visualizar claramente sobre qué etapa del diseño está trabajando.

En la Figura 16 se observa que en el centro del menú se genera una sección con una breve descripción sobre la etapa y un porcentaje de avance en el diseño de esa etapa.

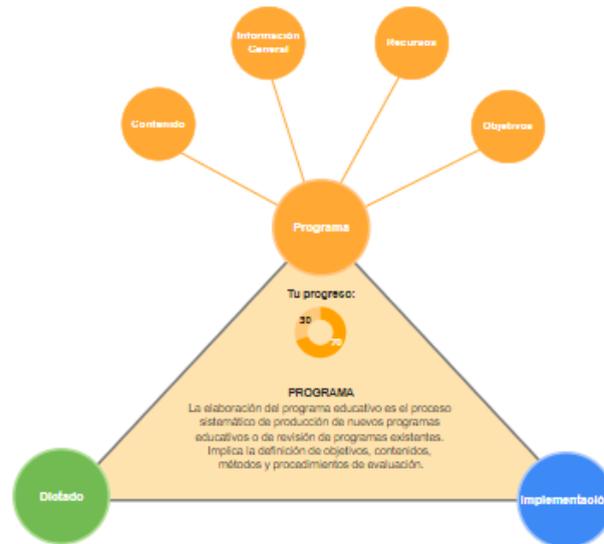


Figura 16. Maqueta - Menú de opciones.

En la siguiente imagen (Figura 17) se clasifican a las opciones del menú como opciones del primer nivel y sub-opciones del segundo nivel.

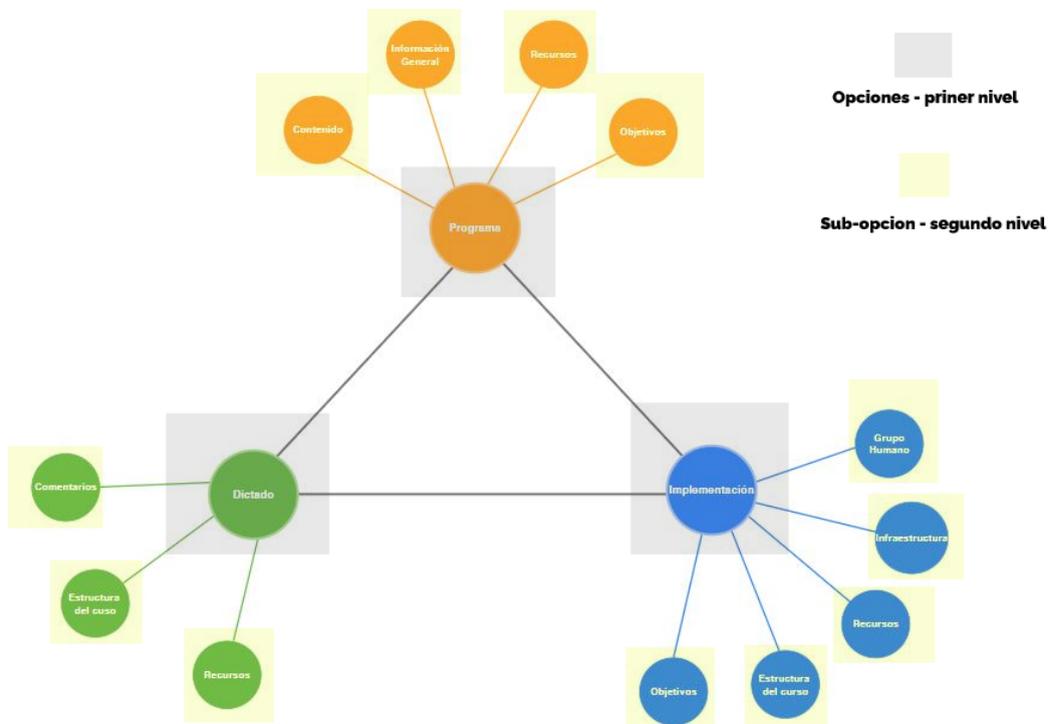


Figura 17
Maqueta - Menú con opciones y subopciones.

Cuando se selecciona una sub-opción del segundo nivel, ésta pasa al centro del menú cómo se observa en la Figura 18, para representar visualmente que se está trabajando sobre ella.

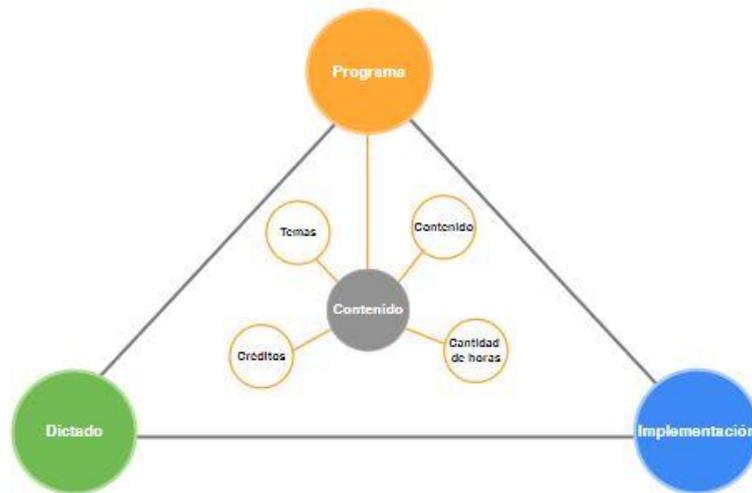


Figura 18
Maqueta - Menú con sub-opción seleccionada en el centro.

Las sub-opciones del segundo nivel se pueden relacionar con más de una opción del primer nivel. En tal caso, la sub-opción se mueve al centro del menú y además se relaciona mediante aristas con todas las opciones que tiene relación en el primer nivel. Además, éstas aristas se pintan del color que corresponde la opción del primer nivel.

Por ejemplo, en la siguiente imagen (Figura 19) se vé qué “Recursos” está relacionado con las etapas “Programa”, “Contextualización” y “Dictado”.

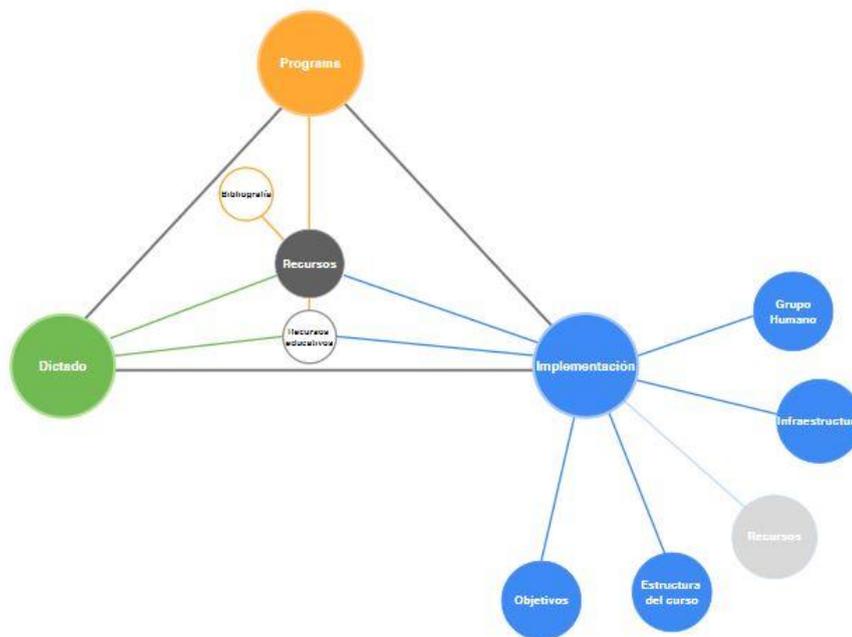


Figura 19. Maqueta - Menú con sub-opción que pertenece a varias etapas.

En un principio se manejó la posibilidad de usar un tercer nivel de nodos que representan los atributos de cada subsección. Luego fué descartado ya que se consideró que el desafío de acomodar el espacio para visualizar todos estos nodos no rendía suficientes beneficios. Otra consideración fue que el nivel mínimo de interacción del usuario con el menú era el segundo nivel, con eso ya se carga la sección del área de trabajo y allí ve los atributos que corresponden a la sub-opción seleccionada.

5.1.7 Área de trabajo

Corresponde a la sección de la interfaz de usuario donde el docente efectivamente va a poder agregar los datos correspondientes a cada atributo del curso.

Se parte del documento que surgió del relevamiento de requerimientos, donde se tiene la colección de atributos y grupos de atributos. En cada reunión con el usuario experto en el dominio se presentan incrementalmente los atributos de una nueva etapa del diseño. Allí se discute la forma de presentar los nuevos y se sugieren mejoras por parte de todo el equipo. En la siguiente reunión se presentan las mejoras aplicadas.

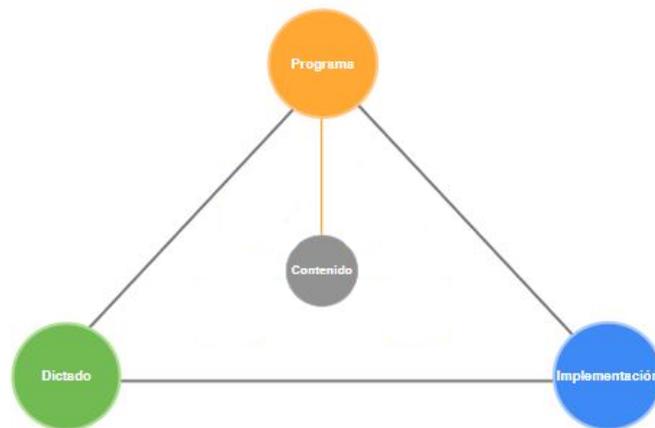
En la Figura 20 se vé el área vacía antes de que el usuario seleccione una sub-opción del menú para trabajar.



Figura 20
Maqueta - Área de trabajo vacía.

Luego de que el usuario selecciona una sub-opción del menú, en la sección del área de trabajo se cargan los atributos correspondientes. Allí se puede cargar la información del curso.

Por ejemplo, si el usuario selecciona la opción Programa y luego la subopción Contenido, se cargan los atributos de Contenido. Este ejemplo se vé representado en la Figura 21.



Contenido

Cantidad de créditos

12 créditos calculados

Normativa

¿Cómo se cuentan los créditos de acuerdo a UDELAR?

<https://dgjuridica.udelar.edu.uy/215-ordenanza-de-estudios-de-grado-y-otros/>

Temas y/o Competencias ?	Descripción ?	Cantidad de horas ?
<input style="width: 100%;" type="text" value="tema, competencias..."/> <input style="width: 100%;" type="text" value="Nuevo tema"/>	<input style="width: 100%;" type="text" value="Contenido"/>	<input style="width: 100%;" type="text" value="15 horas"/>
<input style="width: 100%;" type="text" value="tema, competencias..."/> <input style="width: 100%;" type="text" value="Tema"/>	<input style="width: 100%;" type="text" value="Contenido"/>	<input style="width: 100%;" type="text" value="15 horas"/>

[+ Agregar más](#)

= 30 horas (equivale a 12 creditos)

Figura 21
Maqueta - Área de trabajo con datos de la sub-opción Contenido.

5.1.8 Estándar para atributos

Se definen los tipos de atributos para estandarizar y que todos los de un mismo tipo comparten la visualización. Es importante definir un estándar con el mismo estilo para ciertos atributos, ya que al ser un software flexible, los atributos de la interfaz no son fijos y dependen del esquema definido para el curso que se esté trabajando.

A continuación, en la Tabla 1, se presentan los diferentes tipos de datos.

Tipo y descripción	Ejemplo de Visualización
Texto: cadena de hasta 99 caracteres.	<p>Nombre de la unidad curricular</p> <input data-bbox="831 320 1214 383" type="text" value="Nombre de la unidad curricular"/>
Textarea: cadena de más de 100 caracteres.	<p>Marco pedagógico</p> <div data-bbox="831 488 1369 611" style="border: 1px solid #ccc; padding: 5px;"> <p>En la siguiente etapa podrás definir la Metodología de enseñanza a partir del Marco pedagógico</p> </div>
Numérico	<p>Cantidad de créditos</p> <input data-bbox="831 723 1011 768" type="text" value="10"/>
Radio Button: múltiples opciones, solo una es aceptada.	<p>Forma de aprobación</p> <p>Permite exoneración <input checked="" data-bbox="1034 880 1054 909" type="radio"/> Sí <input data-bbox="1034 898 1054 927" type="radio"/> No</p>
Combobox fijo: lista de datos fijos del sistema, solo una opción es aceptada.	<p>Modalidad de enseñanza</p> <input data-bbox="831 1037 1190 1093" type="text" value="Seleccione..."/>
Combobox dinámico: lista de datos fijos o previamente ingresados por el usuario. Permite múltiples valores.	<p>Temas asociados</p> <div data-bbox="831 1223 1374 1267" style="border: 1px solid #ccc; padding: 2px;"> Tema 1 Tema 2 ▼ </div>
Adjuntar documentos: Permite subir archivos.	<p>Bibliografía ?</p> <div data-bbox="831 1384 1190 1451" style="border: 1px solid #ccc; padding: 2px;"> Libro 1  </div>
Descargar documentos	<p>Bibliografía ?</p> <div data-bbox="831 1547 1286 1653" style="border: 1px solid #ccc; padding: 2px;"> Libro 1   </div>

Tabla 1. Tipo de datos.

Los atributos deben tener una propiedad que sirva de ayuda o guía para el usuario. Es información adicional sobre el atributo, pero que puede resultar de importancia al momento de completar los datos.

Por ejemplo:

Atributo: Cantidad de créditos

Ayuda: Defina la cantidad de créditos si ya los conoce, de lo contrario pueden ser calculados a partir de la cantidad de horas de cada tema.

Por ejemplo, los Temas y Competencias son atributos de éste tipo. Para ellos se mostrará un botón al finalizar la última instancia indicando que se puede agregar una nueva cómo se puede ver en la Figura 22.

Temas y/o Competencias ?	Descripcion?	Cantidad de horas ?
tema, competencias... ▼	Nuevo tema	Contenido
tema, competencias... ▼	Tema	Contenido
		15 horas
		15 horas

[+ Agregar más](#)

Figura 22. Maqueta - Atributos multivaluados.

5.1.9 Puntos críticos

Los puntos críticos son momentos en que la herramienta tiene que brindar la información adicional al docente para un mejor resultado del proceso de diseño de curso. Pueden ser de los siguientes tipos:

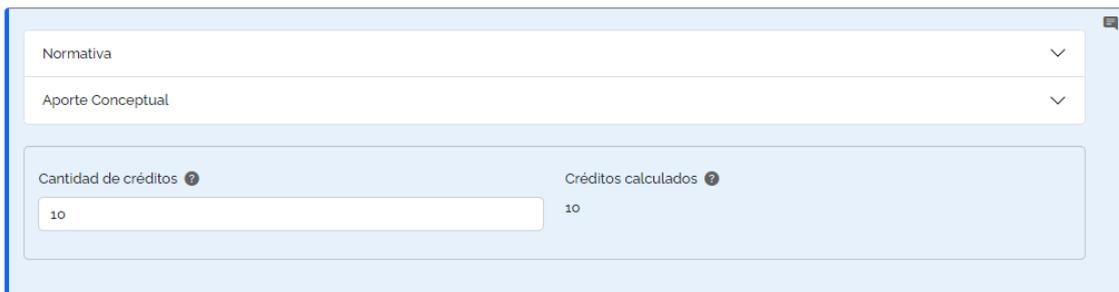
- **Aporte conceptual:** Un breve texto en el cual se explica un concepto de manera acotada y breve. Breve texto que presenta el material del enlace, seguido de un enlace a un sitio que amplía el concepto explicado.
- **Normativa:** una nota que recuerda normativas o requisitos institucionales que vienen al caso. Está seguido de un breve texto que dice el nombre del documento o referencia y el enlace al mismo.
- **Información ampliatoria:** Tema novedoso y dinámico.

Por lo tanto, es importante diferenciar en la interfaz de usuario qué atributos son puntos críticos de una manera que llame la atención al usuario. También se considera la posibilidad de que no sea una información siempre visible, ya que un usuario habitual del software quizá no necesite consultar ésta información todo el tiempo.

Es por eso que se definió para los atributos que son puntos críticos resaltar el color de fondo (#e6f1fc). La información que corresponde al punto crítico estará dentro de un acordeón que podrá ser expandido y colapsado en cualquier momento para no obstruir la visión del usuario.

A continuación podemos ver cómo se ve el punto crítico Cantidad de Créditos colapsado (Figura 23) y luego cómo se ve expandido (Figura 24).

Contenido



Normativa

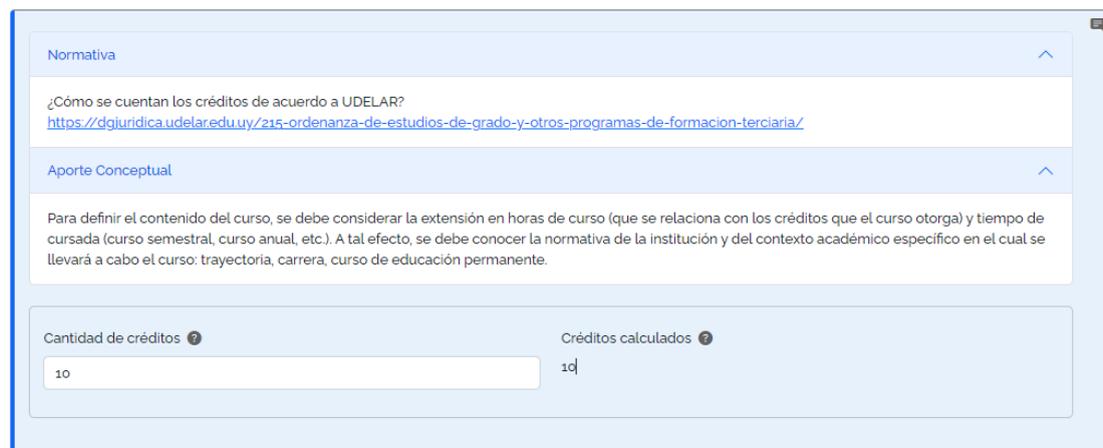
Aporte Conceptual

Cantidad de créditos 10

Créditos calculados 10

Figura 23. Punto crítico Cantidad de Créditos colapsado.

Contenido



Normativa

¿Cómo se cuentan los créditos de acuerdo a UDELAR?
<https://dgiuridica.udelar.edu.uy/215-ordenanza-de-estudios-de-grado-y-otros-programas-de-formacion-terciaria/>

Aporte Conceptual

Para definir el contenido del curso, se debe considerar la extensión en horas de curso (que se relaciona con los créditos que el curso otorga) y tiempo de cursada (curso semestral, curso anual, etc.). A tal efecto, se debe conocer la normativa de la institución y del contexto académico específico en el cual se llevará a cabo el curso: trayectoria, carrera, curso de educación permanente.

Cantidad de créditos 10

Créditos calculados 10

Figura 24. Punto crítico Cantidad de Créditos expandido.

5.2 Arquitectura para la implementación de la solución propuesta

En base a los requerimientos, se diseña una solución con 3 productos:

- Software cliente para Docente: Aplicación de escritorio para que los usuarios Docente puedan realizar el proceso de diseño de un curso. La forma en la que se visualiza este proceso estará definido en un archivo de texto JSON (esquema) que la aplicación interpreta.
- Software cliente para Administrador: Aplicación para que el usuario Administrador pueda modificar el proceso de diseño de un curso (esquema).
- Servidor central: Donde se soportan las funcionalidades en línea como ser la distribución de cambios en el esquema realizado por el usuario Administrador, las alertas que este cambio genera, la recepción de sugerencias realizadas por Docentes al Administrador y la analítica de datos de los cursos subidos al servidor.

En la Figura 25 se puede ver un diagrama de la arquitectura propuesta.

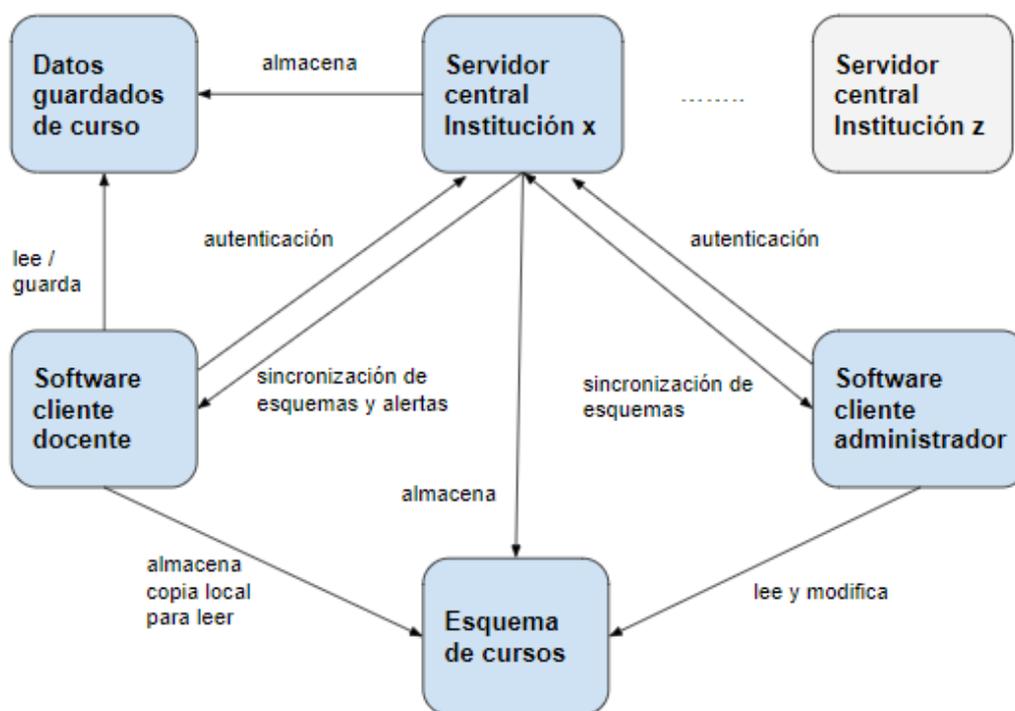


Figura 25
Arquitectura para la solución propuesta.

El software cliente para el docente es una aplicación de escritorio que funciona sin conexión (offline). Esto impacta en la arquitectura ya que se deben mantener los esquemas y los cursos almacenados de forma local en la computadora de cada cliente.

El almacenamiento de los datos se hace en tiempo real, es decir, cada vez que el software en el cliente detecta que se realizó una modificación en un atributo se dispara una secuencia de acciones que derivan en el procesamiento de la información para impactar el resultado en la base de datos del curso.

La Figura 26 representa el flujo de información para procesar los datos ingresados.

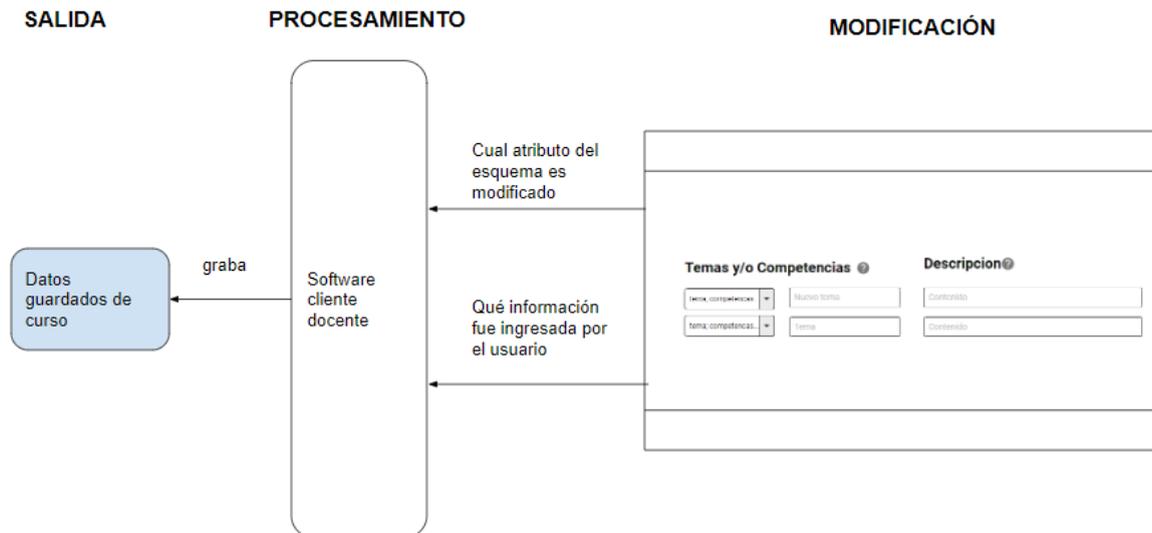


Figura 26. Modificación de los datos para almacenar en la estructura de datos.

5.3 Elección del Software de Desarrollo

La primera decisión de implementación a tomar es la de seleccionar la tecnología con la cual implementar los 3 productos. Para las 2 aplicaciones de escritorio se evaluaron 2 posibles soluciones, una de ellas es utilizar una tecnología con compilación nativa en aplicación de escritorio y la otra una tecnología de desarrollo web con compilación a aplicación de escritorio mediante librerías de código abierto.

La Tabla 2 presenta un estudio de características entre los lenguajes de programación Angular¹¹, React¹² y Java¹³.

	Angular	React	Java
Última versión estable	15.0.1	18.2.0	8 Update 351
Software libre	si	si	si
Tipo	Framework completo	Librería de Javascript	Software
Permite aplicaciones de escritorio	si (usando librerías)	si (usando librerías)	si
Arquitectura	Componentes aislados	Componentes aislados	Orientada a Objetos
Experiencia con el software	Bastante	Poco	Bastante

Tabla 2

Características de tecnologías nativas de escritorio y tecnologías web.

¹¹ <https://angular.io/>

¹² <https://es.reactjs.org/>

¹³ <https://www.java.com/es/>

Basado en el análisis, se ha elegido Angular como el lenguaje para desarrollar las aplicaciones de escritorio debido a varias razones. En primer lugar, el equipo tiene un alto nivel de familiaridad con Angular, lo que les permite aprovechar al máximo las características y herramientas de este framework. Además, el equipo tiene interés en aprender cómo compilar un desarrollo web en una aplicación de escritorio, y Angular es conocido por ser una excelente opción para este propósito.

En segundo lugar, Angular ofrece ventajas como una programación más estructurada, lo que ayuda a separar la lógica de la aplicación de la manipulación de los datos. Esto resulta en una mejora en la calidad del código y los hábitos de programación. Al tener una estructura más clara y ordenada, se vuelve más fácil de mantener y escalar.

Además, Angular cuenta con un gran apoyo de distintas comunidades, lo que permite una evolución continua y una gran cantidad de documentación y componentes de otros desarrolladores. Esto significa que los desarrolladores pueden aprovechar las soluciones y mejores prácticas de otros desarrolladores, lo que ahorra tiempo y esfuerzo en el desarrollo de la aplicación. Por último, Angular permite desarrollar con menos código y cuenta con una gran cantidad de documentación y componentes de otros desarrolladores, lo que ayuda a aumentar la eficiencia y productividad en el desarrollo de la aplicación.

También permite un fácil consumo de API's y servicios, lo que simplifica significativamente el proceso de desarrollo y conlleva grandes beneficios en la manipulación de los datos. Angular facilita la integración de servicios externos en la aplicación. Los componentes se pueden utilizar para manejar la lógica de negocio y para mostrar la información obtenida de los servicios externos.

Angular proporciona una forma sencilla de hacer peticiones HTTP a través de su módulo HttpClient, lo que permite recibir y enviar datos de forma eficiente y segura, esto agiliza el proceso de obtención de información de servicios externos y permite una mejor gestión de los datos, lo que se traduce en una mejora en la calidad y rapidez del desarrollo.

Adicionalmente, permite crear componentes aislados cumpliendo con el objetivo de que el software sea flexible y adaptable a los cambios. Los componentes en un sistema desarrollado con Angular tienen varios beneficios, entre ellos:

1. Reutilización: Los componentes son piezas reutilizables de código que se pueden utilizar en diferentes partes de la aplicación. Esto facilita el mantenimiento y reduce la cantidad de código necesario.
2. Separación de responsabilidades: Los componentes se encargan de una tarea específica y tienen un conjunto limitado de responsabilidades, lo que facilita el desarrollo y el depurado de la aplicación.
3. Interacción fácil: Los componentes se comunican entre sí mediante un sistema de eventos, lo que facilita la creación de aplicaciones con una interfaz de usuario rica y dinámica.
4. Facilidad de testing: Los componentes son unidades independientes de código, lo que facilita el testing y la verificación de la aplicación.
5. Mejora la escalabilidad y seguimiento de cambios.
6. Permite una mejor organización del código y una mayor facilidad en su mantenimiento.

La modularidad permite separar las partes de un sistema en componentes independientes, lo que reduce la cohesión y aumenta la legibilidad. Además, las modificaciones sólo afectan a una parte específica y se minimizan los efectos secundarios.

Para realizar la compilación de código web que genera Angular a una aplicación de escritorio, evaluamos 3 opciones: ToDesktop¹⁴, Nativefier¹⁵ y Electron¹⁶.

- Electron es un framework JavaScript de código abierto creado por GitHub capaz de tomar código HTML, CSS, JavaScript y compilarlo en una aplicación de escritorio para las plataformas Windows, Linux y MacOS. Utiliza Chromium como su motor de renderizado de páginas web y Node.js para acceder a las funciones del sistema operativo. Cuenta con documentación precisa de cómo realizar el desarrollo y distribución de la aplicación de escritorio.
- ToDesktop es un servicio online que dada la URL de un sitio web, se convierte a una aplicación de escritorio para las plataformas Windows, Linux y MacOS. Para realizar la conversión a aplicación de escritorio, utilizan el framework Electron.
- Nativefier es un proyecto de código abierto con la misma premisa que ToDesktop, convertir una URL en una aplicación de escritorio. Para realizar la conversión, por detrás utiliza Electron.

Dado que dos de las opciones analizadas, ToDesktop y Nativefier, utilizan Electron como base, se decide utilizar Electron directamente como compilador. Esto nos brinda la posibilidad de lograr lo mismo que las otras dos opciones ofrecen, pero con la ventaja adicional de poder adaptar la compilación a nuestras necesidades específicas del proyecto. Al utilizar Electron directamente, se evita depender de servicios o herramientas externas, lo que puede ser beneficioso en términos de flexibilidad y control del proceso de compilación.

5.4 Elección de la Base de Datos

La base de datos debe almacenar la información general de un curso. Además, debe también almacenar los datos específicos que el docente irá cargando, como los materiales de estudio, tareas, exámenes y otras actividades relacionadas con el curso. Es importante tener en cuenta que cada curso puede tener una estructura de datos diferente, ya sea por la cantidad de información requerida o por los tipos de datos requeridos.

Además, esta base de datos debe ser fácilmente exportable para compartir un curso con otro docente. Esto significa que debe tener la capacidad de exportar los datos de un curso específico en un formato que pueda ser importado fácilmente por otro docente, permitiendo una mayor colaboración y eficiencia en la planificación y ejecución de los cursos.

En resumen, la base de datos debe ser capaz de almacenar y organizar la información general y específica de un curso, adaptándose a las necesidades de cada curso y permitiendo una fácil exportación para compartir con otros docentes.

¹⁴ <https://www.todesktop.com/works-with/javascript>

¹⁵ <https://github.com/nativefier/nativefier>

¹⁶ <https://www.electronjs.org/>

La siguiente tabla (Tabla 3) es un estudio de las principales características entre los modelos de datos SQL¹⁷, Documento JSON (JavaScript Object Notation)¹⁸ nativo y MongoDB¹⁹.

	SQL	Documento Json	NoSQL- MongoDB
Última versión estable	2022	-	6.0
Licenciamiento	Se requiere licencia	Fuente abierta	Fuente abierta
Modelo de base de datos	Base de datos relacional: formato de tablas	Base de datos no relacional: orientada a documentos json	Base de datos no relacional: orientada a documentos (estructura clave-valor)
Requiere Manejador de Base de Datos	si	no	si
Esquema de definición de datos	Esquema fijo	Esquema dinámico	Esquema dinámico
Exportar e Importar	si, archivo grande.	si, archivo liviano.	si, archivo mediano.
Integración con Angular	si	si	si
Experiencia con la Base de Datos	Bastante	Bastante	Muy poco

Tabla 3
Características de modelos de datos.

Se ha elegido utilizar JSON para almacenar los datos de nuestro software debido a sus ventajas en términos de velocidad y eficiencia. JSON es un formato ligero y fácil de leer y escribir, lo que permite realizar operaciones de manera rápida y con poco consumo de recursos, tanto de CPU como de memoria. Además, Angular, el framework elegido para el desarrollo de la aplicación, proporciona una buena integración para el manejo de objetos JSON, lo que facilita la utilización de este formato en el desarrollo.

Otra ventaja de utilizar JSON es que, al tratarse de documentos en formato texto, hace que sea sencilla la tarea de exportar, compartir e importar el conjunto de datos. Esto es especialmente útil en casos en los que se desea compartir los cursos con otros docentes.

5.4.1 Base de datos de documentos JSON

Las aplicaciones que utilizan diferentes tipos de datos JSON y lenguaje de consultas orientado a dicho formato pueden interactuar con los datos almacenados en una base de datos documental JSON.

¹⁷ <https://www.microsoft.com/es-es/sql-server/>

¹⁸ <https://www.json.org/json-es.html>

¹⁹ <https://www.mongodb.com/docs/>

Características que definen una base de datos de documentos JSON:

- Una base de datos de documentos JSON es una base de datos no relacional diseñada para almacenar y realizar consultas como documentos JSON.
- Los datos JSON de la base de datos son textuales, pero el texto se puede almacenar utilizando el tipo de dato BLOB, VARCHAR2, CLOB o el tipo de datos JSON binario en la versión 21c.
- El acceso a los datos JSON almacenados en la base de datos se lleva cabo de forma similar a otras bases de datos, e incluye el uso de OCI, .NET y JDBC
- Los datos en formato JSON de una base de datos de documentos JSON se pueden almacenar, indexar y consultar sin necesidad de contar con un esquema de definición de datos.

JSON presenta un formato ligero de intercambio de datos que marca la pauta de la transferencia de documentos.

5.5 Servidor Express para una aplicación Angular

Los documentos JSON se almacenan en el dispositivo del cliente, ya sea en la memoria interna o en un disco externo. Aunque Angular no proporciona un servicio nativo para el manejo de archivos locales, existen formas de implementar la gestión de archivos locales en aplicaciones Angular.

Para interactuar con los archivos en el dispositivo del cliente, se requiere un servidor instalado en el mismo. Una opción es utilizar Node.js con Express.

- **Node.js:** Es un entorno que trabaja en tiempo de ejecución, de código abierto, multiplataforma, que permite a los desarrolladores crear toda clase de herramientas de lado servidor y aplicaciones en JavaScript. La ejecución en tiempo real está pensada para usarse fuera del contexto de un explorador web (es decir, ejecutarse directamente en una computadora o sistema operativo de servidor). Como tal, el entorno omite las APIs de JavaScript específicas del explorador web y añade soporte para APIs de sistema operativo más tradicionales que incluyen HTTP y bibliotecas de sistemas de ficheros.
- **Express:** Framework web más popular de Node, y es la librería subyacente. Con una gran comunidad de desarrolladores y seguidores, se considera un estándar para crear aplicaciones web y API con Node.js. Proporciona un conjunto de funciones de middleware para manejar solicitudes HTTP y rutas, así como una interfaz simple para crear aplicaciones web y APIs. Con Express, se pueden crear aplicaciones web con un enfoque en la lógica del servidor y manejar solicitudes HTTP, tales como GET, POST, PUT y DELETE.

Utilizando estas herramientas, se pueden crear rutas para subir y descargar documentos JSON en la aplicación Angular, lo que permite almacenar y recuperar los datos de forma local.

5.6 Frontend

El Frontend se desarrolla en Angular, usando el Design System de Bootstrap que proporciona los estilos, iconos y funcionalidades para una mejor experiencia de usuario.

En la etapa de requerimientos, se relevó que la interfaz gráfica debería ser dinámica y flexible, ya que los cursos no tienen porqué ser iguales y esto dependerá de la institución. Esto implica que para distintas instituciones educativas, la presentación al usuario puede ser diferente sin ser necesario modificar el código fuente del software.

Además, la aplicación debe ser offline y los cursos deben poder ser exportados e importados fácilmente.

Para cumplir con estos objetivos, se decide contar con un archivo que cumpla la función de esquema de presentación. Este archivo contiene todas las pautas para representar la aplicación de tal manera que cumplan los requerimientos y la maqueta. El archivo se representa en formato json utilizando objetos, arreglos y datos primitivos. A partir de ahora, llamaremos a este documento como Esquema del Curso y será explicado en detalle más adelante.

En el esquema se encuentran las etapas, grupos de atributos, atributos, datos y demás particularidades del curso.

En el sistema pueden existir distintos esquemas de cursos. Cada esquema pertenece a una institución y puede ser personalizado con sus parámetros del sistema. Por defecto se proporciona un esquema, que será general, y viene pre cargado con el software. Luego se podrán instalar esquemas personalizados por una institución. Todos los esquemas de una institución estarán almacenados en el servidor correspondiente a dicha institución y los usuarios autenticados en ella podrán usarlos.

Un curso tendrá un único esquema de curso que lo representa y este podrá ser actualizado por el administrador del curso. En ese caso, cuando la aplicación de escritorio del docente tenga acceso a internet, este recibirá una notificación advirtiéndolo de que el curso será actualizado.

5.7 Instalación de las Herramientas

Se instalan las siguientes versiones de las herramientas:

- Angular 14.1.0
- Node 18.12.1
- Bootstrap 5.2.0
- Electron 20.0.1
- Express 4.18.1
- Gitlab

5.8 Interfaz Gráfica

Realizado el proceso de análisis de los diferentes requerimientos y la maqueta, se define la estructura que mantendrá la Interfaz.

El color primario definido para todo el software es #0d6efd por su buen contraste y definición en todas las pantallas.

La fuente usada para todo el software es Raleway, en los siguientes tamaños:

- Título: 2.5rem
- Subtítulo: 1.5rem
- Normal: 1rem

5.8.1 Desarrollo del Menú (Grafo)

La maqueta del menú se parece mucho a un grafo y allí Javascript ofrece distintas bibliotecas para manipular grados.

La implementación del menú se hizo utilizando la biblioteca D3.js²⁰ de Javascript que sirve para manipular documentos basados en datos. Permite realizar representaciones visuales, dinámicas e interactivas en navegadores web, utilizando HTML5 , CSS3 y SVG, de manera de manipular el DOM ajustándose a los datos deseados.

Entre las muchas características a favor que tiene d3.js que nos llevaron a optar por ella, se pueden nombrar la gran cantidad de ejemplos en la web. También la gran comunidad que utiliza d3.js permite que haya muchos recursos de aprendizaje como guías y cursos, así como rápidas respuestas en los foros de preguntas.

Se implementó una interfaz para pasar los datos desde el software en angular hacia la biblioteca D3.js. Allí se toma el esquema del curso y se genera un objeto json con la información requerida para su funcionamiento. El objeto json cuenta con una lista de objetos (nodos) y cada objeto a su vez con una lista de objetos (nodos) dentro. Las relaciones del nodo se representan en otra lista, referenciando los identificadores de todos los nodos relacionados.

Esta interfaz, además, cuenta con un evento que captura si se hace click sobre un nodo y en ese caso se llama a una función de angular que desencadena todo el proceso de visualización correspondiente.

5.8.2 Esquema de la Interfaz

El esquema es un documento JSON. Cada esquema cuenta con un campo institución de tipo string y un campo versión de tipo numérico, donde la institución indica la URL hacia el servidor central de la institución educativa y la versión indica la versión del proceso de diseño de un curso.

Para esa combinación institución - versión, el esquema tiene un arreglo de objetos llamado etapas que representa cómo se visualizará el proceso de diseño de un curso. Cada etapa tiene un campo "id" de tipo entero, "nombre" de tipo string, "descripción" de tipo string y un arreglo de objetos llamado grupos que representan los atributos, agrupados, que tiene la etapa.

Con esa información del esquema se representa el menú cómo se ve en la imagen a continuación (Figura 27).

²⁰ <https://d3js.org/>

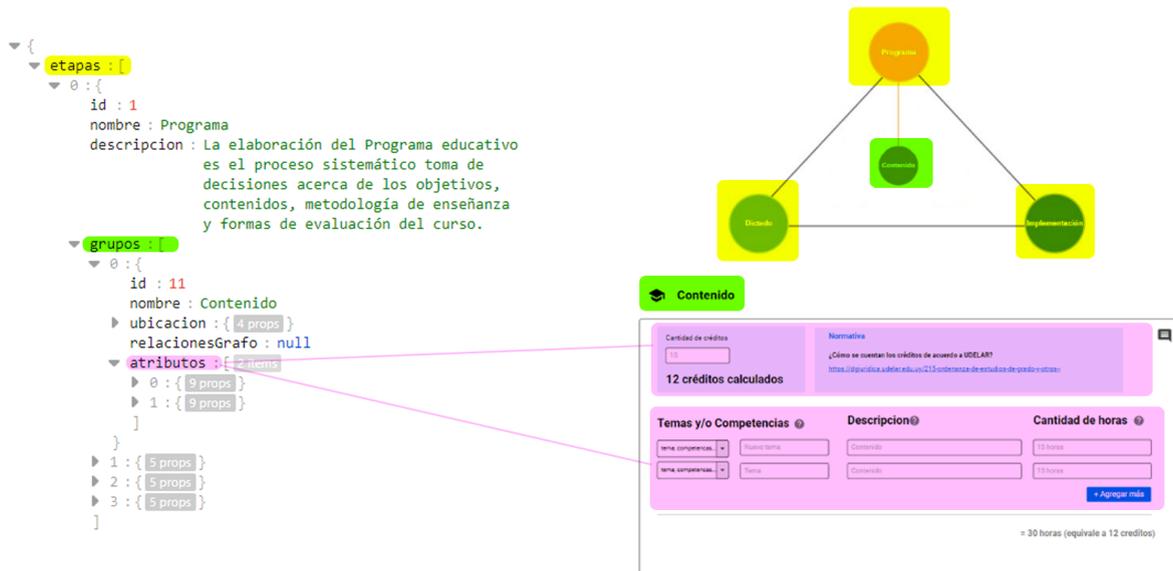


Figura 27
Representación de datos y menú.

Para los atributos se definen propiedades adicionales, cómo si se puede tener más de una instancia, si es puntos críticos, y si está relacionado con un atributo de otro grupo o etapa.

Un objeto que corresponde a un atributo contiene una lista de objetos, las filas de datos. Esas filas de datos serán usadas para representar en la pantalla la información de manera ordenada, tal como sea configurado por el administrador. Cada objeto dentro de la fila de datos representa un dato.

En la Figura 28 se continúa con el ejemplo anterior. Se puede observar el Atributo "Cantidad de créditos", que consta de una sola fila de datos con un valor.

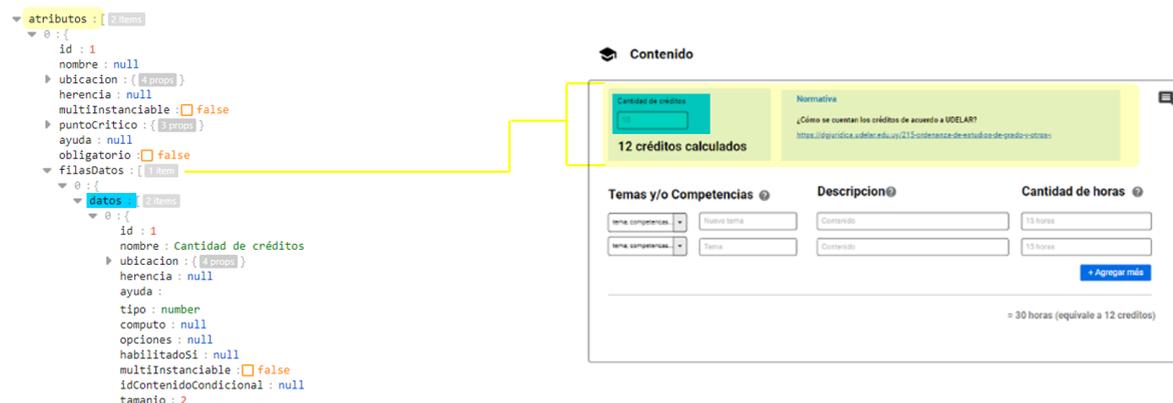


Figura 28
Ejemplo - Atributo cantidad de créditos.

Cada objeto del Json (grupo, atributo, dato) contiene su ubicación. La ubicación es un objeto que almacena la etapa, grupo, atributo y dato al que corresponde. Esa información es usada para saber donde debe ser mostrado el objeto y para almacenar los datos que ingrese el usuario desde la interfaz. Funciona como un identificador único.

En la Figura 29 se tiene un ejemplo de ubicación de atributo:

```
{
  ubicacion : {
    idEtapa : 1
    idGrupo : 11
    idAtributo : 1
    idDato : null
  }
}
```

Figura 29
Ejemplo ubicación de un atributo.

Los objetos que representan los datos, tienen una propiedad de tipo. El tipo define cómo se va a visualizar en la interfaz ese dato tal cómo se detalla en la Tabla 1 - Tipo de datos.

Los atributos que pueden ser multivaluados tienen una propiedad de tipo booleano que los determina de la siguiente manera: "multiInstanciable": false ó "multiInstanciable": true. En el esquema, cada Atributo describe cómo se dibuja en pantalla y en qué espacio de la base de datos persisten los valores ingresados. Para indicar que se pueden grabar en la base de datos más de una instancia de un Atributo, se usa el booleano "multiInstanciable". Sin el booleano, no hay forma de saber si dado un Atributo puede ocupar más de un espacio en la base de datos.

Algunos datos pueden ser calculados en tiempo de ejecución, es decir, toma la información cargada en otros datos para aplicar una función y auto calcula su valor. Es el caso de los créditos sugeridos de un curso. Allí se toma la cantidad de horas de clase o similar sumado a la cantidad de horas de estudio personal y eso se divide entre 10. Para aplicar esa función, se usa un objeto dentro del dato que contiene las ubicaciones de los datos a usar y los operadores que se deben aplicar. Los operadores aceptados son: suma, resta, multiplicación y división. Esto hace que la función sea flexible y dinámica, la lógica está en el software pero los parámetros que se toman están en el esquema.

Hay datos de atributos que dependen de otros datos. Por ejemplo, la cantidad de estudiantes en un cupo depende y será visible si anteriormente se seleccionó el que el curso acepta cupo. Esto se maneja en 2 datos aparte, uno para el "acepta cupo" y por otro lado la cantidad en el cupo. El dato de cantidad de cupo tiene la propiedad "habilitado si" que será un objeto. El objeto tiene una referencia, que indica de qué otro dato depende y una propiedad de valor seleccionado para conocer cuál es la opción que debe haber elegido el docente para que se active el dato actual.

El comportamiento descrito anteriormente se observa representado en las Figuras 30 y 31.

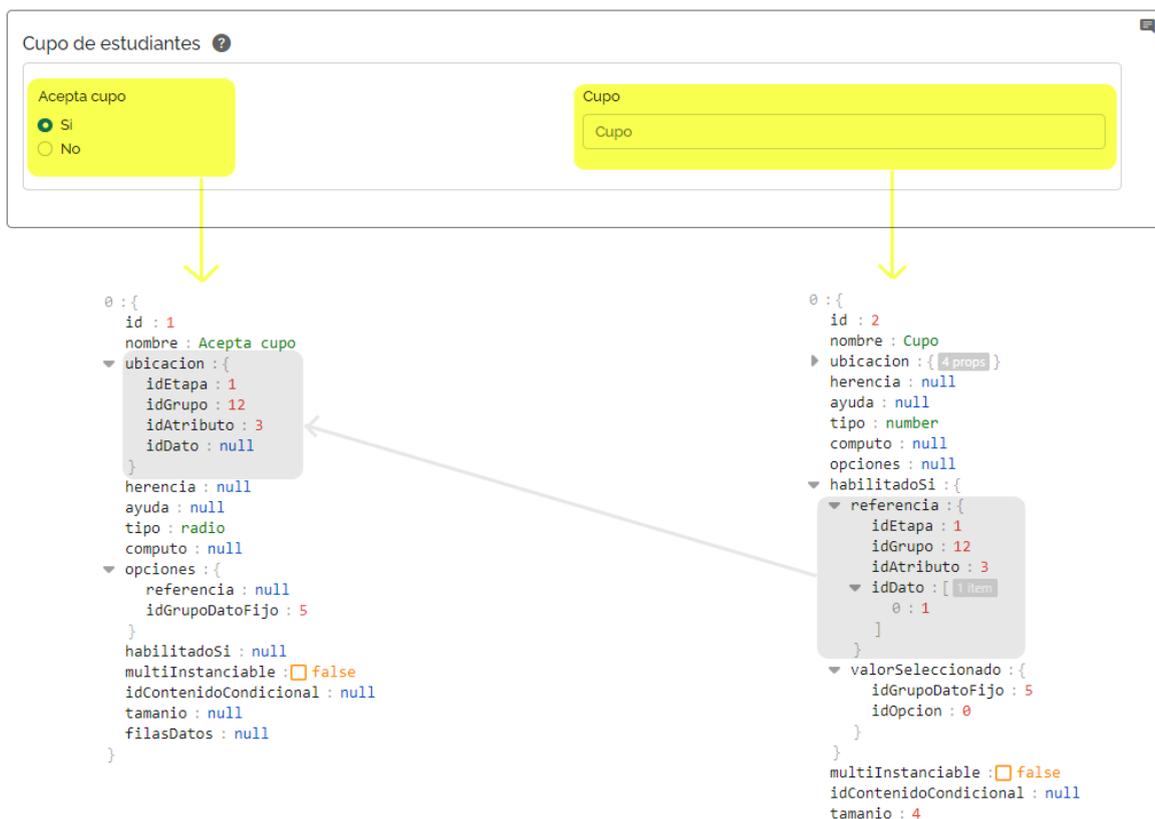


Figura 30
Ejemplo dato de atributo que está habilitado si se da una opción de otro dato.

Figura 31
Ejemplo dato de atributo que no está habilitado si se da una opción de otro dato.

5.8.3 Atributos con Herencia

Existe una relación entre ciertos atributos. Esa relación, que a partir de ahora llamaremos herencia, hace que un atributo pueda heredar de otro. Esto habilita la funcionalidad de cargar la información en los datos del atributo que heredado desde los datos del atributo que hereda. Este comportamiento se observa en los mismos atributos de distintas etapas.

Por ejemplo, Recursos pertenece a la etapa de Programa y de Contextualización. El atributo Recursos de Contextualización hereda del atributo Recursos de Programa, cómo se puede observar en la Figura 32.

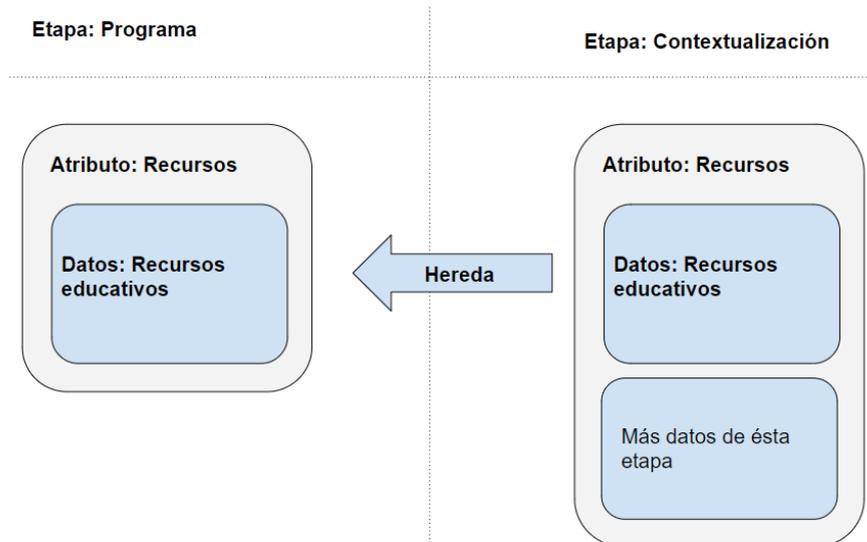


Figura 32
Ejemplo en diagrama de un atributo que se hereda de otro.

El comportamiento sobre estos datos es consultado con el usuario experto en el dominio del proyecto y se le ofrecieron diferentes opciones:

1. Que sean atributos y datos independientes: No guardan una dependencia y ambos podrían tener información distinta.
2. Que sean atributos y datos idénticos: Ambos datos almacenan siempre la misma información. Si se modifica uno, entonces se verá alterado el otro atributo.
3. Los atributos guardan una dependencia, pero el docente decide si quiere cargar la información desde el Heredado y podrá hacerlo en cualquier momento a través de un botón.

Se eligió la tercera opción, ya que es la que mejor se ajusta a la realidad y facilita la tarea al docente.

Entonces, con este nuevo comportamiento y siguiendo con el ejemplo anterior, en Recursos de Contextualización se encuentra un botón con la leyenda: "Cargar datos desde Recursos de Programa" como se observa resaltado en la Figura 33.

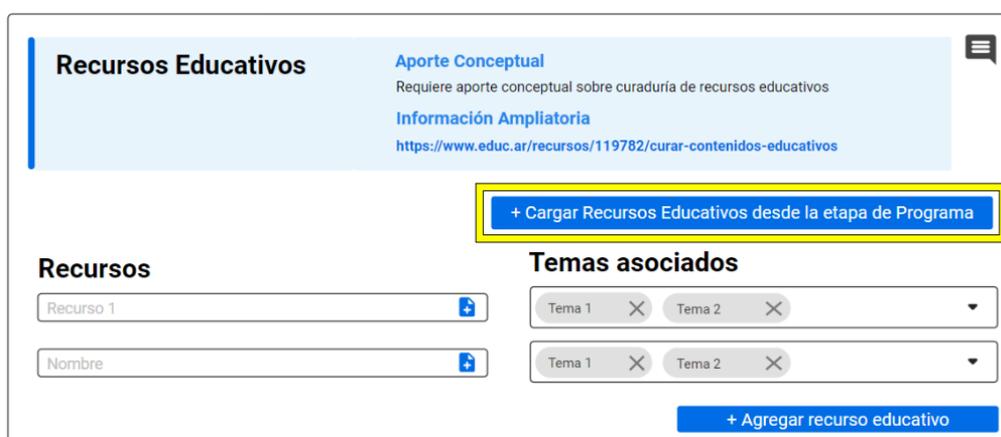


Figura 33

Ejemplo atributo que hereda y se muestra botón para cargar información.

La acción asociada al botón es traer todos los datos del atributo desde el que hereda. La estructura de los datos en el esquema para estos atributos se representa en la Figura 34.

```
0 :{
  id : 1
  nombre : Recursos
  ▾ ubicacion :{
    idEtapa : 2
    idGrupo : 23
    idAtributo : null
    idDato : null
  }
  ▾ herencia :{
    idEtapa : 1
    idGrupo : 14
    idAtributo : 1
    idDato : null
  }
  multiInstanciable :  true
  puntoCritico : null
  filasDatos : null
}
```

Figura 34

Ejemplo de la estructura de un atributo que hereda de otro.

Para implementar este comportamiento se usa una propiedad “*herencia*” en el objeto de tipo Atributo. Esa propiedad es un objeto que representa una ubicación, que tiene los identificadores necesarios para acceder a la ubicación del atributo desde el que hereda.

Además, un atributo que hereda de otro también puede contener datos propios. En ese caso se agrega una colección de fila de datos como hemos visto en casos anteriores. Esos datos serán una extensión a los datos heredados.

5.8.4 Atributos con Orquestación

El atributo de este tipo describe los módulos de un curso y su información adicional en unidades. Es un atributo compuesto por una colección de objetos que representan los módulos del curso. Cada módulo, a su vez, contiene una colección de unidades, también representadas como objetos. Ambas, módulos y unidades, son listas multivaluadas, lo que significa que el usuario puede agregar o eliminar objetos en cualquier momento. La Figura 35 representa como se visualiza un atributo de este tipo.

Orquestación

Aporte
 La estructura del curso debe considerar los tiempos y espacios de trabajo con los que se cuenta y debe asegurar que se cumplen con los requisitos para asignar los créditos que otorga.

Información ampliatoria
<https://cead.pressbooks.com/chapter/11-9-paso-7-diseñar-la-estructura-del-curso-y-las-actividades-de-aprendizaje/>

Contenido agrupado en: ? Confirmar

Modulo 1

Fecha inicio

Fecha fin

Modalidad

Espacios físicos utilizados

Plataformas disponibles

}

Horas	Temas y/o Competencias	Recursos	Bibliografía
<input type="text" value="10"/>	<input type="text" value="Tema1"/> <input type="text" value="Tema2"/>	<input type="text" value="Recurso 1"/>	<input type="text" value="Biblio 1"/> <input type="text" value="Biblio 2"/>

Descripción

+ Agregar unidad

+ Agregar Módulo

} **Unidad dentro del módulo**

Cada módulo

Unidad dentro del módulo

Figura 35

Atributos de tipo orquestación con módulos y unidades.

Dentro del esquema del curso, encontramos que este tipo de atributos se representan como datos dentro de otros datos. Para representar la estructura se necesita que uno de los datos tenga una lista de datos dentro. Es decir, el atributo tendrá una lista de datos, y uno de ellos será un objeto con su propia lista de datos. Esta estructura recursiva se puede ver representada en la Figura 36 y más en detalle en la Figura 37.

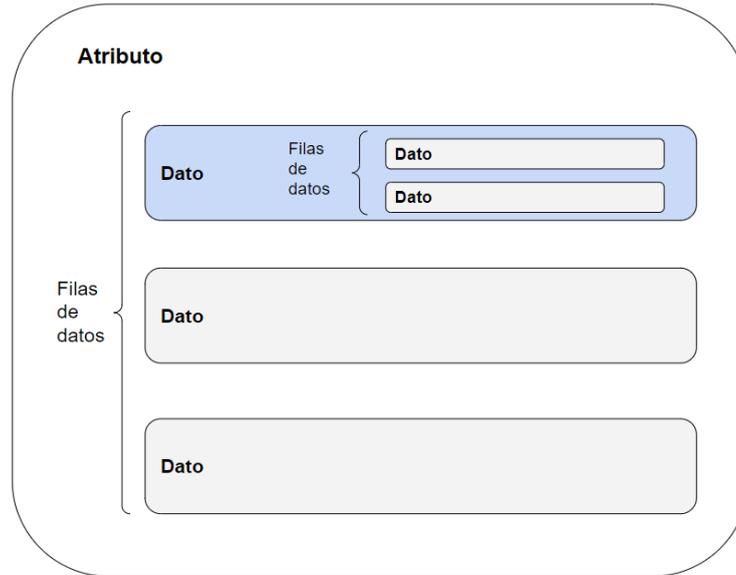


Figura 36
Atributos de tipo orquestación.

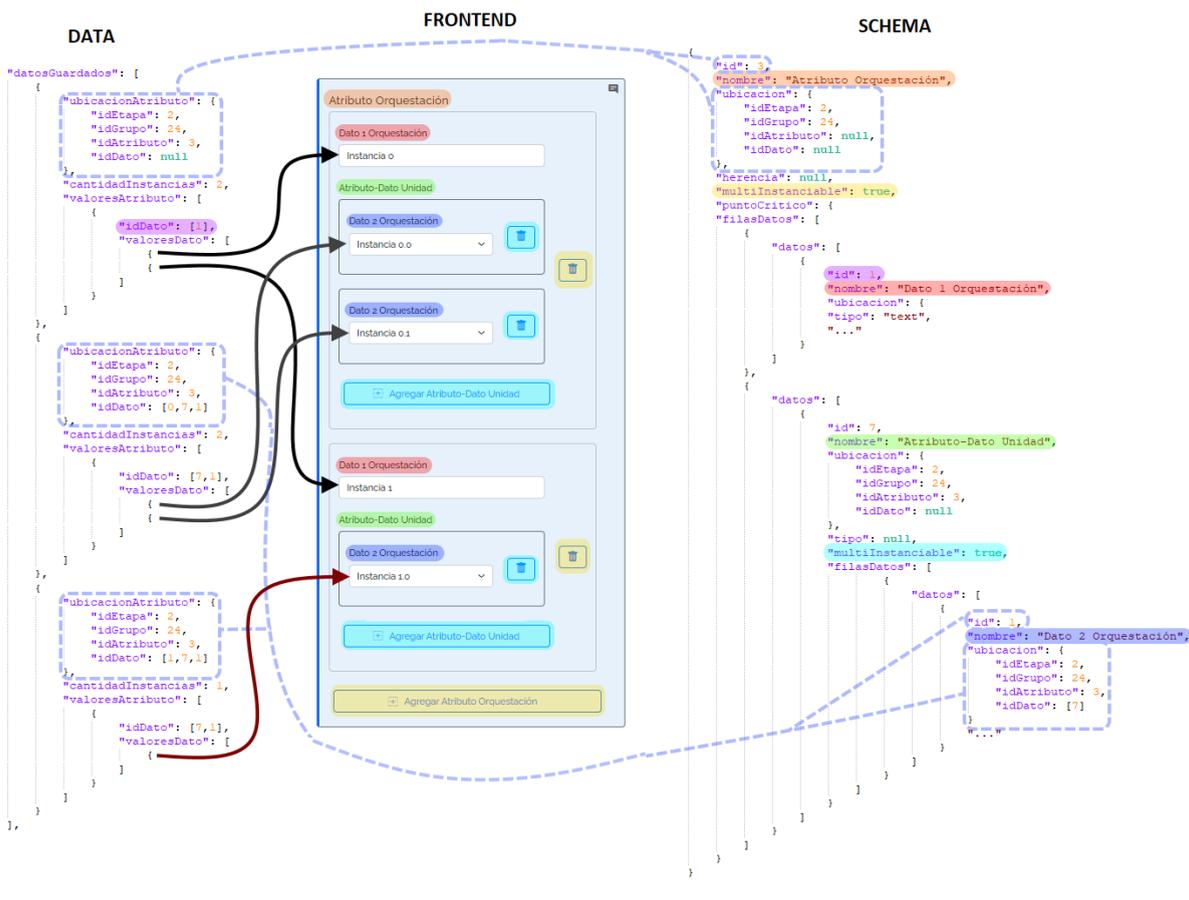


Figura 37
Esquema para atributos de tipo orquestación.

Como se mencionó anteriormente, el proyecto se distribuyó en sprints de Scrum y esta complejidad fue identificada durante desarrollo del cuarto sprint. Hasta ese momento, la estructura de datos utilizada permitía representar correctamente la complejidad requerida.

Sin embargo, debido a la nueva complejidad identificada, fue necesario realizar una revisión y adaptación de la estructura de datos para satisfacer las necesidades del proyecto y lograr una representación adecuada de la información.

Al añadir un nivel adicional de datos, es necesario agregar complejidad a los índices de los datos del nuevo nivel para lograr una representación correcta. Se realiza un cambio en los índices para diferenciar los datos o unidades del módulo. Esto se logra mediante una ligera alteración de la ubicación, añadiendo un "0" al comienzo en la propiedad idDato para la primera unidad y aumentando este índice de manera secuencial. Este proceso garantiza que los datos se almacenen y se identifiquen de manera única y clara en el sistema, lo que permite una representación precisa de la información.

5.8.5 Datos fijos

Se trata de una lista de objetos dentro del esquema del curso que representan los datos estáticos y requeridos por una institución para describir un curso. Por ejemplo, las carreras o las Áreas de conocimiento. Esta lista es configurada de acuerdo a los requerimientos específicos de cada institución y puede ser actualizada y modificada según sea necesario.

Cada objeto en la lista tiene un identificador único, un nombre y una lista de opciones asociadas. Cada opción es un objeto individual con su propio identificador único dentro del dato, identificador de grupo y valor.

Como se puede observar en el ejemplo de la Figura 38, si el atributo es "carreras", se utilizarán los datos específicos de la lista para cargar las opciones disponibles en el combobox correspondiente en el sistema.

Información general

Nombre de la unidad curricular ⓘ
Metodología de la investigación social

Carreras ⓘ
Cio Social del Noreste

Carreras ⓘ

- Cio Social del Noreste
- Técnico en Desarrollo Sustentable
- Lic. en Educación Física

```
0 : {  
  id : 2  
  nombre : Carreras  
  opciones : [ 3 items  
    0 : {  
      id : 0  
      idGrupo : 2  
      valor : Cio Social del Noreste  
      muestroSi : null  
    }  
    1 : {  
      id : 1  
      idGrupo : 2  
      valor : Técnico en Desarrollo Sustentable  
      muestroSi : null  
    }  
    2 : {  
      id : 2  
      idGrupo : 2  
      valor : Lic. en Educación Física  
      muestroSi : null  
    }  
  ]  
}
```

Figura 38

Ejemplo atributo de tipo combobox que es cargado con una lista de opciones.

5.9 Estructura de los datos

El almacenamiento de la información ingresada por los docentes durante el proceso de diseño de un curso se almacena en un objeto dentro de un archivo JSON. Entre los datos de este archivo se encuentra un identificador secuencial (local al PC donde se almacena), un nombre de curso, la URL del servidor central de la institución a la que pertenece y una lista de versiones.

Cada versión es un objeto que almacena una lista de datos. Los datos son objetos que contienen información sobre los atributos. También incluye una versión de esquema que indica con qué versión del proceso de diseño de curso se deben interpretar los datos almacenados.

En la Figura 39 se puede ver un ejemplo de un curso, con su metadata y cómo se representa la información en el Frontend.

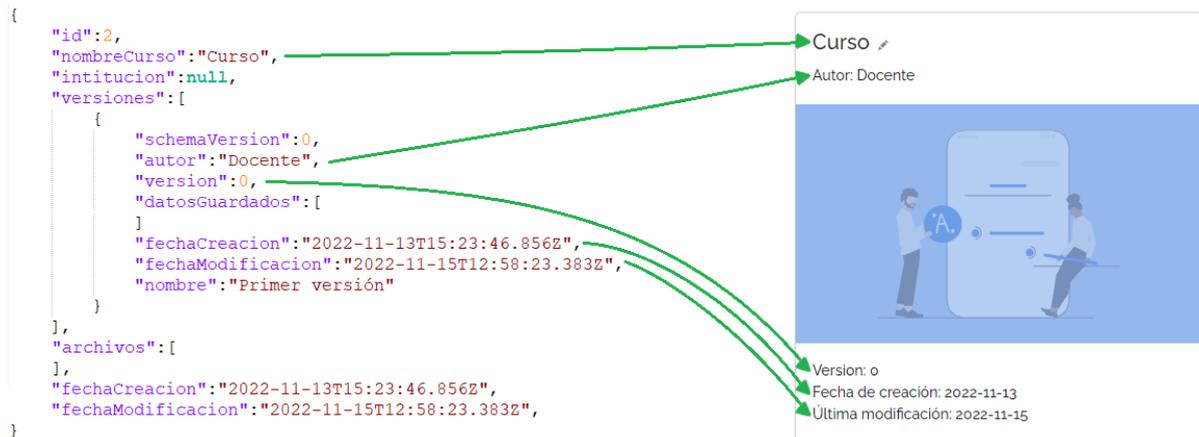


Figura 39
Metadata de un curso almacenado y su representación en el Frontend.

La información generada en el proceso de diseño de curso se guarda en una lista de objetos de atributos. Cada objeto de atributo contiene una colección de valores de datos correspondientes a cada dato definido en el esquema de ese atributo.

La Figura 40 muestra cómo el software interactúa con el esquema y el frontend para mostrar un atributo con dos datos de tipo texto. Para presentar la información en la interfaz gráfica, se obtienen los atributos y los datos del esquema. Cada dato tiene una ubicación clave.

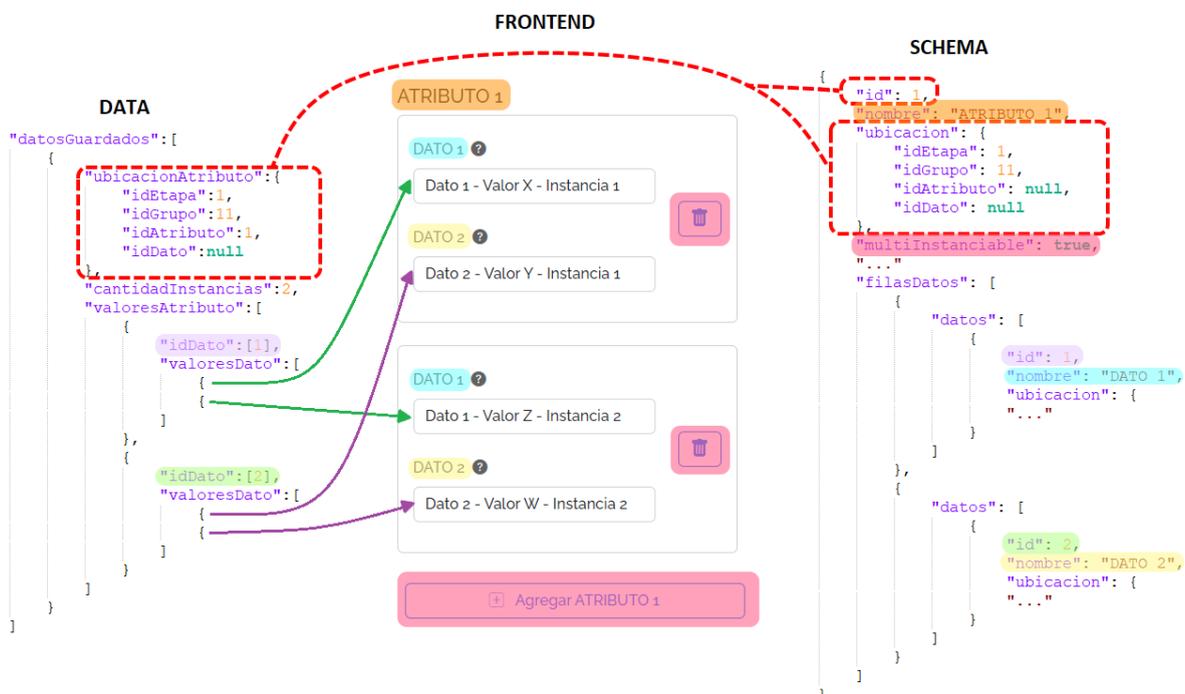


Figura 40
Interacción de la información guardada en el Schema y su visualización en el Frontend.

Después de cargar el frontend, se busca la información almacenada para cada dato presentado en la interfaz, buscando por clave mediante su identificador en los datos guardados. La Figura 41 representa este proceso.

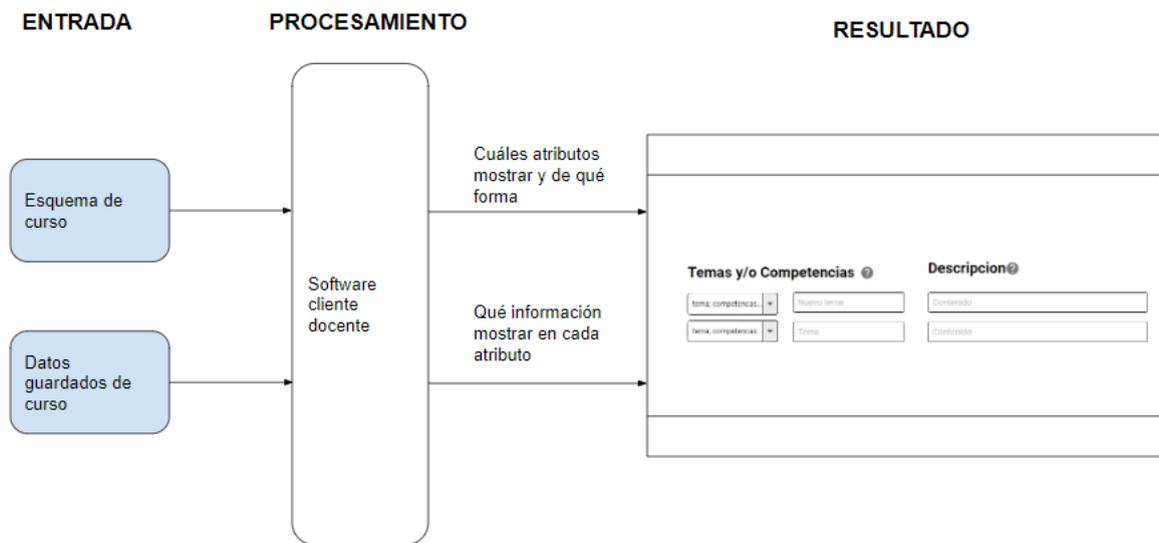


Figura 41
Interacción con el esquema y los datos guardados para armar la pantalla.

Esta forma de interacción entre la información guardada, el frontend y el esquema es común para todos los atributos del sistema. El software utiliza el esquema para obtener la estructura y los tipos de datos de los atributos. Luego, utiliza la información guardada para recuperar los valores de los datos correspondientes a cada atributo. Finalmente, utiliza el frontend para presentar esta información en una interfaz gráfica accesible para el usuario. Este proceso se repite para cada atributo en el sistema.

5.9.1 Sistema de guardado automático de información

El sistema permite que la información ingresada por el usuario sea guardada de manera automática, sin necesidad de que el usuario tenga que hacerlo manualmente. Esto se logra mediante un evento que está atento a la interfaz y que detecta cualquier cambio en los atributos. En ese momento, el sistema almacena automáticamente el valor cambiado en la base de datos.

Este enfoque mejora la experiencia del usuario, ya que este no tiene que preocuparse por recordar guardar los cambios o ejecutar un evento para hacerlo. Esto significa que el usuario puede centrarse en su trabajo y realizar sus tareas de manera más eficiente, sin tener que preocuparse por la pérdida de información por descuido.

Además, la automatización del proceso de guardado de la información garantiza que los últimos cambios realizados por el usuario sean guardados y no se pierdan. De esta manera, se evita una situación en la que el usuario olvide guardar los cambios y luego pierda horas de trabajo. En resumen, el sistema de guardado automático de la información mejora la eficiencia y reduce el riesgo de pérdida de información.

5.9.2 Almacenamiento de Documentos

La herramienta permite almacenar y subir distintos tipos de documentos y archivos en diferentes secciones, por ejemplo en la bibliografía como se puede ver en la Figura 42.

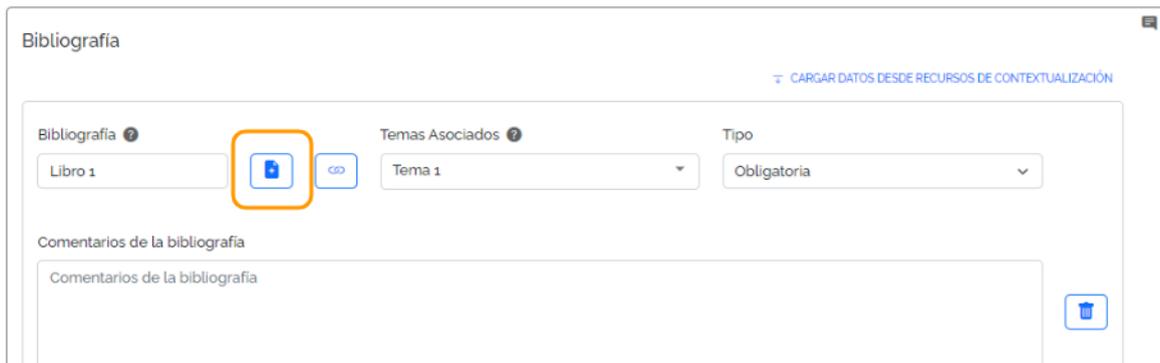


Figura 42
Ejemplo atributo que permite adjuntar archivos.

Inicialmente, estos documentos se guardaban en un directorio específico en el PC del usuario y se tomaban desde allí. Sin embargo, esta solución tenía un problema: al exportar un curso para enviarlo a otro docente, los documentos se perdían o debían adjuntarse junto con el curso. Con un curso con un gran volumen de documentos, esto resultaba trabajoso para el docente y podría olvidarse de qué documentos tenía almacenados.

Para resolver este problema, se decidió convertir los documentos y archivos a binario en formato base64. De esta manera, el esquema de un curso almacena una lista de objetos que funciona como una biblioteca de archivos dentro del curso.

Esta solución resulta eficiente, ya que si un mismo documento se refiere en más de un atributo, el documento no ocupa memoria por cada uno, sino que se refiere al mismo documento en todos los lugares. Además, para el docente esta lógica es transparente y no tiene que preocuparse ni ocuparse de exportar cada documento al momento de exportar un curso.

Cuando se quiere almacenar un archivo en un dato, se tiene un objeto de tipo archivo. En él se almacena un identificador "fileId" que es la clave para encontrar el archivo en la biblioteca de archivos para el curso. Además, se guardan otros datos, como el nombre del documento y un nombre adicional que puede ser ingresado por el docente. En la Figura 43 se puede ver cómo se almacenan estos objetos y cómo son referenciados desde un atributo.

```
, {
  "string": null,
  "number": null,
  "selectFijo": null,
  "selectUsuario": null,
  "archivo": {
    "texto": "resultado de la prueba",
    "fileName": "Metodologia de la investigaciónV1.pdf",
    "fileId": 1,
    "ruta": null
  },
  "date": null
}, {
  "archivos": [
    {
      "id": 1,
      "b64": "data:application/json;base64,ewogICAgImlkIjogNSwKICAgICJub21icmVddXJzbyI6ICJNZXRvAgTmludG10dW9uZmVudC01IiwKICAgICJ2ZXJzaW9uZXMiOiBbewKICAgICJ2ZXJzaW9uIjogMSwKICAgICAgICAgICAgImRhdG9zR3Vh"
```

Figura 43
Estructura para almacenar documentos y archivos.

5.10 Versionado

Un curso puede contar con distintas versiones del curso y el docente elegirá en qué momento crear una nueva versión a través de un botón. Debajo del menú se tiene un botón para ver el historial de versiones como se observa en la Figura 44.



Figura 44
Ver historial de versiones.

Una nueva versión es una copia de la versión actual, pero a partir de esta versión se pueden realizar modificaciones, conservando una versión anterior que se puede restaurar en cualquier momento.

En la Figura 45 se puede ver un ejemplo de un curso con 3 versiones del curso. La versión activa actualmente se encuentra resaltada con un tag.



Figura 45
Ver historial - Lista de versiones.

Cada versión anterior tiene un botón "Cargar versión" que permite restaurar esa versión. Al restaurar una versión anterior del curso, se convierte en la nueva versión activa. Además, la versión actual cuenta con un botón para crear una nueva versión a partir de ella.

5.11 Exportar e importar cursos

Esta funcionalidad resulta importante para realizar un trabajo colaborativo sobre un curso. Un docente puede compartir el trabajo realizado sobre un curso con otros docentes.

El software permite exportar un curso en un archivo, el formato elegido para este archivo es el documento json, ya que es en el formato que se eligió para la base de datos. Por lo tanto, el documento json exportado contiene los datos guardados del curso. En la Figura 46 se puede ver donde se encuentra la funcionalidad para exportar un curso.

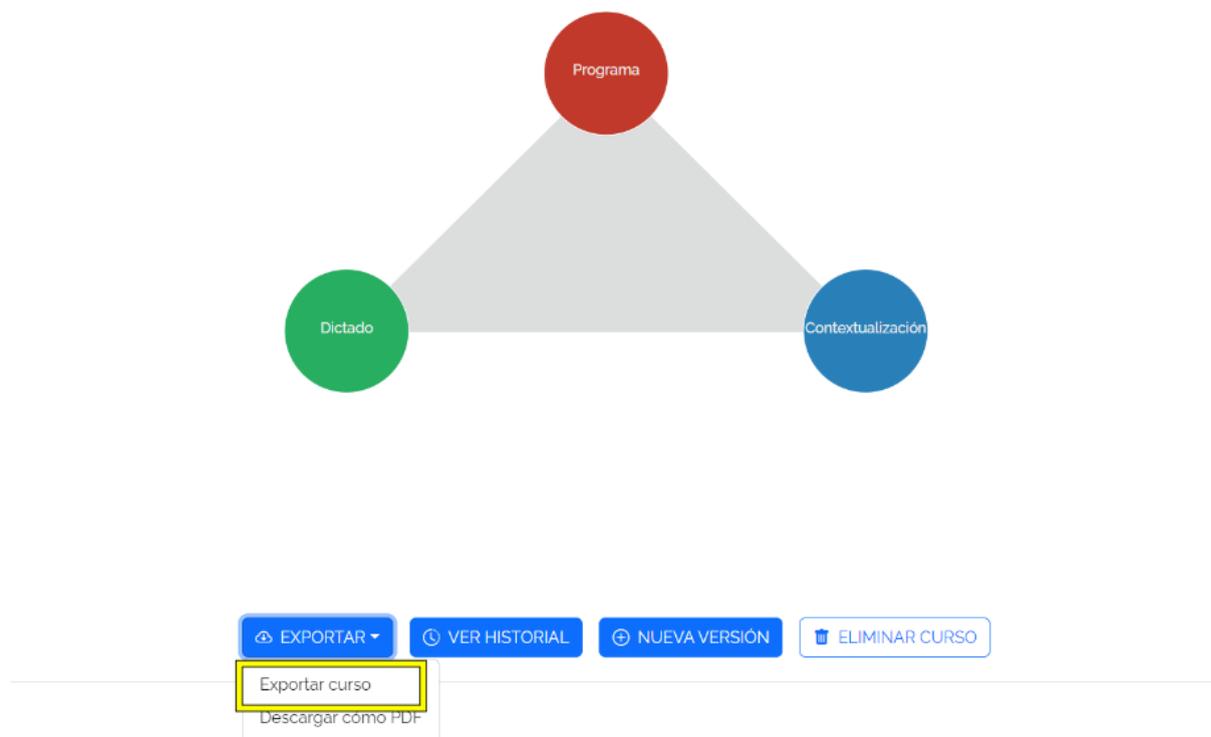


Figura 46
Exportar curso - json.

Luego se puede tomar ese documento json e importarlo en el software. Esta acción se puede realizar desde el botón correspondiente señalado en la Figura 47.

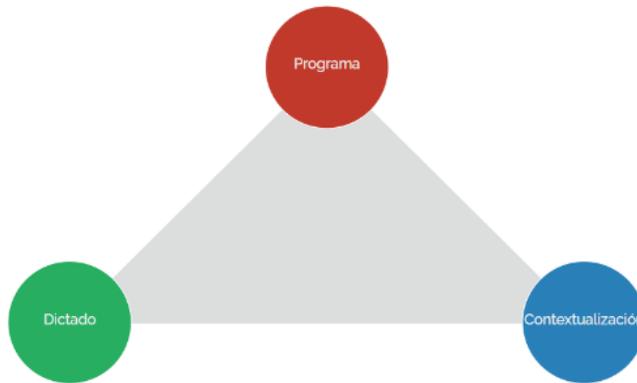
Cuando se importa un curso desde un json, al finalizar el proceso de importación del curso, este se verá en la lista de cursos del docente y a partir de ese momento podrá acceder a él como si fuese suyo.

Mis cursos

<p>Metodología de la investigación /</p> <p>Autor: Mariana Porta</p>  <p>Version: 3 Fecha de creación: 2022-12-29 Última modificación: 2022-10-13</p>	<p>Matemáticas 2022 /</p> <p>Autor: Seba</p>  <p>Version: 1 Fecha de creación: 2022-11-07 Última modificación: 2023-01-05</p>
--	--

Figura 47
Importar curso - json.

Un curso también puede ser exportado en un documento PDF. Este curso hace que sea un documento legible por el docente, donde se imprimen los datos guardados del curso. Para realizar esta tarea, se debe recorrer el esquema del curso, para cada atributo y con la clave de la ubicación se busca en los datos guardados del curso si para un atributo se ingresó información. Esta acción se puede realizar desde el botón resaltado en la Figura 48.



EXPORTAR VER HISTORIAL NUEVA VERSIÓN ELIMINAR CURSO

Exportar curso

Descargar cómo PDF

Figura 48
Exportar curso en documento PDF.

Para implementar esta funcionalidad se usó pdfmake²¹, una biblioteca de JavaScript que permite generar documentos PDF de forma programática.

PDFMake se basa en un lenguaje de marcas para describir la estructura y contenido del documento PDF. Este lenguaje de marcas se utiliza para definir las características del documento, como las fuentes, los encabezados y los pies de página, así como para especificar el contenido del mismo.

Para utilizar pdfmake.js, se debe incluir la biblioteca en el proyecto y luego se puede utilizar para generar un documento PDF. Esto se hace utilizando un objeto de configuración que especifica la estructura y el contenido del documento. En este objeto se especifican las distintas secciones del documento. Una vez que se ha configurado el objeto, se puede llamar a la función de pdfmake para generar el documento PDF.

Se consultó al usuario experto en el dominio y se acordó que en el documento PDF se mostrarán los atributos vacíos para aquellos campos en los que aún no se ha ingresado información.

En la Figura 49 se puede observar cómo se visualiza un curso exportado como documento PDF.

Metodología de la investigación

Autor: Mariana Porta

Programa

Contenido

Cantidad de créditos: 10
Créditos calculados:

Temas y/o Competencias:

Nombre de tema: Módulo 1: Qué es la ciencia. Fundamentos Epistemológicos
Descripción: ¿Qué es la ciencia y cuáles son sus objetivos? ¿Cómo se genera el conocimiento científico? ¿Qué y cómo observa la ciencia? ¿Cuáles son algunas preguntas que han conformado el debate epistemológico del SXX?
• Conceptos, objetos y problemas epistemológicos de la investigación. Conocimiento científico y conocimiento no científico.
• La ciencia: observación y enunciados. Inducción y deducción
• La ciencia y las ciencias sociales, vigilancia epistemológica.
• La ciencia como empresa social: construcción y gestión del conocimiento
Horas de clase o similar: 10

Temas y/o Competencias:

Nombre de tema: Módulo 2: El diseño de investigación
Descripción: ¿Qué requisitos tiene una investigación para generar conocimiento científico?
¿Cómo es el diseño de investigación si se pretende describir o si se pretende explicar un fenómeno? ¿Qué garantiza la validez del conocimiento generado?

• Las preguntas iniciales, el planteo de un problema de investigación y el rol de la teoría y el marco teórico.
• De las preguntas a las hipótesis: los tipos de hipótesis, las hipótesis rivales. Las variables y el proceso de operacionalización.
• El universo y las unidades de análisis
• La pretensión de validez de un diseño. Validez, confiabilidad y generalizabilidad.
• El diseño como garantía de validez. Los tipos de validez: interna, externa, de constructo, de medida. Las amenazas a la validez.
• Los tipos de diseño: experimental, cuasiexperimental, no experimental
• Diseño cuantitativo, cualitativo y mixto
Horas de clase o similar: 32

Temas y/o Competencias:

Nombre de tema: Módulo 3: El proceso de investigación: la estrategia, el proyecto y las técnicas
Descripción:
¿Cuál es el ciclo de investigación desde la idea hasta el proceso, el resultado y su publicación?
¿Qué son las técnicas y cuáles son algunas de las más comúnmente utilizadas? ¿Cómo se escribe un perfil de investigación?

• El ciclo y el proceso de investigación: la noción de helicoide y la formalización en etapas.
• El planteo del problema, las preguntas y las hipótesis, los antecedentes, el marco teórico.

• La toma de decisiones respecto al tipo de diseño y los requisitos que se deben observar para su validez y confiabilidad.
• El tema de los datos, las fuentes y los relevamientos
• La observación, la encuesta y la entrevista
• El proceso de escritura de un perfil de investigación
Horas de clase o similar: 24

Temas y/o Competencias:

Nombre de tema:
Descripción:
Horas de clase o similar:

Información general

Nombre de la unidad curricular: Metodología de la investigación social
Carreras:

Área de conocimiento: Área Ciencias de la Salud
Subárea: Diseño y Urbanismo
Especialización:

Acepta cupo: No
Permite exoneración: Si
Porcentaje:
Nota: 6
Con examen
Ambos

Objetivos

Modalidad de enseñanza: Presencial

Objetivos de aprendizaje: 1. Identificar las diferencias entre conocimiento científico y otras formas de conocimiento y familiarizarse con distintas perspectivas acerca de cómo se genera el conocimiento: la deducción, la inducción y sus límites, la falsación.
2. Reflexionar acerca de las diferencias entre ciencias naturales y ciencias sociales y el rol de la vigilancia epistemológica.
3. Conocer las diferencias entre metodología, método y técnica en el proceso de investigación social.
4. Reconocer y formular el proceso lógico de un ciclo y de un proyecto de investigación social.
5. Distinguir las formas de inferencia lógica en el proceso de investigación; conocer las relaciones y funciones de las teorías y los métodos en el proceso de la investigación social.
6. Identificar y formular problemas de investigación. Formular hipótesis de investigación y comprender la relación entre las hipótesis, los marcos conceptuales y los tipos de diseño de investigación
7. Identificar dimensiones, variables e indicadores de distinto tipo y elaborar en base a lo identificado.
8. Conocer algunas técnicas de relevamiento en el marco de metodologías cuantitativas y cualitativas.
9. Distinguir fuentes de datos primarias y secundarias, unidades de análisis y poblaciones en la investigación y elaborar en base a lo identificado.
10. Identificar las diversas etapas y componentes de un proyecto de investigación y sus resultados, familiarizándose con la estructura de los proyectos de investigación en

Figura 49

Ejemplo de visualización de documento PDF resultado de exportar un curso.

²¹ <http://pdfmake.org/#/>

5.12 Comentarios

En el requerimiento de comentarios privados del [Anexo D.2](#) pág. 101 se estableció que la herramienta debe contar con una opción para ingresar comentarios privados (notas) en cualquier momento del proceso, solo visible dentro de la plataforma.

Se implementa esta funcionalidad mediante un icono de comentarios en la esquina superior derecha, como se puede ver en la Figura 50.

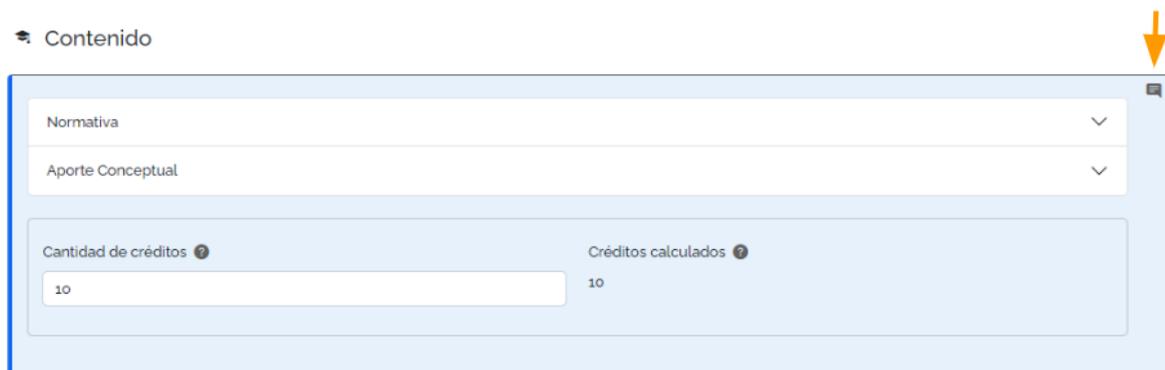


Figura 50
Botón para ver y agregar comentarios.

Cuando se hace clic en el icono, se abre una ventana emergente como la observada en la Figura 51, en la que se muestra una lista de comentarios previamente ingresados para ese atributo, con la fecha y nombre del usuario que lo agregó.

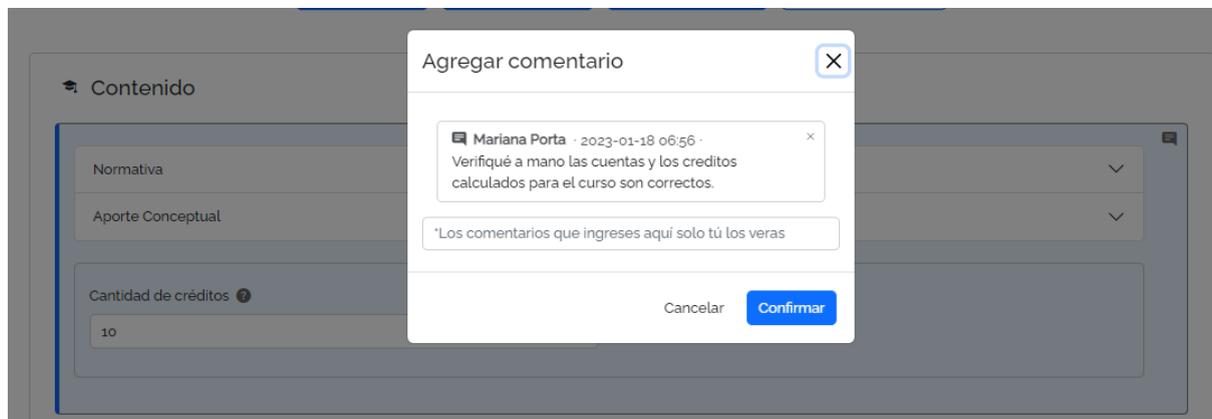


Figura 51
Popup para ver y agregar comentarios.

Los comentarios se almacenan en la base de datos. El objeto "atributo" contiene una lista de objetos "comentario", cada uno con campos para el mensaje, el autor y la fecha de creación.

Una decisión importante a tomar durante el desarrollo fue si exportar o no los comentarios al enviar un curso a otro usuario. Inicialmente, el equipo de desarrollo consideró que serían mensajes privados y por lo tanto, no debían ser exportados. Sin embargo, durante la presentación al usuario experto en el dominio, se decidió que sería más útil exportarlos ya

que podrían servir como información valiosa para otros usuarios, contribuyendo a convertir el software en una herramienta colaborativa.

5.13 Sitio web de presentación

Se ha creado un sitio web de acceso público con el fin de presentar el proyecto: <https://disenaturcurso.netlify.app/>. A través del sitio web que se observa en la Figura X, los usuarios pueden acceder a una descripción general de las funcionalidades del software. Además, desde allí se puede descargar el software en sistemas operativos como Windows, Linux y MacOS. El panel de descargas se puede observar en la Figura X.



Figura 52
Pantalla principal del sitio web.

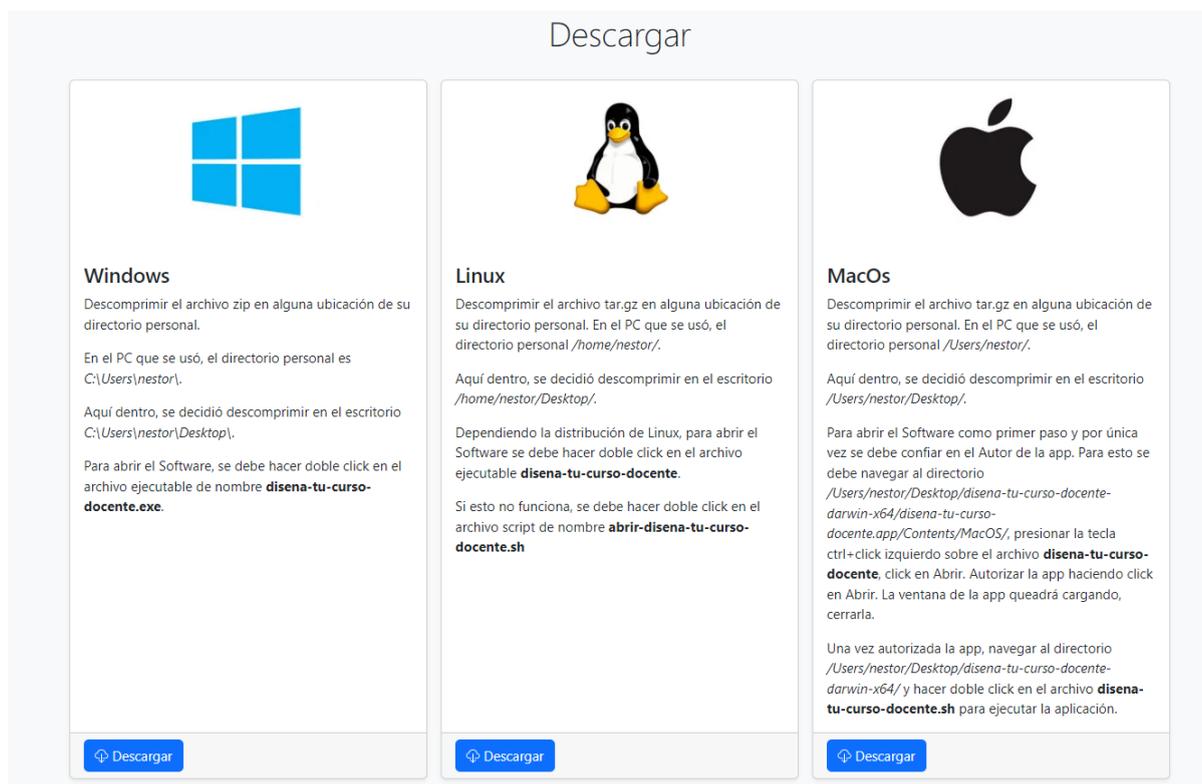


Figura 53
Pantalla de descargas del sitio web.

Antes de que los usuarios puedan descargar el software, se les solicita completar un formulario que incluye información como su nombre completo, correo electrónico, país e institución educativa. Una vez que el usuario ha proporcionado esta información, se le permite descargar el software. Es importante destacar que este formulario tiene como objetivo registrar a los usuarios que descargan y utilizan el software.

Este formulario es de gran utilidad, ya que permite mantener un registro de los usuarios que utilizan el software, lo que facilita la notificación de nuevas versiones o posibles actualizaciones. Además, permite al equipo de trabajo conocer mejor a su audiencia y adaptar el software a sus necesidades y preferencias.

Por otro lado, el sitio web cuenta con una sección dedicada a la presentación del equipo de trabajo responsable de la creación y desarrollo del proyecto. Esta sección, como se puede apreciar en la Figura 54, incluye información detallada acerca de los 3 integrantes y desarrolladores del software, así como de la tutora y el usuario experto en el dominio del proyecto.

Esta sección tiene como objetivo brindar transparencia y confianza a los usuarios al conocer al equipo de trabajo detrás del software, así como su experiencia y trayectoria en el ámbito de desarrollo de software.

Equipo

Somos un grupo de tres estudiantes avanzados de la carrera de Ingeniería en Computación de la Facultad de Ingeniería de la Universidad de la República (UDELAR), el Software Diseña Tu Curso forma parte del Proyecto de Grado de dicha carrera.

Para el diseño del software se lleva a cabo un proceso de co-diseño que implica trabajar junto con un Mariana Porta, nuestro usuario experto en el dominio para identificar problemas, necesidades y expectativas en el diseño de la herramienta.

Integrantes



Nestor Etcheverry

Estudiante
Licenciado en Computación



Sebastián Pelaez

Estudiante



Pablo Hernández

Estudiante



Regina Motz

Tutora
Doctora en Ciencias de la Computación



Mariana Porta

Usuario experto en el dominio
Socióloga, Máster en Educación

Figura 54

Pantalla del sitio web con información del equipo.

6. Evaluación de usabilidad

La identificación de errores de usabilidad es crucial para mejorar la experiencia del usuario y eliminar barreras. Por ello, se realizaron pruebas para evaluar la percepción de usabilidad y estética de la interfaz gráfica de un sitio. Se utilizó la herramienta VisAWI²² (Visual Aesthetics of Websites Inventory) para llevar a cabo la evaluación mediante una encuesta presentada a los usuarios después de haber utilizado la aplicación.

VisAWI es una herramienta o instrumento de evaluación que mide la estética visual de un software. Esta herramienta se compone de una serie de preguntas que se clasifican en categorías específicas, como Simplicidad, Colorido, Construcción y Diversidad, y permite evaluar la apariencia y la usabilidad de un sitio web. El objetivo es identificar fortalezas y debilidades en el diseño visual del sitio y mejorar su apariencia y experiencia de usuario.

La encuesta que se puede observar en el [Anexo E.1](#), se divide en seis secciones, las primeras cuatro corresponden a las descritas por VisAWI y contienen respuestas que van desde 'Muy en desacuerdo' hasta 'Muy de acuerdo', siendo la escala numérica de la siguiente manera: 1- Muy en desacuerdo, 2- En desacuerdo, 3 - Ni de acuerdo ni en desacuerdo, 4 - De acuerdo y 5 - Muy de acuerdo.

Las dos secciones finales están destinadas a preguntas que no están definidas en la herramienta VisAWI y que se refieren a funcionalidades específicas. Es interesante medir cómo los usuarios evaluaron estas funcionalidades. Además, se ha reservado espacio para que los usuarios puedan desarrollar sus respuestas libremente, sin limitarse únicamente a respuestas guiadas.

Las preguntas de las primeras cuatro secciones correspondientes a VisAWI se clasifican en cuatro categorías:

- Simplicidad: se refiere a la facilidad de uso del software.
- Colorido: se enfoca en el uso de los colores en la misma, buscando equilibrio para resaltar partes importantes sin sobrecargar la interfaz.
- Construcción: evalúa la calidad de la estructura del software, asegurándose de que tenga sentido.
- Diversidad: valora la originalidad del software.

El objetivo principal de utilizar VisAWI es lograr la aprobación de los usuarios desde un punto de vista estético. Para ello, se ha establecido un criterio de éxito: el promedio de todas las categorías debe ser al menos "de acuerdo", es decir, mayor o igual a 4 en la escala de clasificación previamente mencionada. Esto significa que los usuarios deben encontrar la estética visual del sitio atractiva y fácil de usar. Al lograr este objetivo, se espera una buena experiencia de los usuarios en el software y, por lo tanto, aumentar su satisfacción y fidelidad.

La documentación de VisAWI indica que es ideal obtener 20 respuestas, pero como esto es demasiado complejo, se decidió realizar la prueba con una muestra de 5 docentes invitados

²² <https://visawi.uid.com/>

por el usuario experto en el dominio para participar, estos docentes participaron en el curso “Rediseñando la Universidad Digital”. Dado que VisAWI solo está disponible en inglés o alemán, se proporcionó a los usuarios una traducción de cada pregunta para reducir la barrera del idioma.

6.1 Resultados

En las cuatro categorías evaluadas en VisAWI, se ha obtenido una puntuación superior a la media, que se estableció en 4 puntos como nivel de acuerdo. Esto significa que en todas las categorías evaluadas, el desempeño ha sido satisfactorio y ha superado las expectativas promedio.

Los resultados promedio de las respuestas se pueden observar en la Figura 55 y el resultado del promedio por categoría en la Figura 56. En ambos casos se pueden analizar en detalle en el [Anexo E.2](#).

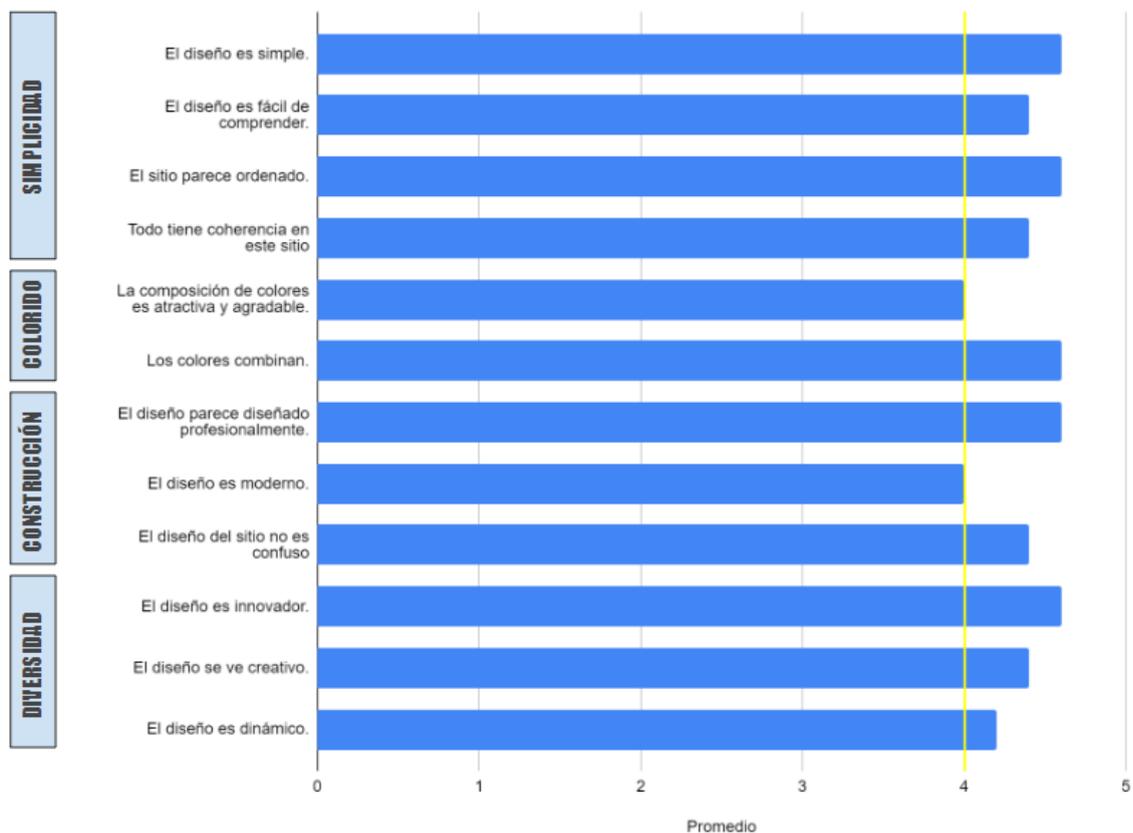


Figura 55
Promedio de respuestas por pregunta.

Promedio de respuestas por categoría

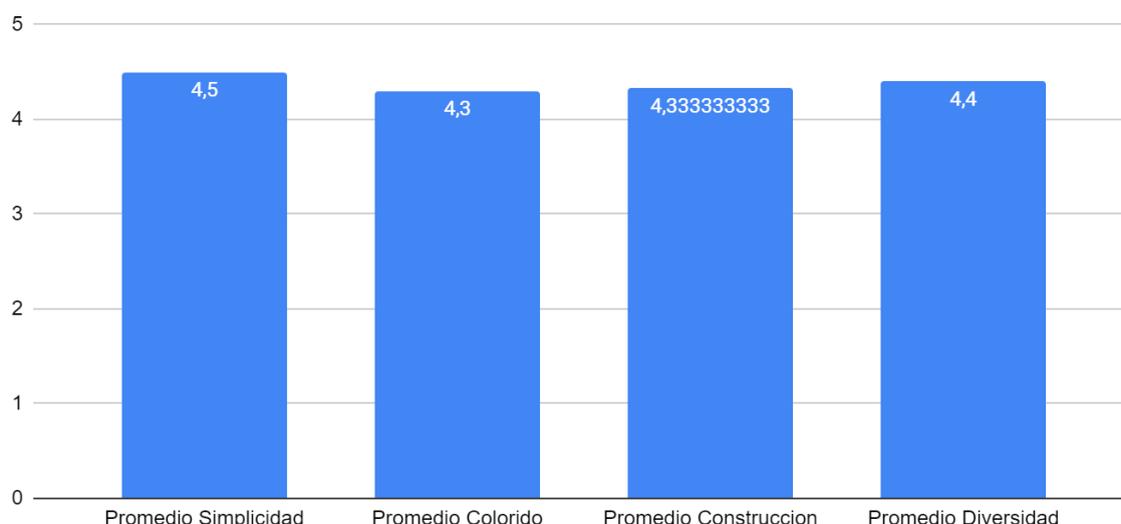


Figura 56
Promedio de respuestas por categoría.

Sin embargo, en cuanto a la funcionalidad del software, el promedio de respuestas a la pregunta “¿Cree que el software puede ser usado sin un manual de usuario?” fue 3,6, lo que está por debajo del valor de 4 que indica estar de acuerdo. Por lo tanto, sería recomendable contar con un manual de usuario o una guía para mejorar la experiencia de uso del software.

De igual manera, es importante destacar que la respuesta a la pregunta “¿Cómo evalúa la usabilidad en la etapa de Orquestación?” arrojó un promedio de 3,4, lo cual se encuentra por debajo del nivel de acuerdo aceptable. Esto sugiere que es necesario mejorar la usabilidad del software en esta etapa y considerar posibles alternativas para garantizar una experiencia de usuario más satisfactoria.

Cabe destacar que uno de los aspectos más positivos del estudio fue que todos los usuarios que probaron el software respondieron que volverían a usarlo para diseñar sus cursos. Esto indica que, a pesar de las posibles limitaciones mencionadas anteriormente, los usuarios encontraron el software lo suficientemente útil y satisfactorio para volver a utilizarlo en el futuro.

En este sentido, el hecho de que todos los usuarios estén dispuestos a volver a utilizar el software es una señal muy positiva y sugiere que el software tiene un gran potencial.

De la encuesta se desprende que las funcionalidades más destacadas del software son:

- Textos de ayuda claros y aportes conceptuales para ayudar en el diseño de cursos.
- Organización y relación entre diferentes aspectos del diseño.
- Autoguardado.
- Funcionalidad práctica para diseñar los contenidos del curso.
- Herramienta innovadora ya que no existe otra con estas características.
- Estructura práctica, posibilidad de importar contenidos y generar un documento con la información organizada.

6.2 Sugerencias

Las dos primeras sugerencias recibidas después de enviarles el software a los usuarios fueron sobre la configuración debido a las particularidades de sus computadoras. En primer lugar, nuestro software estaba compilado para una arquitectura x64, pero un usuario tenía una arquitectura x86, lo que provocó un error en la ejecución. De manera similar, otro usuario solo tenía un sistema operativo Linux, y hasta ese momento el software había sido exportado para Windows/x64. La solución en ambos casos fue exportar el software para que pudiera ejecutarse correctamente en la combinación sistema operativo/arquitectura de los usuarios.

Algunos usuarios enviaron por correo electrónico sugerencias y notas que tomaron durante las pruebas de usabilidad.

Se han recibido las siguientes sugerencias sobre el esquema del curso:

- En el apartado Tipo de la Evaluación, sería conveniente permitir la selección de más de una opción simultánea.
- Sería necesario mejorar la ayuda en el apartado Infraestructura del curso, ya que hay dudas sobre si se refiere a la infraestructura necesaria para el estudiante o para las clases.
- En el apartado Metodología de enseñanza, sería útil permitir la selección de más de una opción.
- En el apartado Temas y/o Competencias de Contenido en la etapa de Programa, es necesario mejorar la ayuda, ya que actualmente solo se refiere a las competencias y no queda claro si se refiere a los temas y/o competencias de las unidades o en general.
- En el apartado Estructura del Curso dentro de Dictado, la distribución de las Unidades resulta tediosa. Se sugiere la posibilidad de agregar imágenes para cargar esquemas del curso.
- Sería recomendable tener un espacio dentro de cada etapa con "Otras Informaciones".
- En el apartado Estructura del Curso, sería más intuitivo comenzar por la Orquestación y luego mostrar las Evaluaciones.
- Es necesario mejorar la información de ayuda en el apartado Bitácora del Curso de la etapa de Dictado.
- En la sección de Recursos, no queda claro lo que se espera en cuanto a elegir una opción de Licenciamiento de Creative Commons.
- En la sección de Contextualización, se sugiere incorporar un ítem sobre el perfil de los estudiantes o el perfil esperado, además del número.

Se recibieron las siguientes sugerencias adicionales del software en general:

- En cuanto al diseño, se sugiere incluir íconos o imágenes que aporten claridad, decoración e innovación.
- Sería conveniente permitir el intercambio de información entre el software y EVA para obtener información de un curso que ya se encuentra ingresado allí.

- Cuando se exporta un curso a formato PDF, el diseño lineal actual hace que se pierda la interconexión entre los elementos. Se sugiere la posibilidad de exportarlo con un diseño más gráfico y con el estilo de la interfaz para una mejor comprensión.
- Se sugiere incluir una guía de imágenes para la instalación.
- No se encontró un ítem sobre previas que se requiera para cursar formalmente, se sugiere incorporarlo.
- Se sugiere desarrollar la opción de elegir qué elementos incorporar en el informe impreso al exportar a PDF.

7. Conclusiones y Trabajos futuros

Durante el proyecto, el equipo enfrentó muchos desafíos que resultaron en una experiencia enriquecedora para todos los miembros. En general, se considera que la metodología de trabajo utilizada tuvo un impacto positivo en el desarrollo de la herramienta.

El proceso de trabajo con el usuario experto en el dominio fue intensivo y requirió una atención especial y una escucha activa para comprender sus expectativas con respecto a la herramienta tecnológica. La idea de cómo desarrollarla evolucionó a lo largo de varias etapas y cambios hasta alcanzar una solución satisfactoria para el usuario experto en el dominio. La maquetación fue fundamental en este proceso, ya que permitió una comprensión más clara de sus necesidades. Aunque este proceso tomó varias semanas, finalmente logramos un diseño que cumplía con sus expectativas.

En la fase de análisis se realizaron encuentros semanales con el usuario experto en el dominio y la tutora para intercambiar ideas. El usuario experto en el dominio presentó la idea del proyecto, qué la motivó y qué solución imagina. Además de proporcionar una herramienta tecnológica, este proyecto también realizó una evaluación del proceso de diseño de cursos y de las etapas que deben seguir los docentes para lograr sus objetivos.

Los encuentros se realizaron de forma virtual, en horarios a convenir y considerando los usos horarios diferentes dado que el usuario experto en el dominio en determinado momento en la vida de este proyecto se encuentra en España presentando su posgrado. Los encuentros se realizaron a través de la plataforma Zoom.

Entre reunión y reunión, se analizaron los puntos conversados, las problemáticas planteadas y los principales puntos o cambios generados a partir de la reunión anterior. Se generaron documentos, diagramas y maquetas que facilitaron el seguimiento de las reuniones.

Se ha desarrollado una herramienta tecnológica, (usamos el término herramienta tecnológica en lugar de software para alinearnos con la terminología más amigable desde el punto de vista docente) que ofrece apoyo a los docentes en el diseño de cursos de enseñanza superior. Este software cumple con los requerimientos funcionales y no funcionales establecidos con el usuario experto en el dominio. Los docentes pueden utilizar este software para diseñar sus cursos atendiendo las tres etapas de vida del curso: Programa, Contextualización y Dictado. La herramienta permite trabajar sin conexión a internet, Es personalizable a las características de los cursos de distintas instituciones educativas y permite exportar el diseño del curso en las distintas etapas para ser compartido con otros colegas. Además, este software se puede instalar en cualquier ordenador con sistema operativo Windows, Linux o MacOs. Es importante mencionar que la compatibilidad con MacOs no era un requisito obligatorio, pero se logró alcanzar como objetivo. El software cuenta con un esquema de curso genérico predeterminado, pero permite a los docentes agregar diferentes esquemas y trabajar juntos de forma colaborativa para diseñar los cursos.

En el desarrollo del software se implementó el concepto de puntos críticos del diseño. Los atributos críticos están bien diferenciados y se han implementado tres formas de desarrollar

la información. Esto permite al docente tener la posibilidad de ser acompañado y mejorar sus diseños a partir de esta información. Esto es muy importante ya que ayuda al docente a identificar los puntos críticos del diseño del curso y a tomar decisiones informadas para mejorarlo. Además, al tener varias formas de desarrollar la información, el docente tiene más opciones para adaptar el diseño a sus necesidades y preferencias.

El software desarrollado comparte varias funcionalidades con el entorno de trabajo ILDE presentado anteriormente, como la capacidad de diseñar cursos, colaborar con otros usuarios, adjuntar documentos y exportar el diseño. No obstante, nuestro software ofrece una interfaz de usuario más moderna y dinámica en comparación con ILDE, el cual presenta un área de trabajo en forma de cuadro de texto lo que hace que los diseños sean lineales. Además, en nuestro software, el diseño del curso se compone de varias secciones, lo que permite que el docente navegue y complete atributos que luego se incorporan en el diseño final. A diferencia de ILDE, cuyos diseños de curso son plantillas de texto que simplemente requieren que el docente complete espacios en blanco. Por otro lado, ILDE es una plataforma en línea, mientras que nuestro software debe ser instalado en la computadora del docente.

Durante el desarrollo del software, se encontraron algunas dificultades relacionadas con la complejidad de los cursos. Algunos cursos pueden tener esquemas muy elaborados con dependencias entre atributos y atributos dentro de otros atributos (como la orquestación), lo que hace que el esquema sea muy complicado y difícil de interpretar. Esto se reflejó en los resultados de la encuesta de usabilidad, donde la orquestación recibió un promedio de respuestas por debajo del aceptable. Para abordar este problema, se pueden considerar en un futuro mejoras en la forma de diseñar los cursos para hacer que el esquema sea más simple y fácil de interpretar. Esto puede incluir una mejor organización de los atributos o una interfaz de usuario mejorada para facilitar la navegación y la interpretación de los esquemas.

En el desarrollo del software, se subestimó el tiempo que llevaría desarrollar la orquestación, lo que causó trabajo extra y retraso en los tiempos del proyecto. La causa de este retraso fue por la subestimación de la complejidad de la funcionalidad.

Dentro de los riesgos materializados y difíciles de superar, se destaca la falta de datos reales de cursos. Estos datos podrían haber sido útiles para tener una mejor comprensión de los datos a ser ingresados en la herramienta. Sin embargo, esto se mitigó con una comunicación efectiva con la cliente, quien proporcionó la información más precisa posible.

En cuanto al requerimiento de recolección de datos, se alcanzó el segundo nivel, denominado el nivel básico. Este nivel se caracteriza por la capacidad de importar documentos generados en el curso en diferentes formatos, como PDF, Excel, CSV, entre otros. Esto permite a los docentes importar y utilizar información relevante para el diseño del curso, como planificaciones y materiales didácticos, y hacer uso de esta información en el proceso de diseño. El nivel básico de recolección de datos permite alcanzar un cierto grado de eficacia en el diseño de cursos, pero no es tan avanzado como el tercer nivel, el nivel avanzado, que incluye herramientas de automatización y sistematización de la recolección de datos, permitiendo una mayor eficacia en el diseño del curso.

Otra característica a resaltar en la herramienta es que permite realizar versiones de cursos. Esto significa que el docente tiene la capacidad de crear una nueva versión del curso en cualquier momento utilizando un botón específico. El software también proporciona una lista de las versiones anteriores del curso, permitiendo al docente visualizar todas las versiones disponibles y restaurar una versión anterior si lo desea. Esta característica es muy útil para garantizar que el docente pueda seguir el progreso del curso y volver a versiones anteriores si es necesario. Además, también ayuda a mantener una trazabilidad de las modificaciones del curso.

Por otro lado, se deseaba desarrollar el servidor central y el cliente para el usuario Administrador, pero quedó fuera del alcance del proyecto por razones de cronograma. Ambos sistemas agregarían valor al software que se desarrolló, ya que el cliente Administrador es el responsable de la gestión de los datos y los esquemas de cursos, mientras que el servidor central es el encargado de centralizar el almacenamiento de los cursos y su gestión. Desarrollar ambos sistemas podría haber requerido una cantidad de tiempo y recursos significativamente mayor al que se disponía para el proyecto.

A continuación se presentan los trabajos futuros:

- Un sistema cliente para que el usuario Administrador pueda crear y editar los esquemas de los cursos en su institución. El sistema permitirá al administrador gestionar los atributos que se mostrarán en la pantalla y ver su evolución en una interfaz en tiempo real.
- Mejorar la estructura de la orquestación para hacerla más intuitiva para el docente y más fácil de procesar por el software.
- El servidor central es responsable de varias funcionalidades. Una de ellas es permitir subir cursos. Para lograr esto, es necesario desarrollar un conjunto de instrucciones o algoritmos en el servidor central que permitan combinar o unir información de varios cursos en uno solo, a medida que los usuarios (docentes) van enviando sus cursos. Esto asegurará que los cursos estén disponibles en un solo lugar y facilita el manejo y actualización de la información. Es importante que el sistema permita una actualización automática y segura de estos cursos, para evitar problemas de consistencia de la información.

Por otro lado, del servidor central también depende la capacidad de crear el inicio de sesión con las instituciones y manejar todos los usuarios. Además, este servidor debe permitir asignar una licencia Creative Commons (CC) de derechos de autor para un grupo específico de usuarios registrados en el sistema.

En el servidor central también se podrá tener un repositorio de los cursos, es decir, el curso se va a almacenar tanto en la computadora del docente como en el servidor. Esto implica que la información sea consistente entre ambas ubicaciones y asegurará que el curso esté disponible en todo momento sin discrepancias en la información. Al tener un repositorio centralizado, se facilita la gestión y la actualización de los cursos. Es importante que se implementen medidas de seguridad y respaldo para evitar la pérdida de información en caso de fallas en el sistema o en la computadora del docente.

Actualmente, cuando se crea un nuevo curso, se le asigna un identificador único dentro del conjunto de cursos locales en la PC en uso. Estos identificadores deben ser corregidos antes de subir el curso al servidor central para garantizar la consistencia y la unicidad a nivel global.

Para el manejo de los documentos y archivos adjuntos de un curso en el servidor, como los documentos de bibliografía, implica permitir que los docentes suban estos archivos al servidor. Además, es importante que se implemente un mecanismo para evitar duplicados cuando varios docentes suban el mismo documento. Esto asegurará que solo exista una versión del documento y evitará confusiones en la información.

- Desarrollar la capacidad de enviar alertas a los docentes cuando se modifica el esquema de un curso. Se debe tener en cuenta la posibilidad de que un docente pueda optar por recibir estas alertas o no mediante una configuración en su perfil.

Esta funcionalidad es importante porque permite a los docentes estar al tanto de los cambios en los esquemas de los cursos de manera instantánea.

- Desarrollar las funcionalidades para generar estadísticas y utilizar inteligencia artificial para enriquecer los análisis con los datos recolectados del curso. Esto puede incluir análisis de rendimiento de los estudiantes, patrones en el uso de los recursos educativos, tendencias en el rendimiento de los cursos, entre otros.

Esta funcionalidad es importante porque permite a los docentes y administradores tener una mejor comprensión del rendimiento de los estudiantes y de los cursos. Además, también puede ayudar a identificar problemas y oportunidades para mejorar la calidad de la educación y adaptarse a las necesidades de los estudiantes.

Con respecto al sitio web que presenta el proyecto, es importante destacar que cumple con su objetivo principal de servir como una página de aterrizaje para que los usuarios puedan descargar el software. Queda pendiente implementar un formulario de registro y validación en dos pasos de los usuarios antes de la descarga. Este formulario adicional permitiría agregar mayor seguridad y trazabilidad al proceso de registro de usuarios, lo que garantiza que se tenga un registro fiable de quienes han descargado el software. Además, el proceso de validación en dos pasos brindaría una capa adicional de protección a los usuarios, lo que les permitiría sentirse más seguros al proporcionar información personal en el formulario. Este proceso adicional al registro requiere de un servidor, lógica de backend y una base de datos para almacenar la información.

En lo que respecta al ámbito personal de los integrantes del equipo, consideramos que ha sido una experiencia positiva. Nos brinda una gran satisfacción haber trabajado sobre este proyecto en particular y haber logrado demostrar y plasmar los conocimientos adquiridos durante todos estos años durante el transcurso de la carrera.

El software presentado tiene un impacto significativo y busca mejorar el proceso de diseño de un curso para los docentes, con desafíos tanto tecnológicos como de integración.

8. Referencias

Arnaz, J. A. (1981). La planeación curricular. (segunda edición). Editorial Trillas. México.

Barriga, F. D. (2005). Principios de diseño instruccional de entornos de aprendizaje apoyados con TIC: un marco de referencia sociocultural y situado. *Revista Tecnología y comunicación educativa*, 41, 5-16. [En línea] (Consultado el 31 de marzo de 2023) Recuperado de <http://investigacion.ilce.edu.mx/tyce/41/art1.pdf>

Belloch, C. (2012). Diseño Instruccional. [En línea] (Consultado el 31 de marzo de 2023) Recuperado de <http://www.uv.es/~bellochc/pedagogia/EVA4.pdf>

Caeiro, M., & Mendoza, M. (2014). Towards an open lesson plan data model. *Ingeniería E Innovación*, 2(2), 2014. [En línea] (Consultado el 31 de marzo de 2023) Recuperado de <https://doi.org/10.21897/23460466.783>

Caeiro-Rodríguez, M., Llamas-Nistal, M., Fernández-Iglesias, M., Mikic-Fonte, F., & Lama-Peñín, M. (2015). Supporting real open educational resources in Edu-AREA: Different views about open educational resources. En 2015 IEEE Frontiers in Education Conference (FIE), (pp. 1-8). El Paso, TX, USA. [En línea] (Consultado el 31 de marzo de 2023) Recuperado de <https://doi.org/10.1109/FIE.2015.7344338>

Michos, K., & Hernández-Leo, D. (2018). Supporting awareness in communities of learning design practice. *Computers in Human Behavior*, 85, 255-270. [En línea] (Consultado el 31 de marzo de 2023) Recuperado de <https://doi.org/10.1016/j.chb.2018.04.008>

Paquette, G. (2012). Technology-Based Instructional Design - Evolution and Major Trends. En N. M. Seel (Ed.), *Encyclopedia of the Sciences of Learning* (pp. 3298-3302). [En línea] (Consultado el 31 de marzo de 2023) Recuperado de https://www.researchgate.net/profile/Gilbert-Paquette-2/publication/245165034_Chapter_56_Technology-Based_Instructional_Design_-_Evolution_and_Major_Trends/links/0046351d6dad5daf2000000/Chapter-56-Technology-Based-Instructional-Design-Evolution-and-Major-Trends.pdf

Porta, M., Casnati, A., Solana, V., & Marrero, C. (2021). Rediseñando la Universidad Digital. [En línea] (Consultado el 31 de marzo de 2023) Recuperado de <https://proeva.udelar.edu.uy/curso-redisenando-la-universidad-digital>

Porta, M., Motz, R., & De Queiroz Lopes, D. (2022). Higher education teachers: problematization of post-pandemic course design. En *International Symposium on Computers in Education (SIIE)*, Coimbra, Portugal, pp. 1-5, doi: 10.1109/SIIE56031.2022.9982347.

Porta, M., Rodés, V., Garófalo, L., Czwerwonogora, A., Rodríguez, C. & Marrero, C. (2022). Rediseñar la Universidad Digital: aperturas reflexivas. *TIES Revista de Tecnología e Innovación en Educación Superior*, v.: 6 p.:2 - 18, 2022 Teacher.

Sommerville, I. (2004). *Software Engineering 7a. ed.* Pearson Education.

9. Glosario

Definición de punto crítico

Durante el proceso de uso de la herramienta, se presentan puntos críticos en los que es crucial que la información brindada al docente sea adecuada para lograr una definición clara de los resultados. En estos momentos, se pueden sugerir herramientas externas al sistema para mejorar la calidad de la información proporcionada. Es responsabilidad del administrador de la herramienta definir estos puntos críticos y brindar al docente un aporte conceptual, información ampliatoria y/o normativa necesaria para el éxito del proceso.

- **Aporte conceptual:** Un breve texto en el cual se explica un concepto de manera acotada y breve. Breve texto que presenta el material del enlace, seguido de un enlace a un sitio que amplía el concepto explicado.
- **Normativa:** una nota que recuerda normativas o requisitos institucionales que vienen al caso. Está seguido de un breve texto que dice el nombre del documento o referencia y el enlace al mismo.
- **Información ampliatoria:** Tema novedoso y dinámico (posibilidad de aplicarlo con un sistema de recomendación). Es dinámico.

Definición de Atributo

Un Atributo indica un espacio en el formulario de entrada donde el sistema despliega una pregunta y un Docente puede ingresar una respuesta. Los atributos pueden ser Puntos Críticos, Requeridos o Opcionales, Modificables o No Modificables.

Derechos de autor Creative Commons (CC)

Es una organización sin fines de lucro dedicada a promover el acceso y el intercambio de cultura. Desarrolla un conjunto de instrumentos jurídicos de carácter gratuito que facilitan usar y compartir tanto la creatividad como el conocimiento.

Las licencias Creative Commons no reemplazan a los derechos de autor, sino que se apoyan en estos para permitir elegir los términos y condiciones de la licencia de una obra de la manera que mejor satisfaga a quien es titular de los derechos. Por tal motivo, estas licencias se han interpretado como una forma de tomar el control para compartir la propiedad intelectual.

Product Backlog

Consiste en la elaboración de un listado de todas aquellas tareas que queremos realizar durante el desarrollo de un proyecto con el objetivo de que estas sean visibles para todo el equipo.²³

Design System

²³ <https://www.scrum.org/resources/scrum-guide>

Es un conjunto de patrones interconectados y prácticas compartidas coherentemente organizadas. Ayudan en el diseño de productos digitales y el desarrollo de productos como aplicaciones o sitios web. Pueden contener, entre otros, bibliotecas de patrones, lenguajes de diseño, guías de estilo, componentes codificados, lenguajes de marca y documentación.

Framework

Un framework es un marco o esquema de trabajo generalmente utilizado por programadores para realizar el desarrollo de software. Utilizar un framework permite agilizar los procesos de desarrollo ya que evita tener que escribir código de forma repetitiva, asegura unas buenas prácticas y la consistencia del código.

Es un conjunto de herramientas y módulos que pueden ser reutilizados para varios proyectos. Uno de los frameworks más conocidos y utilizados es Angular, un framework de código abierto desarrollado en TypeScript y mantenido por Google.

Frontend

Se refiere a la parte visible de una aplicación o sitio web que los usuarios pueden interactuar directamente. Incluye todo lo que un usuario ve, toca y experimenta en su pantalla, como el diseño, la navegación, los botones, los formularios, las animaciones y los efectos visuales.²⁴

HTML (Hypertext Markup Language)

Hace referencia al lenguaje de marcas de hipertexto para la elaboración de páginas web. Mediante marcas el lenguaje describe la estructura de la página web. Los elementos del lenguaje etiquetan partes del contenido como tablas, listas, párrafos, entre otros.²⁵

CSS (Cascading Style Sheets)

Es un lenguaje utilizado para la descripción de la presentación de una página web, incluye los colores a mostrar, la letra, el diseño. Permite generar una presentación que se adapte a distintos tamaños de dispositivos de visualización, con pantallas grandes o pequeñas.

JavaScript

JavaScript²⁶ es un lenguaje de programación interpretado orientado a objetos, que sigue el estándar ECMAScript. Comúnmente es empleado en los sitios web para realizar acciones en el lado del cliente, pero también puede ser utilizado como lenguaje del lado del servidor.

HTTP (Hypertext Transfer Protocol)

²⁴ https://developer.mozilla.org/en-US/docs/Learn/Front-end_web_developer

²⁵ <https://www.w3.org/standards/webdesign/htmlcss>

²⁶ <https://developer.mozilla.org/es/docs/conflicting/Web/JavaScript>

Protocolo de red utilizado para transferir información en la World Wide Web. Es un estándar para la transmisión de información en la web, que permite la comunicación entre el navegador y el servidor web para mostrar páginas web en la web.²⁷

HttpClient

Es una clase en .NET que se utiliza para realizar solicitudes HTTP y recibir respuestas en aplicaciones .NET, incluyendo aplicaciones de escritorio, móviles y de servidor. HttpClient permite enviar solicitudes HTTP, como GET, POST, PUT y DELETE, y recibir respuestas HTTP en forma de objetos HttpResponseMessage. HttpClient es una manera fácil y conveniente de realizar llamadas HTTP en aplicaciones .NET.²⁸

API

La abreviatura de API viene del término en inglés Application Programming Interface (Interfaz de programación de aplicaciones). Las API son un conjunto de comandos, funciones, protocolos y objetos que los programadores pueden utilizar abstrayéndose de la implementación en sí, para desarrollar software o interactuar con un sistema externo.²⁹

²⁷ <https://developer.mozilla.org/es/docs/Web/HTTP>

²⁸ <https://learn.microsoft.com/en-us/dotnet/api/system.net.http.httpclient?view=net-6.0>

²⁹ <https://techterms.com/definition/api>

A. Anexo Formularios de Programa

A.1 Formulario de nuevo programa FCS 2022

Programa de
Nombre de la Actividad curricular (AC)
Plan 2009
Licenciatura
Ciclo (Inicial o Avanzado)

1. **Docentes**

2. **Créditos:**

3. **Régimen de cursado:**

4. **Carga y distribución de horas estimada:**

Actividad		Hs. estimadas
Con supervisión docente presencial	Horas presenciales aula	
	Aula virtual con presencia docente	
	Otros (Especificar)	
Sin supervisión docente presencial	Estudio autónomo	
	Tarea consignada por el equipo docente fuera de horario presencial (grupal, individual, actividades EVA)	
	Trabajo de campo	
	Trabajos finales fuera del itinerario presencial	
	Otros (Especificar)	
Horas totales de la actividad curricular		

5. **Conocimientos previos recomendados**

6. Objetivos de enseñanza y aporte al módulo que integra la UC

7. Contenidos y organización del curso:

8. Método de enseñanza. Marcar los métodos a utilizar en el curso y describir cómo se organizan en el semestre.

Aprendizaje basado en problemas	
Proyectos	
Exposición	
Debate/Coloquio	
Prácticas/Laboratorios (demostración, aplicación, resolución de ejercicios y problemas)	
Talleres	
Seminarios	
Tutorías	
Salidas de campo	
Otros métodos	

- El curso distingue entre grupos teóricos y grupos prácticos - (SI o NO)

Descripción:

--

9. Sistema de evaluación. Marcar las que se prevea utilizar y describir

Actividad	Peso relativo	Descripción (formativa, control de lectura, etc)
Evaluación presencial		

Ev. domiciliaria individual		
Ev. domiciliaria grupal		
Presentaciones/participación en aula		
Informes/trabajo final/Monografía		
Tareas en EVA		
Otras actividades (describir)		

Para la aprobación del curso se requiere:

10. Bibliografía

Obligatoria

Ampliatoria

A.2 Formulario Tercer semestre del Ciclo Inicial de la Facultad de Ciencias Sociales

Programa
Unidad curricular
Edición
Plan

3º semestre del Ciclo Inicial de la Facultad de Ciencias Sociales

Ofrecido también para el__

Docentes:

Docente del Teórico:

Docentes de prácticos:

1. Créditos:

a) Carga horaria:

b) Modalidad de cursado

c) Modalidad de enseñanza

2. Conocimientos previos recomendados

3. Objetivos

4. Contenidos

5. Método de trabajo

6. Sistema de evaluación

7. Bibliografía

A.3 Formulario Humanidades

Nombre de la asignatura:

Edición:

Plan:

Ciclo:

Equipo docente:

	Cargo	Nombre	PDU/Departamento /Sección

Programa unidad curricular (UC)

Créditos

El total de Créditos corresponde a:

Carga horaria presencial	
Trabajos domiciliarios	
Plataforma EVA	
Trabajos de campo	
Monografía	
Otros (describir)	

Asistencia obligatoria:

Permite exoneración: SI nota de aprobación mínima: Otros:

Número de períodos de examen: 4

Conocimientos requeridos

Previas reglamentarias:

Previas sugeridas:

Fundamentación

Objetivos:

Contenido o programa del curso o actividad curricular

Explicitar contenido sintético:

Explicitar contenido desagregado:

Bibliografía obligatoria

regional e independencia del Uruguay, 2a ed.,

Bibliografía opcional: cada módulo tendrá su propia bibliografía opcional.

Modalidad de enseñanza

Modalidad de cursado a emplear

Teórico.

Práctico.

Teórico-práctico.

Taller.

Seminario.

Multimodal

Presencial

Semipresencial

Otro.

B. Anexo Fase de análisis - Documento de conceptos

Existen tres etapas en el diseño:

1. Elaboración del programa

Es el proceso en que el docente crea un curso que podrá ser parte de una carrera de grado o trayectoria educativa, o podrá ser un curso de Educación Permanente o de posgrado. El producto final del proceso es un documento: El Programa.

Según el Tesoro de UNESCO, la *elaboración del programa educativo* es el proceso sistemático de producción de nuevos programas educativos o de revisión de programas existentes. Implica la definición de objetivos, contenidos, métodos y procedimientos de evaluación.

Referencia: <https://skos.um.es/unescothes/C00935/html>

Se dividen la información a ingresar en grupos:

Grupo 1: Contenido

1. Créditos de la materia - Si sabe los créditos los ingresa
 - a. Opcional
 - b. Punto Crítico - Tesoro de UNESCO, Curso con créditos: Curso de enseñanza superior medido por un sistema de unidades según el cual a cada asignatura le corresponden más o menos unidades en función del número de horas lectivas.
<https://skos.um.es/unescothes/C00835/html>. Distribución de créditos en horas de “clase o actividad equivalente” o “estudio personal”.
 - i. Normativa: Este punto requiere un recordatorio sobre cómo se cuentan los créditos de acuerdo a UDELAR. Debe agregar enlace al documento con la normativa que es la Ordenanza de grado.
<https://dgjuridica.udelar.edu.uy/215-ordenanza-de-estudios-de-grado-y-otros-programas-de-formacion-terciaria/>
2. Créditos calculados - Calculado en base a la definición de crédito usada y los contenidos del curso ingresados
 - a. Modificable: NO
3. Contenido del curso - 3 campos, Contenido / Tema / Cantidad de horas.

Grupo 2: Información general

1. Nombre de unidad curricular

- a. Modificable a futuro: NO
- 2. Carrera - Es una lista de Select.
- 3. Área de conocimiento - Es una lista. Son 3 campos, 2 de tipo Select y 1 texto libre. Área de Conocimiento, Subárea, Especialización.
- 4. Cupo de estudiantes - Si/No, si selecciono Si -> Cantidad.
 - a. Opcional
- 5. Forma de aprobación
 - a. Permite exoneración - Si/No
 - b. Porcentaje de aprobación
 - c. Nota de aprobación
 - d. Examen - Radio Con/Sin, cada opción tiene su conjunto de opciones
 - i. Con: reglamentado, libre, ambos
 - ii. Sin: trabajo final, promedio de calificaciones

Grupo 3: Recursos

- 1. Bibliografía - Es una lista. Son 2 campos, Textos libre y/o lista de archivos asociado a una lista de Contenidos (los ingresados en el Grupo 1).
- 2. Recursos Educativos
 - a. Punto Crítico
 - i. Aporte Conceptual: Qué es un recurso educativo y qué es un recurso educativo abierto.
 - ii. Información ampliatoria:
 Enlace a este sitio de UDELAR
[https://proeva.udelar.edu.uy/recursos-abiertos/#:~:text=Los%20Recursos%20Educativos%20Abiertos%20\(REA,pueden%20encontrar%20en%20dominio%20p%C3%BAblico](https://proeva.udelar.edu.uy/recursos-abiertos/#:~:text=Los%20Recursos%20Educativos%20Abiertos%20(REA,pueden%20encontrar%20en%20dominio%20p%C3%BAblico)
 Otro enlace <https://es.unesco.org/themes/tic-educacion/rea>
 Enlaces a repositorios de recursos educativos abiertos
<https://recursosinnova.uniovi.es/buscadores>
 Enlaces a herramientas para crear recursos educativos
<https://recursosinnova.uniovi.es/herramientas/contenidos>
- 3. Licencia CC - Select con opciones
 - a. Punto Crítico
 - i. Normativa: Requiere aporte conceptual sobre licencia Creative Commons.
<https://es.unesco.org/open-access/las-licencias-creative-commons/#:~:text=Las%20licencias%20Creative%20Commons%20son,utilizar%20y%20distribuir%20un%20contenido>

Grupo 4: Objetivos

1. Modalidad de enseñanza - Select con opciones presencial, online, híbrido.
2. Objetivos de aprendizaje - Texto libre
 - a. Punto Crítico
 - i. Aporte Conceptual: Requiere un aporte conceptual sobre cómo se debe formular un objetivo de aprendizaje.
<https://eduteka.icesi.edu.co/articulos/TaxonomiaBloomDigital>
Se explicitan con verbos que corresponden a acciones que los estudiantes deberán demostrar. Debe explicitar el tipo de conocimiento al que el estudiante debe acceder: Ej. identificar, describir, comprender, aplicar, analizar, crear, etc. Los objetivos deben ser específicos y medibles. Deben expresarse desde el punto de vista del estudiante.
Ejemplos:
 - 1) Identificar las diferencias entre conocimiento científico y otras formas de conocimiento.
 - 2) Formular un problema de investigación
 - 3) Identificar dimensiones, variables e indicadores de distinto tipo. Los docentes suelen tener inconvenientes en su definición, por tanto es necesaria una guía detallada para su correcto planteamiento.
3. Marco pedagógico - Texto libre. En la siguiente etapa será parte de Metodología de enseñanza - Opcional

2. Instanciación o Contextualización

Ajuste del diseño para la cursada.

En esta etapa existen algunas definiciones que estarán dadas por las circunstancias de la cursada y otras que estarán definidas por el/la docente. Una vez el curso se ofrece se deberán tener los siguientes datos:

Grupo 1: Grupo Humano

1. Número de estudiantes
 - a. Cantidad
2. Composición del equipo docente - Lista de elementos
 - a. Nombre - Grado
3. Ratio estudiantes/docente: calculado automáticamente

Grupo 2: Infraestructura

1. Espacios - Se pueden agregar varias
 - a. Capacidad
 - b. Tipo: Select con Laboratorio/Aula
 - c. Protocolo de distanciamiento: Si/No
 - d. Equipamiento:

- i. Cámaras
 - ii. Amplificadores
 - iii. Pizarras electronicas
- 2. Equipamiento de préstamo a estudiantes
 - a. Select de Equipo
 - b. Cantidad
- 3. Asistencia técnica - Texto libre. Que se sepa a quién recurrir en casos de necesitar asistencia técnica.
- 4. Plataformas disponibles - Select con opciones
 - a. Eva
 - b. Zoom
 - c. Jitsi meet
 - d. BBB

Grupo 3: Recursos

- 1. Redefinir recursos educativos - Traerse todos los campos de la Etapa Diseño, Grupo Recursos, excepto el de Licenciamiento CC.
 - a. Obligatorio
 - b. Punto Crítico
 - i. Aporte Conceptual - Requiere aporte conceptual sobre curaduría de recursos educativos
 - ii. Información Ampliatoria - <https://www.educ.ar/recursos/119782/curar-contenidos-educativos>

Grupo 4: Estructura del curso

- 1. Duración: fecha inicio, fecha fin. Calcula la duración en función de las fechas.
- 2. Evaluaciones - Lista de Select con las opciones diagnóstica, formativa, sumativa, continua. Herramientas - Texto libre.
 - a. Punto Crítico
 - i. Aporte conceptual - Requiere aporte conceptual sobre cuáles tipos de evaluación existen. Se concibe evaluar como un proceso continuo, sistemático y flexible por el cual se emiten juicios de valor.
 - ii. Información ampliatoria - <https://eva.udelar.edu.uy/course/view.php?id=619§ion=6>

3. Orquestación: Definición (estructura del curso) - Es la orquestación de los contenidos y los espacios, se debe considerar cómo se distribuyen los contenidos, las actividades y las diferentes instancias educativas entre los entornos físicos y digitales y a través del tiempo. La estructura debe asignar espacios y tiempos al contenido desagregado. En la práctica es, a cada punto del contenido desagregado, asignarle tiempo en horas, en que fechas se dicta, en que entorno (digital o físico), tiene actividades asociadas, tiene instancias educativas, etc.

A cada Tema ingresado en la etapa anterior, se le pueden agregar elementos que pueden ser de tipo Actividad o Evaluación. Si se selecciona Evaluación, se puede seleccionar una evaluación de las ingresadas hasta el momento. Si se selecciona Actividad, se despliega un Select con las opciones Teórico, Práctico. Luego, se le asigna una fecha inicio - fecha fin, y una duración en horas. En la Figura 57 se representa un diagrama de alto nivel, a modo de ejemplo de la orquestación.

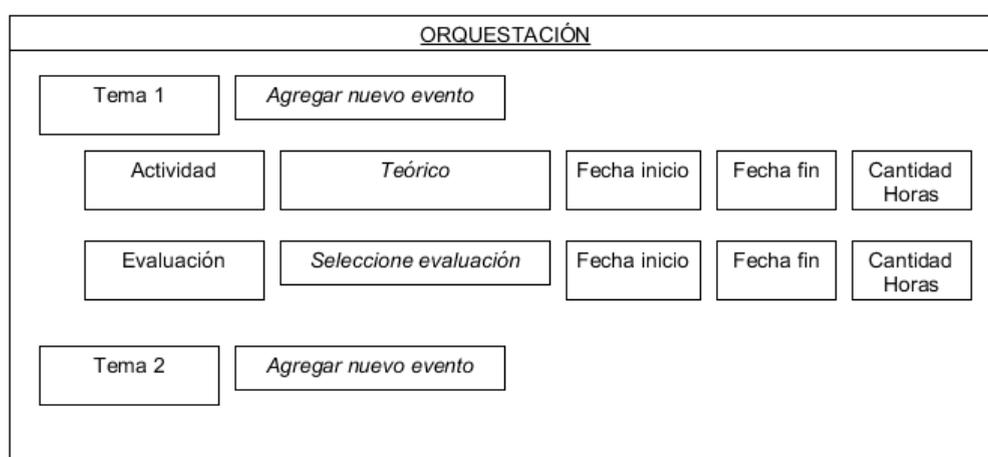


Figura 57
Diagrama de alto nivel de orquestación.

a. Punto Crítico

- i. Aporte Conceptual - La estructura del curso debe considerar los tiempos y espacios de trabajo con los que se cuenta y debe asegurar que se cumplen con los requisitos para asignar los créditos que otorga.
- ii. Información ampliatoria - <https://cead.pressbooks.com/chapter/11-9-paso-7-disenar-la-e-estructura-del-curso-y-las-actividades-de-aprendizaje/>
- iii. Normativa - Este punto requiere un recordatorio sobre cómo se cuentan los créditos de acuerdo a UDELAR. Debe agregar enlace al documento con la normativa que es la Ordenanza de grado. <https://dgjuridica.udelar.edu.uy/215-ordenanza-de-estudios-de-grado-y-otros-programas-de-formacion-terciaria/>

Grupo 5: Objetivos

1. Metodología de enseñanza - Select con opciones: Conferencia, MOOC, Aprendizaje Basado en Proyectos (ABP), Aula invertida, Aprendizaje

cooperativo, Gamificación, Pensamiento de diseño, Método de caso, Aprendizaje basado en problemas. Otro campo con texto libre

a. Obligatorio

b. Punto Crítico

i. Aporte conceptual: Requiere aporte conceptual sobre qué son los métodos de enseñanza.

ii. Información ampliatoria: Los métodos se refieren al “cómo” enseñar. Tienen abordajes diferentes según las teorías de aprendizaje que los fundamentan y según cómo se aborda el objeto de aprendizaje. Existen métodos más centrados en el docente y métodos más centrados en el estudiante. Existen métodos muy estructurados y otros más flexibles. Definir la metodología implica Seleccionar entre métodos conocidos, proponer actividades, seleccionar técnicas, Priorizar tipos de evaluación.

Aula invertida -

https://www.grupoeducar.cl/material_de_apoyo/funciona-clase-invertida/

Mooc -

<https://www.uab.cat/web/estudios/mooc/-que-es-un-curso-mooc-1345668281247.html>

2. Traer los 4 elementos de la Etapa Diseño, Grupo Objetivos. Puede modificarlos.

1. Estructura del curso: Es la orquestación de los contenidos y los espacios, se debe considerar cómo se distribuyen los contenidos, las actividades y las diferentes instancias educativas entre los entornos físicos y digitales y a través del tiempo. La estructura debe asignar espacios y tiempos al contenido desagregado.

En la práctica es, a cada punto del contenido desagregado, asignarle tiempo en horas, en que fechas se dicta, en que entorno (digital o físico), tiene actividades asociadas, tiene instancias educativas, etc.

Punto Crítico:

- Aporte Conceptual: La estructura del curso debe considerar los tiempos y espacios de trabajo con los que se cuenta y debe asegurar que se cumplen con los requisitos para asignar los créditos que otorga.

- Información ampliatoria:

<https://cead.pressbooks.com/chapter/11-9-paso-7-disenar-la-estructura-del-curso-y-las-actividades-de-aprendizaje/>

- Normativa: Este punto requiere un recordatorio sobre cómo se cuentan los créditos de acuerdo a UDELAR. Debe agregar enlace al documento con la normativa que es la Ordenanza de grado.

<https://dgjuridica.udelar.edu.uy/215-ordenanza-de-estudios-de-grado-y-otros-programas-de-formacion-terciaria/>

Alerta: Cada crédito son 15 horas, si las horas ingresadas en este campo son menores a las necesarias (definidas en 1.4 créditos), debe aparecer alerta.

3. Implementación - Dictado

Definiciones de diseño “on the spot” en la edición: implica ajustes que se toman sobre la marcha de acuerdo a decisiones del docente basadas en cómo el curso se desenvuelve y cómo los estudiantes responden. Se registran los cambios hechos.

Grupo 1: Comentarios

Texto libre

Grupo 2: Estructura del curso

Traer lo definido en este grupo de la etapa anterior

Grupo 3: Recursos

Traer lo definido en este grupo de la etapa anterior

C. Anexo Sprints

El software final se divide en 3 versiones, cada una correspondiente a una etapa del diseño del curso. Esta división permite entregar al usuario experto en el dominio un software funcionando en cada etapa del desarrollo. De esta manera, el usuario experto en el dominio recibe incrementos del software en cada entrega, que corresponden a las etapas del diseño.

El sprint es el ciclo de desarrollo del proyecto. Se definen 7 sprints para el desarrollo de software. Cada sprint corresponde a 2 semanas de trabajo.

Antes del sprint 1, se realizan reuniones de trabajo durante 3 semanas para tomar decisiones del equipo de desarrollo. En estas reuniones se decide sobre la estructura de datos a usar, el lenguaje de programación del software, las bibliotecas adicionales a usar para cumplir con las funcionalidades y la organización de tareas en el tiempo. Este trabajo no implica directamente la creación de un producto para el cliente y por eso no es contado como un sprint.

En la Figura 58 vemos un diagrama con los sprints y las etapas del diseño.

TAREAS	SPRINT 1	SPRINT 2	SPRINT 3	SPRINT 4	SPRINT 5	SPRINT 6	SPRINT 7
ETAPA 1 PROGRAMA							
ETAPA 2 CONTEXUALIZACIÓN							
ETAPA 3 DICTADO							

Figura 58
Sprints y tareas.

Los dos primeros sprints tienen como resultado final el software donde se puede diseñar todo el programa de un curso, esto corresponde a la etapa 1 en el diseño del curso. Al finalizar el sprint 1, se le envía al usuario experto en el dominio la lista con los parámetros del sistema que son requeridos para completar esta etapa. Una vez finalizado el sprint 2, se envía al usuario experto en el dominio la primera versión del software.

Para el desarrollo del software de la etapa de contextualización se definieron 3 sprints, ya que al relevar los requerimientos, como se verá más adelante en este documento, se observa una mayor carga de tareas que en el resto de las etapas.

Al finalizar el sprint 4, se le envía al usuario experto en el dominio la lista con los parámetros del sistema necesarios para completar las etapas de contextualización y dictado. Una vez finalizado el sprint 5, se finaliza la segunda versión del software, incluyendo las etapas de Programa y Contextualización.

Los últimos 2 sprints están destinados al desarrollo de la etapa de dictado. Al finalizar el sprint 7, se le entrega al usuario experto en el dominio la última versión del software, incluyendo todas las etapas para el diseño de un curso.

Al completar el desarrollo de cada nueva versión, se le presenta al usuario experto en el dominio del software en una reunión donde se hace una demostración del funcionamiento. Las semanas siguientes a la presentación, el usuario experto en el dominio prueba el software y hace sugerencias que son evaluadas por el equipo de desarrollo e ingresadas en la lista de tareas para la siguiente versión.

La planificación del sprint se realiza en el fin de semana previo a comenzar un nuevo sprint, allí se dividen las tareas entre los miembros del equipo de manera que las tareas sean equitativas en cuanto a esfuerzo.

D. Anexo Requerimientos

D.1 Requerimientos Funcionales

A continuación se detallan los requisitos funcionales que el sistema deberá implementar.

Autenticación y autorización de usuarios

Dado que el sistema es público, no será necesaria la autenticación para su uso general.

La autenticación es necesaria para acceder a los diseños de una institución, y asegurar que a los diseños de curso privados sean accedidos solo por los docentes autenticados y autorizados puedan hacer uso de la herramienta. Para esto se debe contar con un servidor central para cada institución.

El sistema deberá permitir a los usuarios iniciar sesión. Esto incluye brindar un sistema de soporte a la autenticación y autorización de usuarios, así como de gestión de los mismos. También deberá permitirles cerrar una sesión iniciada previamente.

Se deberán soportar los siguientes roles: docente y administrador del sistema. El administrador podrá modificar las etapas del ciclo de vida de un curso para adaptar el sistema a su institución. Cada uno de estos usuarios tendrá diferentes permisos y visibilidad de recursos del sistema dependiendo del rol que cumplan.

Permisos sobre el diseño

Permisos anónimos

El sistema debe permitir a cualquier persona que cuente con el software crear el diseño de un curso, y visualizar o modificar en función de la licencia CC elegida.

Permisos nominados

El sistema debe permitir asignar una licencia CC de derecho de autor para un grupo específico de usuarios registrados en el sistema. Los usuarios de este grupo podrán realizar las acciones definidas en la licencia CC elegida para el diseño.

Etapas

Por defecto se programará una definición para cada Etapa. Cada Etapa consiste de un conjunto de Atributos (par pregunta/respuesta) que el sistema despliega al docente y que se podrán completar en cualquier orden. Es necesario para cada Atributo saber si es modificable o no en cada Etapa. Hay atributos opcionales y atributos requeridos; los atributos requeridos necesitan ser respondidos por el docente para poder finalizar la etapa.

Actor Administrador

El usuario Administrador es el único que puede crear, modificar y eliminar los Atributos que componen cada Etapa.

Actor Docente

El sistema dará soporte al Docente para poder completar las etapas (sin poder modificarlas) del ciclo de vida de un curso.

Etapa Elaboración del programa

El sistema le permitirá al docente completar con datos los atributos de un curso.

Etapa Contextualización

El sistema le permitirá al docente realizar ajustes al diseño. En esta etapa el Docente puede modificar algún dato ingresado durante la etapa de Elaboración y completar los datos de ésta etapa.

Etapa Dictado

El sistema permitirá realizar ajustes que se toman durante el dictado del curso de acuerdo a decisiones del docente basadas en cómo el curso se desarrolla y cómo los estudiantes responden.

En esta etapa el Docente puede modificar algunos atributos ingresados durante la etapa de Instanciación.

Almacenamiento y edición

El sistema permitirá abandonar la sesión en cualquier momento y que el avance de su trabajo quede almacenado. Posteriormente, puede retomar la edición sobre el curso.

Reutilización

El sistema permitirá compartir el diseño de un curso.

Exportar

El sistema permitirá exportar un documento PDF con el resultado del proceso de diseño del curso.

Compartir diseño de curso mediante repositorio

El sistema permitirá compartir el diseño de un curso a partir de un repositorio. El docente propietario del diseño debe poder ver las modificaciones realizadas y aceptarlas o descartarlas. Además, se debe tener trazabilidad sobre los cambios incluyendo el autor y la fecha del mismo.

Compartir diseño de curso fuera de línea

El sistema permitirá exportar e importar el diseño de un curso a través de un archivo.

Sugerencias

El sistema le brindará al docente sugerencias y guías para enriquecer su elaboración. Estas sugerencias serán mostradas al docente en los Atributos, que son puntos críticos, definidos por el administrador del sistema.

Enviar sugerencias

El docente podrá agregar una herramienta o descripción que no está entre las recomendadas por el sistema, y le solicitará al administrador que la agregue.

Recibir sugerencias

El administrador recibirá una notificación con la sugerencia enviada por un docente. Tendrá la posibilidad de aceptarla o descartarla.

Recolección de datos

El docente podrá ingresar datos en el sistema en la Etapa de Dictado con dos niveles de complejidad.

Nivel básico

El docente podrá agregar a mano atributos sobre el curso. Estos atributos serán definidos por el administrador, asignándoles a cada uno un tipo de datos.

Nivel básico con documento

Se pueden importar documentos generados en el curso en formato pdf, excel, csv, etc.

Nivel avanzado

Transformar los datos importados en datos abiertos para realizar análisis de datos, generar estadísticas, etc.

Analítica

El sistema debe permitir recolectar y mostrar estadísticas acerca de los cursos ingresados en el sistema.

Alertas

El sistema deberá mostrar alertas al docente. Las alertas serán programadas para ser ejecutadas cuando la respuesta ingresada en un Atributo sea incompatible con la respuesta asociada a otro Atributo que mantienen cierta relación.

El sistema deberá enviar alertas sobre actualizaciones de alguna etapa del ciclo de vida de los cursos. El docente puede optar por actualizar los cursos ya creados que son impactados por este cambio.

Versionado

El sistema debe guardar los cambios realizados por los usuarios autorizados, y un identificador de versión. Además, debe permitir a los usuarios navegar por las distintas versiones de un elemento, y seleccionarlas como activas.

D.2 Requerimientos no Funcionales

Requerimientos de interfaces externas

El sistema funcionará sobre un sistema operativo de una computadora de escritorio. Deberá ser un sistema offline, sin necesidad de estar conectado a internet para ser utilizado.

Una vez que se tiene acceso a internet, se deberá sincronizar el trabajo realizado con un servidor central, validando los datos contra el modelo del proceso que se tiene actualmente en dicho servidor.

Interfaces de usuario

El sistema tendrá un panel donde cada usuario podrá acceder para diseñar o visualizar un curso.

La interfaz de usuario deberá estar suficientemente desacoplada como para que el motor (o backend³⁰) acepte distintos tipos de visualizaciones.

El sistema debe poseer interfaces gráficas basadas en las reglas definidas por un Design System³¹.

Interfaces de hardware

Los dispositivos soportados por el sistema son: Computadoras de escritorio y portátiles.

Interfaces de software

El cliente del sistema será desarrollado para la plataforma Windows, Linux y opcionalmente para MacOS.

Tiempos de respuesta

Toda funcionalidad del sistema y transacción de negocio debe responder en un tiempo cómodo o aceptable para el usuario.

Permisos

Los permisos de acceso al sistema serán asignados o revocados por el usuario administrador.

Seguridad

Todas las comunicaciones externas entre el cliente del sistema y el servidor central deben estar encriptadas.

Derechos de autor

El sistema debe respetar los criterios de derechos de autor definidos por el estándar Creative Commons (CC).

Su definición es un punto crítico del sistema.

El docente lo define antes de finalizar la primera etapa del diseño, y luego podrá modificarlos en cualquier momento.

Manual de usuario

El sistema debe contar con manuales de usuario para el administrador.

Mensajes de error

El sistema debe proporcionar mensajes de error que sean informativos y orientados al usuario final.

³⁰ En diseño de software, back-end refiere a la realización del procesamiento de los datos de la aplicación. Se encarga de la interacción con la base de datos y otros servidores.

³¹ Es un conjunto de patrones interconectados y prácticas compartidas coherentemente organizadas. Ayudan en el diseño de productos digitales y el desarrollo de productos como aplicaciones o sitios web. Pueden contener, entre otros, bibliotecas de patrones, lenguajes.

Comentarios privados (notas)

El docente podrá ingresar comentarios privados en cualquier momento del proceso. Estos comentarios sólo los verá dentro de la plataforma.

E. Anexo Encuesta de Usabilidad

E.1 Formulario de la encuesta

Encuesta de satisfacción

La siguiente encuesta de satisfacción es una herramienta de medición utilizada para evaluar el grado de satisfacción de los clientes o usuarios con respecto al Software para Diseñar Cursos.

La encuesta consta de una serie de preguntas abiertas o cerradas que se responden en una escala de valores, en este caso del 1 al 5, siendo 1: "Muy en desacuerdo" y 5: "Muy de acuerdo".

Los resultados de la encuesta serán analizados y utilizados para identificar áreas de mejora.

Simplicidad

Un diseño simple en un software significa que la página y su contenido están presentados de manera clara y directa, sin distracciones innecesarias o elementos complicados.

La coherencia en un software significa que todos los elementos en la página tienen un sentido de armonía y uniformidad en cuanto a su diseño, presentación de información y navegación. Esto incluye elementos como la tipografía, los colores, los iconos, las imágenes, los botones y las secciones de la página.

Un software bien ordenado significa que la información y los elementos en la página están organizados de manera lógica y fácil de navegar para los usuarios. Esto incluye la ubicación clara de la información, la utilización de encabezados, subencabezados y títulos apropiados, y la presentación de la información en secciones claramente identificadas y separadas.

1. El diseño es simple.

1 2 3 4 5

2. El diseño es fácil de comprender.

1 2 3 4 5

3. El sitio parece ordenado.

1 2 3 4 5

4. Todo tiene coherencia en este sitio

1 2 3 4 5

Colorido

La composición de colores atractiva y agradable significa que los colores utilizados en la página crean un equilibrio visual atractivo y placentero para los usuarios. Esto incluye la elección de una paleta de colores coherente y apropiada, la utilización de contraste suficiente para hacer destacar la información importante, y la consideración de cómo los colores interactúan entre sí para crear un ambiente visual agradable.

1. La composición de colores es atractiva y agradable.

1 2 3 4 5

2. Los colores combinan.

1 2 3 4 5

Construcción

Que un diseño parezca desarrollado profesionalmente significa que el sitio presenta un aspecto cuidado y refinado que transmite competencia y credibilidad a los usuarios. Esto incluye la utilización de una tipografía legible y coherente, una paleta de colores atractiva, y una estructura bien organizada de la información.

Que el diseño sea moderno significa que el sitio presenta un aspecto actual y acorde con las tendencias actuales. Esto incluye la utilización de elementos gráficos y de diseño atractivos y coherentes, una navegación intuitiva y fácil de usar.

1. El diseño parece diseñado profesionalmente.

1 2 3 4 5

2. El diseño es moderno.

1 2 3 4 5

3. El diseño del sitio no es confuso.

1 2 3 4 5

Diversidad

Que el diseño sea innovador significa que el sitio presenta características o soluciones de diseño únicas y originales que lo diferencian de otros sitios y mejoran la experiencia de usuario.

Por otro lado, que el diseño sea creativo significa que el sitio presenta un enfoque original y atractivo para la presentación de la información y la interacción con los usuarios. Esto incluye la utilización de elementos gráficos atractivos, la incorporación de soluciones de diseño novedosas y efectivas.

Un software es dinámico si presenta elementos interactivos y animaciones que dinamizan la experiencia de usuario. Esto incluye la utilización de elementos interactivos como botones,

formularios, gráficos, y efectos de transición, que permiten a los usuarios interactuar con el sitio de manera atractiva y efectiva.

1. El diseño es innovador.

1 2 3 4 5

2. El diseño se ve creativo.

1 2 3 4 5

3. El diseño es dinámico.

1 2 3 4 5

Funcionalidades

Busca medir la experiencia de usuario con algunas funcionalidades del Software.

1. ¿Cree que el software puede ser usado sin un manual de usuario?

1 2 3 4 5

2. ¿Cree que los textos de ayuda son claros y guían el proceso?

Si No

3. ¿Cuáles considera que son las funcionalidades más destacadas del software?

(texto libre)

4. Si pudieras mejorar algo de la aplicación, ¿qué sería?

(texto libre)

5. ¿Cómo evalúa la usabilidad para la gestión de versiones de un curso?

1 2 3 4 5

6. ¿Cómo evalúa la usabilidad en la etapa de Orquestación?

1 2 3 4 5

Comentarios finales

1. ¿Volvería a usar el Software para el diseño de un curso?

Si No

2. ¿Cómo calificarías tu experiencia general con nuestra aplicación?

(texto libre)

3. Otros comentarios.

(texto libre)

E.2 Respuestas recibidas

Simplicidad				
	El diseño es simple.	El diseño es fácil de comprender.	El sitio parece ordenado.	Todo tiene coherencia en este sitio
	5	5	5	4
	4	3	5	5
	5	5	4	4
	5	5	5	5
	4	4	4	4
Promedio	4,6	4,4	4,6	4,4

Promedio Simplicidad: 4,5

Promedio de respuestas en categoría Simplicidad

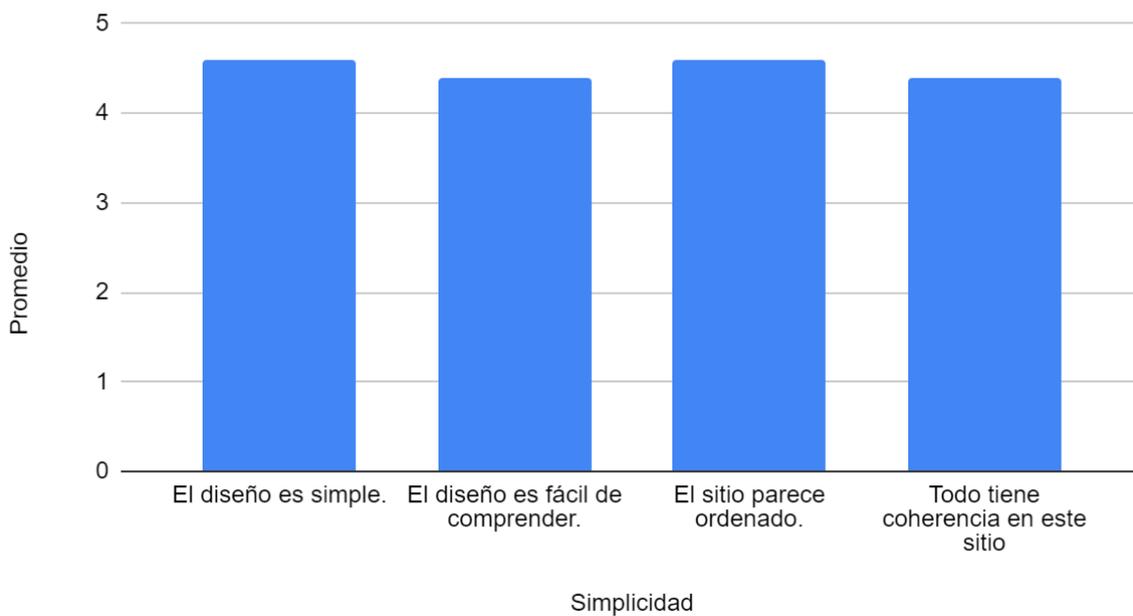


Figura 59
Promedio de respuestas en categoría Simplicidad.

Colorido		
	La composición de colores es atractiva y agradable.	Los colores combinan.
	5	5
	2	4
	4	5
	4	4

	5	5
Promedio	4	4,6

Promedio Colorido: 4,3

Promedio de respuestas en categoría Colorido

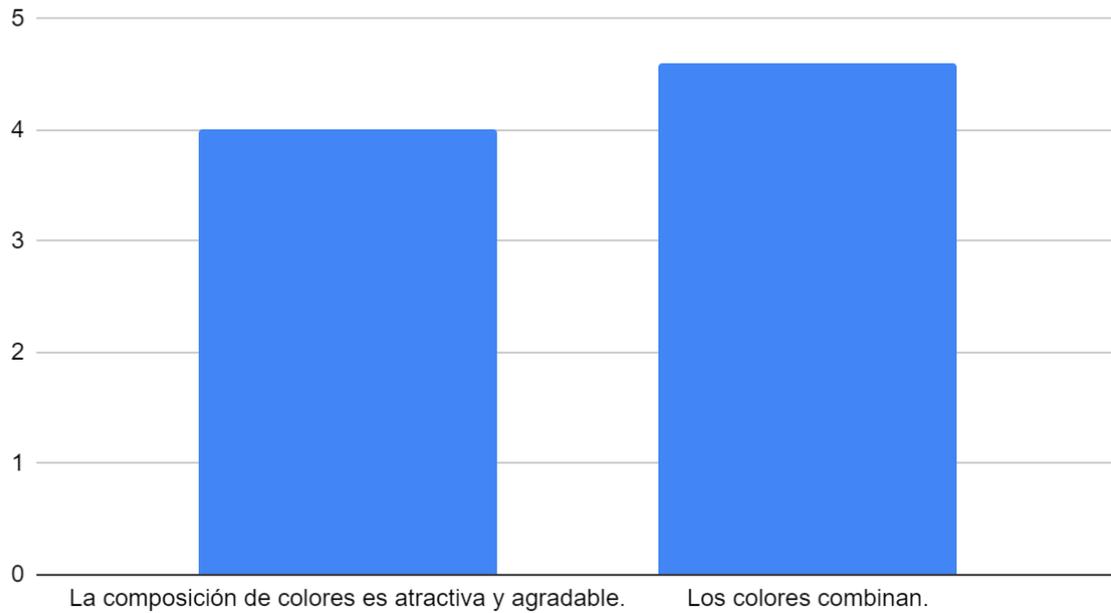


Figura 60
Promedio de respuestas en categoría Colorido.

Construcción			
	El diseño parece diseñado profesionalmente.	El diseño es moderno.	El diseño del sitio no es confuso
	4	4	5
	5	3	4
	5	4	4
	4	4	5
	5	5	4
Promedio	4,6	4	4,4

Promedio Construcción: 4,3

Promedio de respuestas en categoría Construcción

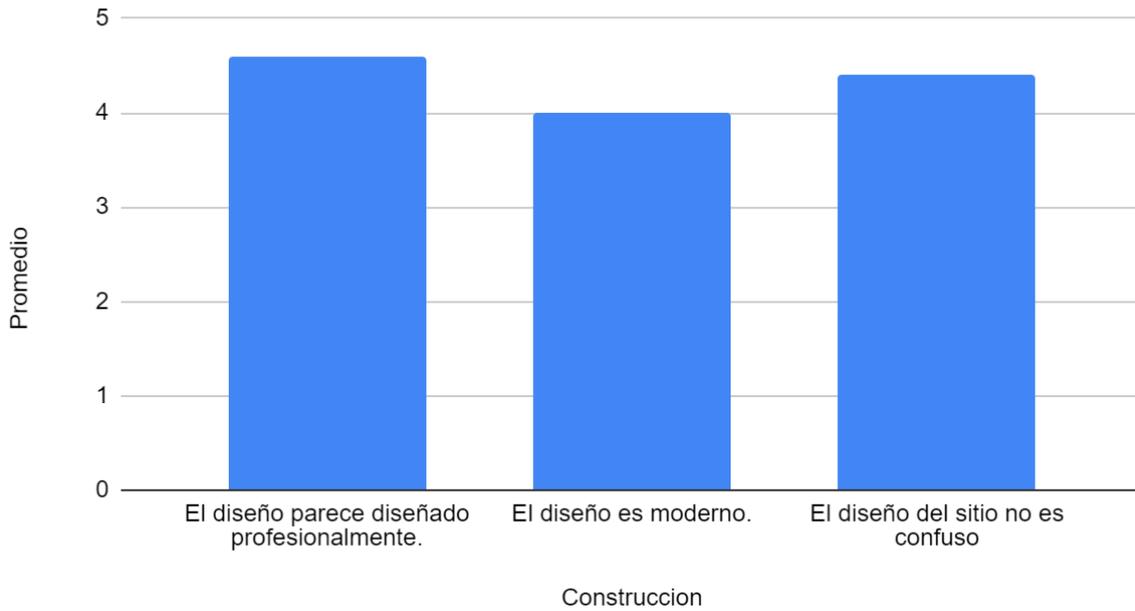


Figura 61
Promedio de respuestas en categoría Construcción.

Diversidad			
	El diseño es innovador.	El diseño se ve creativo.	El diseño es dinámico.
	3	3	4
	5	5	3
	5	5	5
	5	4	5
	5	5	4
Promedio	4,6	4,4	4,2

Promedio Diversidad: 4,4

Promedio de respuestas en categoría Diversidad

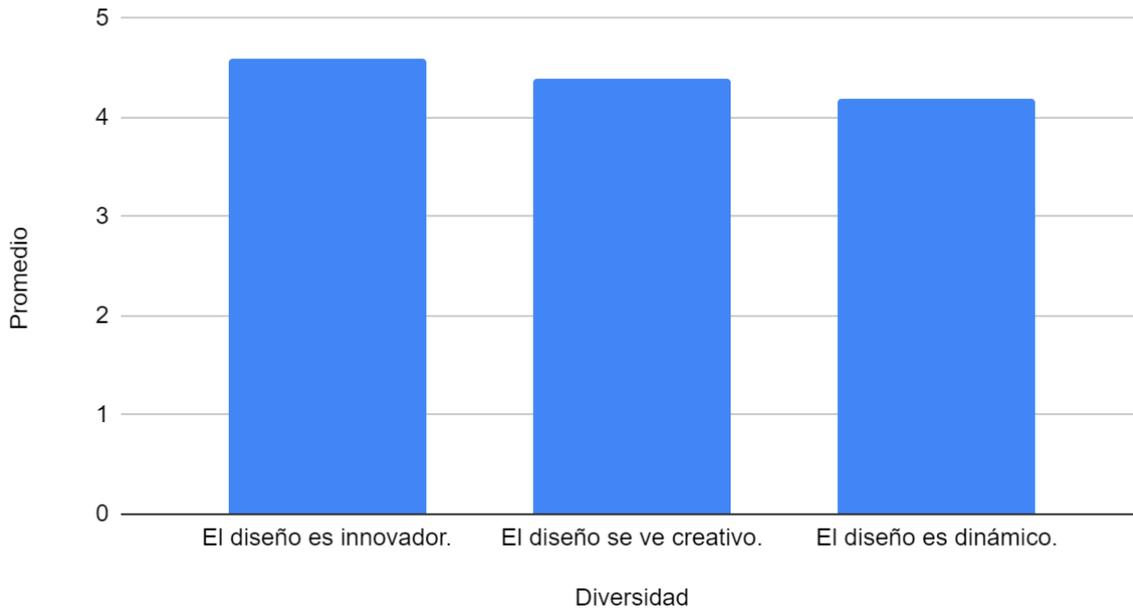


Figura 62
Promedio de respuestas en categoría Diversidad.

Funcionalidad*					
	¿Cree que el software puede ser usado sin un manual de usuario?	¿Cree que los textos de ayuda son claros y guían el proceso?	¿Cómo evalúa la usabilidad para la gestión de versiones de un curso?	¿Cómo evalúa la usabilidad en la etapa de Orquestación?	¿Volvería a usar el Software para el diseño de un curso?
	5	Si	5	3	Si
	3	Si	4	2	Si
	4	Si	3	3	Si
	3	Si	5	5	Si
	3	No	5	4	Si
Promedio	3,6	-	4,4	3,4	-

*Estas preguntas no corresponden a VisAWI. Son formuladas en el contexto de este proyecto entendiendo que puedan aportar información útil para el análisis.

Preguntas de desarrollo

¿Cuáles considera que son las funcionalidades más destacadas del software?

1. Los textos de ayuda son claros y aportan en positivo. La info ampliatoria y los aportes conceptuales son una excelente idea para ayudar a personas que tengan que realizar el diseño de sus cursos pero no tengan los insumos para hacerlo.
2. Organización. Interrelación entre diferentes aspectos del diseño.

3. Desde mi experiencia no he utilizado otros software de estas características, pero considero que su funcionalidad es bastante práctica para diseñar los contenidos de nuestro curso.
4. Facilita el tener un diseño ya resuelto.
5. La estructura es práctica y permite sistematizar la diversidad de contenidos que manejamos habitualmente en la preparación y dictado del curso; la interconexión y la posibilidad de "importar" contenidos de un área a otra es una muy buena funcionalidad; la posibilidad de generar un documento con la información organizada.

Si pudieras mejorar algo de la aplicación, ¿qué sería?

1. Si bien el diseño es muy limpio y eso es muy positivo, creo que se beneficiaría con algunos íconos. O sea, entiendo que le faltaría dentro de lo visual, íconos o imágenes que no hagan ruido pero que sí aporten claridad, decoración e innovación al diseño. Tengo un comentario sobre el contenido: en metodología de enseñanza es imposible elegir una sola (salvo que el/la docente dé cátedra). Propongo que en ese ítem se puedan elegir varias opciones.
2. Tengo varios audios, detallando mi propia experiencia. Los envío por email. En algunas partes se requiere explicación suplementaria.
3. Tuve dificultades para la instalación del software, quizás los formatos que se manejan. Debo destacar la pronta respuesta en la asistencia para solucionar estos problemas.
4. Una guía de imágenes para la instalación.
5. En la pregunta de los textos opté por NO pero en realidad sería "PARCIALMENTE" (en realidad creo que los textos aclaran, pero me parece que talvés se mejorar en algunos aspectos).

Incorporar un texto inicial que explique brevemente la estructura propuesta y que por ejemplo, explique las opciones de interconexión (que tienen un orden lógico o una trayectoria recomendada para completar: ej: el ítem Recursos: es necesario completar en Programa para poder traspasar a Contextualización y de allí a Dictado).

Tal vez sería bueno diferenciar más gráficamente los cuadros de aclaraciones, aportes conceptuales de los espacios que son para completar (poner con fondo de otro color o en cursiva-negrita, etc). Por ejemplo, en Programa/Objetivos/Objetivos de aprendizaje aparece en el mismo cuadro gris "Aporte conceptual" (con fondo blanco) y debajo el cuadro Objetivos de aprendizaje con el mismo fondo blanco. En otro caso hay un cuadro aclaratorio sobre créditos en el área de Programa / Contenidos que comienza con una nota aclaratoria sobre Contenidos y en el mismo cuadro hay un casillero de número de créditos (que sería para completar) y otro que se genera por suma de los créditos declarados más abajo. O sea, queda un poco confuso en el mismo cuadro (gris) texto aclaratorio, casilla a completar y número que se autogenera en base a lo que se escriba más abajo.

Otros detalles: En Programa/Información general/Subárea no se contemplan los ciclos iniciales de los cenures (que no se corresponden con facultades); podría subsanarse si se permitiera marcar todas las facultades a las que se accede con el

egreso del CIO pero sólo deja marcar una opción. Igual creo que lo mejor sería incluir el ítem de los ciclos iniciales de cada área (cyt, social y artística, salud).

En Recursos no queda claro (al menos para mí) que se espera en cuanto a elegir una opción de Licenciamiento de Creative Commons, porque se viene detallando toda la bibliografía y recursos educativos. Y el licenciamiento entiendo sería para recursos propios (o si refiere a la bibliografía y recursos listados, puede haber de distinto tipo, etc.)

En contextualización tal vez podría ser interesante incorporar un ítem sobre el perfil de los estudiantes (o el perfil esperado) además del número, por ejemplo: las competencias o contenidos que se espera tengan antes de iniciar el curso (podrían incorporarse otro tipo de informaciones que son útiles tener sistematizadas, tipo: edades esperadas, trayectorias curriculares anteriores, si se espera que sea un grupo con muchos estudiantes trabajadores, etc. (esto es en caso sobre todo si se pudieran tener distintas opciones para imprimir pdf, ver más abajo)

No encontré tampoco un ítem sobre previaturas que se requiera para cursar formalmente, creo que es importante incorporarlo.

En el cupo del curso tal vez es mejor hablar de "tiene" cupo y no de "acepta" cupo.

Finalmente (para no complicarlo más). La opción de exportar a pdf me parece excelente. Tal vez sería muy bueno si más adelante pudiesen desarrollar la opción de elegir qué elementos incorporar en el informe impreso, porque eso permitiría tener productos con objetivos muy diversos y que requieren insumos de información distintos: desde generar un pdf para información a los estudiantes, a otro que permita planificar en el equipo docente, otro que sea para presentar en otros ámbitos, etc.

¿Cómo calificarías tu experiencia general con nuestra aplicación?

1. El comienzo fue difícil ya que la app no funcionaba en ninguno de los sistemas operativos que yo uso: iOS, Linux y pude conseguir prestado Win7. Cuando solicité ayuda, me fue ofrecida inmediatamente y me enviaron una versión para Win7 que funcionó perfectamente. La usabilidad de la app en términos generales fue muy buena y mi experiencia de usuaria excelente.
2. Buena. Lo hice sin leer mucho las instrucciones. En general creo que es fácil de usar, aunque (como todo) se requiere de una adaptación a la interfaz. Creo que es práctico y que vale la pena pasar por ese período de adaptación para poder utilizarlo de forma más fácil y rápida. Creo que es útil y dispara la reflexión del docente sobre aspectos que quizás no había considerado. Los colores los pondría más pastel, y no tan brillosos.
3. Es una muy buena herramienta que simplifica la información.
4. Muy buena.
5. No he podido aún terminar de completarla, traté de mirar y completar ítems de las distintas áreas para poder hacer esta devolución (si encuentro algo más, después se los hago llegar). Me ha resultado interesante sistematizar la propuesta de curso e imagino que la posibilidad de hacer nuevas versiones a partir de una anterior puede ser también muy bueno.

Otros comentarios.

1. Los y las felicito por esta herramienta y quedo a las órdenes para apoyar en cualquier forma que necesiten.
2. Tengo varios audios para comentarles sobre las experiencias que fui teniendo al usarlo. Se los mando por email.
3. Agradezco la oportunidad de participar, me gustaría poder trabajar con el
4. Creo que puede ser una herramienta muy útil y sobre todo en la medida que puedan incorporar algunas mejoras y funcionalidades (como la de informes con contenidos parciales, etc.). En lo personal agradezco que se incursione en este tipo de apoyos técnicos para la función de enseñanza en la UdelaR, habitualmente tan relegada y sujeta a esfuerzos aislados y competencias particulares de los equipos docentes o docentes.