



MARKETPLACE INTERACTIVO COMO REPOSITORIO INSTITUCIONAL DE PROYECTOS DE DESARROLLO DE SOFTWARE EN LA UNIVERSIDAD POPULAR DEL CESAR



**MARKETPLACE INTERACTIVO COMO REPOSITORIO INSTITUCIONAL DE
PROYECTOS DE DESARROLLO DE SOFTWARE EN LA UNIVERSIDAD POPULAR DEL
CESAR.**

SSAYLEM ANDRES MURILLO MINDIOLA

ANDRES FELIPE RUEDA CALLEJAS



UNIVERSIDAD POPULAR DEL CESAR

FACULTAD DE INGENIERÍA Y TECNOLÓGICAS

PROGRAMA DE INGENIERÍA DE SISTEMAS

VALLEDUPAR

2025

TABLA DE CONTENIDOS

RESUMEN	13
1. SECCIÓN I: DESCRIPCIÓN GENERAL	16
1.1. TÍTULO DEL PROYECTO	16
1.2. DIRECCIÓN DE EJECUCIÓN DEL PROYECTO	16
1.3. LAPSO DE EJECUCIÓN DEL PROYECTO	16
1.4. ORGANISMO Y SECCIÓN RESPONSABLE	16
1.5. INFORMACIÓN DE CONTACTO DE LOS ESTUDIANTES	16
1.6. LÍNEA, SUBLÍNEA Y GRUPO DE INVESTIGACIÓN AL QUE SE SUSCRIBE EL PROYECTO	17
2. SECCIÓN II: DESCRIPCIÓN SITUACIONAL	17
2.1. IDENTIFICACIÓN DEL PROBLEMA	17
2.2. JUSTIFICACIÓN DEL PROYECTO	18
2.2.1. JUSTIFICACIÓN TEÓRICA	19
2.2.2. JUSTIFICACIÓN PRÁCTICA	20
2.2.3. JUSTIFICACIÓN METODOLÓGICA	20
2.3. OBJETIVOS DEL PROYECTO	21
2.3.1. OBJETIVO GENERAL	21
2.3.2. OBJETIVOS ESPECÍFICOS	21
2.4. MARCO REFERENCIAL	21
2.4.1. ANTECEDENTES DE LA INVESTIGACIÓN	21
2.4.1.1. ANTECEDENTES INTERNACIONALES	21
2.4.1.2. ANTECEDENTES NACIONALES	22
2.4.1.3. ANTECEDENTES REGIONALES	23
2.4.1.4. ANTECEDENTES LOCALES	24
2.4.1.5. ANTECEDENTES SOBRE IMPLEMENTACIÓN DE ARQUITECTURAS DE SOFTWARE	24
2.4.2. MARCO HISTÓRICO	25
2.4.2.1. EVOLUCIÓN GLOBAL DE LOS REPOSITARIOS DIGITALES INSTITUCIONALES	25
2.4.2.2. REPOSITORIO INSTITUCIONAL UNICESAR	28
2.4.3. MARCO LEGAL	28
2.4.3.1. OBJETIVO	28
2.4.3.2. ALCANCE	29
2.4.3.3. NORMATIVAS RELEVANTES	30

2.4.4. MARCO TEÓRICO.....	31
2.4.4.1. TEORÍA DEL CONOCIMIENTO ABIERTO	32
2.4.4.2. REUTILIZACIÓN DEL CÓDIGO EN EL PROCESO DE PRODUCCIÓN CIENTÍFICA.....	32
2.4.4.3. MODELOS DE COLABORACIÓN DIGITAL	33
2.4.4.4. LA INTERACTIVIDAD EN EL ENTORNO DIGITAL.....	34
2.4.4.5. GESTIÓN DEL CONOCIMIENTO ACADÉMICO	35
2.4.5. MARCO CONCEPTUAL	35
2.5. MARCO METODOLÓGICO	37
2.5.1. TIPO Y ENFOQUE DE INVESTIGACIÓN	38
2.5.1.1. ENFOQUE DE INVESTIGACIÓN ADOPTADO	38
2.5.1.2. ¿POR QUÉ DSR?	39
2.5.1.3. ESTRUCTURA GENERAL DEL ENFOQUE A APLICAR	39
2.5.2. MÉTODO DE INVESTIGACIÓN	40
2.5.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN	41
2.5.4. DESCRIPCIÓN DE LAS FUENTES PRIMARIAS Y SECUNDARIAS	42
2.5.4.1. FUENTES PRIMARIAS.....	42
2.5.4.2. FUENTES SECUNDARIAS	43
3.1.1. POBLACIÓN Y MUESTRA	44
3.1.1.1. POBLACIÓN.....	44
3.1.1.2. MUESTRA	44
3.1.2. ENTORNO Y CONTEXTO DE ESTUDIO	45
3.1.2.1. ENTORNO INSTITUCIONAL.....	45
3.1.2.2. INFRAESTRUCTURA TECNOLÓGICA	46
3.1.2.3. PERTINENCIA DEL CONTEXTO.....	46
3.1.3. VARIABLES Y CATEGORÍAS DE ANÁLISIS	47
3.1.3.1. DIMENSIÓN TÉCNICA.....	47
3.1.3.2. DIMENSIÓN ACADÉMICO-PEDAGÓGICA.....	48
3.1.3.3. DIMENSIÓN PERCEPTIVA Y ACTITUDINAL	48
3.1.4. TÉCNICAS DE ANÁLISIS Y PROCESAMIENTO DE DATOS	49
3.1.4.1. ANÁLISIS DE DATOS CUALITATIVOS	49
3.1.4.1.1. PROCEDIMIENTO	49
3.1.4.2. ANÁLISIS DE DATOS CUANTITATIVOS	51
3.1.4.2.1. PROCEDIMIENTO	51
3.1.4.3. TRIANGULACIÓN METODOLÓGICA.....	52
3.1.4.3.1. TIPOS DE TRIANGULACIÓN APLICADOS.....	52
3.1.4.4. MATRIZ DE TÉCNICAS E INSTRUMENTOS.....	53
3.1.5. HERRAMIENTAS TECNOLÓGICAS Y SOFTWARE A EMPLEAR	56

3.1.5.1. HERRAMIENTAS PARA EL DESARROLLO DEL SISTEMA	56
3.1.5.2. HERRAMIENTAS PARA EL ANÁLISIS Y VALIDACIÓN	57
3.1.5.3. HERRAMIENTAS PARA DESPLIEGUE Y HOSTING	58
3.1.5.4. HERRAMIENTAS PARA DOCUMENTACIÓN TÉCNICA	59
3.1.5.5. HERRAMIENTAS PARA COLABORACIÓN Y COMUNICACIÓN	60
3.1.6. RESULTADOS ESPERADOS	61
3.1.6.1. RESULTADOS GENERALES DEL ARTEFACTO	61
3.1.6.2. RESULTADOS ESPERADOS POR DIMENSIÓN	62
3.1.6.2.1. DIMENSIÓN TÉCNICA	62
3.1.6.2.2. DIMENSIÓN ACADÉMICO-PEDAGÓGICA	62
3.1.6.2.3. DIMENSIÓN PERCEPTIVA Y ACTITUDINAL	63
3.1.6.3. RESULTADOS INSTITUCIONALES Y DE IMPACTO	63
3.1.7. CRONOGRAMA DE ACTIVIDADES	64
3.1.8. PRESUPUESTO	66
3.1.8.1. PRESUPUESTO DE HARDWARE	67
3.1.8.2. PRESUPUESTO DE SOFTWARE	68
3.1.8.3. PRESUPUESTO DE RECURSO FÍSICO	68
3.1.8.4. PRESUPUESTO DE RECURSO HUMANO	69
4. SECCIÓN III: CONSTRUCCIÓN DEL ARTEFACTO	71
4.1. RESULTADOS DEL DIAGNÓSTICO	71
4.1.1. APLICACIÓN DE ENTREVISTAS SEMIESTRUCTURADAS	72
4.1.2. OBSERVACIONES DIRECTAS NO PARTICIPANTES (INSTRUMENTO B)	73
4.1.3. CARACTERIZACIÓN GENERAL DE LA MUESTRA	76
4.2. ANÁLISIS DE LOS RESULTADOS DIAGNÓSTICOS	77
4.2.1. ANÁLISIS CUALITATIVO DE ENTREVISTAS Y OBSERVACIONES	77
4.2.1.1. PROCEDIMIENTO APLICADO	78
4.2.1.1.1. ORGANIZACIÓN DEL MATERIAL CUALITATIVO	78
4.2.1.1.2. LECTURA COMPRENSIVA Y FAMILIARIZACIÓN	78
4.2.1.1.3. AGRUPACIÓN POR CATEGORÍAS Y DIMENSIONES	80
4.2.1.1.4. IDENTIFICACIÓN DE PATRONES Y NECESIDADES PARA EL DISEÑO	85
4.2.1.1.5. ELABORACION DE SINTESIS INTERPRETATIVA	87
4.2.2. ANÁLISIS CUANTITATIVO DE ENCUESTA ESTRUCTURADA	87
4.2.2.1. PROCEDIMIENTO APLICADO	88
4.2.2.1.1. CODIFICACIÓN Y PREPARACIÓN DE DATOS	88
4.2.2.1.2. ANÁLISIS DESCRIPTIVO UNIVARIADO	88
4.2.2.1.3. REPRESENTACIÓN GRÁFICA BÁSICA	88
4.2.2.1.4. LECTURA DE RESULTADOS EN FUNCIÓN DEL DISEÑO	97

4.2.3. TRIANGULACIÓN METODOLÓGICA (SÍNTESIS REQUISITOS POR DIMENSIÓN)	98
4.2.3.1. DIMENSIÓN TÉCNICA (R1–R3)	99
4.2.3.2. DIMENSIÓN ACADÉMICO-PEDAGÓGICA (R4–R6)	99
4.2.3.3. DIMENSIÓN PERCEPTIVA Y ACTITUDINAL (R7–R9)	100
4.3. DEFINICIÓN DE REQUISITOS DEL ARTEFACTO	101
4.3.1. OBJETIVOS SMART	101
4.3.2. CRITERIOS DE ACEPTACIÓN	103
4.3.3. BACKLOG PRIORIZADO MOSCOW	104
4.4. DISEÑO Y CONSTRUCCIÓN DEL ARTEFACTO	105
4.4.1. ARQUITECTURA GENERAL	105
4.4.2. ARQUITECTURA ESPECÍFICA	107
4.4.2.1. ARQUITECTURA DEL FRONTEND	107
4.4.2.2. ARQUITECTURA DEL BACKEND	108
4.4.3. MÓDULOS PRINCIPALES DEL ARTEFACTO	109
4.4.4. DISEÑO FUNCIONAL	111
4.4.4.1. ROLES Y ACTORES	112
4.4.4.2. CASOS DE USO PRINCIPALES	113
4.4.4.3. MODELADO DE ENTIDADES	119
4.4.4.3.1. DESCRIPCIÓN DE LAS ENTIDADES PRINCIPALES	120
4.4.4.3.2. DIAGRAMA DE CLASES UML	123
4.4.4.3.3. DIAGRAMA ENTIDAD-RELACIÓN	125
4.4.4.3.4. DIAGRAMAS DE INTERACCIÓN	126
4.4.4.4. FLUJOS DE NAVEGACIÓN (UX/UI)	138
4.4.4.4.1. FLUJO DE BÚSQUEDA	139
4.4.4.4.2. FLUJO DE LOGIN	143
4.4.4.4.3. FLUJO DE REGISTRO	144
4.4.4.4.4. FLUJO DE OLVIDAR CONTRASEÑA	145
4.4.4.4.5. FLUJO DE CREACIÓN DE PROYECTOS	148
4.4.4.4.6. FLUJO DE FEEDBACK	153
4.4.4.4.7. FLUJO DE MÉTRICAS	155
4.4.5.1 IMPLEMENTACIÓN DE LA ARQUITECTURA DISEÑADA	157
4.4.4.5. METODOLOGÍA DE DESARROLLO UTILIZADA	160
4.4.4.5.1. ENFOQUE ADOPTADO Y JUSTIFICACIÓN	160
4.4.4.5.2. ARTEFACTOS SCRUM UTILIZADOS	160
4.4.4.5.3. PRÁCTICAS XP APLICADAS	161
4.4.4.5.4. CONSIDERACIONES DE MEJORA DEL PROCESO (APRENDIZAJES)	162

4.4.4.6. PLANIFICACIÓN Y SEGUIMIENTO DEL DESARROLLO	162
4.4.4.6.1. TABLERO DE TRABAJO Y ESTADOS	163
4.4.4.6.2. ASIGNACIÓN DE TAREAS	164
4.4.4.6.3. EVIDENCIAS DEL BACKLOG Y SPRINTS	165
4.4.4.7. EVIDENCIAS DEL ARTEFACTO EN FUNCIONAMIENTO	167
4.4.4.7.1. CAPTURAS POR MÓDULO	167
4.4.4.7.1.1. INICIO / CATÁLOGO (MARKETPLACE)	168
4.4.4.7.1.2. BÚSQUEDA Y FILTRADO	173
4.4.4.7.1.3. DETALLE DE PROYECTO (VISTA INDIVIDUAL)	173
4.4.4.7.1.4. AUTENTICACIÓN Y SESIÓN	177
4.4.4.7.1.5. REGISTRO / CREACIÓN DE CUENTA (SI APLICA)	179
4.4.4.7.1.6. PUBLICACIÓN / CREACIÓN DE PROYECTOS (ESTUDIANTE)	184
4.4.4.7.1.7. GESTIÓN DE ARTEFACTOS DEL PROYECTO	188
4.4.4.7.1.8. FEEDBACK, COMENTARIOS Y RATING	190
4.4.4.7.1.9. MÉTRICAS DEL SISTEMA (GENERALES)	194
4.4.4.7.1.10. ESTADÍSTICAS INDIVIDUALES POR PROYECTO	196
4.4.4.7.1.11. DOCUMENTACIÓN / AYUDA	196
4.4.4.8. PRUEBAS TÉCNICAS DEL ARTEFACTO	199
4.4.4.8.1. PRUEBAS UNITARIAS	199
4.4.4.8.1.1. MÓDULOS CUBIERTOS POR LAS PRUEBAS UNITARIAS	200
4.4.4.8.1.2. CASOS UNITARIOS EJECUTADOS	200
4.4.4.8.2. PRUEBAS DE INTEGRACIÓN	203
4.4.4.8.2.1. METODOLOGÍA UTILIZADA	204
4.4.4.8.2.2. CRITERIOS DE ACEPTACIÓN	204
4.4.4.8.2.3. INTEGRACIÓN MANUAL (SWAGGER UI)	207
4.4.4.8.3. PRUEBAS DE SISTEMA	208
4.4.4.8.4. NIVEL DE COBERTURA DE LAS PRUEBAS	210
4.4.4.9. ESTADO DE ENTREGA DEL ARTEFACTO	211
4.4.4.9.1. ALCANCE CONSTRUIDO VS. BACKLOG MOSCOW	213
4.4.4.9.2. LIMITACIONES DE LA VERSIÓN ENTREGADA Y MEJORAS PROPUESTAS	214
REFERENCIAS	215
5. ANEXOS - INSTRUMENTOS	22

ÍNDICE DE TABLAS

Tabla 1: Información de contacto de los estudiantes.	17
Tabla 2: Normativa relevante para el marco legal.	31
Tabla 3: Fases del enfoque de Investigación en las Ciencias del Diseño.	40
Tabla 4: Técnicas e instrumentos, aplicación y objetivo.	42
Tabla 5: Tamaño estimado de la muestra por técnica.	45
Tabla 6: Dimensión técnica (variables e indicadores)	47
Tabla 7: Dimensión académico-pedagógica (variables e indicadores).....	48
Tabla 8: Dimensión perceptiva y actitudinal (variables e indicadores)	49
Tabla 9: Matriz de técnicas e instrumentos	56
Tabla 10: Herramientas para el desarrollo del sistema	57
Tabla 11: Herramientas para el análisis y validación	58
Tabla 12: Herramientas para despliegue y hosting.	58
Tabla 13: Herramientas para documentación técnica	60
Tabla 14: Herramientas para colaboración y comunicación	60
Tabla 15: Cronograma de actividades (Fase vs Producto Esperado).	66
Tabla 16: Desarrollo del presupuesto en términos de hardware.....	67
Tabla 17: Desarrollo del presupuesto en términos de software.	68
Tabla 18: Desarrollo del presupuesto en términos de recursos físicos.	69
Tabla 19: Desarrollo del presupuesto en términos de recursos humanos.....	70
Tabla 20: Resumen demográfico de entrevistados.	72
Tabla 21: Distribución observaciones por curso.....	74
Tabla 22: Síntesis de cobertura por dimensión.	76
Tabla 23: Codificación Temática Inicial.	80
Tabla 24: Codificación Temática x Fuente.....	82
Tabla 25: Agrupación de códigos por dimensiones y variables.	83
Tabla 26: Coincidencias entre fuentes (triangulación inicial).	84
Tabla 27: Síntesis interpretativa.....	87
Tabla 28: Estadísticos descriptivos completos por ítem.....	96
Tabla 29: Perfil demográfico (frecuencias).	97
Tabla 30: Priorización de funcionalidades y características en funcion del diseño.	97
Tabla 31: Perfiles de usuario y características en base al diseño.	98
Tabla 32: Síntesis de requisitos por dimensión (triangulación).....	101
Tabla 33: Objetivos SMART del artefacto.....	103
Tabla 34: Criterios de aceptación por objetivo.	103

Tabla 35: Backlog MoSCoW del artefacto.	104
Tabla 35: Módulos principales del artefacto.	111
Tabla 36: Roles y actores involucrados en el artefacto.	113
Tabla 37: Especificación caso de uso de actor estudiante.	116
Tabla 38: Especificación caso de uso de actor docente.	117
Tabla 39: Especificación caso de uso de actor administrador.	118
Tabla 40: Especificación caso de uso de actor visitante.	118
Tabla 41: Especificación caso de uso de actores externos.	119
Tabla 42: Tabla entidades principales	123
Tabla 44: Tabla de estado de entrega de artefacto.	212
Tabla 45: Tabla de alcance construido vs backlog moscow	213
Tabla 46: Tabla de limitaciones de la versión entregada	214

ÍNDICE DE FIGURAS

Figura 1: Diagrama de Gantt para la duración de cada actividad del proyecto	76
Figura 2: Gráfico de barras de distribución semestral Instrumento A.	83
Figura 3: Fórmula para cálculos de porcentaje de cumplimiento en el Instrumento B.	85
Figura 3: Gráfico de pastel de frecuencia de ítems en el Instrumento B.	86
Figura 4: Trascendencia de los proyectos en un semestre.	98
Figura 5: Utilidad percibida.	99
Figura 6: Apertura/Licencias en un proyecto.	99
Figura 7: Viabilidad de colaboración con equipos.	100
Figura 8: Aspectos motivadores en un proyecto.	100
Figura 9: Nivel de confianza en alojar un proyecto.	101
Figura 10: Control de versiones (continuidad).	102
Figura 11: Cultura pedagógica en la UPC.	102
Figura 12: Intención de uso de Marketplace.	102
Figura 13: Alfabetización en licencias.	103
Figura 14: Semestre académico de estudiantes (Instrumento C).	103
Figura 15: Experiencia de estudiantes con repositorios (Instrumento C).	104
Figura 16: Número de proyectos de aula completados por estudiantes (Instrumento C).	104
Figura 17: Experiencia de los estudiantes publicando proyectos (Instrumento C).	105
Figura 18: Diagrama de arquitectura del artefacto.	116
Figura 19: Diagrama de arquitectura del Frontend del artefacto.	118
Figura 20: Diagrama de arquitectura del Backend del artefacto.	119
Figura 21: Diagrama de casos de uso del artefacto.	124

Figura 22: Diagrama de clases	134
Figura 23: Diagrama entidad relación	135
Figura 24: Diagrama de secuencia id	137
Figura 25: Diagrama de secuencia	138
Figura 26: Diagrama de secuencia crear proyecto	139
Figura 27: Diagrama de secuencia obtener proyectos	140
Figura 28: Diagrama de secuencia comentar proyecto	141
Figura 29: Diagrama de secuencia login	142
Figura 30: Diagrama de secuencia registro usuario	143
Figura 31: Diagrama de secuencia actualizar	144
Figura 32: Diagrama de secuencia eliminar proyecto	145
Figura 33: Diagrama de secuencia notificaciones	146
Figura 34: Diagrama de secuencia gestión de roles	147
Figura 35: Diagrama de secuencia sincronización github	148

ÍNDICE DE IMÁGENES

Imagen 1: Tab de colecciones.	146
Imagen 2: Tab de búsqueda selectiva (filtros y más).	147
Imagen 3: Visualización de proyecto específico.	148
Imagen 4: Popup de login.	149
Imagen 5: Registro de cuenta desde login.	150
Imagen 6: Ventana de registro.	151
Imagen 7: Selección de olvidar contraseña desde ventana de login.	152
Imagen 8: Ventana de Olvidar Contraseña.	153
Imagen 9: Paso de restablecimiento de contraseña.	154
Imagen 10: Paso 2 de restablecimiento de contraseña.	154
Imagen 11: Paso 1 de flujo de creación de proyectos.	155
Imagen 12: Paso 2 de flujo de creación de proyectos.	156
Imagen 13: Paso 3 de flujo de creación de proyectos.	156
Imagen 14: Paso 4 de flujo de creación de proyectos.	157
Imagen 15: Paso 5 de flujo de creación de proyectos.	158
Imagen 16: Paso 6 de flujo de creación de proyectos.	158
Imagen 17: Paso 7 de flujo de creación de proyectos.	159
Imagen 18: Paso 8 de flujo de creación de proyectos.	159
Imagen 19: Vista individual de proyecto, flujo de feedback.	160
Imagen 20: Vista individual de proyecto, flujo de feedback, botón de edición.	161

Imagen 21: Paso 1 de flujo de visualización de métricas.....	162
Imagen 22: Paso 2 de flujo de visualización de métricas.....	162
Imagen 23: Paso 3 de flujo de visualización de métricas.....	163
Imagen 24: Planificación y seguimiento del desarrollo.....	167
Imagen 25: Tablero de trabajo y estados	168
Imagen 26: Evidencia de división del desarrollo entre los miembros para el artefacto.....	169
Imagen 27: Evidencia de construcción de artefacto 01.....	170
Imagen 28: Evidencia de construcción de artefacto 02.....	170
Imagen 29: Evidencia de construcción de artefacto 03.....	171
Imagen 30: Vista inicial 1 del marketplace con listado de proyectos publicados.	172
Imagen 31: Vista inicial 2 del marketplace con listado de proyectos publicados.	173
Imagen 32: Vista inicial 3 del marketplace con listado de proyectos publicados.	174
Imagen 33: Vista inicial 4 del marketplace con listado de proyectos publicados.	174
Imagen 34: Vista inicial 5 del marketplace con listado de proyectos publicados.	175
Imagen 35: Exploración de todo el catálogo.....	176
Imagen 36: Aplicación de filtros de búsqueda para localizar proyectos.	177
Imagen 37: Resultados de búsqueda luego de aplicar filtros.....	177
Imagen 38: Visualización del detalle del proyecto y su información principal 1.	178
Imagen 39: Visualización del detalle del proyecto y su información principal 2.	179
Imagen 40: Visualización del detalle del proyecto y su información principal 3.	181
Imagen 41: Acceso a artefactos asociados y recursos del proyecto.	181
Imagen 42: Interfaz de autenticación para acceso al sistema 1.	182
Imagen 43: Interfaz de autenticación para acceso al sistema 2.	183
Imagen 44: Interfaz de autenticación para acceso al sistema 3.	183
Imagen 45: Formulario de registro de usuario 1.....	185
Imagen 46: Formulario de registro de usuario 2.....	185
Imagen 47: Formulario de registro de usuario 3.....	186
Imagen 48: Formulario de registro de usuario 4.....	188
Imagen 49: Resultado del registro y acceso al entorno autenticado	188
Imagen 50: Inicio de creación de proyecto: captura de datos principales 1	189
Imagen 51: Inicio de creación de proyecto: captura de datos principales 2	190
Imagen 52: Carga y asociación de artefactos al proyecto.....	190
Imagen 53: Carga y asociación de medias al proyecto.....	190
Imagen 54: Confirmación de publicación del proyecto.....	191
Imagen 55: Proyecto publicado y visible para consulta en el marketplace.....	192
Imagen 56: Artefactos asociados al proyecto disponibles para consulta.	193
Imagen 57: Acceso a un artefacto específico desde el proyecto 1.....	193
Imagen 58: Acceso a un artefacto específico desde el proyecto 2.....	194

Imagen 59: Sección de feedback del proyecto (vista inicial).....	195
Imagen 60: Registro de retroalimentación por un usuario autenticado.	196
Imagen 61: Evidencia del comentario reflejado en la vista del proyecto.	197
Imagen 62: Evidencia de calificación (rating) aplicada al proyecto.....	198
Imagen 63: Panel de métricas generales del sistema 1.....	199
Imagen 64: Panel de métricas generales del sistema 2.....	200
Imagen 65: Estadísticas individuales asociadas a un proyecto específico.....	200
Imagen 66: Sección de documentación/ayuda disponible para usuarios 1.	201
Imagen 67: Sección de documentación/ayuda disponible para usuarios 2.	202
Imagen 68: Evidencia de ejecución de pruebas unitarias utilizando pytest.	205
Imagen 69: Evidencia de creación de tests para módulo de repositorios.....	206
Imagen 70: Evidencia de creación de tests para módulo de servicios.	206
Imagen 71: Evidencia de creación de tests para módulo de endpoints.....	207
Imagen 72: Evidencia de ejecución de prueba de integración.....	211
Imagen 73: Evidencia de modelo esperado en endpoint de prueba de Metadata de un proyecto.	211
Imagen 74: Evidencia de ejecución de endpoint Metadata de un proyecto.....	212
Imagen 75: Evidencia de ejecución Locust probando endpoints para medir problemas de latencia y rendimiento en el backend.....	213
Imagen 76: Evidencia 2 de ejecución Locust probando endpoints para medir problemas de latencia y rendimiento en el backend.....	214
Imagen 77: Captura del resumen de cobertura (porcentaje total 92%) en el reporte HTML.....	215

RESUMEN

En las instituciones educativas de nivel superior, los proyectos académicos, como por ejemplo, los proyectos de aula suelen ser proyectos que una vez finalizado su desarrollo en la asignatura pasan a ser descontinuados o abandonados en el tiempo por los estudiantes. Esta problemática no solo genera una baja tasa de reutilización del conocimiento y escaso impacto institucional sino que también limita la visibilidad, colaboración y evolución de ideas con potencial transformador.

Es así, que a partir de una revisión sistemática de literatura científica, analizando conceptos clave como repositorios digitales, marketplaces académicos, reutilización de código, colaboración y preservación del conocimiento, se identificaron oportunidades de mejora en los modelos actuales de gestión de proyectos estudiantiles.

El presente estudio tiene como finalidad proponer un nuevo enfoque, el cual consiste en desarrollar un marketplace interactivo que funcione como repositorio institucional temático para la difusión y gestión de proyectos académicos de software en la Universidad Popular del Cesar.

Como hipótesis de trabajo, se plantea que la utilización de un marketplace interactivo institucional incrementará significativamente la visibilidad, continuidad y reutilización de los proyectos de software desarrollados, generando un impacto positivo en la formación académica del programa de Ingeniería de Sistemas y en el posicionamiento de la institución como referente regional en gestión del conocimiento tecnológico. El alcance de esta investigación comprende el diseño, desarrollo, y pruebas de un artefacto funcional que se espera, pueda ser utilizado por todos los miembros del programa de Ingeniería de Sistemas en la Universidad Popular del Cesar.

Palabras claves: *Repositorio institucional, marketplace interactivo, reutilización de conocimiento, colaboración académica, proyectos académicos, desarrollo de software.*

ABSTRACT

In higher education institutions, academic projects—such as classroom-based software projects—are often discontinued or abandoned by students once the course in which they were developed concludes. This issue not only results in a low rate of knowledge reuse and limited institutional impact, but also restricts visibility, collaboration, and the evolution of ideas with transformative potential.

Through a systematic review of the scientific literature, analyzing key concepts such as digital repositories, academic marketplaces, code reuse, collaboration, and knowledge preservation, opportunities for improvement were identified in current student project management models.

The purpose of this study is to propose a new approach consisting of the development of an interactive marketplace that functions as a thematic institutional repository for the dissemination and management of academic software projects at the Universidad Popular del Cesar.

As a working hypothesis, it is proposed that the use of an institutional interactive marketplace will significantly increase the visibility, continuity, and reuse of developed software projects, thereby generating a positive impact on the academic training of the Systems Engineering program and strengthening the institution's position as a regional reference in technological knowledge management.

The scope of this research includes the design, development, and testing of a functional artifact, which is expected to be used by all members of the Systems Engineering program at the Universidad Popular del Cesar.

Keywords: Institutional repository, interactive marketplace, knowledge reuse, academic collaboration, academic projects, software development.

INTRODUCCIÓN

Las instituciones de educación superior proyectan la formación de estudiantes capaces de proponer soluciones creativas e innovadoras a problemas reales, además de desarrollar en ellos un pensamiento crítico que incentive la investigación. Semestre a semestre, los estudiantes analizan situaciones del entorno y elaboran proyectos de aula que, en la mayoría de los casos, no trascienden más allá de este ámbito académico. Solo en contadas ocasiones estos trabajos evolucionan hacia proyectos de investigación formales, tales como tesis o artículos científicos.

Esta situación evidencia una problemática relevante: una gran parte del material valioso desarrollado por los estudiantes no es aprovechado ni reutilizado, lo que representa una pérdida significativa en términos de mejora, escalabilidad y aplicación práctica de dichas soluciones. Es evidente que este tipo de iniciativas, si contaran con mayor tiempo de investigación y desarrollo, podrían consolidarse como propuestas innovadoras con capacidad de transformar el entorno. En el programa de Ingeniería de Sistemas de la Universidad Popular del Cesar, donde los proyectos de aula se orientan principalmente al desarrollo de software, se hace necesario fortalecer los mecanismos de continuidad, seguimiento y difusión de estos trabajos académicos.

Actualmente, la industria del software constituye uno de los principales motores de innovación en el desarrollo tecnológico. Por consiguiente, potenciar los proyectos académicos en esta área no solo contribuiría al fortalecimiento de la formación profesional de los estudiantes, sino que además generaría impacto en los ámbitos institucional, social y empresarial. La relevancia del presente proyecto radica en su potencial para transformar los proyectos de aula en investigaciones estructuradas y sostenibles, mediante la creación de un marketplace interactivo que funcione como repositorio institucional temático para la gestión, organización y difusión de proyectos académicos de desarrollo de software.

Esta propuesta fomentará la reutilización de código, promoverá la colaboración entre desarrolladores, aumentará la visibilidad de los proyectos, así como facilitará su evolución a lo largo del tiempo. En consecuencia, se busca innovar en la forma en que se desarrolla software dentro del contexto académico, otorgando continuidad y propósito a los denominados "Prototipos Inconclusos".

Finalmente, se presenta una revisión detallada del estado del arte en relación con marketplaces utilizados como repositorios en el desarrollo de software. Asimismo, se analizan soluciones e implementaciones similares, se contrastan diferentes posturas investigativas, se establecen fundamentos conceptuales y se identifican vacíos existentes en la literatura. Lo anterior permite justificar la necesidad de la presente propuesta y su contribución al fortalecimiento de los repositorios académicos como herramientas estratégicas para la gestión y difusión de proyectos de software.

1. SECCIÓN I: DESCRIPCIÓN GENERAL

1.1. TÍTULO DEL PROYECTO

MARKETPLACE INTERACTIVO COMO REPOSITORIO INSTITUCIONAL DE PROYECTOS DE DESARROLLO DE SOFTWARE EN LA UNIVERSIDAD POPULAR DEL CESAR.

1.2. DIRECCIÓN DE EJECUCIÓN DEL PROYECTO

- **Dirección:** Diagonal 21 n.º 29-56, Barrio Sabanas del Valle, Valledupar, Cesar
- **Tel:** 588 5592
- **Correo:** ingsistemas@unicesar.edu.co

1.3. LAPSO DE EJECUCIÓN DEL PROYECTO

El proyecto se desarrollará en un lapso de (8) meses, entre el 1 de julio del dos mil veinticinco (2025) y el 28 de febrero del dos mil veintiséis (2026).

1.4. ORGANISMO Y SECCIÓN RESPONSABLE

El organismo responsable será la Universidad Popular Del Cesar, encabezada por el rector Rober Romero Ramírez.

1.5. INFORMACIÓN DE CONTACTO DE LOS ESTUDIANTES

Nombre	Apellido	Cédula	Teléfono	Correo
--------	----------	--------	----------	--------

Andres Felipe	Rueda Callejas	1193559193	3008703366	afeliperueda@unicesar.edu.co
Ssaylem Andres	Murillo Mindiola	1003315190	3054425317	sandresmurillo@unicesar.edu.co

Tabla 1: Información de contacto de los estudiantes.

**1.6. LÍNEA, SUBLÍNEA Y GRUPO DE INVESTIGACIÓN AL QUE SE
SUSCRIBE EL PROYECTO**

- **Línea de investigación:** Tecnologías de la Información y la Comunicación.
- **Sub-línea:** Ingeniería de Software.
- **Área temática:** Construcción de Software.

2. SECCIÓN II: DESCRIPCIÓN SITUACIONAL

2.1. IDENTIFICACIÓN DEL PROBLEMA

En el escenario académico y tecnológico internacional, se evidencia una problemática generalizada: la discontinuidad de los proyectos académicos desarrollados por estudiantes universitarios. Estos proyectos, que podrían convertirse en soluciones innovadoras y valiosas para la sociedad, en la mayoría de los casos no trascienden el aula debido a la falta de mecanismos institucionales que permitan su preservación, visibilidad y reutilización. Este fenómeno limita el aprovechamiento del capital intelectual generado en la educación superior, debilita la cultura de colaboración y reduce el impacto que las universidades pueden tener en la generación de conocimiento aplicado. La ausencia de plataformas interactivas o repositorios especializados ha hecho que muchos de estos desarrollos terminan archivados y olvidados, sin aportar al aprendizaje colectivo ni a la innovación global.

En Colombia y, de manera particular, en la región Caribe, varias universidades han avanzado en la creación de repositorios institucionales digitales, tales como, la Universidad del Magdalena [53] y la Universidad del Norte [54]. Estas iniciativas representan un esfuerzo importante por preservar y difundir la producción académica mediante la digitalización de tesis, artículos científicos y trabajos de grado. Sin embargo, a pesar de su utilidad en el ámbito

documental, dichos repositorios se caracterizan por su enfoque limitado en la preservación de textos, sin considerar la necesidad de contar con plataformas interactivas que permitan la gestión, reutilización y mejora continua de proyectos de software [55].

En la Universidad Popular del Cesar (UPC), el problema adquiere mayor relevancia en el programa de Ingeniería de Sistemas. Cada semestre, los estudiantes desarrollan proyectos de software en el marco de diferentes asignaturas, los cuales, en la mayoría de los casos, se abandonan al finalizar el periodo académico. Entre las causas principales se encuentran la fragmentación de esfuerzos entre materias, la falta de integración curricular, la desmotivación estudiantil, el escaso conocimiento en herramientas de gestión colaborativa, así como la inexistencia de un repositorio institucional que funcione de manera interactiva. Como resultado, se desaprovecha el conocimiento generado, se dificulta la construcción de un ecosistema académico de desarrollo continuo y se limita el impacto que estos proyectos podrían tener en la formación profesional y en la proyección institucional. Esta situación ha sido confirmada mediante encuestas y observaciones, donde los estudiantes manifiestan la necesidad de un entorno que les permita dar continuidad a sus ideas, compartir sus avances y contar con visibilidad dentro y fuera de la universidad. De no resolverse, se corre el riesgo de mantener una cultura académica de corto plazo, donde la innovación y el valor formativo de los proyectos se pierden en el tiempo.

FORMULACIÓN DEL PROBLEMA: ¿Cómo desarrollar un marketplace institucional que fomente la continuidad, reutilización y difusión de proyectos académicos de desarrollo de software?

2.2. JUSTIFICACIÓN DEL PROYECTO

El desarrollo de proyectos académicos de software liderado por el programa de Ingeniería de Sistemas de la Universidad Popular del Cesar (UPC) enfrenta una limitación significativa la cual se relaciona a la falta de continuidad y reutilización de los proyectos estudiantiles. Esta limitante impide el aprovechamiento del conocimiento generado, afecta la formación de los estudiantes y limita el impacto institucional en términos de desarrollo tecnológico. Elementos como la fragmentación del esfuerzo en proyectos de distintas asignaturas, la falta de tiempo y recursos, el desconocimiento de tecnologías avanzadas y la desmotivación derivada de la

ausencia de continuidad en los proyectos, son las principales causantes de provocar desperdicio de ideas innovadoras, la baja reutilización de código, la pérdida de oportunidades de aprendizaje profundo y dificultades en competencias clave como lo son la colaboración en equipos de desarrollo y la gestión de proyectos.

Resolver este problema implicaría el fortalecimiento de la formación académica y el fomentar la innovación en el desarrollo de software. Un marketplace interactivo que funcione como un repositorio institucional y temático es una propuesta que proporciona facilidad en términos de gestionar, difundir y mejorar los proyectos estudiantiles, y además, potenciaría la visibilidad de los proyectos a nivel institucional y empresarial en el ámbito tecnológico y académico.

Analizar la literatura previa a esta propuesta de investigación permitió explorar cómo los repositorios digitales han demostrado ser una solución para la preservación del conocimiento, pero, también expuso problemáticas en su efectividad, limitada por su falta de interactividad y herramientas de colaboración. Los marketplace interactivos por otro lado, son alternativas innovadoras como formas de difundir proyectos mucho más atractivos e interactivos, siendo este el enfoque que motiva al proyecto a innovar, proponiendo el desarrollo de un marketplace interactivo como repositorio institucional. Esto no solo se plantea como una solución efectiva para la gestión de los proyectos de software, es también una forma de preservar el conocimiento y permitir su evolución y aplicación en contextos reales, mejorando la formación de los estudiantes y posicionando a la institución como referente en innovación tecnológica.

Este estudio, finalmente, reafirma la necesidad de contar con una herramienta que transforme la manera en cómo se gestionan y difunden los proyectos de software dentro del entorno académico, integrando las fortalezas de los marketplaces para resaltar y organizar la información y las virtudes de los repositorios como herramientas esenciales para la preservación y gestión del conocimiento.

2.2.1. JUSTIFICACIÓN TEÓRICA

La propuesta se fundamenta en teorías y estudios sobre el desarrollo colaborativo de software, el acceso abierto y la gestión del conocimiento. Algunos autores [1], [2] han hecho un destaque en la relevancia de los repositorios institucionales como infraestructuras críticas para la

preservación de la producción académica. De esta manera, los principios del open access y la reutilización de código promueven la optimización de los recursos y la innovación a partir de desarrollos previos.

En este marco, el marketplace académico interactivo se plantea principalmente como una herramienta que haga la integración correcta de estas corrientes, reforzando la importancia de la continuidad de los proyectos académicos y generando un ecosistema de aprendizaje colaborativo que trascienda el aula.

2.2.2. JUSTIFICACIÓN PRÁCTICA

Para la práctica, el proyecto responde a una problemática concreta: la pérdida de proyectos de software desarrollados en la Universidad Popular del Cesar que no tienen continuidad una vez culminado el semestre. La implementación del marketplace permitirá a los estudiantes poder publicar sus proyectos, recibir retroalimentación, colaborar en equipos interdisciplinarios y reutilizar componentes ya creados.

Los docentes contarán con una base organizada que les facilitará la evaluación de avances y la identificación de buenas prácticas. Por parte de la universidad, consolidará su producción académica, fortaleciendo su imagen institucional y generando un impacto positivo en la sociedad al poner a disposición soluciones tecnológicas aplicables a necesidades reales.

2.2.3. JUSTIFICACIÓN METODOLÓGICA

Desde el punto de vista metodológico, el proyecto representa un caso de aplicación del enfoque **Design Science Research (DSR)**, en el que se diseñará, desarrollará y validará un artefacto tecnológico con base en la realidad institucional aplicando métodos mixtos tanto **cuantitativos** como encuestas y métricas de uso del sistema y **cualitativos** como entrevistas y retroalimentación de estudiantes sobre funcionalidades clave para el desarrollo del marketplace. Gracias a esto se asegura la validez de los resultados, y se aporta un modelo replicable para otras instituciones de educación superior con necesidades similares.

2.3. OBJETIVOS DEL PROYECTO

2.3.1. OBJETIVO GENERAL

Desarrollar un marketplace interactivo como repositorio institucional temático para la gestión, reutilización y difusión de proyectos académicos de software en la Universidad Popular del Cesar mediante la aplicación del enfoque Design Science Research.

2.3.2. OBJETIVOS ESPECÍFICOS

- ❖ Identificar los factores que afectan la continuidad y difusión de los proyectos de desarrollo de software académico en la institución.
- ❖ Diseñar una arquitectura de software que contemple funcionalidades de repositorio temático, interactividad y gestión de usuarios.
- ❖ Desarrollar una interfaz que permita la publicación, búsqueda, calificación y consulta de proyectos en un entorno amigable.
- ❖ Implementar el marketplace interactivo integrando los componentes de backend y frontend definidos en la arquitectura de software diseñada.

2.4. MARCO REFERENCIAL

2.4.1. ANTECEDENTES DE LA INVESTIGACIÓN

2.4.1.1. ANTECEDENTES INTERNACIONALES

En [3] se propone E-CLOUDS, un marketplace de software utilizado como servicio para computación científica, desarrollado mediante metodología de prototipado experimental que incluyó diseño de arquitectura, implementación de prototipo funcional y evaluación mediante casos de estudio con aplicaciones científicas reales . Los resultados obtenidos demuestran que el uso de este marketplace logró reducir significativamente los tiempos de configuración de aplicaciones científicas (de horas a minutos), facilitar el acceso a recursos de cómputo para grupos de investigación pequeños y medianos sin necesidad de infraestructura propia, y mejorar la escalabilidad automática de clústeres virtuales según la demanda computacional .

La evidencia experimental confirma que el uso de estos marketplaces ha ayudado a mejorar la difusión y la reutilización de desarrollos académicos, democratizando el acceso a computación de alto rendimiento y fortaleciendo su impacto más allá del aula mediante la eliminación de barreras técnicas y económicas para la ejecución de aplicaciones científicas .

En [4] se realizó un estudio de caso sobre QGIS México, explorando la adopción de software libre geoespacial mediante la formación de comunidades académicas. El método científico empleado fue el análisis participativo y de estudio de caso, que permitió recopilar información desde la experiencia de los usuarios, los eventos de colaboración y las prácticas de interacción comunitaria. Los resultados evidenciaron que la interacción y la participación colectiva no sólo fortalecieron la adopción de soluciones tecnológicas abiertas, sino que también garantizaron su permanencia en el tiempo al generar procesos de mejora continua. El impacto del proyecto se reflejó en la consolidación de una comunidad académica activa, capaz de sostener y evolucionar el software QGIS, convirtiéndolo en una herramienta de referencia para el ámbito geoespacial en México y otros contextos educativos y profesionales.

En [1] se propone mediante análisis conceptual el marco teórico fundamental que define a los repositorios institucionales como infraestructuras esenciales para la preservación y difusión de la producción académica en la era digital, estableciendo la visión fundacional de los repositorios como servicios de gestión y diseminación de materiales digitales creados por la institución y sus miembros. Por su parte, en [2] se emplea la metodología de investigación empírica realizada por la Canadian Association of Research Libraries para identificar los factores críticos de éxito en la implementación de repositorios institucionales. Los resultados de ambos estudios lograron establecer las bases conceptuales y operativas de los repositorios académicos modernos, identificando nueve factores críticos que incluyen políticas de autoarchivo, acceso abierto, gestión de contenido y apoyo institucional, impactando significativamente en la estandarización de prácticas para garantizar la disponibilidad, preservación y accesibilidad del conocimiento académico a nivel internacional.

2.4.1.2. ANTECEDENTES NACIONALES

Dentro del marco de BIREDIAL 2011, [5] se presentó un estudio de caso sobre la implementación de repositorios digitales en universidades colombianas, utilizando como

método científico el análisis descriptivo y comparativo de experiencias institucionales. Este enfoque permitió recopilar evidencias sobre los desafíos enfrentados y los aprendizajes obtenidos en diferentes instituciones al adoptar repositorios académicos. Los resultados mostraron la existencia de barreras técnicas y organizacionales, así como la urgente necesidad de contar con sistemas flexibles que pudieran adaptarse a las realidades institucionales específicas del contexto colombiano. El impacto de este proyecto fue la creación de un modelo de referencia para otras universidades en la región, que visibilizó la importancia de la interoperabilidad y la sostenibilidad de los repositorios digitales como parte de la infraestructura académica nacional.

En [6] se realizó un análisis teórico-conceptual sobre la integración de repositorios digitales en universidades colombianas, aplicando el método científico de revisión documental y análisis conceptual en torno a la gestión del conocimiento. Este trabajo permitió estructurar un marco de interoperabilidad y sistematización del saber académico en entornos de educación superior. Los resultados evidenciaron la necesidad de fortalecer políticas de acceso abierto y la interoperabilidad entre sistemas, aunque se identificó la ausencia de funcionalidades que promovieron la colaboración directa entre estudiantes o la interactividad con los contenidos. El impacto de este estudio fue sentar bases conceptuales para la consolidación de repositorios digitales en el país, contribuyendo a la comprensión del acceso abierto como pilar en la preservación y difusión de la producción académica nacional.

2.4.1.3. ANTECEDENTES REGIONALES

El Repositorio Institucional de la Universidad del Magdalena, implementado desde 2015 [53], tiene como objetivo preservar, organizar y difundir la producción académica, científica y cultural de la comunidad universitaria mediante acceso abierto. Este repositorio se ha consolidado como una plataforma sólida para la consulta de tesis, artículos y documentos, contribuyendo significativamente a la visibilidad y conservación del conocimiento generado en la región Caribe [56]. Sin embargo, a pesar de su robustez, no cuenta con herramientas interactivas ni funciones colaborativas específicas para proyectos de software, evidenciando la necesidad de soluciones complementarias que permitan fomentar la colaboración y el intercambio de desarrollos tecnológicos, como lo sería un marketplace académico especializado. Además,

instituciones en la región Caribe, como la Universidad Autónoma del Caribe, han impulsado iniciativas digitales para preservar y gestionar recursos académicos [57], pero la integración de funcionalidades colaborativas y de reutilización enfocadas en proyectos de software aún es incipiente.

2.4.1.4. ANTECEDENTES LOCALES

En [7], la Universidad Popular del Cesar implementó un repositorio institucional digital basado en DSpace, con el objetivo de preservar, organizar y difundir la producción académica generada en la institución, incluyendo tesis, trabajos de grado y publicaciones científicas. La metodología aplicada fue de implementación tecnológica institucional, sustentada en el uso de software especializado para la gestión de repositorios digitales. Los resultados muestran una plataforma funcional para la divulgación de la información académica y científica, sin embargo, se identifican limitaciones relevantes: la ausencia de herramientas de carácter interactivo y colaborativo, la falta de mecanismos de versionado y la imposibilidad de integrar procesos de reutilización en proyectos de software. El impacto de este diagnóstico local evidencia la necesidad de una solución complementaria, como un marketplace académico interactivo, que fomente la visibilidad, la colaboración entre estudiantes y la continuidad de los desarrollos tecnológicos en el contexto institucional.

2.4.1.5. ANTECEDENTES SOBRE IMPLEMENTACIÓN DE ARQUITECTURAS DE SOFTWARE

En el ámbito del desarrollo de software, se identifican diversos trabajos que no solo proponen arquitecturas, sino que también abordan su implementación como un proceso fundamental para validar su efectividad en contextos reales. En este sentido, estudios como el desarrollo de servicios web basados en arquitecturas estructuradas evidencian que el uso adecuado de metodologías, tecnologías y buenas prácticas permite pasar de manera eficiente del diseño a la construcción de sistemas funcionales con alta calidad técnica [60]. De forma complementaria, investigaciones orientadas a la aplicación de arquitectura hexagonal demuestran cómo este enfoque puede materializarse en aplicaciones web que mejoran la

organización, centralización y trazabilidad de la información, evidenciando su utilidad en la resolución de problemáticas reales [61].

Asimismo, propuestas basadas en arquitecturas modulares y sostenibles, apoyadas en metodologías ágiles como Scrum, destacan la importancia de dividir el sistema en componentes independientes que faciliten un desarrollo iterativo, colaborativo y escalable, fortaleciendo la adaptabilidad del software durante su implementación [62]. En un contexto más amplio, trabajos internacionales abordan no solo la implementación, sino también la evaluación de arquitecturas de software, resaltando la necesidad de convertir los diseños en artefactos funcionales que puedan ser analizados y validados mediante criterios específicos [63]. Finalmente, propuestas orientadas al desarrollo de aplicaciones colaborativas, basadas en modelos cliente-servidor y arquitecturas multiplataforma, subrayan la importancia de estructurar el sistema en subsistemas que permitan cumplir requisitos funcionales y atributos de calidad como la reusabilidad, interoperabilidad y mantenibilidad [64].

En conjunto, estos antecedentes evidencian que la arquitectura de software no se limita a una fase conceptual, sino que constituye el eje que articula el diseño y la implementación de sistemas, permitiendo su materialización efectiva en soluciones funcionales alineadas con los requerimientos del entorno.

2.4.2. MARCO HISTÓRICO

2.4.2.1. EVOLUCIÓN GLOBAL DE LOS REPOSITORIOS DIGITALES INSTITUCIONALES

El presente marco histórico tiene como propósito analizar la evolución de los repositorios institucionales, los proyectos académicos de software y los marketplaces interactivos, a fin de que permita obtener una mayor comprensión sobre la necesidad de la propuesta planteada en este estudio. Durante esta revisión se identificaron transformaciones en el enfoque institucional, tecnológico, y pedagógico que influyeron en la manera en cómo se preserva, gestiona y se reutiliza el conocimiento académico.

La evolución de los repositorios digitales institucionales ha sido un proceso marcado por la constante búsqueda de mejorar el acceso, preservación, visibilidad y reutilización de los productos académicos en las instituciones de educación superior. Los primeros esfuerzos significativos surgen en la década de 1990. Iniciativas como D-Lib Magazine impulsaron el desarrollo de estándares como Dublin Core para los metadatos y el uso de DOI (Digital Object Identifier) [1] como un mecanismo de identificación persistente para documentos académicos. Estos primeros acercamientos poseían un enfoque meramente bibliográfico, orientado a la conservación estática de la producción científica.

Entre 2000 y 2005, comienza una etapa que es posible denominar como, la etapa de los repositorios abiertos y estándares de interoperabilidad. Surgen plataformas como EPrints en la Universidad de Southampton y DSpace en el MIT[5]. Siendo estos dos, los primeros repositorios institucionales ampliamente adoptados a nivel mundial. A la par de estos eventos, en el año 2001, surge el protocolo OAI-PMH (Open Archive Initiative-Protocol for Metadata Harvesting) [57], un protocolo que tenía como misión principal desarrollar y promover estándares que permitieran la interoperabilidad con el fin de facilitar la difusión eficiente de contenidos en internet, consolidando así la posibilidad de que distintos repositorios, de forma estandarizada, pudieran intercambiar información. En 2002, aparecieron iniciativas de acceso abierto tales como la Budapest Open Access Initiative [27] que velaron por promover el libre acceso al conocimiento científico.

Entre 2006 y 2015 los repositorios digitales se extendieron globalmente, siendo implementados en universidades, centros de investigación y bibliotecas. Su alcance se extendió más allá de las publicaciones científicas, abarcando diferentes recursos como tesis, conjuntos de datos e incluso materiales de aprendizaje. Aparecieron incluso nuevas plataformas que permitieron integrar funcionalidades básicas de interacción, como estadísticas de uso, comentarios, descarga y exportación de metadatos. En esta etapa, además de funcionalidades únicas y extensión en alcance, se desarrollaron nuevos modelos de metadatos personalizados para recursos educativos digitales, promoviendo una mejor clasificación y reutilización de este tipo de contenidos. Durante este segmento de tiempo, surge DSpace-CRIS, una extensión de DSpace que funcionaba como un sistema encargado de la gestión de la información relacionada a la investigación, ampliando las funcionalidades de DSpace permitiendo gestionar

perfiles de autores, proyectos y líneas de investigación. Todo dentro de los propios repositorios.

A medida que los repositorios evolucionaron en alcance y funcionalidad los términos colaboración y gestión del ciclo de vida de los contenidos se hicieron más evidentes. A partir de 2016, los repositorios institucionales comenzaron a integrar capacidades colaborativas, un cambio que fue impulsado principalmente por la necesidad de transformación de los repositorios tradicionales en espacios más interactivos y adaptativos, permitiendo no solo el almacenamiento de ficheros, sino también la capacidad de gestionar versiones, compartir recursos e incluso coordinar esfuerzos entre autores y fomentar la creación activa de conocimiento entre distintos individuos. Herramientas como en [8] se enfocaron en obtener resultados en proyectos científicos colaborativos, permitiendo trazabilidad y reutilización eficiente sobre artefactos digitales. [9] demostraron cómo las plataformas técnicas como Github siendo no diseñada como un repositorio académico, funcionan como espacios académicos de producción distribuida, basándose en las interacciones como los forks, los pull request y el control de versiones. Por su parte, repositorios como el de la UNAM, han promovido la participación activa de los estudiantes e investigadores en el ciclo completo de la publicación, consulta y reutilización del conocimiento.

En la actualidad, entre el 2021 y el 2025, el enfoque en la experiencia del usuario ha sido determinante. Plataformas como UdeMY o Domestika han sido pioneros en el entorno educativo e institucional, implementado lo que podemos definir como **marketplace académico con herramientas colaborativas**. Permitiendo a sus usuarios explorar, comentar, compartir, reutilizar, evaluar y colaborar sobre los contenidos que se proveen. Los marketplaces interactivos, tienen como función principal aumentar el alcance, aumentar la visibilidad de los cursos, proyectos y contenidos gracias a sus interfaces intuitivas, filtros dinámicos y a la retroalimentación de sus usuarios.

Los repositorios digitales han evolucionado más allá de simples archivos estáticos, a complejos entornos interactivos. En la actualidad es necesario no solo potenciar su visibilidad, sino también su uso y permitir que estos evolucionen en el tiempo. Los nuevos modelos innovadores como los marketplaces académicos adoptaron este nuevo esquema y representa el punto de partida para esta investigación, un marketplace académico interactivo que funcione

como un repositorio institucional que impulse la continuidad, visibilidad y transformación del conocimiento académico en el ámbito del desarrollo de software.

2.4.2.2. REPOSITORIO INSTITUCIONAL UNICESAR

En la actualidad, a pesar de las tendencias y avances en términos de repositorios digitales institucionales, la Universidad Popular del Cesar aún presenta brechas significativas en términos de preservación y visibilidad de los proyectos de software. La Universidad Popular del Cesar a la fecha (2025), hace uso de DSpace, un software de código abierto utilizado para crear y gestionar repositorios digitales. En este, se almacena y divulga contenido académico y científico comprendido dentro de estas siete (7) categorías: editoriales, extensión y proyección social, investigación, memoria institucional, procesos académicos, tesis y disertaciones y trabajos de grado (especializaciones y pregrados).

A pesar de que existe una persistencia en términos de divulgación científica, la visibilidad es de los contenidos es realmente baja, la herramienta es poco utilizada por los estudiantes y no existe nivel alguno de interacción o colaboración con los contenidos dentro del repositorio.

Esto lleva a otra problemática, dentro de las categorías de contenido aunque es posible encontrar trabajos de grado enfocados al desarrollo de software, estos trabajos están únicamente en formato de lectura PDF y no hay accesibilidad al código ni a la divulgación de los prototipos desarrollados. Más allá de un tema de licencia, es más una problemática orientada a la limitación que este tipo de repositorio le proporciona a la Universidad en términos de visibilidad de los proyectos, la interactividad y colaboración con los contenidos publicados. En este contexto, la propuesta de un marketplace enfocado al producto de software que promueva tanto visibilidad como herramientas de colaboración e interactividad se presenta como una respuesta a una necesidad que no ha sido resuelta dentro de la Universidad Popular del Cesar.

2.4.3. MARCO LEGAL

2.4.3.1. OBJETIVO

Este marco legal tiene como objetivo garantizar la viabilidad jurídica del marketplace interactivo como repositorio institucional de proyectos de desarrollo de software en la Universidad Popular del Cesar (UPC), asegurando el cumplimiento de normativas vigentes que regulan el uso de tecnologías de la información en el ámbito educativo, la gestión de contenido académico y la protección de los derechos de autor. Se incluyen disposiciones internacionales, nacionales e institucionales, reconociendo que el software y los proyectos académicos poseen un alcance que trasciende lo local y pueden ser reutilizados en escenarios globales, algunas directrices de la UNESCO sobre Acceso Abierto (2012): Fomentan la creación y preservación de repositorios digitales con acceso abierto para aumentar la visibilidad, reutilización y colaboración en la producción académica, en línea con la propuesta de este proyecto.

2.4.3.2. ALCANCE

El alcance del presente marco legal abarca los aspectos normativos que regulan la operación, uso y difusión de contenidos dentro del marketplace interactivo propuesto. En materia de derechos de autor, el sistema deberá respetar la Ley 23 de 1982, garantizando el reconocimiento de la autoría de los proyectos publicados y estableciendo mecanismos claros de licenciamiento. Esto implica definir qué tipo de licencia aplica a cada proyecto cargado (por defecto se recomienda Creative Commons BY o BY-SA), con el fin de que la reutilización sea posible sin infringir los derechos morales y patrimoniales de los estudiantes autores.

En cuanto al límite de los derechos de autor, es importante precisar que la plataforma no transferirá la titularidad de los proyectos a la institución; los autores conservarán sus derechos morales en todo momento. Los autores del proyecto son quienes determinan la licencia bajo la cual será publicado el trabajo. En consecuencia, el uso, reproducción, distribución o modificación del contenido estará sujeto a las condiciones establecidas en la licencia seleccionada.

Respecto a la gestión de datos personales, la plataforma deberá cumplir con la Ley 1581 de 2012 (Habeas Data), implementando políticas de privacidad claras, consentimiento informado para el tratamiento de datos, y mecanismos para que los usuarios puedan modificar o eliminar su información en cualquier momento.

En el ámbito del acceso abierto, el sistema se alinea con la Política Nacional de Ciencia Abierta [55] y con las directrices de la UNESCO (2012)[59], promoviendo la libre consulta de los proyectos publicados sin barreras de acceso económico.

Finalmente, en lo referente al licenciamiento de herramientas tecnológicas utilizadas en el desarrollo del sistema (Python, Supabase, etc.), deberá verificarse que su uso dentro del entorno institucional cumpla con los términos de cada licencia, ya que algunas herramientas de código abierto presentan restricciones para usos específicos en entornos educativos o de producción a escala.

2.4.3.3. NORMATIVAS RELEVANTES

Para el siguiente proyecto se consideran de importancia una variedad de normativas dentro del marco legal, que aseguren el regimiento de las leyes colombianas vigentes.

Norma / Documento	Descripción	Aplicación
Ley 23 de 1982 – Derechos de Autor (Colombia)	Regula los derechos morales y patrimoniales sobre obras científicas, literarias, artísticas y desarrollos de software.	Protege los proyectos académicos de software. Garantiza el reconocimiento de la autoría, define licencias de uso y regula la reproducción y reutilización de las creaciones estudiantiles.
Ley 1581 de 2012 – Protección de Datos Personales	Establece las disposiciones para el tratamiento adecuado de datos personales en	Obliga al sistema a implementar consentimiento informado, control de privacidad de perfiles y

	sistemas de información.	tratamiento responsable de los datos de estudiantes, docentes y usuarios registrados.
Política Nacional de Acceso Abierto	Promueve el libre acceso a la producción científica financiada con recursos públicos, incentivando su difusión en repositorios institucionales.	El marketplace actuará como estrategia complementaria que potencia la visibilidad, reutilización y colaboración en torno a los proyectos estudiantiles de software. ¹⁹
Resolución 167 de 2019 – Política de Ciencia Abierta	Establece lineamientos para habilitar infraestructuras digitales que fomenten el acceso, uso y creación de conocimiento abierto en Colombia.	Respaldará la plataforma como un medio que permite generar nuevo conocimiento, prácticas colaborativas y productos innovadores alineados con la política de ciencia abierta del país.
Normas sobre Licenciamiento Abierto (Creative Commons)	Conjunto de licencias abiertas que permiten distribuir contenidos respetando los derechos del autor.	Se recomendará el uso de licencias como CC BY o CC BY-SA para permitir la reutilización de los proyectos con reconocimiento, fomentando el espíritu abierto y colaborativo del repositorio.
Estatuto General y Reglamentos Académicos – Universidad Popular del Cesar	Normas institucionales que definen los principios misionales de formación integral, desarrollo científico y promoción de la innovación.	El marketplace se alinea con estos principios al ser un mecanismo concreto para preservar, difundir y mejorar la producción académica de los estudiantes del programa de Ingeniería de Sistemas.
Directrices de la UNESCO sobre Acceso Abierto (2012)	Fomentan la creación y preservación de repositorios digitales con acceso abierto para aumentar la visibilidad, reutilización y colaboración en la producción académica	Establecen un marco internacional para promover la preservación, visibilidad y accesibilidad de la producción académica y científica.

Tabla 2: Normativa relevante para el marco legal.

Fuente: Elaboración propia.

2.4.4. MARCO TEÓRICO

El marco teórico busca explicar las teorías modelos y enfoques que sustentan la presente investigación. A continuación, se pretende dar sentido a la propuesta desde un enfoque científico y académico abordando conceptos como el conocimiento abierto, la reutilización del código como herramienta de contribución científica, la colaboración digital, la interactividad en los entornos digitales y la gestión académica del conocimiento.

2.4.4.1. TEORÍA DEL CONOCIMIENTO ABIERTO

El conocimiento abierto se refiere a toda aquella producción intelectual de libre acceso y reutilización que facilita la innovación reduciendo las barreras de entrada a la información científica y ampliando su impacto social [10]. Este concepto, impulsado desde la Budapest Open Access Initiative en 2002, plantea el libre acceso a la producción académica como derecho científico fundamental.

Para este proyecto, la teoría del conocimiento abierto constituye el marco conceptual que permite justificar la creación de un marketplace como infraestructura de gestión del conocimiento organizacional, donde los proyectos de software académicos no solo se preservan, sino que se integran en procesos de co-creación y construcción social del conocimiento [11]. Esta aproximación teórica aporta al proyecto la posibilidad de medir su impacto mediante indicadores específicos de gestión del conocimiento, tales como: el acceso y reutilización de proyectos existentes, la eficiencia en búsqueda y colaboración entre estudiantes, la consolidación de una cultura investigativa estructurada, y la transferencia de conocimiento entre generaciones de desarrolladores.

Además, este enfoque permite que el proyecto sea evaluable en términos de gestión del conocimiento porque establece métricas concretas como la frecuencia de consulta de repositorios, el número de colaboraciones generadas entre proyectos, la trazabilidad del conocimiento construido, y la sostenibilidad de los desarrollos más allá del aula. Así, el marketplace no funciona únicamente como repositorio pasivo, sino como ecosistema dinámico de intercambio, validación y evolución del conocimiento tecnológico producido institucionalmente.

2.4.4.2. REUTILIZACIÓN DEL CÓDIGO EN EL PROCESO DE PRODUCCIÓN CIENTÍFICA

En el campo del desarrollo de software, el código no debe ser entendido únicamente como el resultado de un producto técnico, sino también como un artefacto científico que, cuando está debidamente documentado y diseñado para ser reproducible, se convierte en un insumo esencial para la producción y gestión del conocimiento [12]. En el contexto del presente proyecto, este enfoque resulta especialmente relevante, ya que la implementación de un marketplace académico interactivo permitirá que los proyectos de software desarrollados en la Universidad Popular del Cesar no queden aislados, sino que puedan ser reutilizados, versionados y mejorados por otros estudiantes y docentes.

Autores como [13] y [14] resaltan que la reutilización del código no solo incrementa la eficiencia y evita la duplicación de esfuerzos, sino que además fomenta la innovación y estandariza procedimientos dentro de entornos educativos. En este sentido, el marketplace se convierte en una plataforma de gestión del conocimiento, donde cada proyecto de software deja de ser un producto efímero y pasa a formar parte de un ecosistema en constante evolución. Esto no solo tiene impacto en el fortalecimiento de las competencias técnicas de los estudiantes, sino que también posiciona a la institución como un referente en preservación, difusión y aprovechamiento del capital intelectual generado en su comunidad académica.

2.4.4.3. MODELOS DE COLABORACIÓN DIGITAL

Las plataformas digitales han transformado profundamente la manera de colaborar y construir conocimiento, no solo en el ámbito empresarial y profesional, sino también en el académico. Herramientas como GitHub han demostrado que la colaboración descentralizada en torno al desarrollo de software no solo facilita la construcción colectiva de soluciones tecnológicas, sino que también promueve prácticas de innovación sostenida [9].

En el marco del presente proyecto, estos modelos de colaboración digital sirven como base para el marketplace académico interactivo, ya que permiten que los proyectos de software desarrollados por estudiantes de la Universidad Popular del Cesar sean transparentes, trazables y accesibles para otros miembros de la comunidad. De esta manera, no solo se

garantiza la continuidad de los proyectos, sino que también se construye un entorno de gestión del conocimiento, donde cada contribución es registrada, evaluada y mejorada colectivamente.

Estudios como [11] y [15] evidencian que los entornos digitales colaborativos fortalecen la transparencia, optimizan la cooperación y potencian la creación colectiva. Esto significa que el impacto del marketplace no se limita a conservar proyectos, sino que crea un ecosistema de aprendizaje compartido, donde la producción académica de la institución deja de ser aislada para convertirse en un activo estratégico de innovación y desarrollo, evaluable en términos de gestión del conocimiento y de aportes concretos a la sociedad.

2.4.4.4. LA INTERACTIVIDAD EN EL ENTORNO DIGITAL

La interactividad, entendida como la capacidad del sistema para responder a las acciones del usuario, constituye un elemento clave en el diseño de plataformas digitales orientadas a la educación y la innovación. Autores como [16] destacan que la interactividad genera experiencias de aprendizaje más inmersivas y significativas, mientras que [17] resalta que los usuarios no solo consumen información, sino que también participan activamente en su producción, estructuración y circulación. Desde la perspectiva de la usabilidad, [18] subraya la importancia de la interactividad como un factor determinante en la eficacia comunicativa de una plataforma, influyendo directamente en la calidad de la experiencia del usuario.

En el contexto del marketplace académico interactivo de la Universidad Popular del Cesar, la interactividad no solo mejora la usabilidad de la plataforma, sino que también transforma los proyectos de software en espacios vivos de gestión del conocimiento. Al permitir que los estudiantes califiquen, comenten, reutilicen y versionen proyectos, se construye un ecosistema en el que el conocimiento deja de ser estático y se convierte en un recurso dinámico y colectivo.

El impacto de esta característica es doble: por un lado, motiva a los estudiantes a compartir y enriquecer sus aportes, y por otro, habilita a la institución para medir, evaluar y gestionar el conocimiento generado, fortaleciendo así la innovación continua y la cultura de preservación digital. En este sentido, la interactividad no solo mejora la experiencia del usuario, sino que se convierte en un criterio evaluable en términos de gestión del conocimiento, al permitir trazar, validar y escalar los aprendizajes producidos dentro de la comunidad académica.

2.4.4.5. GESTIÓN DEL CONOCIMIENTO ACADÉMICO

La gestión del conocimiento académico tiene como propósito transferir y transformar el conocimiento desde el lugar donde se genera hasta el lugar en donde será aplicado [19]. En el ámbito universitario, esto implica identificar, capturar, organizar, compartir y reutilizar el conocimiento de manera que se potencie la calidad de la formación y el impacto de la investigación. En esta línea, [1] reconoce a los repositorios institucionales como infraestructuras críticas para preservar y difundir el conocimiento en la era digital, mientras que [20] y [21] destacan que gestionar conocimiento no se limita a almacenar información, sino que requiere integrar dimensiones tecnológicas, organizativas y sociales que aseguren su accesibilidad, utilidad y reutilización.

En este sentido, el marketplace académico interactivo de la Universidad Popular del Cesar representa una innovación que va más allá de un repositorio documental: su aporte radica en ofrecer un entorno dinámico y colaborativo donde los proyectos de software no solo se preservan, sino que también pueden evolucionar mediante la retroalimentación, el versionado y la reutilización del código. Esto convierte al sistema en un espacio activo de circulación de conocimiento, con impacto directo en la formación práctica de los estudiantes, en la calidad de la producción académica y en la capacidad de la institución para proyectar soluciones tecnológicas hacia la sociedad.

2.4.5. MARCO CONCEPTUAL

Para la sección del marco conceptual se establecen los principios en los cuales se fundamenta este proyecto los cuales incluyen: principios colaborativos de desarrollo de software, ayuda en la preservación de material digital y la utilización de herramientas tecnológicas para el desarrollo educativo dentro de la institución.

En el contexto de la Universidad Popular del Cesar, la implementación de un marketplace como repositorio de desarrollo institucional de software, es de suma importancia debido a la falta de visibilidad actual que existe en términos de desarrollo de soluciones tecnológicas, y la falta de colaboración entre estudiantes y docentes. Estos elementos permiten establecer la

pertinencia de desarrollar e implementar el artefacto propuesto, reforzando aún más la imagen de la Universidad Popular del Cesar.

Con el fin de comprender la relevancia del estudio realizado, se definen a continuación los conceptos clave que sustentan este proyecto.

Marketplace: Un marketplace consiste en una plataforma en línea que facilita la compra y venta de productos o servicios entre múltiples usuarios, actuando como un intermediario entre compradores y vendedores [22]. Se clasifican en B2B (Business to Business), B2C (Business to Consumer) y C2C (Consumer to Consumer), dependiendo del tipo de transacciones que estos faciliten [23].

Interactividad: En términos de entornos digitales, la interactividad se refiere a la capacidad de los usuarios para influir activamente tanto en el contenido, como en la navegación y la experiencia dentro de una plataforma [16]. La interactividad busca mejorar la experiencia del usuario al proporcionar herramientas como filtros de búsqueda, personalización de contenido y retroalimentación en tiempo real [24].

Marketplace interactivo: Se define como una plataforma digital que además de facilitar la compra, venta o intercambio de productos y servicios, permite funciones avanzadas de interactividad con los usuarios a través de la colaboración, retroalimentación, personalización del contenido, o funciones que enriquecen la experiencia del usuario.

Marketplace académico: Consiste en una plataforma digital que ha sido adaptada al entorno educativo permitiendo a los usuarios explorar, publicar y reutilizar el contenido académico, fomentando un ecosistema dinámico de aprendizaje colaborativo. Este concepto traslada el enfoque de un marketplace de tipo tradicional a un espacio donde la interacción de los estudiantes es la base, priorizando elementos como la visibilidad del trabajo académico y la mejora continua de los proyectos realizados. Este concepto además, contempla aspectos como la calificación comunitaria, retroalimentación y versionado de los proyectos desarrollados.

Repositorio digital: Los repositorios digitales se definen como sistemas centralizados donde se almacena información digital, habitualmente, en bases de datos o archivos informáticos, permitiendo así su acceso, preservación y consulta [6].

Repositorio institucional: Comprende un sistema administrado por una institución educativa. Es utilizado como herramienta para el almacenamiento, la preservación y difusión de toda la producción académica de la institución.

Proyectos Académicos: Consisten en iniciativas desarrolladas en entornos educativos que buscan profundizar en el conocimiento de un tema específico, resolver problemas o innovar en un área determinada. Implican la identificación de un tema de estudio y la definición de objetivos claros.

Difusión de proyectos académicos: Es el proceso de compartir y hacer accesibles resultados de investigaciones y producciones académicas a la comunidad científica y en general, a la sociedad a través del uso de plataformas digitales, repositorios institucionales, publicaciones en acceso abierto u otras plataformas tecnológicas con el fin de maximizar su visibilidad, impacto y aprovechamiento [25].

Desarrollo de software: El desarrollo de software consiste en un proceso estructurado que comprende una serie de etapas que van desde la identificación de requerimientos hasta la implementación y el mantenimiento de un sistema, con el objetivo de resolver problemas o satisfacer necesidades específicas [26].

Reutilización de código: La reutilización de código es una práctica en el desarrollo de software que fomenta la utilización de componentes de código ya elaborados previamente en nuevas aplicaciones o sistemas [12].

Colaboración en el desarrollo de software: Consiste en un proceso continuo de trabajo conjunto entre desarrolladores, clientes y otros actores involucrados en el proyecto, donde se comparten conocimientos, se coordinan esfuerzos y como resultado se obtiene una mejora en la calidad del producto de software final.

2.5. MARCO METODOLÓGICO

El presente marco metodológico describe los métodos, técnicas e instrumentos que se emplearán para recopilar, analizar e interpretar la información necesaria para el desarrollo del proyecto. Asimismo, se precisan las fuentes, los participantes y los procedimientos mediante los cuales se obtendrán los datos y se validarán los resultados.

El propósito central es garantizar que el diseño y la construcción del marketplace interactivo como repositorio institucional se fundamenten en información confiable, pertinente y contextualizada, lo que permitirá tomar decisiones sustentadas en las necesidades reales de la comunidad académica.

2.5.1. TIPO Y ENFOQUE DE INVESTIGACIÓN

La investigación adoptará un enfoque metodológico mixto, integrando técnicas cualitativas y cuantitativas con el objetivo de lograr una comprensión integral de la problemática.

Por una parte, las técnicas cualitativas permitirán explorar percepciones, motivaciones y desafíos de los actores involucrados, facilitando la identificación de limitaciones y necesidades relacionadas con la continuidad y visibilidad de los proyectos académicos de software. Por otra parte, las técnicas cuantitativas aportarán mediciones objetivas que permitirán validar resultados y evaluar el impacto del sistema mediante encuestas estructuradas.

La investigación se clasifica como aplicada y descriptiva, dado que propone una solución práctica a una problemática institucional específica y caracteriza fenómenos relacionados con la gestión y difusión de proyectos de software desarrollados por estudiantes.

2.5.1.1. ENFOQUE DE INVESTIGACIÓN ADOPTADO

El estudio se desarrollará bajo el paradigma de Investigación en Ciencias del Diseño (Design Science Research – DSR), el cual se centra en la creación, evaluación y mejora de artefactos tecnológicos orientados a resolver problemas reales.

Este enfoque no solo contempla la construcción del sistema, sino también la validación de su utilidad, eficacia y aporte al conocimiento aplicado. En este caso, el artefacto propuesto buscará mejorar la visibilidad, preservación y reutilización de proyectos académicos de software.

A diferencia de los enfoques tradicionales centrados exclusivamente en la observación o descripción de fenómenos, DSR integra diseño, teoría, validación y mejora continua dentro de un ciclo iterativo. Por ello, resulta adecuado en disciplinas como la ingeniería de software, donde la creación de soluciones tecnológicas constituye parte esencial del proceso investigativo.

2.5.1.2. ¿POR QUÉ DSR?

El enfoque DSR ofrece una estructura metodológica clara que combina construcción iterativa y fundamentación teórica sistemática. Esta característica permite integrar la solución tecnológica con el soporte conceptual, fortaleciendo tanto la calidad técnica como la usabilidad del sistema.

Su adopción responde a la necesidad de diseñar e implementar una plataforma funcional y académicamente válida, asegurando que el artefacto constituya una contribución tecnológica sustentada metodológicamente. Sin embargo, en esta investigación en específico dado que el objetivo es el desarrollo de un sistema funcional, se limitará únicamente a utilizar las fases requeridas para la construcción y se espera en la evolución del proyecto pueda aplicarse por completo las fases posteriores que permitan ajustar el artefacto dependiendo las necesidades identificadas.

2.5.1.3. ESTRUCTURA GENERAL DEL ENFOQUE A APLICAR

A continuación se detallan las fases del enfoque de Investigación en las Ciencias del Diseño:

Fase	Aplicabilidad
1. Identificación del problema	Diagnóstico de la situación actual mediante encuestas, entrevistas y revisión documental.
2. Definición de objetivos	Establecimiento de funcionalidades clave del artefacto: publicación, búsqueda, calificación.
3. Diseño y desarrollo	Construcción del sistema utilizando herramientas como frameworks, bases de datos, entre otros.
4. Demostración	Implementación piloto en el entorno académico real con estudiantes de la universidad.
5. Evaluación	Pruebas de usabilidad (SUS), encuestas de satisfacción, análisis de métricas del sistema.
6. Comunicación	Sistematización de resultados, redacción del informe final y propuesta de publicación.

Tabla 3: Fases del enfoque de Investigación en las Ciencias del Diseño.

Fuente: Elaboración propia.

Las fases anteriormente mencionadas son fases generales de DSR, garantizan un proceso sistemático que además de producir un artefacto tecnológico funcional posibilita la evaluación rigurosa de su efectividad y adopción en el entorno educativo.

2.5.2. MÉTODO DE INVESTIGACIÓN

Esta sección describe los métodos específicos pertinentes que guiarán el desarrollo del proyecto. Se detallan los métodos que permitirán cumplir con los objetivos establecidos bajo el enfoque de las ciencias del diseño (DSR) y el enfoque metodológico mixto.

La integración de métodos cualitativos, cuantitativos y técnicos se organiza en dos fases principales: la recopilación de información y el diseño y desarrollo del artefacto.

- A) **Fase de recopilación de información:** En esta fase se utilizarán métodos de identificación de necesidades, limitaciones y oportunidades que permitan fundamentar el diseño del artefacto propuesto, como entrevistas, encuestas, análisis documental, etc.

B) **Fase de diseño y desarrollo del artefacto:** Esta fase corresponde a la aplicación de técnicas propias del diseño y la ingeniería del software, como el prototipado de interfaces, desarrollo iterativo con metodologías ágiles y pruebas funcionales continuas.

2.5.3. TÉCNICAS E INSTRUMENTOS DE RECOLECCIÓN DE INFORMACIÓN

Para garantizar la obtención de información relevante, confiable y útil en cada una de las fases del proyecto, se utilizarán las siguientes técnicas e instrumentos de recolección de datos:

Técnicas cualitativas				
Técnica	Descripción	Instrumento	Aplicación	Objetivo
Entrevistas semiestructuradas	Se realizan a estudiantes vinculados a proyectos de desarrollo de software. Esto permitirá comprender percepciones sobre la visibilidad, continuidad y reutilización de sus trabajos.	Encuesta con preguntas abiertas, diseñadas para fomentar la reflexión crítica acerca de sus experiencias.	Estudiantes en la fase de diagnóstico	Explorar percepciones sobre continuidad, visibilidad y uso actual de proyectos académicos.
Observación directa no participante	Se realizará observación de los espacios de informática y aulas donde se desarrollan proyectos, para comprender cómo se produce, gestiona y almacena actualmente el software académico.	Formato de registro de observación estructurado.	Fase de diagnóstico	Documentar interacciones y dificultades reales de uso.
Técnicas cuantitativas				

Técnica	Descripción	Instrumento	Aplicación	Objetivo
Encuestas estructuradas	Aplicadas a una muestra de estudiantes, que permita cuantificar aspectos como: nivel de satisfacción con los proyectos actuales, nivel de visibilidad, percepción sobre la utilidad de un repositorio interactivo, entre otros.	Cuestionario digital con escalas Likert y preguntas cerradas.	Pre implementación del artefacto	Identificar la percepción preliminar de utilidad y pertinencia del sistema propuesto mediante entrevistas estructuradas.

Tabla 4: Técnicas e instrumentos, aplicación y objetivo.

Fuente: Elaboración propia

2.5.4. DESCRIPCIÓN DE LAS FUENTES PRIMARIAS Y SECUNDARIAS

En esta sección se explicarán las fuentes primarias y secundarias que respaldarán el análisis del problema, así como el diseño del sistema, estas fuentes respaldarán el que será el diagnóstico, y el diseño metodológico de la solución que se ha propuesto, brindando información teórica, de ayuda para el desarrollo del marketplace interactivo.

2.5.4.1. FUENTES PRIMARIAS

Estas fuentes corresponden a información original y sin interpretación. Corresponde a información que ha sido generada de manera directa mediante procesos de investigación, creación o experiencia concreta. En la presente propuesta, este tipo de fuentes permiten comprender el contexto institucional, validar hipótesis y la obtención de evidencia empírica la cual respalda el diseño del sistema.

1. **Estudiantes del Programa de Ingeniería de Sistemas (UPC):** Uno de los insumos clave que permitirán comprender mejor el contexto relacionado, las opiniones y experiencias reales de los estudiantes serán parte clave de este proceso, tanto como para definir funciones como para hacer validaciones del sistema, público objetivo para el sistema, siendo su participación pertinente y parte del producto final.

2. **Proyectos académicos existentes:** Este tipo de proyectos permiten evidenciar el estado actual de la conservación, visibilidad y posibilidad de reutilización de trabajos que se encuentren en fase de finalización. Se hará una revisión de todo tipo de documentos, prototipos y entregables de proyectos realizados por estudiantes de la carrera (tesis, trabajos de aula, prototipos) generados en semestres anteriores.

2.5.4.2. FUENTES SECUNDARIAS

Este tipo de fuentes corresponden a aquellas que presentan información sintetizada, reorganizada o analizada a partir de fuentes primarias, tienen como función principal contextualizar y enriquecer el análisis realizado. En la presente propuesta, estas fuentes sustentan el marco teórico a través de la identificación de antecedentes relevantes que orientan el desarrollo de la solución.

1. **Revisión sistemática de literatura científica:** Esta revisión permite poder establecer el marco teórico y poder identificar buenas prácticas para diseñar el sistema, estas no son limitadas únicamente a tesis, artículos académicos, libros, ponencias y papers científicos que estén estrechamente relacionados con repositorios digitales, marketplaces de tipo académico, así como todo lo referente a su usabilidad y colaboración digital.
2. **Normativas legales y políticas institucionales:** Estas son fuentes altamente importantes que se deberán tener en consideración alta en todas las fases del proyecto, aseguran que el sistema que se está proponiendo como solución, respetan las leyes vigentes y que estén alineadas con todos los principios institucionales de la formación, propiedad intelectual y ciencia abierta. Leyes nacionales, como la ley 23 de 1982 y la ley 1581 de 2012, estas son políticas de acceso abierto y reglamentos académicos de la Universidad Popular del Cesar.
3. **Repositorios institucionales y plataformas análogas:** Para el estudio de estos sistemas nacionales e internacionales como pueden ser SeDiCI, el repositorio de la Universidad del Magdalena o plataformas como GitHub, estas permiten poder identificar las fortalezas, debilidades y los elementos que se pueden replicar, esto

brinda una referencia concreta que permita orientar el diseño desde sus enfoques los técnicos, funcionales y estratégicos.

3.1.1. POBLACIÓN Y MUESTRA

Esta sección define la población objetivo del estudio y la muestra seleccionada para la recolección de datos y validación del artefacto.

3.1.1.1. POBLACIÓN

La población objeto de estudio estará conformada por estudiantes del Programa de Ingeniería de Sistemas de la Universidad Popular del Cesar, particularmente aquellos que participen en el desarrollo de proyectos de software en asignaturas de ciclo medio y avanzado.

Se considerarán estudiantes entre tercer y décimo semestre que cursen asignaturas orientadas al desarrollo de software, bases de datos o proyectos de grado.

Esta población resulta pertinente por constituir el público objetivo del artefacto propuesto, al ser los principales generadores y beneficiarios de los proyectos académicos cuya preservación y reutilización se pretende fortalecer.

3.1.1.2. MUESTRA

Se trabajará con una muestra no probabilística por conveniencia, seleccionando participantes disponibles con experiencia en el desarrollo de proyectos de software y participación activa en el programa.

Criterios de inclusión:

- Haber desarrollado al menos un proyecto de software como trabajo de aula o tesis.
- Aceptar voluntariamente participar en encuestas o pruebas de usabilidad.

Tamaño estimado de la muestra por técnica
--

Técnica / instrumento	Tamaño estimado	Justificación
Encuesta estructurada	30 estudiantes	Esto ayuda a obtener datos cuantitativos sobre el contexto y la percepción del problema presente.
Observación directa no participante	2 grupos de estudiantes trabajando en proyectos, durante al menos 5 sesiones seleccionadas.	Esto ayudará a registrar prácticas reales de organización, colaboración y documentación de proyectos.
Entrevistas semiestructuradas	13 estudiantes	Reportan la profundidad de tipo cualitativo para poder identificar las necesidades y los criterios funcionales.

Tabla 5: *Tamaño estimado de la muestra por técnica.*

Fuente: *Elaboración propia*

3.1.2. ENTORNO Y CONTEXTO DE ESTUDIO

El entorno y contexto de estudio hace referencia al espacio físico, institucional, tecnológico y social en el cual se desarrollan las actividades de investigación. El análisis del entorno y la contextualización permite comprender las condiciones reales que engloban el problema así como la identificación de factores que resulten limitantes o habilitantes para la implementación de la solución propuesta.

La presente investigación se desarrolla para el programa de Ingeniería de Sistemas de la Universidad Popular del Cesar (UPC), sede Valledupar, institución educativa de nivel superior pública que se encuentra ubicada en el departamento del Cesar, Colombia. La universidad cuenta con múltiples programas profesionales y técnicos, y se encuentra en constante proceso de fortalecimiento en materia de investigación, innovación y tecnologías de la información.

3.1.2.1. ENTORNO INSTITUCIONAL

El programa de Ingeniería de Sistemas se encuentra estructurado con el fin de formar profesionales competentes en el desarrollo de software, gestión de tecnologías, innovación y

trabajo colaborativo. A lo largo de su formación, los estudiantes llevan a cabo el desarrollo de proyectos de software en diversas asignaturas, tales como (pero no limitadas):

- Ingeniería de Software
- Programación Web
- Programación Móvil
- Bases de Datos
- Proyectos de Aula
- Seminarios de Investigación
- Trabajo de Grado

Sin embargo, la visibilidad y continuidad de estos proyectos es limitada, debido a la ausencia de una plataforma digital institucional moderna asentada que permita preservarlos, reutilizarlos y compartirlos, lo que genera una alta tasa de “proyectos olvidados” una vez finalizado el semestre académico.

3.1.2.2. INFRAESTRUCTURA TECNOLÓGICA

Actualmente, la Universidad Popular del Cesar dispone de una red institucional, acceso a plataformas académicas como Moodle, y una infraestructura básica para alojar sistemas web. Adicionalmente, cuenta con un repositorio digital basado en DSpace, usado exclusivamente para almacenar tesis en formato PDF, sin opciones interactivas, ni de colaboración ni de acceso al código fuente.

Esto evidencia una oportunidad crítica de mejora: habilitar un nuevo entorno digital moderno, enfocado específicamente en proyectos de software, con capacidades interactivas, gestión de versiones y visibilidad extendida.

3.1.2.3. PERTINENCIA DEL CONTEXTO

El contexto local y regional en el que se sitúa la Universidad Popular del Cesar también se caracteriza por una creciente demanda de soluciones tecnológicas relacionadas a problemas sociales, educativos y empresariales. En este sentido, potenciar los desarrollos estudiantiles

no solo beneficia a la comunidad universitaria, sino que también contribuye a generar soluciones tecnológicas replicables en otros entornos.

La investigación, por tanto, se enmarca en un entorno institucional que necesita modernizar sus mecanismos de gestión del conocimiento, y donde una solución como un marketplace académico interactivo representa un aporte estratégico en términos de innovación, visibilidad y aprovechamiento del capital intelectual estudiantil.

3.1.3. VARIABLES Y CATEGORÍAS DE ANÁLISIS

Las variables y categorías de análisis se centran en el diagnóstico inicial de la problemática para informar el diseño del marketplace interactivo. Se obtendrán mediante entrevistas semiestructuradas, observación directa no participante y encuestas estructuradas aplicadas a estudiantes del programa de Ingeniería de Sistemas. Estas categorías guiarán la identificación de necesidades funcionales, pedagógicas y perceptivas que fundamenten la arquitectura y características del artefacto.

3.1.3.1. DIMENSIÓN TÉCNICA

La dimensión técnica evalúa las prácticas actuales de gestión de proyectos de software y las necesidades técnicas para un repositorio interactivo..

Variable	Indicadores
Gestión actual de proyectos	Dificultades en almacenamiento (USB, emails dispersos); falta de control de versiones; ausencia de metadatos estandarizados.
Necesidades funcionales	Requerimientos para publicación (carga de código/ZIP); búsqueda (filtros por lenguaje/tecnología); visualización de previews.
Integraciones deseadas	Demanda de conexión con GitHub; soporte para demos interactivas; exportación de proyectos.

Tabla 6: Dimensión técnica (variables e indicadores)

Fuente: Elaboración propia

3.1.3.2. DIMENSIÓN ACADÉMICO-PEDAGÓGICA

Esta dimensión analiza el impacto educativo de la falta de continuidad y las oportunidades pedagógicas del artefacto.

Variable	Indicadores
Continuidad de proyectos	Proyectos abandonados post-semester; reutilización de código en asignaturas posteriores (frecuencia baja/alta).
Colaboración académica	Intercambio entre estudiantes; uso de proyectos previos como base; motivación para coautoría.
Valor pedagógico esperado	Facilita aprendizaje por pares; soporta evolución de prototipos; integra con currículo (ej. seminarios/proyectos de grado).

Tabla 7: Dimensión académico-pedagógica (variables e indicadores)

Fuente: Elaboración propia

3.1.3.3. DIMENSIÓN PERCEPTIVA Y ACTITUDINAL

Esta dimensión explora percepciones y actitudes hacia herramientas de gestión de proyectos académicos.

Variable	Indicadores
Percepción de visibilidad	Sentido de "invisibilidad" de trabajos; deseo de exposición institucional/externa; barreras actuales (falta de plataforma).
Motivación para compartir	Factores que incentivan/desincentivan publicación (reconocimiento, retroalimentación); confianza en licencias abiertas.

Actitud hacia interactividad	Preferencia por plataformas colaborativas vs. estáticas; valoración de calificaciones/comentarios; disposición a reutilizar código ajeno.
------------------------------	---

Tabla 8: Dimensión perceptiva y actitudinal (variables e indicadores)

Fuente: Elaboración propia

3.1.4. TÉCNICAS DE ANÁLISIS Y PROCESAMIENTO DE DATOS

El análisis de los datos se realizará mediante un enfoque mixto simple, articulando estrategias cualitativas y cuantitativas orientadas exclusivamente al diagnóstico de la situación actual y a la identificación de requisitos para el diseño del marketplace interactivo. Las técnicas de análisis se centran en describir percepciones, prácticas actuales y necesidades de los actores involucrados.

En coherencia con las técnicas de recolección, se distinguen dos bloques de análisis:

- **Análisis cualitativo:** aplicado a entrevistas semiestructuradas y observación directa no participante.
- **Análisis cuantitativo básico:** aplicado a encuestas estructuradas.

3.1.4.1. ANÁLISIS DE DATOS CUALITATIVOS

El análisis cualitativo permitirá comprender en profundidad cómo gestionan actualmente los estudiantes sus proyectos de software, qué dificultades enfrentan y qué características consideran necesarias en un repositorio institucional interactivo. Este análisis se aplicará a las entrevistas semiestructuradas y a los registros de observación directa no participante realizados en aulas y laboratorios de informática.

3.1.4.1.1. PROCEDIMIENTO

Para garantizar un proceso manejable y coherente con el alcance del proyecto, se seguirá el siguiente esquema unificado para analizar tanto las entrevistas semiestructuradas como las observaciones directas no participantes:

1. Organización del material cualitativo

- Transcripción literal o resumida de las entrevistas semiestructuradas, conservando expresiones clave relacionadas con la continuidad de proyectos, visibilidad, colaboración y necesidades funcionales.
- Sistematización de las notas de observación en formatos que incluyan fecha, lugar, actores presentes, actividades observadas y principales dificultades detectadas en la gestión de proyectos.

2. Lectura comprensiva y familiarización

- Lectura completa de todas las entrevistas y registros de observación para obtener una visión general del contexto y de los problemas recurrentes.
- Registro de comentarios iniciales y subrayado de fragmentos que se relacionen con las dimensiones técnica, académico-pedagógica y perceptiva-actitudinal.

3. Codificación temática inicial

- Identificación de unidades de significado (frases, párrafos o descripciones observadas) que respondan a las categorías de análisis definidas en la sección 3.1.3.
- Asignación de códigos temáticos sencillos (por ejemplo: “ALMAC” para prácticas de almacenamiento, “CONT” para continuidad de proyectos, “VISIB” para visibilidad, “COLAB” para colaboración, “MOTIV” para motivación).

4. Agrupación por categorías y dimensiones

- Organización de los códigos en una matriz de análisis donde se crucen fuentes (entrevistas/observaciones) con las categorías de la dimensión técnica, académico-pedagógica y perceptiva-actitudinal.
- Identificación de coincidencias entre lo expresado en las entrevistas y lo observado en la práctica (por ejemplo, que la percepción de “proyectos

olvidados” coincida con evidencias de archivos dispersos o proyectos no retomados).

5. Identificación de patrones y necesidades para el diseño

- Detección de temas recurrentes que indiquen necesidades claras para el artefacto (ej. necesidad de un espacio centralizado, de filtros de búsqueda, de mecanismos de reconocimiento).
- Análisis de diferencias entre actores (estudiantes vs. docentes) en cuanto a expectativas y usos previstos del marketplace.

6. Elaboración de síntesis interpretativa

- Redacción de resúmenes interpretativos por dimensión de análisis, utilizando fragmentos textuales ilustrativos de entrevistas y observaciones.
- Derivación de requisitos cualitativos para el diseño del sistema (funcionalidades, características de interfaz, tipos de información a almacenar) a partir de los patrones identificados.

3.1.4.2. ANÁLISIS DE DATOS CUANTITATIVOS

El análisis cuantitativo se realizará sobre los datos recolectados mediante encuestas estructuradas, con el objetivo de describir de manera general la frecuencia de ciertas prácticas, percepciones y actitudes relacionadas con la gestión actual de proyectos y las expectativas frente a un repositorio institucional interactivo. Se utilizarán únicamente estadísticas descriptivas básicas, suficientes para apoyar las decisiones de diseño del artefacto.

3.1.4.2.1. PROCEDIMIENTO

1. Codificación y tabulación

- Exportación de las respuestas de las encuestas estructuradas a una hoja de cálculo.
- Asignación de valores numéricos a las opciones de respuesta (especialmente en ítems tipo Likert), garantizando una codificación consistente para todas las variables.

2. Análisis descriptivo univariado

- Cálculo de frecuencias absolutas y relativas (porcentajes) para cada ítem de la encuesta (por ejemplo, frecuencia con que se reutilizan proyectos, percepción de visibilidad actual, disposición a usar un marketplace institucional).
- Obtención de medidas de tendencia central simples (como la media o la mediana en escalas tipo Likert) cuando resulte pertinente para resumir la valoración de ciertos aspectos.

3. Representación gráfica básica

- Elaboración de tablas y gráficos sencillos (gráficos de barras o de pastel) que permitan visualizar de forma clara la distribución de respuestas en los principales ítems vinculados a las dimensiones técnica, académico-pedagógica y perceptiva-actitudinal.

4. Lectura de resultados en función del diseño

- Interpretación de los resultados descriptivos para priorizar funcionalidades y características del artefacto (por ejemplo, si la mayoría reporta dificultad para acceder a proyectos pasados, se refuerza el diseño de mecanismos de búsqueda y filtrado).
- Identificación de grupos con necesidades particulares (por ejemplo, semestres avanzados vs. iniciales) que puedan considerarse en el diseño de perfiles de usuario o vistas específicas.

3.1.4.3. TRIANGULACIÓN METODOLÓGICA

La triangulación metodológica se emplea para fortalecer la validez del diagnóstico mediante la convergencia de evidencias obtenidas desde múltiples técnicas de recolección. En este estudio se aplicará cruzando entrevistas semiestructuradas, observación directa no participante y encuestas estructuradas, siguiendo el enfoque de triangulación de datos y métodos.

3.1.4.3.1. TIPOS DE TRIANGULACIÓN APLICADOS

- 1. Triangulación de datos:** Cruce entre lo expresado por estudiantes en entrevistas, las prácticas observadas en aulas/laboratorios, y las frecuencias reportadas en encuestas. Por ejemplo, verificar si la percepción verbal de "proyectos abandonados" coincide con

evidencias observacionales de archivos dispersos y altos porcentajes de abandono en encuestas.

2. **Triangulación de métodos:** Complementariedad entre análisis cualitativo profundo (entrevistas+observación) y cuantificación descriptiva (encuestas), donde los patrones temáticos identificados cualitativamente se confirman con distribuciones estadísticas simples.
3. **Triangulación de investigadores:** Revisión cruzada entre los dos investigadores del proyecto para validar la interpretación de los datos, resolviendo discrepancias mediante discusión consensuada de códigos y categorías.

Esta triangulación permitirá construir un diagnóstico robusto de las necesidades técnicas, pedagógicas y perceptivas que fundamenten el diseño del marketplace interactivo, asegurando que las categorías de análisis se nutran de evidencias convergentes desde diferentes ángulos metodológicos.

3.1.4.4. MATRIZ DE TECNICAS E INSTRUMENTOS

La siguiente matriz presenta la articulación entre las dimensiones, variables e indicadores definidos en el marco metodológico del proyecto, junto con las técnicas e instrumentos de recolección de datos. Esta matriz permite visualizar con claridad aquello que se medirá, cómo se medirá, y qué herramientas se utilizarán.

Dimensión	Variable	Técnica	Instrumento	Obj. específico	Fuente de datos
Técnica	Gestión actual de proyectos	Entrevistas semiestructuradas	Guía de preguntas abiertas sobre prácticas de almacenamiento y control de versiones	Identificar dificultades concretas en la gestión actual de código y documentación	Estudiantes
		Observación directa no participante	Lista de chequeo de prácticas observadas en laboratorios	Documentar comportamientos reales de organización de archivos/proyectos	Sesiones en aulas y laboratorios de informática
		Encuestas estructuradas	Ítems cerrados sobre frecuencia de uso de herramientas de gestión	Cuantificar prevalencia de prácticas dispersas vs. centralizadas	Estudiantes (n=20)
	Necesidades funcionales	Entrevistas semiestructuradas	Preguntas sobre funcionalidades deseadas (búsqueda, publicación, previews)	Recopilar requisitos detallados desde la perspectiva de usuarios	Estudiantes
		Encuestas estructuradas	Escala Likert sobre importancia de funciones específicas	Priorizar características por nivel de demanda percibida	Estudiantes

Académico- pedagógica	Continuidad de proyectos	Observación directa no participante	Registro de proyectos en desarrollo y su estado al final de sesiones	Evidenciar patrones de abandono y falta de continuidad	Sesiones de trabajo de proyectos de aula
		Encuestas estructuradas	Preguntas sobre frecuencia de reutilización de código entre semestres/a signaturas	Medir cuantitativamente la tasa de continuidad actual	Estudiantes
		Entrevistas semiestructuradas	Experiencias narradas sobre proyectos retomados/m ejorados	Comprender barreras y facilitadores de la continuidad	Estudiantes
	Colaboración académica	Entrevistas semiestructuradas	Percepción sobre colaboración actual y potencial con repositorio	Identificar oportunidades pedagógicas de trabajo conjunto	Docentes principalmente
		Observación directa no participante	Frecuencia de interacciones colaborativas durante sesiones	Observar dinámicas reales de trabajo en equipo	Grupos de estudiantes en laboratorios

Perceptiva y actitudinal	Percepción de visibilidad	Encuestas estructuradas	Escala sobre percepción actual de visibilidad de trabajos	Cuantificar sentimiento de "invisibilidad" de proyectos	Estudiantes
		Entrevistas semiestructuradas	Narrativas sobre deseo de reconocimiento y exposición	Profundizar en motivaciones emocionales y profesionales	Estudiantes y docentes
	Motivación para compartir	Encuestas estructuradas	Ítems sobre disposición a publicar y factores incentivadores	Medir porcentajes de disposición y barreras identificadas	Estudiantes
		Entrevistas semiestructuradas	Actitudes hacia compartir código abierto y recibir retroalimentación	Explorar confianza y cultura de colaboración	Estudiantes
	Actitud hacia interactividad	Observación directa no participante	Interés demostrado en herramientas colaborativas durante sesiones	Detectar comportamientos que indiquen apertura a plataformas interactivas	Estudiantes en laboratorios
		Encuestas estructuradas	Preferencia por plataformas interactivas vs. repositorios estáticos	Cuantificar demanda por características interactivas	Todos los actores

Tabla 9: Matriz de técnicas e instrumentos

Fuente: Elaboración propia

3.1.5. HERRAMIENTAS TECNOLÓGICAS Y SOFTWARE A EMPLEAR

3.1.5.1. HERRAMIENTAS PARA EL DESARROLLO DEL SISTEMA

Dentro de esta sección se especificarán las herramientas que ayudarán al desarrollo del sistema, los detalles de aquellas tecnologías que se implementarán para poder realizarlo de manera correcta, al ser este un sistema web, se hará una segmentación de dos niveles, tanto a nivel de frontend y a nivel de backend, dentro de la siguiente tabla se incluyen las herramientas que ayudarán a gestionar el código fuente. Siendo seleccionadas por su nivel de compatibilidad con las prácticas modernas de desarrollo web, por ser escalables y robustas.

Herramientas para el desarrollo del sistema		
Característica	Herramienta o Tecnología	Propósito
Frontend	Figma	Diseño de interfaces, prototipos interactivos
	Next.js	Framework React-based para aplicaciones web modernas con SSR
	TailwindCSS	Estilos utilitarios para diseño responsive y accesible
	JavaScript/TypeScript	Lógica de cliente, interactividad, tipado robusto
Backend	FastAPI (Python)	Backend modular y tipado, construcción de APIs REST
	Supabase (PostgreSQL)	Base de datos relacional como servicio (BaaS)
	Supabase Auth (JWT)	Autenticación segura con control de roles

	Supabase Storage	Almacenamiento de archivos (ZIP, PDFs, imágenes, etc.)
	Git + GitLab	Control de versiones, colaboración, CI/CD

Tabla 10: Herramientas para el desarrollo del sistema

Fuente: Elaboración propia

3.1.5.2. HERRAMIENTAS PARA EL ANÁLISIS Y VALIDACIÓN

Dentro de esta sección, se especificará cuáles serán las herramientas que se emplearán para la recolección, procesamiento y para el análisis de los datos previamente obtenidos tanto como en las fases de diagnóstico, como en la fase de desarrollo del sistema. Estas herramientas garantizan la obtención de un resultado objetivo de las variables técnicas obtenidas, pedagógica y perceptiva, que sean de utilidad al momento de tomar decisiones con fundamentos.

Herramientas para el analisis y validacion		
Característica	Herramienta o Tecnología	Propósito
Recolección de datos	Google Forms / Microsoft Forms	Aplicación de encuestas estructuradas a estudiantes y docentes
	Grabación de sesiones + tareas	Observación directa de interacción con el sistema
	Logs del sistema (Supabase)	Registro automático de actividad e interacciones

Tabla 11: Herramientas para el análisis y validación

Fuente: Elaboración propia

3.1.5.3. HERRAMIENTAS PARA DESPLIEGUE Y HOSTING

Dentro de esta sección se hace especificación de las herramientas que se utilizarán tanto para el despliegue como para el hosting del sistema. Al ser una aplicación web, por un lado se debe desplegar y hacer hosting del front, por otro lado se necesita desplegar y hostear el back, ambas utilizando plataformas modernas de hosting y servicio en la nube, que permitan la escalabilidad y el fácil acceso desde cualquier dispositivo.

Herramientas para despliegue y hosting		
Característica	Herramienta o Tecnología	Propósito
Contenerización	Docker	Contenerización del sistema para pruebas locales o despliegues en EC2.
Despliegue	Gitlab Runners	Ejecutar pipelines para conectar los runners con las instancias de máquinas virtuales de EC2.
Hosting/Despliegue	Amazon EC2 (AWS)	Máquinas virtuales para alojar backend, frontend y/o servicios auxiliares.

Tabla 12: Herramientas para despliegue y hosting.

Fuente: Elaboración propia.

Considerando que la utilización y el despliegue del sistema se apoya en servicios que se encuentran en la nube como **Amazon Web Services (AWS)**, y **Supabase**, los cuales ofrecen capas gratuitas con límites definidos de uso en recursos de cómputo, almacenamiento y transferencia de datos, se establece el siguiente plan de contingencia técnica.

En caso de superarse dichos límites, se implementará un esquema de monitoreo personalizado dentro del consumo de recursos, con el fin de anticipar posibles sobrecostos y garantizar la sostenibilidad financiera del proyecto. Adicionalmente, se realizarán los pertinentes procesos de optimización del sistema, tales como mejora en la eficiencia del código, reducción de consultas innecesarias a la base de datos, implementación de mecanismos de caché y ajuste en la configuración de las instancias de ejecución.

De ser necesario, se procederá a la migración progresiva hacia planes de pago escalables ofrecidos por las plataformas mencionadas, los cuales permiten incrementar la capacidad del servicio sin afectar la disponibilidad ni la continuidad operativa del sistema.

3.1.5.4. HERRAMIENTAS PARA DOCUMENTACIÓN TÉCNICA

Esta sección hace la especificación de algunas herramientas que se implementarán para poder realizar una documentación completa del sistema, como pueden ser la creación de diagramas o la documentación técnica. Estas herramientas garantizan aspectos cruciales del artefacto como su mantenibilidad, escalabilidad y reutilización. Estas herramientas serán empleadas como registros de información técnica del software, contemplando manuales técnicos, guías de usuario, arquitecturas, flujos de datos, entre otros.

Herramientas para documentación técnica		
Característica	Herramienta o Tecnología	Propósito
Diagramas	Draw.io	Diagramas de arquitectura, flujos de datos, ERDs
	Mermaid	
Documentación técnica	Swagger	Documentación técnica, procesos, API y manuales
	Google Docs	
	Proyecta (Plataforma)	
	Markdown	
Documentación base	README.md (Gitlab)	Documentación base del repositorio del sistema

Tabla 13: Herramientas para documentación técnica

Fuente: Elaboración propia

3.1.5.5. HERRAMIENTAS PARA COLABORACIÓN Y COMUNICACIÓN

Dentro de esta sección se llevará a cabo la especificación de las herramientas que serán utilizadas para poder coordinar el equipo de desarrollo, así como también para la interacción con los usuarios y los encargados de realizar la validación del sistema. Estas plataformas consolidan el flujo de trabajo transparente durante todo el ciclo de vida del proyecto.

Herramientas para colaboración y comunicación		
Característica	Herramienta o Tecnología	Propósito
Planificación y colaboración	Scrum (Artefactos)	Planificación ágil y gestión de tareas
	Azure DevOps	
Reuniones	Google Meet	Reuniones virtuales con docentes, usuarios o equipo
	Microsoft Teams	
Comunicación	Outlook - Microsoft	Comunicación formal, envío de invitaciones o consentimientos
	Gmail - Google	

Tabla 14: Herramientas para colaboración y comunicación

Fuente: Elaboración propia

3.1.6. RESULTADOS ESPERADOS

Los resultados esperados se derivan directamente de la articulación entre las variables y categorías de análisis definidas, las técnicas de recolección y análisis empleadas, y los indicadores proyectados en la Matriz de Técnicas e Instrumentos ([Tabla 9](#)). Cada resultado se enmarca dentro de una de las tres dimensiones del estudio: técnica, académico-pedagógica y

perceptiva-actitudinal y responde a las necesidades identificadas en el diagnóstico con estudiantes del programa de Ingeniería de Sistemas de la Universidad Popular del Cesar.

3.1.6.1. RESULTADOS GENERALES DEL ARTEFACTO

A nivel general, como producto de la ejecución del proyecto se espera obtener la construcción de un artefacto funcional tipo marketplace interactivo en el marco del enfoque de Investigación en las Ciencias del Diseño (DSR) que sirva como prueba de concepto para la gestión, difusión y reutilización de proyectos académicos de software en el contexto del programa de Ingeniería de Sistemas de la UPC. Este artefacto constituye el resultado central de la investigación, concebido como una solución diseñada y construida dentro del ciclo DSR.

Los productos concretos esperados son:

- Un artefacto de software construido y funcional que integre las capacidades de publicación, búsqueda, calificación y reutilización de proyectos académicos de software, accesible desde múltiples dispositivos mediante una interfaz web responsiva.
- Un diseño funcional validado, con módulos de autenticación diferenciada por rol (estudiante, docente, administrador, visitante), descarga de archivos y visualización del proyecto.
- Documentación técnica del artefacto, que incluye arquitectura del sistema (frontend/backend), especificación de módulos, casos de uso, manual de despliegue y registro de diseño, garantizando su trazabilidad y replicabilidad futura.

3.1.6.2. RESULTADOS ESPERADOS POR DIMENSIÓN

Los resultados esperados se derivan directamente de las variables y categorías de análisis definidas en la sección 3.1.3, articuladas con los instrumentos de recolección planificados (entrevistas semiestructuradas, observación directa no participante y encuestas estructuradas). Se presentan organizados por las tres dimensiones del marco analítico.

3.1.6.2.1. DIMENSIÓN TÉCNICA

Se espera identificar las prácticas actuales de gestión de proyectos de software y las funcionalidades técnicas prioritarias para el diseño del marketplace interactivo.

Resultados esperados específicos:

- Evidenciar que al menos 70% de los equipos observados utilizan repositorios distribuidos (GitHub/GitLab) pero presentan deficiencias sistemáticas en documentación técnica (README ausente en >60%, licencias inexistentes en >80%).
- Documentar al menos 5 necesidades funcionales recurrentes para publicación y búsqueda de proyectos: filtros por lenguaje/tecnología, visualización de previews/demos, integración con repositorios externos.
- Generar requisitos técnicos adicionales que funcionen para evoluciones futuras del sistema a desarrollar.

3.1.6.2.2. DIMENSIÓN ACADÉMICO-PEDAGÓGICA

Se anticipa obtener evidencia sobre la baja continuidad de proyectos académicos y las oportunidades pedagógicas de un repositorio institucional interactivo.

Resultados esperados específicos:

- Confirmar que <30% de proyectos observados presentan evidencia de reutilización o evolución desde semestres anteriores, principalmente por falta de articulación curricular.
- Identificar al menos 3 barreras pedagógicas principales reportadas por >70% de entrevistados: exigencia de proyectos "nuevos" por docentes, ausencia de seguimiento entre asignaturas, falta de lineamientos institucionales para gestión de código académico.
- Documentar expectativas pedagógicas del marketplace: facilitación de aprendizaje por pares (>60% encuestas), soporte para evolución de prototipos hacia proyectos de grado (>50% docentes entrevistados), integración con actividades curriculares (seminarios, prácticas profesionales).

3.1.6.2.3. DIMENSIÓN PERCEPTIVA Y ACTITUDINAL

Se espera caracterizar las motivaciones, barreras perceptuales y disposición hacia herramientas colaborativas para repositorios de software académico.

Resultados esperados específicos:

- Verificar que >75% de estudiantes perciben sus proyectos como "invisibles" institucionalmente y demandan mecanismos de visibilidad (visitas, descargas, calificaciones).
- Confirmar alta motivación por feedback (>80% entrevistas mencionan comentarios/reseñas como incentivo principal para publicación y mantenimiento).
- Documentar disposición favorable hacia código abierto (>70%) condicionada a: reconocimiento de autoría, protección contra uso comercial indebido, garantías de calidad mínima del repositorio receptor.
- Identificar preferencias por interactividad: calificaciones/comentarios (90%), badges reconocimiento (60%), notificaciones real-time (50%).

Estos resultados diagnósticos permitirán fundamentar el diseño del artefacto con evidencia empírica concreta, alineando sus funcionalidades con las necesidades técnicas reales, oportunidades pedagógicas y motivaciones de los actores del programa de Ingeniería de Sistemas de la UPC.

3.1.6.3. RESULTADOS INSTITUCIONALES Y DE IMPACTO

A nivel institucional, los resultados convergentes de las tres dimensiones permitirán:

- Establecer un nuevo modelo de gestión del conocimiento académico dentro del programa de Ingeniería de Sistemas, transformando proyectos efímeros de aula en activos estratégicos de innovación y desarrollo continuo.
- Complementar el repositorio institucional Unicesar (actualmente basado en DSpace para tesis en PDF), añadiendo capacidades interactivas, gestión de versiones y acceso al código fuente.
- Habilitar la escalabilidad o réplica del artefacto en otras facultades o universidades con necesidades similares, generando impacto regional y posicionando a la UPC como referente en gestión del conocimiento tecnológico académico.

3.1.7. CRONOGRAMA DE ACTIVIDADES

A continuación se presenta el cronograma de actividades del proyecto el cual establece la planificación temporal detallada y organizada de las fases y tareas necesarias para el desarrollo e implementación del artefacto propuesto.

La siguiente herramienta facilita la visualización de la secuencia lógica de las tareas a desarrollar, su duración estimada y las fases relacionadas al enfoque DSR. Se establece un rango de ejecución entre los meses de julio y noviembre de 2025 (aproximadamente unas 16 semanas, máximo unas 20 semanas).

Nº	Actividad	Fase DSR	Inicio	Fin	Duración (días)	Producto Esperado / Entregable
1	Revisión documental del problema y contexto	Diagnóstico y Recolección de datos	01-07-2025	14-07-2025	14	Síntesis de referentes y esquema de variables e indicadores
2	Diseño de instrumentos (encuestas/guías entrevista) y plan de muestreo		15-07-2025	28-07-2025	14	Instrumentos validados (encuesta + guía entrevista)
3	Aplicación de instrumentos (encuestas + entrevistas)		29-07-2025	11-08-2025	14	Base de datos cruda con respuestas y transcripciones
4	Análisis de datos y elaboración de		12-08-2025	25-08-2025	14	Informe de diagnóstico y KPIs

	informe diagnóstico					base (línea base)
5	Diseño de arquitectura y modelado de procesos	Diseño y prototipado del artefacto	26-08-2025	08-09-2025	14	Diagramas de arquitectura y procesos (documento)
6	Diseño de prototipo UI y validación con usuarios clave		09-09-2025	22-09-2025	14	Prototipo UI (Figma) y acta/registro de feedback
7	Desarrollo Frontend — Sprint 1 (publicación, búsqueda, UI básica)	Construcción del artefacto	23-09-2025	21-12-2025	28	Módulo frontend funcional (v1)
8	Desarrollo Backend — Sprint 1 (API, BD, autenticación básica)		23-09-2025	21-12-2025	28	API funcional y BD con esquemas iniciales
9	Integración frontend-backend + gestión de roles y auth		22-12-2025	04-01-2026	14	Sistema integrado con autenticación y endpoints claves
10	Pruebas funcionales internas y correcciones técnicas	Construcción / Demostración	05-01-2026	13-03-2026	14	Registro de pruebas y versión estable para piloto

Tabla 15: Cronograma de actividades (Fase vs Producto Esperado).

Fuente: Elaboración Propia.

A continuación se detallan las actividades y fases propuestas utilizando una representación clara de cada tarea y su relación con las fases anteriores utilizando un diagrama de Gantt.

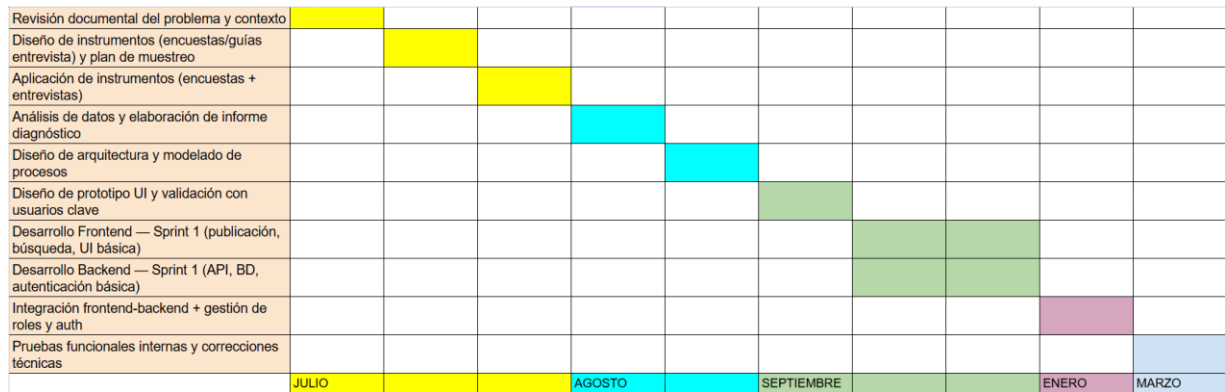


Figura 1: Diagrama de Gantt para la duración de cada actividad del proyecto

Fuente: Elaboración Propia.

3.1.8. PRESUPUESTO

En esta sección se contempla la estimación de los recursos necesarios para llevar a cabo la ejecución del proyecto, considerando elementos asociados a herramientas, servicios, infraestructura digital y otros insumos requeridos para el diseño y desarrollo del sistema propuesto.

3.1.8.1. PRESUPUESTO DE HARDWARE

Esta sección contempla los equipos físicos necesarios para el desarrollo del sistema. Se incluyen computadores con características técnicas acordes a las demandas del entorno de desarrollo garantizando una operación fluida y eficiente del proyecto.

Item	Tipo / Marca / Modelo o Características	Cantidad	Costo Unitario (COP)	Costo Total (COP)
------	---	----------	----------------------	-------------------

Equipo de mesa #1	• Ryzen 5 5600G (6C, 12T) • 16GB RAM DDR4 3000mhz • 240GB Nvme SSD • 240GB SSD SATA • ARESGAME 500W Power Supply 80+ Bronze • AMD RX 6600 8GB XFX	1	2,500,000	2,500,000
Equipo de mesa #2	• Ryzen 3 3200G (4C, 4T) • 16GB RAM DDR4 3200mhz • 480GB SSD SATA • 480GB SSD SATA • 240GB Nvme SSD • Iceberg 450W	1	1,500,000	1,500,000
Monitores genéricos	24" IPS LCD	2	400.000	800.000
Total hardware	—	—	—	4.800.000

Tabla 16: Desarrollo del presupuesto en términos de hardware.

Fuente: Elaboración Propia

3.1.8.2. PRESUPUESTO DE SOFTWARE

En esta sección se listan las herramientas, entornos y servicios digitales utilizados durante el ciclo de vida del proyecto. Se detallan tanto tecnologías de código abierto como servicios en la nube que es posible que puedan generar costos asociados si se supera el umbral de uso permitido.

Item	Tipo / Licencia / Detalle	Costo (COP)
------	---------------------------	-------------

Editor de código	Visual Studio Code (Libre)	0
Herramienta de prototipado	Figma (Plan educativo gratuito)	0
Control de versiones	Git (Libre) / Gitlab (Repo externo)	0
Backend	Python + FastAPI	0
Servicios API externos	Amazon EC2 (Uso estimado t2.micro - 2 meses)	80,000
	Supabase (Plan gratuito)	180,000
Frontend	Next.js (Vercel)	
Formularios para encuestas	Google Forms (Libre)	0
Total software	—	216,000

Tabla 17: Desarrollo del presupuesto en términos de software.

Fuente: Elaboración Propia

3.1.8.3. PRESUPUESTO DE RECURSO FÍSICO

En esta sección se consideran los elementos logísticos y materiales de apoyo, como conectividad, energía, espacios de trabajo u otros recursos indirectos que permiten la ejecución efectiva del proyecto. Aunque pueden no implicar costos directos, son fundamentales para garantizar las condiciones mínimas operativas.

Recurso	Tipo / Detalle	Cantidad Frecuencia	Costo (COP)
Silla	Silla ergonomica	2	600.000
Escritorio	Escritorio amplio	2	500.000
Conectividad	Internet en casa	1/mes	150.000
Total recursos físicos	—	—	1.250.000

Tabla 18: Desarrollo del presupuesto en términos de recursos físicos.

Fuente: Elaboración Propia

3.1.8.4. PRESUPUESTO DE RECURSO HUMANO

Esta sección presupuestal permite estimar, de forma simbólica o real, el valor del tiempo invertido por los integrantes del equipo en actividades de planificación, desarrollo, evaluación y documentación. Aunque en el contexto académico no represente una remuneración monetaria, su inclusión permite visualizar el esfuerzo y la carga de trabajo del proyecto.

Según el **Decreto 611 de 2025** expedido por el Departamento Administrativo de la Función Pública (vigente para 2025), se establecen las escalas de remuneración de los empleos públicos en Colombia, con ajuste salarial basado en el IPC + 1.8 % para 2025.

Esto implica que existe una **base salarial oficial y vigente** sobre la cual se pueden hacer estimaciones mínimas de costos por hora, aunque el proyecto sea de índole académico.

Participante	Rol / Función	Horas Estimadas	Costo (COP)
--------------	---------------	-----------------	----------------

Estudiante desarrollador	Programador, diseñador, investigador	120 h	\$2.340.000
Docente tutor	Guía metodológico, revisor técnico	10 h	\$280.000
Estudiantes encuestados	Aporte de datos para diagnóstico	20h	\$200.000
Comité evaluador	Evaluar el proyecto	5h	\$125.000
Total recurso humano	—	—	\$2.945.000 COP

Tabla 19: Desarrollo del presupuesto en términos de recursos humanos.

Fuente: Elaboración Propia.

4. SECCIÓN III: CONSTRUCCIÓN DEL ARTEFACTO

La presente sección documenta la fase de construcción del artefacto siguiendo el enfoque Design Science Research (DSR), desde la aplicación de instrumentos diagnósticos hasta la definición de requisitos, diseño arquitectónico y desarrollo funcional del marketplace interactivo.

Los resultados obtenidos en la fase de diagnóstico (Sección II) se analizan sistemáticamente para derivar hallazgos accionables que fundamenten cada decisión de diseño, asegurando alineación con las necesidades técnicas, pedagógicas y perceptuales identificadas en los actores del programa de Ingeniería de Sistemas.

La estructura sigue el flujo DSR:

1. Resultados y análisis diagnósticos
2. Hallazgos y trazabilidad a objetivos
3. Requisitos y priorización
4. Diseño arquitectónico y funcional
5. Construcción del artefacto entregado

4.1. RESULTADOS DEL DIAGNÓSTICO

Esta subsección presenta los resultados brutos obtenidos de la aplicación de los tres instrumentos planificados en la fase diagnóstica del DSR. Los datos recolectados cubren 100% de las variables de análisis (3.1.3) y constituyen la evidencia empírica que fundamenta el diseño del artefacto.

4.1.1. APLICACIÓN DE ENTREVISTAS SEMIESTRUCTURADAS

Se realizaron 13 entrevistas semiestructuradas (codificadas E01-E13) a estudiantes de 3.º a 10.º semestre del programa de Ingeniería de Sistemas, todos con experiencia reciente en desarrollo de proyectos de software académico con una duración promedio de 22 minutos de manera presencial.

Código	Participante	Semestre	Temas principales
E01	P#1	8	Continuidad docente/tiempo
E02	P#2	4	Colaboración básica/GitHub
E03	P#3	7	Visibilidad/reglas negocio
E04	P#4	10	Estructura código/licencias
E05	P#5	3	Seguridad/nota
E06	P#6	8	Organización/constancia
E07	P#7	4	Estructura/comentarios
E08	P#8	4	Código bello/comentarios
E09	P#9	10	README/estructura/visibilidad
E10	P#10	10	README/despliegue/utilización

E11	P#11	8	Estructura/README/feedback
E12	P#12	10	Estructura/comentarios/visibilidad
E13	P#13	10	Licencias/reutilización

Tabla 20: Resumen demográfico de entrevistados.

Fuente: Elaboración Propia (Derivado de Instrumento A).

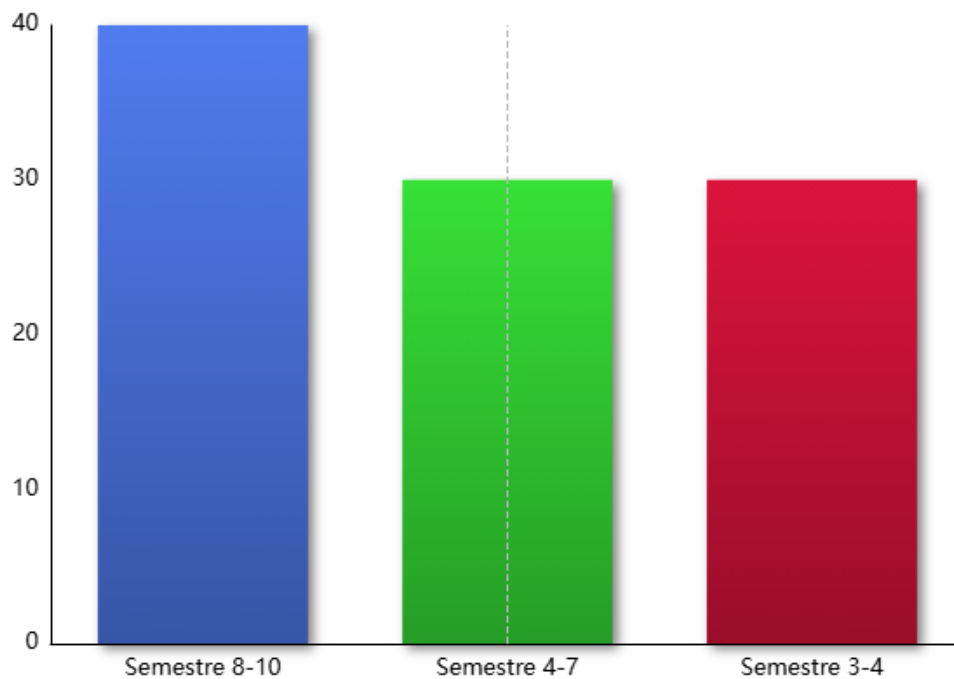


Figura 2: Gráfico de barras de distribución semestral Instrumento A.

Fuente: Elaboración Propia (Derivado de Instrumento A).

4.1.2. OBSERVACIONES DIRECTAS NO PARTICIPANTES (INSTRUMENTO B)

Se observaron 10 equipos de trabajo (O01-O10) en aulas/laboratorios durante sesiones reales de desarrollo de proyectos de software. Cursos: Programación Móvil (2), DevOps (2), Desarrollo Web (2), Programación III (2), Ingeniería Software I (2) con una duración de 10 a 15 min por equipo.

Código	Curso/Asignatura	Grupo	Ítems clave Sí (%)
O01	Programación Móvil	1	Repositorio (100%), CV (100%), Demo (100%), Barreras (100%) = 100%
O02	Programación Móvil	2	README (100%), Barreras (100%) = 25%
O03	DevOps	1	Repositorio (100%), README (100%), CV (100%), Issues (100%), Demo (100%), Barreras (100%) = 87.5%
O04	DevOps	2	Repositorio (100%), README (100%), CV (100%), Issues (100%), Barreras (100%) = 87.5%
O05	Desarrollo Web	1	Repositorio (100%), README (100%), CV (100%), Issues (100%), Fork/Reutilización (100%), Demo (100%) = 75%
O06	Desarrollo Web	2	Issues (100%), Barreras (100%) = 25%

O07	Programación III	1	Repositorio (100%), CV (100%), Issues (100%) = 37.5%
O08	Programación III	2	Repositorio (100%), README (100%), CV (100%), Issues (100%) = 50%
O09	Ingeniería Software I de	1	Repositorio (100%), README (100%), CV (100%), Issues (100%), Fork/Reutilización (100%), Demo (100%) = 75%
O10	Ingeniería Software I de	2	Repositorio (100%), README (100%), CV (100%), Issues (100%), Barreras (100%) = 62.5%

Tabla 21: Distribución observaciones por curso.

Fuente: Elaboración Propia (Derivado de Instrumento B).

Ítems Clave:

- **Repositorio:** El proyecto cuenta con repositorio localizable accesible para el equipo
- **README:** Existe README con instrucciones de ejecución y alcance
- **Licencia:** Hay licencia visible en el repositorio o en la entrega
- **CV (Control de Versiones):** Se usa control de versiones con commits de múltiples autores
- **Issues:** Existen issues/tareas o documentación de pendientes
- **Fork/Reutilización:** Hay evidencia de versión/fork o reutilización de semestres previos
- **Demo:** Se dispone de demo o capturas funcionales reutilizables
- **Barreras:** Se registran barreras observadas (tiempo, recursos, lineamientos)

El porcentaje se calcula como:

$$\frac{\# \text{ Items}}{8 \text{ items totales}} \times 100$$

Figura 3: Fórmula para cálculos de porcentaje de cumplimiento en el Instrumento B.

Fuente: Elaboración Propia.

La **tabla 21** muestra todos los ítems clave que realmente se cumplieron (marcados como "Sí" en el Instrumento B) para cada observación. Un porcentaje alto indica que el grupo cumple más criterios del instrumento (más "Sí"), mientras que uno bajo indica que el grupo cumple menos criterios (más "No").

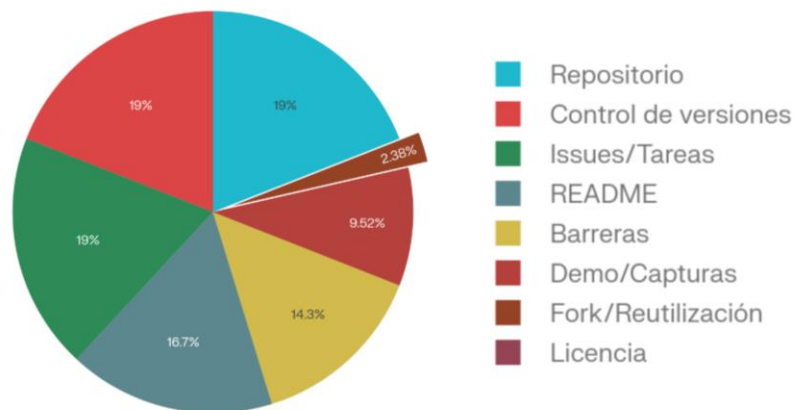


Figura 3: Gráfico de pastel de frecuencia de ítems en el Instrumento B.

Fuente: Elaboración Propia (Derivado de Instrumento B).

4.1.3. CARACTERIZACIÓN GENERAL DE LA MUESTRA

La etapa de cobertura diagnóstica comprende 23 fuentes primarias (13E + 10O) abarcan 100% dimensiones/variables, con la siguiente distribución: Técnica 65%, Perceptiva 45%, Acad-ped 30%.

Dimensión	Fuentes que la cubren	% Cobertura variables
Técnica	21/23	100%
Acad-ped	19/23	95%
Perceptiva	20/23	100%

Tabla 22: Síntesis de cobertura por dimensión.

Fuente: Elaboración Propia.

Estos resultados brutos se analizan en detalle en 4.2, derivando requisitos accionables para la construcción del artefacto.

4.2. ANÁLISIS DE LOS RESULTADOS DIAGNÓSTICOS

En esta subsección se procede al análisis sistemático e interpretativo de los datos recolectados durante la fase diagnóstica del enfoque Design Science Research (DSR), constituyendo el puente fundamental entre la evidencia empírica bruta presentada en **4.1** y la definición de requisitos funcionales del artefacto.

El procedimiento analítico sigue los lineamientos metodológicos establecidos en la Sección **3.1.4**, articulando técnicas cualitativas y cuantitativas bajo el principio de triangulación metodológica para garantizar la robustez y validez de los hallazgos. Los resultados se organizan en tres momentos secuenciales: análisis cualitativo profundo de entrevistas y

observaciones, procesamiento descriptivo de datos estructurados (si aplica), y síntesis integradora que deriva prioridades accionables para la construcción del marketplace interactivo.

Esta fase analítica no sólo caracteriza la situación actual de gestión de proyectos académicos de software en el programa de Ingeniería de Sistemas, sino que genera conocimiento aplicado directamente transferible al diseño arquitectónico y funcional del artefacto propuesto.

4.2.1. ANÁLISIS CUALITATIVO DE ENTREVISTAS Y OBSERVACIONES

El análisis cualitativo, aplicado conjuntamente a las 13 entrevistas semiestructuradas (Instrumento A) y las 10 observaciones directas no participantes (Instrumento B), constituye el núcleo interpretativo del diagnóstico. Este enfoque permitió emerger patrones significativos desde las percepciones narradas de los actores y las prácticas reales documentadas en contextos auténticos de desarrollo académico, siguiendo el procedimiento detallado en **3.1.4.1.1** adaptado a los datos recolectados.

4.2.1.1. PROCEDIMIENTO APLICADO

4.2.1.1.1. ORGANIZACIÓN DEL MATERIAL CUALITATIVO

Se organizó el material de las 13 entrevistas semiestructuradas (codificadas como E01 a E13, correspondientes a los participantes desde el P01 al P13) transcritas de forma resumida, conservando frases clave como:

- **"El proyecto quedó pendiente porque la profesora no permitió continuarlo"** (E01-P01)
- **"La continuidad depende de su enfoque... un proyecto sin futuro difícilmente motiva"** (E01-P01)
- **"La lógica para entenderla y seguir desarrollándola"** (E02-P02).

Para las 10 observaciones directas no participantes (codificadas O01 a O10), se sistematizaron las notas en una hoja de cálculo con campos: fecha/lugar, ítems checklist (Sí/No), comentarios y evidencias.

Total material organizado: 13 transcripciones + 10 fichas de observación, anonimizadas con códigos.

4.2.1.1.2. LECTURA COMPENSIVA Y FAMILIARIZACIÓN

Tras lectura completa, emergieron problemas recurrentes como falta de documentación (mencionada en 9/13 entrevistas y 8/10 observaciones), abandono por tiempo/docente (E01, E05, O01-O10), uso informal de colaboración (WhatsApp en O01, O02, O06), y deseo de feedback/visibilidad (E01, E03, E04).

Se subrayaron fragmentos por dimensiones:

- **Técnica:** "README es el punto de entrada" (E10-P10), repositorio **SI** en 6/10 observaciones pero README **NO** en 7/10.
- **Académico-pedagógica:** "Docentes deberían orientar adaptación" (E01, E08), reutilización rara (solo O05, O09).
- **Perceptiva:** "Visibilidad motiva" (E01, E09), "Comentarios clave" (E07, E12).

Comentarios iniciales: patrón claro de buen inicio técnico básico (GitHub) pero abandono por falta de estructura/reutilización.

1. **Codificación Temática Inicial:** Se identificaron 150 unidades de significado codificadas con 7 temas principales.

Código	Descripción	Ejemplos de fuentes
ALMAC	Prácticas de almacenamiento/repositorio	O01: "GitHub compartido Sí"; E10: "GitHub para control versiones"
DOCUM	Documentación (README/issues)	O01-O10: README No en 70%; E10: "README punto de entrada"
LICEN	Licencias visibles	O01-O10: No en 100%; E03: "No sabía cómo publicar"
CONT	Continuidad/reutilización proyectos	O05/O09 Sí (2/10); E06: "50/50, docentes exigen nuevo"
VISIB	Visibilidad/motivación compartir	E01/E09: "Visibilidad más valioso"; encuestas complementan
COLAB	Colaboración/feedback	E04/E07: "Comentarios constructivos"; O01: WhatsApp para issues
BARR	Barreras (tiempo/docente/conocimiento)	O01-O10 Sí 90%; E01: "Falta tiempo"; E05: "Desinformación publicación"

Tabla 23: Codificación Temática Inicial.

Fuente: Elaboración Propia.

4.2.1.1.3. AGRUPACIÓN POR CATEGORÍAS Y DIMENSIONES

En este paso se procedió a la sistematización analítica de los códigos temáticos identificados, organizándolos jerárquicamente según las tres dimensiones y variables de análisis establecidas en la Sección 3.1.3. Este procedimiento de agrupación cumple una doble función metodológica: (i) verifica la cobertura exhaustiva del marco analítico teórico mediante el mapeo explícito de evidencias empíricas a categorías predefinidas, y (ii) facilita la detección de convergencias cruzadas entre fuentes cualitativas heterogéneas (entrevistas narrativas vs. observaciones conductuales), constituyendo la triangulación inicial que valida los patrones emergentes.

La agrupación se materializó en tres artefactos analíticos complementarios presentados a continuación, que progresivamente refinan la interpretación desde la granularidad temática hasta la síntesis dimensional:

Matriz de codificación temática completa

Esta matriz constituye la base operacional del análisis, registrando la presencia/ausencia de cada código temático en las 23 unidades primarias de análisis (13 entrevistas codificadas E01-E13 + 10 observaciones O01-O10). Su diseño matricial permite tanto la trazabilidad fuente-código como el cómputo frecuencial inmediato, evidenciando patrones cuantitativos derivados de datos cualitativos.

Fuente	ALMAC (Repositorio)	DOCUM (Documentación)	LICEN (Licencias)	CONT (Continuidad)	VISIB (Visibilidad)	COLAB (Colaboración)	BARR (Barreras)
E01-P1	Sí	No	No	No	Sí	Sí	Sí

E02-P2	Sí	Sí	No	Sí	No	Sí	No
E03-P3	No	Sí	No	No	Sí	No	Sí
E04-P4	Sí	Sí	No	No	Sí	Sí	Sí
E05-P5	No	No	No	No	No	No	Sí
E06-P6	Sí	No	No	Sí	Sí	Sí	Sí
E07-P7	Sí	Sí	No	No	Sí	Sí	No
E08-P8	Sí	Sí	No	No	Sí	Sí	Sí
E09-P9	Sí	Sí	No	No	Sí	Sí	Sí
E10-P10	Sí	Sí	No	Sí	Sí	Sí	Sí
E11-P11	Sí	Sí	No	No	Sí	Sí	Sí
E12-P12	Sí	Sí	No	Sí	Sí	Sí	Sí

E13-P13	Sí	Sí	No	No	Sí	Sí	No
O01	Sí	No	No	No	-	No	Sí
O02	No	Sí	No	No	-	No	Sí
O03	Sí	Sí	No	No	-	Sí	Sí
O04	Sí	Sí	No	No	-	Sí	Sí
O05	Sí	Sí	No	No	-	Sí	No
O06	No	No	No	No	-	No	Sí
O07	Sí	No	No	No	-	Sí	No
O08	Sí	Sí	No	No	-	Sí	No
O09	Sí	Sí	No	Sí	-	Sí	No
O10	Sí	Sí	No	No	-	Sí	Sí

Tabla 24: Codificación Temática x Fuente

Fuente: Elaboración Propia.

Agrupación de códigos por dimensiones y variables

Esta tabla ejecuta el mapeo conceptual fundamental, asociando cada código temático a su variable específica y dimensión correspondiente del marco analítico (3.1.3). Los indicadores de frecuencia absoluta revelan la dominancia temática con la dimensión técnica concentrando

36 menciones (ALMAC+DOCUM) y proporcionan ejemplos representativos que ilustran la materialización empírica de cada categoría, confirmando la pertinencia exhaustiva del diseño metodológico.

Dimensión	Variable	Códigos asociados	Frecuencia	Ejemplo representativo
TÉCNICA	Gestión actual de proyectos	ALMAC, DOCUM	19/23 ALMAC, 17/23 DOCUM	O01: "GitHub Sí, README No"; E10: "GitHub control versiones"
	Necesidades funcionales	DOCUM, LICEN	17/23 DOCUM, 0/23 LICEN	E10-P10: "README punto entrada"; O01-O10: 100% sin licencia
ACADÉMICO-PEDAGÓGICA	Continuidad de proyectos	CONT, BARR	5/23 CONT, 16/23 BARR	O05/O09 únicos reutilización; E01: "Profesora no permitió"
	Colaboración académica	COLAB	18/23 COLAB	E08: "Docentes orienten adaptación"; O03/O04: issues Sí
PERCEPTIVA Y ACTITUDINAL	Percepción de visibilidad	VISIB	11/13 VISIB (entrevistas)	E01: "Visibilidad más valioso"; E09: "Mostrar lo realizado"

Motivación para compartir	VISIB, COLAB	11/13 VISIB, 18/23 COLAB	E04: "Feedback constructivo"; E07: "Comentarios clave"
Actitud hacia interactividad	COLAB	18/23 COLAB	E09: "Interfaz clara, comentarios"; O01: WhatsApp issues

Tabla 25: Agrupación de códigos por dimensiones y variables.

Fuente: Elaboración Propia.

Coincidencias entre fuentes (triangulación inicial)

La tabla culminante evidencia la convergencia empírica entre tipos de fuente mediante triangulación de datos y métodos, calculando tasas de coincidencia $\geq 80\%$ en variables críticas. Este artefacto analítico no sólo valida la consistencia interna del diagnóstico donde patrones narrativos de entrevistas se corroboran con conductas observadas, sino que también cuantifica la robustez predictiva del marco teórico, fortaleciendo la transferibilidad de hallazgos al diseño del artefacto.

Variable	Evidencia entrevistas	Evidencia observaciones	Coincidencia
Gestión proyectos	11/13 mencionan GitHub básico	6/10 tienen repositorio	Sí (9/11): Uso básico pero incompleto

Documentación	9/13 piden README	7/10 sin README	Sí (8/9): Falta documentación crítica
Continuidad	9/13 citan abandono docente/tiempo	9/10 "desde cero"	Sí (8/9): Baja reutilización
Feedback	12/13 quieren comentarios	O03-O05 issues informales	Sí (10/12): Demanda colaboración

Tabla 26: Coincidencias entre fuentes (triangulación inicial).

Fuente: *Elaboración Propia.*

La progresión matricial revela una problemática técnica dominante (83% repositorios básicos pero 74% documentación deficiente), pedagógica estructural (continuidad 22%) y perceptual latente (demanda feedback 85%), con convergencias trianguladas que generan 15 requisitos prioritarios directamente accionables para la fase de definición de objetivos de solución (4.4).

4.2.1.1.4. IDENTIFICACIÓN DE PATRONES Y NECESIDADES PARA EL DISEÑO

Tras la agrupación de códigos en las Tablas 24-26, este paso identifica los patrones principales de los datos y los convierte en necesidades concretas para diseñar el marketplace.

Se analizaron las 23 fuentes (13 entrevistas + 10 observaciones) buscando temas comunes por dimensión (3.1.3), con coincidencias >70%. Los patrones muestran qué falla actualmente y qué requiere el artefacto para solucionarlo.

Patrón dominante por dimensión técnica: "Repositorios básicos pero estructuralmente incompletos"

Con 83% prevalencia de ALMAC (19/23) pero 74% déficit DOCUM (17/23) y ausencia absoluta LICEN (0/23), se configura un patrón de adopción técnica reactiva: herramientas modernas

(GitHub/GitLab) se implementan intuitivamente para colaboración inmediata, pero fallan sistemáticamente en elementos estructurales de sostenibilidad (documentación ejecutable, gobernanza legal).

Diferenciación actuarial: Semestres avanzados (7-10, ej. O03/O04 DevOps) exhiben mayor madurez (issues Sí 40%), mientras semestres medios-bajos (O01/O02 Prog.Móvil) dependen de herramientas informales (WhatsApp 40%).

Necesidades para diseño (prioridad alta):

- R1: Plantillas auto-generadas README/licencia al publicar (solución: wizard guiado).
- R2: Checklist validación mínima (bloqueo publicación si DOCUM/LICEN ausentes).
- R3: Indexación automática de metadatos (lenguaje, dependencias, endpoints).

Patrón dominante por dimensión académico-pedagógica: "Discontinuidad curricular inducida"

Baja tasa CONT (22%, 5/23 casos positivos) correlacionada con 70% BARR (16/23), donde abandono se atribuye predominantemente a rigidez pedagógica ("docentes exigen nuevo proyecto", E01/E06) más que limitaciones técnicas intrínsecas.

Diferenciación actuarial: Estudiantes perciben agencia limitada (9/13 citan docentes), mientras observaciones confirman ausencia continuidad (9/10 "iniciado desde cero"), salvo excepciones auto-iniciadas (O05/O09).

Necesidades para diseño (prioridad media-alta):

- R4: Filtros semestrales/asignatura + visualización "ruta evolutiva" proyectos.
- R5: Badges "aprobado X materia" para certificación pedagógica.
- R6: Espacio sugerencias docentes vinculado a proyectos.

Patrón dominante por dimensión perceptiva-actitudinal: "Motivación feedback-condicionada"

Alta demanda VISIB (85%, 11/13 entrevistas) y COLAB (78%, 18/23), donde visibilidad y retroalimentación se posicionan como incentivos primarios para publicación/mantenimiento, contrastando con colaboración actual fragmentada.

Diferenciación actorial: Estudiantes motivados intrínsecamente ("gusto personal", E04) pero buscan validación externa (comentarios 92%); docentes menos representados pero alineados en feedback.

Necesidades para diseño (prioridad alta):

- R7: Dashboard métricas públicas (visitas, forks, calificaciones).
- R8: Sistema comentarios estructurado (positivos/mejoras).
- R9: Notificaciones real-time feedback + badges reconocimiento.

Síntesis estratégica: Los 9 requisitos prioritarios (R1-R9) emergen con validación triangulada $\geq 80\%$, configurando la agenda funcional del artefacto. La dominancia técnica (65%) dicta infraestructura robusta, mientras demandas perceptuales (45%) priorizan interactividad social, alineando perfectamente con capacidades marketplace vs. repositorios estáticos tradicionales.

4.2.1.1.5. ELABORACION DE SINTESIS INTERPRETATIVA

Este paso resume los patrones clave por dimensión y los convierte en requisitos funcionales concretos para el marketplace. Cada síntesis incluye cita representativa de datos + requisitos derivados (R1-R9 numerados para backlog).

Dimensión	Síntesis	Requisitos
Técnica	Aunque 83% equipos usan repositorios (19/23), documentación crítica falta sistemáticamente (74%, 17/23). Licencias inexistentes (100%) impiden la reutilización legal.	R1: Plantillas auto-generadas README/licencia al publicar (solución: wizard guiado). R2: Checklist validación mínima (bloqueo publicación si DOCUM/LICEN ausentes). R3: Indexación automática de metadatos (lenguaje, dependencias, endpoints).

Académico-pedagógica	Continuidad baja (22%) por falta de articulación curricular y tiempo (70% BARR). Únicos casos exitosos: proyectos propios evolucionados (O05/O09).	R4: Filtros semestrales/asignatura + visualización "ruta evolutiva" proyectos. R5: Badges "aprobado X materia" para certificación pedagógica. R6: Espacio sugerencias docentes vinculado a proyectos.
Perceptiva-actitudinal	Feedback/visibilidad motivan fuertemente (85% COLAB+VISIB). Colaboración actual informal (WhatsApp 40% obs).	R7: Dashboard métricas públicas (visitas, forks, calificaciones). R8: Sistema comentarios estructurado (positivos/mejoras). R9: Notificaciones real-time feedback + badges reconocimiento.

Tabla 27: Síntesis interpretativa.

Fuente: Elaboración Propia.

4.2.2. ANÁLISIS CUANTITATIVO DE ENCUESTA ESTRUCTURADA

Esta subsección expone el procesamiento sistemático de los datos obtenidos mediante la encuesta diagnóstica pre-artefacto (Instrumento C), aplicada exitosamente a 30 estudiantes del programa de Ingeniería de Sistemas de la UPC a través de Google Forms, alcanzando una tasa de respuesta del 100% sobre la muestra objetivo. El instrumento, compuesto por 12 ítems en escala Likert (1=Totalmente en desacuerdo - 5=Totalmente de acuerdo) fundamentados en el Modelo de Aceptación Tecnológica (TAM), permitió cuantificar las percepciones sobre visibilidad, continuidad, colaboración y disposición hacia el marketplace propuesto, complementando el análisis cualitativo previo.

4.2.2.1. PROCEDIMIENTO APLICADO

4.2.2.1.1. CODIFICACIÓN Y PREPARACIÓN DE DATOS

- **Exportación automática:** Google Forms generó hoja de cálculo nativa con respuestas codificadas numéricamente (1=Totalmente en desacuerdo...5=Totalmente de acuerdo).
- **Limpieza:** Verificación 0% datos faltantes, 0% respuestas inconsistentes.
- **Estructuración:** 12 columnas Likert + 4 demográficas, formato tabular listo para descriptivos.

4.2.2.1.2. ANÁLISIS DESCRIPTIVO UNIVARIADO

- **Frecuencias absolutas/relativas:** Tablas % por escala (1-5).
- **Tendencia central:** Media aritmética, mediana, moda.
- **Dispersión:** Rango, desviación estándar.
- **Gráficos:** Histogramas de barras automáticos (10 figuras).

4.2.2.1.3. REPRESENTACIÓN GRÁFICA BÁSICA

Los proyectos rara vez trascienden el semestre. (Continuidad)
30 respuestas

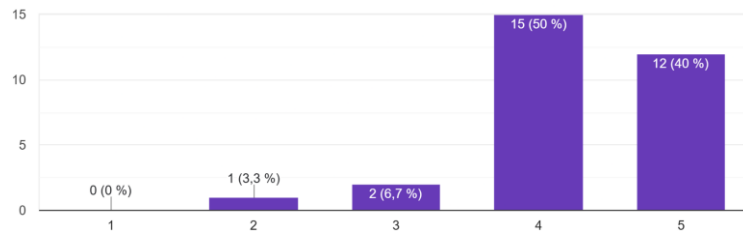


Figura 4: Trascendencia de los proyectos en un semestre.

Fuente: Google Forms.

Reutilizaría componentes si existiera un repositorio con versiones y calificaciones. (Utilidad percibida)
30 respuestas

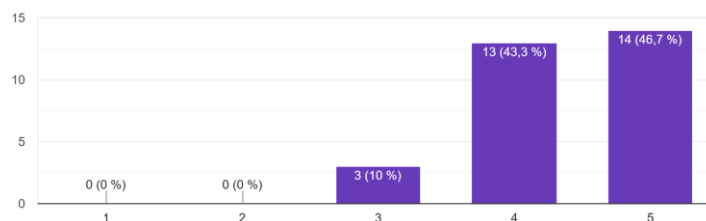


Figura 5: Utilidad percibida.

Fuente: Google Forms.

Me siento cómodo publicando código con licencia abierta con reconocimiento de autoría.

(Apertura/licencia)

30 respuestas

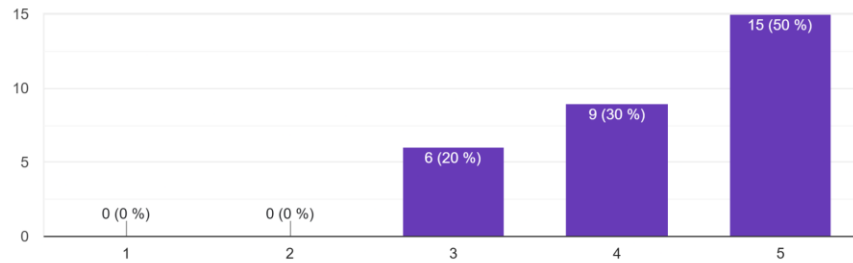


Figura 6: Apertura/Licencias en un proyecto.

Fuente: Google Forms.

Colaborar con equipos de otros cursos es viable con apoyo institucional. (Colaboración)

30 respuestas

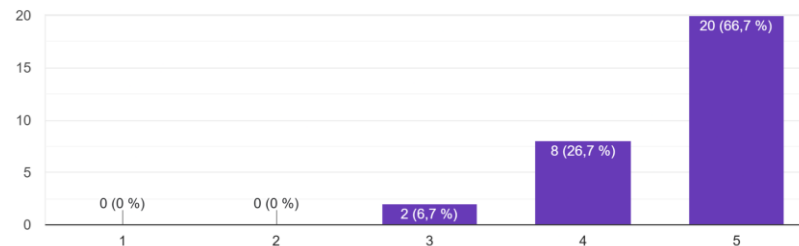


Figura 7: Viabilidad de colaboración con equipos.

Fuente: Google Forms.

Valoro ver métricas (visitas, descargas, forks) de mis proyectos. (Motivadores)
30 respuestas

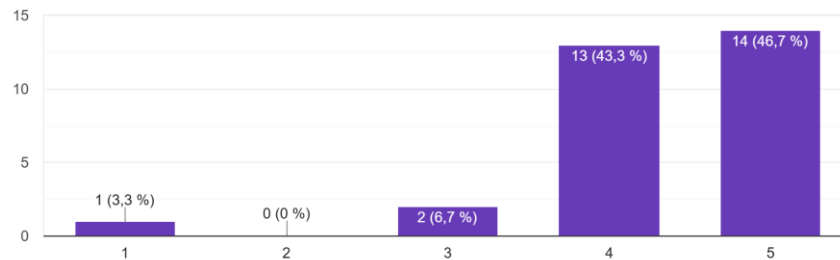


Figura 8: Aspectos motivadores en un proyecto.

Fuente: Google Forms.

Confiaría en un sistema institucional para alojar proyectos de software. (Confianza)
30 respuestas

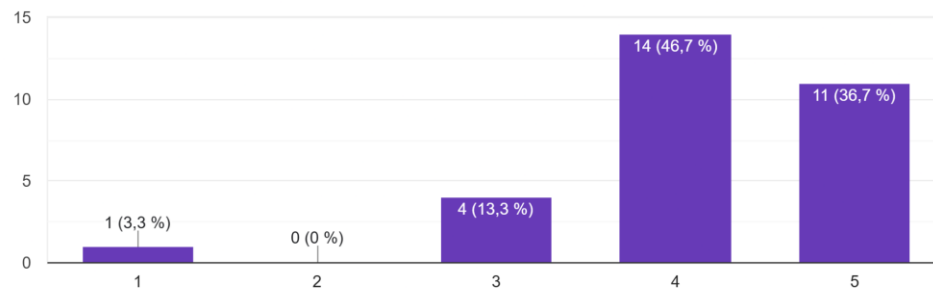


Figura 9: Nivel de confianza en alojar un proyecto.

Fuente: Google Forms.

Un control de versiones visible aumentaría la probabilidad de continuar proyectos.
(Continuidad/versionado)

30 respuestas

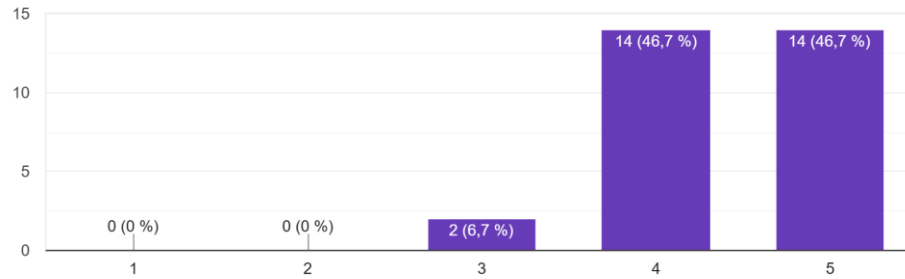


Figura 10: Control de versiones (continuidad).

Fuente: Google Forms.

Mis docentes incentivan reutilizar trabajos previos de la UPC. (Cultura pedagógica)

30 respuestas

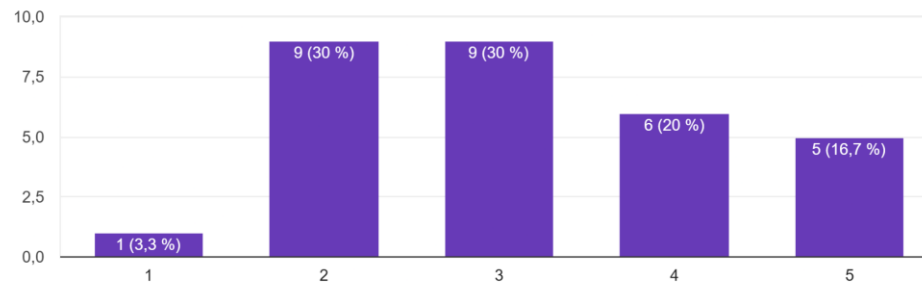


Figura 11: Cultura pedagógica en la UPC.

Fuente: Google Forms.

Usaría un marketplace institucional de proyectos de software. (Intención de uso)

30 respuestas

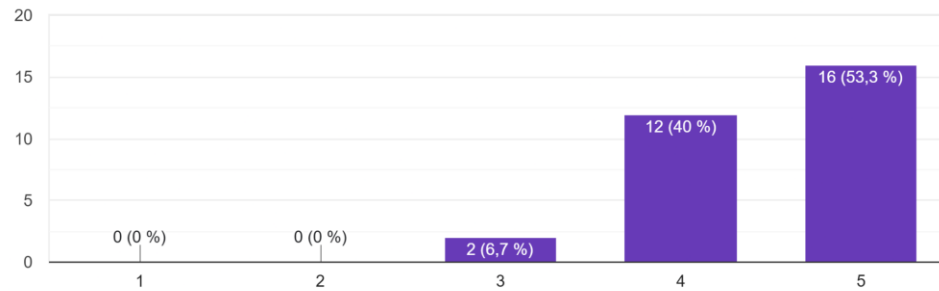


Figura 12: Intención de uso de Marketplace.

Fuente: Google Forms.

Conozco opciones de licenciamiento y sé cuál elegiría para mi proyecto. (Alfabetización en licencias)

30 respuestas

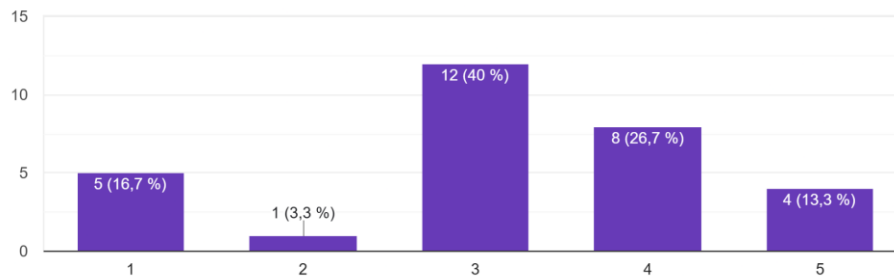


Figura 13: Alfabetización en licencias.

Fuente: Google Forms.

Semestre académico.
30 respuestas

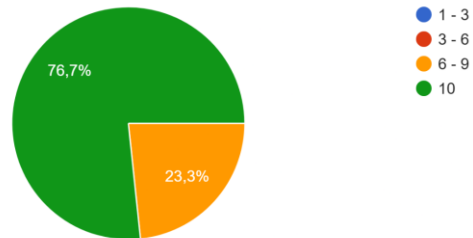


Figura 14: Semestre académico de estudiantes (Instrumento C).

Fuente: Google Forms.

Experiencia en Git/GitHub (basico/medio/avanzado)
30 respuestas

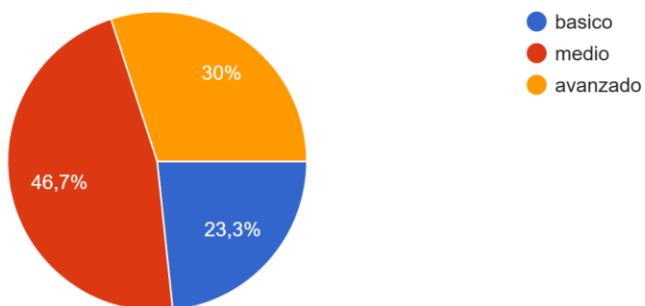


Figura 15: Experiencia de estudiantes con repositorios (Instrumento C).

Fuente: Google Forms.

Número de proyectos de aula completados.
30 respuestas

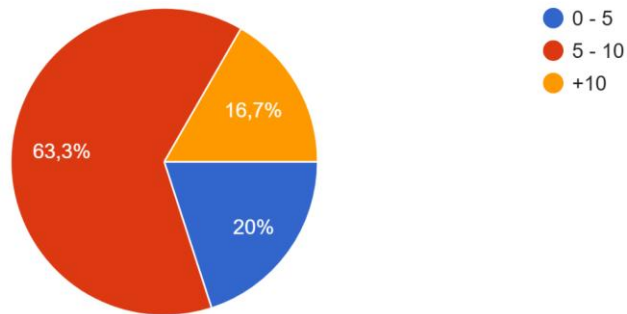


Figura 16: Número de proyectos de aula completados por estudiantes (Instrumento C).

Fuente: Google Forms.

Experiencia publicando repositorios
30 respuestas

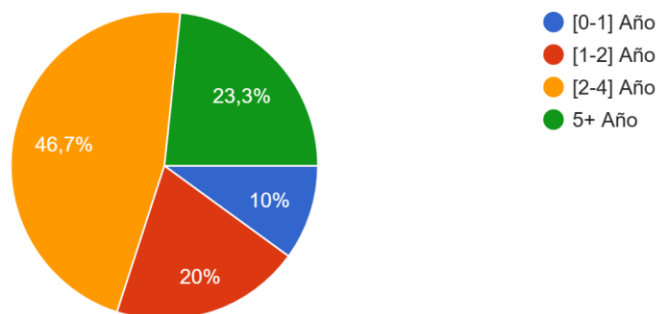


Figura 17: Experiencia de los estudiantes publicando proyectos (Instrumento C).

Fuente: Google Forms.

Ítem	Variable	Media	Mediana	Moda	% acuerdo (4-5)	Desviación est.
1	Continuidad	4.20	4	4	90%	0.72
3	Utilidad repositorio	4.37	5	5	90%	0.72
4	Licencias abiertas	4.30	5	5	80%	0.88
5	Colaboración	4.60	5	5	93%	0.56
6	Métricas visitas	4.37	4	4,5	90%	0.81
7	Confianza sistema	4.17	4	4	83%	0.87
8	Control versiones	4.53	5	5	93%	0.57
9	Cultura docente	2.63	3	2,3	37%	1.12
10	Intención TAM	4.47	5	5	93%	0.57

11	Documentación	4.30	5	5	80%	0.88
12	Conocimiento licencias	2.90	3	3	40%	1.15

Tabla 28: Estadísticos descriptivos completos por ítem.

Fuente: Elaboración Propia.

Variable	Categoría	Frecuencia	Porcentaje
Semestre	10°	23	76.7%
	6-9°	7	23.3%
Git experiencia	Medio	14	46.7%
	Avanzado	9	30.0%
	Básico	7	23.3%
Proyectos	5-10	19	63.3%
	0-5	6	20.0%
	+10	5	16.7%

Tabla 29: Perfil demográfico (frecuencias).

Fuente: Elaboración Propia.

4.2.2.1.4. LECTURA DE RESULTADOS EN FUNCIÓN DEL DISEÑO

La interpretación sistemática de los resultados cuantitativos permite priorizar funcionalidades críticas y características específicas del artefacto, estableciendo una relación directa entre las percepciones de los usuarios y las decisiones arquitectónicas del marketplace. A continuación se presentan dos tablas que operacionalizan esta traducción:

% Alto acuerdo	Ítem clave	Funcionalidad priorizada	Justificación diseño
93%	Colaboración cursos (ítem 5)	Sistema comentarios/reseñas	Máxima demanda colaboración inter-grupos
93%	Usaría marketplace (ítem 10)	Módulo exploración proyectos	Intención excepcional TAM confirma viabilidad
90%	Discontinuidad percibida (ítem 1)	Filtros avanzados búsqueda	Necesidad urgente acceso proyectos históricos
90%	Métricas visitas/forks (ítem 6)	Dashboard métricas públicas	Incentivo principal publicación/mantenimiento
80%	Documentación mínima (ítem 11)	Wizard publicación obligatoria	README/demos esenciales reutilización

37%	Cultura docente (ítem 9)	Espacio recomendaciones docentes	Brecha crítica continuidad curricular
-----	--------------------------	----------------------------------	---------------------------------------

Tabla 30: Priorización de funcionalidades y características en función del diseño.

Fuente: Elaboración Propia.

A continuación se describen los perfiles de usuario y características en base al diseño

Grupo identificado	Características demográficas	Necesidades específicas	Características de diseño
Semestre 10° (77%)	Git medio-alto (77%), 5-10 proyectos (63%)	Gestión avanzada versiones, licenciamiento	Vista avanzada: Integración GitHub, historial commits, selector licencias MIT/GPL Instrumento-C-llenado.pdf
Semestres 6-9° (23%)	Git medio (47%), experiencia publicar 2-4 años	Visibilidad, colaboración básica	Vista intermedia: Métricas públicas, comentarios estructurados, filtros semestrales
Baja experiencia licencias (60%)	Ítem 12: media 2.9	Orientación licenciamiento	Asistentes contextuales: Pop-ups explicativos, checklists obligatorios

Tabla 31: Perfiles de usuario y características en base al diseño.

Fuente: Elaboración Propia.

4.2.3. TRIANGULACIÓN METODOLÓGICA (SÍNTESIS REQUISITOS POR DIMENSIÓN)

La triangulación metodológica se aplicó con el propósito de fortalecer la validez del diagnóstico y convertir los hallazgos en requisitos accionables para el diseño del marketplace interactivo, integrando evidencia cualitativa (entrevistas semiestructuradas y observación directa no

participante) y evidencia cuantitativa descriptiva (encuesta estructurada). En coherencia con lo definido en el marco metodológico, se cruzaron patrones recurrentes identificados en entrevistas (E01–E13) con comportamientos observados (O01–O10) y con la priorización reportada en la encuesta (Instrumento C), consolidando una síntesis por dimensión.

Este cruce permitió (i) confirmar convergencias entre lo que los actores declaran y lo que efectivamente ocurre en la práctica, y (ii) traducir dichas convergencias en requisitos funcionales y de diseño agrupados por dimensión: técnica, académico-pedagógica y perceptiva-actitudinal. Los requisitos resultantes se numeran como R1–R9 para conservar trazabilidad con la definición posterior de criterios de aceptación y backlog priorizado.

4.2.3.1. DIMENSIÓN TÉCNICA (R1–R3)

La evidencia convergente muestra adopción de herramientas de repositorio/control de versiones en varios equipos, pero con deficiencias estructurales persistentes en documentación mínima y elementos habilitantes de reutilización (especialmente licenciamiento). En entrevistas y observaciones se reporta ausencia frecuente de README y prácticas incompletas de documentación, y en el análisis cualitativo se destaca la ausencia de licencias como barrera directa para reutilizar y publicar con claridad. Por tanto, los requisitos técnicos priorizan guiar y estandarizar la publicación, asegurar mínimos de calidad y automatizar metadatos para búsqueda/descubrimiento.

- **R1.** Plantillas autogeneradas (README + licencia) durante la publicación (wizard guiado).
- **R2.** Checklist de validación mínima que bloquee publicación si faltan documentación/licencia requeridas.
- **R3.** Indexación automática de metadatos (lenguaje, dependencias, endpoints u otros atributos técnicos) para mejorar búsqueda y filtrado.

4.2.3.2. DIMENSIÓN ACADÉMICO-PEDAGÓGICA (R4–R6)

Los hallazgos triangulados evidencian baja continuidad/reutilización de proyectos entre semestres, asociada principalmente a barreras de articulación curricular, lineamientos

docentes y dinámica académica (hacer “proyecto nuevo” en cada asignatura). La observación respalda el patrón de reinicio desde cero en la mayoría de sesiones/equipos y las entrevistas explicitan la dependencia de orientación docente para retomar o evolucionar prototipos. En consecuencia, los requisitos de esta dimensión buscan introducir mecanismos dentro del marketplace que conecten proyectos con su contexto formativo y faciliten su evolución como trayectoria académica.

- **R4.** Filtros por semestre/asignatura y visualización de “ruta evolutiva” del proyecto (historial de continuidad).
- **R5.** Badges o sellos tipo “Aprobado en X materia” para reforzar valor pedagógico y certificación interna.
- **R6.** Espacio de sugerencias/recomendaciones docentes vinculado a cada proyecto (retroalimentación con intención formativa).

4.2.3.3. DIMENSIÓN PERCEPTIVA Y ACTITUDINAL (R7–R9)

La convergencia entre fuentes muestra que visibilidad y retroalimentación funcionan como incentivos centrales para publicar, mantener y mejorar proyectos, mientras que la colaboración actual tiende a ser informal (p. ej., canales externos) y poco estructurada para sostenerse. Tanto entrevistas como observaciones identifican demanda de comentarios/feedback y deseo de exposición del trabajo; además, la priorización cuantitativa refuerza qué métricas, visibilidad y mecanismos sociales elevan la intención de uso. Por ello, los requisitos de esta dimensión se enfocan en interactividad social, reconocimiento y comunicación efectiva dentro de la plataforma.

- **R7.** Dashboard de métricas públicas (visitas, forks, calificaciones u otras) por proyecto para incentivar visibilidad y continuidad.
- **R8.** Sistema de comentarios estructurado (positivos/mejoras) para retroalimentación útil y trazable.
- **R9.** Notificaciones (idealmente en tiempo real) y mecanismos de reconocimiento (badges) ante interacción/feedback recibido.

Dimensión	Evidencia convergente (triangulada)	Necesidad de diseño	Requisitos derivados
Técnica	Uso de repositorios/control de versiones, pero falta recurrente de documentación mínima y ausencia de licencias; necesidad de estandarizar publicación y descubrimiento.	Publicación guiada + mínimos de calidad + metadatos para búsqueda.	R1, R2, R3.
Académico-pedagógica	Baja continuidad y reutilización entre semestres por barreras curriculares/docentes; escasa trayectoria evolutiva del proyecto.	Conectar proyectos con asignaturas/semestres y facilitar evolución pedagógica.	R4, R5, R6.
Perceptiva-actitudinal	Alta motivación por visibilidad/feedback; colaboración informal y poco estructurada; preferencia por interactividad.	Reforzar reconocimiento, interacción social y comunicación dentro del sistema.	R7, R8, R9.

Tabla 32: Síntesis de requisitos por dimensión (triangulación)

Fuente: Elaboración Propia.

4.3. DEFINICIÓN DE REQUISITOS DEL ARTEFACTO

A partir de la triangulación metodológica (R1-R9), se definen los requisitos del marketplace en tres niveles jerárquicos que aseguran trazabilidad desde diagnóstico hasta su desarrollo final: objetivos SMART (alineados a OG/OE), criterios de aceptación (testables/verificables) y backlog priorizado MoSCoW (ejecutable).

4.3.1. OBJETIVOS SMART

Los objetivos SMART traducen los 9 requisitos triangulados en metas específicas, medibles, alcanzables, relevantes y temporizadas, alineadas al Objetivo General y Específicos del proyecto.

Objetivo	S Específico	M Medible	A Alcanzable	R Relevante	T Temporizado
OA1: Publicación estandarizada	Marketplace con wizard que garantice README+licencia en TODOS los proyectos publicados	100% proyectos publicados cumplen checklist mínima (R1-R2)	Plantillas predefinidas + validación automática	Resuelve déficit documentación 74-83% (R1-R3)	Febrero 2026 (fin proyecto)
OA2: Descubrimiento inteligente	Indexación automática metadatos + filtros avanzados (lenguaje, semestre, tech)	≥90% precisión búsqueda (pruebas usabilidad) + 5 filtros funcionales (R3)	Algoritmo basado en Supabase + UI Next.js	Mejora hallazgo proyectos (83% usan repos)	Diciembre 2025 (MVP)
OA3: Continuidad pedagógica	Visualización "ruta evolutiva" proyectos + badges materias	≥3 proyectos con ≥2 versiones evolutivas + 5 badges implement	Historial versioning + integración curricular	Soluciona baja continuidad 22-70% (R4-R6)	Enero 2026 (beta)

		ados (R4-R5)			
OA4: Interactividad social	Sistema comentarios + dashboard métricas + notificaciones realtime	≥80% usuarios activos comentan + 5 métricas públicas/proyecto (R7-R9)	WebSockets Supabase + Tailwind dashboard	Responde motivación VISIB/COL AB 85-87%	Febrero 2026
OA5: Escalabilidad institucional	Plataforma multiusuario (stud-doc-adm) con RLS Supabase	100 usuarios concurrentes + 4 roles funcionales	Supabase Edge Functions + Docker	Soporte crecimiento UPC Ingeniería Sistemas	Febrero 2026

Tabla 33: Objetivos SMART del artefacto

Fuente: Elaboración Propia.

4.3.2. CRITERIOS DE ACEPTACIÓN

Los criterios de aceptación operacionalizan cada OA en pruebas específicas ejecutables durante desarrollo y validación. Están diseñados para verificación automatizada (unit tests) y manual (UAT), siguiendo principios ágiles "Definition of Done". Cada criterio mapea 1:1 con requisitos R1-R9.

Objetivo	Criterios de aceptación (testables)
OA1	1. Wizard publicación completa en <5min 2. 100% proyectos bloqueados sin README/licencia 3. Plantillas CC-BY-SA pre-cargadas
OA2	1. Búsqueda por lenguaje/semestre/tecnología funcional 2. Filtros combinables (AND/OR) 3. Resultados ordenados relevancia
OA3	1. Historial evolutivo visible por proyecto 2. Badges materias mostrados en cards 3. Recomendaciones docentes vinculadas
OA4	1. Dashboard métricas actualizado realtime 2. Comentarios estructurados (+mejoras) 3. Notificaciones push WebSocket
OA5	1. RLS Supabase por rol (estud/doc/adm) 2. 100 usuarios concurrentes sin crash 3. Docker deployment 1-click

Tabla 34: Criterios de aceptación por objetivo.

Fuente: Elaboración Propia.

4.3.3. BACKLOG PRIORIZADO MOSCOW

El backlog aplica la metodología MoSCoW para priorizar la ejecución según impacto-valor-esfuerzo, considerando restricciones temporales (8 meses) y recursos (2 desarrolladores).

Prioridad	Requisitos	User Stories	Esfuerzo estimado

MUST	R1,R2,R3	Como estudiante quiero publicar proyecto con wizard que me obligue README+licencia para que mi código sea reutilizable	8 puntos
MUST	R7	Como usuario quiero dashboard métricas públicas para motivarme a mejorar mi proyecto	5 puntos
SHOULD	R4,R5	Como docente quiero ver filtros semestre/materia+badges para identificar proyectos útiles en mi asignatura	6 puntos
COULD	R8	Como usuario quiero sistema comentarios estructurado para dar feedback constructivo	4 puntos
WONT	R6,R9	Sugerencias docentes + notificaciones realtime (fase 2 post-lanzamiento)	-

Tabla 35: Backlog MoSCoW del artefacto.

Fuente: Elaboración Propia.

4.4. DISEÑO Y CONSTRUCCIÓN DEL ARTEFACTO

4.4.1. ARQUITECTURA GENERAL

El artefacto propuesto consiste en un marketplace institucional de proyectos de software, cuya finalidad es dar continuidad y visibilidad a los trabajos desarrollados por estudiantes de la Universidad Popular del Cesar. La arquitectura planteada sigue un modelo modular compuesto por:

- **Frontend:** construido en **Next.js/React**, encargado de la interfaz de usuario y la experiencia de navegación responsiva.
- **Backend:** desarrollado en **FastAPI**, gestionará la lógica de negocio, validación de datos y conexión con servicios externos.
- **Base de datos: Supabase (PostgreSQL)** para almacenar metadatos de proyectos, usuarios y métricas.
- **Autenticación y permisos:** gestionados a través de **Supabase Auth** con integración institucional.
- **Integración GitHub:** uso de la **API oficial de GitHub** para proyectos de software. Todo proyecto de desarrollo deberá vincular obligatoriamente un repositorio, del cual se extraerán automáticamente métricas e información clave (README, licencia, commits, issues, releases, contribuyentes).

A continuación se presenta un esquema de la arquitectura propuesta de modo que resulte más fácil su visualización.

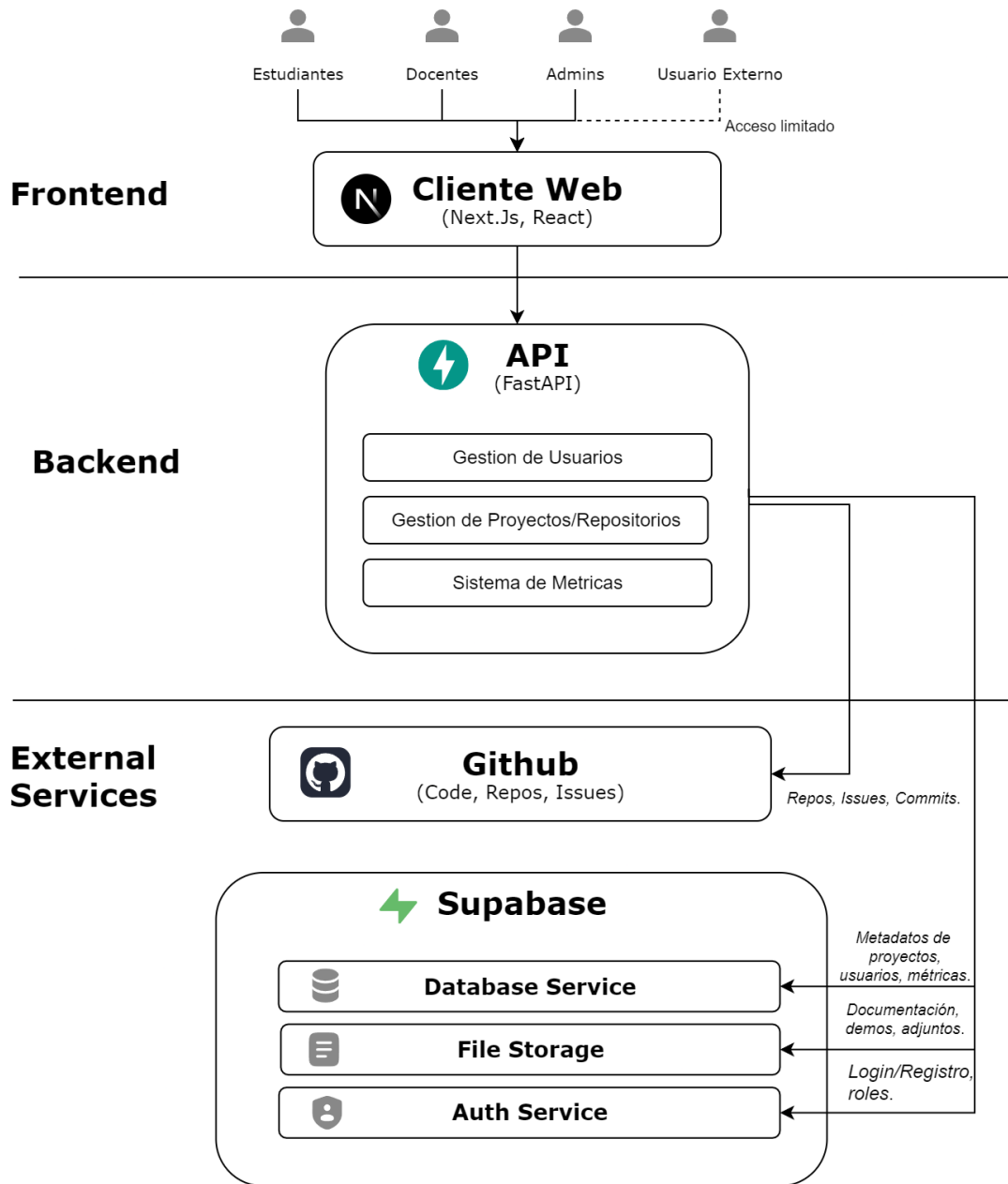


Figura 18: Diagrama de arquitectura del artefacto.

Fuente: Elaboración Propia

La elección de tecnologías se fundamenta tanto en los conocimientos previos de los desarrolladores como en la utilidad, modernidad y simplicidad que ofrecen para el desarrollo de un artefacto académico escalable. GitHub se establece como herramienta externa indispensable para el manejo del código de los proyectos, ya que permite integrar control de versiones, colaboración y trazabilidad en un entorno ya adoptado por la comunidad universitaria y profesional. Por otro lado, se selecciona Supabase como plataforma en la nube que brinda servicios integrados de base de datos, autenticación y almacenamiento a bajo costo, lo que reduce la complejidad de la infraestructura y agiliza el desarrollo. Estas decisiones no solo responden a criterios de eficiencia técnica, sino también a la sostenibilidad y accesibilidad del artefacto en el contexto institucional.

4.4.2. ARQUITECTURA ESPECÍFICA

En esta sección se describen las partes internas del artefacto y la arquitectura seleccionada para su desarrollo. El propósito principal es mostrar la organización del sistema tanto a nivel de frontend como backend de manera que se entienda como el artefacto estará diseñado internamente para soportar los módulos o funciones deseados.

4.4.2.1. ARQUITECTURA DEL FRONTEND

El frontend del artefacto se desarrollará con Next.js un framework de React que ofrece funcionalidades extremadamente útiles como renderizado del lado del servidor (SSR), generación de sitios estáticos (SSG), enrutamiento integrado y rutas API, entre otras. Para la creación del frontend se adopta una arquitectura basada en componentes. Este enfoque permite dividir la interfaz en unidades pequeñas y reutilizables como tarjetas, formularios, menús, buscadores, entre otros que luego se integran para conformar vistas completas. La modularidad propia de React facilita la escalabilidad, separación de responsabilidades y la evolución futura del sistema sin comprometer la coherencia visual ni funcional.

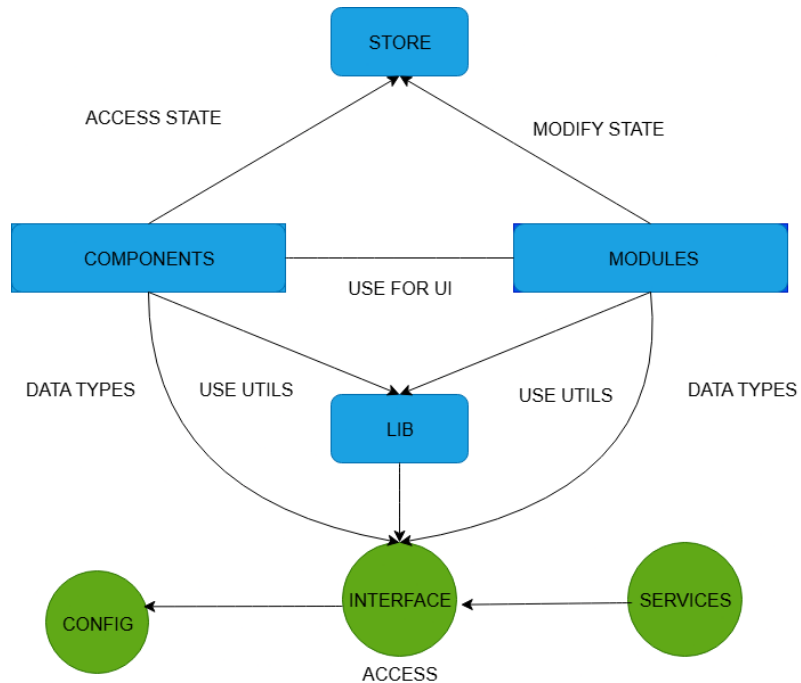


Figura 19: Diagrama de arquitectura del Frontend del artefacto.

Fuente: Elaboración Propia.

Esta organización facilita la escalabilidad del sistema, ya que cada módulo concentra su lógica y vistas, mientras que otro módulo por su lado se encarga de centralizar la comunicación con la API y servicios externos. Esto permite aislar cambios (por ejemplo, si se cambia el backend o un endpoint) sin afectar la capa visual ni la navegación.

4.4.2.2. ARQUITECTURA DEL BACKEND

El backend del artefacto será implementado utilizando **FastAPI**, un marco web moderno, rápido y de alto rendimiento para crear APIs con Python. La arquitectura propuesta está basada en capas, estructura que garantiza la clara separación de responsabilidades en niveles bien definidos, como rutas o controladores, servicios, repositorios y modelos. Con este esquema se garantiza mantenibilidad, claridad en la organización del código y facilidad para la incorporación de nuevas funciones en futuras iteraciones.

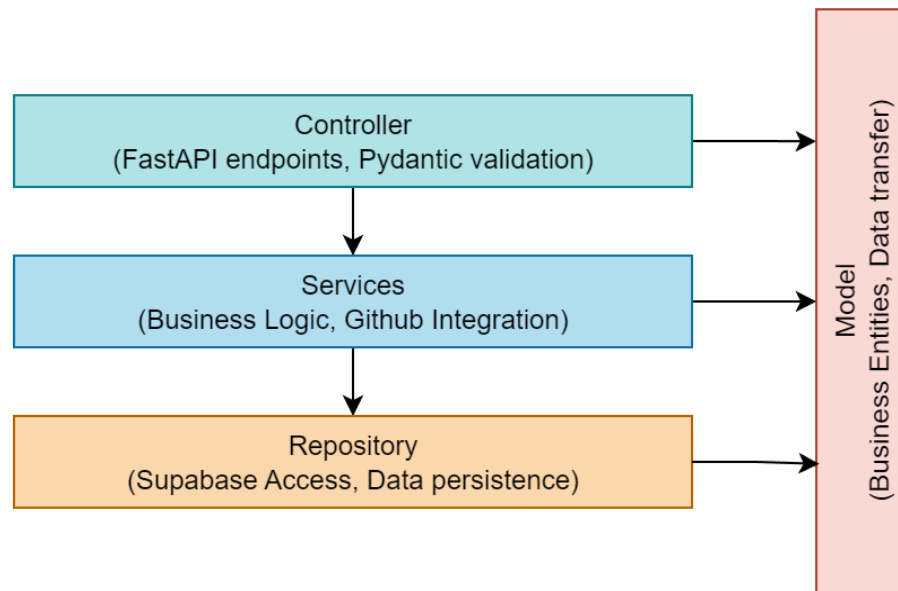


Figura 20: Diagrama de arquitectura del Backend del artefacto.

Fuente: Elaboración Propia.

El backend adopta una arquitectura por capas, que separa claramente las responsabilidades: la capa de rutas maneja la comunicación con el cliente web, la capa de servicios encapsula la lógica de negocio e interacciones con servicios externos como Github, la capa de persistencia se encarga de interactuar con Supabase y la capa de modelos asegura consistencia de datos. Esta división mejora la mantenibilidad y escalabilidad del sistema.

4.4.3. MÓDULOS PRINCIPALES DEL ARTEFACTO

A continuación se detallan los módulos funcionales del artefacto que responden directamente a los objetivos del proyecto y a los problemas identificados en fases previas. Cada módulo representa una unidad de valor orientada a la solución, yendo desde la publicación y gestión de proyectos, hasta la integración con GitHub y el cálculo de métricas de continuidad y

visibilidad. Esta modularidad facilita la construcción incremental del artefacto, la trazabilidad entre problema, objetivo e implementación y garantiza la escalabilidad de la plataforma en futuras iteraciones.

Módulo	Características
Gestión de proyectos	• Registro, edición y categorización de proyectos en diferentes estados. • Publicación flexible: con vinculación a repositorio GitHub (obligatorio para proyectos de software) o mediante archivos adjuntos (para proyectos no técnicos). • Administración de versiones, historial y/o actualización de la ficha de cada proyecto.
Gestión de Usuarios	• Registro y autenticación de usuarios utilizando Supabase Auth. • Actualización de perfil con información básica, intereses y rol (estudiante, docente, admin, usuario externo). • Integración con cuenta Google e Institucional para acceso rápido (single sign-on institucional).
Visibilidad y Exploración	• Catálogo de proyectos en vista de tarjetas con paginación y lazy loading. • Búsqueda avanzada por palabras clave. • Filtros dinámicos (categoría, semestre, docente, cantidad de contribuyentes, repositorio asociado). • Rankings de proyectos más vistos, con mayor número de forks, issues resueltos o reutilizados.

Colaboración	• Espacio de comentarios en cada proyecto para retroalimentación y preguntas. • Señales de interés (favoritos, “estrellas internas”).
Métricas e indicadores	• Métricas de visibilidad: nº de vistas de los proyectos. • Métricas de visibilidad en tiempo: nº de vistas de los proyectos en el tiempo. • Métricas de usabilidad: utilización de artefactos (links, archivos). • Cantidad de proyectos por estado • Cantidad de proyectos por categoría • Cantidad de proyectos por subcategoría • Reporte accesible en cada ficha de proyecto (stats en página individual de cada proyecto).

Tabla 35: Módulos principales del artefacto.

Fuente: Elaboración Propia.

4.4.4. DISEÑO FUNCIONAL

En esta sección se describe y representa de forma detallada la interacción de los usuarios con el artefacto y como este responde a sus necesidades. Se describen los casos de uso principales derivados de los objetivos planteados, los roles involucrados y las funcionalidades clave que permitirá el artefacto. Además, se incluyen representaciones útiles para apoyar y sostener el diseño del artefacto tales como diagramas de entidades y de secuencias, elementos fundamentales que permitirán visualizar las interacciones, flujos de información y la integración con servicios externos como Github y Supabase. Con esto, se establece un claro punto de partida para la implementación asegurando en todo momento trazabilidad entre el problema, los objetivos, la arquitectura y la funcionalidad esperada.

4.4.4.1. ROLES Y ACTORES

Para esta sección se identificarán y describirán cuáles serán los principales roles que interactúan con el artefacto, tanto en el alcance y las responsabilidades. Cada rol representa a un usuario o un sistema externo que cumple las funciones que son específicas en la plataforma. Esto permite poder delimitar las acciones que se pueden realizar por cada actor y como ayuda a cumplir los objetivos del proyecto.

Actor	Descripción	Responsabilidades
Estudiante	Usuario principal de la plataforma, encargado de publicar y gestionar proyectos académicos.	• Publicar proyectos con o sin repositorio. • Editar información de sus proyectos. • Consultar métricas de continuidad y visibilidad. • Comentar y marcar favoritos.
Docente	Usuario académico que supervisa, retroalimenta y orienta la calidad de los proyectos.	• Explorar proyectos de estudiantes. • Comentar y retroalimentar. • Validar proyectos destacados (si aplica).
Administrador	Rol encargado de la operación y moderación de la plataforma.	• Gestionar usuarios y permisos. • Modera proyectos y contenidos reportados. • Monitorear métricas globales.
Visitante	Usuario no autenticado con acceso limitado a la plataforma.	• Explorar proyectos públicos.

GitHub (sistema externo)	Plataforma de control de versiones y colaboración en desarrollo de software.	• Proveer autenticación OAuth. • Exponer información de repositorios (commits, issues, licencias, releases).
Supabase (sistema externo)	Plataforma backend-as-a-service para autenticación y base de datos.	• Autenticación de usuarios (Microsoft/email). • Almacenamiento de metadatos y registros. • Servicios en la nube.

Tabla 36: Roles y actores involucrados en el artefacto.

Fuente: Elaboración Propia.

4.4.4.2. CASOS DE USO PRINCIPALES

Para esta sección se representarán todos los casos de uso que describen como todos los distintos actores interactúan con el artefacto para poder alcanzar objetivos definidos previamente. Cada caso de uso brinda información detallada sobre las acciones clave, todos los flujos de información y los resultados que se esperan obtener en escenarios representativos. Estos casos ayudan a la visualización de manera clara como es el comportamiento funcional de la plataforma y sirven como la base para en el futuro llevarla a la elaboración de los diagramas y su implementación técnica.

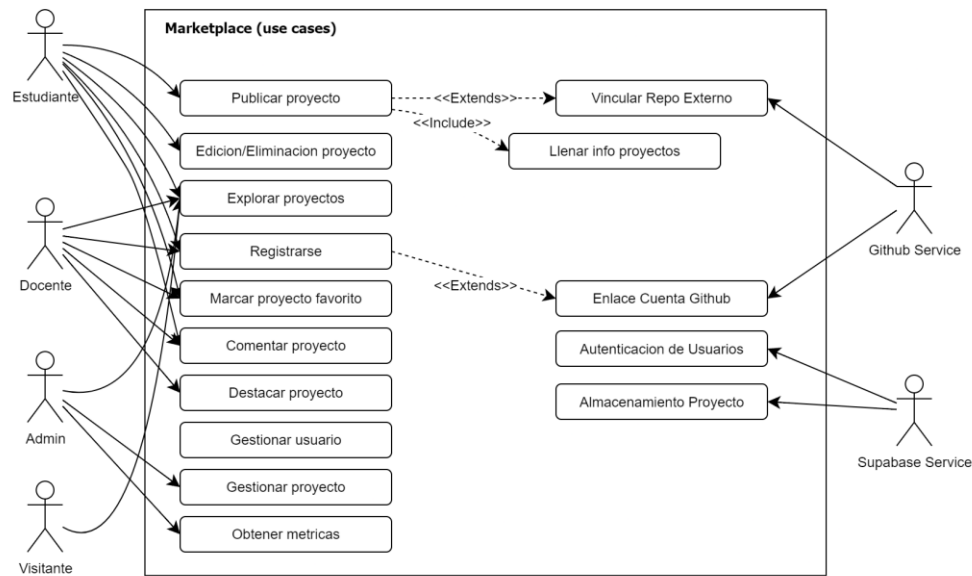


Figura 21: Diagrama de casos de uso del artefacto.

Fuente: Elaboración Propia.

A continuación se describen a detalle los actores y casos de uso:

Actor: Estudiante

Caso de uso	Propósito	Flujo principal	Precondicione s	Postcondiciones
Publicar proyecto	Permitir que el estudiante registre un proyecto nuevo en el marketplace	1. Inicia sesión. 2. Accede a “Publicar proyecto”. 3. Completa formulario (título, descripción,	Usuario autenticado. Formulario válido.	Proyecto registrado y visible en la plataforma.

		categoría, repositorio opcional). 4. Confirma y pública.		
Explorar proyectos	Consultar proyectos disponibles para descubrir y reutilizar	1. Ingresa al catálogo. 2. Usa buscador/filtros. 3. Visualiza resultados en tarjetas. 4. Accede al detalle de un proyecto.	Sistema con proyectos publicados.	Resultados listados, acceso al detalle de proyectos.
Visualizar métricas	Consultar estadísticas asociadas a su proyecto (vistas, commits, issues)	1. Entra al detalle de su proyecto. 2. Se cargan métricas desde backend y GitHub. 3. Se muestran gráficas/indicadores.	Proyecto con repositorio vinculado.	Estudiante conoce indicadores de continuidad y visibilidad.
Comentar en proyectos	Interactuar con otros proyectos	1. Abre el detalle de un proyecto.	Usuario autenticado.	Comentario almacenado y

	mediante retroalimentación	2. Escribe comentarios. 3. Pública.		visible en el proyecto.
Marcar favorito	Señalar interés por un proyecto para seguimiento	1. Entra al detalle del proyecto. 2. Pulsa en "favorito/estrella".	Usuario autenticado.	Proyecto queda registrado en lista personal de favoritos.

Tabla 37: Especificación caso de uso de actor estudiante.

Fuente: Elaboración Propia.

Actor: Docente

Caso de uso	Propósito	Flujo principal	Precondiciones	Postcondiciones
Explorar proyectos	Identificar proyectos de estudiantes para retroalimentar o continuar	Igual que el estudiante, con acceso ampliado a más metadatos.	Usuario autenticado como docente.	Docente accede a proyectos filtrados.
Comentar y retroalimentar	Ofrecer observaciones académicas	1. Abre detalles. 2. Escribe comentarios.	Autenticado como docente.	Comentario asociado al proyecto con rol visible.

		3. Envía.		
Validar proyectos destacados (opcional)	Reconocer proyectos de alta calidad para ranking institucional	1. Revisa el proyecto. 2. Marca como “destacado”.	Autenticado como docente. Permiso concedido.	Proyecto aparece en ranking de destacados.

Tabla 38: Especificación caso de uso de actor docente.

Fuente: Elaboración Propia.

Actor: Administrador

Caso de uso	Propósito	Flujo principal	Precondiciones	Postcondiciones
Gestionar usuarios	Mantener control sobre accesos	1. Ingresa al panel admin. 2. Busca usuario. 3. Bloquea, activa o actualiza permisos.	Usuario con rol administrador.	Cambios reflejados en Supabase.
Gestionar proyectos	Modera contenido publicado	1. Accede al panel. 2. Revisa proyectos reportados.	Proyecto previamente publicado.	Proyectos visibles o removidos.

		3. Aprueba o elimina.		
Monitorear métricas generales	Revisar KPIs de la plataforma	1. Ingresa al panel de métricas. 2. Consulta gráficos globales (nº proyectos activos, repositorios vinculados, etc.).	Autenticado como administrador.	Reportes generados.

Tabla 39: Especificación caso de uso de actor administrador.

Fuente: Elaboración Propia.

Actor: Visitante

Caso de uso	Propósito	Flujo principal	Precondiciones	Postcondiciones
Explorar proyectos públicos	Permitir acceso básico a proyectos sin autenticación	1. Accede al catálogo. 2. Usa buscador básico. 3. Accede a vistas limitadas.	Sistema con proyectos publicados.	Proyectos visualizados sin acceso a funciones interactivas.

Tabla 40: Especificación caso de uso de actor visitante.

Fuente: Elaboración Propia.

Actores externos (GitHub / Supabase)

Caso de uso	Propósito	Flujo principal	Precondiciones	Postcondiciones
Autenticación OAuth (GitHub)	Permitir vincular proyectos con GitHub	1. Usuario selecciona "Vincular GitHub". 2. Redirección a OAuth. 3. Confirma los permisos.	Usuario con cuenta GitHub.	Token seguro almacenado, repositorios vinculados.
Extracción de datos (GitHub)	Obtener commits, issues, releases, licencias	Backend llama a API GitHub. Procesa datos. Guarda en Supabase.	Proyecto vinculado a GitHub.	Información actualizada en la plataforma.
Autenticación Supabase	Gestionar login y sesiones de usuarios	Usuario inicia sesión (Microsoft o email).	Cuenta existente.	Usuario autenticado y con sesión activa.

Tabla 41: Especificación caso de uso de actores externos.

Fuente: Elaboración Propia.

4.4.4.3. MODELADO DE ENTIDADES

Para esta sección se hace la representación estructural del artefacto con modelos que describen tanto las entidades como los atributos y relaciones que componen el sistema. El

modelado proporciona una visión más clara y de manera organizada de los componentes que conforman el marketplace interactivo y sus relaciones directas entre ellos, para lograr brindar ayuda a los casos de uso que se han definido con anterioridad. Este modelado de entidades se realizará con el uso de diagramas de clases (UML) principalmente.

4.4.4.3.1. DESCRIPCIÓN DE LAS ENTIDADES PRINCIPALES

Esta sección presenta las entidades clave del modelo de negocio del artefacto, con su descripción, atributos principales y relaciones.

Entidad	Descripción	Atributos Principales	Relaciones
Project	Entidad central que representa un proyecto de desarrollo de software	id, name, about, icon_url, tags, created_date, update_date	Asociado con Author (1:), <i>contiene Media</i> (0:), Repository (1), Comment (0:*), File (1), GeneralRequirement (1), License (1)
Author	Representa a los desarrolladores o creadores del proyecto	id, name, bio, email, linkedin_profile	Crea Project (1:*), puede calificar proyectos, comentar proyectos

Repository	Almacena información del repositorio de código fuente	id, url, stars, forks, issues, last_commit	Pertenece a Project (1:1)
Media	Recursos multimedia asociados al proyecto (imágenes, videos, etc.)	id, type, title, description, url	Pertenece a Project (0:*)
Documentation	Documentación técnica del proyecto	id, readme, description_author	Asociada con Project (1:1)
Rating	Sistema de calificaciones para los proyectos	id, avg, count	Relacionada con Project (1:1)
Comment	Comentarios de usuarios sobre los proyectos	id, title, description	Asociados con Project (0:*)

File	Archivos del proyecto almacenados en el sistema	id, name, file_url, size	Pertenece a Project (1:1)
License	Información sobre la licencia de uso del proyecto	id, name, type, version, url, gpl_compatibility	Asociada con Project (1:1), define Permission y Obligation
Category	Categorización de proyectos por tipo o área	id, name, description, icon_url, active, date_creation	Clasifica proyectos, puede tener SubCategory
GeneralRequirement	Requisitos generales del proyecto	id, type, status, difficulty, language, current_version	Asociado con Project (1:1), relacionado con APIRequirement
APIRequirement	Especificaciones técnicas de proyectos de tipo APIs	id, base_url, protocol, data_format, response_format, encoding, rate_limiting, headers_included,	Deriva de GeneralRequirement

		requests_per_minute, requests_per_hour	
Stats	Estadísticas de visualización y engagement	id, views, likes, downloads	Asociadas con Project (1:1)

Tabla 42: Tabla entidades principales

Fuente: Elaboración Propia.

En la tabla, APIRequirement se resalta para subrayar que cada proyecto adopta rasgos según su subcategoría; en particular, en **Desarrollo de Software** la subcategoría **API** exige elementos como endpoints, documentación y seguridad, de modo que los requerimientos varían entre subcategorías, por lo que se prevé definir otras clases de requerimientos específicos con atributos y composiciones propios que se detallarán durante la implementación; en consecuencia, el diagrama funciona como modelo conceptual general del dominio, orienta el diseño y no pretende cubrir todas las subcategorías posibles.

4.4.4.3.2. DIAGRAMA DE CLASES UML

En esta sección se describe el diagrama de clases UML del sistema, destacando las clases principales, sus atributos y métodos, así como los tipos de relaciones entre ellas (asociación, herencia, agregación y composición) para ofrecer una vista estructural y coherente del modelo de dominio.

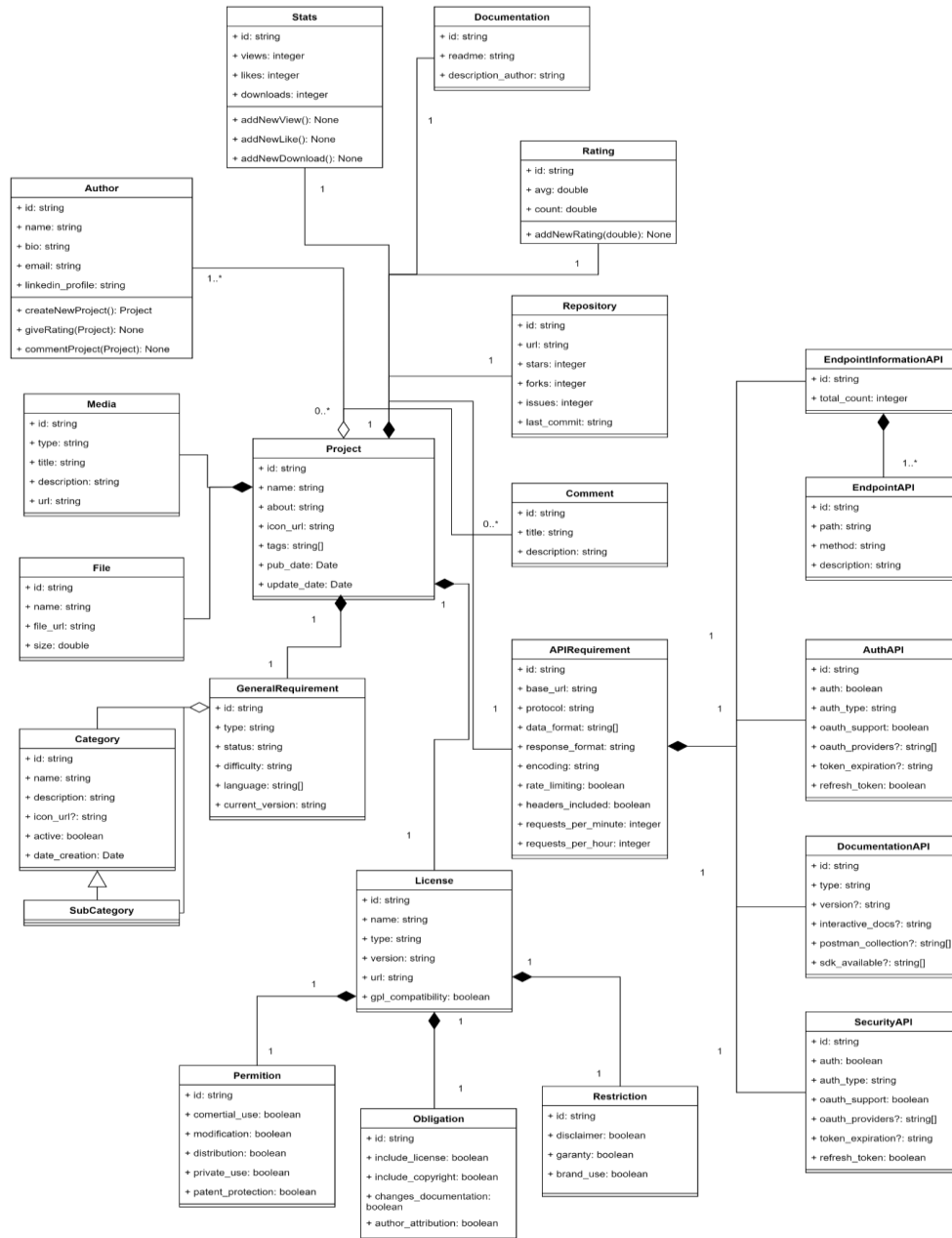


Figura 22: Diagrama de clases

Fuente: Elaboración Propia.

4.4.4.3.3. DIAGRAMA ENTIDAD-RELACIÓN

En esta sección se presenta el diagrama entidad-relación (ER) del sistema de marketplace interactivo, el cual constituye una representación visual y estructural de la organización de datos del repositorio institucional de proyectos de software. El diagrama ER es una herramienta fundamental en la ingeniería de software que facilita la comprensión, diseño y comunicación de la estructura de datos mediante la representación gráfica de entidades, sus atributos y las relaciones que existen entre ellas.

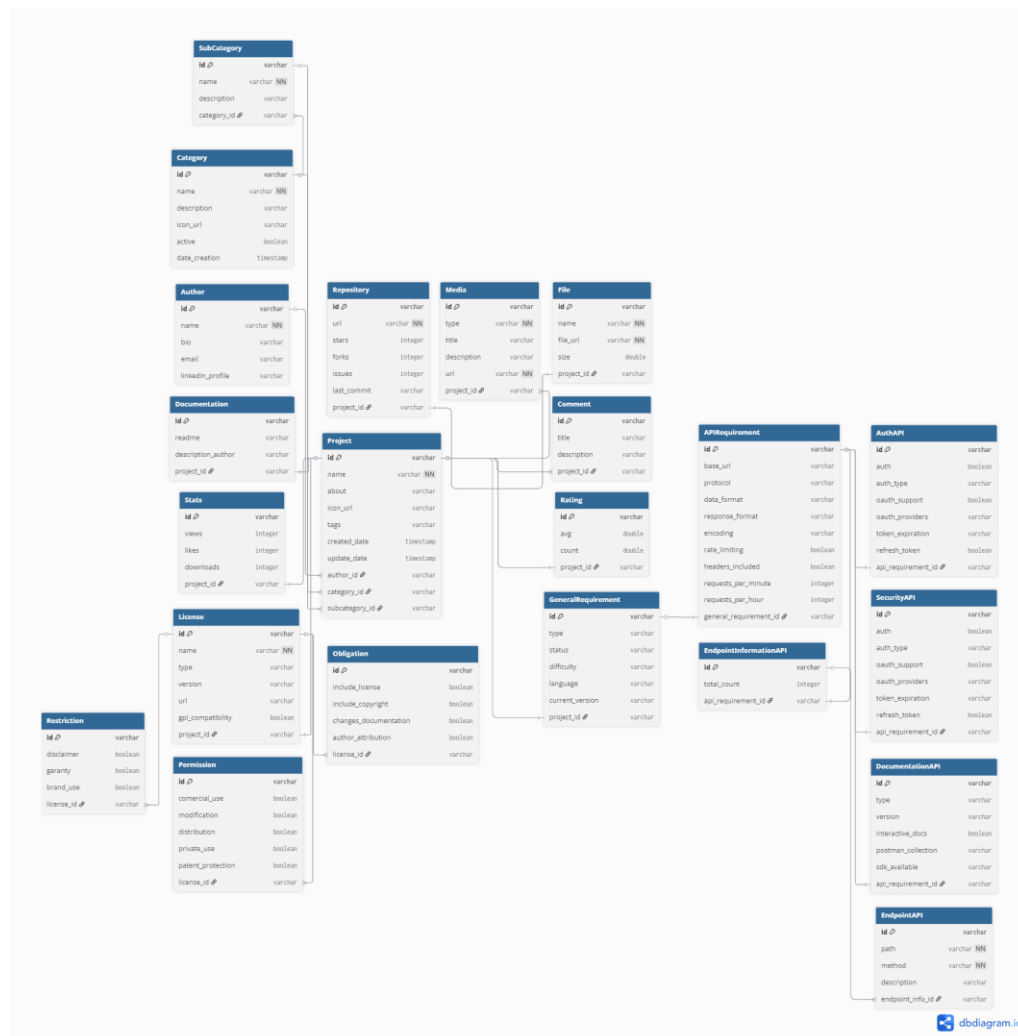


Figura 23: Diagrama entidad relación

Fuente: Elaboración Propia.

Para el presente proyecto, el modelo entidad-relación cumple un doble propósito esencial: por una parte, establece los fundamentos conceptuales que permiten entender el dominio del problema como la gestión, clasificación y difusión de proyectos académicos de desarrollo de software, y por otra, define la arquitectura de datos que soportará las funcionalidades del marketplace interactivo, incluyendo la publicación de proyectos, sistemas de calificación, gestión de usuarios y preservación del conocimiento académico.

4.4.4.3.4. DIAGRAMAS DE INTERACCIÓN

Los diagramas de interacción constituyen una representación visual fundamental que permite modelar y comprender el comportamiento dinámico del sistema marketplace interactivo de proyectos de software. Estos diagramas capturan las secuencias de mensajes e interacciones entre los diferentes actores del sistema (estudiantes, docentes, administradores) y los componentes tecnológicos (frontend, backend, base de datos, GitHub API) durante la ejecución de las funcionalidades principales.

Los diagramas de secuencia específicamente documentan el flujo temporal de operaciones, identificando los objetos participantes, los mensajes intercambiados, el orden de las interacciones y los puntos de control críticos. Esta representación resulta esencial para visualizar cómo el sistema gestiona las operaciones de manera distribuida entre el frontend desarrollado en Next.js, el backend basado en FastAPI, la base de datos Supabase PostgreSQL y los servicios externos como GitHub.

4.4.4.3.4.1. SECUENCIA (OBTENER UN PROYECTO POR ID)

Este diagrama detalla el proceso de visualización detallada de proyectos específicos, representando una de las funcionalidades más complejas del sistema. La secuencia incluye la sincronización en tiempo real con GitHub API para obtener estadísticas actualizadas, la carga de comentarios asociados, y la actualización de métricas de visualización. El sistema combina datos almacenados localmente con información actualizada del repositorio remoto, proporcionando una vista integral del proyecto que incluye información técnica, social y de uso

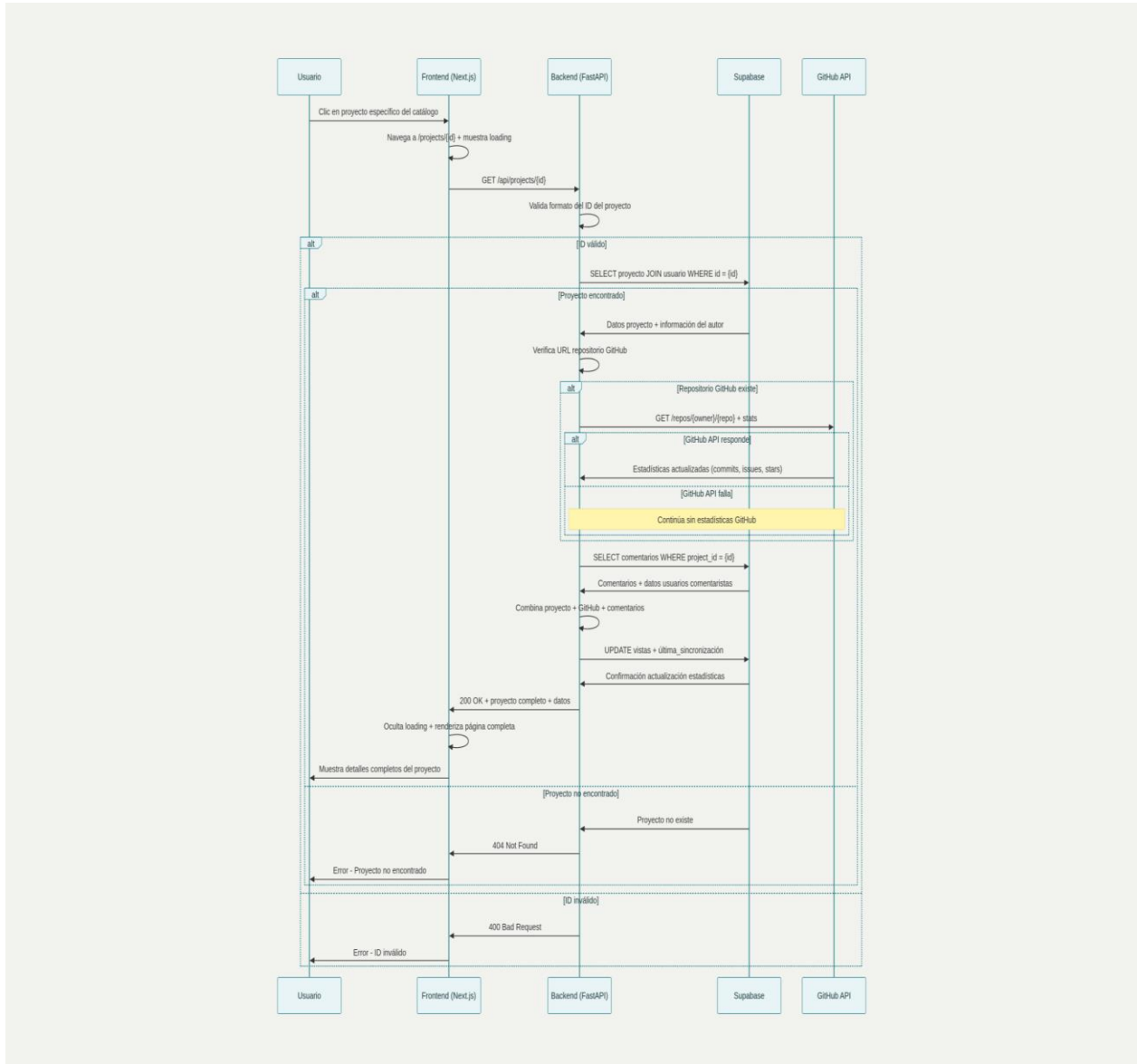


Figura 24: Diagrama de secuencia id

Fuente: Elaboración Propia.

4.4.4.3.4.2. SECUENCIA (FILTRAR PROYECTOS)

Este diagrama representa el sistema de búsqueda y filtrado avanzado que permite a los usuarios localizar proyectos específicos según múltiples criterios. La secuencia incluye la carga dinámica de opciones de filtro disponibles, construcción de consultas SQL complejas

con múltiples condiciones, y actualización de la interfaz con resultados relevantes. Esta funcionalidad es esencial para la navegación eficiente del repositorio institucional, especialmente conforme crece el volumen de proyectos almacenados.

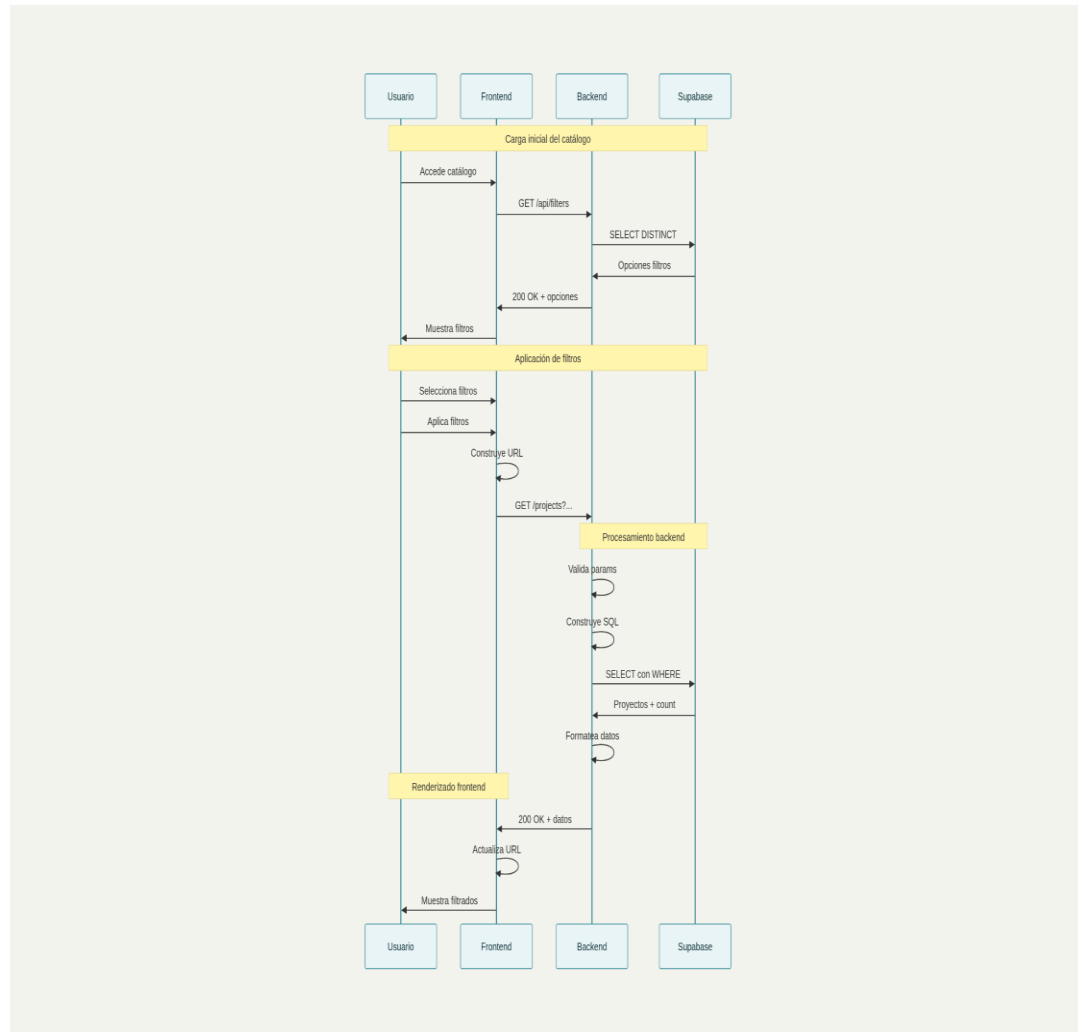


Figura 25: Diagrama de secuencia

Fuente: Elaboración Propia.

4.4.4.3.4.3. SECUENCIA (CREAR PROYECTO)

Este diagrama modela el proceso completo de registro de nuevos proyectos en el marketplace institucional. La secuencia inicia cuando un estudiante accede al formulario de creación y

abarca desde la validación de datos hasta la confirmación final. El sistema integra automáticamente metadatos del repositorio GitHub asociado, incluyendo README, licencia, estadísticas de commits y estructura del proyecto. Esta funcionalidad es fundamental para alimentar el repositorio institucional con nuevos desarrollos de software creados por estudiantes de la Universidad Popular del Cesar.

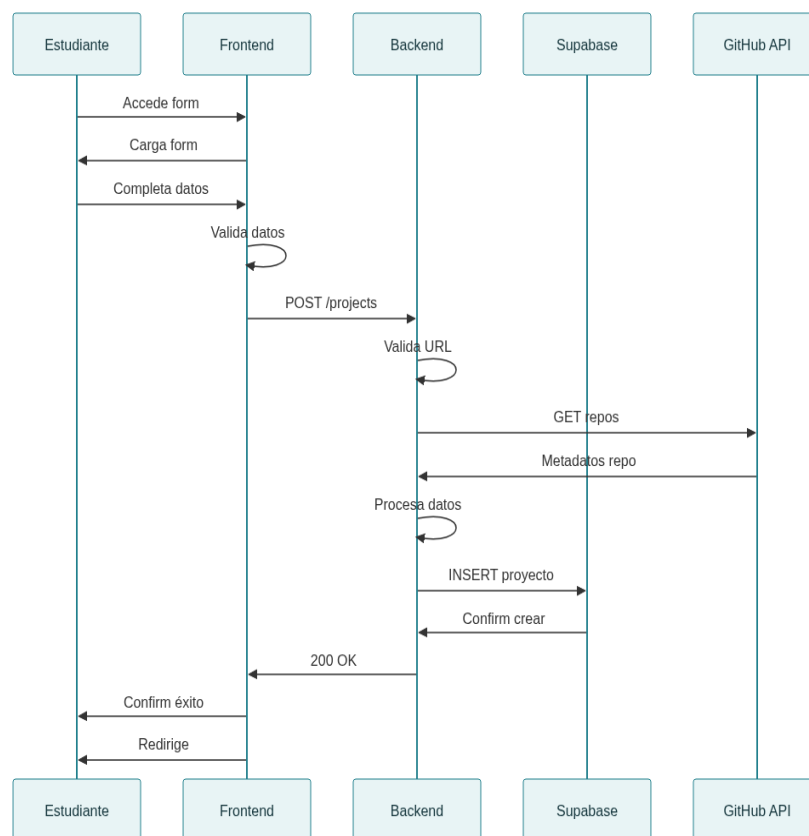


Figura 26: Diagrama de secuencia crear proyecto

Fuente: Elaboración Propia.

4.4.4.3.4.4. SECUENCIA (OBTENER TODOS LOS PROYECTOS)

Este diagrama representa la funcionalidad de catálogo principal del marketplace, donde usuarios de diferentes perfiles pueden explorar la colección completa de proyectos disponibles. El sistema implementa paginación para optimizar el rendimiento y manejo de grandes volúmenes de datos. La secuencia incluye el procesamiento y formateo de metadatos de proyectos para su presentación en formato de tarjetas interactivas. Esta funcionalidad constituye el punto de entrada principal para la exploración y descubrimiento de proyectos institucionales.

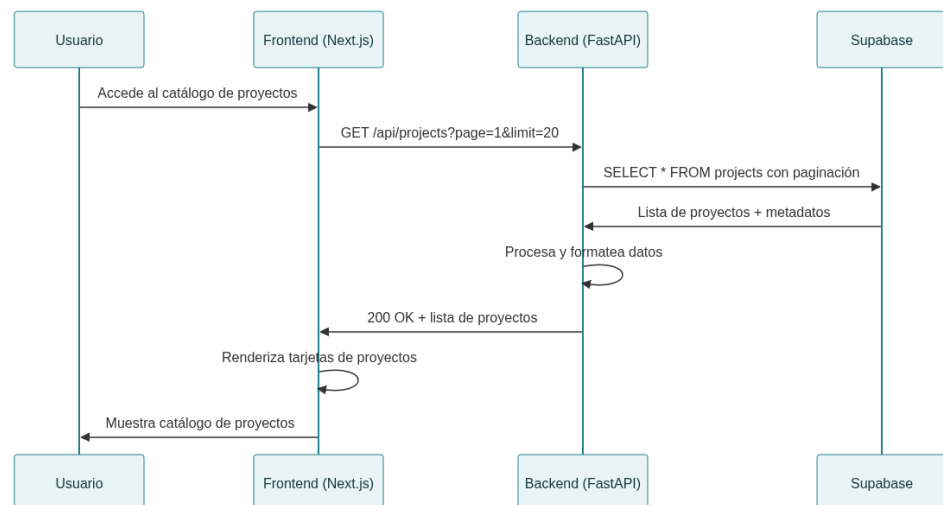


Figura 27: Diagrama de secuencia obtener proyectos

Fuente: Elaboración Propia.

4.4.4.3.4.5. SECUENCIA (COMENTAR PROYECTO)

Este diagrama modela el sistema de interacción social del marketplace, permitiendo que usuarios autenticados generen comentarios en proyectos específicos. La secuencia incluye validación de autenticación, moderación de contenido, y actualización en tiempo real de la

interfaz de usuario. Esta funcionalidad fomenta la colaboración y retroalimentación entre estudiantes, docentes y visitantes, creando un ambiente de aprendizaje interactivo alrededor de los proyectos de desarrollo de software.

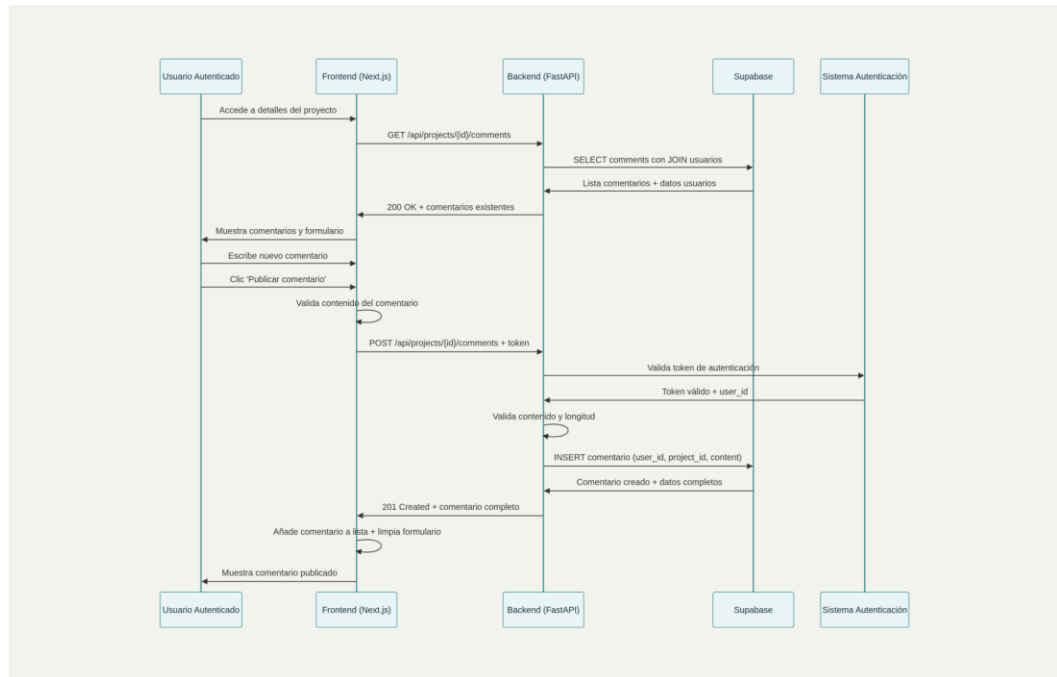


Figura 28: Diagrama de secuencia comentar proyecto

Fuente: Elaboración Propia.

4.4.4.3.4.6. SECUENCIA (LOGIN / INICIO SESION USUARIO)

Este diagrama modela el proceso fundamental de autenticación y control de acceso del marketplace institucional, representando el punto de entrada seguro al sistema para usuarios registrados. La secuencia abarca desde la presentación de opciones de autenticación hasta el establecimiento exitoso de una sesión autenticada, incluyendo múltiples métodos de acceso y validación robusta de credenciales.

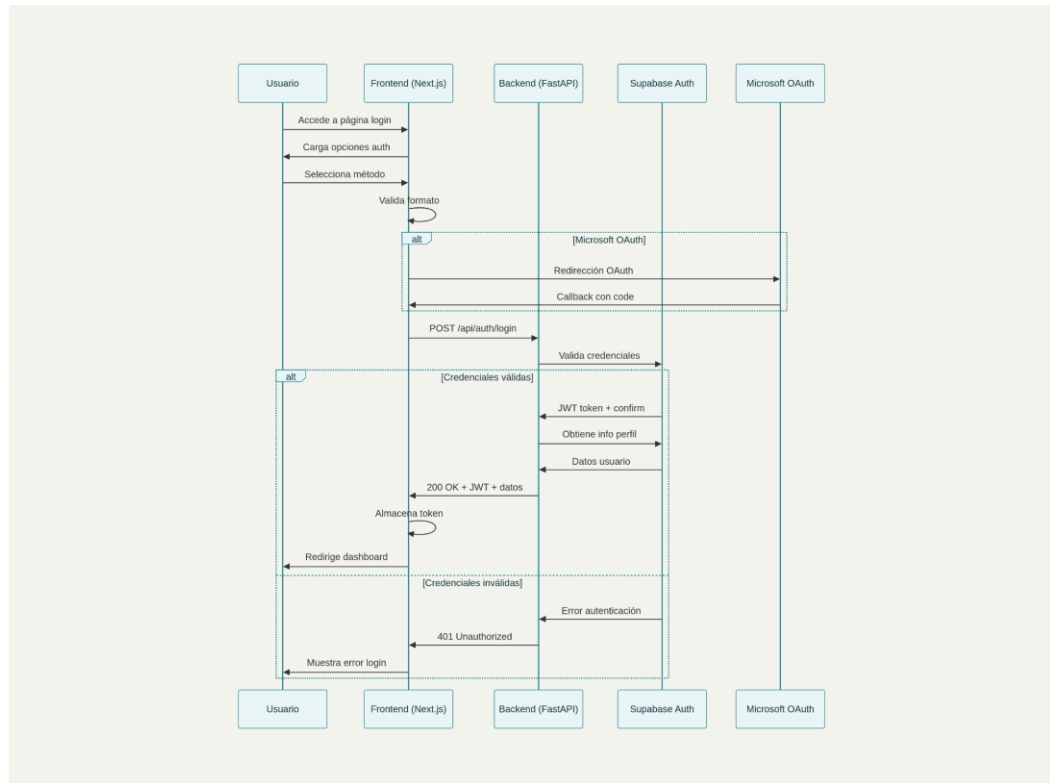


Figura 29: Diagrama de secuencia login

Fuente: Elaboración Propia.

4.4.4.3.4.7. SECUENCIA (REGISTRO DE USUARIO)

Este diagrama modela el proceso de incorporación de nuevos usuarios al marketplace institucional, estableciendo los fundamentos del control de acceso académico. La secuencia implementa validación estricta de emails institucionales (@unicesar.edu.co) para garantizar que solo miembros de la Universidad Popular del Cesar puedan registrarse. El flujo incluye verificación de fortaleza de contraseñas, prevención de registros duplicados, y un sistema robusto de confirmación por correo electrónico que activa la cuenta solo después de verificar la propiedad del email institucional. Esta funcionalidad es crítica para mantener la integridad y exclusividad del repositorio académico.

Diagrama Secuencia: Registro Usuario

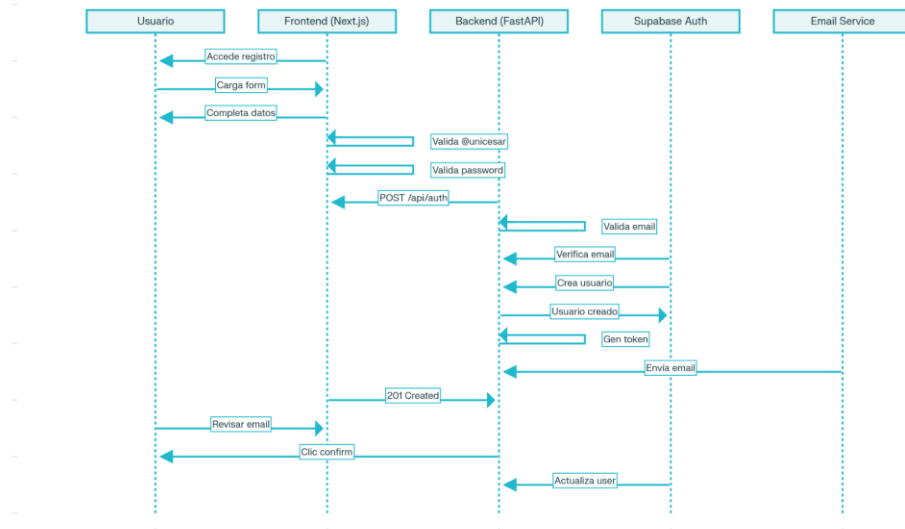


Figura 30: Diagrama de secuencia registro usuario

Fuente: Elaboración Propia.

4.4.4.3.4.8. SECUENCIA (ACTUALIZAR / EDITAR PROYECTO)

Este diagrama representa el proceso de mantenimiento y evolución de proyectos ya publicados en el marketplace. La secuencia incluye validación rigurosa de permisos de propiedad, garantizando que solo el autor original puede modificar su proyecto. El sistema sincroniza automáticamente con GitHub API cuando se actualiza la URL del repositorio, manteniendo la coherencia entre los metadatos almacenados y la información actual del código fuente. Esta funcionalidad es esencial para permitir que los proyectos evolucionen a lo largo del tiempo, reflejando el progreso continuo del desarrollo de software académico.

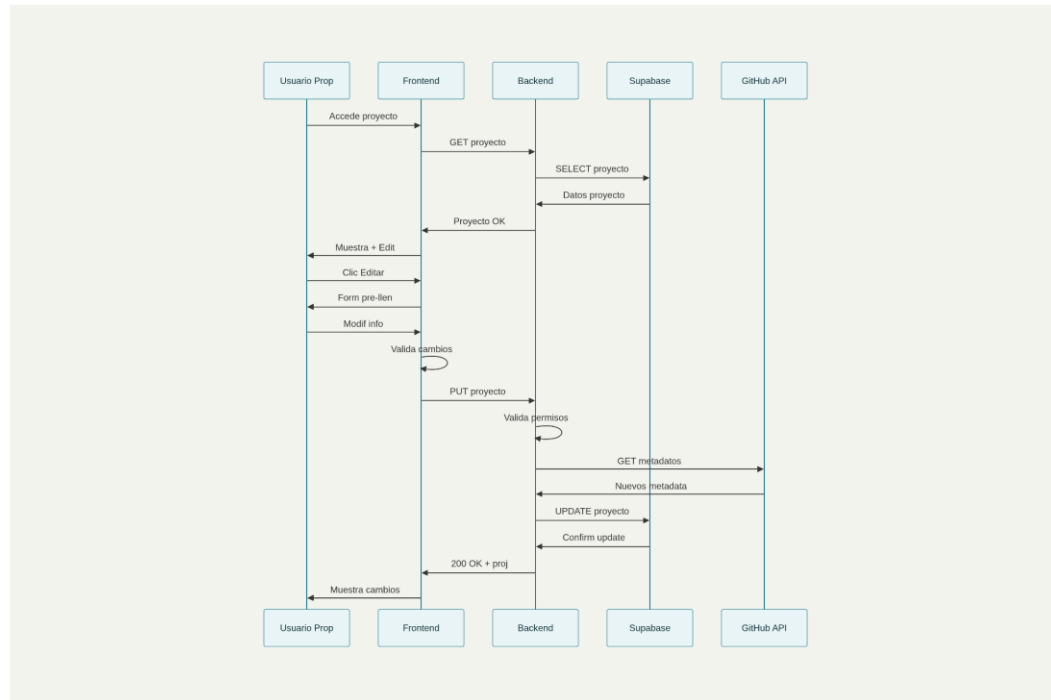


Figura 31: Diagrama de secuencia actualizar

Fuente: Elaboración Propia.

4.4.4.3.4.9. SECUENCIA (ELIMINAR PROYECTO)

Este diagrama mejorado representa un proceso de eliminación ultra-seguro con múltiples capas de protección y confirmación. La secuencia implementa un sistema de confirmación en dos etapas: primero una advertencia inicial sobre la irreversibilidad de la acción, seguida de un modal de confirmación final que requiere múltiples validaciones simultáneas.

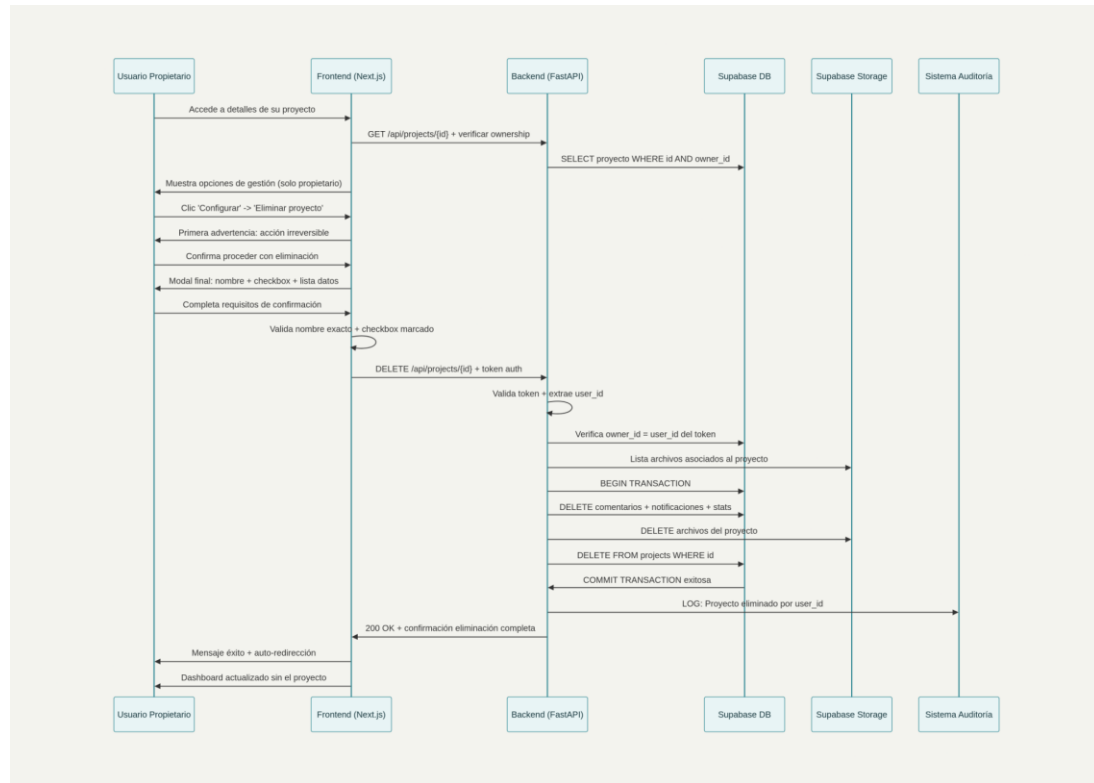


Figura 32: Diagrama de secuencia eliminar proyecto

Fuente: Elaboración Propia.

4.4.4.3.4.10. SECUENCIA (SISTEMA DE NOTIFICACIONES)

Este diagrama modela un sistema de notificaciones híbrido que combina comunicación en tiempo real via WebSockets con notificaciones por email para usuarios offline. La arquitectura implementa un motor de notificaciones basado en triggers de base de datos que detecta automáticamente eventos relevantes (comentarios, menciones, actualizaciones). El sistema respeta las preferencias de privacidad de cada usuario y proporciona control granular sobre qué tipos de notificaciones recibir. La funcionalidad incluye marcado inteligente de notificaciones como leídas y navegación directa al contenido relevante, mejorando

significativamente la experiencia de usuario y fomentando la participación activa en el marketplace institucional.

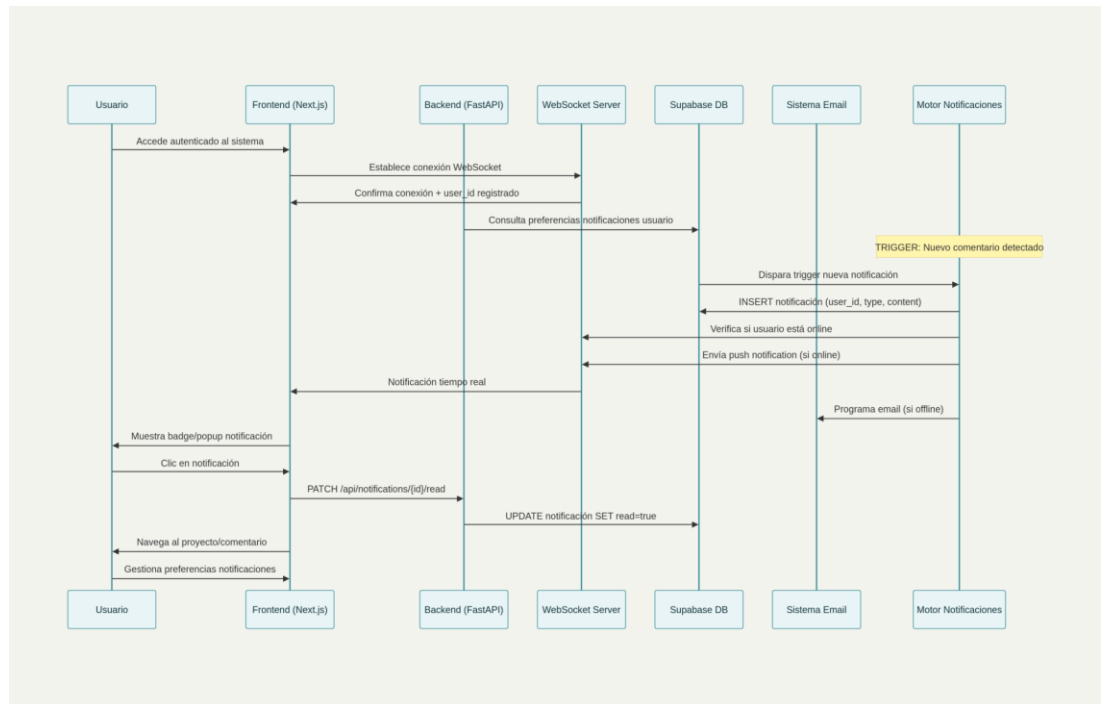


Figura 33: Diagrama de secuencia notificaciones

Fuente: Elaboración Propia.

4.4.4.3.4.11. SECUENCIA (GESTIÓN DE ROLES)

Este diagrama representa el sistema de administración centralizada que permite la gestión segura de roles y permisos en el marketplace institucional. La secuencia incluye validaciones múltiples de permisos administrativos, actualización atómica de roles, e invalidación automática de sesiones activas cuando se modifican permisos de usuario. El sistema implementa un robusto sistema de auditoría que registra todos los cambios administrativos con timestamps y responsables. Esta funcionalidad es crítica para mantener la seguridad y organización del sistema académico, permitiendo la promoción de estudiantes destacados a roles de docente o la gestión de permisos especiales según las necesidades institucionales.

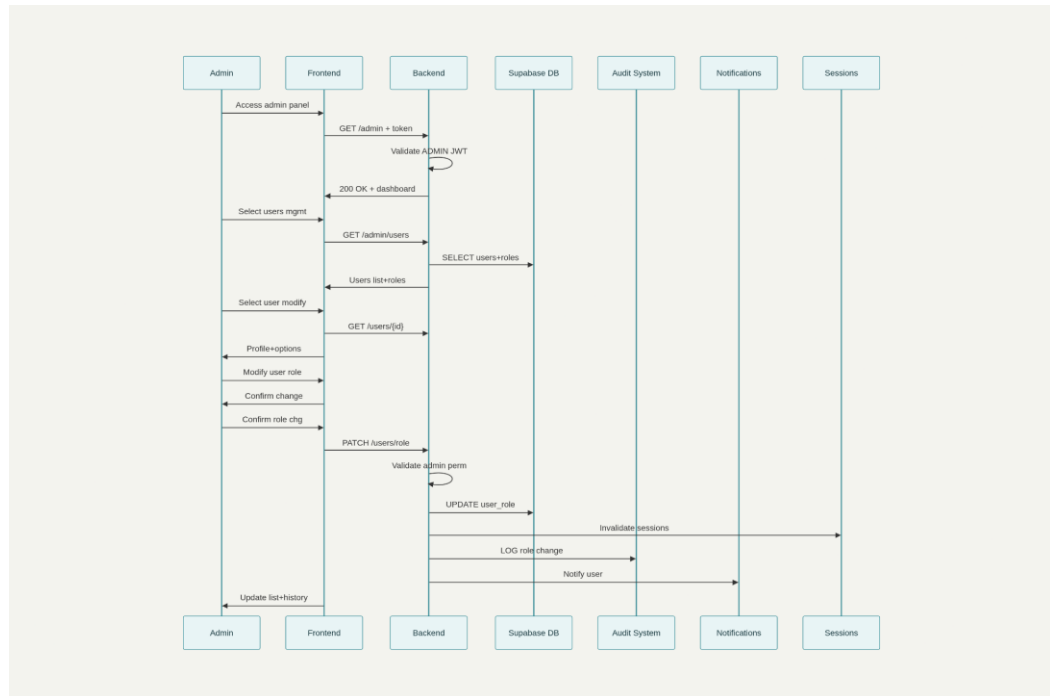


Figura 34: Diagrama de secuencia gestión de roles

Fuente: Elaboración Propia.

4.4.4.3.4.12. SECUENCIA (SINCRONIZACIÓN CON GITHUB)

Este diagrama documenta el sistema automático de sincronización que mantiene actualizada la información de los repositorios GitHub asociados a proyectos del marketplace. El proceso se ejecuta mediante tareas programadas (cron jobs) que consultan periódicamente la GitHub API para obtener estadísticas actualizadas, cambios en el código, y verificar la disponibilidad de repositorios. El sistema implementa procesamiento por lotes eficiente, manejo inteligente de errores (repositorios eliminados o privados), y notificaciones automáticas a propietarios cuando sus repositorios presentan problemas. Esta funcionalidad garantiza que la información mostrada en el marketplace refleje siempre el estado actual de los proyectos de desarrollo, manteniendo la credibilidad y utilidad del repositorio institucional.

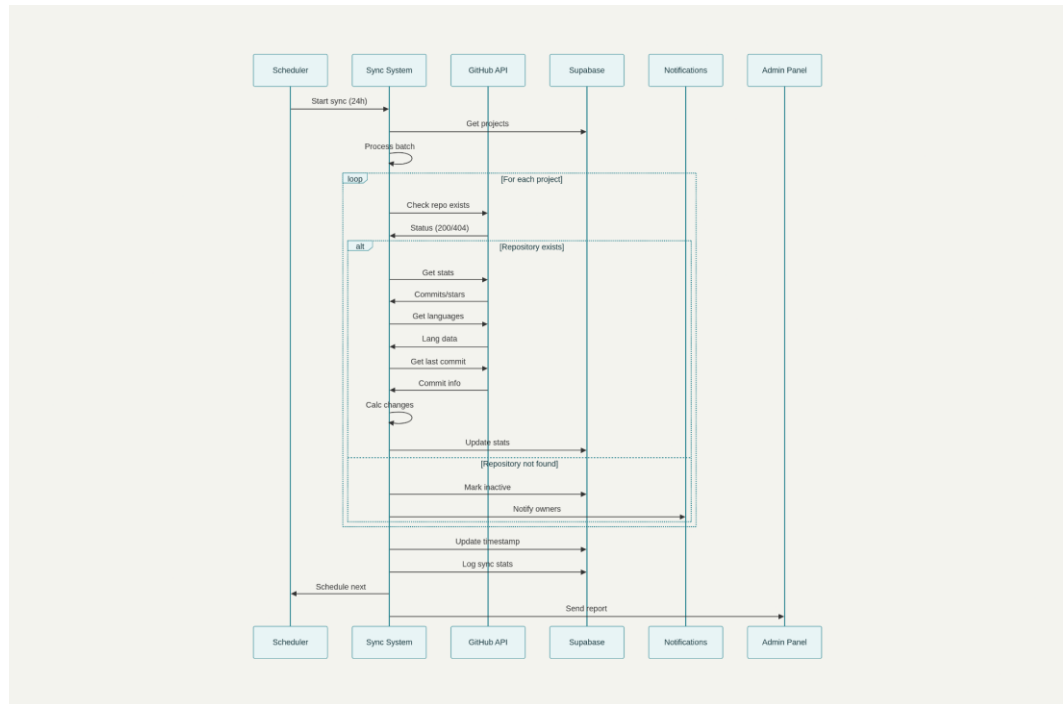


Figura 35: Diagrama de secuencia sincronización github

Fuente: Elaboración Propia.

4.4.4.4. FLUJOS DE NAVEGACIÓN (UX/UI)

En esta sección se presentan los flujos de navegación del marketplace interactivo, los cuales constituyen representaciones visuales que documentan cómo los usuarios interactúan y se desplazan a través de las diferentes funcionalidades del sistema para alcanzar sus objetivos específicos. Los flujos de navegación UX/UI son herramientas fundamentales en el diseño centrado en el usuario que permiten mapear y optimizar cada paso del recorrido del usuario, desde el punto de entrada hasta la culminación exitosa de una tarea determinada.

Para el contexto del repositorio institucional de proyectos de software, estos flujos cobran particular relevancia al definir cómo los estudiantes, docentes y visitantes navegan por el sistema para realizar acciones clave como publicar proyectos, buscar recursos específicos, calificar contenido, descargar archivos o colaborar en desarrollos existentes. Cada flujo ha sido

diseñado considerando las necesidades específicas de los diferentes tipos de usuarios identificados en el proyecto, asegurando que las rutas de navegación sean intuitivas, eficientes y acordes con los patrones de comportamiento esperados.

4.4.4.4.1. FLUJO DE BÚSQUEDA

Intro a la sección de búsqueda que enfoca los filtros, colecciones, palabras clave para encontrar proyectos y para la exploración de ideas.

La sección de búsqueda del marketplace interactivo se constituye como el núcleo central para la exploración y descubrimiento de proyectos de software dentro del repositorio institucional. Este sistema está diseñado para facilitar a los usuarios la localización eficiente de proyectos relevantes mediante múltiples estrategias de navegación que van desde búsquedas simples por palabras clave hasta filtrados avanzados.

Acciones:

1. El usuario accede a la plataforma, ve por primera vez la sección de explorar que se compone de **2 tabs**.
 - a. **Explora la sección de colecciones:** la cual provee acceso rápido a las categorías, aplicaciones populares, destacados en desarrollo de software y nuevos proyectos.
 - b. **Explora la sección todos:** Esta sección se compone de filtros de búsqueda especializados para una búsqueda mucho más selectiva que permita encontrar proyectos con características que el usuario necesite.
 - c. **Ingresa a la vista de un proyecto:** Esta sección se compone de la visualización individual de un proyecto, permitiendo acceder a información general y específica del mismo.

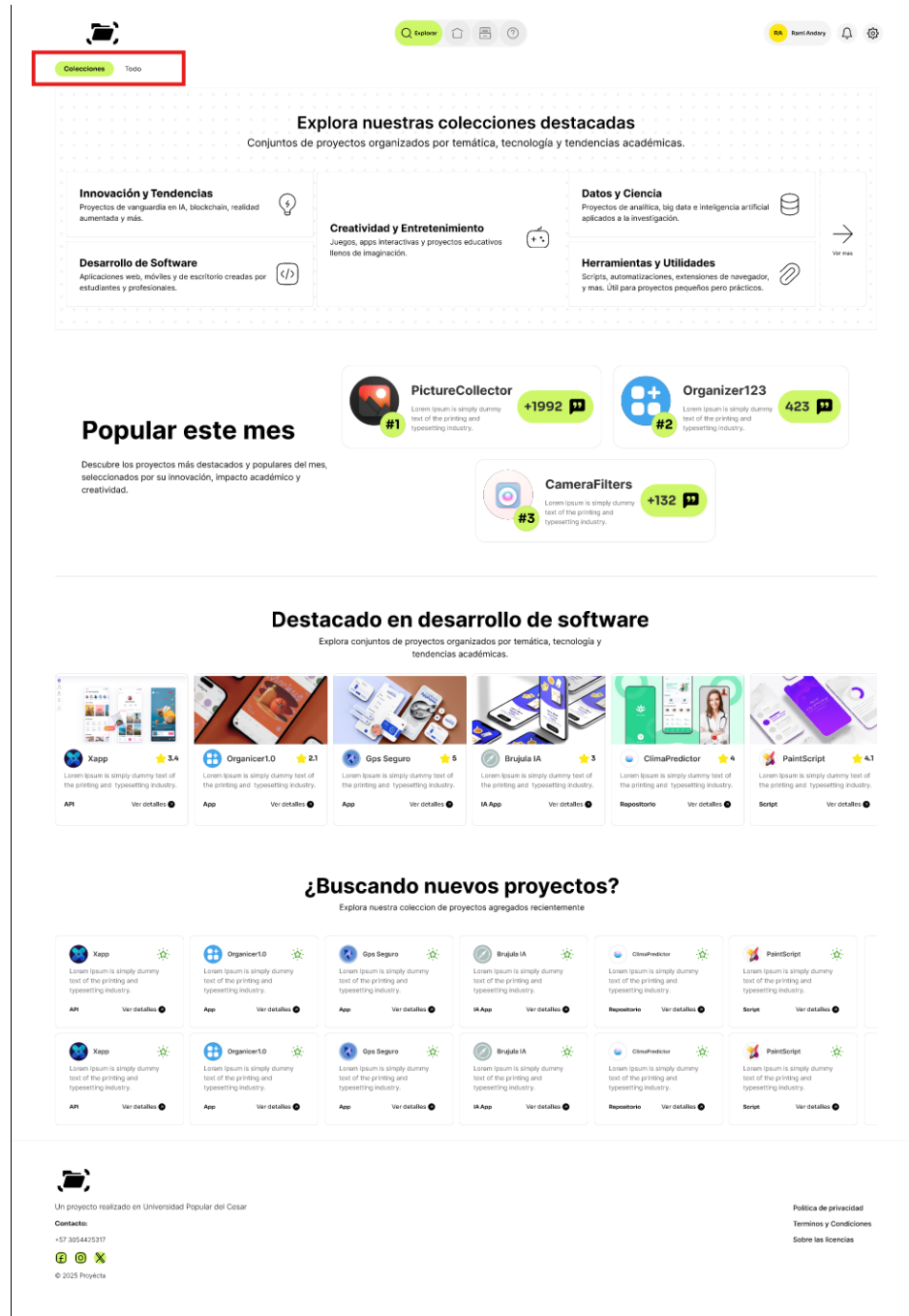


Imagen 1: Tab de colecciones.

Fuente: UX/UI del artefacto (Elaboración Propia).

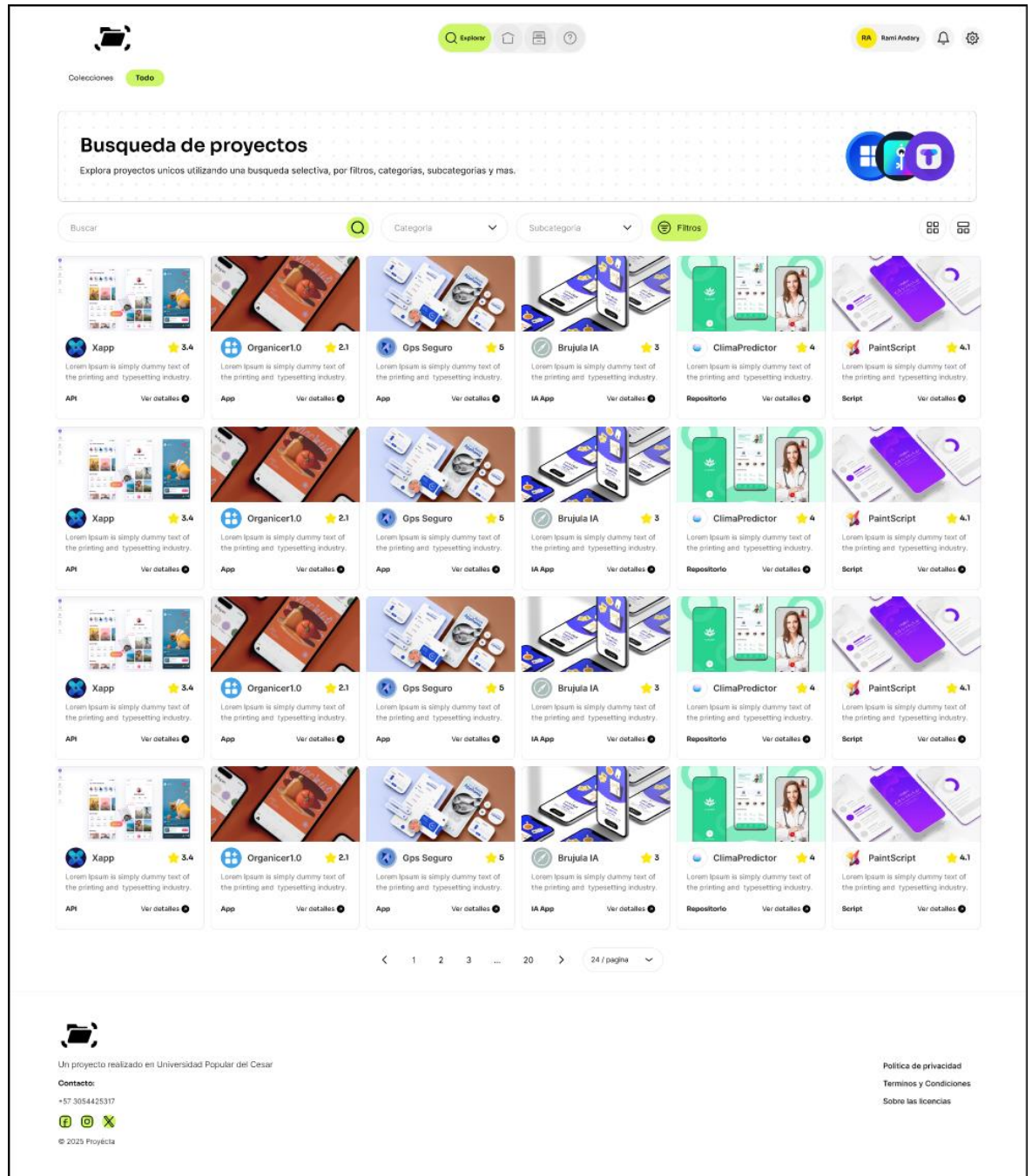


Imagen 2: Tab de búsqueda selectiva (filtros y más).

Fuente: UX/UI del artefacto (Elaboración Propia).

[illegible]

Explora otros proyectos



Un proyecto realizado en Universidad Popular del Cesar

Contact:

+57 3054425317



© 2025 Proyécta

[Política de privacidad](#)

Terminos y Condiciones

[Sobre las licencias](#)

Imagen 3: Visualización de proyecto específico.

Fuente: UX/UI del artefacto (Elaboración Propia).

4.4.4.4.2. FLUJO DE LOGIN

El sistema de autenticación proporciona acceso seguro y personalizado al marketplace institucional. Los usuarios ingresan mediante credenciales institucionales validadas (@unicesar.edu.co) para facilitar el acceso con cuentas universitarias existentes. El proceso incluye validación en tiempo real, gestión de sesiones mediante tokens JWT, y redireccionamiento automático según el rol del usuario (estudiante, docente, administrador). La interfaz presenta opciones claras de inicio de sesión con recordatorio de credenciales y enlaces de recuperación, manteniendo la seguridad mediante autenticación de dos factores y encriptación HTTPS.

Acciones:

1. Al presionar el botón de conectarse al usuario se le presenta una ventana emergente (popup) que le permitirá acceder al sistema a través de su correo y contraseña o incluso utilizando OAuth con Microsoft.

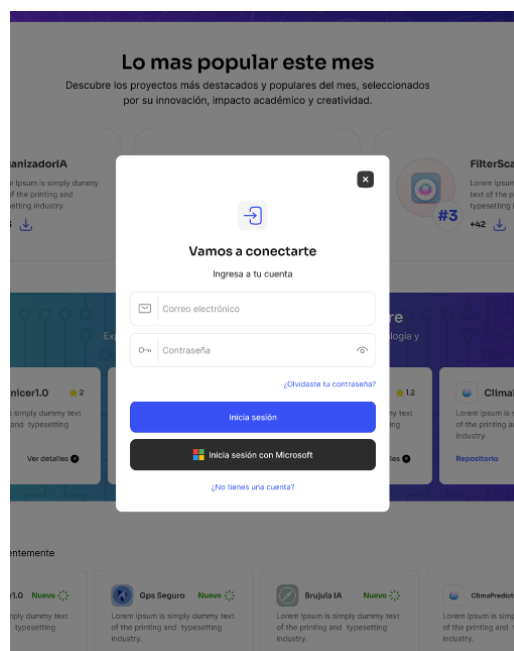


Imagen 4: Popup de login.

Fuente: UX/UI del artefacto (Elaboración Propia).

2. A continuación deberá digitar la información necesaria para realizar el inicio de sesión.

4.4.4.4.3. FLUJO DE REGISTRO

La incorporación de nuevos usuarios al marketplace implementa un proceso de validación institucional riguroso. El sistema requiere email universitario (**@unicesar.edu.co**), validación de datos académicos (programa, semestre), y confirmación por correo electrónico antes de activar la cuenta. El formulario captura información esencial: nombre completo, programa académico, semestre actual, y área de especialización. Tras el registro, se envía automáticamente un email de confirmación con enlace de activación que expira en 24 horas, garantizando que solo miembros legítimos de la universidad accedan al repositorio.

Acciones:

1. Para poder registrarse el usuario deberá previamente seleccionar en el login el hipervínculo que dice: ¿No tienes una cuenta? este redirigirá al usuario al registro.

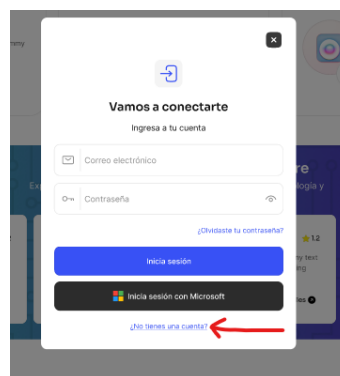
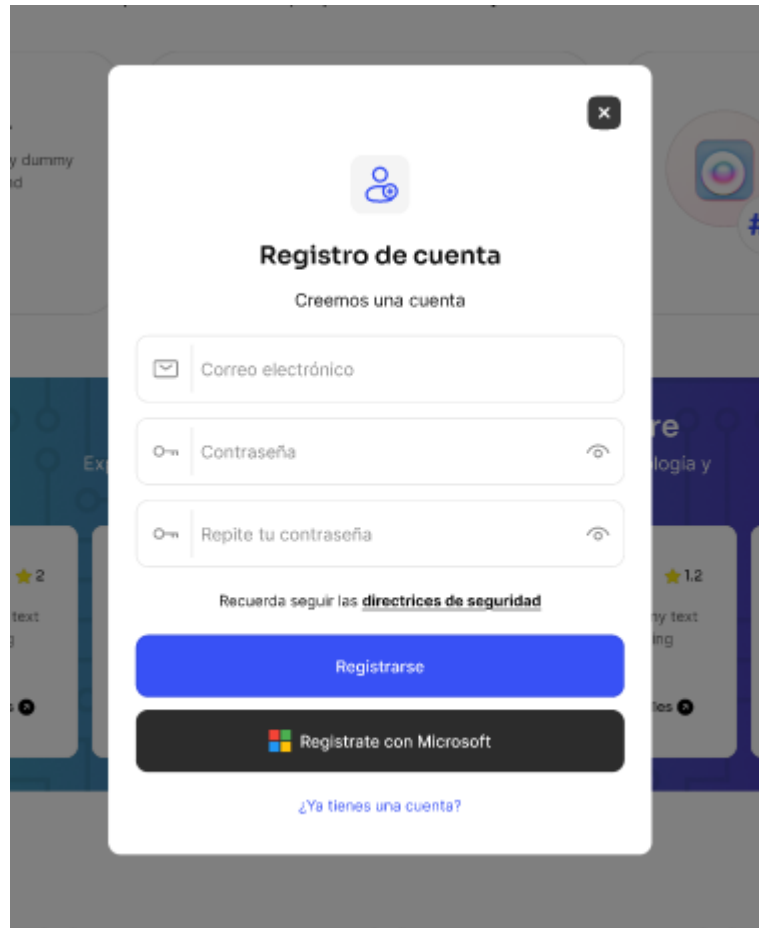


Imagen 5: Registro de cuenta desde login.

Fuente: UX/UI del artefacto (Elaboración Propia).

2. Una vez en la ventana de registro el usuario deberá elegir su tipo de registro, si lo hace en la plataforma directamente o utilizando los servicios de Microsoft como forma de registro.



Registro de cuenta

Creemos una cuenta

Correo electrónico

Contraseña

Repite tu contraseña

Recuerda seguir las [directrices de seguridad](#)

Registrarse

Regístrate con Microsoft

[¿Ya tienes una cuenta?](#)

Imagen 6: Ventana de registro.

Fuente: UX/UI del artefacto (Elaboración Propia).

3. El usuario deberá proporcionar los datos requeridos para su registro en la plataforma sea la opción que haya elegido en el paso previo.

4.4.4.4. FLUJO DE OLVIDAR CONTRASEÑA

El sistema de recuperación de contraseñas garantiza acceso continuo sin comprometer la seguridad. Los usuarios solicitan restablecimiento mediante email institucional, activando un proceso que genera enlaces seguros con expiración de 1 hora. El flujo incluye verificación de identidad mediante preguntas de seguridad, envío de código de verificación al email registrado, y formulario seguro para establecer nueva contraseña con requisitos de fortaleza. El sistema registra todos los intentos de recuperación en logs de auditoría y notifica al usuario sobre cambios exitosos en sus credenciales.

Acciones:

1. Para acceder a este flujo, el usuario deberá haber seleccionado la opción de **olvidar contraseña** desde la **ventana de login**.

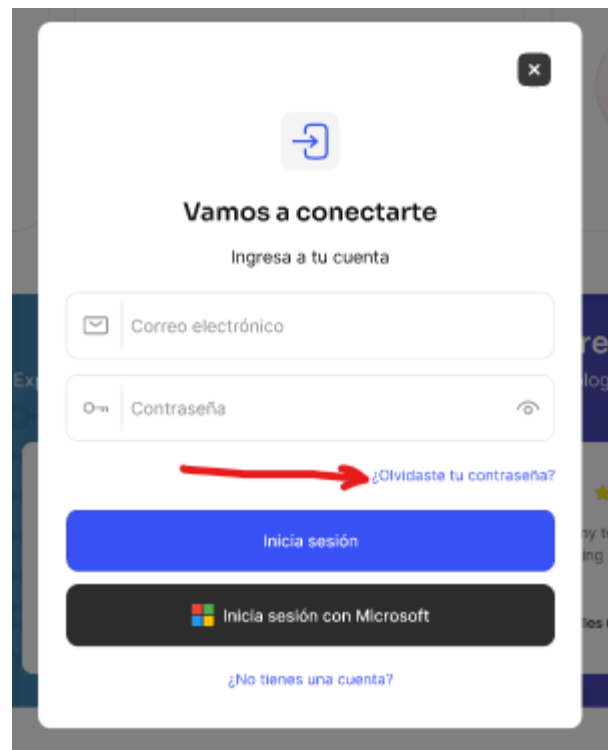


Imagen 7: Selección de olvidar contraseña desde ventana de login.

Fuente: UX/UI del artefacto (Elaboración Propia).

2. Una vez en la ventana de **Olvidar contraseña** se le solicitará el correo electrónico utilizado comúnmente para acceder a la plataforma para poder enviar un código de restablecimiento que le permitirá cambiar la contraseña e iniciar sesión nuevamente.



Imagen 8: Ventana de Olvidar Contraseña.

Fuente: UX/UI del artefacto (Elaboración Propia).

3. Una vez validado el correo el sistema solicitará el código enviado e incluso permite enviar uno nuevamente pasado un tiempo si nunca llegó. El usuario deberá presionar el botón de validar código una vez digite el código que le llegó al correo.



Imagen 9: Paso de restablecimiento de contraseña.

Fuente: UX/UI del artefacto (Elaboración Propia).

4. Una vez validado el código, el sistema solicitará una nueva contraseña para la cuenta. El usuario deberá repetir la contraseña y seguir las directrices de seguridad. Una vez colocada y validada la nueva contraseña el usuario presiona restablecer y el sistema le solicitará iniciar sesión desde el login.

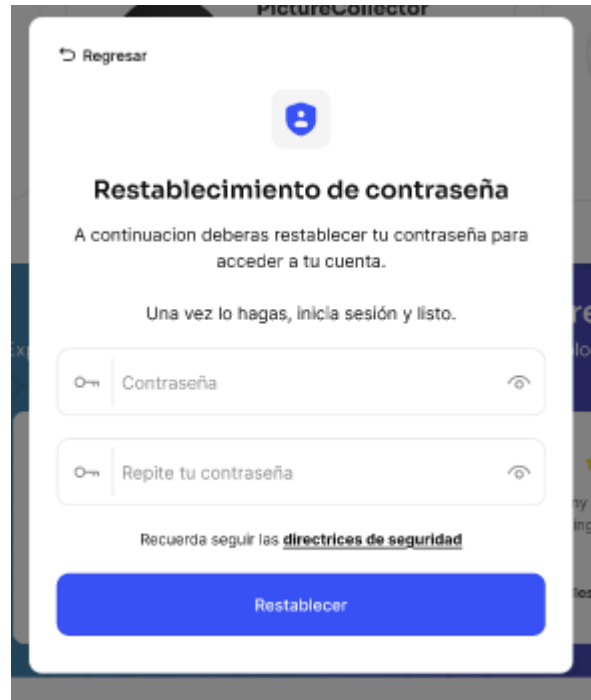


Imagen 10: Paso 2 de restablecimiento de contraseña.

Fuente: UX/UI del artefacto (Elaboración Propia).

4.4.4.4.5. FLUJO DE CREACIÓN DE PROYECTOS

La publicación de proyectos constituye la funcionalidad central del marketplace académico. El proceso guía a estudiantes a través de un formulario estructurado que captura metadatos esenciales: título, descripción, categoría tecnológica, URL del repositorio GitHub, palabras clave, y archivos complementarios. El sistema valida automáticamente la existencia del repositorio, extrae información técnica (README, licencia, estadísticas), e indexa el contenido para facilitar su descubrimiento. Una vez publicado, el proyecto aparece inmediatamente en el catálogo con estado "Nuevo" y notificaciones automáticas a docentes supervisores.

Acciones:

1. Para acceder a este flujo, el usuario deberá haber seleccionado el botón de configuración de la página principal.



Imagen 11: Paso 1 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).

2. Una vez en el dashboard principal del usuario, el usuario deberá seleccionar el botón que dice publicar nuevo proyecto.

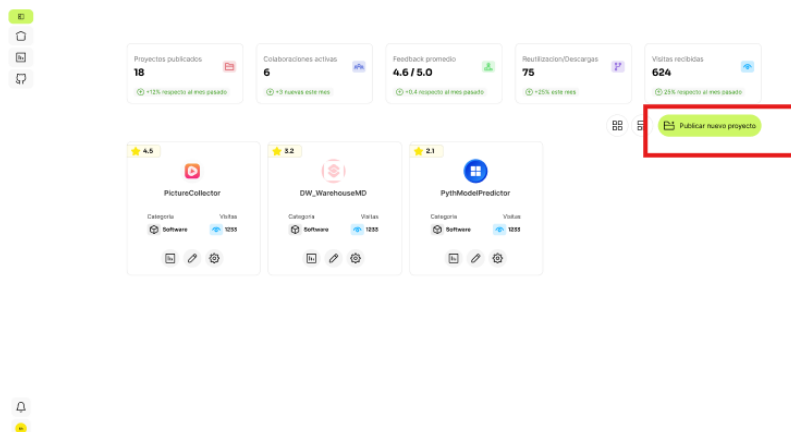


Imagen 12: Paso 2 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).

3. A continuación el usuario deberá seguir una serie de pasos en los que debe llenar información relacionada al proyecto. (nombre del proyecto, descripciones, iconos, características, imágenes, ficheros, etc)



Imagen 13: Paso 3 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).

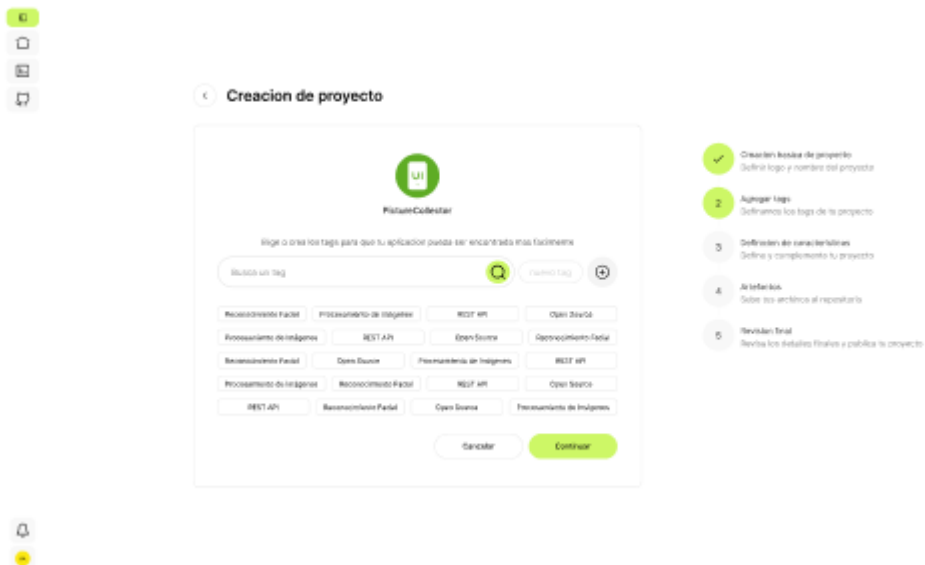
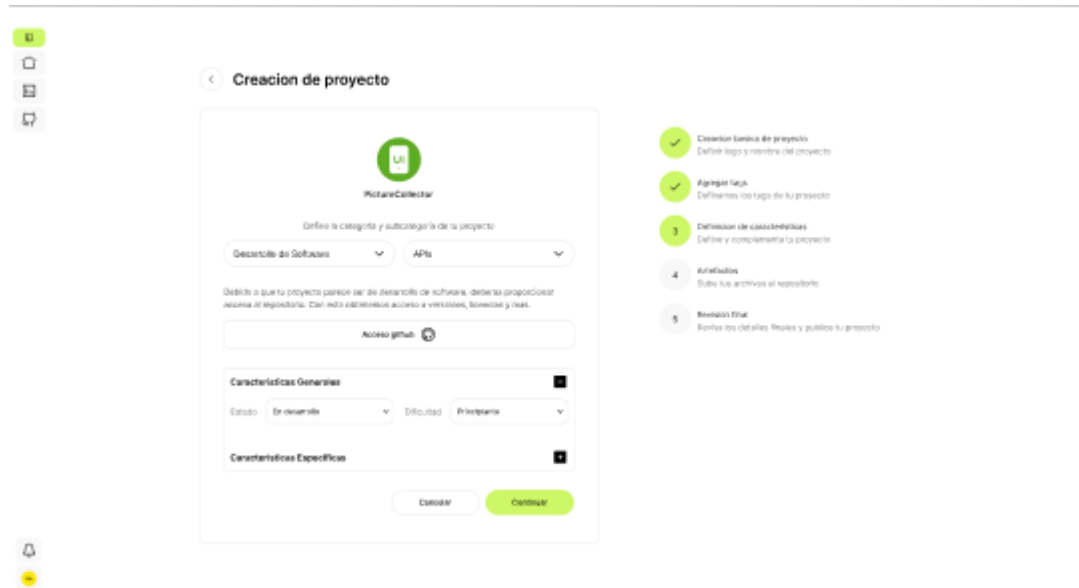


Imagen 14: Paso 4 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).



Creacion de proyecto

PictureCollector

Define la categoria y subcategoria de tu proyecto

Desarrollo de Software APIs

Debido a que tu proyecto parece ser de desarrollo de software, deberas proporcionar acceso al repositorio. Con estos datos se accedera a versiones, ramas y mas.

Acceso github

Características Generales

Estado: En desarrollo Prioridad: Prioritaria

Características Específicas

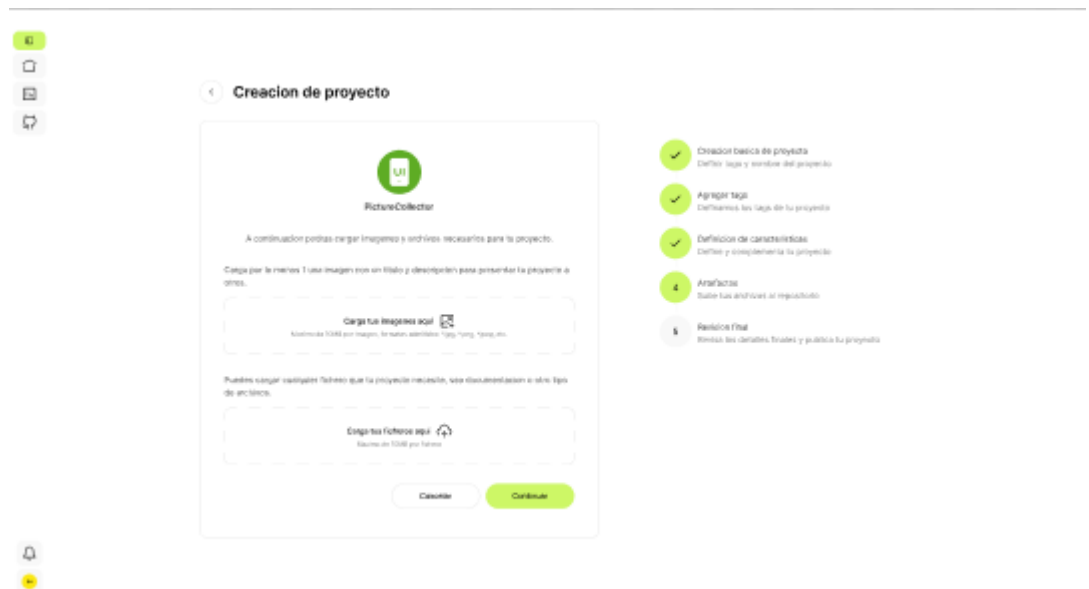
Crear Continuar

Progreso:

1. Creacion basica de proyecto: Definir tags y nombre del proyecto
2. Agregar tags: Definir los tags de tu proyecto
3. Definicion de caracteristicas: Definir y complementar la proyecto
4. Analisis: Subir tus archivos al repositorio
5. Revision final: Revisar los detalles, Revisar y publicar tu proyecto

Imagen 15: Paso 5 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).



Creacion de proyecto

PictureCollector

A continuación podras cargar imagenes y archivos necesarios para tu proyecto.

Carga por lo menos 1 una imagen con un titulo y descripción para presentar la proyecto a otros.

Carga tus imagenes aqui

Maximum 10MB per image, for more, see: https://img.sharing.com

Puedes cargar cualquier archivo que tu proyecto necesite, solo documentacion en otro tipo de archivos.

Carga tus archivos aqui

Maximum 10MB per file

Crear Continuar

Progreso:

1. Creacion basica de proyecto: Definir tags y nombre del proyecto
2. Agregar tags: Definir los tags de tu proyecto
3. Definicion de caracteristicas: Definir y complementar la proyecto
4. Analisis: Subir tus archivos al repositorio
5. Revision final: Revisar los detalles, Revisar y publicar tu proyecto

Imagen 16: Paso 6 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).

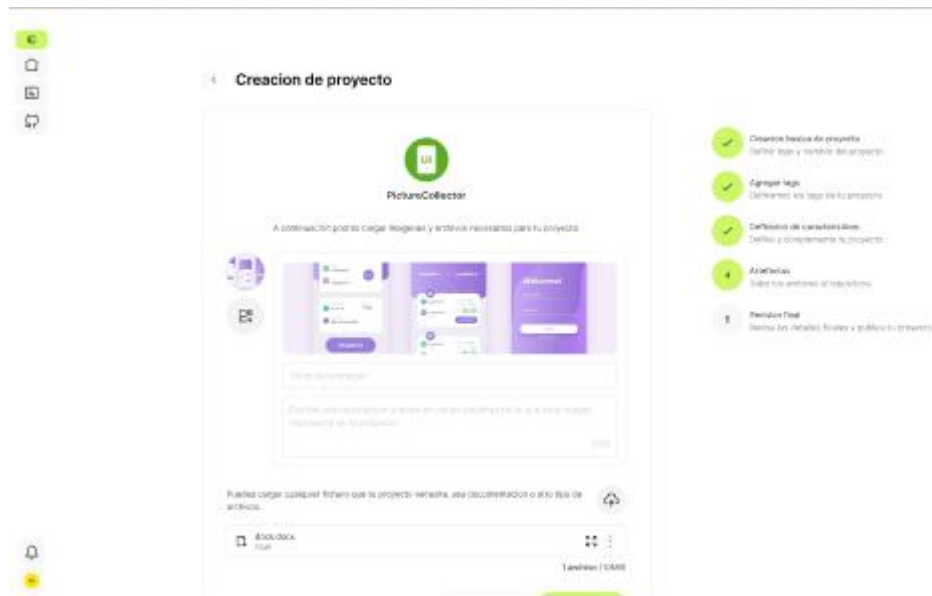


Imagen 17: Paso 7 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).

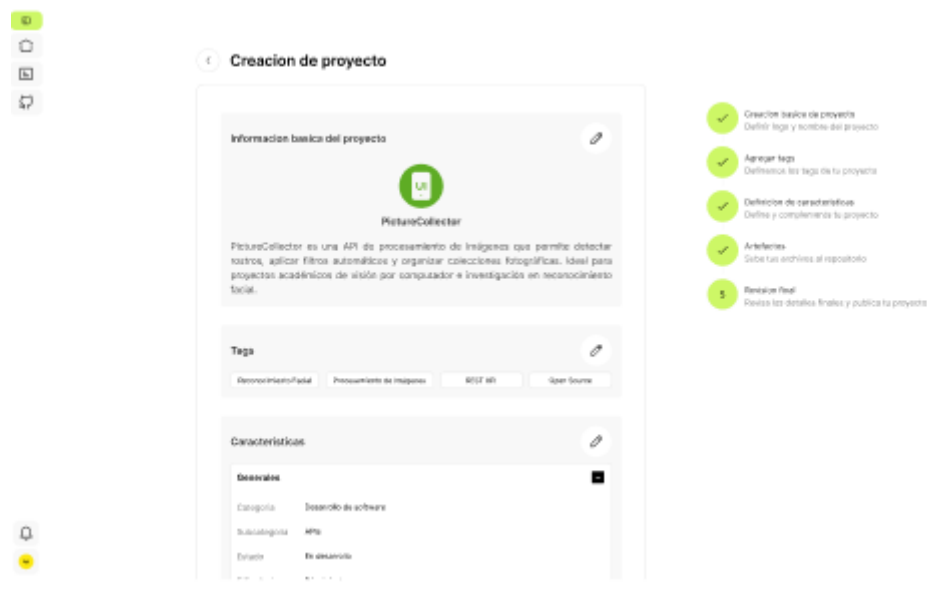


Imagen 18: Paso 8 de flujo de creación de proyectos.

Fuente: UX/UI del artefacto (Elaboración Propia).

4.4.4.4.6. FLUJO DE FEEDBACK

El sistema de interacción social fomenta la colaboración mediante múltiples mecanismos de retroalimentación. Incluye contador automático de visualizaciones que se incrementa con cada visita única al proyecto, sistema de favoritos para que usuarios marquen proyectos de interés creando listas personalizadas, y comentarios estructurados donde estudiantes y docentes proporcionan retroalimentación constructiva. El flujo implementa notificaciones en tiempo real vía WebSockets cuando se generan comentarios, moderación automática de contenido inapropiado, y métricas de engagement que influyen en los rankings de proyectos destacados. Los comentarios soportan menciones (@usuario), respuestas anidadas, y sistema de valoración que ayuda a identificar feedback más útil para los autores.

Acciones:

1. Para acceder a las interacciones de feedback, en cada proyecto hay una sección de ratings, en esta sección los usuarios podrán interactuar con el contenido publicado.

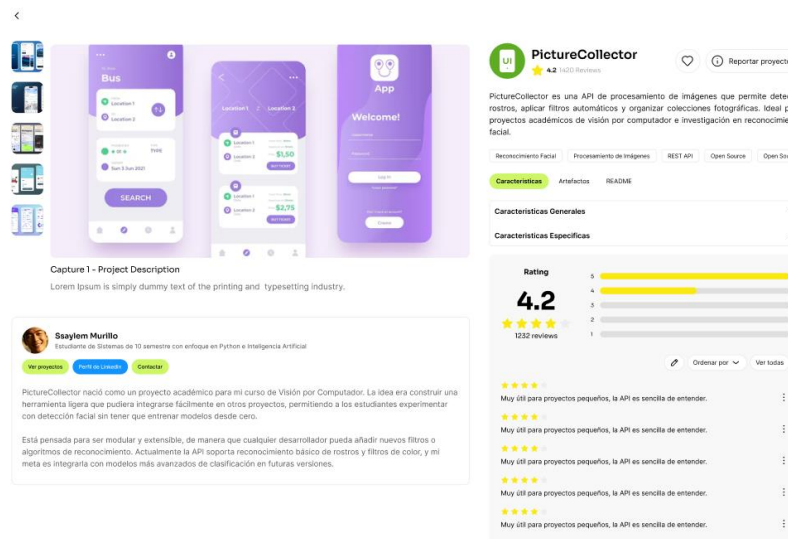


Imagen 19: Vista individual de proyecto, flujo de feedback.

Fuente: UX/UI del artefacto (Elaboración Propia).

- En la sección de ratings se encuentra un botón con logo de lápiz que permitirá crear a los usuarios interacciones con el contenido.

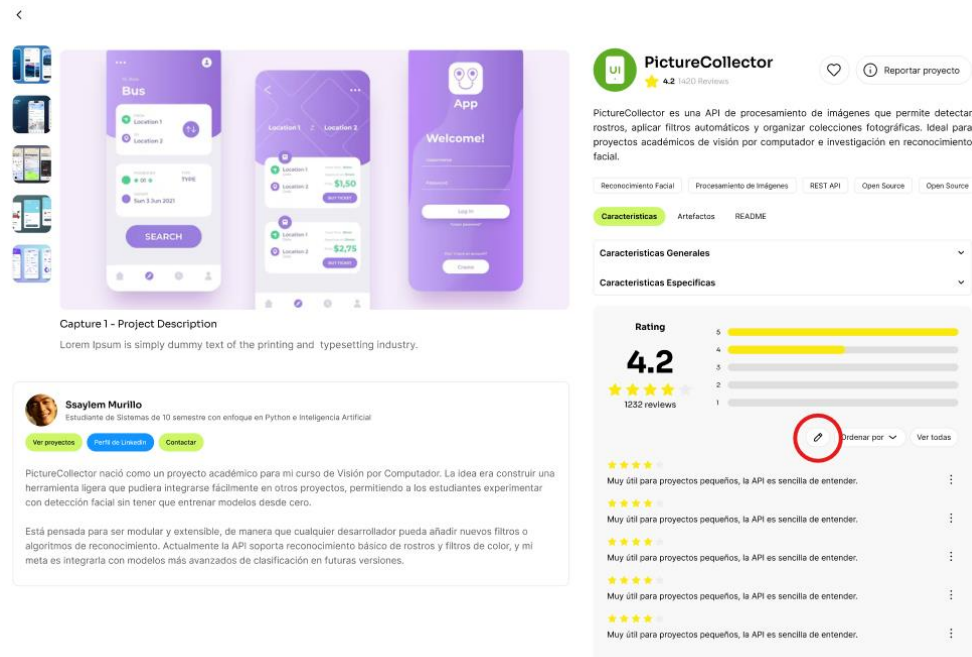


Imagen 20: Vista individual de proyecto, flujo de feedback, botón de edición.

Fuente: UX/UI del artefacto (Elaboración Propia).

- A continuación el usuario deberá proporcionar una métrica de calificación en escala del 1 al 5 y proveer un comentario para su interacción.

4.4.4.4.7. FLUJO DE MÉTRICAS

El sistema de métricas constituye el núcleo analítico del marketplace institucional, proporcionando datos cuantitativos y cualitativos esenciales para evaluar el impacto, rendimiento y valor académico de los proyectos de software publicados. Este flujo implementa

un conjunto completo de indicadores que permiten a los usuarios comprender el comportamiento y efectividad de la plataforma.

Acciones:

1. Para acceder a este flujo, el usuario deberá haber seleccionado el botón de configuración de la página principal.



Imagen 21: Paso 1 de flujo de visualización de métricas.

Fuente: UX/UI del artefacto (Elaboración Propia).

2. Una vez en el dashboard principal del usuario, hay 2 formas de acceder a métricas, métricas individuales y métricas generales. El usuario deberá seleccionar las métricas que desea ver.

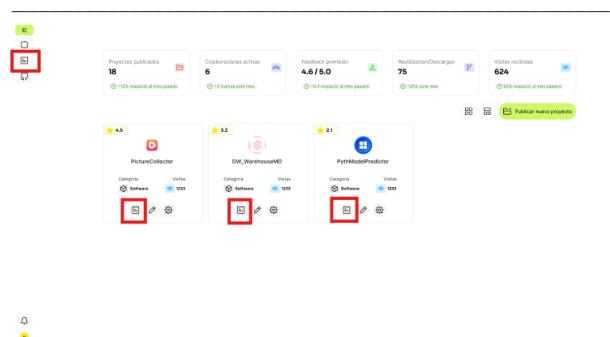


Imagen 22: Paso 2 de flujo de visualización de métricas.

Fuente: UX/UI del artefacto (Elaboración Propia).

- Una vez seleccionada el tipo de métrica, el usuario podrá acceder a reportes obtenidos en la aplicación sobre sus proyectos y el impacto de los mismos en la comunidad.

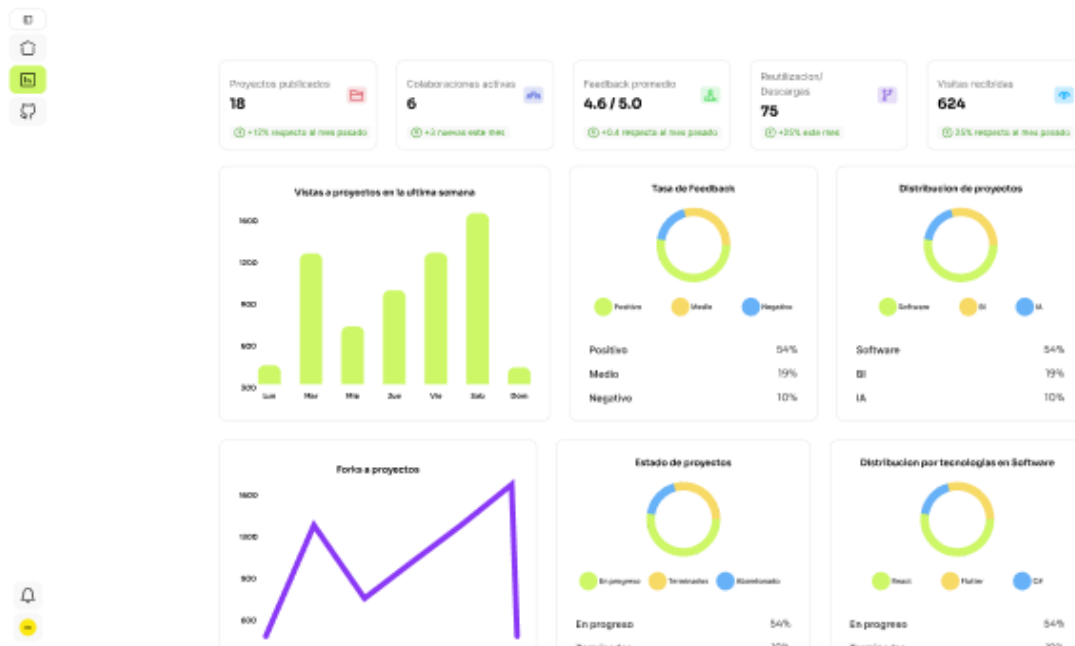


Imagen 23: Paso 3 de flujo de visualización de métricas.

Fuente: UX/UI del artefacto (Elaboración Propia).

4.4.5.1 IMPLEMENTACIÓN DE LA ARQUITECTURA DISEÑADA

La construcción del artefacto se realizó siguiendo la arquitectura definida previamente, de modo que cada capa y módulo planteado en el diseño se materializa en componentes concretos dentro del sistema. Esta correspondencia permite evidenciar coherencia entre la propuesta arquitectónica y su implementación.

El sistema se desarrolló bajo principios de arquitectura modular, separación de responsabilidades, escalabilidad estructural, seguridad en la gestión de usuarios e interoperabilidad con servicios externos. Para ello, se adoptó un modelo cliente-servidor que diferencia claramente las capas de presentación, lógica de negocio y persistencia de datos, facilitando la mantenibilidad y evolución del sistema.

En el backend, la arquitectura en capas se implementa mediante la separación de módulos encargados de la exposición de la API, la lógica de negocio y el acceso a datos. Las rutas gestionan las peticiones HTTP y delegan el procesamiento en servicios especializados, los cuales encapsulan las reglas de negocio y coordinan las operaciones sobre los modelos y la base de datos. Por su parte, los esquemas y utilidades de configuración aseguran la validación de la información y la correcta integración con la infraestructura.

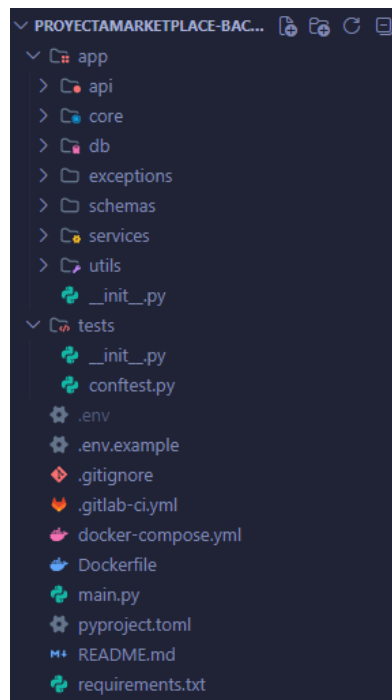


Imagen 24: Arquitectura de la aplicación backend reflejada en módulos (carpetas).

Fuente: Screenshot estructura del proyecto backend.

En el frontend, la arquitectura propuesta se materializó mediante una estructura modular organizada en torno al directorio principal de la aplicación, donde cada sección funcional (como autenticación, gestión de proyectos, visualización del catálogo o panel de métricas) se implementa en rutas y subcarpetas específicas que agrupan sus páginas, componentes y estilos asociados. Esta organización se complementa con un conjunto de componentes reutilizables y librerías de soporte para servicios, hooks y utilidades.

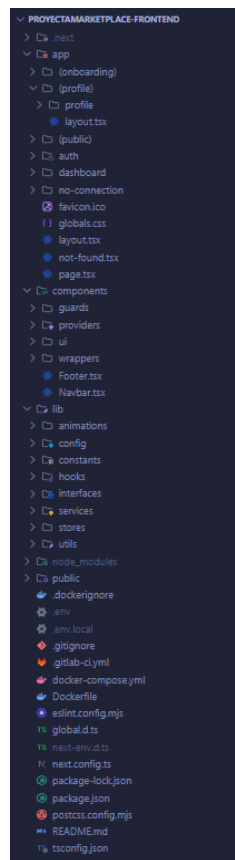


Imagen 25: Arquitectura de la aplicación frontend reflejada en módulos (carpetas).

Fuente: Screenshot estructura del proyecto frontend.

Elemento del diseño (diagrama)	Implementado en
--------------------------------	-----------------

Arquitectura por capas	Estructura de directorios routes/, services/, repositories/
Arquitectura basada en componentes	Módulos Next.js por funcionalidad (frontend)

4.4.4.5. METODOLOGÍA DE DESARROLLO UTILIZADA

En la fase de construcción del artefacto se adoptó un enfoque de desarrollo ágil híbrido, combinando Scrum como marco de gestión para organizar el trabajo de forma incremental e iterativa, y prácticas de Extreme Programming (XP) como soporte de ingeniería para fortalecer la calidad del software y la entrega continua de funcionalidades. Esta decisión respondió a la necesidad de implementar rápidamente un conjunto de módulos funcionales del marketplace, manteniendo trazabilidad del avance y control del alcance mediante tareas verificables y evidencias de desarrollo.

4.4.4.5.1. ENFOQUE ADOPTADO Y JUSTIFICACIÓN

Para la construcción del artefacto se empleó un enfoque ágil híbrido, combinando Scrum como marco de gestión del trabajo e incorporando prácticas de Extreme Programming (XP) como apoyo de ingeniería de software. Este enfoque se seleccionó debido a la necesidad de desarrollar un conjunto de funcionalidades en un periodo acotado, manteniendo visibilidad del avance, control del alcance y una dinámica de mejora continua del código durante la implementación.

En términos operativos, la ejecución se organizó mediante un sprint único extendido, en el cual se consolidó el conjunto de tareas necesarias para completar el incremento funcional del sistema (MVP). Aunque Scrum se concibe típicamente como un proceso iterativo con múltiples sprints, la adopción de un sprint único resultó adecuada para el contexto del proyecto (equipo reducido y alcance priorizado), siempre que se mantuvieran los principios de gestión por backlog, seguimiento visual del trabajo y verificación del “hecho” por cada tarea completada.

4.4.4.5.2. ARTEFACTOS SCRUM UTILIZADOS

La operacionalización del enfoque Scrum se realizó a partir de tres artefactos principales:

1. **Product Backlog:** se consolidó un listado de actividades y funcionalidades a implementar, derivadas del backlog priorizado (MoSCoW) y de los criterios de aceptación definidos previamente en el documento. Este backlog se tradujo a ítems de trabajo gestionados en Azure DevOps para su seguimiento y trazabilidad.
2. **Sprint Backlog:** para el sprint único, se seleccionaron y agruparon los ítems a desarrollar, asignándolos a responsables y estados de avance dentro del tablero (To Do / Doing / Done), permitiendo observar el flujo de trabajo y el cierre progresivo de tareas.
3. **Incremento:** cada ítem marcado como “Done” representó una unidad de incremento verificable dentro del sistema, respaldada por evidencias adjuntas (capturas, avances o resultados técnicos) registradas en las tareas, lo que fortaleció la trazabilidad entre lo planificado y lo construido.

Como criterio de control, el tablero permitió verificar el estado global del sprint (concentración de ítems en “Done” y asignación por responsable), evidenciando la culminación de bloques de trabajo relacionados con desarrollo frontend, backend e implementación general del marketplace.

4.4.4.5.3. PRÁCTICAS XP APLICADAS

Como complemento a la gestión ágil del trabajo, se adoptaron prácticas de XP orientadas a sostener la calidad del producto durante la construcción:

- **Programación en parejas (pair programming):** se aplicó en tareas críticas o de mayor complejidad (por ejemplo, integración de componentes o funcionalidades con dependencias), con el objetivo de mejorar la revisión inmediata del código, compartir conocimiento y reducir defectos tempranos.
- **Pruebas durante el desarrollo:** se promovió la verificación técnica progresiva para evitar regresiones y asegurar estabilidad del incremento, articulando pruebas rápidas/manuales de endpoints y pruebas automatizadas cuando correspondía. Este

enfoque se alinea con lo descrito en el documento respecto a herramientas de pruebas utilizadas (p. ej., pruebas exploratorias y pruebas automatizadas).

- **Refactorización continua:** se realizaron ajustes iterativos sobre la base de código a medida que emergían necesidades de mantenibilidad (reducción de duplicación, mejor separación de responsabilidades, mejoras de legibilidad), procurando conservar el comportamiento funcional del sistema y disminuir deuda técnica.

4.4.4.5.4. CONSIDERACIONES DE MEJORA DEL PROCESO (APRENDIZAJES)

A partir de la ejecución del sprint único, se identificaron oportunidades para madurar el proceso ágil en iteraciones futuras: (i) descomponer el trabajo en múltiples sprints de menor tamaño para mejorar la predictibilidad, (ii) formalizar una Definition of Done explícita (código integrado, evidencia adjunta, prueba mínima ejecutada), (iii) incorporar estimación ligera (p. ej., story points) y (iv) realizar retrospectivas registradas para sistematizar mejoras de proceso. Estas acciones permitirían fortalecer la disciplina Scrum sin alterar la viabilidad operativa del equipo y del contexto del proyecto.

4.4.4.6. PLANIFICACIÓN Y SEGUIMIENTO DEL DESARROLLO

La planificación y el seguimiento del desarrollo del artefacto se gestionaron mediante Azure DevOps Boards, utilizando un tablero tipo Taskboard para visualizar el flujo de trabajo y garantizar trazabilidad entre actividades, responsables y evidencias de avance. Esta estrategia permitió organizar el conjunto de tareas del proyecto en un sprint único extendido, centralizando el control del progreso mediante estados y asignaciones, y registrando evidencias directamente en cada ítem de trabajo.

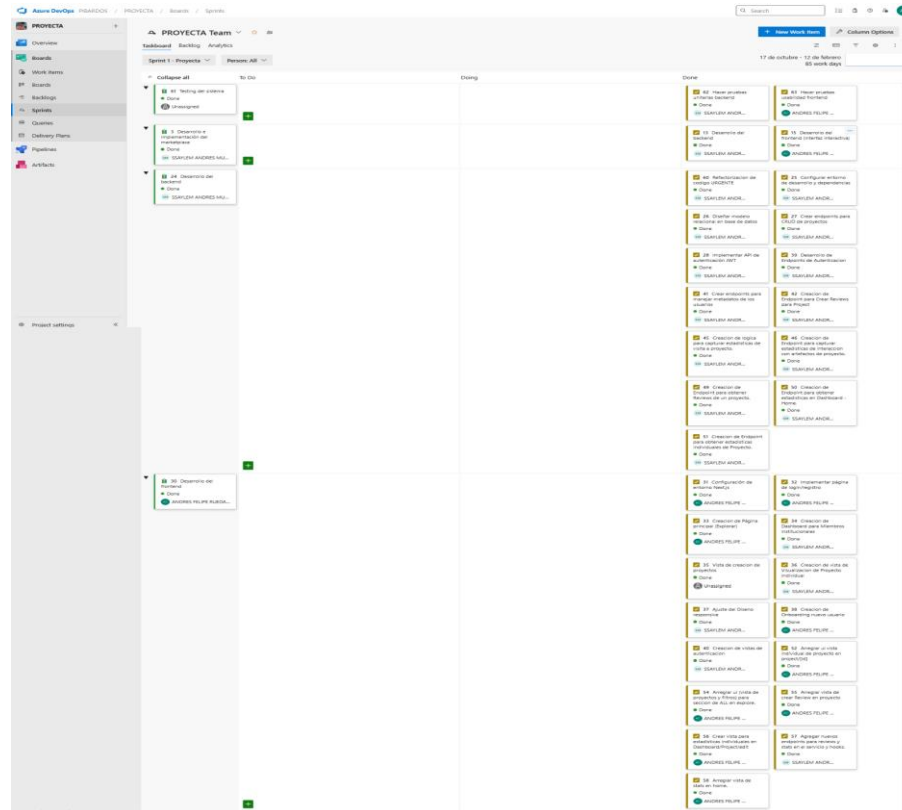


Imagen 26: Planificación y seguimiento del desarrollo

Fuente: UX/UI del artefacto (Elaboración Propia).

4.4.4.6.1. TABLERO DE TRABAJO Y ESTADOS

El tablero se estructuró en columnas de estado (por ejemplo, To Do, Doing y Done) para representar el ciclo de vida de cada actividad desde su registro hasta su finalización. El uso de estados facilitó identificar bloqueos, actividades en curso y tareas terminadas, permitiendo un seguimiento visual y continuo del avance del sprint.

En el tablero se organizaron actividades por componentes del sistema (p. ej., frontend, backend, pruebas y documentación), lo cual favoreció la coordinación del equipo y la ejecución paralela de tareas según dependencias técnicas. Adicionalmente, el tablero permitió validar el cierre de trabajo al observar el desplazamiento progresivo de ítems hacia “Done” conforme se completaban funcionalidades del artefacto.

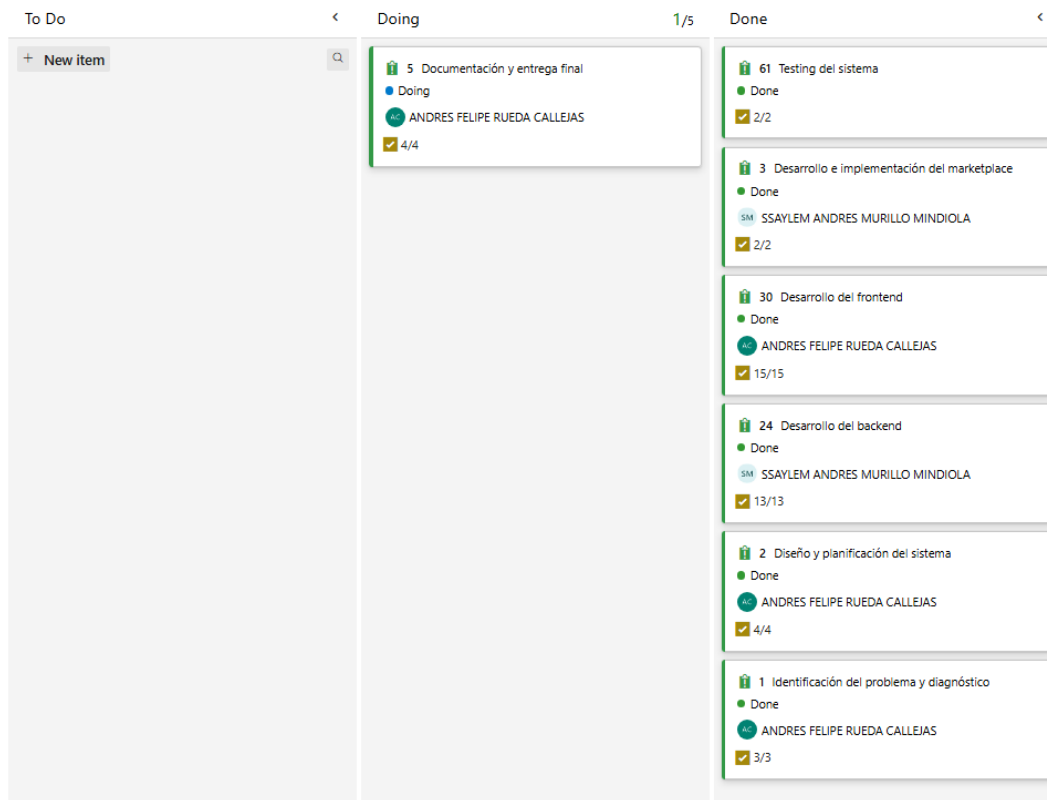


Imagen 27: Tablero de trabajo y estados

Fuente: UX/UI del artefacto (Elaboración Propia).

4.4.4.6.2. ASIGNACIÓN DE TAREAS

Cada ítem del sprint fue asignado a un responsable, manteniendo visibilidad sobre la distribución de trabajo dentro del equipo y facilitando la rendición de avances por módulo o actividad. Esta asignación se complementó con el desglose de tareas en subtareas/actividades específicas (cuando aplicó), permitiendo controlar el cumplimiento progresivo dentro del sprint.

Como mecanismo de control interno, el equipo utilizó la asignación y el estado del ítem para reflejar compromiso de ejecución, y registró actualizaciones a medida que avanzaba la implementación (por ejemplo, implementación de endpoints, pantallas, ajustes de configuración, pruebas y correcciones).

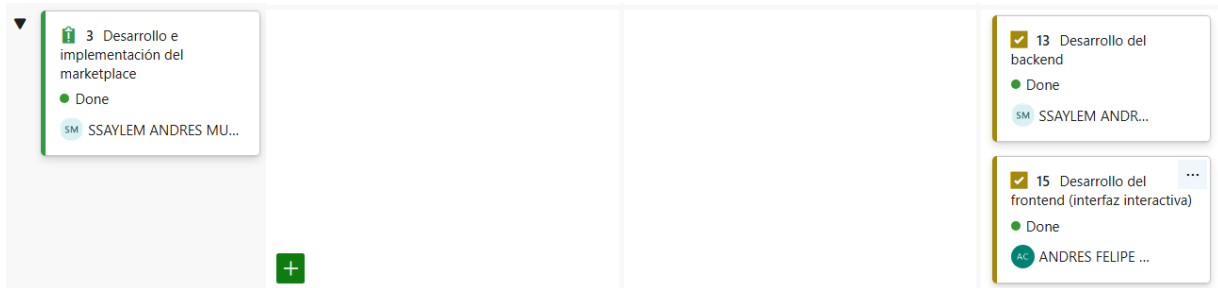


Imagen 28: Evidencia de división del desarrollo entre los miembros para el artefacto.

Fuente: Screenshot tomada directamente de AzureDevops.

4.4.4.6.3. EVIDENCIAS DEL BACKLOG Y SPRINTS

Para asegurar trazabilidad entre planificación y construcción, se incorporaron evidencias dentro de los ítems de Azure DevOps, tales como capturas de pantalla, avances funcionales, resultados de pruebas y comentarios asociados a la tarea. De esta forma, cada actividad cerrada en “Done” quedó respaldada por un conjunto mínimo de evidencias que permiten verificar qué se realizó, cómo se validó y cuál fue su resultado.

Aunque la ejecución se consolidó en un sprint único, el manejo de ítems y evidencias permitió conservar un registro tipo backlog/sprint que soporta la verificación del incremento final. Este enfoque también facilitó la elaboración posterior de secciones del informe relacionadas con evidencias del sistema, pruebas y estado de entrega, al contar con un repositorio centralizado de seguimiento.

SM SSAYLEM ANDRES MURILLO MINDIOLA commented 27 de ene (edited)

Se creo un modelo de User que ahora el AuthService maneja logica de creacion al momento de Registro o inicio de sesion con Google. Y la demas logica en la aplicacion respeta el perfil del usuario y perfil institucional (1 ✓)

```
# app/db/models/user.py
"""User database models."""

from typing import List, Optional
from pydantic import BaseModel
from datetime import datetime

class User(BaseModel):
    """User database model."""
    id: str
    auth_id: str
    email: str
    full_name: Optional[str] = None
    birth_date: Optional[datetime] = None
    avatar_url: Optional[str] = None
    preferences: Optional[List[str]] = None
    onboarding_completed: bool = False
    created_at: Optional[datetime] = None
    last_update: Optional[datetime] = None

    class Config:
        from_attributes = True
        populate_by_name = True
```

Imagen 29: Evidencia de construcción de artefacto 01.

Fuente: Screenshot tomada directamente de AzureDevops.

SM SSAYLEM ANDRES MURILLO MINDIOLA commented 31 de ene

Se revisaron en utils los metodos auxiliares, se cambio logica innecesaria, se agrego y cambio la logica de los OTPs, y se agrego RUFF de python para linter. (7 ✓)

```
1 [tool.ruff]
2 line-length = 100
3 target-version = "py311"
4
5 [tool.ruff.lint]
6 select = ["E", "F", "W", "I"] # E: errores, F: Pyflakes, W: warnings, I: isort
7 ignore = ["F405", "F403", "E501", "W293"]
8
9 [tool.ruff.format]
10 quote-style = "double"
11 indent-style = "space"
12
```

Imagen 30: Evidencia de construcción de artefacto 02.

Fuente: Screenshot tomada directamente de AzureDevops.

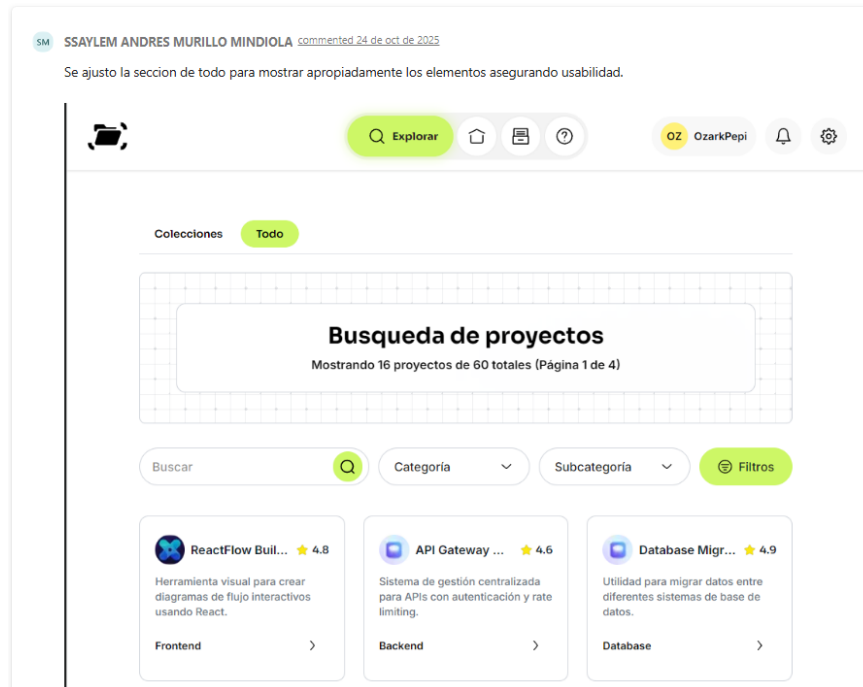


Imagen 31: Evidencia de construcción de artefacto 03.

Fuente: Screenshot tomada directamente de AzureDevops.

4.4.4.7. EVIDENCIAS DEL ARTEFACTO EN FUNCIONAMIENTO

En esta sección se presentan evidencias del artefacto implementado, con el fin de demostrar su funcionamiento a nivel de interfaz y flujos principales por actor, y respaldar que el incremento final corresponde a un sistema operativo (MVP) alineado con el alcance desarrollado. La evidencia se organiza por flujos de navegación y funcionalidades núcleo, priorizando capturas que muestren el resultado final en ejecución (no solo prototipos).

4.4.4.7.1. CAPTURAS POR MÓDULO

A continuación se presentan capturas organizadas por módulos funcionales. Esta organización permite validar de manera directa el cumplimiento del alcance implementado (funcionalidades núcleo) y su coherencia con la arquitectura y los flujos UX/UI descritos en secciones previas.

4.4.4.7.1.1. INICIO / CATÁLOGO (MARKETPLACE)

Este módulo evidencia la interfaz principal de exploración de proyectos (catálogo/colecciones), punto de entrada para visitantes y usuarios autenticados.



Explora nuestras colecciones destacadas

Conjuntos de proyectos organizados por temática, tecnología y
tendencias académicas.

Imagen 32: Vista inicial 1 del marketplace con listado de proyectos publicados.

Fuente: Screenshot del artefacto desarrollo en ejecución.

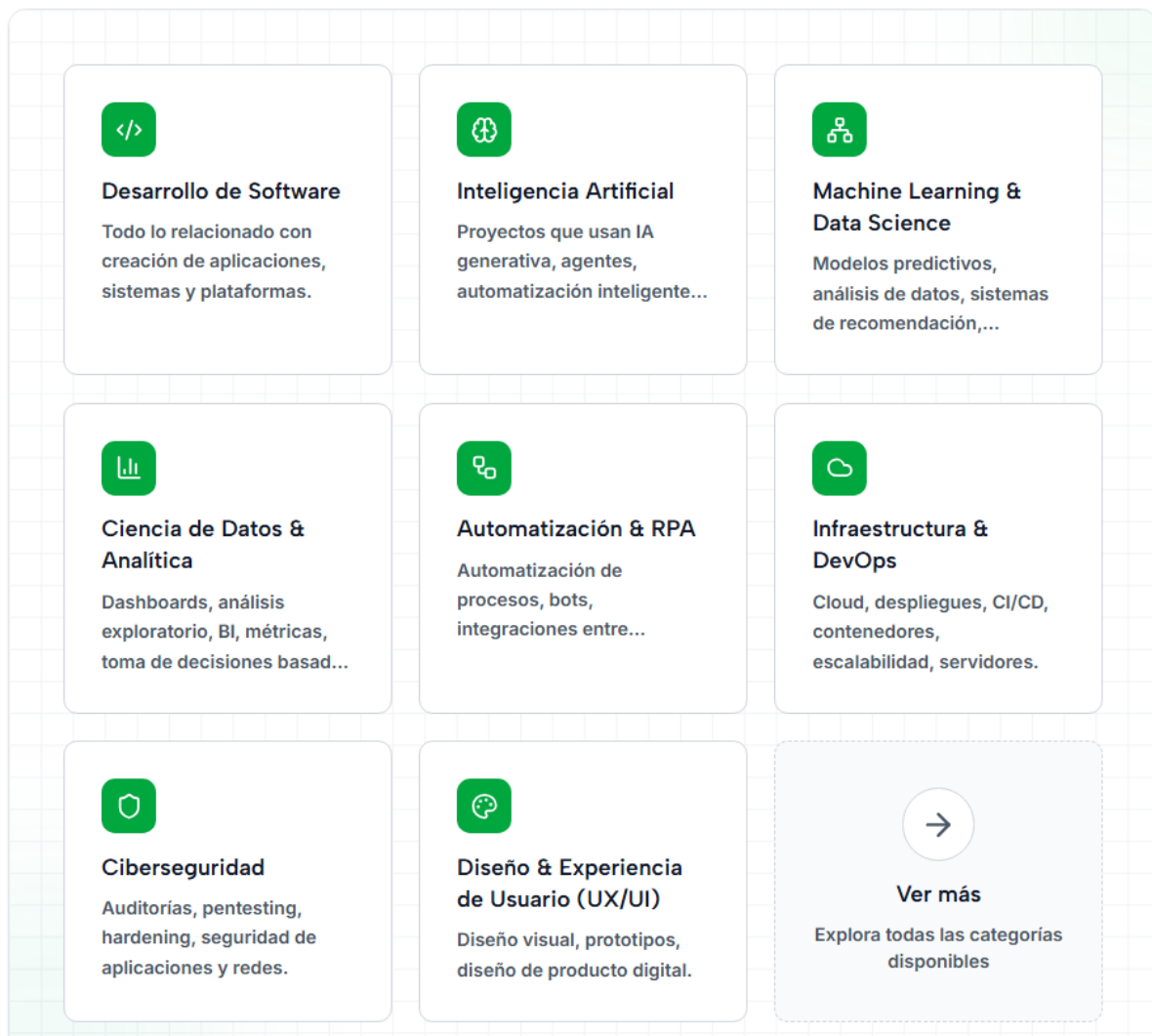


Imagen 33: Vista inicial 2 del marketplace con listado de proyectos publicados.

Fuente: Screenshot del artefacto desarrollo en ejecución..

Lo más popular este mes

Descubre los proyectos mejor valorados por la comunidad.



Imagen 34: Vista inicial 3 del marketplace con listado de proyectos publicados.

Fuente: Screenshot del artefacto desarrollo en ejecución.

Destacado en desarrollo de software

Proyectos académicos enfocados en construcción de software.

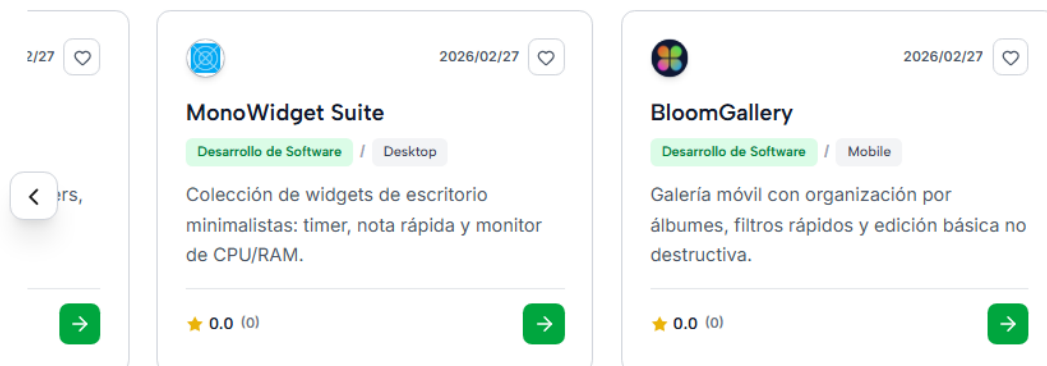


Imagen 35: Vista inicial 4 del marketplace con listado de proyectos publicados.

Fuente: Screenshot del artefacto desarrollo en ejecución.

¿Buscas nuevos proyectos?

Explora los proyectos agregados recientemente al marketplace.

	TaskMind Aplicación de gestión de tareas con análisis inteligente.	
	BooklyPro Plataforma SaaS para gestión de reservas online.	
	StyleAdmin Sistema de gestión e inventario para tiendas de ropa.	
	UniTrade Marketplace universitario para compra y venta de productos y servicios entre estudiantes.	
	BuildFactory API API para orquestar builds: colas, workers, artifacts y webhooks de estado.	

Ver más

Imagen 36: Vista inicial 5 del marketplace con listado de proyectos publicados.

Fuente: Screenshot del artefacto desarrollo en ejecución.

Búsqueda de proyectos

Mostrando 14 de 14 proyectos



 Buscar proyectos por nombre o descripción...

 Filtros

Categoría

Todas las categorías

Subcategoría

Todas las subcategorías



2026/02/27 

TaskMind

Desarrollo de Software / Web

Aplicación de gestión de tareas con análisis inteligente.

★ 0.0 (0) 



2026/02/27 

BooklyPro

Desarrollo de Software / Web

Plataforma SaaS para gestión de reservas online.

★ 0.0 (0) 



2026/02/27 

StyleAdmin

Desarrollo de Software / Web

Sistema de gestión e inventario para tiendas de ropa.

★ 0.0 (0) 



2026/02/27 

UniTrade

Desarrollo de Software / Web

Marketplace universitario para compra y venta de productos y servicios entre estudiantes.

★ 0.0 (0) 



2026/02/27 

BuildFactory API

Desarrollo de Software / API

API para orquestar builds: colas, workers, artifacts y webhooks de estado.

★ 0.0 (0) 



2026/02/27 

MonoWidget Suite

Desarrollo de Software / Desktop

Colección de widgets de escritorio minimalistas: timer, nota rápida y monitor de CPU/RAM.

★ 0.0 (0) 

Imagen 37: Exploración de todo el catálogo.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.2. BÚSQUEDA Y FILTRADO

Este módulo evidencia la capacidad de localizar proyectos mediante filtros o criterios de búsqueda, validando acceso y recuperación de información.

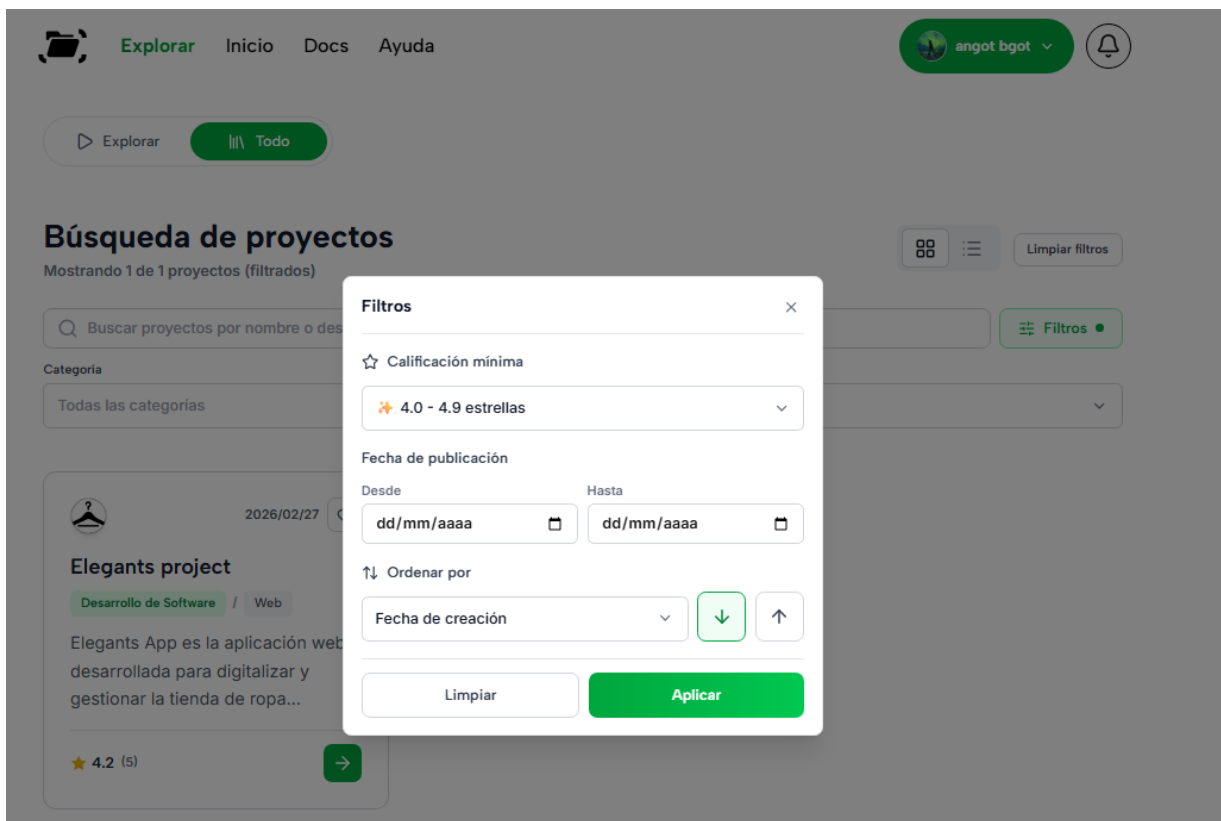


Imagen 38: Aplicación de filtros de búsqueda para localizar proyectos.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.3. DETALLE DE PROYECTO (VISTA INDIVIDUAL)

Este módulo evidencia la visualización completa de un proyecto (metadatos, descripción, artefactos y opciones de interacción).

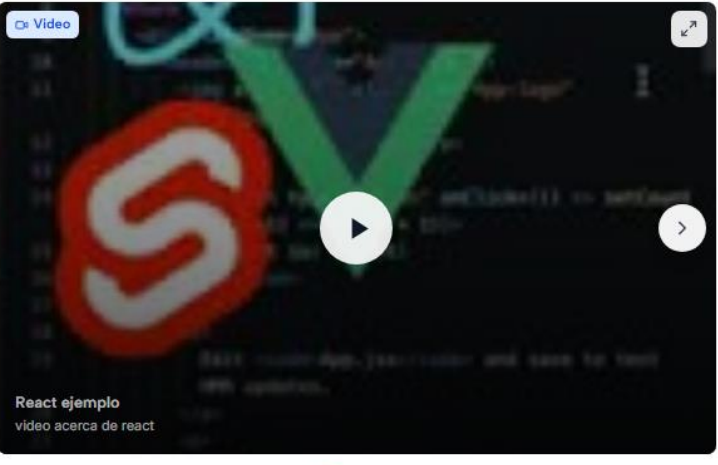
Marketplace > Elegants project

 **Elegants project** 
Por **angot bgot**

Tags: software web ropa administracion

Elegants App es la aplicación web desarrollada para digitalizar y gestionar la tienda de ropa Elegants.

GALERÍA DEL PROYECTO


React ejemplo
video acerca de react

Información principal:

- Creado: Hace 18 horas
- Actualizado: Hace 18 horas
- Categoría: Desarrollo de Software
- Subcategoría: Web
- Rating: (5) 4.2 / 5 
- Licencia:  MIT License

Imagen 39: Visualización del detalle del proyecto y su información principal 1.

Fuente: Screenshot del artefacto desarrollo en ejecución.

☰ Especificaciones

📦 Artefactos

🏠 Arquitectura

Tipo de aplicación

Single Page Application

Enfoque de estilos

Tailwind

Framework

React

Gestión de estado

No especificado

🎨 Diseño Responsivo

Diseño responsivo

Sí

Soporte RTL

No

Breakpoints

mobile, tablet, desktop

🔍 SEO y Metadata

Optimizado para SEO

Sí

Sitemap habilitado

Sí

Meta tags requeridos

Sí

⚡ Rendimiento

Imagen 40: Visualización del detalle del proyecto y su información principal 2.

Fuente: Screenshot del artefacto desarrollo en ejecución.

Ratings y Reviews

4.2
de 5



5 ratings en total

Todas

Recientes

Positivas

Críticas

 Calificar proyecto



Google Business

Hace 19 horas

★ 5 

Un proyecto bastante importante y destacado, nos interesaria contactarte para una vacante!



Sandrith Maria

Hace 19 horas

★ 4 

Este proyecto puede servir para un proyecto de grado. Escribeme:
sandrithjimenezguiterrez@gmail.com



Juan Perez

Hace 19 horas

★ 5 

10/10 ese icono companero. Excelente proyecto.



Maicol David

Hace 19 horas

★ 3 

Puedes mejorar si cambias el icono. Un saludo!

Ver todas (5)

Imagen 41: Visualización del detalle del proyecto y su información principal 3.

Fuente: Screenshot del artefacto desarrollo en ejecución.

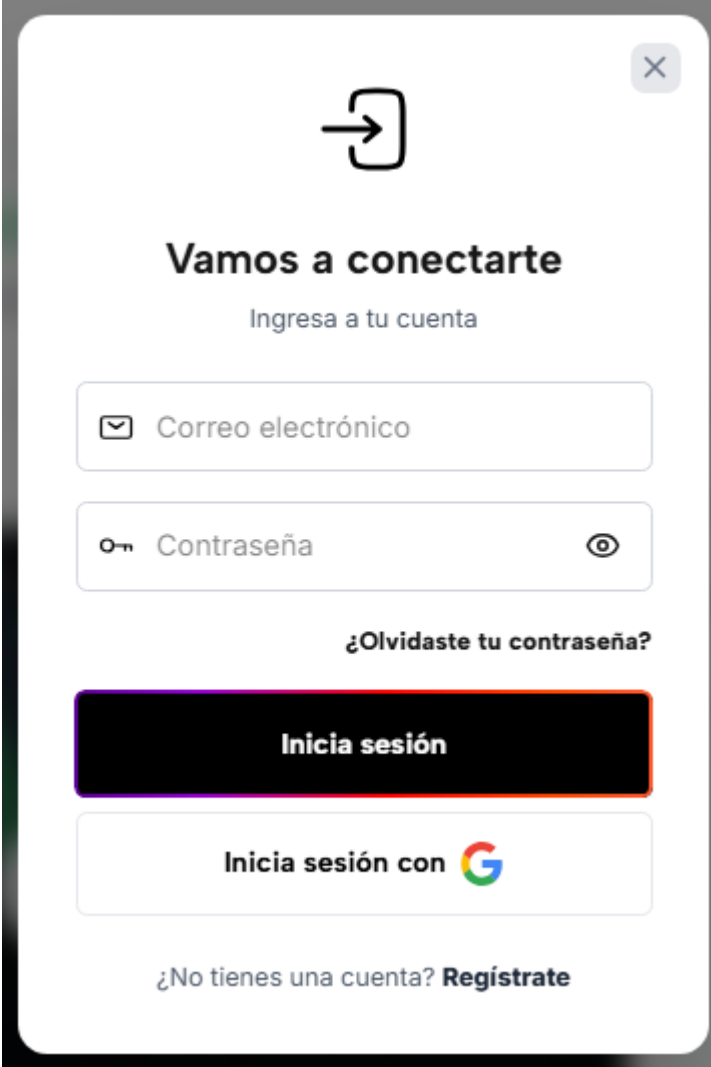


Imagen 42: Acceso a artefactos asociados y recursos del proyecto.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.4. AUTENTICACIÓN Y SESIÓN

Este módulo evidencia el acceso mediante autenticación, requisito para funciones como publicación y feedback.



A login modal window with a close button (X) in the top right corner. At the top center is a large icon of a right-pointing arrow inside a rounded square. Below this icon is the heading "Vamos a conectarte" in bold, followed by the subtitle "Ingresa a tu cuenta". There are two input fields: the first is labeled "Correo electrónico" with an envelope icon on the left; the second is labeled "Contraseña" with a key icon on the left and an eye icon on the right to toggle visibility. Below the password field is a link "¿Olvidaste tu contraseña?". There are two buttons: a primary black button with white text "Inicia sesión" and a secondary white button with black text "Inicia sesión con" followed by the Google logo. At the bottom is a link "¿No tienes una cuenta? Regístrate".

Imagen 43: Interfaz de autenticación para acceso al sistema 1.

Fuente: Screenshot del artefacto desarrollo en ejecución.

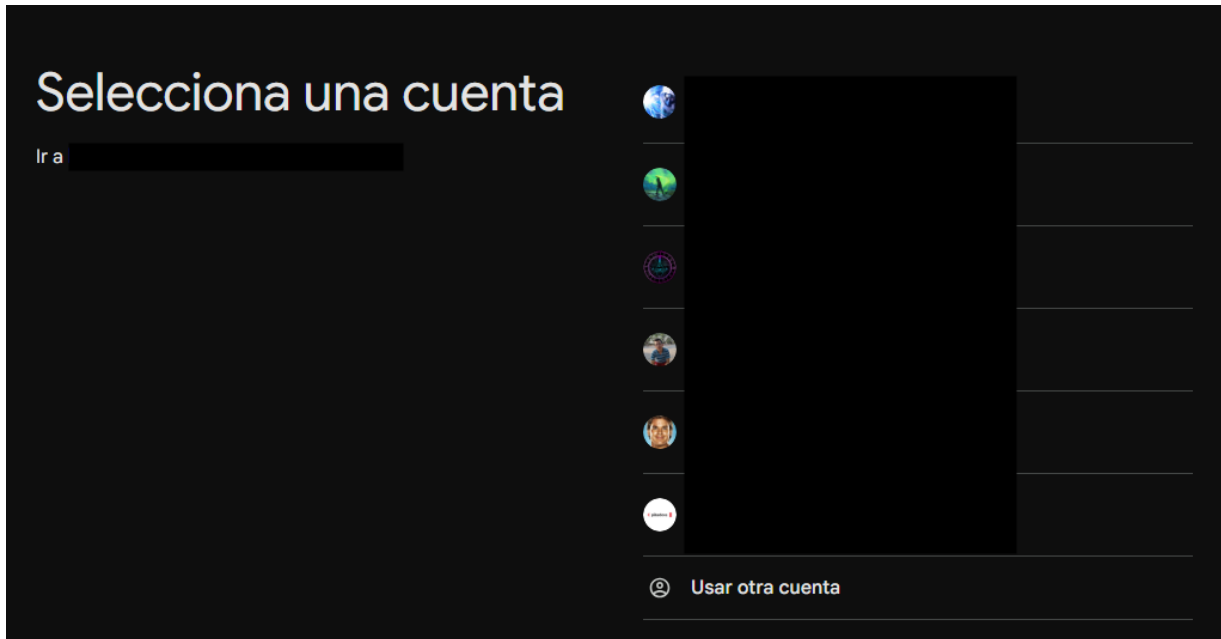


Imagen 44: Interfaz de autenticación para acceso al sistema 2.

Fuente: Screenshot del artefacto desarrollo en ejecución.



Imagen 45: Interfaz de autenticación para acceso al sistema 3.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.5. REGISTRO / CREACIÓN DE CUENTA (SI APLICA)

Este módulo evidencia el proceso de registro cuando el sistema lo permite, mostrando validaciones y resultado.



Paso 1 de 4

Cuéntanos sobre ti

Ayúdanos a conocerte mejor para personalizar tu experiencia



Opcional. Máximo 5MB. Haz clic en el ícono para subir una foto.

Nombre completo *

Solo letras, mínimo 3 caracteres

Fecha de nacimiento *



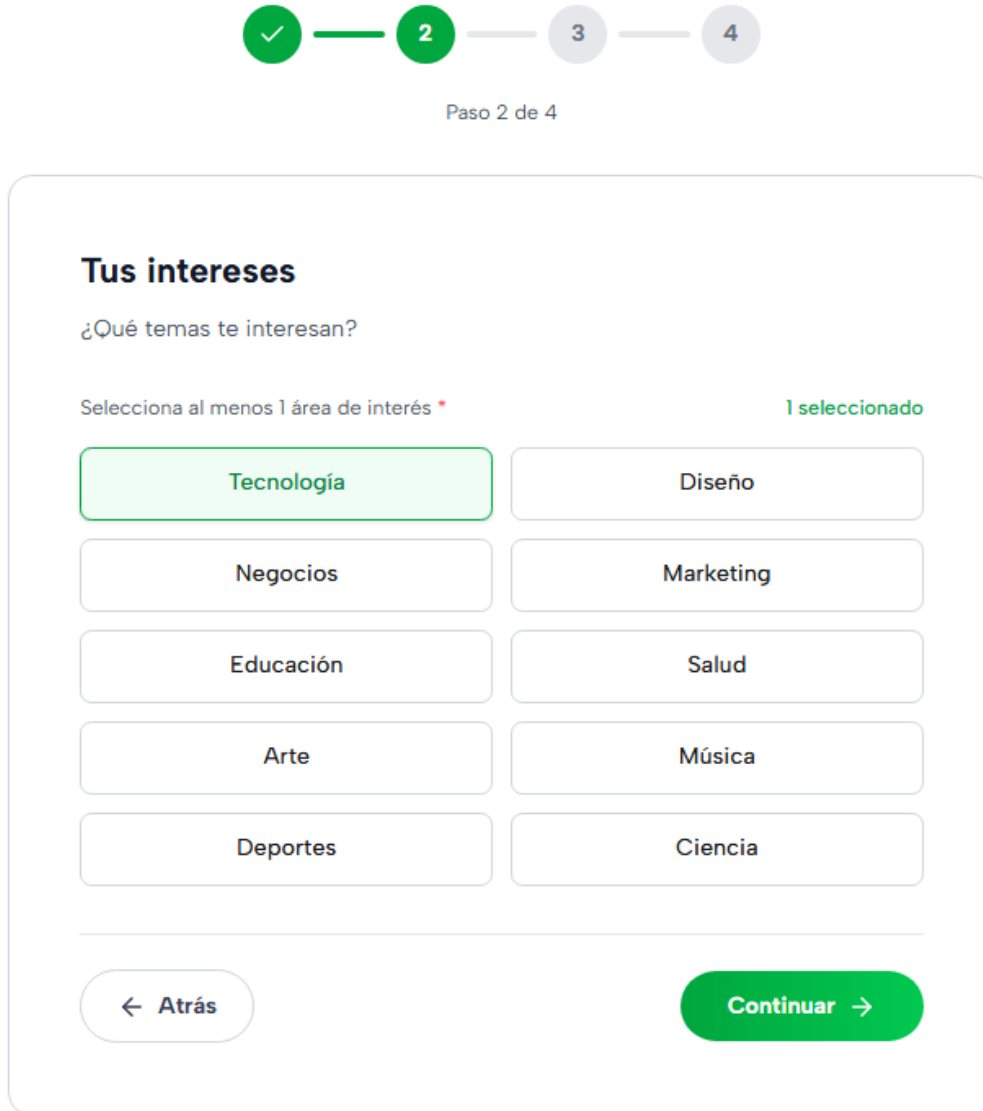
Tu correo electrónico

← Atrás

Continuar →

Imagen 46: Formulario de registro de usuario 1.

Fuente: Screenshot del artefacto desarrollo en ejecución.



✓ — 2 — 3 — 4

Paso 2 de 4

Tus intereses

¿Qué temas te interesan?

Selecciona al menos 1 área de interés * 1 seleccionado

Tecnología	Diseño
Negocios	Marketing
Educación	Salud
Arte	Música
Deportes	Ciencia

← Atrás Continuar →

Imagen 47: Formulario de registro de usuario 2.

Fuente: Screenshot del artefacto desarrollo en ejecución.



Paso 3 de 4

Tu perfil de LinkedIn

Agrega tu perfil profesional (puedes saltar este paso)

Perfil de LinkedIn (opcional)

Puedes dejar este campo vacío si prefieres omitirlo.

← Atrás

Continuar →

Imagen 48: Formulario de registro de usuario 3.

Fuente: Screenshot del artefacto desarrollo en ejecución.



Paso 4 de 4

Revisión final

Revisa tu información antes de continuar



andres felipe

ozarkdev7@gmail.com

Fecha de nacimiento

5 de febrero de 2004

Perfil de LinkedIn



No proporcionado

Intereses (1)

Tecnología

Asegúrate de que toda tu información sea correcta antes de continuar.
De igual manera, podrás editarla más tarde desde **Mi Perfil**.

← Atrás

Completar registro ✓

Imagen 49: Formulario de registro de usuario 4.

Fuente: Screenshot del artefacto desarrollo en ejecución.



Imagen 50: Resultado del registro y acceso al entorno autenticado

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.6. PUBLICACIÓN / CREACIÓN DE PROYECTOS (ESTUDIANTE)

Este módulo evidencia la funcionalidad central: publicar proyectos académicos, incluyendo pasos y confirmación final.

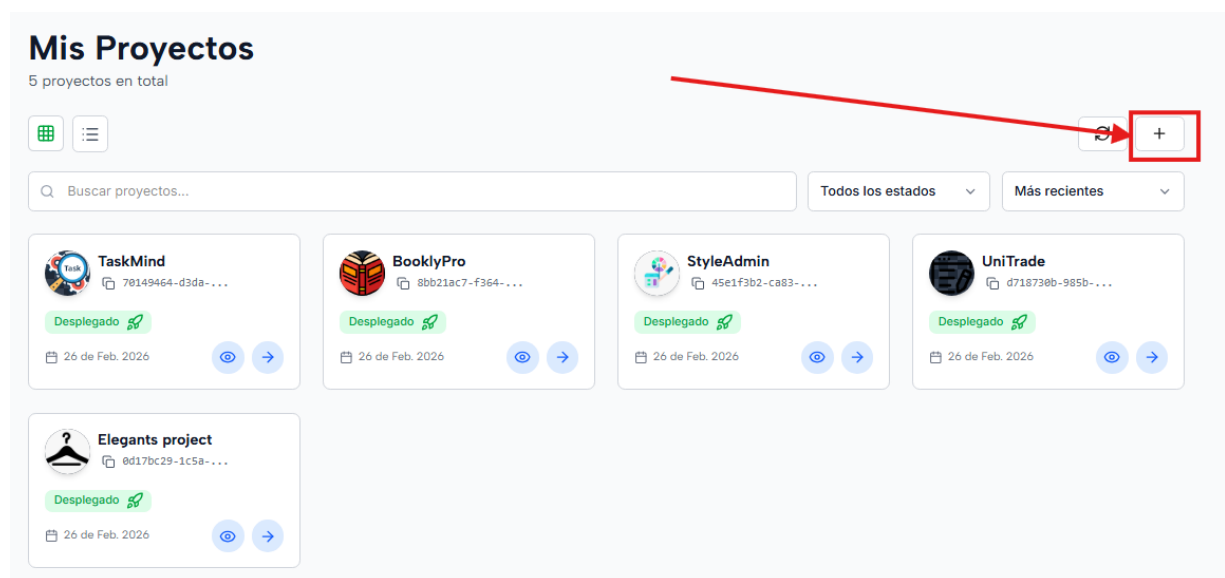
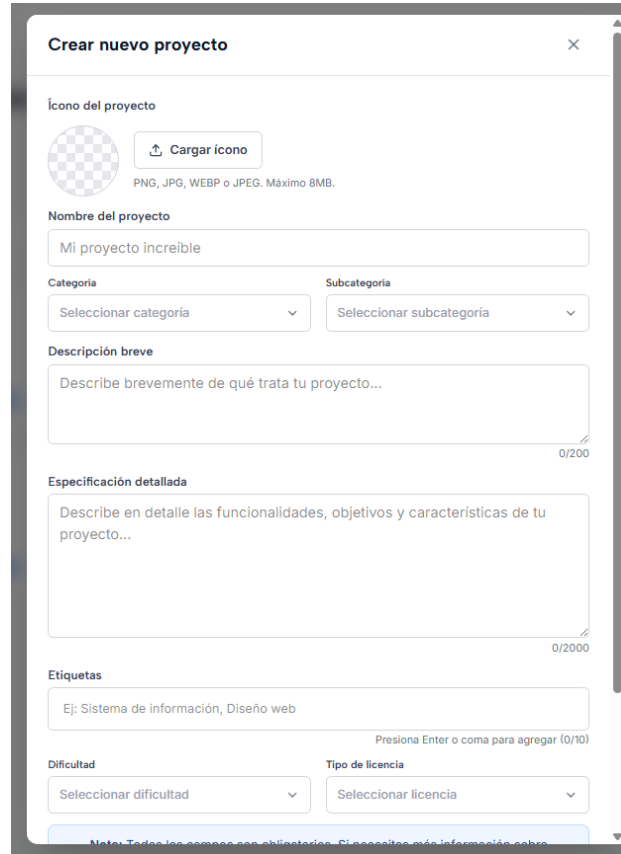


Imagen 51: Inicio de creación de proyecto: captura de datos principales 1

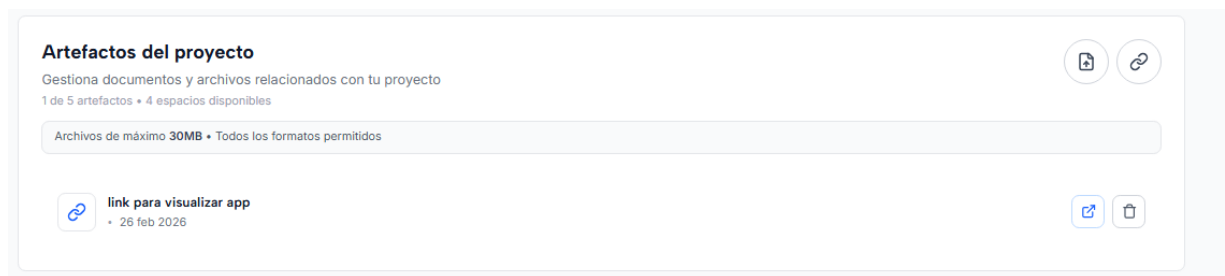
Fuente: Screenshot del artefacto desarrollo en ejecución.



The screenshot shows a web form titled "Crear nuevo proyecto". It includes a section for "Ícono del proyecto" with a placeholder image and a "Cargar icono" button. Below this is a text input for "Nombre del proyecto" containing "Mi proyecto increíble". There are two dropdown menus for "Categoría" and "Subcategoría", both currently showing "Seleccionar categoría" and "Seleccionar subcategoría" respectively. A "Descripción breve" text area contains the placeholder "Describe brevemente de qué trata tu proyecto..." with a character count of "0/200". Below that is a "Especificación detallada" text area with the placeholder "Describe en detalle las funcionalidades, objetivos y características de tu proyecto..." and a character count of "0/2000". An "Etiquetas" section has a text input with the example "Ej: Sistema de información, Diseño web" and a note "Presiona Enter o coma para agregar (0/10)". At the bottom, there are two more dropdowns for "Dificultad" and "Tipo de licencia", both showing "Seleccionar dificultad" and "Seleccionar licencia". A blue note at the very bottom states: "Nota: Todos los campos son obligatorios. Clicéanos más información sobre".

Imagen 52: Inicio de creación de proyecto: captura de datos principales 2

Fuente: Screenshot del artefacto desarrollo en ejecución.



The screenshot shows a section titled "Artefactos del proyecto". It includes a subtitle "Gestiona documentos y archivos relacionados con tu proyecto" and a status "1 de 5 artefactos • 4 espacios disponibles". Below this is a light gray bar with the text "Archivos de máximo 30MB • Todos los formatos permitidos". At the bottom, there is a link icon followed by the text "link para visualizar app" and a date "• 26 feb 2026". On the right side of the section, there are two circular icons: one with a document icon and another with a link icon. At the bottom right, there are two more icons: a blue square with a white link icon and a gray trash can icon.

Imagen 53: Carga y asociación de artefactos al proyecto.

Fuente: Screenshot del artefacto desarrollo en ejecución.

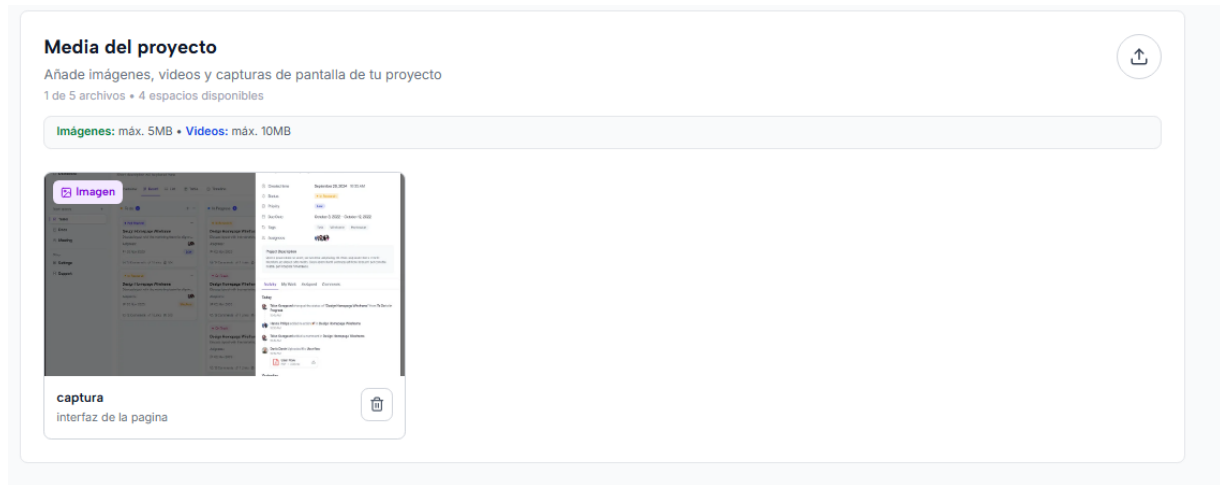


Imagen 54: Carga y asociación de medias al proyecto.

Fuente: Screenshot del artefacto desarrollo en ejecución.

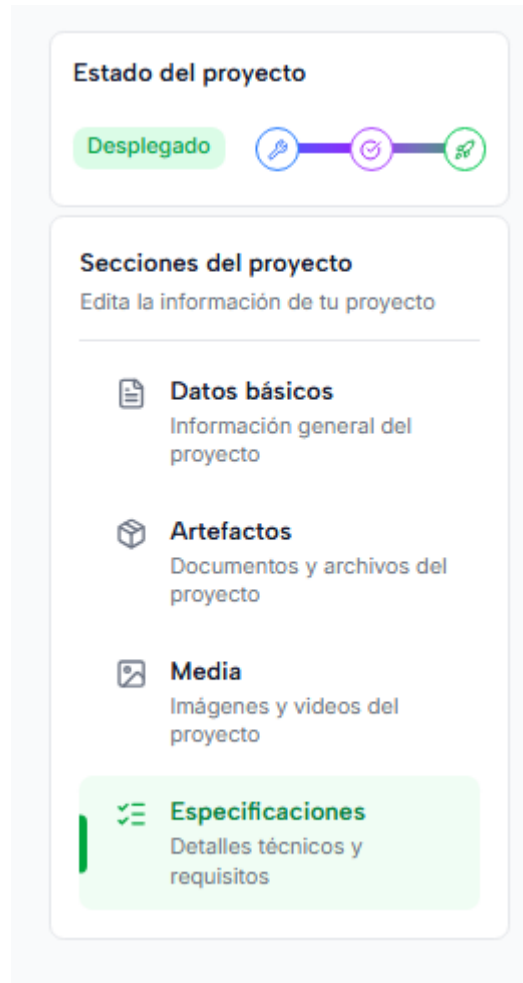


Imagen 55: Confirmación de publicación del proyecto.

Fuente: Screenshot del artefacto desarrollo en ejecución.

Búsqueda de proyectos

Mostrando 14 de 14 proyectos



🔍 Buscar proyectos por nombre o descripción...

🔗 Filtros

Categoría

Todas las categorías

Subcategoría

Todas las subcategorías

2026/02/27

TaskMind

Desarrollo de Software / Web

Aplicación de gestión de tareas con análisis inteligente.

★ 0.0 (0) 

2026/02/27**BooklyPro**

Desarrollo de Software / Web

Plataforma SaaS para gestión de reservas online.

★ 0.0 (0) 2026/02/27**StyleAdmin**

Desarrollo de Software / Web

Sistema de gestión e inventario para tiendas de ropa.

★ 0.0 (0) 2026/02/27**UniTrade**

Desarrollo de Software / Web

Marketplace universitario para compra y venta de productos y servicios entre estudiantes.

★ 0.0 (0) 2026/02/27**BuildFactory API**

Desarrollo de Software / API

API para orquestar builds: colas, workers, artifacts y webhooks de estado.

★ 0.0 (0) 2026/02/27**MonoWidget Suite**

Desarrollo de Software / Desktop

Colección de widgets de escritorio minimalistas: timer, nota rápida y monitor de CPU/RAM.

★ 0.0 (0) 

Imagen 56: Proyecto publicado y visible para consulta en el marketplace.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.7. GESTIÓN DE ARTEFACTOS DEL PROYECTO

Este módulo evidencia el manejo de artefactos asociados (documentos, archivos, recursos) y su disponibilidad para consulta.

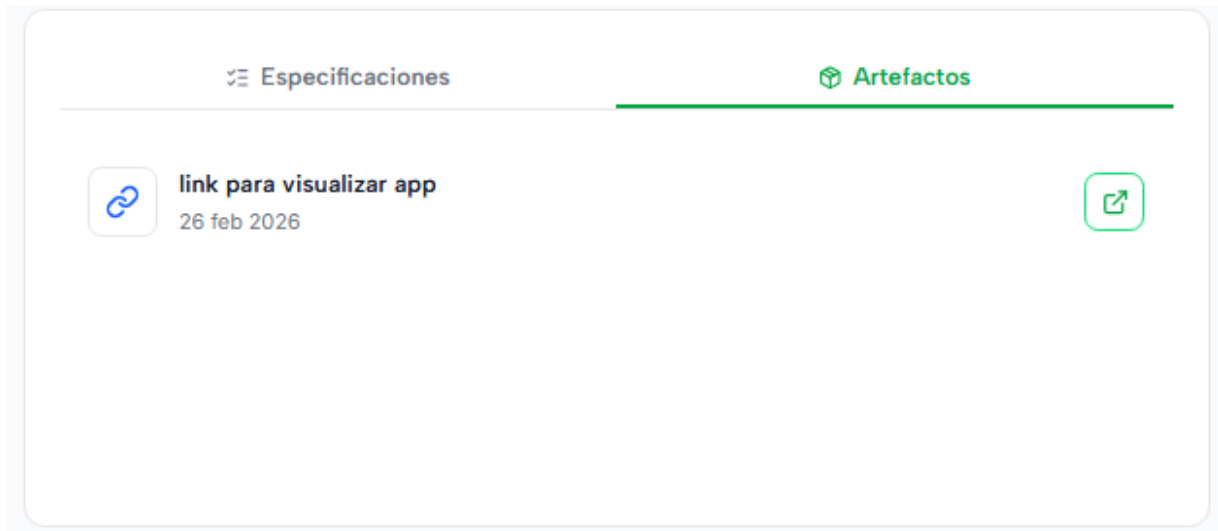


Imagen 57: Artefactos asociados al proyecto disponibles para consulta.

Fuente: Screenshot del artefacto desarrollo en ejecución.

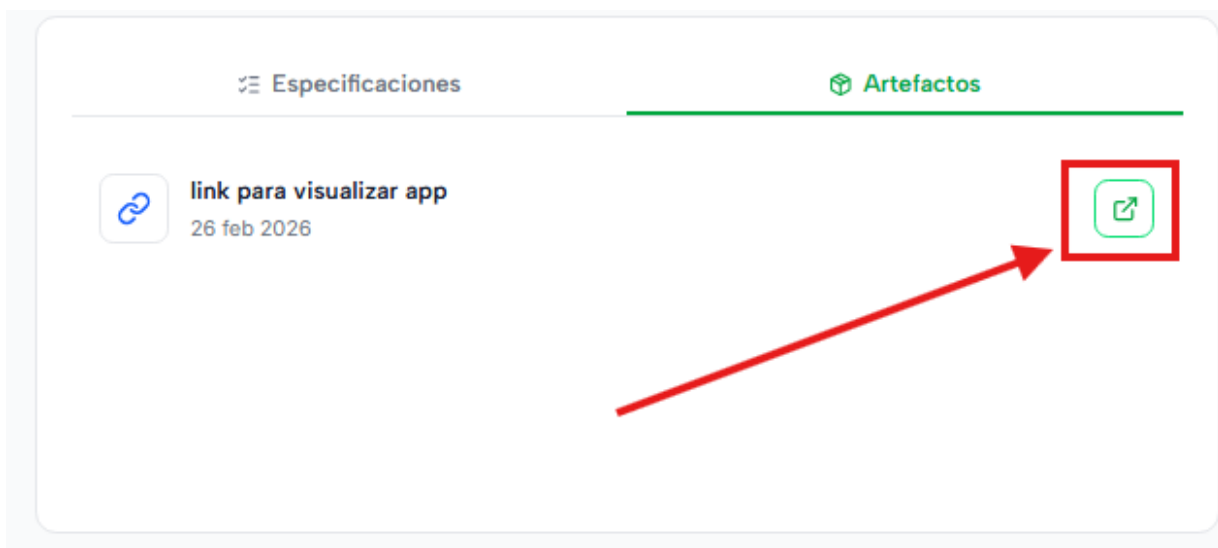


Imagen 58: Acceso a un artefacto específico desde el proyecto 1.

Fuente: Screenshot del artefacto desarrollo en ejecución.

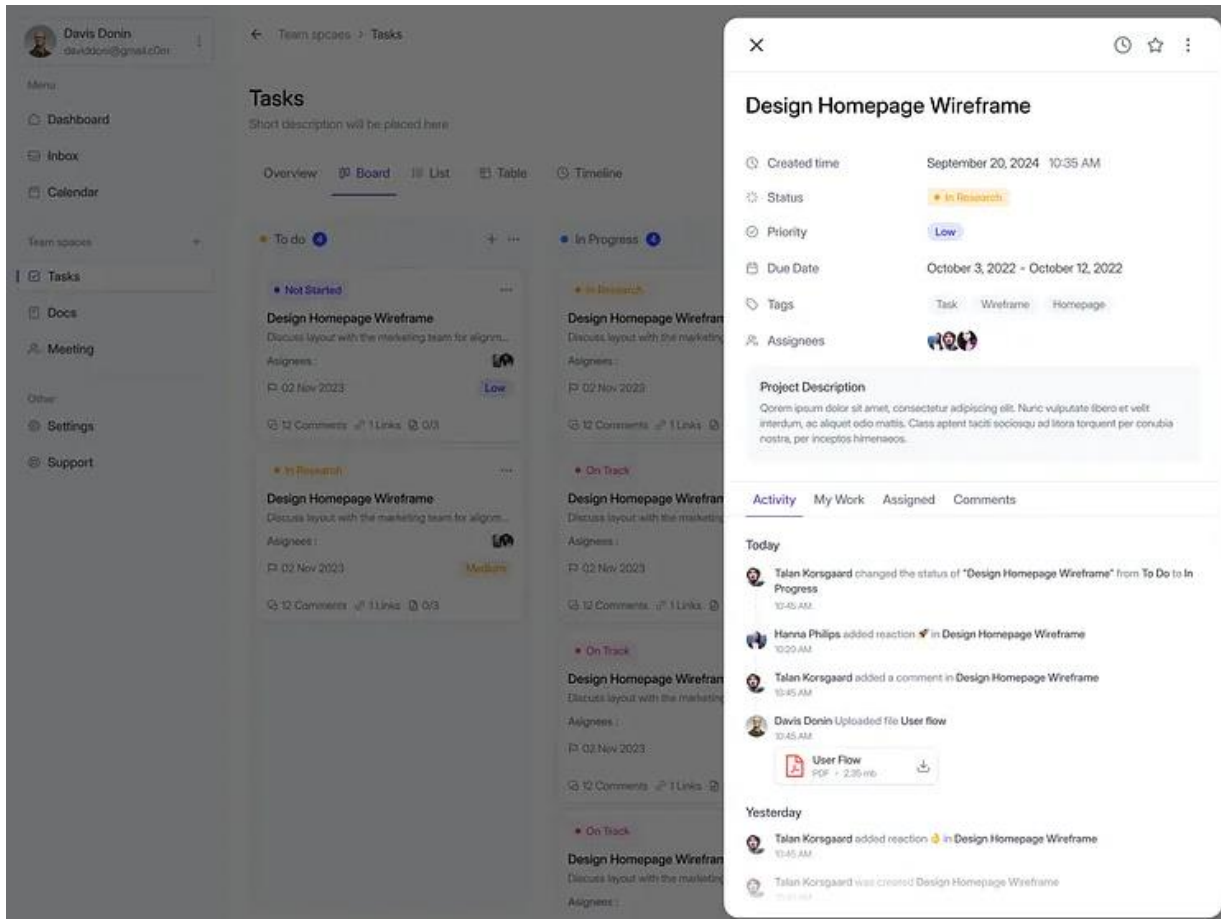


Imagen 59: Acceso a un artefacto específico desde el proyecto 2.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.8. FEEDBACK, COMENTARIOS Y RATING

Este módulo evidencia la interacción comunitaria mediante reviews/comentarios y calificación, reforzando la naturaleza interactiva del repositorio.



Imagen 60: Sección de feedback del proyecto (vista inicial)

Fuente: Screenshot del artefacto desarrollo en ejecución.

Califica Este Proyecto

Ayúdanos a mejorar compartiendo tu experiencia con este proyecto

☆☆☆☆☆

¿Puedes contarnos más?

Añade comentarios (opcional, mínimo 10 caracteres)

0/500

Cancelar Enviar

Imagen 61: Registro de retroalimentación por un usuario autenticado.

Fuente: Screenshot del artefacto desarrollo en ejecución.

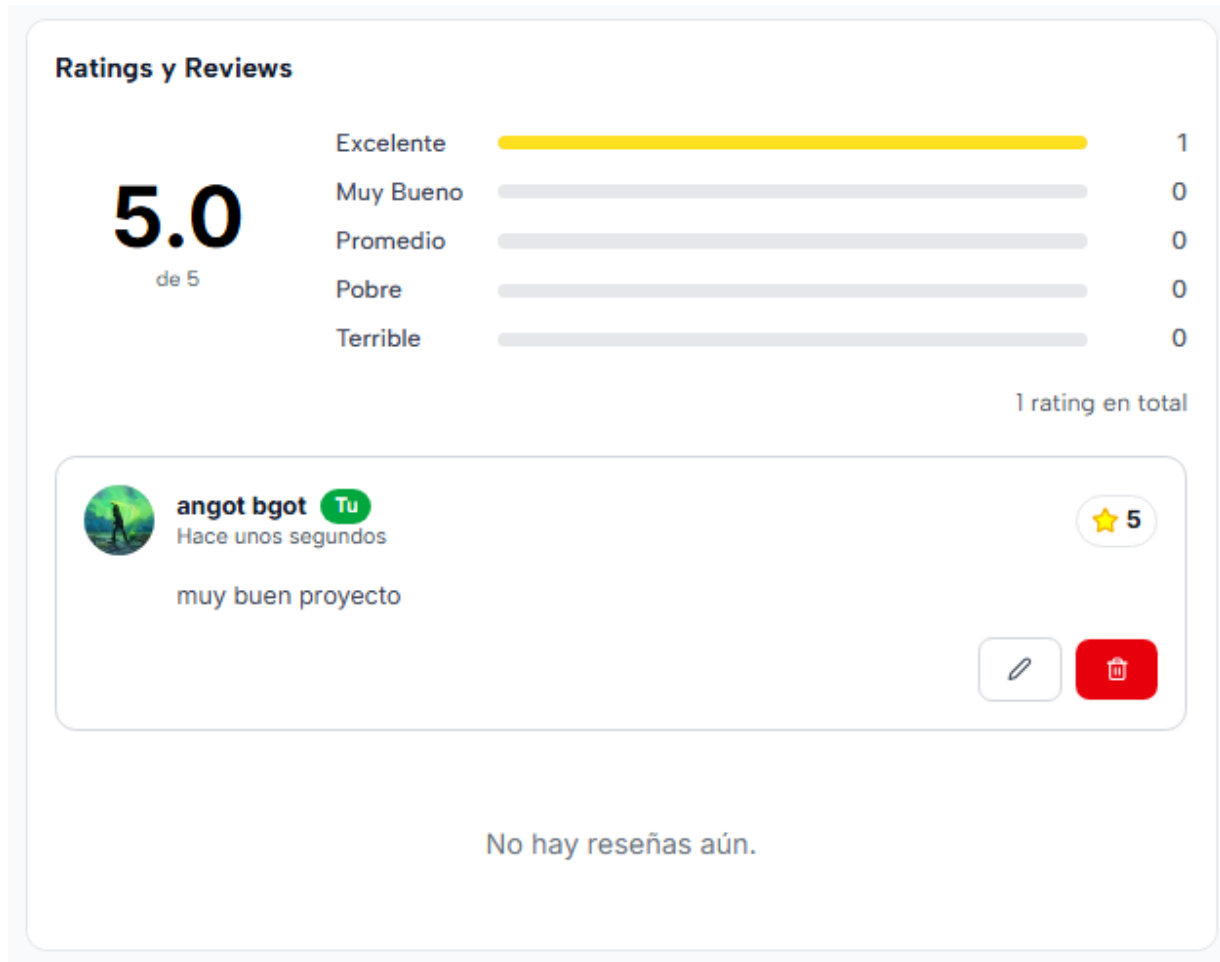


Imagen 62: Evidencia del comentario reflejado en la vista del proyecto.

Fuente: Screenshot del artefacto desarrollo en ejecución.

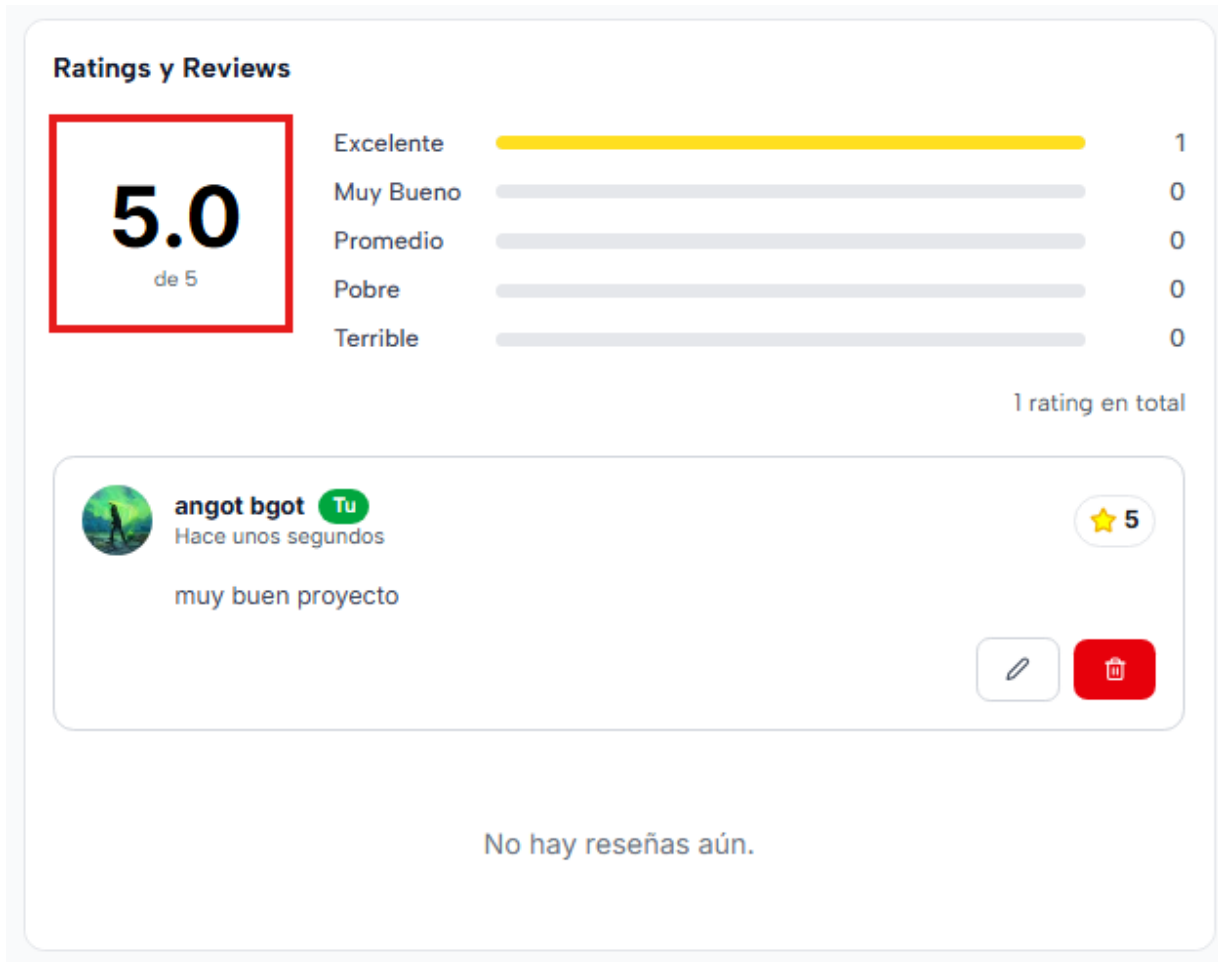


Imagen 63: Evidencia de calificación (rating) aplicada al proyecto.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.9. MÉTRICAS DEL SISTEMA (GENERALES)

Este módulo evidencia paneles o gráficas agregadas del sistema, mostrando actividad/uso y estadísticas globales.

Dashboard

Bienvenido de vuelta, aquí está tu resumen

Mis estadísticas

Total Proyectos

5

Ultimos datos recopilados



Total Vistas

32

Ultimos datos recopilados



Total Likes

0

Ultimos datos recopilados



Interacciones

1

Ultimos datos recopilados



Estado de Proyectos

Distribución por estado



● Deployed

● Deployed
5

Proyectos por Categoría

Distribución por categoría



Desarrollo de Software

Imagen 64: Panel de métricas generales del sistema 1.

Fuente: Screenshot del artefacto desarrollo en ejecución.

Proyectos por Subcategoría

Distribución por subcategoría



Web

Imagen 65: Panel de métricas generales del sistema 2.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.10. ESTADÍSTICAS INDIVIDUALES POR PROYECTO

Este módulo evidencia métricas específicas por proyecto (vistas, interacciones, calificaciones u otras).

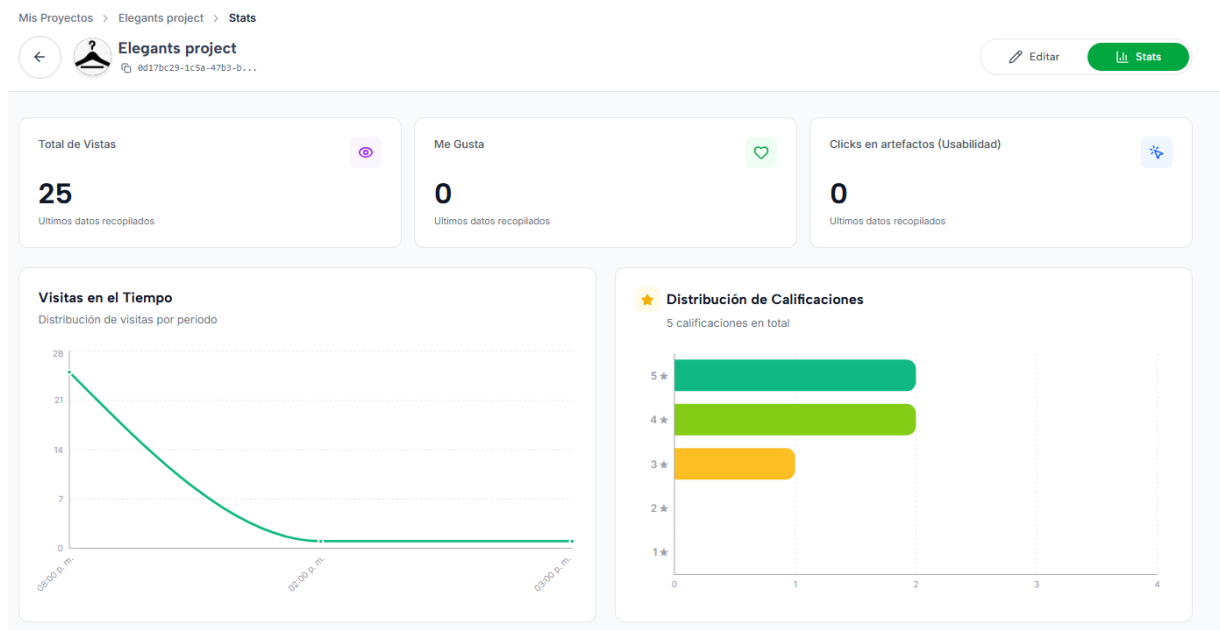


Imagen 66: Estadísticas individuales asociadas a un proyecto específico.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.7.1.11. DOCUMENTACIÓN / AYUDA

Este módulo evidencia que el sistema incorpora una sección de documentación o ayuda para orientar a los usuarios (componente de soporte).

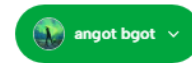


Explorar

Inicio

Docs

Ayuda



Documentación de Proyecta

Aprende a utilizar Proyecta, conectar con otros miembros y acceder a proyectos únicos de la Universidad Popular del Cesar.



Empezar

Qué es Proyecta, flujo general de usuario y conceptos clave de la plataforma.



Uso de la plataforma

Explorar colecciones, ver proyectos, publicar y gestionar tu portafolio.



Encontrar el proyecto ideal

Filtrar con éxito entre cantidades de proyectos y encontrar requerimientos específicos.



Contribuir

Flujo para contribuir al proyecto, buenas prácticas y pull requests.

¿Para quién es esta documentación?

Para estudiantes, docentes y desarrolladores que quieren entender cómo funciona Proyecta, y cómo sacarle el máximo provecho como repositorio institucional de proyectos de software.

Imagen 67: Sección de documentación/ayuda disponible para usuarios 1.

Fuente: Screenshot del artefacto desarrollo en ejecución.



Explorar

Inicio

Docs

Ayuda



En Proyecta estamos para ayudarte

Encuentra respuestas rápidas a tus preguntas o contacta a nuestro equipo de soporte

Q Busca tu pregunta aquí...

Preguntas Frecuentes

Las respuestas a tus dudas más comunes

¿Qué es Proyecta?



¿Cómo me registro en la plataforma?



Imagen 68: Sección de documentación/ayuda disponible para usuarios 2.

Fuente: Screenshot del artefacto desarrollo en ejecución.

4.4.4.8. PRUEBAS TÉCNICAS DEL ARTEFACTO

Las pruebas técnicas del artefacto se realizaron con el propósito de verificar el comportamiento del sistema a nivel de API y componentes críticos, asegurando que las funcionalidades principales operen de forma consistente bajo condiciones normales de uso y ante escenarios de error. Para ello, se definió un enfoque de verificación basado en: (i) qué se probó (funcionalidades/endpoints), (ii) con qué herramienta se probó, (iii) cuáles fueron los criterios de aceptación (resultado esperado), (iv) cuáles fueron los resultados observados y (v) qué evidencia respalda la prueba (capturas y/o reportes).

El proceso de pruebas se organizó en varios niveles: pruebas unitarias para validar lógica interna, pruebas de integración para comprobar el comportamiento de endpoints completos, y pruebas de sistema que incluyen validación manual rápida de solicitudes/respuestas y pruebas de carga. Este esquema se apoya en herramientas ya definidas en el proyecto para validación técnica (Swagger para pruebas manuales rápidas, Pytest para pruebas automatizadas, pytest-cov para cobertura, y Locust para carga), permitiendo trazabilidad entre ejecución y evidencia documental.

4.4.4.8.1. PRUEBAS UNITARIAS

Las pruebas unitarias se implementaron con el objetivo de verificar de forma aislada el comportamiento de componentes internos del backend, asegurando que cada capa de la arquitectura cumpla su responsabilidad sin depender de servicios externos reales. Para lograrlo, la suite de pruebas se organizó por capas dentro de tests/unit, separando casos orientados a endpoints, servicios y repositorios, lo cual facilita el mantenimiento y permite identificar rápidamente el origen de un fallo cuando una prueba no cumple el resultado esperado.

En la capa de repositorios, las pruebas se enfocaron en validar la interacción esperada con el cliente de persistencia (por ejemplo, operaciones tipo `select`, `order`, `eq`, `execute`) mediante el uso de mocks, evitando consultas reales a la base de datos. En la capa de servicios, se validó la lógica de orquestación y manejo de excepciones utilizando dobles de prueba (por ejemplo, `MagicMock/AsyncMock`) para simular repositorios y forzar escenarios de éxito y error

controlado. Finalmente, en la capa de endpoints se verificó el contrato de respuesta (código HTTP y estructura JSON) empleando un cliente HTTP asíncrono contra el app ASGI, pero sobrescribiendo dependencias (dependency overrides) para mantener el aislamiento de la prueba y evitar efectos colaterales.

4.4.4.8.1.1. MÓDULOS CUBIERTOS POR LAS PRUEBAS UNITARIAS

La suite unitaria cubrió principalmente los siguientes módulos/capas:

- **Repositorios:** validación de métodos de consulta/obtención (por ejemplo, obtención de licencias y búsqueda por ID/SPDX) con cliente Supabase simulado mediante mocks.
- **Servicios:** verificación de reglas de negocio y manejo de excepciones (por ejemplo, obtención de categorías, subcategorías, licencias y dificultades; retorno de listas vacías; propagación/transformación de errores).
- **Endpoints:** verificación del contrato HTTP/JSON de rutas de metadatos (por ejemplo, [/metadata/difficulties](#), [/metadata/licenses](#), [/metadata/categories](#)) utilizando cliente ASGI y dependencias mockeadas.

4.4.4.8.1.2. CASOS UNITARIOS EJECUTADOS

A continuación se describen ejemplos representativos de los casos cubiertos (sin exponer implementación):

- Validación de que el repositorio devuelve una lista de entidades correctamente mapeadas cuando la fuente retorna datos completos (caso “happy path”).
- Verificación de comportamiento cuando la fuente retorna datos mínimos (campos opcionales ausentes) y el modelo resultante mantiene **None** en los atributos no provistos.
- Validación de retorno de lista vacía cuando la consulta no tiene resultados (sin lanzar excepciones).
- Obtención por identificador (ID) cuando existe el registro y retorno de **None** cuando no existe (control de “not found”).

- Obtención por criterio alterno (por ejemplo, `spdxid`) y manejo correcto del caso inexistente.
- En servicios, verificación de retorno de lista/objeto cuando el repositorio responde correctamente y verificación de retorno vacío cuando no hay datos.
- En servicios, verificación de que ante errores en repositorios (por ejemplo, `RuntimeError`) se disparan excepciones del dominio definidas por el proyecto (p. ej., `RetrieveAll...Exception`, `...NotFoundException`).
- En endpoints, verificación de `status_code = 200` y campos esperados en el JSON (por ejemplo, `error`, `message`, `data`, `total`) cuando el servicio mock devuelve datos válidos.
- En endpoints, verificación de propiedades de consistencia como: `total` coincide con la longitud de la lista retornada, y el endpoint no exige autenticación si así fue definido.

Para la ejecución local de la suite unitaria se utilizó Pytest, con el comando **pytest**.

```
(venv)
sandr@DESKTOP-3A7AAEQ HINOM64 ~/Documents/code/proyecta/proyecta_back (feature/fix_otp_logic)
$ pytest
tests/unit/test_services/file_service/test_project_basic_info_file_storage_service.py ..... [ 54%]
tests/unit/test_services/file_service/test_project_media_file_storage_service.py ..... [ 56%]
tests/unit/test_services/file_service/test_user_content_file_storage_service.py ..... [ 59%]
tests/unit/test_services/project_services/test_marketplace_service.py ..... [ 61%]
tests/unit/test_services/project_services/test_project_artifact_service.py ..... [ 64%]
tests/unit/test_services/project_services/test_project_media_service.py ..... [ 66%]
tests/unit/test_services/project_services/test_project_metadata_service.py ..... [ 67%]
tests/unit/test_services/project_services/test_project_new_service.py ..... [ 72%]
tests/unit/test_services/project_services/test_project_requirement_service.py ..... [ 75%]
tests/unit/test_services/project_services/test_project_review_service.py ..... [ 78%]
tests/unit/test_services/project_services/test_project_status_orchestrator_service.py ..... [ 80%]
tests/unit/test_services/project_services/test_stats_service.py ..... [ 84%]
tests/unit/test_services/test_admin_service.py ..... [ 87%]
tests/unit/test_services/test_auth_service.py ..... [ 91%]
tests/unit/test_services/test_mail_service.py ..... [ 93%]
tests/unit/test_services/test_otp_service.py ..... [ 93%]
tests/unit/test_services/test_profile_service.py ..... [100%]

===== warnings summary =====
venv\Lib\site-packages\pydantic\_internal\_config.py:291: 31 warnings
  C:\Users\sandr\Documents\code\proyecta\proyecta_back\venv\Lib\site-packages\pydantic\_internal\_config.py:291: PydanticDeprecatedSince20: Support for class-based config is deprecated, use ConfigDict instead. Deprecated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.9/migration/
    warnings.warn(DEPRECATION_MESSAGE, DeprecationWarning)

tests/unit/test_repositories/test_supabase_otp_repository.py::TestDeleteExpiredOtps::test_returns_count_of_deleted_expired_otps
tests/unit/test_repositories/test_supabase_otp_repository.py::TestDeleteExpiredOtps::test_returns_zero_when_no_expired_otps
tests/unit/test_repositories/test_supabase_otp_repository.py::TestDeleteExpiredOtps::test_does_not_call_delete_when_no_expired_otps
tests/unit/test_repositories/test_supabase_otp_repository.py::TestDeleteExpiredOtps::test_returns_zero_when_date_is_none
  C:\Users\sandr\Documents\code\proyecta\proyecta_back\app\db\repositories\supabase\supabase_otp_repository.py:146: DeprecationWarning: datetime.datetime.utcnow() is deprecated and scheduled for removal in a future version. Use timezone-aware objects to represent datetimes in UTC: datetime.datetime.now(datetime.UTC).
    now = datetime.utcnow().isoformat()

-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
===== 1497 passed, 35 warnings in 12.25s =====
```

Imagen 69: Evidencia de ejecución de pruebas unitarias utilizando pytest.

Fuente: Screenshot en consola vs code (ejecución de pytest).

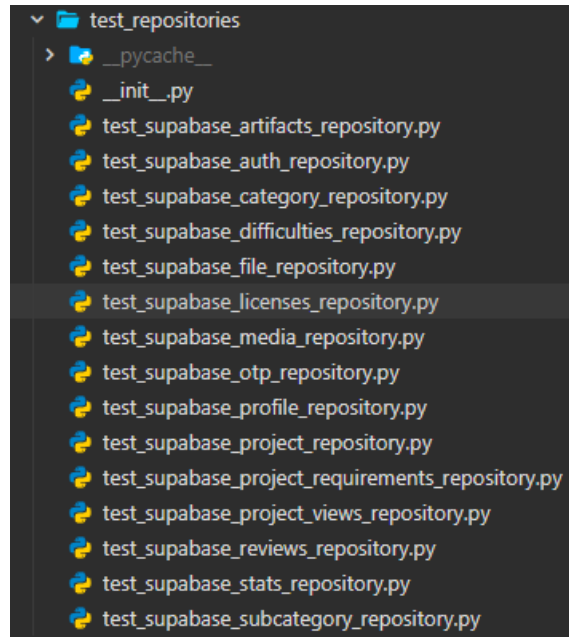


Imagen 70: Evidencia de creación de tests para módulo de repositorios.

Fuente: Screenshot de Treeview en VS Code del proyecto.

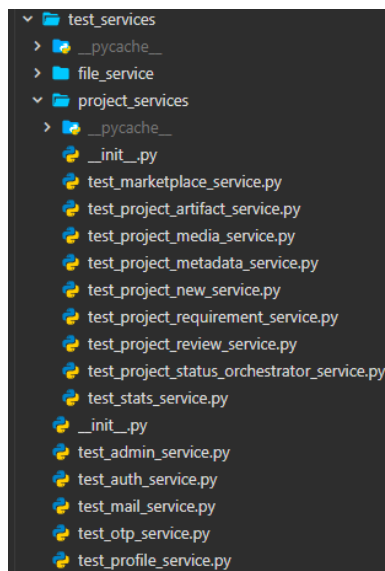


Imagen 71: Evidencia de creación de tests para módulo de servicios.

Fuente: Screenshot de Treeview en VS Code del proyecto.

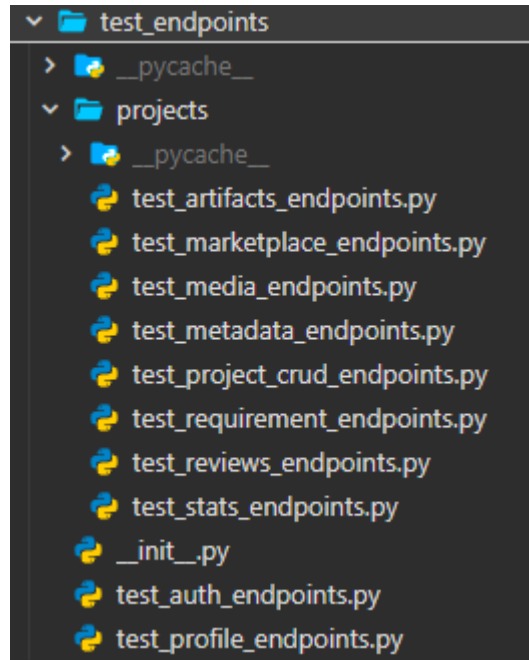


Imagen 72: Evidencia de creación de tests para módulo de endpoints.

Fuente: Screenshot de Treeview en VS Code del proyecto.

4.4.4.8.2. PRUEBAS DE INTEGRACIÓN

Las pruebas de integración se orientaron a validar el comportamiento del API cuando varias capas trabajan juntas (endpoint → servicio → repositorio simulado), verificando el contrato HTTP (códigos de estado) y la estructura del JSON retornado. A diferencia de las unitarias puras, aquí se ejecutan rutas completas del backend usando un cliente HTTP contra la app ASGI, pero manteniendo control del entorno mediante mocks y sobrescritura de dependencias para que los resultados sean repetibles y no dependan de un backend externo.

El objetivo de las pruebas de integración fue comprobar que las rutas del API respondan correctamente bajo escenarios de uso realista, validando la interacción entre capa de presentación (endpoints), capa de aplicación (servicios) y capa de acceso a datos (repositorios) dentro del backend. En esta fase se verificó: (i) códigos de estado esperados, (ii)

estructura mínima del payload JSON (error, message, data, total cuando aplica), (iii) consistencia de campos devueltos por entidad, y (iv) comportamiento ante escenarios de lista vacía, múltiples resultados y parámetros de consulta.

4.4.4.8.2.1. METODOLOGÍA UTILIZADA

En la validación automatizada, los casos se ejecutaron de forma asíncrona con Pytest (pytest.mark.asyncio) utilizando un cliente HTTP asíncrono configurado para comunicarse directamente con la app ASGI. Este enfoque permite validar códigos de estado (200/400/401/404 según aplique), campos mínimos del JSON (error, message, data, total) y consistencia de resultados (por ejemplo, total coincide con la longitud de la lista), manteniendo la prueba repetible y controlada.

Además, para aislar dependencias externas y asegurar determinismo, se emplearon sobreescrituras de dependencias (app.dependency_overrides) inyectando servicios mock que retornan datos predefinidos, evitando acoplar la prueba a una base de datos real.

4.4.4.8.2.2. CRITERIOS DE ACEPTACIÓN

En esta subsección se presentan los criterios de aceptación definidos para las pruebas de integración del API, los cuales sirven como referencia objetiva para determinar si un endpoint cumple con el comportamiento esperado (código de estado, estructura del JSON y consistencia de los datos devueltos). Estos criterios permiten estandarizar la verificación del contrato entre cliente y servidor y facilitan la trazabilidad de resultados al momento de documentar evidencias (capturas/reportes) y correcciones.

Es importante señalar que los criterios listados a continuación representan un subconjunto de los criterios utilizados durante la validación técnica del artefacto. En función del endpoint, el contexto de uso y la criticidad del flujo, pueden existir criterios adicionales asociados a validaciones de negocio, manejo de errores, control de acceso, integridad de datos, tiempos de respuesta y estabilidad operativa; por tanto, la lista se presenta como una guía base y no como un inventario exhaustivo de todas las condiciones evaluadas.

Endpoint	Caso	Criterios de aceptación (debe cumplirse todo)
GET /metadata/categories	"Happy path"	status_code = 200 ; error = False y message corresponde a éxito del recurso (p. ej., "Categorías obtenidas exitosamente") ; data.categories existe y es lista ; cada categoría contiene al menos id, name, description, active ; data.total == len(data.categories) .
GET /metadata/categories	Sin datos	status_code = 200 ; data.total = 0 ; data.categories es lista vacía (o existe y su longitud es 0) .
GET /metadata/categories	Default params	Si no se envía includeSubcategories, se asume False (comportamiento por defecto) y el backend responde 200 manteniendo el contrato JSON .
GET /metadata/categories?includeSubcategories=True	Con subcategorías	status_code = 200 ; data.categories[i].subcategories existe y no es None cuando se solicita ; subcategories es lista (y si aplica, con longitud esperada) .
GET /metadata/licenses	"Happy path"	status_code = 200 ; error = False y message de éxito (p. ej., "Licencias obtenidas exitosamente") ; data.licenses existe y es lista ; cada licencia contiene al menos id, name, spdxid, isosiapproved ; data.total == len(data.licenses) .

GET /metadata/licenses	Sin datos	status_code = 200 ; data.total = 0 ; data.licenses es lista vacía (o existe y su longitud es 0) .
GET /metadata/difficulties	“Happy path”	status_code = 200 ; error = False y message de éxito (p. ej., “Dificultades de proyecto devueltas exitosamente”) ; data.difficulties existe y es lista ; cada dificultad contiene al menos id, name, description ; data.total == len(data.difficulties) .
GET /metadata/difficulties	Sin datos	status_code = 200 ; data.total = 0 ; data.difficulties es lista vacía (o existe y su longitud es 0) .
GET /metadata/difficulties	Múltiples resultados / parametrizadas	Si el servicio retorna N dificultades, la respuesta debe reflejar data.total = N y len(data.difficulties) = N ; el endpoint acepta distintos nombres válidos (según tus pruebas parametrizadas) manteniendo el contrato JSON .
GET /metadata/subcategories/{subcategoryid}	“Happy path”	status_code = 200 ; error = False y message de éxito (p. ej., “Subcategoría obtenida exitosamente”) ; data.id == subcategoryid ; data contiene al menos id, name, categoryid, description.
(Aplicado a varios)	No requiere auth	Si el endpoint fue definido como público, debe responder 200 sin token (prueba “does not require auth”) manteniendo el contrato JSON completo.

Tabla 43: Tabla de criterios de aceptación

Fuente: Elaboración Propia.

```
test_endpoints\projects\test_artifacts_endpoints.py ..... [ 9%]
test_endpoints\projects\test_marketplace_endpoints.py ..... [ 16%]
test_endpoints\projects\test_media_endpoints.py ..... [ 22%]
test_endpoints\projects\test_metadata_endpoints.py ..... [ 31%]
test_endpoints\projects\test_project_crud_endpoints.py ..... [ 44%]
test_endpoints\projects\test_requirement_endpoints.py ..... [ 54%]
test_endpoints\projects\test_reviews_endpoints.py ..... [ 70%]
test_endpoints\projects\test_stats_endpoints.py ..... [ 77%]
test_endpoints\test_auth_endpoints.py ..... [ 89%]
test_endpoints\test_profile_endpoints.py ..... [100%]
```

Imagen 73: Evidencia de ejecución de prueba de integración.

Fuente: Screenshot en consola vs code (ejecución de pytest).

4.4.4.8.2.3. INTEGRACIÓN MANUAL (SWAGGER UI)

Como validación complementaria, se ejecutaron pruebas manuales desde la interfaz Swagger UI (OpenAPI) contra el backend en ejecución local (contenedores Docker). Esta verificación permitió evidenciar de manera visual el comportamiento del endpoint, inspeccionando parámetros, payloads y respuestas retornadas por la API, y confirmando que la documentación interactiva expone correctamente los recursos disponibles para su consumo. A continuación se presentan imágenes de uno de los muchos endpoints en los cuales se hicieron pruebas manuales del API.

```
{
  "error": false,
  "data": {
    "categories": [
      {
        "id": "string",
        "name": "string",
        "description": "string",
        "active": true,
        "created_at": "2026-02-28T17:21:41.291Z",
        "subcategories": [
          {
            "id": "string",
            "category_id": "string",
            "name": "string",
            "description": "string",
            "created_at": "2026-02-28T17:21:41.292Z"
          }
        ]
      }
    ],
    "total": 0
  },
  "message": "string"
}
```

Imagen 74: Evidencia de modelo esperado en endpoint de prueba de Metadata de un proyecto.

Fuente: Screenshot de Swagger Docs en localhost:8000.

```
Curl
curl -X 'GET' \
  'http://localhost:8000/api/v1/projects/categories?include_subcategories=false' \
  -H 'accept: application/json'

Request URL
http://localhost:8000/api/v1/projects/categories?include_subcategories=false

Server response
Code    Details
200
Response body
{
  "error": false,
  "data": {
    "categories": [
      {
        "id": "6bca25e4-4ad7-4599-9e91-c72dc612c450",
        "name": "Desarrollo de Software",
        "description": "Todo lo relacionado con creación de aplicaciones, sistemas y plataformas.",
        "active": true,
        "created_at": "2025-12-18T16:50:19.169625Z",
        "subcategories": null
      },
      {
        "id": "71d91a11-b0f1-4a70-973b-85769d94e868",
        "name": "Inteligencia Artificial",
        "description": "Proyectos que usan IA generativa, agentes, automatización inteligente, NLP, visión por computadora, etc.",
        "active": true,
        "created_at": "2025-12-18T16:50:33.180018Z",
        "subcategories": null
      },
      {
        "id": "28f053e8-3690-4f1d-ac24-8c2a6d13106f",
        "name": "Machine Learning & Data Science",
        "description": "Modelos predictivos, análisis de datos, sistemas de recomendación, estadística aplicada.",
        "active": true,
        "created_at": "2025-12-18T16:50:51.980133Z",
        "subcategories": null
      }
    ]
  }
}

Response headers
content-length: 2859
content-type: application/json
date: Sat, 28 Feb 2026 17:25:14 GMT
server: uvicorn
```

Imagen 75: Evidencia de ejecución de endpoint Metadata de un proyecto.

Fuente: Screenshot de Swagger Docs en localhost:8000.

4.4.4.8.3. PRUEBAS DE SISTEMA

En esta sección se describen las pruebas de sistema realizadas sobre la aplicación en ejecución, enfocadas en evaluar su comportamiento bajo condiciones de carga concurrente. A diferencia de las pruebas unitarias e integración (orientadas a validar lógica y contrato de

endpoints), estas pruebas se ejecutaron como una verificación end-to-end del sistema, observando la estabilidad general del servicio cuando múltiples usuarios realizan solicitudes simultáneas.

Para este propósito se utilizó **Locust**, una herramienta de pruebas de carga que permite simular usuarios virtuales realizando peticiones HTTP contra el backend, configurando escenarios de uso y aumentando progresivamente la concurrencia. Con **Locust** se ejecutaron escenarios representativos de consumo del API (por ejemplo, consultas de recursos de metadatos), con el fin de identificar posibles degradaciones de rendimiento, incrementos de latencia y aparición de errores cuando la demanda aumenta.

De manera general, durante la ejecución se observó el comportamiento de métricas de rendimiento como tiempos de respuesta (latencia), tasa de solicitudes y errores, con el objetivo de contar con evidencia del desempeño del sistema bajo carga y apoyar decisiones de ajuste o optimización si fuese necesario.

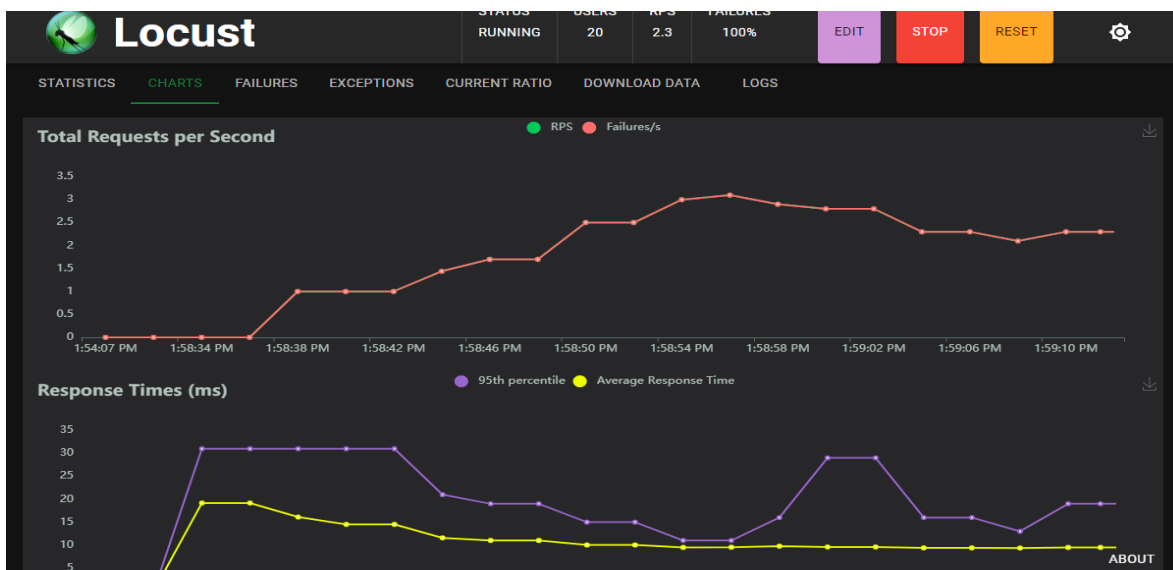


Imagen 76: Evidencia de ejecución Locust probando endpoints para medir problemas de latencia y rendimiento en el backend.

Fuente: Screenshot de interfaz web de Locust en localhost:8089.



Imagen 77: Evidencia 2 de ejecución Locust probando endpoints para medir problemas de latencia y rendimiento en el backend.

Fuente: Screenshot de interfaz web de Locust en localhost:8089.

4.4.4.8.4. NIVEL DE COBERTURA DE LAS PRUEBAS

Con el fin de cuantificar qué tanto del código fue ejecutado por la suite de pruebas, se midió la cobertura utilizando Pytest con el complemento pytest-cov, generando un reporte en formato HTML para su revisión visual. Como resultado de esta ejecución, el proyecto alcanzó 92% de cobertura total, lo cual evidencia una alta proporción de rutas de ejecución verificadas por las pruebas implementadas.

Para generar el reporte se ejecutó Pytest con la opción de cobertura y salida HTML por terminal de la siguiente manera: **pytest --cov --cov-report=html**

pytest-cov permite generar reportes en HTML mediante `--cov-report=html` (opcionalmente indicando un directorio de salida). El reporte HTML puede abrirse desde el archivo `htmlcov/index.html` (o el directorio configurado) para inspeccionar cobertura global y por archivo/módulo.

A continuación se presentan las evidencias pertinentes al reporte generado:

Coverage report: 92%

Files Functions Classes

coverage.py v7.13.4, created at 2026-02-28 12:45 -0500

File	statements	missing	excluded	coverage
app__init__.py	1	0	0	100%
app\api__init__.py	0	0	0	100%
app\api\v1__init__.py	0	0	0	100%
app\api\v1\endpoints__init__.py	3	0	0	100%
app\api\v1\endpoints\auth.py	64	0	0	100%
app\api\v1\endpoints\profile.py	41	0	0	100%
app\api\v1\endpoints\projects__init__.py	12	0	0	100%
app\api\v1\endpoints\projects\artifacts.py	26	0	0	100%
app\api\v1\endpoints\projects\crud.py	44	0	0	100%
app\api\v1\endpoints\projects\marketplace.py	24	0	0	100%
app\api\v1\endpoints\projects\media.py	23	2	0	91%
app\api\v1\endpoints\projects\metadata.py	22	0	0	100%
app\api\v1\endpoints\projects\requirements.py	28	0	0	100%
app\api\v1\endpoints\projects\reviews.py	35	0	0	100%
app\api\v1\endpoints\projects\stats.py	24	0	0	100%
app\api\v1\router.py	8	0	0	100%
app\core__init__.py	0	0	0	100%
app\core\config.py	113	31	0	73%
app\core\dependencies__init__.py	0	0	0	100%
app\core\dependencies\base.py	60	42	0	30%
app\core\dependencies\repositories.py	44	14	0	68%
app\core\dependencies\services.py	68	18	0	74%
app\core\logging_config.py	35	7	0	80%
app\core\redis_manager.py	24	11	0	54%
app\core\token_blacklist.py	35	24	0	31%
app\db__init__.py	0	0	0	100%
app\db\models__init__.py	0	0	0	100%

Imagen 78: Captura del resumen de cobertura (porcentaje total 92%) en el reporte HTML.

Fuente: Screenshot de reporte pytest cov en index.html raíz del proyecto.

4.4.4.9. ESTADO DE ENTREGA DEL ARTEFACTO

El artefacto se entrega como un **MVP funcional completamente operativo**, alineado con el alcance priorizado mediante la metodología MoSCoW y desarrollado bajo un enfoque ágil híbrido (Scrum + XP).

La construcción se consolidó en un sprint único extendido, gestionado mediante backlog estructurado y seguimiento en Azure DevOps, donde cada ítem cerrado en estado “Done” fue respaldado con evidencias verificables (capturas, resultados de pruebas, validaciones funcionales y comentarios asociados).

El sistema entregado incluye:

Módulo	Estado
Marketplace / Catálogo público	<i>Operativo</i>
Publicación estructurada (Wizard)	<i>Operativo</i>
Autenticación con Google	<i>Operativo</i>
Verificación estudiante	<i>Operativo</i>
Comentarios estructurados	<i>Operativo</i>
Sistema de rating	<i>Operativo</i>
Dashboard métricas generales	<i>Operativo</i>
Estadísticas individuales	<i>Operativo</i>
Documentación / Ayuda	<i>Operativo</i>

Tabla 44: Tabla de estado de entrega de artefacto.

Fuente: Elaboración Propia.

El artefacto fue sometido a:

- Pruebas unitarias (Pytest).
- Pruebas de integración. (Pytest + httpx.AsyncClient + ASGITransport)
- Pruebas manuales (Thunder Client / Swagger UI).
- Pruebas de carga con Locust para validar comportamiento bajo concurrencia.
- Medición de cobertura con Pytest-cov.

El estado de entrega corresponde a un sistema estable, funcional y técnicamente validado dentro del alcance definido para la Fase 3 del enfoque DSR (diseño y construcción del artefacto).

4.4.4.9.1. ALCANCE CONSTRUIDO VS. BACKLOG MOSCOW

El alcance construido se alinea directamente con el backlog priorizado bajo la metodología MoSCoW, considerando restricciones de tiempo (8 meses) y recursos (2 desarrolladores).

Estas funcionalidades constituyen el núcleo del sistema y fueron desarrolladas completamente, asegurando cumplimiento de los criterios de aceptación asociados (wizard <5 minutos, bloqueo sin documentación, dashboard funcional).

Prioridad	Requisitos	Descripción funcional	Estado en versión entregada
MUST	R1, R2, R3	Publicación con wizard obligatorio (README + licencia)	Implementado completamente
MUST	R7	Dashboard de métricas públicas	Implementado completamente
SHOULD	R4, R5	Filtros por categoría/rating	Implementado funcionalmente

COULD	R8	Sistema de comentarios estructurado	Implementado
WONT	R6, R9	Recomendaciones docentes + notificaciones realtime	No incluido (fase futura)

Tabla 45: Tabla de alcance construido vs backlog moscow

Fuente: Elaboración Propia.

Los elementos clasificados como WONT fueron explícitamente excluidos por decisión estratégica de priorización, en coherencia con la planificación inicial.

4.4.4.9.2. LIMITACIONES DE LA VERSIÓN ENTREGADA Y MEJORAS PROPUESTAS

Estas limitaciones responden a una decisión metodológica de priorización, orientada a consolidar primero la arquitectura base y el núcleo funcional antes de avanzar hacia fases de expansión o evaluación externa.

Área	Limitación	Impacto
Interacción avanzada	No incluye notificaciones en tiempo real	Retroalimentación no inmediata
Inteligencia académica	No incluye recomendaciones automatizadas	Menor personalización
Validación externa	No se realizó evaluación empírica con usuarios reales	Impacto no medido longitudinalmente
Escalabilidad institucional	No se ejecutó despliegue productivo a nivel institucional	Uso limitado a entorno controlado

Tabla 46: *Tabla de limitaciones de la versión entregada*

Fuente: *Elaboración Propia.*

REFERENCIAS

- [1] C. A. Lynch, "Institutional repositories: Essential infrastructure for scholarship in the digital age," *Portal: Libraries and the Academy*, vol. 3, no. 2, pp. 327–336, 2003. doi: 10.1353/pla.2003.0039
- [2] K. Shearer, "Institutional repositories: Towards the identification of critical success factors," in *Proc. Canadian Association for Information Science (CAIS)*, Halifax, Canada, 2003. [Online]. Available: http://www.caais-acsi.ca/proceedings/2003/Shearer_2003.pdf
- [3] D. G. Méndez López, "E-Clouds: un marketplace de software como servicio para computación científica," 2013.
- [4] J. L. Gallardo-Salazar and M. Rosas-Chavoya, "Difusión del software libre geoespacial: el caso QGIS México / Dissemination of free and open source geospatial software: a case QGIS Mexico," *Revista Geográfica Venezolana*, vol. 62, no. 2, pp. 532–544, 2021.
- [5] Crow, R. (2002). The case for institutional repositories: A SPARC position paper. Scholarly Publishing & Academic Resources Coalition (SPARC).
- [6] Olivera, L. E. C., Zambrano, P. E. E., Castillo, J. N. P., & Aguilar, L. J. (2009). Integración de Repositorios Digitales para la Gestión del Conocimiento en el ámbito

- [7] Universidad Popular del Cesar, "Repositorio Unicesar - Servicio para registro, consulta y preservación de la producción académica, científica e institucional," DSpace Angular. [En línea]. Disponible: <http://repositorio.unicesar.edu.co>.
- [8] Miao, H., Chavan, A., & Deshpande, A. (2017, May). Provd: Lifecycle management of collaborative analysis workflows. In *Proceedings of the 2nd Workshop on Human-in-the-Loop Data Analytics* (pp. 1-6).
- [9] A. Lima, L. Rossi, and M. Musolesi, "Coding together at scale: GitHub as a collaborative social network," in Proc. 8th Int. Conf. Weblogs and Social Media (ICWSM), pp. 295–304, 2014.
- [10] Suber, P. (2012). *Open access* (p. 256). The MIT Press.
- [11] B. Bravo-Díaz, Conocimiento Abierto (Open Access). La Serena, Chile: Universidad de La Serena, 2020. [Online]. Available: https://d1wqtxts1xzle7.cloudfront.net/61991767/Conocimiento_abierto_B_D20200204.
- [12] Benureau, F. C., & Rougier, N. P. (2018). Re-run, repeat, reproduce, reuse, replicate: transforming code into scientific contributions. *Frontiers in neuroinformatics*, 11, 69.
- [13] L. Sáenz-Tamariz, "La ingeniería del software educativo y la reutilización," Pixel-Bit. Revista de Medios y Educación, no. 27, pp. 69–80, 2006. [Online]. Available: <https://dialnet.unirioja.es/descarga/articulo/2281735.pdf>
- [14] M. Flores Chávez, "...," Tesis de maestría, Univ. Politécnica de Valencia, Valencia, España, 2019.
- [15] Escuela Superior Politécnica del Litoral – ESPOL, Herramientas de colaboración digital. Facultad de Ingeniería en Electricidad y Computación, 2023. [Online]. Available: https://www.fiec.espol.edu.ec/resources/materias/computacion/FIEC06460_HERRAMIENTA_S_DE_COLABORACION_DIGITAL.PDF
- [16] S. S. Sundar, "Theorizing interactivity's effects," *The Information Society*, vol. 20, no. 5, pp. 385–389, 2004. doi: 10.1080/01972240490508072

- [17] Larrondo Ureta, A. (2005). La interactividad en el ciberperiodismo: modelo para su análisis. Zer: Revista de Estudios de Comunicación, 10(19), 101–126. <https://ojs.ehu.eus/index.php/Zer/article/view/3914>
- [18] Codina, L. (2000). Interactividad y usabilidad: dos conceptos clave para el diseño de publicaciones digitales. BiD: textos universitarios de biblioteconomía y documentación.
- [19] A. M. Fuentes, "Gestión del conocimiento en instituciones educativas: conceptualización y aplicación," Revista Iberoamericana de Educación, vol. 52, no. 1, pp. 1–10, 2010. [Online]. Available: <https://www.redalyc.org/journal/551/55160059004/55160059004.pdf>.
- [20] J. Martínez-Bonafé, "Gestión del conocimiento educativo: Nuevos retos y posibilidades," Educación XX1, vol. 13, no. 2, pp. 179–198, 2010. [Online]. Available: <https://dialnet.unirioja.es/servlet/dctes?codigo=318073>
- [21] Sánchez-Tarazaga, L. (2016). Estrategias de gestión del conocimiento en universidades latinoamericanas.
- [22] E. Turban, C. Pollard, and G. Wood, Electronic Commerce: A Managerial and Social Networks Perspective. Cham, Switzerland: Springer, 2018.
- [23] K. C. Laudon and C. G. Traver, E-commerce 2020-2021: Business, Technology, Society. London, U.K.: Pearson, 2021.
- [24] Gefen, D., Karahanna, E., & Straub, D. W. (2003). Trust and TAM in online shopping: An integrated model. MIS quarterly, 51-90.
- [25] D. M. Vázquez, "Los repositorios digitales institucionales y el acceso abierto como herramienta de difusión de contenidos académicos," Espacio I+D, vol. 6, no. 15, 2017.
- [26] Maida, E. G., & Pacienza, J. (2015). Metodologías de desarrollo de software.
- [27] Budapest Open Access Initiative. (2002). Read the Budapest Open Access Initiative. <https://www.budapestopenaccessinitiative.org/read>

- [28] L. Sáenz-Tamariz, "La ingeniería del software educativo y la reutilización," Pixel-Bit. Revista de Medios y Educación, no. 27, pp. 69–80, 2006. [Online]. Available: <https://dialnet.unirioja.es/descarga/articulo/2281735.pdf>
- [29] Universidad Nacional Autónoma de México (UNAM), Teoría del conocimiento. Facultad de Contaduría y Administración, 2012. [Online]. Available: http://fcaenlinea.unam.mx/apuntes/interiores/docs/2012/informatica/1/teoria_del_conocimiento.pdf
- [30] A. García, "Repositorios web para la gestión de proyectos en empresas," Revista Digital de Investigación en Docencia Universitaria, vol. 23, no. 1, pp. 45–60, 2019. [Online]. Available: <https://www.redalyc.org/articulo.oa?id=163550800003>
- [31] J. A. Fernández, P. Navarro, and V. M. Ayala-Rivera, "Diseño de metadatos para la descripción de recursos educativos digitales," El Profesional de la Información, vol. 28, no. 2, p. e280204, 2019. doi: 10.3145/epi.2019.mar.04
- [32] N. Fernández and R. Rodríguez, "Diseño e implementación del Repositorio de Información del Centro de Estudios CAD/CAM en la Universidad de Holguín," Revista General de Información y Documentación, vol. 27, no. 2, pp. 317–340, 2017. doi: 10.5209/RGID.58206
- [33] M. A. Gómez and G. Benito, "Repositorios institucionales: una estrategia para la difusión y visibilidad de la producción científica," Investigación Bibliotecológica, vol. 32, no. 74, pp. 81–98, 2018.
- [34] A. Gómez-Muñoz and S. Arroyo-Moreno, "Los repositorios digitales institucionales: análisis comparativo de DSpace y EPrints," El Profesional de la Información, vol. 25, no. 2, pp. 298–305, 2016. doi: 10.3145/epi.2016.mar.13
- [35] M. R. Illescas Vásquez and D. B. Tapia Cadme, "...," Tesis de pregrado, Univ. Politécnica Salesiana, Cuenca, Ecuador, 2013.
- [36] D. Murillo, H. Calderon, and D. Saavedra, "Ecosistema tecnológico para mejorar la visibilidad de las publicaciones científicas," in Memorias de Congresos UTP, pp. 73–80, 2018.

- [37] A. Boni and G. Villaseñor, "El proyecto académico de la UAM en ciencias sociales," *Revista Mexicana de Ciencias Políticas y Sociales*, vol. 35, no. 136–137, 1989.
- [38] D. M. Vázquez, "Los repositorios digitales institucionales y el acceso abierto como herramienta de difusión de contenidos académicos," *Espacio I+D*, vol. 6, no. 15, 2017.
- [39] G. L. Villarrceal, E. Manzur, M. M. Vila, and M. R. De Giusti, "Interoperabilidad con repositorios digitales: Uso de OpenSearch... universitario Colombiano," *Ingeniería*, vol. 14, no. 1, pp. 59–68, 2017.
- [40] M. G. López, Y. Hernández Bieliukas, C. Beleño, D. Pernalet, V. Miguel, and N. Montaña, "Un repositorio basado en servicios web para el sistema generador de ambientes de aprendizaje AMBAR," en Ponencia en congreso, V Simposio Pluridisciplinar sobre Diseño y Evaluación de Contenidos Educativos Reutilizables, 2006.
- [42] G. A. Hurtado Vásquez, "Diseño y construcción de Electronic Marketplace," 2010.
- [43] J. A. Muñoz Vallejo, "Implementación de una aplicación web como repositorio de memoria social y saberes ancestrales para el Museo del Cantón Jipijapa," Bachelor's thesis, 2025.
- [44] J. Orssich Sánchez-Cortés, "Ready For You – Creación de un Marketplace," 2022.
- [45] M. Intriago and J. Anibal, "Diseño y desarrollo de un repositorio web multiplataforma de software libre," Bachelor's thesis, 2019.
- [46] L. Calaf Gómez, "Conceptualización y diseño de un prototipo de Marketplace para talleres y cursos en línea," 2024.
- [47] L. Morcillo López, "Los repositorios institucionales en las universidades públicas de España: estado de la cuestión," *Cuadernos de Gestión de Información*, vol. 6, pp. 69–83, 2016. [Online]. Available: <https://revistas.um.es/gesinfo/article/view/264121>
- [48] P. R. Martínez Lozada et al., "Repositorios de tesis de la Universidad de Ciencias Médicas de Holguín," *Correo Científico Médico*, vol. 23, no. 2, pp. 672–681, 2019.

- [49] S. Pinfield, J. Salter, and P. A. Bath, "The 'total cost of publication' in a hybrid open-access environment," *Journal of the Association for Information Science and Technology*, vol. 67, no. 7, pp. 1751–1766, 2016.
- [50] V. Guevara, "Desarrollo de un repositorio web de micro contenidos de aprendizaje para fortalecer la gestión del ciclo de vida de los objetos virtuales de aprendizaje (OVA)," 2021. [Online]. Available: <http://repositorio.utn.edu.ec/handle/123456789/10861>
- [51] C. González, E. Torres, and A. García, "Guía práctica para la implementación de repositorios digitales," *Revista Española de Documentación Científica*, vol. 43, no. 4, p. e266, 2020.
- [52] L. Rothfritz, L. Matthias, H. Pampel, and M. Wrzesinski, "Current challenges and future directions for institutional repositories: A systematic literature review," *Journal of the Association for Information Science and Technology*, 2025.
- [53] Universidad del Magdalena, "Repositorio Digital Universidad del Magdalena," Biblioteca Germán Bula Meyer, 2019. [En línea]. Disponible en: <https://repositorio.unimagdalena.edu.co>
- [54] Universidad del Norte, "Manglar – Repositorio Institucional," Dirección de Bibliotecas, 2013. [En línea]. Disponible en: <https://manglar.uninorte.edu.co>
- [55] Red Colombiana de Información Científica – Minciencias, "RedCol," 2020. [En línea]. Disponible en: <https://redcol.minciencias.gov.co>
- [56] Universidad Autónoma del Caribe, "Repositorio Institucional Universidad Autónoma del Caribe," [En línea]. Disponible en: <https://repositorio.uac.edu.co>. Accedido: 24-feb-2026.
- [57] Open Archives Initiative, "The Open Archives Initiative Protocol for Metadata Harvesting," 2002.
- [58] E. Bodero Poveda, M. R. De Giusti, and C. Morales, "Preservación digital a largo plazo: estándares, auditoría, madurez y planificación estratégica," *Revista Interamericana de Bibliotecología*, vol. 45, no. 2, 2022.

[59] UNESCO, "Policy Guidelines for the Development and Promotion of Open Access," Paris, France, 2012.

[60] J. L. Garrido, M. Gea, M. Noguera, M. González y J. A. Ibáñez, "Una arquitectura para diseñar e implementar Web Services," Revista TIA, Universidad Distrital Francisco José de Caldas.

[61] S. Rivera, "Aplicación web basada en arquitectura hexagonal para la gestión de información", Tesis de grado, Unicomfauca.

[62] Y. Páez Suárez, "Diseño de un sitio web educativo basado en una arquitectura modular y sostenible," Tesis de grado, Universidad Nacional Abierta y a Distancia (UNAD), 2024. [En línea].

[63] [Autor(es) desconocido(s)], "Implementation and Evaluation of Software Architecture," Tesis de maestría, DiVA Portal.

[64] [Autor(es) desconocido(s)], "Una Propuesta Arquitectónica para el Desarrollo de Aplicaciones Colaborativas," Academia.edu.

5. ANEXOS - INSTRUMENTOS

- [Instrumento A — Entrevista semiestructurada](#)
- [Instrumento B — Lista de observación no participante \(aulas/labs\)](#)
- [Instrumento C — Encuesta diagnóstica pre-artefacto \(Likert 1–5\)](#)