

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO- CHILE



**“SISTEMA DE APOYO BASADO EN INTELIGENCIA
ARTIFICIAL GENERATIVA PARA MEJORAR LA
ADMISIBILIDAD DE POSTULACIONES DE PYMES A
FONDOS CORFO”**

DIEGO ARMANDO PÉREZ OSORIO

**MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA**

Profesor Guía: Pedro Godoy
Profesor Correferente: Luis Hevia

Julio - 2026



CONSTANCIA DE VALIDACIÓN Y CONFIDENCIALIDAD DE MONOGRAFÍA A REPOSITORIO ACADÉMICO

1.- IDENTIFICACIÓN DEL TRABAJO ACADÉMICO

Tipo de monografía (marcar una opción): ☒ Memoria o trabajo de título ☐ Tesis de Postgrado

Título del trabajo: SISTEMA DE APOYO BASADO EN INTELIGENCIA ARTIFICIAL GENERATIVA PARA MEJORAR LA ADMISIBILIDAD DE POSTULACIONES DE PYMES A FONDOS CORFO

Nombre del candidato(a): Diego Armando Pérez Osorio

Carrera / Grado: Ingeniería Civil Informática

Campus: Casa Central

Departamento: Informática

2.- VALIDACIÓN DEL PROFESOR GUÍA/DIRECTOR DE TESIS

Yo, Pedro Francisco Godoy Barrera, en mi calidad de profesor(a) guía/director(a) del trabajo académico mencionado anteriormente **DEJO CONSTANCIA** que:

- He revisado esta versión del documento y corresponde a la versión final aprobada del trabajo.
- El trabajo cumple con los requisitos académicos y de formato establecidos por la institución.

3.- EVALUACIÓN DE CONFIDENCIALIDAD POR PROPIEDAD INDUSTRIAL (marcar una opción)

☒ El trabajo **NO contiene** información que amerite confidencialidad y puede ser publicado de inmediato en repositorio con acceso abierto.

☐ El trabajo **CONTIENE** información con potenciales implicancias de propiedad industrial o intelectual y requiere un periodo de confidencialidad (**embargo**) por (**marcar una opción**):

☐ 6 meses ☐ 12 meses ☐ 2 años ☐ 3 años ☐ 5 años ☐ 10 años

Fundamentación de la necesidad de confidencialidad (obligatorio si se solicita embargo):

4.- FIRMAS

Profesor(a) guía o director(a) de memoria o tesis:

Fecha: 29 de Julio de 2026 **Firma:** 

Estudiante o Candidato(a):

Fecha: 28-07-2026 **Firma:** Diego Pérez Osorio

Este formulario debe ser insertado como página 2 de la memoria o tesis, completado y firmado por estudiante y profesor(a) antes de la entrega en portal PRISMA de Biblioteca USM.

DEDICATORIA

Dedicado a mi familia y amigos. Ellos me dieron todo el apoyo que necesitaba y más, en las buenas, en las malas. Mis padres, Ana y Armando, me soportaron todos mis dramas y me hicieron llegar donde estoy ahora. La groupie, que me dio las mejores juntas para distender, e incluso para estudiar.

Catalina, quien me apañó todas y sigue haciéndolo.

Gracias

AGRADECIMIENTOS

Agradecer a mi familia querida, que no solo queda en padre, madre y hermanos.

Abuelas, tíos y primos que me apoyaron durante toda la etapa universitaria, que me escucharon mientras hablaba de que finalmente iba a terminar la carrera, que me apoyaron y hasta hicieron mandas para sacarme de acá.

En especial a mi Mamá y Papá, que siempre me mantuvieron con buen ánimo, que me enseñaron a perseverar, que me enseñaron a seguir adelante, que se preocuparon por mí y me llamaban casi todos los días para saber cómo estaba. Todavía está la foto pegada en el refrigerador de cuando entré a la universidad y estamos mi papá y yo, felices de que entrara.

Agradecer a mi pareja Catalina, que me acompañó en todas mis locuras y este largo proceso. Muchas gracias.

La Groupie fue mi centro, mis amigos, donde me sentía a gusto y podía sentirme yo, gracias por todos estos buenos momentos que quedan y por los que vendrán.

Agradecer también de corazón a toda la gente que conocí durante estos largos años que no diré cuántos fueron. Promoción Informática que me recibió siendo un pollito. El Centro de alumnos de informática del que también fui parte, Grande sea el Labpro y toda la gente que asistía, La salita de estar y el team BIS (Ex JIM). Fue un largo camino, pero fue muy entretenido.

Y por último, pero no menos importante, mi profesor guía Pedro Godoy, que me acompañó con todas mis situaciones, todos mis cachos y me acompañó para poder sacar esto adelante.

RESUMEN

Resumen Las pequeñas y medianas empresas (PYMEs) chilenas enfrentan una alta tasa de rechazo administrativo al postular a fondos CORFO, originada más en la asimetría de información que en la falta de mérito de sus proyectos. Este trabajo presenta Admisivo, un sistema de apoyo computacional basado en inteligencia artificial generativa que centraliza y clasifica convocatorias según el perfil del proyecto, evalúa el cumplimiento de los criterios de admisibilidad frente a un checklist estructurado, y entrega retroalimentación de redacción en tiempo real mediante un Taller de IA. Implementado como un producto mínimo viable (FastAPI, Next.js, Google Gemini), el sistema se validó mediante una prueba de usabilidad SUS con usuarios reales (91.25/100, categoría Excelente) y una validación técnica del motor de admisibilidad sobre 20 casos de prueba en 4 instrumentos CORFO, que permitió detectar y corregir dos defectos reales de robustez. Admisivo reduce la asimetría de información y sistematiza el proceso de postulación de forma medible, aunque su fiabilidad semántica frente a un experto humano en múltiples casos reales queda como validación pendiente y trabajo futuro.

Palabras Clave Admisibilidad CORFO; inteligencia artificial generativa; PYME; sistema de apoyo a la decisión; validación de precisión técnica.

ABSTRACT

Abstract Chilean small and medium-sized enterprises (SMEs) face a high administrative rejection rate when applying to CORFO public funding calls, driven more by information asymmetry than by a lack of merit in their projects. This work presents Admisivo, an AI-based support system that centralizes and classifies funding calls by project profile, evaluates compliance with admissibility criteria against a structured checklist, and provides real-time drafting feedback through an AI Workshop. Implemented as a minimum viable product (FastAPI, Next.js, Google Gemini), the system was validated through a System Usability Scale test with real users (91.25/100, rated Excellent) and a technical validation of the admissibility engine over 20 test cases across 4 CORFO instruments, which enabled detecting and fixing two real robustness defects. Admisivo measurably reduces information asymmetry and systematizes the application process, although its semantic reliability against human expert judgment across multiple real cases remains an open validation and future work.

Keywords CORFO admissibility; generative artificial intelligence; SME; decision support system; technical precision validation.

GLOSARIO

CORFO: Corporación de Fomento de la Producción. Agencia del Gobierno de Chile encargada de apoyar el emprendimiento, la innovación y la competitividad en el país.

CoT: *Chain-of-Thought* (Cadena de Pensamiento). Técnica de *prompting* que instruye a un LLM a descomponer un problema complejo en pasos lógicos para mejorar la calidad de su razonamiento.

Equity: Término en inglés para referirse a la participación accionaria o al capital de una empresa. En el contexto de las *startups*, a menudo se cede un porcentaje a cambio de inversión o asesoría especializada.

FOSIS: Fondo de Solidaridad e Inversión Social. Institución del Gobierno de Chile que apoya a personas en situación de pobreza y vulnerabilidad.

GPT: *Generative Pre-trained Transformer* (Transformador Generativo Pre-entrenado). Tipo de arquitectura de modelo de lenguaje grande desarrollada por OpenAI.

I+D: Investigación y Desarrollo. Actividades destinadas a la creación de nuevos conocimientos o la aplicación de estos para el desarrollo de nuevos productos, servicios o procesos.

LLM: *Large Language Model* (Modelo de Lenguaje Grande). Modelo de inteligencia artificial entrenado con grandes cantidades de datos de texto para comprender y generar lenguaje natural.

Mipyme: Acrónimo para Micro, Pequeña y Mediana Empresa.

MVP: *Minimum Viable Product* (Producto Mínimo Viable). Versión de un nuevo producto que permite a un equipo recolectar la máxima cantidad de aprendizaje validado sobre los clientes con el mínimo esfuerzo.

NER: *Named Entity Recognition* (Reconocimiento de Entidades Nombradas). Técnica de Procesamiento de Lenguaje Natural (PLN) para identificar y clasificar entidades específicas en un texto, como nombres, fechas o montos de dinero.

PLN: Procesamiento de Lenguaje Natural. Campo de la inteligencia artificial centrado en la interacción entre las computadoras y el lenguaje humano.

PYME: Pequeña y Mediana Empresa.

RAG: *Retrieval-Augmented Generation* (Generación Aumentada por Recuperación). Enfoque que combina un LLM con una base de conocimiento externa para generar respuestas más precisas y actualizadas.

Sercotec: Servicio de Cooperación Técnica. Organismo del Gobierno de Chile que apoya a las micro y pequeñas empresas y a los emprendedores.

VC: *Venture Capital* (Capital de Riesgo). Forma de financiamiento privado proporcionado a *startups* o empresas emergentes que se considera tienen un alto potencial de crecimiento.

TABLA DE CONTENIDOS

DEDICATORIA	2
AGRADECIMIENTOS	3
RESUMEN.....	4
ABSTRACT	4
GLOSARIO	5
TABLA DE CONTENIDOS	6
ÍNDICE DE FIGURAS	9
ÍNDICE DE TABLAS	10
INTRODUCCIÓN.....	11
CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA	12
1.1 Contexto.....	12
1.2 Dificultades actuales de los actores involucrados	14
1.2.1 Pequeñas y medianas empresas (PYMEs)	14
1.2.2 Empresas aceleradoras, incubadoras y venture capital	16
1.2.3 CORFO	16
1.3 Consecuencias de no abordar el problema.....	16
1.4 Cuantificación y Naturaleza del Problema.....	17
1.4.1 Análisis Cuantitativo de la Barrera de Admisibilidad	17
1.5 Objetivos y alcances de la memoria	23
1.5.1 Objetivo general	23
1.5.2 Objetivos específicos	23
1.5.3 Alcances y Limitaciones.....	23
CAPÍTULO 2: MARCO CONCEPTUAL	24
2.1 Análisis y Estructura de Postulaciones a CORFO	24
2.1.1 El Postulante: Perfil de la PYME Innovadora en Chile	24
2.1.2 Misión y Prioridades Estratégicas de CORFO.....	26
2.1.3 Componentes Clave de una Postulación.....	26
2.1.4 Criterios de Evaluación	27
2.1.5 Desafíos y Complejidades en la Postulación.....	28
2.2 Herramientas Tecnológicas para la Generación de Conocimiento	29
2.2.1 Descripción General de los <i>LLMs</i>	29
2.2.2 Conceptos Técnicos Fundamentales: <i>Tokenización</i> y Ventana de Contexto	30
2.2.3 Capacidades de los <i>LLMs</i> Relevantes para la Asistencia en Postulaciones	30
2.2.4 Limitaciones y Desafíos Inherentes	31

2.2.5 Análisis del Estado del Arte en Ecosistemas de IA	31
2.3 Ingeniería de <i>Prompts</i>	32
2.4 Generación Aumentada por Recuperación (RAG)	33
2.4.1 Componentes de un Sistema de Recuperación	33
2.4.2 Proceso Técnico de RAG	34
2.5 Extracción de Información y Análisis de Coherencia	34
2.5.1 Reconocimiento de Entidades Nombradas (NER)	34
2.6 Evaluación de la Calidad de la Asistencia	35
2.7 Consideraciones Éticas y de IA responsable	36
CAPÍTULO 3: PROPUESTA DE SOLUCIÓN	37
3.1 Visión General y Arquitectura del Sistema (Modelo C4)	37
3.1.1 Diagrama de Contexto (C4 - Nivel 1).....	39
3.1.2 Arquitectura de Contenedores (C4 - Nivel 2).....	40
3.1.3 Diagrama de Componentes (C4 - Nivel 3: <i>Backend</i>).....	41
3.1.4 Mecanismos de Robustez del Scraper y Autenticación ante ClaveUnica	43
3.1.5 Diseño de Prompts para el Matching Híbrido	44
3.2 Pipeline de Procesamiento de Documentos: El Paradigma "In-Prompt"	44
3.2.1 Justificación Técnica de la Ausencia de RAG	44
3.2.2 Fases del Pipeline de Extracción y Procesamiento	45
3.3 Modelo de Datos Relacional.....	46
3.3.1 Contrato de Datos: Esquemas de Pydantic	48
3.4 Motor de Emparejamiento (<i>Matching</i> Híbrido).....	48
3.4.1 Capa 1: Filtros Técnicos Determinísticos	48
3.4.2 Capa 2: Refinamiento Cualitativo IA (Discovery).....	49
3.5 Motor de Admisibilidad Dinámica	49
3.5.1 Evaluación por IA contra el <i>Checklist</i> y Reconciliación del Veredicto	49
3.5.2 Formulario Dinámico y Análisis Combinado	50
3.6 Ingeniería de <i>Prompts</i> Avanzada y Mitigación de Alucinaciones	51
3.6.1 Modo JSON Restringido y Validación Pydantic	51
3.6.2 Control Térmico (Temperature Control).....	51
3.6.3 Resiliencia a Fallas y Trazabilidad (<i>Fallbacks</i>).....	52
3.7 Taller de Mejora Iterativa (Taller IA)	52
3.7.1 Vista Doble y Edición Asistida	52
3.7.2 Regeneración Controlada de Sugerencias	53
3.7.3 Persistencia, Reanálisis y Cierre del Ciclo	53
3.7.4 Telemetría de Uso.....	53

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN	54
4.1 Estrategia General de Validación	54
4.2 Validación de Usabilidad: Test SUS con Usuarios	55
4.2.1 Alianza institucional y diseño del instrumento	55
4.2.2 Perfil de los Participantes	56
4.2.3 Resultados Cuantitativos del SUS	56
4.2.4 Observaciones Cualitativas del Protocolo de Tareas	58
4.2.5 Hallazgos Críticos y Defectos de Usabilidad Detectados	61
4.3 Validación Técnica: Precisión de Detección de Admisibilidad	62
4.3.1 Diseño del Set de Casos de Prueba (5N)	62
4.3.2 Hallazgo: No Conformidad de Vocabulario en el Veredicto Agregado	63
4.3.3 Correcciones Aplicadas y Resultados de la Corrida Piloto	64
4.3.4 Caso Real Complementario: MunIA (Start-Up Chile)	65
4.3.5 Alcance y Limitaciones	66
4.4 Síntesis de la Validación	66
CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES	68
5.1 Cumplimiento de los Objetivos	68
5.2 Validación Temprana de Requerimientos	69
5.3 Decisiones de Infraestructura Técnica	70
5.4 Limitaciones	71
5.5 Trabajo Futuro	71
ANEXOS	73
Anexo A: Recorrido visual de la plataforma	73
Referencias bibliográficas	77

ÍNDICE DE FIGURAS

Figura 1: Árbol del problema Fuente: Elaboración propia	21
Figura 2: Sectores económicos Fuente: Data Innovación 2023	25
Figura 3: Diagrama de flujo Fuente: Elaboración propia.....	38
Figura 5: Contenedores Fuente: Elaboración propia	40
Figura 6: Pseudocódigo de estabilización del DOM Fuente: Elaboración Propia	43
Figura 7: Modelo de datos funcionalidades Fuente: Elaboración propia	46
Figura 8: Estados contradictorios en el checklist del Taller IA Fuente: Captura de sesión participante P3	60
Figura 9: Módulo de Análisis Rápido Fuente: Elaboración propia	73
Figura 10: Formulario dinámico Fuente: Elaboración propia.....	74
Figura 11: Diagnóstico de admisibilidad Fuente: Elaboración propia	74
Figura 12: Taller IA Fuente: Elaboración propia	75
Figura 13: Panel de proyectos Fuente: Elaboración propia	76

ÍNDICE DE TABLAS

Tabla 1: Comparación de Asesores Fuente: Elaboración propia..... 15

Tabla 2: Matriz de Errores Frecuentes en Postulaciones Fuente: Elaboración propia 19

Tabla 3: Actores y descripciones Fuente: Elaboración Propia 40

Tabla 4: Modelo Pydantic Fuente: Elaboración propia 48

Tabla 5: Tareas del protocolo de usabilidad y módulo evaluado Fuente: Elaboración Propia
..... 55

Tabla 6: Respuestas brutas del cuestionario SUS (escala 1-5) Fuente: Elaboración Propia 56

Tabla 7: Puntaje SUS individual y subescalas Fuente: Elaboración Propia 57

INTRODUCCIÓN

Las pequeñas y medianas empresas (PYMEs) representan un motor central de innovación y crecimiento económico en Chile. Los fondos concursables de la Corporación de Fomento de la Producción (CORFO) constituyen una de las vías de financiamiento más relevantes para escalar sus proyectos; sin embargo, entre el 70% y el 80% de las postulaciones son descartadas en las etapas iniciales de elegibilidad y admisibilidad. Este fenómeno responde mayoritariamente a errores formales evitables como formularios mal completados o criterios mal interpretados, y no a una carencia de mérito técnico o comercial en los proyectos. Esta brecha deriva de una asimetría de información entre las exigencias institucionales y la capacidad operativa de un postulante típico, quien a menudo carece de experiencia previa en fondos públicos y de recursos para contratar asesoría especializada.

Las alternativas actuales, como consultoras privadas y programas de mentoría, imponen costos económicos o condiciones de participación accionaria (*equity*) prohibitivos para PYMEs en etapas tempranas. Ante esta realidad, este trabajo presenta *Admisivo*, un sistema de apoyo computacional basado en inteligencia artificial generativa diseñado para democratizar el acceso a información estratégica. La plataforma integra tres pilares funcionales: un módulo de búsqueda inteligente que identifica la convocatoria óptima para el perfil del proyecto; un motor que audita el cumplimiento de criterios de admisibilidad mediante un *checklist* estructurado; y un Taller de IA que optimiza la calidad de la propuesta con retroalimentación en tiempo real. La solución fue implementada como un producto mínimo viable (*stack FastAPI*, Next.js y Google Gemini) y validada mediante un estudio de usabilidad con usuarios reales y una validación técnica sobre un conjunto controlado de casos de prueba.

Para exponer el trabajo desarrollado, el presente documento se estructura de la siguiente manera: el Capítulo 1 define el problema y los objetivos del trabajo; el Capítulo 2 expone el marco conceptual; el Capítulo 3 detalla la arquitectura y el diseño técnico de la solución; el Capítulo 4 presenta los resultados de la validación de usabilidad y precisión técnica; finalmente, el Capítulo 5 discute el cumplimiento de los objetivos, las limitaciones técnicas y las líneas de trabajo futuro.

CAPÍTULO 1: DEFINICIÓN DEL PROBLEMA

1.1 Contexto

En el dinámico ecosistema empresarial de Chile, las pequeñas y medianas empresas (PYMES) son un motor fundamental para la innovación y el crecimiento económico. Para materializar su potencial, escalar sus operaciones y desarrollar nuevas soluciones, el acceso a financiamiento es un paso crítico. En este contexto, la Corporación de Fomento de la Producción (CORFO) se presenta como una de las principales entidades que ofrece fondos para fomentar la inversión, la innovación y el emprendimiento en el país.

Sin embargo, el proceso de postulación a estos fondos representa una barrera significativa para muchas PYMES. La complejidad de los requisitos y la necesidad de articular una propuesta de valor robusta y bien fundamentada, demandan un nivel de experiencia y acceso a información que no siempre está al alcance de estos microempresarios. Frecuentemente, quienes buscan apoyo recurren a entidades asesoras como incubadoras o aceleradoras de negocios, las cuales brindan un acompañamiento valioso.

Este acompañamiento o mentoría se realiza periódicamente, revisando la propuesta completa con el objetivo de mejorarla. Específicamente, se analizan aspectos clave como:

- La definición y cuantificación del segmento de mercado.
- El impacto potencial del proyecto.
- La solución propuesta: qué se ofrece, cómo se define, su impacto cuantificado, los ciclos de venta y producción, y la escalabilidad del proyecto.
- Proyecciones a 3 años: financieras y de modelo de negocio.

La complejidad del proceso no radica únicamente en la dificultad de redactar un buen proyecto, sino también en la navegación del propio ecosistema de financiamiento de CORFO. La corporación ofrece una multiplicidad de instrumentos diseñados para distintas etapas, sectores y objetivos: desde capital semilla para ideas incipientes (como Semilla Inicia), programas de escalamiento (Start-Up Chile), incentivos para la innovación (Crea y Valida), hasta créditos para la sostenibilidad (Crédito Verde). Ante esta multiplicidad de instrumentos, y en ausencia de un mecanismo de orientación centralizado, el postulante debe optar entre dos rutas: un proceso de investigación manual, revisando individualmente las bases técnicas de cada convocatoria para determinar cuál corresponde a su perfil, o recurrir a asesoría externa de incubadoras o consultores, cuyo costo se detalla en la sección 1.2.1. En ausencia de ambas, la selección incorrecta del instrumento se traduce en un rechazo evitable, cuyo único efecto para el postulante es la constatación tardía del error, después de haber invertido los recursos requeridos para preparar la postulación.

Esta diversidad de fondos, presentada sin una guía clara, deriva en que la elección incorrecta del programa sea una causa frecuente y evitable de rechazo, con el consiguiente

costo de tiempo y recursos para el postulante. A esta complejidad se suma la superposición de prioridades estratégicas que CORFO ha ido incorporando, como el fomento a la descentralización, el impulso a la sostenibilidad y la promoción de la equidad de género, lo que eleva el desafío de una sobrecarga informativa simple a uno de alineación estratégica.

1.2 Dificultades actuales de los actores involucrados

Para comprender mejor la problemática, es fundamental centrarse en las dificultades que enfrentan las PYMES, así como los desafíos que esto genera en los demás actores del ecosistema.

1.2.1 Pequeñas y medianas empresas (PYMES)

Falta de experiencia en formulación de proyectos: Más allá de la escritura, esto se manifiesta en la dificultad para traducir una visión de negocio en un plan de trabajo coherente, con hitos medibles y un presupuesto justificado. A menudo, los emprendedores son expertos en su producto o servicio, pero no en la metodología de formulación de proyectos que exigen las agencias de fomento, lo que resulta en propuestas que carecen de la estructura y el rigor necesarios para ser competitivas. Estos errores no son aleatorios, sino que siguen patrones recurrentes que evidencian un desconocimiento sistemático de los criterios formales de evaluación. Se evidencia que una proporción significativa de las propuestas presenta deficiencias en la generación de valor informativo. Estos documentos tienden a priorizar aspectos estéticos, resultando en descripciones ambiguas que carecen de la precisión técnica y cuantitativa necesaria para justificar la solución propuesta ante los evaluadores. Un error crítico, señalado por la propia CORFO, es la incapacidad de destacar la innovación; muchas postulaciones se limitan a declarar que su solución es "innovadora" sin explicar por qué o en qué se diferencia de las alternativas existentes. Algunos ejemplos de errores comunes que cometen al momento de postular son: no cumplir con requisitos de tipo de postulante, estado de desarrollo incorrecto, alcance comercial acotado, o desalineación entre objetivos y gastos, entre otros. [1]

Acceso limitado a asesoría especializada: La barrera no es solo económica. Involucrarse con una aceleradora a menudo implica un costo de oportunidad significativo, como ceder un porcentaje de participación futura (equity), que suele ser no negociable y oscila entre el 3% y el 7% del capital del startup. [2] Esta cesión de capital, si bien abre puertas a inversión y mentoría, implica una pérdida de autonomía en la toma de decisiones y la presión de los inversores por maximizar su retorno. Dentro de estas existen múltiples tipos de entidades de apoyo. Un ejemplo son las consultoras privadas, cuyo objetivo específico es maximizar las probabilidades de adjudicación a través de su conocimiento experto del sistema. Este servicio tiene costos directos que pueden ser una barrera inicial importante. Por ejemplo, algunas firmas operan con un modelo de pago que incluye un honorario fijo inicial, que puede partir desde los \$750.000 + IVA, y un segundo pago de igual monto condicionado al éxito de la postulación [3]. Para muchas PYMES en etapa temprana, estos costos, tanto directos como indirectos, son prohibitivos. Estos pueden ser vistos en la siguiente tabla.

SISTEMA DE APOYO BASADO EN INTELIGENCIA ARTIFICIAL GENERATIVA PARA MEJORAR LA ADMISIBILIDAD
DE POSTULACIONES DE PYMES A FONDOS CORFO

Tabla 1: Comparación de Asesores
Fuente: Elaboración propia

Tipo de Asesor	Oferta de Servicio Principal	Modelo de Costo	Etapas Pyme Objetivo	Ejemplos
Consultoría Privada	Formulación y postulación de proyectos a fondos públicos.	Honorarios por servicio (pago inicial + pago por éxito).	Idea, Etapa Temprana, Crecimiento.	AsesoriasCorfo.cl, CIDIF, Vía Pública Consultores.
Incubadora Universitaria	Programa de incubación integral: mentoría, redes, validación, apoyo en postulación.	Tarifa de programa, a menudo subsidiada o con becas. Algunas pueden tomar <i>equity</i> .	Idea, Pre-semilla, Etapa Temprana.	UDD Ventures, Incuba UC, 3iE (USM), Incubatec UFRO.
Aceleradora de Negocios (VC)	Programa de aceleración enfocado en escalamiento rápido y levantamiento de capital.	Participación accionaria (<i>equity</i>) no negociable (típicamente 3-7%).	Etapas Temprana (con MVP), Crecimiento.	Magical Startups, Platanus Ventures.
Programas Públicos de Apoyo	Asesoría, capacitación y acompañamiento técnico y de gestión (no formulan proyectos).	Gratuito para la Pyme (financiado por el Estado).	Micro, Pequeña y Mediana Empresa.	Centros de Negocios Sercotec, Fortalece Pyme (CORFO).

Brecha de información para la formulación estratégica: Existe una dificultad notable para acceder a datos cruciales que fundamentan la propuesta, como proyecciones de mercado, tasas de crecimiento (CAGR), o cuantificación de segmentos. Esto ralentiza el proceso, resta solidez a la propuesta y hace que las mentorías sean menos eficientes, ya que se invierte tiempo en la búsqueda de información básica en lugar de en la estrategia.

Desconocimiento de la oferta de financiamiento: Como se mencionó, no se trata de una escasez de opciones, sino de lo contrario: un volumen tan vasto y fragmentado de instrumentos públicos que genera una "niebla" informativa. Cada uno de estos fondos tiene un nombre, objetivo, monto, plazo y público objetivo distintos. Además, el financiamiento no emana de una única ventanilla. Sino que provienen de una diversidad de agencias como CORFO, Sercotec, *FOSIS*, y ministerios sectoriales (como el de las Culturas, las Artes y el Patrimonio o el de Agricultura), cada uno con sus propios portales, criterios y procesos. Las PYMEs invierten tiempo y esfuerzo en postular a convocatorias para las cuales no cumplen el perfil de etapa, ventas o enfoque temático, llevando a rechazos automáticos que podrían haberse evitado con una mejor orientación inicial.

1.2.2 Empresas aceleradoras, incubadoras y venture capital

Limitaciones en la escalabilidad y cuello de botella operativo: El modelo de negocio de estas entidades depende de procesar un alto volumen de *startups* para identificar y potenciar aquellas con mayor proyección. Sin embargo, debido a que las propuestas iniciales de las PYMEs suelen carecer de estructura formal y coherencia estratégica, los evaluadores se ven obligados a invertir una cantidad desproporcionada de horas-hombre en corregir errores básicos de admisibilidad y elegibilidad. Esta fricción técnica satura su capacidad operativa, impidiéndoles escalar su cartera y desviando su foco del análisis real de viabilidad técnica y comercial.

Restricciones de monetización y alto costo de oportunidad: Las PYMEs en etapas tempranas rara vez disponen de liquidez para costear honorarios de consultoría por adelantado, obligando a las incubadoras a operar bajo modelos diferidos o exigiendo participación accionaria (*equity*). Si la entidad invierte semanas de mentoría en un proyecto que finalmente es rechazado por un error administrativo en el formulario estatal, asume una pérdida total del retorno sobre el tiempo invertido. Por tanto, la asimetría de información del ecosistema representa para ellas un riesgo financiero directo que limita a quiénes pueden ayudar.

1.2.3 CORFO

Para la corporación, la alta tasa de fallos en postulaciones tempranas no responde a una incapacidad operativa, sino que representa una oportunidad de optimización de su proceso de evaluación.

Sobrecarga de la capacidad de evaluación: Al recibir un alto volumen de proyectos anuales, una proporción crítica llega con deficiencias formales que no superan la etapa de admisibilidad. El sistema actual obliga a CORFO a destinar considerables recursos humanos y técnicos a auditar el cumplimiento de formatos, reglas y requisitos básicos, actuando como filtro administrativo en lugar de evaluador de mérito.

Oportunidad de maximización del impacto público: Si se logra depurar la calidad de las postulaciones que ingresan al proceso, permitiendo que las PYMEs pre-validen y estructuren formalmente sus propuestas, CORFO podría reasignar ese esfuerzo institucional hacia la evaluación profunda del grado de innovación y tracción real. Sistematizar la mejora en la calidad de origen de las propuestas permite inyectar los recursos con mayor precisión técnica, elevando la eficiencia del gasto fiscal.

1.3 Consecuencias de no abordar el problema

Si no se implementa una solución que cierre esta brecha de herramientas y acceso a la información para las PYMEs, podrían persistir o agravarse las siguientes situaciones:

Perpetuación de la brecha de acceso al financiamiento: Empresas con ideas y proyectos de alto potencial innovador podrían seguir quedando sin acceso a fondos cruciales, no por falta

de mérito, sino por no poder plasmarlo eficazmente en una propuesta, dado que la evaluación de CORFO se basa exclusivamente en el contenido del formulario presentado. Una propuesta mal redactada, con ausencia de datos cuantitativos o con proyecciones poco realistas, resulta en un rechazo en la primera etapa, la admisibilidad [5]. Esto crea un sesgo que no favorece necesariamente a los mejores proyectos, sino a los mejor formulados.

Ineficiencia en el proceso de formulación: El tiempo es un recurso limitado para un emprendedor. Cada hora dedicada a interpretar las bases de postulación, recopilar datos de mercado de forma manual o corregir errores de formulación es una hora no dedicada al desarrollo del producto, la atención de clientes o la operación del negocio. Un tiempo de formulación prolongado más allá de lo necesario puede, por tanto, comprometer las operaciones centrales de la PYME.

Baja eficiencia en la asignación de recursos públicos: CORFO podría destinar fondos a proyectos que, si bien están bien formulados, no necesariamente tienen el mayor potencial de impacto, simplemente porque otras propuestas superiores no lograron articularse correctamente.

1.4 Cuantificación y Naturaleza del Problema

Para dimensionar la magnitud del desafío, es crucial analizar los datos concretos del proceso de postulación y las causas de fallo.

1.4.1 Análisis Cuantitativo de la Barrera de Admisibilidad

A nivel global, el ecosistema de financiamiento de CORFO presenta un embudo de selección sumamente restrictivo en sus etapas iniciales. Solo en el año 2022, de las 10.349 postulaciones recibidas transversalmente, más de 8.400 fueron declaradas no admisibles. Esto significa que el esfuerzo del 81% de los postulantes de ese año fue desestimado por errores formales o de pertinencia, sin que la calidad o el potencial técnico de sus ideas llegaran siquiera a ser evaluados por el comité de mérito [4].

Esta fricción sistémica se evidencia de manera elocuente al hacer un doble clic en instrumentos específicos de alta demanda, como el programa **Semilla Inicia**. Los datos históricos consolidados de este fondo (actualizados al ciclo 2023 en la plataforma DataInnovación) muestran una drástica caída en su tasa de admisibilidad: desplomándose de un 57,2% en 2019 a solo un 18,9% en 2022 [5]. Si bien el volumen de emprendedores postulantes mantiene una tendencia al alza, esta severa tasa de atrición técnica demuestra empíricamente que el verdadero desafío para las PYMEs no se concentra en la competencia final por los cupos, sino en lograr flanquear el primer filtro administrativo.

Aunque las razones específicas de rechazo caen dentro de la confidencialidad de cada postulante, CORFO provee múltiples instancias para categorizar los errores estructurales más comunes que explican este 81% de mortalidad inicial.

1.4.1.1 Factores Críticos de Rechazo en Postulaciones CORFO

Comprender las causas del fallo es fundamental para plantear una solución. Si bien la estructura exacta puede variar entre convocatorias, el proceso de descarte de proyectos en CORFO generalmente ocurre en tres grandes etapas: **Elegibilidad, Admisibilidad y Evaluación Técnica o de Mérito**. Un alto volumen de postulaciones no logra superar los dos primeros filtros formales.

1.4.1.2 La barrera de la Elegibilidad: ¿Cumple el postulante?

La primera fase se centra en verificar que el *postulante* (la persona o empresa) cumple con todos los requisitos legales, administrativos y formales definidos en las bases. Es un filtro de cumplimiento estricto. Se estima que, en conjunto con la admisibilidad, estas etapas iniciales descartan a un **70% u 80% de los proyectos**.^[5] Los errores comunes que llevan a la no elegibilidad incluyen:

- **Incumplimiento del perfil del postulante:** No cumplir con la antigüedad requerida de la empresa, tener deudas previsionales o tributarias, no figurar en registros de deudores, o tener incompatibilidades por haberse adjudicado otros fondos CORFO recientemente.
- **Documentación incompleta o incorrecta:** No adjuntar todos los documentos solicitados en los formatos adecuados, o presentar archivos ilegibles es una causa directa de descarte en esta fase.

1.4.1.3 La barrera de la admisibilidad: ¿Es pertinente el proyecto?

Una vez que el postulante es declarado elegible, se evalúa si el *proyecto* en sí es pertinente para la convocatoria. El objetivo es asegurar que la propuesta se alinea con los fines del instrumento de financiamiento. Las causas de inadmisibilidad incluyen:

- **Postulación a un programa incorrecto:** Presentar un proyecto que no corresponde a la etapa de desarrollo exigida (ej. una idea para un fondo de escalamiento) o cuyo foco temático no se alinea con la convocatoria (ej. un proyecto sin impacto ambiental para un "Crédito Verde").
- **Alcance limitado o inadecuado:** Proponer un proyecto con un impacto meramente local cuando el programa busca explícitamente un alcance nacional o global.
- **Errores técnicos:** Un enlace de video que no funciona o requiere contraseña es una causa directa de inadmisibilidad.

1.4.1.4 Las debilidades en la Evaluación de Mérito

Solo los proyectos que superan ambos filtros formales avanzan a la evaluación de fondo, donde se analiza su calidad y potencial. En esta etapa, las debilidades son más cualitativas:

- **Grado de innovación insuficiente:** La propuesta no logra demostrar claramente su diferenciación sustancial respecto a lo que ya existe.
- **Modelo de negocio débil o poco escalable:** Falta de claridad en cómo el proyecto generará ingresos de forma sostenible y crecerá significativamente.
- **Equipo emprendedor insuficiente:** No demostrar que el equipo posee las competencias técnicas, de gestión y comerciales para ejecutar el proyecto.

Estos factores, que abarcan desde el cumplimiento formal del postulante hasta la solidez de la propuesta, subrayan la necesidad de una herramienta que guíe a las PYMEs a través de estas complejidades, ayudándolas a evitar errores comunes y a fortalecer su postulación en todas las etapas.

Tabla 2: Matriz de Errores Frecuentes en Postulaciones
Fuente: Elaboración propia

Etapas de Rechazo	Tipo de Error	Descripción del Error	Ejemplo Concreto
Elegibilidad	Formal/Administrativo	Incumplimiento del perfil del postulante.	La empresa tiene 20 meses de antigüedad y postula a Semilla Inicia (límite 18 meses).
Elegibilidad	Formal/Administrativo	Deudas o incompatibilidades.	El postulante figura con deudas tributarias o ya se adjudicó un fondo CORFO incompatible.
Admisibilidad	Coherencia del Proyecto	Postulación a un programa incorrecto.	Se presenta un proyecto en etapa de idea a un fondo de escalamiento que exige ventas.
Admisibilidad	Coherencia del Proyecto	Alcance del proyecto inadecuado.	Se propone una solución de impacto local para un programa que busca explícitamente alcance nacional.
Admisibilidad	Formal/Administrativo	Video de postulación inaccesible.	El enlace al video en el formulario está roto o requiere una contraseña no proporcionada.
Admisibilidad	Coherencia del Proyecto	Desalineación entre objetivos y gastos.	Se declara que el subsidio se usará para comprar maquinaria, actividad no financiable por el programa.

SISTEMA DE APOYO BASADO EN INTELIGENCIA ARTIFICIAL GENERATIVA PARA MEJORAR LA ADMISIBILIDAD
DE POSTULACIONES DE PYMES A FONDOS CORFO

Evaluación de Mérito	Contenido Estratégico	Grado de innovación no justificado.	La propuesta afirma ser "altamente innovadora" pero no describe cómo se diferencia de soluciones existentes.
Evaluación de Mérito	Contenido Estratégico	Modelo de negocio débil o poco claro.	No se explica de forma convincente cómo el proyecto generará ingresos de forma sostenible.
Evaluación de Mérito	Contenido Estratégico	Equipo emprendedor insuficiente.	No se demuestra que el equipo posee las competencias técnicas y de gestión para ejecutar el proyecto.

La Tabla 2 categoriza y sintetiza la matriz de errores más frecuentes detectados en las postulaciones, estructurándolos de acuerdo con la fase del proceso en la que provocan el descarte del proyecto (Elegibilidad, Admisibilidad y Evaluación de Mérito). Como ilustra el desglose, una vasta mayoría de las propuestas que fracasan tempranamente no lo hacen por carecer de potencial comercial intrínseco, sino por desalineaciones estrictamente formales o estratégicas, tales como postular a un fondo incorrecto o incumplir directrices administrativas. Esta matriz evidencia que el desconocimiento de las reglas formales actúa como un bloqueador inicial insalvable, justificando técnicamente la necesidad de un sistema algorítmico que verifique estas causales antes del envío definitivo.

Todo esto se sintetiza en el siguiente árbol del problema:

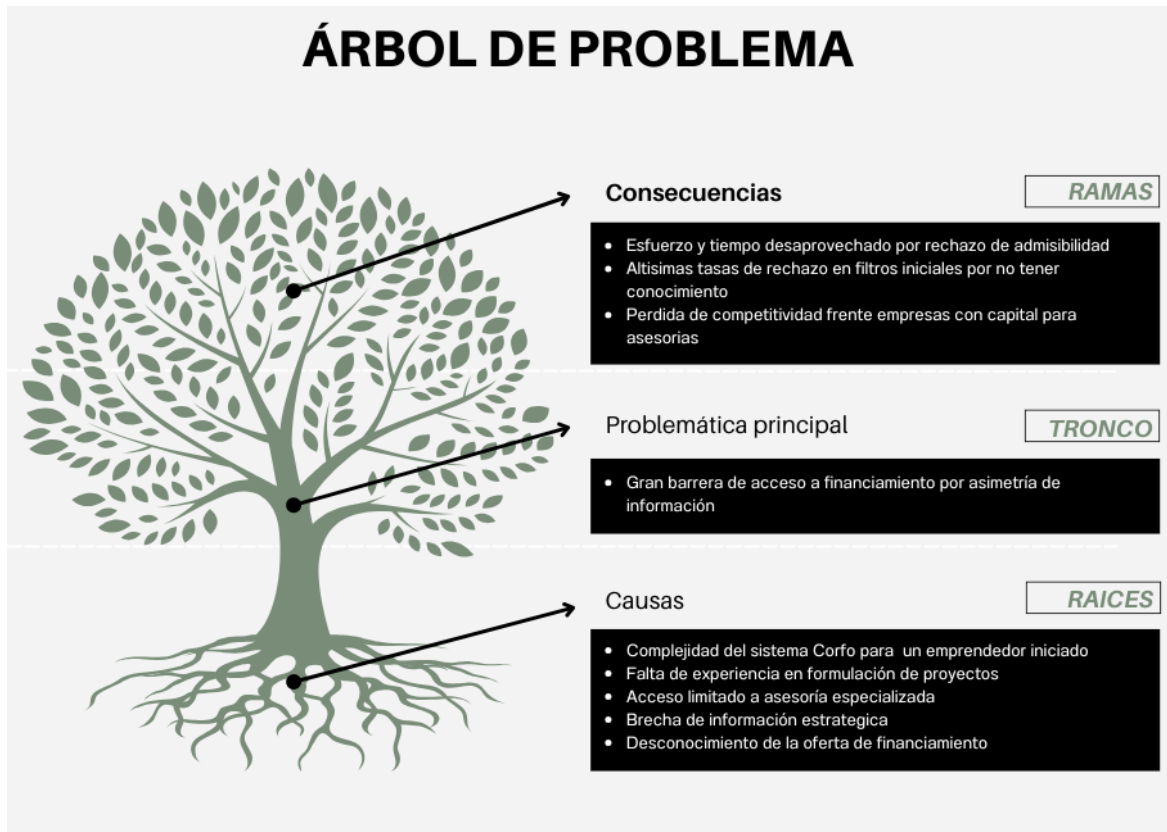


Figura 1: Árbol del problema
Fuente: Elaboración propia

Lo que se ha analizado anteriormente del árbol del problema se sintetiza en que: Tomando la perspectiva del postulante, la gran barrera de acceso al financiamiento para PYMEs, generada por una profunda asimetría de información en el ecosistema CORFO, se fundamenta en diversas causas. En primer lugar, la complejidad inherente al ecosistema con su multiplicidad de programas, requisitos y procesos, resulta abrumadora para un emprendedor que se inicia en la búsqueda de fondos, dificultando la identificación del instrumento adecuado. A esto se suma la falta de experiencia en la formulación de proyectos, donde las PYMEs a menudo carecen de las habilidades metodológicas para traducir sus ideas en propuestas sólidas y competitivas según los estándares exigidos. El acceso limitado a una asesoría especializada que pueda guiar este proceso se convierte en otra causa fundamental, ya sea por los costos prohibitivos de las consultoras privadas o por las exigentes condiciones de participación de las aceleradoras, que no siempre son viables en etapas tempranas. Asimismo, la brecha de información estratégica dificulta el acceso a datos cruciales para fundamentar la viabilidad de los proyectos, debilitando las postulaciones desde su base. Finalmente, el desconocimiento generalizado de la vasta oferta de financiamiento, que lleva a las PYMEs a invertir tiempo valioso en convocatorias erróneas, agrava el problema central.

Como consecuencia directa de esta barrera de acceso, se observan efectos perjudiciales significativos para el ecosistema emprendedor. Las altísimas tasas de rechazo en los filtros formales iniciales son una manifestación empírica de esta asimetría, impidiendo que proyectos innovadores alcancen la instancia donde verdaderamente se evalúa su mérito técnico y comercial. Esto conduce a un esfuerzo y tiempo desaprovechado por parte de los emprendedores, quienes invierten recursos escasos en procesos fallidos, distrayéndose de la operación y el crecimiento de sus negocios. En última instancia, esta dinámica genera un sesgo de selección estructural: considerando que más del 80% de los proyectos son descartados en la etapa de admisibilidad, el avance inicial de una postulación termina condicionado por la capacidad administrativa y financiera de la empresa para costear o formular correctamente los requerimientos. Esto margina tempranamente a iniciativas que, al carecer de estas herramientas metodológicas, no logran superar la estricta barrera formal para que su mérito real sea evaluado.

1.5 Objetivos y alcances de la memoria

Como respuesta directa a la asimetría de información y a las altas tasas de rechazo inicial descritas, este trabajo establece una hoja de ruta tecnológica para mitigar dichas barreras. A continuación, se detallan los objetivos generales y específicos que guiarán el desarrollo y validación del sistema 'Admisivo', junto con los alcances que delimitan funcional y operativamente la solución propuesta.

1.5.1 Objetivo general

Implementar y evaluar un sistema de apoyo computacional, basado en inteligencia artificial, que permita a las PYMES optimizar el **cumplimiento de criterios de admisibilidad técnica** de sus postulaciones para fondos CORFO, democratizando el acceso a información estratégica, informando sobre puntos clave y asistencia experta para optimizar su proceso de postulación.

1.5.2 Objetivos específicos

- Analizar en profundidad los desafíos, brechas de información y necesidades de las PYMES durante el proceso de formulación de postulaciones a CORFO, identificando patrones que afectan la calidad de las propuestas.
- Diseñar una plataforma tecnológica que facilite la estructuración, revisión y mejora de postulaciones de manera escalable y sistemática.
- Desarrollar un módulo de búsqueda inteligente que centralice y clasifique las convocatorias vigentes según el perfil y necesidades de cada empresa.
- Implementar un motor de asistencia basado en IA que genere retroalimentación en tiempo real para mejorar la coherencia y el cumplimiento de los puntos clave de la postulación.
- Validar la precisión técnica y la fiabilidad de la solución implementada mediante un análisis comparativo basado en juicio de experto, utilizando un conjunto de casos de prueba controlados para medir la capacidad de detección de criterios de admisibilidad.

1.5.3 Alcances y Limitaciones

- Alcance Funcional: El sistema actuará estrictamente como una herramienta de apoyo a la toma de decisiones y mejora de la redacción. No realizará el envío automático de la postulación hacia las plataformas del Estado, ni garantizará la adjudicación del financiamiento.
- Dominio de Datos: La extracción de bases legales y la evaluación técnica se limitarán exclusivamente al ecosistema de financiamiento de CORFO, excluyendo otros instrumentos públicos (como Sercotec o FOSIS) o fondos de Venture Capital privados.
- Alcance Tecnológico: El desarrollo se enmarcará en la construcción de un Producto Mínimo Viable (PMV) web, utilizando integraciones a través de API con modelos fundacionales de IA generativa (como Google Gemini), sin contemplar el entrenamiento o *fine-tuning* de un modelo fundacional propio desde cero.

CAPÍTULO 2: MARCO CONCEPTUAL

Este capítulo establece los fundamentos teóricos y tecnológicos sobre los cuales se desarrolla el sistema de apoyo para postulaciones a CORFO. Se abordan los conceptos clave relacionados con los modelos de lenguaje grande (*LLMs*), las particularidades de las postulaciones a CORFO, y las técnicas de inteligencia artificial necesarias para implementar una asistencia efectiva. Se define como alcance de la memoria solo el ecosistema CORFO.

2.1 Análisis y Estructura de Postulaciones a CORFO

Para que un sistema de IA pueda asistir eficazmente en la postulación a fondos CORFO, es fundamental comprender la estructura típica de estas postulaciones, los criterios de evaluación empleados y los desafíos comunes que enfrentan los postulantes. Esta comprensión se basa tanto en la guía general del proceso de postulación como en el análisis de bases específicas de convocatorias, considerando el programa Scale ME apoyado por CORFO [6] y los datos de las convocatorias de InnovaChile.

2.1.1 El Postulante: Perfil de la PYME Innovadora en Chile

Para diseñar una herramienta de apoyo efectiva, es crucial comprender el perfil del usuario objetivo. Lejos de ser grandes corporaciones con equipos dedicados, el universo de postulantes a las líneas de innovación de CORFO está compuesto mayoritariamente por Pequeñas y Medianas Empresas. Los datos de las convocatorias de InnovaChile en 2023 ofrecen una caracterización detallada de este perfil.[5]

- **Tamaño de la Empresa:** El ecosistema postulante es predominantemente Mipyme. El **93,9%** de todos los proyectos presentados en 2023 provinieron de micro, pequeñas y medianas empresas. Esto subraya la importancia de una solución accesible y diseñada para equipos con recursos limitados.
- **Antigüedad y Experiencia:** Se observa una alta concentración de empresas jóvenes. El **47,7%** de las empresas postulantes tenían entre 0 y 5 años de antigüedad desde su inicio de actividades. Esto sugiere una alta probabilidad de que se enfrenten por primera vez a la complejidad de una postulación estatal. De hecho, los datos de recurrencia lo confirman: el **39,4%** de las empresas que postularon en 2023 nunca lo habían hecho antes (en el período 2010-2022).
- **Experiencia en Adjudicación:** La falta de experiencia no es solo en la postulación, sino también en la gestión de fondos. Entre las empresas que lograron adjudicarse un subsidio en 2023, una mayoría del **56% eran "nuevos beneficiarios"**, es decir, nunca habían recibido un subsidio de InnovaChile. Esto refuerza la idea de que cada proceso es un desafío significativo, sin necesariamente contar con un historial de éxito previo.
- **Sectores Económicos:** Si bien las postulaciones provienen de una amplia gama de sectores económicos, los datos actualizados a 2023 revelan que la mayor

concentración de iniciativas se agrupa en **Tecnologías de la Información (278 proyectos)**, **Servicios de Ingeniería (251 proyectos)** y el sector de **Alimentos, exceptuando el vitivinícola (250 proyectos)**. Esta distribución cuantitativa ratifica que, aunque la demanda por financiamiento es transversal a la industria, existe una predominancia significativa y sostenida de sectores intensivos en conocimiento y desarrollo tecnológico.

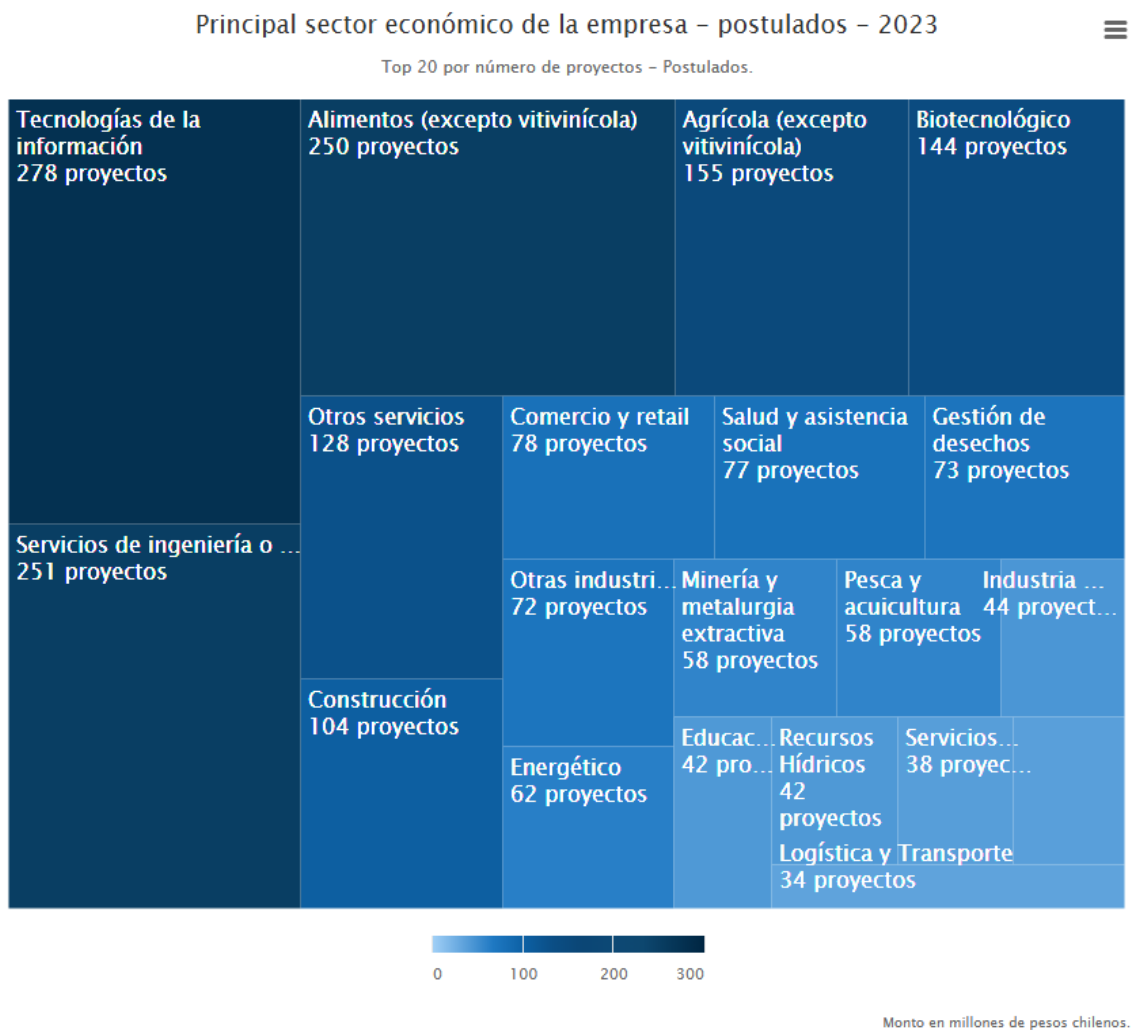


Figura 2: Sectores económicos
Fuente: Data Innovación 2023

2.1.2 Misión y Prioridades Estratégicas de CORFO

CORFO tiene como misión mejorar la competitividad y diversificación productiva de Chile fomentando la inversión, la innovación y el emprendimiento. Sus objetivos estratégicos incluyen impulsar la innovación sostenible, la digitalización, fortalecer capacidades de I+D (especialmente en PYMEs), y promover el desarrollo regional. Alinear las propuestas con estas prioridades (innovación, sostenibilidad, impacto regional, escalabilidad) es crucial para aumentar las posibilidades de éxito. Programas específicos pueden tener focos adicionales, como el apoyo a empresas lideradas por mujeres o la resiliencia al cambio climático. [7]

2.1.3 Componentes Clave de una Postulación

Aunque varía según el programa, una postulación a CORFO típicamente requiere detallar diversos aspectos del proyecto. Basado en la práctica general y ejemplos específicos, estos suelen incluir:

- **Identificación del Problema u Oportunidad:** Descripción clara del problema que se aborda o la oportunidad de mercado detectada, y su relevancia.
- **Solución Propuesta e Innovación:** Descripción del producto, servicio o proceso, destacando su carácter innovador o diferenciador respecto a alternativas existentes. Los datos de InnovaChile muestran que las innovaciones propuestas suelen ser de servicio (46,3%) y de producto (40,3%).
- **Modelo de Negocio:** Cómo se generarán ingresos, perfil del cliente, canales de venta y comunicación. Los modelos B2B (67,7%) y B2C (27,8%) son los más comunes en las postulaciones.
- **Mercado y Escalabilidad:** Análisis del tamaño del mercado, competencia, barreras de entrada y potencial de crecimiento nacional e internacional. Se requiere a menudo un plan de expansión.
- **Equipo Emprendedor:** Descripción del equipo, su experiencia, capacidades, complementariedad y dedicación al proyecto.
- **Estado de Avance:** Hitos alcanzados, validación comercial, ventas, tracción, capital levantado, entre otros...
- **Proyecciones y Presupuesto:** Plan financiero, necesidades de inversión, estructura de costos y proyecciones de ventas. Se debe detallar el uso de los fondos solicitados y el cofinanciamiento aportado por el beneficiario.
- **Impacto y Sostenibilidad:** Descripción del impacto económico, social y/o ambiental esperado. A menudo se requiere una estrategia de operación sostenible.
- **Documentación de Respaldo:** Antecedentes legales de la empresa, CVs del equipo, cartas de apoyo, cotizaciones, formularios específicos (ej. F29 para ventas).

La postulación generalmente se realiza a través de plataformas en línea y requiere completar formularios específicos y adjuntar la documentación requerida.

2.1.4 Criterios de Evaluación

Si bien los criterios exactos y las ponderaciones varían dinámicamente según el instrumento de financiamiento, la evaluación de CORFO (o sus entidades delegadas) se estructura sobre un núcleo de requerimientos transversales. Para efectos de esta memoria, se ha seleccionado el programa Scale ME como arquetipo o modelo base de análisis. La justificación técnica de esta elección radica en que sus bases administrativas representan el estándar promedio de exigencia del ecosistema: condensa los pilares fundamentales de validación de negocios sin descender a la simplificación extrema de los fondos de ideación inicial, ni exigir la hiper-especialización técnica de las líneas de I+D pura (como el programa Crea y Valida). Por lo tanto, este instrumento ilustra con alta representatividad los siguientes criterios comunes exigidos a las PYMEs:

- **Relevancia del Problema/Oportunidad:** Magnitud y pertinencia del problema abordado.
- **Innovación y Propuesta de Valor:** Grado de diferenciación y valor agregado de la solución.
- **Escalabilidad y Modelo de Negocio:** Potencial de crecimiento acelerado, viabilidad comercial, plan de expansión.
- **Equipo:** Capacidades, experiencia, complementariedad y compromiso del equipo emprendedor.
- **Impacto y Sostenibilidad:** Contribución al desarrollo económico, social y/o ambiental; viabilidad a largo plazo; alineación con focos estratégicos (ej. cambio climático, equidad de género).
- **Viabilidad Técnica y Financiera:** Coherencia del presupuesto, realismo de las proyecciones, capacidad de ejecución.

La evaluación integral de estos fondos suele implicar una rigurosa revisión de admisibilidad (cumplimiento de requisitos formales y de elegibilidad) seguida de la evaluación técnica y financiera descrita, culminando a menudo en instancias de defensa oral frente a un comité evaluador o "*pitch*". [6]

2.1.5 Desafíos y Complejidades en la Postulación

El proceso de postulación presenta varios desafíos conceptuales y prácticos para las PYMEs:

- **Identificación del Programa Adecuado:** Navegar la diversidad de programas y filtros (perfil, etapa, necesidad, ventas, territorio) para encontrar el más pertinente.
- **Cumplimiento de Requisitos Específicos:** Cada programa tiene criterios de elegibilidad detallados (antigüedad, nivel de ventas, personalidad jurídica, foco temático, historial de financiamiento previo) que deben cumplirse estrictamente.
- **Formulación de la Propuesta:** Articular claramente todos los componentes requeridos, asegurando coherencia interna y alineación con los criterios de evaluación.
- **Requisitos de Cofinanciamiento:** Demostrar la capacidad de aportar la contrapartida exigida (que varía según tamaño de empresa o si es liderada por mujeres), diferenciando aportes pecuniarios y preexistentes.
- **Gestión de Plazos y Documentación:** Cumplir estrictamente con las fechas límite y reunir toda la documentación solicitada en el formato correcto.
- **Navegación de Normativas:** Comprender y cumplir con las Bases Administrativas Generales de CORFO, manuales de rendición, y otras normativas aplicables.

Estos desafíos subrayan la necesidad de un apoyo estructurado que guíe a los postulantes a través de la complejidad del proceso y les ayude a formular propuestas de alta calidad.

2.2 Herramientas Tecnológicas para la Generación de Conocimiento

Para abordar el desafío de procesar grandes volúmenes de texto no estructurado (como bases administrativas) y asistir en la formulación estratégica de proyectos, el estado del arte en ciencias de la computación ofrece diversos paradigmas de procesamiento y generación de conocimiento. Entre las alternativas tecnológicas más consolidadas se encuentran:

- **Sistemas Basados en Reglas y Expresiones Regulares:** Históricamente utilizados para la extracción de información determinística. Son altamente precisos para identificar patrones predefinidos (como montos, RUTs o fechas clave), pero carecen de la flexibilidad semántica necesaria para interpretar el contexto de un documento legal complejo o la redacción de una propuesta de valor.
- **Procesamiento de Lenguaje Natural (PLN) Tradicional y *Machine Learning*:** Utilizan modelos estadísticos (como *TF-IDF*) o arquitecturas de clasificación estándar. Permiten categorizar documentos o realizar análisis de tópicos, pero no poseen la capacidad de generar texto nuevo, articular retroalimentación dinámica o razonar sobre la coherencia cruzada de un formulario.
- **Inteligencia Artificial Generativa y Modelos Fundacionales:** El paradigma más reciente, capaz de comprender el contexto profundo, realizar inferencias lógicas complejas y generar contenido original. Debido a la necesidad de articular propuestas cualitativas y analizar normativas extensas en lenguaje natural, este enfoque se posiciona como el más apto para tareas de asistencia en redacción y auditoría de admisibilidad.

Dentro de este último paradigma, destacan los Modelos de Lenguaje Grandes (*LLMs*).

2.2.1 Descripción General de los *LLMs*

Los Modelos de Lenguaje Grandes (*LLMs*, por sus siglas en inglés) representan el estado del arte en inteligencia artificial generativa aplicada al procesamiento de lenguaje natural (PLN). A nivel estructural, estos modelos abandonan las redes neuronales secuenciales clásicas para fundamentarse en la arquitectura *Transformer*, la cual procesa secuencias de texto de manera paralela [8].

El núcleo técnico que dota a los *LLMs* de su alta capacidad de comprensión es el mecanismo de autoatención (*Self-Attention*). Este algoritmo permite que la red neuronal calcule y pondere matemáticamente la relación semántica y sintáctica entre todos los *tokens* de un texto simultáneamente, independientemente de la distancia física entre ellos. En el contexto del análisis documental, esta característica es la que permite a los modelos mantener la coherencia a lo largo de textos extensos, conectando dependencias y referencias cruzadas entre distintas secciones de un mismo corpus.

El desarrollo de un LLM contemporáneo consta típicamente de dos fases principales. En la fase de preentrenamiento no supervisado, el modelo ingiere volúmenes masivos de datos textuales para aprender la estructura estadística del lenguaje, optimizando su capacidad para predecir el siguiente *token* en una secuencia. Sin embargo, para que el modelo transicione de un generador probabilístico de texto a un sistema capaz de seguir directrices complejas, se somete a una fase de afinamiento o alineación, frecuentemente a través de *Instruction Tuning* y aprendizaje por refuerzo a partir de retroalimentación humana (*RLHF*). Este proceso es el que habilita al modelo para interpretar instrucciones condicionadas, adoptar roles lógicos específicos y estructurar sus respuestas en formatos estrictos, características que los hacen aptos para tareas analíticas y deductivas de alta precisión.

2.2.2 Conceptos Técnicos Fundamentales: *Tokenización* y Ventana de Contexto

Para comprender la viabilidad de los *LLMs* en el procesamiento de documentos legales extensos, y poder comparar objetivamente las capacidades de distintos modelos, es imperativo definir dos conceptos técnicos que rigen su funcionamiento:

- ***Tokenización***: Es el proceso mediante el cual el modelo descompone el texto de entrada en unidades léxicas mínimas llamadas "tokens" (que pueden ser palabras enteras, sílabas o caracteres). Como referencia estándar de la industria algorítmica, 1.000 tokens equivalen aproximadamente a 750 palabras.
- **Ventana de Contexto**: Se define como el número máximo de tokens que el modelo puede retener y procesar en la memoria de trabajo durante una única iteración (es decir, la suma de la entrada del usuario más la salida generada). En el procesamiento de bases de financiamiento público, una ventana de contexto amplia es un requisito técnico crítico, ya que evita el truncamiento del documento y previene la pérdida de excepciones o anexos legales vinculantes.

2.2.3 Capacidades de los *LLMs* Relevantes para la Asistencia en Postulaciones

En el contexto teórico de la evaluación de proyectos, estas arquitecturas ofrecen capacidades altamente pertinentes:

- **Procesamiento y Comprensión Semántica**: Capacidad de analizar secciones extensas de una postulación, identificar patrones o detectar carencias frente a una rúbrica predefinida, generando sugerencias de mejora.
- **Adaptabilidad Contextual**: Habilidad algorítmica para ajustar el tono, formato y contenido de una salida en base a un contexto inyectado mediante instrucciones (*prompts*), como las bases específicas de una convocatoria o el perfil de la PYME.
- **Síntesis y Análisis Cruzado**: Potencial para resumir documentos normativos extensos y analizar la coherencia lógica cruzada entre distintas secciones de un

formulario (por ejemplo, asegurar que el presupuesto calce con las actividades propuestas).

2.2.4 Limitaciones y Desafíos Inherentes

Es crucial reconocer las limitaciones intrínsecas de los *LLMs* al aplicarlos a dominios de alta rigurosidad técnica:

- **Dependencia de la Calidad del Input (*Prompting*):** La utilidad de la salida depende críticamente de la claridad, restricciones y completitud del contexto proporcionado. Entradas vagas generan respuestas algorítmicas genéricas.
- **Conocimiento Estático y “Alucinaciones”:** Los *LLMs* carecen de acceso autónomo a información en tiempo real post entrenamiento. Pueden generar información sintácticamente correcta pero factualmente falsa (“alucinaciones”). Esto hace imperativa la inyección de la fuente de verdad (como las bases del concurso) en tiempo de inferencia.
- **Propagación de Sesgos:** Pueden reflejar sesgos presentes en sus datos de entrenamiento, lo que requiere un diseño cuidadoso para mitigar resultados indeseados.
- **Falta de Razonamiento de Negocio Real:** Aunque la arquitectura simula el razonamiento deductivo, los modelos no comprenden intrínsecamente la viabilidad comercial o el grado de innovación real de un negocio. Son herramientas probabilísticas que complementan, pero no reemplazan, el juicio experto humano.

2.2.5 Análisis del Estado del Arte en Ecosistemas de IA

Para la implementación de soluciones algorítmicas modernas, la industria actual está dominada por ecosistemas que ofrecen *APIs* comerciales de alto rendimiento. A continuación, se comparan sus modelos de frontera, con especial énfasis en su capacidad de retención de memoria (ventana de contexto) y razonamiento:

A. Ecosistema OpenAI: Pioneros en la masificación de los *LLMs*, ofrecen modelos iterativos orientados tanto al usuario final como a soluciones empresariales, destacando por sus altas capacidades de razonamiento deductivo [9]:

- **GPT-4o / GPT-4o-mini:** Modelos de alta capacidad multimodal. Ambos poseen una ventana de contexto estandarizada de 128.000 **tokens** (aprox. 300 páginas de texto). Son adecuados para tareas de inferencia estándar, aunque su contexto puede verse limitado al cruzar múltiples PDFs legales simultáneamente.
- **Serie OpenAI o1 (y o1-mini):** Evolución algorítmica diseñada para tareas de alta complejidad. Integran aprendizaje por refuerzo para generar una "Cadena de Pensamiento" (*Chain-of-Thought*) interna antes de responder. Soportan una

ventana de contexto ampliada a 200.000 **tokens** y destacan en resolución de problemas lógicos matemáticos y de programación.

B. Ecosistema Google: La familia de modelos Gemini se distingue por su arquitectura nativamente multimodal y un diseño ingenieril enfocado en integrarse orgánicamente con aplicaciones de alto rendimiento [10]. Estos modelos buscan expandir drásticamente los límites de la memoria a corto plazo, utilizando técnicas como *Ring Attention*:

- **Gemini 2.0 Pro / 1.5 Pro:** Modelos de escala mayor, destacados en la industria por poseer una ventana de contexto masiva de 2.000.000 **de tokens**. Esta capacidad teórica es la que permite la ingesta de bibliotecas de documentos legales íntegros sin recurrir a técnicas de fragmentación (*chunking*).
- **Gemini 2.5 Flash:** Arquitectura de última generación optimizada para baja latencia y alta concurrencia. Soporta una ventana de contexto masiva que oscila entre 1.000.000 y 2.000.000 **de tokens**, presentando ventajas comparativas inigualables en la relación costo-beneficio para inferencias repetitivas, interacciones en tiempo real o análisis de corpus documentales completos en milisegundos.

2.3 Ingeniería de *Prompts*

La efectividad de un *LLM* depende en gran medida de cómo se le instruye. La ingeniería de *prompts* es el proceso de diseñar y refinar las entradas para obtener las salidas deseadas. La aplicación efectiva de un *LLM*, no se trata simplemente de hacer preguntas, sino de construir un sistema de *prompts* estructurado que guíe al *LLM* a través de tareas de análisis complejas.

A continuación, se describen las técnicas fundamentales documentadas en la literatura actual:

- **Prompting de Rol o "Persona":** Esta técnica, también denominada asignación de "Persona", consiste en instruir al modelo para que adopte una identidad o perfil específico antes de ejecutar una tarea. Al establecer un marco de referencia profesional o técnico (por ejemplo, el de un experto en una materia determinada), se condiciona el tono, el vocabulario y la profundidad del análisis. La literatura sugiere que esta contextualización permite que el modelo priorice patrones de respuesta alineados con el dominio de conocimiento asignado, mejorando la relevancia de la salida.
- **Prompting de Cadena de Pensamiento (Chain-of-Thought - CoT):** El método de Cadena de Pensamiento se utiliza para abordar tareas que requieren razonamiento lógico o descomposición de problemas. Consiste en forzar al modelo a generar pasos intermedios de razonamiento antes de entregar un resultado final. En lugar de una respuesta directa, el modelo articula una secuencia de premisas y deducciones. Esta técnica ha demostrado ser particularmente efectiva para reducir errores en tareas de aritmética, razonamiento simbólico y evaluaciones de coherencia interna en textos extensos [11].

- **Clasificación de *Prompts* por Funcionalidad:** En arquitecturas de software complejas, la ingeniería de *prompts* evoluciona hacia un enfoque sistemático donde las instrucciones se categorizan según su funcionalidad operativa. Las categorías principales identificadas en la práctica estándar incluyen:
 - ***Prompts* de Validación y Verificación:** Orientados a contrastar un texto de entrada frente a un conjunto de reglas o requisitos explícitos. Se centran en la detección de cumplimiento normativo o técnico.
 - ***Prompts* de Análisis Crítico:** Diseñados para evaluar la calidad, robustez y estructura de un argumento o contenido, identificando vacíos de información o inconsistencias lógicas.
 - ***Prompts* de Síntesis y Generación:** Enfocados en la creación de contenido nuevo a partir de puntos clave o datos estructurados, facilitando la redacción inicial y la organización de ideas complejas.

2.4 Generación Aumentada por Recuperación (RAG)

La Generación Aumentada por Recuperación es una arquitectura diseñada para mitigar las limitaciones intrínsecas de los Modelos de Lenguaje Extensos (LLMs), tales como el corte temporal en su conocimiento pre-entrenado y la propensión a generar alucinaciones [12]. Este enfoque permite que el modelo acceda a fuentes de datos externas y actualizadas durante el proceso de inferencia, condicionando su respuesta a un contexto verídico y específico.

2.4.1 Componentes de un Sistema de Recuperación

Un sistema RAG robusto se fundamenta en la creación de un repositorio de conocimiento especializado. A diferencia del conocimiento generalista del modelo, este componente permite integrar:

- **Documentación Técnica y Normativa:** Archivos que rigen procesos específicos o regulaciones vigentes.
- **Manuales y Guías Operativas:** Instrucciones procedimentales que requieren precisión factual.
- **Criterios de Evaluación y Ponderaciones:** Reglas explícitas que definen el éxito o cumplimiento de una tarea en un dominio determinado.

2.4.2 Proceso Técnico de RAG

El flujo operativo estándar de la arquitectura RAG se descompone en tres fases principales:

1. **Indexación y Vectorización:** Los documentos son segmentados en fragmentos (*chunks*) y transformados en vectores numéricos (*embeddings*) mediante modelos de representación semántica. Estos vectores se almacenan en bases de datos vectoriales para permitir búsquedas eficientes.
2. **Recuperación (*Retrieval*):** Ante una consulta, el sistema convierte dicha entrada en un vector y realiza una búsqueda de similitud para extraer los fragmentos de información más relevantes del repositorio.
3. **Aumentación y Generación:** Los fragmentos recuperados se insertan en el *prompt* del LLM. Esto permite que el modelo genere respuestas fundamentadas en evidencia externa, citando fuentes y reduciendo la ambigüedad.

2.5 Extracción de Información y Análisis de Coherencia

El Procesamiento de Lenguaje Natural (PLN) ofrece herramientas críticas para verificar la integridad de documentos complejos. El análisis de coherencia se basa en la capacidad de los algoritmos para identificar y conectar datos aislados dentro de un flujo de texto.

2.5.1 Reconocimiento de Entidades Nombradas (NER)

La técnica de Reconocimiento de Entidades Nombradas (NER) permite identificar y clasificar unidades de información en categorías predefinidas, tales como organizaciones, magnitudes financieras, fechas o roles profesionales. En el análisis de documentos estructurados, el NER facilita:

- **Extracción de Datos Cuantitativos:** Identificación automática de montos, presupuestos y proyecciones.
- **Mapeo de Atributos:** Clasificación de plazos, hitos y perfiles técnicos.
- **Validación Cruzada:** La información extraída permite realizar chequeos lógicos entre distintas secciones de un mismo documento para asegurar la consistencia interna de la información.

2.6 Evaluación de la Calidad de la Asistencia

La validación de sistemas basados en IA requiere de marcos métricos que garanticen la fiabilidad de las sugerencias generadas. La literatura identifica diversos enfoques para medir esta eficacia:

- **Métricas de Calidad de Salida:** Se evalúan dimensiones como la veracidad (ausencia de alucinaciones), la relevancia respecto al contexto y la completitud de la respuesta.
- **Metodologías de evaluación:**
 - **Evaluación Humana:** Juicio de expertos o usuarios finales sobre la utilidad percibida.
 - **LLM-as-judge:** Uso de modelos de lenguaje de orden superior para calificar el desempeño de otros modelos basándose en rúbricas predefinidas.
 - **Benchmarking:** Comparación del rendimiento frente a estándares o casos de referencia aprobados históricamente.

Dentro de la Evaluación Humana, uno de los instrumentos estandarizados más utilizados en la literatura de interacción humano-computador es el *System Usability Scale (SUS)*, desarrollado por John Brooke en 1996 mientras trabajaba en Digital Equipment Corporation (Reino Unido) como una herramienta “*quick and dirty*” rápida y de bajo costo para medir la usabilidad percibida de un sistema [13]. Consiste en un cuestionario de 10 ítems en escala Likert de 5 puntos, que alterna afirmaciones formuladas en sentido positivo y negativo para mitigar el sesgo de aquiescencia, y cuyo procesamiento produce un puntaje único en el rango 0-100. Pese a su simplicidad, SUS se ha consolidado como un estándar de facto de la industria, validado empíricamente en cientos de estudios y sobre miles de sistemas evaluados, lo que permite comparar el resultado de un sistema específico contra un puntaje de referencia agregado (percentil 50 $\approx$ 68 puntos) y contra una escala adjetival de interpretación de “Peor imaginable” a “Mejor imaginable” propuesta por Bangor, Kortum y Miller [14]. Por su bajo costo de aplicación y su validez cruzada, este trabajo adopta SUS como el instrumento principal del eje de Evaluación Humana descrito en el Capítulo 4.

2.7 Consideraciones Éticas y de IA responsable

La implementación de soluciones de IA en entornos profesionales exige el cumplimiento de principios éticos que aseguren la confianza del usuario y la integridad del proceso.

1. **Privacidad y Confidencialidad:** Garantizar que los datos sensibles (estratégicos o financieros) sean tratados bajo protocolos de protección de datos rigurosos, evitando su filtrado hacia los conjuntos de entrenamiento de modelos públicos.
2. **Transparencia (Explicabilidad):** Informar al usuario sobre la naturaleza artificial del interlocutor y proporcionar trazabilidad sobre el origen de las sugerencias, especialmente cuando se utiliza RAG.
3. **Gestión de Sesgos:** Identificar y mitigar sesgos algorítmicos que podrían influir de manera desequilibrada en las recomendaciones o evaluaciones.
4. **Responsabilidad y Expectativas:** Definir claramente los límites de la herramienta, posicionándose como un sistema de apoyo a la decisión humana y no como una autoridad autónoma infalible.

CAPÍTULO 3: PROPUESTA DE SOLUCIÓN

En este capítulo se detalla el diseño, la arquitectura y la implementación del sistema "Admisivo". La solución propuesta consiste en un Producto Mínimo Viable (PMV) de una plataforma web integral que utiliza Inteligencia Artificial Generativa con el objetivo de reducir la asimetría de información y asistir técnicamente a las PYMEs en el proceso de postulación a fondos CORFO. A diferencia de los buscadores tradicionales o los asistentes basados en palabras clave, esta solución implementa un pipeline de procesamiento de lenguaje natural capaz de interpretar tanto las bases administrativas (documentos PDF no estructurados) como la propuesta de valor del emprendedor, realizando un cruce semántico y normativo en tiempo real.

3.1 Visión General y Arquitectura del Sistema (Modelo C4)

Para documentar la arquitectura de software de Admisivo, se ha adoptado el modelo C4, el cual permite una descomposición jerárquica del sistema desde el contexto del negocio hasta la implementación a nivel de código. El sistema se define como una plataforma *full-stack* que combina *web scraping* automatizado, procesamiento masivo de documentos, un motor de *matching* híbrido y generación de formularios dinámicos respaldados por modelos fundacionales de la familia Google Gemini. El modelo específico se resuelve mediante la variable de entorno GEMINI_MODEL, lo que permite alternar entre versiones sin recompilar el servicio; el despliegue productivo actual opera sobre gemini-2.5-flash-lite, priorizando la relación costo-latencia por sobre la capacidad de razonamiento máxima del modelo base.

SISTEMA DE APOYO BASADO EN INTELIGENCIA ARTIFICIAL GENERATIVA PARA MEJORAR LA ADMISIBILIDAD DE POSTULACIONES DE PYMES A FONDOS CORFO

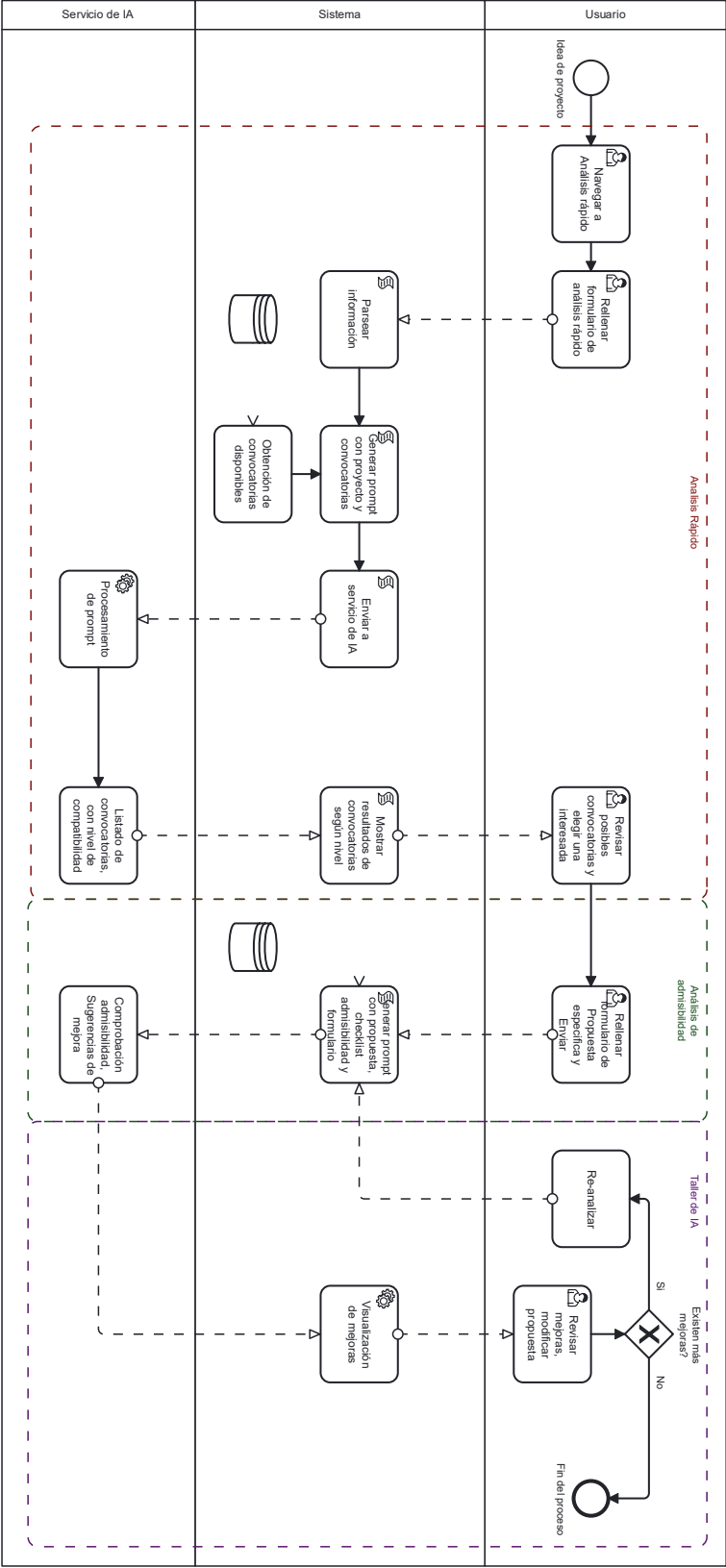


Figura 3: Diagrama de flujo
Fuente: Elaboración propia
Página 38 de 78

3.1.1 Diagrama de Contexto (C4 - Nivel 1)

El contexto del sistema establece las fronteras de Admisivo y sus interacciones con los actores externos. El actor principal es el **Emprendedor (Usuario)**, quien interactúa con la plataforma web para descubrir convocatorias compatibles y recibir asistencia en la estructuración de su propuesta.

Adicionalmente, la gestión de identidad de los usuarios no es resuelta por el propio sistema, sino delegada a un proveedor externo: Firebase Authentication (Google). El Emprendedor se autentica mediante Google Login o correo a través del SDK Web de Firebase en el *frontend*; el *backend*, por su parte, no gestiona contraseñas ni sesiones propias, sino que verifica la validez de los tokens JWT emitidos por Firebase en cada petición protegida. Este servicio se incorpora como un quinto actor externo del sistema.

Para lograr esto, el sistema se integra de manera automatizada con entidades externas:

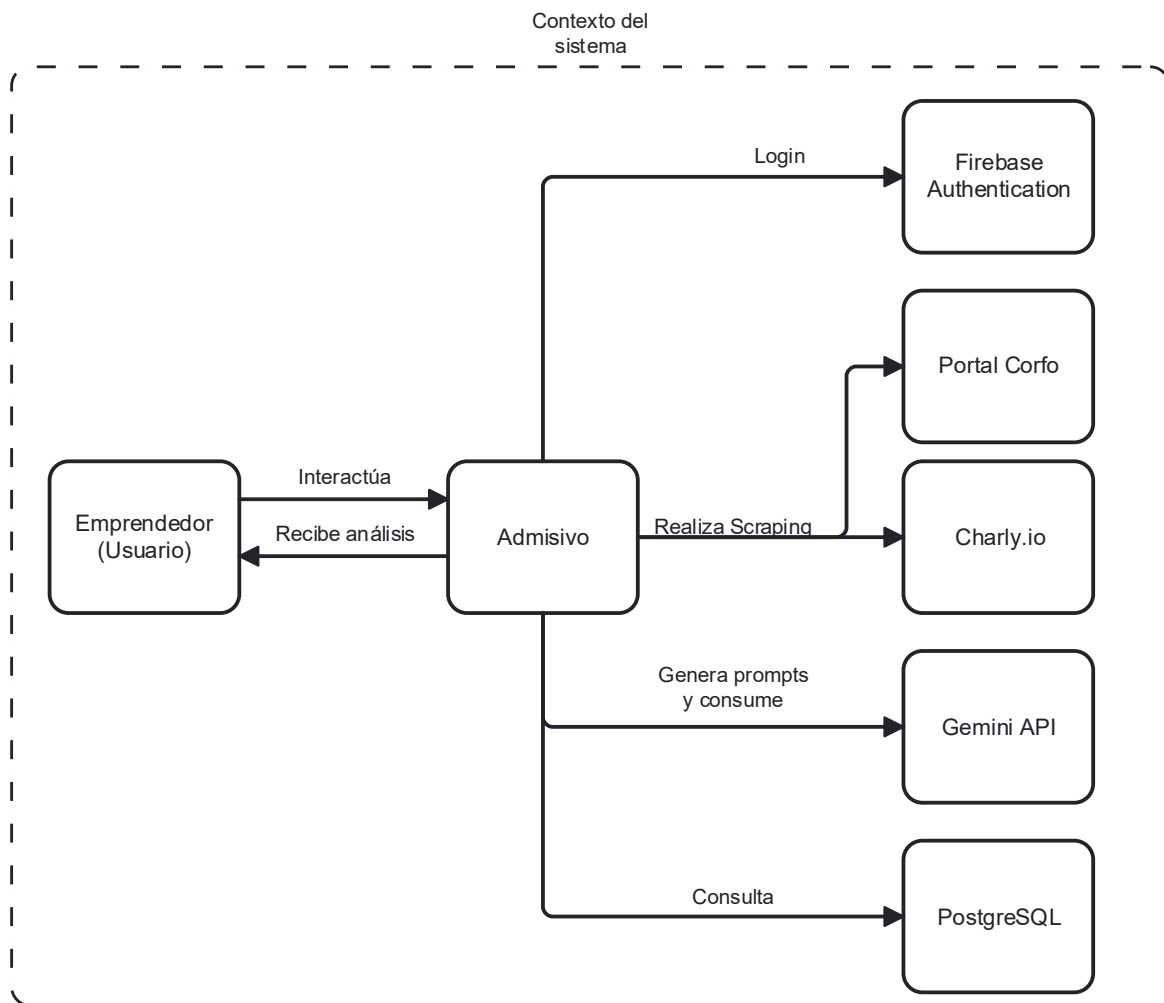


Figura 4: Diagrama de interacciones
Fuente: Elaboración Propia

Tabla 3: Actores y descripciones
Fuente: Elaboración Propia

Elemento	Descripción
Emprendedor (Usuario)	Persona natural o representante de empresa que busca fondos CORFO. Interactúa con el <i>frontend</i> web para describir su proyecto, encontrar convocatorias compatibles y recibir análisis de compatibilidad de alguna en específico.
Portal CORFO	Sitio web gubernamental donde se publican convocatorias. El sistema hace <i>scraping</i> automatizado con <i>Selenium</i> + Clave Única.
Charly.io	Plataforma SaaS donde CORFO aloja parte de sus formularios de postulación. El sistema extrae estructura de formularios incluyendo campos, validaciones y <i>metadata</i> .
Google Gemini API	Servicio de IA generativa de Google, el cual se utiliza para: Generar fichas técnicas, <i>Matchmaking</i> , análisis de admisibilidad y generación de formularios.
PostgreSQL (Railway)	Base de datos relacional gestionada como plugin en Railway. Almacena convocatorias, documentos, usuarios, proyectos y registros de auditoría.
Firebase Authentication (Google)	Proveedor externo de identidad como servicio (IDaaS). Emite y firma tokens JWT de sesión tras un login exitoso (Google o correo). El <i>backend</i> valida estos tokens mediante el SDK Admin de Firebase antes de autorizar operaciones sobre recursos del usuario (proyectos, organizaciones, comentarios).

3.1.2 Arquitectura de Contenedores (C4 - Nivel 2)

El sistema sigue un patrón de despliegue contenerizado operado en la Plataforma as a Service (PaaS) Railway. La separación de responsabilidades se materializa en los siguientes contenedores:

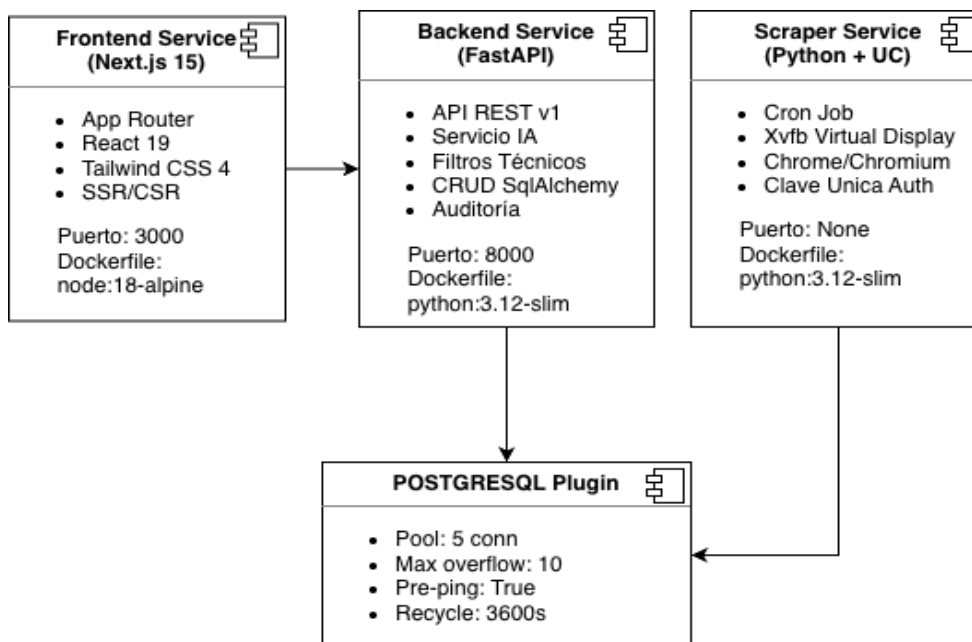


Figura 5: Contenedores
Fuente: Elaboración propia

- **Frontend Service (Next.js 15):** Construido sobre React 19 y Tailwind CSS 4, opera bajo el patrón *App Router*. Maneja estrategias híbridas de renderizado (SSR/CSR) según la ruta (por ejemplo, */analisis_rapido* o */taller*). Utiliza TypeScript 5 para garantizar el tipado estático estricto, interactuando con el *backend* mediante la API nativa *fetch()* y reconstruyendo interfaces complejas dinámicamente a través de componentes como *FormularioDinamico.tsx*.
- **Backend Service (FastAPI):** Representa el núcleo lógico de la aplicación. Desarrollado en Python 3.12 y servido vía Uvicorn, implementa una arquitectura basada en capas (Routers > Services > Infra > CRUD). Expone una API RESTful versionada y gestiona las conexiones concurrentes a la base de datos utilizando SQLAlchemy con un *pool* de conexiones optimizado (tamaño de 5, *max overflow* de 10) para entornos de alta concurrencia. El *Backend Service* integra además el SDK Admin de Firebase para verificar, en cada solicitud autenticada, la validez del token de sesión emitido por el proveedor de identidad externo. Esta dependencia se degrada de forma controlada: si las credenciales de servicio de Firebase no están configuradas en el entorno, el sistema no falla en el arranque, sino que responde con HTTP 503 únicamente en las rutas que requieren identidad de usuario, manteniendo operativas las funcionalidades públicas (búsqueda de convocatorias, *matching*).
- **Scraper Service (Cron Job/Worker):** Contenedor aislado ejecutado en segundo plano dedicado a la ingesta de datos. Para evadir mecanismos *anti-bot* aplicados en plataformas gubernamentales chilenas, implementa Selenium WebDriver junto con *Xvfb (X Virtual Framebuffer)*. Esto permite renderizar Chromium en un *display* virtual, sorteando la detección de navegadores *headless* que habitualmente bloquean el acceso vía Clave Única.

3.1.3 Diagrama de Componentes (C4 - Nivel 3: *Backend*)

Mientras que el Nivel 2 define los contenedores físicos o lógicos del sistema, el Nivel 3 profundiza en la estructura interna del contenedor más complejo y crítico de la solución: el **Backend Service (FastAPI)**.

Para garantizar la mantenibilidad y el principio de responsabilidad única, el *backend* de Admisivo se ha diseñado utilizando un **patrón de arquitectura por capas (Layered Architecture)**. Esta separación aísla la lógica de red, las reglas de negocio, la interacción con servicios externos y la persistencia de datos.

El diagrama de componentes del *backend* se estructura en las siguientes cuatro capas principales:

- **Capa de API (API Layer - Routers):** Es el punto de entrada de las peticiones HTTP HTTP procedentes del *frontend*. Utiliza *Routers* de FastAPI para definir los *endpoints* agrupados por dominio.
 - convocatorias.py: Gestiona la lectura del catálogo de fondos y expone el *endpoint* protegido para gatillar el *scraper*.
 - propuestas.py: Recibe los datos del emprendedor y orquesta las llamadas para el *matching* híbrido y el análisis de admisibilidad.
 - formularios.py: Provee las estructuras JSON necesarias para que el frontend renderice los formularios dinámicos.
- **Capa de Servicios (Service Layer):** Concentra la lógica de negocio central y orquesta las operaciones complejas, siendo independiente del protocolo de transporte (HTTP).
 - ai_service.py: Actúa como un *wrapper* o fachada para la API de Google Gemini. Centraliza la inyección de *prompts*, la configuración del modelo y la normalización de los objetos JSON de salida.
 - filter_service.py: Implementa la lógica determinística de la plataforma, calculando el cumplimiento de filtros duros y asignando el puntaje de los filtros blandos sin incurrir en costos de inferencia de IA.
 - audit_service.py: Servicio transversal encargado de capturar la telemetría, tiempos de respuesta y datos de entrada/salida para el registro de auditoría.
- **Capa de Infraestructura (Infrastructure Layer):** Encapsula las interacciones con sistemas externos asíncronos o no estructurados.
 - scraper_optimizado.py: Módulo robusto encargado de la navegación automatizada (Selenium), evasión de mecanismos *anti-bot* (Xvfb) y extracción de metadatos desde el portal de CORFO.
 - analizador_formularios.py: Módulo especializado en interactuar con la plataforma Charly.io, aplicando heurísticas de espera (Estabilidad DOM) para extraer los árboles de campos y validaciones de los formularios SPA.
- **Capa de Datos (Data Layer - CRUD & Models):** Abstrae la comunicación con el motor PostgreSQL.
 - **Módulos CRUD:** Centralizan las consultas SQL a través del ORM SQLAlchemy (ej. crud_convocatoria.py, crud_usuario.py), previniendo inyecciones SQL y manejando las transacciones de la sesión.
 - **Modelos ORM:** Representaciones en objetos de las tablas de la base de datos, mapeando campos complejos como JSON en variables de tipo TEXT a nivel de base de datos para asegurar flexibilidad estructural.

5. Módulo Transversal de Autenticación: A diferencia de las cuatro capas anteriores, que se ordenan verticalmente, este componente opera de forma transversal (*cross-cutting*) sobre la Capa de API. `app/core/firebase_admin.py` encapsula la verificación de tokens JWT contra el proyecto de Firebase configurado, mientras que `app/api/dependencies/auth.py` expone esta verificación como una dependencia inyectable de FastAPI (`get_current_user`), consumida por los routers que requieren identidad de usuario (`usuarios.py`, `propuestas.py`, `organizaciones.py`, `mentor.py`, `comentarios.py`, entre otros).

3.1.4 Mecanismos de Robustez del Scraper y Autenticación ante ClaveÚnica

Para asegurar la continuidad operativa del sistema frente a las protecciones anti-bot de los portales gubernamentales, se implementó una lógica de navegación avanzada. El servicio de *scraping* gestiona la sesión mediante un flujo que emula el comportamiento humano para evitar bloqueos por IP o detección de *headless browsers*.

Este flujo se basa en dos pilares:

- **Emulación de Entorno:** Se utiliza un **X Virtual Framebuffer (Xvfb)** para ejecutar Chromium en un *display* virtual, lo que permite sortear la detección de navegadores sin interfaz gráfica que bloquean el acceso vía Clave Única.
- **Manejo de Timeouts:** El sistema implementa esperas explícitas basadas en la visibilidad de elementos clave del DOM, evitando errores por carga asíncrona.

El siguiente algoritmo asegura que los scripts de la SPA (*Single Page Application*) hayan terminado de inyectar los campos condicionales y validaciones técnicas antes de proceder con la extracción de datos:

```
Mientras (tiempo_actual - inicio) < timeout:
    conteo_campos = contar_elementos_visibles("#seccionRender")

    Si conteo_campos == conteo_previo y conteo_campos > 0:
        conteo_estabilidad += 1
        Si conteo_estabilidad >= 3: # 3 comprobaciones exitosas cada 0.5s
            Confirmar_Carga_Completa()
            Salir_del_Bucle()
    Sino:
        conteo_estabilidad = 0

    conteo_previo = conteo_campos
    Esperar (0.5 segundos)
```

Figura 6: Pseudocódigo de estabilización del DOM

Fuente: Elaboración Propia

3.1.5 Diseño de Prompts para el Matching Híbrido

La ingeniería de *prompts* en el motor de emparejamiento se estructura para reducir la "alucinación" y forzar un análisis determinístico basado en los datos extraídos. Para lograr esto en la Capa 2 del *matching*, el sistema inyecta un *prompt* dinámico al modelo Gemini

2.5 Flash siguiendo tres pilares técnicos:

- **Variables de Entrada:** Se provee la propuesta del emprendedor (Título, Problema, Solución, Diferenciador, TRL, Sector y Región) contrastada contra el JSON de la "Ficha Técnica" de las convocatorias finalistas.
- **Restricciones de Salida:** Se impone el uso obligatorio de *JSON Mode* (`response_mime_type: "application/json"`) para eliminar saludos o texto narrativo, garantizando la integridad estructural que el *backend* requiere.
- **Control de Temperatura:** Para el emparejamiento factual, se utiliza una temperatura de 0.1 a 0.2, priorizando la exactitud sobre la creatividad algorítmica

3.2 Pipeline de Procesamiento de Documentos: El Paradigma "In-Prompt"

Una de las decisiones arquitectónicas más significativas de Admisivo es la desviación deliberada de la arquitectura RAG (*Retrieval-Augmented Generation*) clásica, optando por una inyección de **contexto completo in-prompt**.

3.2.1 Justificación Técnica de la Ausencia de RAG

Los sistemas RAG tradicionales dividen los documentos en fragmentos (*chunks*), calculan sus *embeddings* vectoriales y los almacenan en bases de datos como Pinecone o ChromaDB. Durante una consulta, se recuperan por similitud coseno los fragmentos más relevantes para alimentar al LLM.

Sin embargo, para el dominio específico de CORFO, este enfoque presenta fallas críticas:

- **Riesgo de pérdida de contexto legal:** Los sistemas RAG tradicionales fundamentan su operación en la división de los documentos fuente en fragmentos (*chunks*), de los cuales el sistema recupera solo aquellos con mayor similitud semántica. Sin embargo, en el análisis de las bases administrativas de CORFO, esta técnica provoca una fragmentación de la integridad normativa. Debido a que estos documentos poseen fuertes dependencias cruzadas donde un requerimiento expuesto en el cuerpo principal puede estar sujeto a excepciones o definiciones críticas detalladas en anexos o cláusulas aisladas, el *chunking* segmenta la información de forma que el modelo pierde la trazabilidad de estas dependencias legales, evaluando así la normativa de manera descontextualizada.
- **Viabilidad de la Ventana de Contexto:** Las bases de CORFO oscilan habitualmente entre 10 y 80 páginas PDF, lo que se traduce en un máximo de 300.000 caracteres. El modelo seleccionado, Gemini 2.5 Flash, posee una ventana de contexto masiva de aproximadamente 1 millón de tokens.

- **Simplicidad y Precisión:** Al proveer el documento íntegro en cada inferencia, se elimina la latencia y la complejidad de mantener una base de datos vectorial, reduciendo la tasa de "alucinaciones por recuperación" a cero, garantizando que el modelo tenga acceso absoluto a la fuente de verdad.

3.2.2 Fases del Pipeline de Extracción y Procesamiento

El procesamiento de la información en Admisivo no ocurre en tiempo real durante la consulta del usuario, sino que sigue un flujo secuencial automatizado en *background* para preparar y estructurar los datos. Este *pipeline* se divide en cuatro fases críticas:

3.2.2.1 Fase de Adquisición

El proceso inicia superando la barrera de autenticación de ClaveÚnica. Para lograr esto sin ser detectado como tráfico automatizado (bot), el contenedor del *scraper* utiliza Selenium WebDriver en conjunto con Xvfb (*X Virtual Framebuffer*). Esto permite ejecutar el navegador Chromium en un entorno virtualizado "visible", sorteando los bloqueos estrictos de los portales gubernamentales. Una vez autenticado, el sistema navega al listado de convocatorias y utiliza la librería [BeautifulSoup4](#) para parsear el DOM (*Document Object Model*). En esta instancia se extraen metadatos estructurados fundamentales como títulos, fechas de apertura y cierre (soportando múltiples formatos de fecha chilenos), descripciones, requisitos del postulante y el calendario detallado de las etapas de postulación.

3.2.2.2 Fase de Extracción de Texto Plano:

Tras identificar las convocatorias, el sistema localiza e itera sobre los enlaces a las bases administrativas y anexos en formato PDF. Utilizando la librería *fitz* (PyMuPDF), el sistema descarga y extrae el contenido página por página. En esta etapa se aplica una rutina de preprocesamiento de texto que normaliza los espacios en blanco y elimina caracteres de control problemáticos (hexadecimales `\x00-\x1F`). Un aspecto crucial de esta fase es la ausencia deliberada de truncamiento o *chunking*; el texto se consolida en una única cadena extensa (generalmente entre 20.000 y 300.000 caracteres) que se almacena temporalmente para ser inyectada de forma íntegra en el *prompt* del modelo de lenguaje, preservando el contexto legal completo.

3.2.2.3 Fase de Extracción de Formularios:

Dado que los formularios de postulación de CORFO están alojados en la plataforma de terceros Charly.io y contruidos como *Single Page Applications* (SPA) altamente dinámicas, la extracción directa del HTML estático es inviable. Para solucionar este desafío de *web scraping* dinámico, el módulo AnalizadorFormulariosCorfo implementa un algoritmo heurístico denominado "3-Check Verification". Este algoritmo monitorea el DOM y cuenta los campos de entrada visibles cada 0.5 segundos; solo cuando este conteo permanece idéntico durante tres comprobaciones consecutivas, el sistema confirma que los *scripts*

asíncronos han terminado de cargar la interfaz. Posteriormente, el *driver* interactúa de forma secuencial con los componentes visuales tipo acordeón (*Bootstrap Collapse*), forzando el renderizado de campos condicionales y extrayendo metadatos técnicos críticos como el identificador del campo, la etiqueta, el tipo de dato y las validaciones exigidas.

3.2.2.4 Fase de Análisis y Pre-computación con IA

Con el texto bruto de las bases y la estructura del formulario extraídos, el sistema realiza una pre-computación para optimizar los tiempos de respuesta futuros. Se ejecutan llamadas automatizadas a la API de Gemini para generar dos artefactos en formato JSON: la **Ficha Técnica**, que estandariza variables cuantitativas como el porcentaje de cofinanciamiento y sectores económicos válidos; y el **Checklist de Filtros**, que destila las bases en una lista concisa de criterios de admisibilidad eliminatorios. Estos objetos JSON persisten en la base de datos PostgreSQL, quedando indexados y listos para la fase de inferencia en tiempo real cuando el usuario ingrese su propuesta de proyecto.

3.3 Modelo de Datos Relacional

Para soportar las complejas operaciones de filtrado y el almacenamiento de objetos JSON heterogéneos generados por la IA, se adoptó PostgreSQL como motor de base de datos principal, interactuando mediante SQLAlchemy bajo un patrón CRUD modularizado.

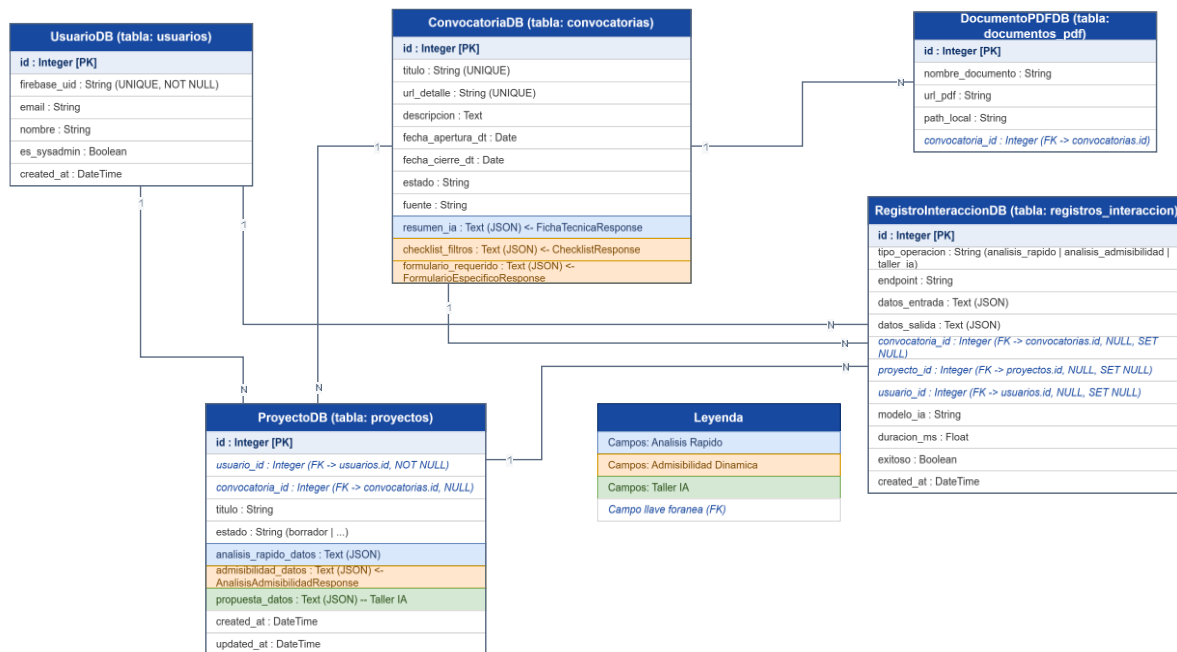


Figura 7: Modelo de datos funcionalidades

Fuente: Elaboración propia

El esquema relacional central orbita alrededor de las siguientes entidades:

- **UsuarioDB:** Entidad de identidad, sincronizada con el proveedor externo de autenticación (Firebase). Almacena el `firebase_uid` como llave de vinculación y actúa como raíz de las relaciones de propiedad: cada proyecto pertenece a un único usuario.
- **Convocatorias:** Entidad maestra que almacena los metadatos transaccionales y estructurales extraídos del portal. Contiene campos de tipo TEXT que almacenan nativamente objetos JSON pre-computados por la IA, tales como `resumen_ia` (Ficha Técnica), `checklist_filtros` y `formulario_requerido`.
- **DocumentoPDFDB:** Entidad de soporte del pipeline de ingesta (sección 3.2). Referencia, mediante llave foránea, los archivos PDF de bases administrativas asociados a cada convocatoria; no es consumida directamente por las tres funcionalidades núcleo, sino que alimenta el proceso de *scraping* y pre-cómputo que las precede.
- **Proyectos:** Representa la propuesta de valor del usuario y es la entidad donde convergen las tres funcionalidades núcleo. Posee relaciones foráneas hacia el creador (UsuarioDB) y la convocatoria objetivo (ConvocatoriaDB), y alberga en campos JSON independientes el resultado de cada una: `analisis_rapido_datos` (Análisis Rápido), `admisibilidad_datos` (Admisibilidad Dinámica) y `propuesta_datos` (Taller IA).
- **RegistroInteraccionDB:** Entidad desacoplada crítica para el cumplimiento de normativas de IA responsable. Captura telemetría detallada de cada inferencia: *endpoints*, tiempos de ejecución en milisegundos, *IPs* de clientes, *payloads* de entrada/salida y estado de éxito, proveyendo un sustrato fundamental para la auditoría técnica del sistema.

3.3.1 Contrato de Datos: Esquemas de Pydantic

La comunicación entre la IA y la base de datos PostgreSQL se rige por modelos de datos estrictos que validan la integridad de los objetos JSON almacenados en los campos TEXT descritos en la Figura 6.

Tabla 4: Modelo Pydantic
Fuente: Elaboración propia

Modelo (Pydantic)	Campo en BD	Descripción del Contenido JSON	Generado por
FichaTecnicaResponse	resumen_ia	Regiones, sectores, TRL, montos máximos y porcentajes de cofinanciamiento.	ai_service
ChecklistResponse	checklist_filtros	Lista de criterios de admisibilidad eliminados (nombre y requisitos).	ai_service
FormularioEspecificoResponse	formulario_requerido	Estructura técnica de campos, tipos de datos, etiquetas y validaciones de Charly.io.	Scraper Service
AnálisisAdmisibilidadResponse	admisibilidad_datos	Combina la evaluación determinística de reglas duras (bloqueos, no verificables y advertencias) con el análisis cualitativo de la IA; expone un estado_general (ADMISIBLE / INADMISIBLE / REQUIERE_MEJORAS) consumido tanto por el Motor de Admisibilidad Dinámica (sección 3.5) como por el Taller IA (sección 3.7).	ai_service y filter_service (evaluación híbrida)

Estos esquemas permiten que el campo **TEXT** de la base de datos sea interpretado nativamente como un objeto estructurado por el *frontend* en Next.js, facilitando la renderización de formularios dinámicos.

3.4 Motor de Emparejamiento (*Matching* Híbrido)

El corazón de la asistencia inicial a la PYME es la reducción del universo de opciones de financiamiento. Para evitar la dependencia exclusiva en la inferencia estocástica de los *LLMs*, se diseñó un algoritmo de *matching* de dos capas:

3.4.1 Capa 1: Filtros Técnicos Determinísticos

Esta capa evalúa condiciones binarias y calculables matemáticamente utilizando la "Ficha Técnica" pre-generada en formato JSON durante la etapa de *scraping*.

- **Filtros Duros (Eliminatorios):** Si la fecha actual supera la fecha_cierre, o si la región de la PYME difiere de las regiones_aplicacion especificadas, la convocatoria se descarta inmediatamente del espacio de búsqueda, ahorrando ciclos de cómputo.
- **Filtros Blandos (Ponderación 0-100):** Se asignan puntajes ponderados según proximidad. Por ejemplo, la compatibilidad del Sector Económico aportó 25 puntos utilizando *matching* parcial de cadenas; el Nivel de Madurez Tecnológica (TRL) aporta hasta 20 puntos si el estado de la propuesta se alinea con las expectativas de la base; y los montos requeridos se comparan heurísticamente con los subsidios máximos disponibles.

Es importante distinguir esta capa de filtros de descubrimiento orientada a acotar rápidamente el universo de convocatorias candidatas mientras el emprendedor aún no ha seleccionado una convocatoria específica de la evaluación de admisibilidad que ocurre una vez seleccionada esa convocatoria (sección 3.5). Ambas etapas contrastan variables similares (sector económico, monto, duración, cofinanciamiento, TRL, tipo de postulante), pero la función `evaluar_reglas_duras_admisibilidad` (`filter_service.py`) que las materializa aquí, clasificando cada hallazgo en bloqueos, no verificables y advertencias, se invoca únicamente dentro del modo estricto de este *endpoint* de descubrimiento (POST `/find-matches`). El Motor de Admisibilidad Dinámica descrito en 3.5 no reutiliza esta capa determinística: por decisión de diseño explícita, la evaluación de la propuesta ya seleccionada queda completamente en manos de la IA contra el *checklist* real de la convocatoria, según se detalla en 3.5.1.

3.4.2 Capa 2: Refinamiento Cualitativo IA (Discovery)

Las convocatorias que superan el umbral técnico pasan al motor Gemini. Se construye un *prompt* dinámico que inyecta la propuesta estructurada del emprendedor (Problema, Solución, Diferenciador, TRL) frente al JSON resumen de las convocatorias finalistas. La IA ejerce como un "Evaluador Experto", retornando un puntaje cualitativo y generando "Objetivos de Taller": un vector de recomendaciones accionables y concretas (ej. "Alinear la declaración de innovación con la sustentabilidad exigida en el párrafo 3") diseñadas para elevar el estándar de la propuesta antes de proceder al formulario. El Anexo A muestra esta pantalla del sistema.

3.5 Motor de Admisibilidad Dinámica

Una vez que el emprendedor selecciona una convocatoria específica entre los resultados del *matching* híbrido (sección 3.4), el sistema transita de un modo de descubrimiento amplio a uno de verificación formal profunda. Este segundo módulo, expuesto a través del *endpoint* POST `/api/v1/propuestas/analisis-admisibilidad-dinamico/{convocatoria_id}`, constituye el segundo núcleo funcional del sistema y responde directamente a la pregunta que motiva esta memoria: ¿cumple esta propuesta, tal como está redactada, los criterios formales de admisibilidad de esta convocatoria en particular?

3.5.1 Evaluación por IA contra el *Checklist* y Reconciliación del Veredicto

A diferencia de la capa de reglas duras de descubrimiento (sección 3.4.1), que opera sobre metadatos generales antes de que el emprendedor seleccione una convocatoria, esta etapa evalúa la coherencia formal entre la propuesta concreta ya redactada y el *checklist_filtros* específico de la convocatoria seleccionada, ya extraído de sus bases administrativas. La función `analizar_admisibilidad_con_checklist` (`ai_service.py`) delega esta evaluación en Gemini: por cada filtro del *checklist* eliminatorio, de evaluación, financiero o de restricción temporal y por cada documento obligatorio, el modelo retorna un estado y una razón textual. A diferencia del motor de *matching* (3.4.1), aquí no se invoca la capa determinística `evaluar_reglas_duras_admisibilidad`; la evaluación queda completamente en manos de la IA contra el *checklist* real, según lo documenta explícitamente un comentario en el código

del *endpoint* (app/api/propuestas.py). Cada filtro puede quedar en uno de cuatro estados, con una criticidad asociada (ALTA, MEDIA o BAJA):

- **CUMPLE:** el requisito se satisface según lo declarado en la propuesta; no aporta puntos de fricción al veredicto.
- **NO_CUMPLE:** incumplimiento detectado por la IA a partir del *checklist* real; por sí solo impide alcanzar el estado ADMISIBLE mientras no se corrija.
- **PARCIAL y NO_VERIFICABLE:** el primero indica cumplimiento incompleto o ambiguo; el segundo, una condición que la IA no puede confirmar con los datos disponibles en la propuesta. Ambos se marcan para revisión del emprendedor en vez de resolverse silenciosamente hacia un extremo, evitando falsos positivos o negativos.

Dado que Gemini a veces entrega un *estado_general* desalineado con el detalle real de los filtros por ejemplo, todos los filtros en CUMPLE pero *estado_general* = REQUIERE_MEJORAS, o incluso INADMISIBLE sin ningún filtro en NO_CUMPLE que lo respalde, una función de reconciliación determinística (*_reconciliar_estado_admisibilidad*, *ai_service.py*) corrige ese veredicto agregado después de recibir la respuesta del modelo, en dos direcciones simétricas: promueve hacia ADMISIBLE cuando ningún filtro quedó en NO_CUMPLE, ningún documento obligatorio está AUSENTE, y además todos los filtros evaluados están en CUMPLE o el puntaje ya es 100; y corrige un INADMISIBLE hacia REQUIERE_MEJORAS cuando ese mismo veredicto no tiene ningún filtro NO_CUMPLE ni documento AUSENTE que lo sustente. Esta segunda regla se agregó a partir de evidencia empírica encontrada durante la validación técnica (sección 4.3): en la corrida piloto del set 5N, un caso etiquetado ADMISIBLE fue declarado INADMISIBLE por Gemini con puntaje 60/100 y cero filtros NO_CUMPLE, una contradicción interna que la función original no corregía. Si la llamada a Gemini falla o el JSON del *checklist* es inválido, un *fallback* determinístico separado retorna directamente REQUIERE_MEJORAS en vez de arriesgar un veredicto no fundamentado.

3.5.2 Formulario Dinámico y Análisis Combinado

En paralelo, el sistema renderiza un formulario específico para la convocatoria seleccionada, generado a partir de la estructura de campos, tipos de dato y validaciones extraída previamente por el módulo *AnalizadorFormulariosCorfo* durante la fase de scraping (sección 3.2.2.3). El emprendedor completa este formulario, y sus respuestas se envían junto al *checklist* real a la evaluación por IA descrita en 3.5.1 que además de contrastar cada filtro revisa coherencia interna, claridad de redacción y alineación estratégica para producir un objeto *AnalisisAdmisibilidadResponse* con un estado general que, tras la reconciliación descrita en 3.5.1, puede tomar tres valores: ADMISIBLE, INADMISIBLE o REQUIERE_MEJORAS. Este último estado es el punto de entrada natural al tercer núcleo funcional del sistema: el Taller de Mejora Iterativa (sección 3.7). El Anexo A muestra esta pantalla del sistema.

3.6 Ingeniería de *Prompts* Avanzada y Mitigación de Alucinaciones

La integración de la IA no se limita al envío de consultas asíncronas, sino a la construcción de un marco de restricciones lingüísticas formales que fuerzan a Gemini a operar como un procesador de datos determinístico.

3.6.1 Modo JSON Restringido y Validación Pydantic

Para asegurar la comunicación bidireccional perfecta entre la IA y el *backend* (FastAPI), todas las configuraciones de generación (*generation_config*) tienen forzado el parámetro *response_mime_type*: "application/json". Esto prohíbe sintácticamente que el modelo responda con delimitadores Markdown (````json`), saludos o justificaciones narrativas fuera de la estructura de datos.

Una vez recibida la cadena JSON, se procesa a través de la capa de Esquemas de Pydantic v2. Esto garantiza la integridad referencial y de tipos. Si la IA debiera generar una estructura tipada como *AnalisisAdmisibilidadResponse*, Pydantic verifica la presencia de cada atributo (ej. *resumen_admisibilidad*, *evaluacion_filtros*, etc.). Cabe precisar que esta validación se restringe a la presencia y el tipo de cada campo, no al vocabulario que dicho campo contiene por defecto: el campo *estado_general* fue originalmente tipado como *str* simple, sin *Literal* ni *Enum* sobre los tres valores documentados (*ADMISIBLE*, *INADMISIBLE*, *REQUIERE_MEJORAS*; `app/schemas/ai_responses.py`), por lo que Pydantic no rechazaba variantes textuales que Gemini generaba fuera de ese vocabulario. La sección 4.3 cuantifica el impacto real de esta brecha sobre datos de auditoría en producción y sobre una corrida piloto controlada, y documenta la corrección aplicada: un validador de normalización (*mode*="before") que mapea las variantes observadas *NO_ADMISIBLE*, "NO ADMISIBLE", *NO_CUMPLE*, *PARCIAL*, *PARCIALMENTE_ADMISIBLE*, *ADMISIBLE_CON_OBSERVACIONES*, *CUMPLE*, y equivalentes en inglés para criticidad (*HIGH/MEDIUM/LOW*) a los valores canónicos antes de la validación de tipo *Literal*, en vez de restringir el tipo sin normalizar, lo que habría descartado análisis cualitativos válidos solo por un desajuste de vocabulario. Cualquier variante no reconocida cae de forma conservadora a *REQUIERE_MEJORAS*, nunca a *ADMISIBLE*, consistente con el criterio de costo asimétrico de errores ya declarado en el "Foco de interés especial" de la sección 4.3.

3.6.2 Control Térmico (Temperature Control)

Se aplicó un ajuste fino de los hiperparámetros del modelo según el caso de uso:

- Para tareas de extracción factual estricta, como la "Ficha Técnica" o el "Análisis de Admisibilidad" contra el *Checklist*, la temperatura se fija en **0.1 - 0.2**. Esto prioriza las probabilidades de tokens más altos, minimizando la creatividad algorítmica y maximizando la exactitud factual.
- Para tareas de síntesis cruzada, como el "Taller de Mejora IA", la temperatura se relaja a valores por defecto, permitiendo a la red neuronal explorar variaciones semánticas útiles para la redacción de propuestas de valor y sugerencias creativas.

3.6.3 Resiliencia a Fallas y Trazabilidad (*Fallbacks*)

Ante anomalías como filtros de seguridad de Google o respuestas truncadas, la arquitectura previene caídas del sistema (*Status 500*) mediante:

- **Recuperación de Truncamiento:** Si un error `JSONDecodeError` es arrojado, una función heurística inspecciona la cadena en reversa buscando el último carácter }, descartando datos corruptos finales y cerrando la lista lógica con un], salvando el procesamiento del 95% de la carga útil.
- **Logs Crudos Estructurados:** Independientemente de la validación, la tupla completa del intercambio (*Prompt* Inyectado + Output Crudo) se vuelca a disco bajo el directorio `ai_raw_responses/` en formato JSONL. Esto no solo provee trazabilidad técnica (*Debugging*), sino que establece una base metodológica para la validación experta posterior.

Precisión de ubicación: la función de recuperación heurística ante truncamiento (búsqueda del último carácter } válido y cierre de la lista con]) está implementada específicamente en `_intentar_recortar_lista_json_truncada` (`ai_service.py`), y no en el módulo genérico de utilidades JSON (`app/core/json_utils.py`), cuya función `safe_json_loads` se limita a un parseo simple sin lógica de recuperación. De forma análoga, el volcado de la tupla *prompt*/respuesta cruda se realiza mediante las funciones `_guardar_respuesta_cruda*` del propio `ai_service.py` hacia el directorio `app/data/ai_raw_responses/` un mecanismo distinto y complementario al de `usage_logger.py`, que registra telemetría de uso (tiempos, éxito/fracaso) y no el contenido crudo de las respuestas.

3.7 Taller de Mejora Iterativa (Taller IA)

Cuando el Motor de Admisibilidad Dinámica (sección 3.5) retorna un `estado_general` de `REQUIERE_MEJORAS`, el sistema no se limita a informar el diagnóstico: ofrece un espacio de trabajo dedicado el Taller IA diseñado para cerrar el ciclo entre la detección de un problema y su corrección efectiva, sin que el emprendedor abandone la plataforma. Esta es la tercera funcionalidad núcleo del sistema y, en términos de superficie de código, la más extensa de las tres.

3.7.1 Vista Doble y Edición Asistida

La interfaz presenta dos columnas sincronizadas: a la izquierda, el borrador original de la propuesta en modo de solo lectura; a la derecha, una versión editable donde los campos que el análisis de admisibilidad marcó como problemáticos se resaltan visualmente (`renderCampoConHighlight`). Esta disposición permite comparar en todo momento el estado previo y el estado en edición, evitando que el usuario pierda de vista qué fue señalado y por qué. El Anexo A muestra esta pantalla del sistema.

3.7.2 Regeneración Controlada de Sugerencias

Para cada campo problemático, la IA ofrece una redacción sugerida inicial. El usuario puede solicitar hasta tres regeneraciones adicionales de esa sugerencia (intentos 2, 3 y 4 el intento 1 corresponde a la sugerencia original del análisis) a través del *endpoint* POST `/api/v1/propuestas/sugerencia/regenerar`, validado en el rango $2 \leq \text{nro_intento} \leq 4$. Cada regeneración no es una simple repetición: el servicio aplica una pauta de redacción distinta según el número de intento priorizando primero concisión, luego precisión técnica, y finalmente impacto narrativo e incluye en el *prompt* la sugerencia previa para evitar que el modelo repita la misma idea. A diferencia de las tareas de extracción factual descritas en 3.6.2, aquí la temperatura se fija en 0.7: una decisión deliberada, ya que en esta etapa el objetivo no es exactitud determinística sino diversidad genuina entre alternativas de redacción.

3.7.3 Persistencia, Reanálisis y Cierre del Ciclo

El estado de edición (`inputEditable`) se sincroniza de forma incremental hacia *localStorage* con un *debounce* de 600 ms, permitiendo que el usuario cierre y retome el taller sin perder trabajo. Al recargar la página, si existe un borrador guardado, este se fusiona con el formulario original y se notifica al usuario que se restauró una edición pendiente. Deliberadamente, el sistema no ofrece una opción de "marcar como cumple" manual sobre los filtros: dado que el *backend* no soporta la sobreescritura de un criterio de admisibilidad, permitir ese atajo en la interfaz induciría al usuario a una falsa sensación de cumplimiento. En su lugar, aplicar una sugerencia solo marca el campo como "pendiente de reanálisis", sin inflar artificialmente el conteo de criterios cumplidos. El cierre del ciclo ocurre cuando el usuario solicita un reanálisis: se invoca el mismo *endpoint* de Admisibilidad Dinámica descrito en la sección 3.5 sobre el formulario editado. Si el resultado alcanza `estado_general = ADMISIBLE`, la interfaz transiciona a una vista de éxito *in-place* sin cambiar de ruta que resume los criterios cumplidos y permite descargar un PDF de la postulación mediante la función nativa de impresión del navegador (`window.print()`), apoyada en una hoja de estilos dedicada que oculta el resto de la interfaz durante la impresión.

3.7.4 Telemetría de Uso

Cada interacción relevante del taller inicio, edición de campo, aplicación de una sugerencia, solicitud de regeneración, solicitud y finalización de reanálisis, cierre del taller y descarga del PDF se reporta de forma asíncrona y no bloqueante (*fire-and-forget*) al *endpoint* POST `/api/v1/audit/evento-taller`, deliberadamente sin autenticación de administrador, ya que es el propio usuario reportando su actividad. Estos eventos se persisten junto con el resto de la auditoría del sistema (`tipo_operacion = "taller_ia"`), constituyendo el sustrato cuantitativo que se utiliza en el Capítulo 4 para caracterizar el uso real del taller: tiempo efectivo invertido, tasa de aceptación de sugerencias, uso de regeneraciones y proporción de sesiones que alcanzan el estado ADMISIBLE.

CAPÍTULO 4: VALIDACIÓN DE LA SOLUCIÓN

La validación de Admisivo responde directamente al objetivo específico 5 (sección 1.5.2), el cual establece dos ejes de evaluación complementarios: (i) la fiabilidad percibida y la usabilidad de la plataforma por parte de usuarios reales, y (ii) la precisión técnica del motor de admisibilidad frente al juicio de un experto sobre un conjunto de casos de prueba controlados. Este capítulo presenta la metodología y los resultados de ambos ejes: el primero, mediante una prueba de usabilidad SUS con usuarios reales; el segundo, dentro de las restricciones de tiempo y de acceso a un evaluador experto disponibles para esta memoria, según se detalla en la sección 4.3, donde también se declara explícitamente el alcance de esa restricción.

4.1 Estrategia General de Validación

Siguiendo el marco de evaluación descrito en la sección 2.6 (Evaluación de la Calidad de la Asistencia), se optó por combinar dos metodologías de naturaleza distinta:

- **Evaluación con usuarios (Evaluación Humana):** mide si la herramienta resuelve el problema de asimetría de información y complejidad descrito en el Capítulo 1 desde la perspectiva del emprendedor, es decir, si el sistema es utilizable, comprensible y genera confianza. Para esto se aplicó una prueba de usabilidad basado en el instrumento System Usability Scale (SUS).
- **Evaluación de precisión técnica:** mide si los veredictos de admisibilidad que entrega el sistema (ADMISIBLE / INADMISIBLE / REQUIERE_MEJORAS, sección 3.5.2) coinciden con lo esperado, mediante dos fuentes complementarias: un conjunto de 20 casos de prueba (5N) contruidos deliberadamente para cubrir escenarios claros y límite, contrastados contra una verdad base definida por el propio autor y anclada en datos reales de CORFO; y un caso real de postulación (MunIA, Start-Up Chile, sección 4.3.4) contrastado contra el juicio efectivo de 279 evaluadores externos. Un panel de juicio experto ciego dedicado específicamente al set de 20 casos queda propuesto como extensión de este eje de validación (sección 4.3.5).

Ambos ejes son necesarios y no intercambiables: un sistema puede ser percibido como fácil de usar y aun así entregar diagnósticos de admisibilidad incorrectos (falsos positivos que generan una falsa sensación de seguridad, o falsos negativos que desalientan a un postulante viable). Por ello, la naturaleza parcial de la validación de precisión técnica basada en una verdad base propia y en un único caso real, sin panel de juicio experto ciego, según se detalla en la sección 4.3 se declara explícitamente como una limitación del presente informe, no como un eje omitido del diseño de validación.

4.2 Validación de Usabilidad: Test SUS con Usuarios

4.2.1 Alianza institucional y diseño del instrumento

La prueba de usabilidad se coordinó con el Laboratorio de Experiencia de Usuario de la Universidad Técnica Federico Santa María (UTFSM), aplicando un protocolo moderado de tipo *think-aloud* sobre un guion de cinco tareas secuenciales, seguido de la encuesta System Usability Scale (SUS) descrita en la sección 2.6.

Se seleccionó SUS por sobre instrumentos ad-hoc por dos razones adicionales a su condición de estándar de facto (sección 2.6): es rápido de aplicar, adecuado para una muestra reducida en etapa de prototipo funcional, y su cálculo produce un puntaje 0-100 con una escala adjetival de interpretación ya establecida [14], lo que facilita comunicar el resultado sin ambigüedad.

El protocolo moderado consistió en:

- Entrega de una "Hoja del Emprendedor" con un perfil ficticio (proyecto StockPro Inteligente, persona natural sin ventas previas, con \$500.000 de aporte propio, dedicación *part-time*), de manera que todos los participantes operaran sobre el mismo caso de negocio y los resultados fueran comparables entre sí.
- Ejecución de cinco tareas secuenciales sobre la plataforma real, sin intervención del moderador salvo fallos críticos del sistema.
- Aplicación de la encuesta SUS al cierre de la sesión.

La Tabla 5 resume las cinco tareas del guion y el módulo del sistema que cada una ejercita, estableciendo la trazabilidad entre el protocolo de prueba y la arquitectura descrita en el Capítulo 3.

Tabla 5: Tareas del protocolo de usabilidad y módulo evaluado
Fuente: Elaboración Propia

Tarea	Objetivo para el usuario	Módulo del sistema evaluado (Cap. 3)
1. Descubrimiento	Encontrar el fondo con mayor % de compatibilidad para su idea	Análisis Rápido / Motor de Emparejamiento Híbrido (3.4)
2. Extracción de datos	Ubicar monto máximo y % de cofinanciamiento exigido	Detalle de convocatoria (Ficha Técnica, 3.2.2.4)
3. Llenado de formulario	Completar las 4 secciones del formulario dinámico con los datos del perfil	Admisibilidad Dinámica / Formulario Dinámico (3.5.2)
4. Interpretación del diagnóstico	Leer el estado general y las razones de incumplimiento (cofinanciamiento, innovación)	Motor de Admisibilidad Dinámica / Reglas Duras (3.5.1)
5. Uso del Taller de Mejora	Editar un campo guiándose por la sugerencia de la IA	Taller IA (3.7)

4.2.2 Perfil de los Participantes

Participaron 4 personas, adultas jóvenes, con experiencia tecnológica básica-media y que en algún momento estuvieron vinculadas a un proyecto de base tecnológica. De los 4 participantes, solo 1 había postulado efectivamente a fondos concursables con anterioridad.

Este perfil es cercano al "óptimo" para la plataforma usuarios familiarizados con el vocabulario emprendedor y con cierta soltura tecnológica, lo que constituye una limitación reconocida de la muestra: no representa completamente al universo de postulantes descrito en la sección 2.1.1, donde el 47,7% de las empresas tiene entre 0 y 5 años de antigüedad y una fracción relevante nunca ha postulado a un instrumento público. Un perfil con menor experiencia tecnológica o sin trayectoria emprendedora previa podría reportar una usabilidad percibida distinta, particularmente en la Tarea 4 (interpretación de diagnóstico) y la Tarea 5 (uso del Taller IA), que exigen comprender conceptos propios del dominio (cofinanciamiento, admisibilidad, TRL). Se recomienda ampliar la muestra en una segunda ronda con usuarios de perfil menos favorable, según se detalla en el Capítulo 5.

Respecto al tamaño muestral (n=4), se sigue el criterio de Nielsen para pruebas de usabilidad cualitativas/moderadas, según el cual entre 4 y 5 usuarios permiten detectar la mayoría de los problemas de usabilidad severos de una interfaz [15], siendo un tamaño adecuado para una iteración de validación temprana de un PMV, aunque insuficiente para generalizar estadísticamente el puntaje SUS obtenido a la población total de postulantes.

4.2.3 Resultados Cuantitativos del SUS

La Tabla 6 presenta las respuestas brutas de los 4 participantes (P1-P4) a los 10 ítems del instrumento. P1-P4 identifican a la misma persona en la Tabla 6, la Tabla 7 y las observaciones cualitativas de la sección 4.2.4.

Tabla 6: Respuestas brutas del cuestionario SUS (escala 1-5)
Fuente: Elaboración Propia

#	Ítem	P1	P2	P3	P4
1	Creo que me gustaría utilizar este sistema frecuentemente	5	4	5	5
2	Encuentro este sistema innecesariamente complejo	1	1	3	2
3	Pienso que el sistema es fácil de usar	5	5	3	4
4	Creo que necesitaría soporte técnico para usar el sistema	1	1	2	1
5	Encuentro las funciones del sistema bien integradas	5	5	4	5
6	He encontrado demasiada inconsistencia en el sistema	1	1	1	1
7	Creo que la mayoría aprendería a usarlo rápidamente	4	5	4	4
8	He encontrado el sistema bastante incómodo de usar	1	1	1	2
9	Me siento muy seguro haciendo uso del sistema	5	4	5	5
10	Necesitaría aprender muchas cosas antes de poder manejarlo	1	1	1	1

Siguiendo el procedimiento estándar de Brooke [13] para ítems impares (1,3,5,7,9) se resta 1 al puntaje bruto; para ítems pares (2,4,6,8,10) se resta el puntaje bruto de 5; la suma de las 10 contribuciones se multiplica por 2.5 para obtener un puntaje en el rango 0-100 se obtuvieron los siguientes puntajes individuales:

Tabla 7: Puntaje SUS individual y subescalas
Fuente: Elaboración Propia

Participante	SUS Total	Subescala Usabilidad	Subescala Aprendibilidad
P1	97.5	96.9	100.0
P2	95.0	93.8	100.0
P3	82.5	81.3	87.5
P4	90.0	87.5	100.0
Promedio	91.25	89.84	96.88

El puntaje "SUS" promedio obtenido es de **91.25/100**. Según la escala adjetival de Bangor, Kortum y Miller [14] que ubica los puntajes sobre 80.3 en la categoría "Excelente" y sobre ~90 en el rango superior de esa categoría, cercano a "El mejor sistema imaginable" el resultado se interpreta como una usabilidad excelente, muy por encima del promedio de referencia de la industria (SUS $\approx$ 68, percentil 50 según Sauro, 2011). Los cuatro participantes puntuaron sobre 80, sin ningún caso por debajo del umbral de aceptabilidad (SUS $\geq$ 68).

Al descomponer el puntaje en las subescalas propuestas por Lewis y Sauro [16] se observa que la dimensión de **Aprendibilidad alcanza 96.88/100**, prácticamente el máximo posible, con tres de los cuatro participantes marcando el valor óptimo (1 = "en desacuerdo") en ambos ítems relacionados con la necesidad de soporte técnico o de aprendizaje previo. Esto es consistente con el diseño de flujo lineal del sistema (Análisis Rápido $\rightarrow$ Admisibilidad Dinámica $\rightarrow$ Taller IA), que no exige al usuario un modelo mental complejo antes de obtener valor.

El único participante que se apartó del patrón de puntajes casi perfectos del resto de la muestra fue **P3** (SUS = 82.5, el más bajo de la muestra), concentrando sus respuestas más bajas en los ítems 2 y 3 ("complejidad" y "facilidad de uso"). Este es el mismo participante cuya sesión registró, en el análisis cualitativo (secciones 4.2.4 y 4.2.5), el incidente más severo detectado en todo el protocolo: no logró completar la Tarea 5 (Taller IA) sin intervención del moderador, debido a un defecto de la interfaz que hace convivir en pantalla dos estados contradictorios sobre un mismo criterio (texto de falla original + chip "PENDIENTE REANÁLISIS") y a que el botón "Reanalizar" queda visualmente oculto al revisar el *checklist* completo. La correspondencia entre el puntaje cuantitativo más bajo de la muestra y el único bloqueo de tarea que requirió ayuda externa es la evidencia más sólida de validez del instrumento SUS obtenida en este estudio: confirma que el cuestionario efectivamente capturó, de forma agregada, una fricción real y no percibida al azar. La sección 4.2.5 detalla este hallazgo como un defecto a corregir, no solo como una percepción subjetiva.

4.2.4 Observaciones Cualitativas del Protocolo de Tareas

A continuación, se sintetiza, por tarea, el comportamiento observado de los 4 participantes durante la sesión moderada, a partir de la revisión de las grabaciones.

Tareas 1 y 2: Descubrimiento y extracción de datos. Los cuatro participantes lograron identificar la convocatoria y reportar correctamente los valores solicitados (monto máximo y % de cofinanciamiento), aunque por rutas distintas: P1 usó los filtros del listado y completó la tarea en aproximadamente 1 minuto; P2 encontró la convocatoria por *scroll* simple, pero se adelantó espontáneamente a iniciar el Análisis Rápido antes de que se le pidiera, debiendo devolverse para reportar los valores; P3 utilizó el buscador directamente; P4 no fue de inmediato a la sección "Convocatorias" sino que primero exploró todo el *landing page* buscándola, y una vez ahí usó el buscador sin problemas. La tarea no presentó bloqueos, pero la conducta de P4 sugiere que el punto de entrada a "Convocatorias" no es igualmente evidente desde la página de inicio para todos los perfiles.

Tarea 3: Llenado del formulario dinámico. Sin incidentes: los cuatro completaron las secciones navegando el formulario con fluidez, escribiendo a medida que leían los datos del perfil entregado. No se registraron dudas relevantes en ningún participante.

Tarea 4: Interpretación del diagnóstico. Aquí surgen las primeras divergencias relevantes:

- **P1** entendió que faltaba ser admisible y que había puntos por corregir, aunque sin el detalle específico en esta pantalla comportamiento consistente con el diseño intencional de "experiencia *freemium*" del resumen (el detalle completo se reserva para el Taller IA).
- **P2** obtuvo el puntaje, pero no tuvo claro si correspondía a un porcentaje o a "puntos sobre 100", asumió que era lo segundo. Sí comprendió con claridad que el resultado no era ADMISIBLE, e intentó descargar el PDF esperando encontrar más detalle el PDF disponible en este punto del flujo no aporta información adicional a la ya mostrada en pantalla, generando una expectativa incumplida.
- **P3** leyó correctamente los puntos a corregir, pero al no contar con información adicional suficiente en ese momento, dudó de si esos eran realmente los puntos correctos por corregir. También intentó descargar el PDF, con el mismo resultado que P2: información idéntica a la ya vista, generando una sensación de redundancia.
- **P4** identificó correctamente cada uno de los puntos solicitados, sin incidentes.

Tarea 5: Taller IA. La tarea con mayor variabilidad de la sesión completa:

- **P1** completó el flujo hasta alcanzar ADMISIBLE. Como retroalimentación espontánea, señaló que hay partes mejorables sin poder detallarlas, que las alertas visuales del Taller le costaron seguir por exceso de información simultánea en pantalla, que la navegación tipo carrusel no tiene botón "Siguiente", y propuso que

las secciones no activas se colapsen (plegado) para mantener la atención del usuario en los cambios que sí está aplicando.

- **P2** identificó intuitivamente que la columna izquierda mostraba lo que estaba mal, revisó cada sección, leyó la razón entregada por la IA, aceptó la propuesta inicial, pidió una segunda versión para comparar alternativas y finalmente la aceptó. Tuvo dificultad para ubicar el botón "Reanalizar". Completó el flujo exitosamente (propuesta correcta), pero calificó la cantidad de información en pantalla como "un poco chocante" la primera vez que se enfrenta a ella.
- **P3** sufrió el incidente más severo del estudio: aplicó cambios repetidamente porque el botón de aplicar no desaparecía como esperaba; el texto de la sugerencia cambió correctamente, pero **el panel derecho seguía mostrando el motivo de la falla original al mismo tiempo que el chip "PENDIENTE REANÁLISIS"** (Figura 7), comunicando de facto dos estados contradictorios sobre el mismo criterio. El participante se declaró "perdido" ante esta contradicción y permaneció bloqueado un período prolongado. Adicionalmente, el botón "Reanalizar" queda oculto por el propio *scroll* de la vista cuando el *checkbox* está expandido, por lo que **P3 no logró completar la tarea sin intervención directa del moderador**. Su retroalimentación espontánea fue precisa: "vi un cambio físico, pero no vi algo que me indicara que estaba listo (...) no incitó a reanalizar".
- **P4** tampoco encontró de inmediato el botón para ir directo al Taller IA desde el resultado del análisis; usó la barra de navegación superior, lo que lo llevó a una página general "Taller IA" (de tipo landing/explicativa) en vez de a su postulación activa. Al reintentar, sí logró acceder a la postulación en curso. Ya dentro, fue el único participante en generar múltiples reanálisis consecutivos porque el monto de cofinanciamiento reportado por la IA en las sucesivas regeneraciones de una misma sugerencia fue inconsistente (385 → 412 → 490 → 500), convergiendo recién en el cuarto intento al valor correcto que además coincidía con el que la IA ya había entregado en el primer intento. El resto de los cambios se aplicaron correctamente y el proyecto alcanzó finalmente el estado ADMISIBLE.

SISTEMA DE APOYO BASADO EN INTELIGENCIA ARTIFICIAL GENERATIVA PARA MEJORAR LA ADMISIBILIDAD DE POSTULACIONES DE PYMES A FONDOS CORFO

ADMISIVO IA

Edita para mejorar

IA MARCA ESTE CAMPO - PARCIAL

El postulante indica una dedicación 'Part-time', lo cual es parcialmente admisible y requiere revisión según las bases de la convocatoria.

PROPOSTA DE LA IA

Aplicar al campo

Pedir otra versión (3)

Copiar

Full-time exclusivo (100%)

8. Dedicación del equipo emprendedor

Full-time exclusivo (100%)

IA MARCA ESTE CAMPO - PARCIAL

El modelo de ingresos describe la forma de cobro ('suscripción mensual de \$10000 a cada almacén') pero no aborda la escalabilidad comercial ni el mercado objetivo de manera explícita.

PROPOSTA DE LA IA

Aplicar al campo

Pedir otra versión (3)

Copiar

Nuestro modelo de ingresos se basa en una suscripción mensual de \$10.000 CLP por almacén. Apuntamos a un mercado objetivo inicial de 5.000 almacenes de barrio en la Región Metropolitana, con potencial de expansión a nivel nacional. La escalabilidad se logrará a través de alianzas estratégicas con distribuidores y gremios de almaceneros, y un modelo de bajo costo de adquisición de clientes digital, permitiendo un crecimiento sostenido de usuarios y, por ende, de ingresos.

7. Modelo de Ingresos

CLP por almacén. Apuntamos a un mercado objetivo inicial de 5.000 almacenes de barrio en la Región Metropolitana, con potencial de expansión a nivel nacional. La escalabilidad se logrará a través de alianzas estratégicas con distribuidores y gremios de almaceneros, y un modelo de bajo costo de adquisición de clientes digital, permitiendo un crecimiento sostenido de usuarios y, por ende, de ingresos.

83% 3 de 6 reglas

Tienes ediciones sin reanalizar. El puntaje y el checklist no reflejan tus cambios hasta que pases **Reanalizar**.

El aporte pecuniario del postulante es de \$5.000.000, lo que representa exactamente el 25% del costo total del proyecto (\$15.000.000 CORFO + \$5.000.000 propio + \$20.000.000 total).

Dedicación del equipo emprendedor - PENDIENTE REANÁLISIS

El postulante indica una dedicación 'Part-time', lo cual es parcialmente admisible y requiere revisión según las bases de la convocatoria.

Mérito innovador a nivel nacional

La descripción de la solución ('Un Software web básico para que los almacenes lleven el control de su inventario usando el celular, reemplazando el físico cuaderno') no detalla la novedad tecnológica significativa a nivel nacional, pareciendo una digitalización de un proceso existente sin elementos innovadores claros.

Viabilidad del modelo comercial - PENDIENTE REANÁLISIS

El modelo de ingresos describe la forma de cobro ('suscripción mensual de \$10000 a cada almacén') pero no aborda la escalabilidad comercial ni el mercado.

Figura 8: Estados contradictorios en el *checklist* del Taller IA
Fuente: Captura de sesión participante P3

4.2.5 Hallazgos Críticos y Defectos de Usabilidad Detectados

Más allá del puntaje agregado, el análisis cualitativo de las 4 sesiones permite clasificar los hallazgos por severidad, siguiendo una escala de 4 niveles adaptada de Nielsen (0 = no es un problema, 4 = catástrofe de usabilidad) [17]:

Tabla 8 Hallazgos de usabilidad detectados
Fuente: Elaboración Propia

#	Hallazgo	Participante(s)	Módulo	Severidad
1	Estados contradictorios simultáneos: tras aplicar una sugerencia, el criterio muestra a la vez el chip "PENDIENTE REANÁLISIS" y el texto original de la falla, sin distinguir "cambio capturado, pendiente de confirmar" de "sigue fallando"	P3	Taller IA <i>checklist</i> de criterios (3.5.1 / 3.7.3)	Crítica (4) bloqueó la tarea sin ayuda externa
2	Botón "Reanalizar" queda oculto por el <i>scroll</i> cuando el <i>checklist</i> completo está expandido	P3	Taller IA	Alta (3)
3	Ausencia de una confirmación positiva explícita tras aplicar un cambio ("se aplicó, continúa")	P3	Taller IA	Alta (3)
4	Inconsistencia numérica entre regeneraciones sucesivas de una misma sugerencia (monto de cofinanciamiento variando 385→412→490→500 antes de converger al valor correcto, ya presente desde el primer intento)	P4	Taller IA Regeneración de Sugerencias (3.7.2)	Alta (3) riesgo de exactitud factual, no solo de UX
5	Navegación tipo carrusel sin botón "Siguiente" explícito, junto con densidad alta de información simultánea en pantalla	P1, P2	Taller IA	Media (2)
6	Ambigüedad en la etiqueta del puntaje de admisibilidad (no queda claro si es un porcentaje o "puntos sobre 100")	P2	Resultado de Admisibilidad Dinámica (3.5.2)	Media (2)
7	Expectativa incumplida: el PDF descargable desde la pantalla de resultados no agrega información respecto de lo ya visible en pantalla	P2, P3	Resultado de Admisibilidad Dinámica	Media (2)
8	Punto de entrada al Taller IA no siempre evidente desde el resultado; la barra de navegación puede llevar a una página genérica en vez de a la postulación activa	P4	Navegación global / Taller IA	Media (2)

El hallazgo n.º 4 merece una lectura técnica adicional, más allá de la usabilidad de la interfaz: la sección 3.7.2 fija deliberadamente la temperatura del modelo en 0.7 durante la regeneración de sugerencias del Taller IA, para favorecer diversidad genuina en la redacción. La sesión de P4 sugiere que esa misma temperatura elevada puede perturbar no solo el estilo de la redacción, sino también **cifras factuales que quedan embebidas en el mismo texto sugerido** (el monto de cofinanciamiento), un efecto no deseado que la arquitectura actual no aísla. Este hallazgo debe considerarse un insumo directo para el Capítulo 5 (recomendaciones): una posible mitigación es separar, a nivel de *prompt* o de esquema de salida, los campos numéricos factuales (que deberían mantenerse fijos entre

regeneraciones, independiente de la temperatura) de los campos de redacción libre (que sí se benefician de la variabilidad).

Es relevante notar que los hallazgos 1, 2 y 3 los de mayor severidad corresponden todos a la sesión de P3, el mismo participante que obtuvo el SUS más bajo de la muestra (82.5, sección 4.2.3). Esta convergencia entre la señal cuantitativa (puntaje bajo) y la señal cualitativa (bloqueo real que requirió ayuda) es la evidencia más fuerte de validez interna obtenida en este estudio, y es precisamente el tipo de hallazgo que un puntaje SUS agregado, sin el respaldo de observación cualitativa por tarea, habría dejado invisible dentro de un promedio grupal que de todas formas se mantiene en la categoría "Excelente".

4.3 Validación Técnica: Precisión de Detección de Admisibilidad

El objetivo específico 5 de esta memoria exige adicionalmente validar la precisión técnica del motor de admisibilidad (secciones 3.5.1 y 3.5.2) mediante un análisis comparativo basado en juicio de experto sobre un conjunto de casos de prueba controlados. Esta sección documenta el diseño metodológico de dicha validación un set de 20 casos de prueba estratificados en 4 instrumentos CORFO distintos (5N, sección 4.3.1), los hallazgos técnicos detectados durante su implementación junto con las correcciones aplicadas sobre el propio sistema (secciones 4.3.2 y 4.3.3), y los resultados cuantitativos finales: 100% de conformidad de vocabulario, 100% de precisión en la detección de casos ADMISIBLE e INADMISIBLE, y un *recall* promedio de criterios bloqueantes de 91% sobre el set completo. Esta validación se complementa con el contraste de un caso real de postulación contra juicio humano externo (sección 4.3.4); el alcance de ambas fuentes de evidencia y las extensiones metodológicas propuestas se detallan en la sección 4.3.5.

4.3.1 Diseño del Set de Casos de Prueba (5N)

Siguiendo la recomendación del profesor guía de estratificar la muestra en al menos 5 casos por categoría de proyecto expresada como 5N, con N el número de convocatorias o instrumentos CORFO distintos cubiertos, en vez de depender solo de la cantidad de expertos consultados, se extendió el *framework* de validación ya implementado en `tests/validation/test_admisibilidad_precision.py`. Ese *framework*, que originalmente cubría 5 casos sobre una única convocatoria de prueba uno de ellos anclado a una carta de rechazo CORFO real, `ground_truth_22IR-209067.json`, se amplió a N=4 convocatorias reales de distinto instrumento (Semilla Inicia, Crece, Escala y Subsidio Semilla de Asignación Flexible), sembradas con el mismo patrón de `test_scenario_seeder.py` usado para el escenario SUS (sección 4.2).

Cada convocatoria aporta 5 perfiles de proyecto sintéticos que cubren deliberadamente los tres estados posibles del sistema (ADMISIBLE, INADMISIBLE, REQUIERE_MEJORAS) y casos límite, por ejemplo, un monto solicitado apenas sobre el máximo permitido, o un cofinanciamiento apenas insuficiente, ejercitando específicamente los cuatro estados de filtro y la función de reconciliación del veredicto agregado descritos en 3.5.1. El resultado es un set de 20 casos (5N) ejecutable de punta a punta, sobre el cual se realizó una corrida

piloto técnica 2 repeticiones de IA por caso, 40 observaciones totales para validar el instrumento de medición antes de reportar cualquier cifra de precisión como definitiva.

Respecto al protocolo de construcción de esa verdad base, cabe explicitar cómo se definieron las etiquetas esperadas de cada caso, dado que de ello depende la lectura de las cifras reportadas más adelante. La etiqueta de cada uno de los 20 casos (ADMISIBLE, INADMISIBLE o REQUIERE_MEJORAS) se fijó a partir de la lectura directa de las bases administrativas de la convocatoria correspondiente, y se escribió antes de construir y afinar los *prompts* del motor de admisibilidad, de modo que el comportamiento observado del sistema no pudiera influir retroactivamente sobre la definición de lo que se consideraba una respuesta correcta. Esta anterioridad temporal mitiga el riesgo de ajustar la verdad base al resultado, pero no lo elimina: el etiquetado y la posterior comparación contra los *checklists* generados los realizó un único anotador, el propio autor, sin condición de ciego y sin una segunda lectura independiente que permitiera calcular concordancia entre evaluadores. Es precisamente esta característica la que impide interpretar las cifras de la sección 4.3.3 como una medida de acuerdo con el juicio experto, y la que motiva el panel de juicio experto ciego declarado como trabajo futuro en las secciones 4.3.5 y 5.5.

4.3.2 Hallazgo: No Conformidad de Vocabulario en el Veredicto Agregado

Antes de interpretar cualquier cifra de precisión, dos fuentes de datos independientes revelaron un problema de robustez en el motor de admisibilidad de vocabulario, no de contenido. La primera fue el registro de auditoría del sistema en producción (RegistroInteraccionDB): de 116 registros acumulados a la fecha generados por pruebas internas repetidas sobre la convocatoria de prueba [TEST-SUS] Semilla Inicia, cada uno evaluado de forma independiente por corresponder a un estado distinto de una propuesta en evolución, 22 correspondieron a llamadas de análisis de admisibilidad con un estado_general registrado. De esos 22, solo 10 (45,5%) coincidían exactamente con uno de los tres valores documentados en el *schema* (ADMISIBLE, INADMISIBLE, REQUIERE_MEJORAS); los 12 restantes (54,5%) eran variantes generadas por Gemini fuera de ese vocabulario NO_ADMISIBLE, "NO ADMISIBLE" con espacio en vez de guion bajo, NO_CUMPLE, CUMPLE y PARCIALMENTE_ADMISIBLE, y en un caso adicional el campo criticidad apareció en inglés (HIGH/MEDIUM) en vez del español documentado (ALTA/MEDIA/BAJA).

La segunda fuente la corrida piloto del set 5N descrito en 4.3.1, antes de aplicar ninguna corrección confirmó el hallazgo a mayor escala y bajo condiciones controladas: de 40 observaciones de estado_general, solo 5 (12,5%) calzaban exactamente con el vocabulario documentado, y el valor INADMISIBLE no apareció ni una sola vez en las 40 corridas el modelo prefirió sistemáticamente variantes no documentadas como NO_ADMISIBLE, PARCIAL, NO_CUMPLE, PARCIALMENTE_ADMISIBLE o ADMISIBLE_CON_OBSERVACIONES.

Se trata de una alucinación de vocabulario, no de contenido: el veredicto de fondo que la IA construye puede ser razonable, y aun así queda expresado en un estado que ni la función de reconciliación (3.5.1) ni el *frontend* que gatilla la vista de éxito comparando textualmente

contra "ADMISIBLE" reconocen, dejando al usuario atascado en una postulación que podría ya cumplir todos los criterios. La causa técnica es que `estado_general` y `criticidad` estaban tipados como `str` simple en `app/schemas/ai_responses.py`, sin un `Literal` o `Enum` que restringiera el vocabulario aceptado (sección 3.6.1): Pydantic validaba la presencia y el tipo de cada campo, pero no su vocabulario. A diferencia del resto de esta sección, este hallazgo no dependió del juicio del experto ni de datos adicionales: fue una brecha de robustez detectable por conteo de cadenas sobre datos ya recolectados.

4.3.3 Correcciones Aplicadas y Resultados de la Corrida Piloto

El hallazgo anterior motivó dos correcciones sobre el motor de admisibilidad, aplicadas antes de reportar cualquier cifra de precisión como válida. La primera fue un validador de normalización (Pydantic, modo "before") en `app/schemas/ai_responses.py` que mapea las variantes de vocabulario observadas `NO_ADMISIBLE`, `"NO ADMISIBLE"`, `NO_CUMPLE`, `PARCIAL`, `PARCIALMENTE_ADMISIBLE`, `ADMISIBLE_CON_OBSERVACIONES`, `CUMPLE`, y los equivalentes en inglés de criticidad a los valores canónicos, en vez de restringir el tipo sin normalizar, lo que habría descartado análisis cualitativos completos y válidos solo por un desajuste de vocabulario. Cualquier variante no reconocida cae de forma conservadora a `REQUIERE_MEJORAS`, nunca a `ADMISIBLE`, consistente con un criterio de costo asimétrico de errores: un falso `ADMISIBLE`, donde el sistema aprueba un proyecto que un evaluador rechazaría, es más costoso para el usuario que un falso `REQUIERE_MEJORAS`, donde el sistema pide mejoras a un proyecto ya aceptable.

Al re-ejecutar el set 5N con esta corrección, la conformidad de vocabulario subió de 12,5% a 100% pero la precisión de admisibilidad reportada bajó de 100% a 66,7%, mientras que la de inadmisibilidad subió a 100%. Este resultado no fue un efecto espurio de la corrección: como el *harness* solo clasificaba un caso como "inadmisibile detectado por IA" cuando el texto coincidía exactamente con `"INADMISIBLE"` *string* que antes casi nunca aparecía, la detección de inadmisibilidad dependía casi enteramente de la capa determinística, ocultando desacuerdos reales del juicio semántico de la IA. Con el vocabulario corregido, esos desacuerdos se hicieron visibles y expusieron una segunda inconsistencia: en el caso C1 (esperado `ADMISIBLE`), Gemini declaró `INADMISIBLE` con puntaje 60/100 y cero filtros marcados `NO_CUMPLE` una contradicción interna entre el veredicto agregado y el detalle por filtro que la función de reconciliación original no corregía, porque `_reconciliar_estado_admisibilidad` (`ai_service.py`, sección 3.5.1) solo promovía hacia `ADMISIBLE` cuando el detalle lo respaldaba, sin una regla simétrica para un `INADMISIBLE` que el propio detalle contradecía.

Esta segunda corrección se aplicó agregando esa regla simétrica: un `INADMISIBLE` sin ningún filtro en `NO_CUMPLE` ni documento obligatorio `AUSENTE` se corrige hacia `REQUIERE_MEJORAS`, nunca al revés. Con ambas correcciones aplicadas, la corrida final del set 5N (20 casos, 40 observaciones, contra Gemini en vivo) alcanzó 100% de conformidad de vocabulario, 100% de precisión en la detección de casos `ADMISIBLE`, 100% en `INADMISIBLE`, y un *recall* promedio de criterios bloqueantes de 91%, sin ningún *mismatch* entre el estado esperado y el detectado en los 20 casos. Estas cifras deben leerse con la

misma cautela que motivó este ejercicio: se calculan contra las etiquetas propias definidas al construir el set, no contra un juicio ciego independiente, por lo que constituyen evidencia de que el motor de admisibilidad y el instrumento de medición funcionan correctamente de punta a punta frente a un conjunto amplio y diverso de escenarios no una confirmación de que el juicio semántico de la IA coincide en cada caso con el de un evaluador humano experto, distinción que un panel de juicio experto ciego permitiría zanjar (sección 4.3.5).

4.3.4 Caso Real Complementario: MunIA (Start-Up Chile)

Dado que Admisivo no cuenta a la fecha con usuarios reales activos en producción ni con acceso sistemático a postulaciones de PYMEs en curso, el conjunto de casos 5N descrito en 4.3.1 se construye de forma sintética pero anclada a datos reales de bases CORFO, siguiendo el mismo criterio ya aplicado en el perfil ficticio del estudio SUS (sección 4.2.1). Se cuenta, sin embargo, con un caso real puntual: el autor cuenta con una postulación real propia a Start-Up Chile (código 26BLD-318143, rechazada el 26-06-2026), con acceso tanto al formulario enviado como al informe oficial de evaluación. Es importante precisar el alcance de este caso: Start-Up Chile no evalúa admisibilidad estructural contra un *checklist*, como sí hacen las convocatorias modeladas en el set 5N, sino mérito mediante ranking competitivo entre 821 postulaciones, calificado por 279 evaluadores. MunIA superó la admisibilidad (674/821 admisibles) y fue rechazado en la fase de mérito (puesto 127, nota final 4,68 frente a un promedio de 5,30 entre los proyectos que avanzaron), con Escalabilidad (4,00), Potencial de crecimiento (4,13) y Tracción y/o validación comercial (4,25) como los criterios más débiles, e Impacto en Chile (5,50) como el más fuerte. Este caso, por lo tanto, no prueba si Admisivo clasifica correctamente ADMISIBLE/INADMISIBLE dimensión que no modela el ranking competitivo, sino si su análisis cualitativo (mecanismo de 3.5.1) detecta las mismas debilidades que detectaron los evaluadores reales, antes de que la propuesta fuera enviada, y se reporta como evidencia complementaria al set 5N, no como sustituto de un panel amplio de proyectos reales.

Se construyó un *checklist* modelado sobre los 7 criterios reales de evaluación citados en el informe oficial (Participantes; Valor del Proyecto desagregado en oportunidad de mercado, producto, tracción e impacto en Chile; Potencial de Crecimiento desagregado en escalabilidad y estrategia) y se transcribió el formulario real de MunIA sin editar, corriéndolo dos veces por analizar_admisibilidad_con_checklist (script tests/validation/caso_real_startupchile_munia.py). El resultado es una validación parcial, no una confirmación limpia: Admisivo coincidió con los evaluadores reales en "Participantes" (PARCIAL en ambas corridas, con una razón textual que menciona casi literalmente la falta de "experiencia específica del equipo fundador para liderar un crecimiento global" señalada en la carta oficial) y en "Estrategia de crecimiento" (PARCIAL en ambas corridas, coherente con que Potencial de Crecimiento fue la segunda categoría más débil). Sin embargo, marcó "Tracción y validación comercial" como CUMPLE en ambas corridas, pese a ser uno de los tres criterios reales más débiles (4,25) un falso negativo del sistema respecto del juicio experto real. Adicionalmente, "Escalabilidad" el criterio real más débil de todos (4,00) fue PARCIAL en la primera corrida, pero CUMPLE en la segunda, pese a que la temperatura 0.2 usada en este tipo de análisis está diseñada específicamente para

minimizar esa variación (sección 3.6.2). Este único caso no permite generalizar una tasa de acierto, y no sustituye a un panel de juicio experto ciego sobre el set 5N evalúa mérito competitivo, no el *checklist* estructural, pero es, dentro de las restricciones de esta memoria, la única fuente disponible de juicio experto real y externo sobre el desempeño semántico del sistema. Confirma la misma conclusión que motiva la línea de trabajo futuro descrita en la sección 4.3.5: antes de confiar en la precisión semántica de Admisivo a escala se necesita un panel de casos más amplio evaluado por un experto ciego.

4.3.5 Alcance y Limitaciones

El set 5N valida el motor de admisibilidad y el instrumento de medición de punta a punta frente a un conjunto amplio y diverso de escenarios, pero lo hace contra una verdad base definida por el propio autor, no contra un juicio independiente. El caso real complementario (sección 4.3.4) aporta el único contraste disponible con un juicio humano externo, aunque sobre un instrumento de evaluación distinto (mérito competitivo, no *checklist* estructural). Cerrar esa brecha requiere un panel de juicio experto ciego sobre el set 5N evaluando matriz de confusión, coeficiente Kappa de Cohen y tasa de falsos positivos críticos junto con la clasificación experta de las sugerencias reales del Taller IA, ambos declarados trabajo futuro prioritario (sección 5.5). Por las mismas razones de alcance y tiempo, tampoco se ejecutan en esta versión la obtención de un panel amplio de proyectos reales en curso ni la comparación cuantitativa entre el motor de *matching* de Admisivo y el buscador oficial de CORFO, sin embargo, ambas son propuestas como trabajo futuro (sección 5.5). Las razones específicas de estas limitaciones y su priorización se discuten en el Capítulo 5.

4.4 Síntesis de la Validación

Los resultados disponibles a la fecha permiten afirmar, con evidencia cuantitativa, que Admisivo es percibido por sus usuarios como un sistema **altamente usable** (SUS = 91.25, categoría Excelente) y **particularmente fácil de aprender** (subescala de Aprendibilidad = 96.88), lo que respalda de forma directa el objetivo específico 2 (diseñar una plataforma que facilite la estructuración, revisión y mejora de postulaciones de forma escalable) y el objetivo específico 4 (motor de asistencia con retroalimentación en tiempo real), en la medida en que los usuarios lograron completar el flujo completo de cinco tareas desde el descubrimiento de un fondo hasta la edición asistida en el Taller IA sin intervención del moderador.

Sin embargo, esta validación es deliberadamente parcial: un puntaje SUS alto certifica que la interfaz no es una barrera para el usuario, pero no certifica que los diagnósticos de admisibilidad que entrega sean correctos. Ambas afirmaciones son lógicamente independientes, y solo la segunda validada en la sección 4.3 contra una verdad base propia y un único caso real, sin el panel de juicio experto ciego que dicha sección declara como trabajo futuro responde de forma completa al problema central planteado en el Capítulo 1: la reducción de la tasa de rechazo por causas formales evitables. En consecuencia, las conclusiones de este trabajo (Capítulo 5) deben leerse a la luz de esta limitación, reconociendo que tanto la validación de usabilidad como la validación técnica disponible

son condiciones necesarias, pero no suficientes para sostener la efectividad completa de la solución frente a un evaluador humano experto.

El análisis cualitativo por tarea (secciones 4.2.4 y 4.2.5) matiza además el resultado agregado: aunque los cuatro participantes lograron, con o sin ayuda, llevar su postulación a estado ADMISIBLE, se identificaron ocho hallazgos de usabilidad concretos tres de ellos de severidad alta o crítica, todos concentrados en el módulo Taller IA que no son visibles si se reporta únicamente el promedio SUS = 91.25. El más severo (contradicción de estados entre el chip "PENDIENTE REANÁLISIS" y el texto de falla original, hallazgo n.º 1) impidió que un participante completara la Tarea 5 sin intervención del moderador, y coincide exactamente con el único puntaje SUS individual que se aparta del resto de la muestra (P3, 82.5). El hallazgo n.º 4 (inconsistencia numérica del monto de cofinanciamiento entre regeneraciones sucesivas de una sugerencia) es de naturaleza distinta y más relevante aún para el objetivo específico 5: sugiere que la temperatura de 0.7 usada para diversificar la redacción (sección 3.7.2) puede introducir imprecisión en cifras factuales embebidas en el mismo texto, un riesgo que conecta directamente con la validación de precisión técnica de la sección 4.3 y que debiera incorporarse como un caso de prueba específico el día en que se ejecute el panel de juicio experto propuesto como trabajo futuro en la sección 5.5.

CAPÍTULO 5: CONCLUSIONES Y RECOMENDACIONES

5.1 Cumplimiento de los Objetivos

Esta sección revisa, uno a uno, el grado de cumplimiento de los cinco objetivos específicos definidos en la sección 1.5.2, para luego cerrar con una evaluación del objetivo general (sección 1.5.1).

El objetivo específico 1 analizar en profundidad los desafíos, brechas de información y necesidades de las PYMEs durante el proceso de formulación de postulaciones a CORFO se considera cumplido. La investigación presentada en los Capítulos 1 y 2 no quedó como un ejercicio exploratorio aislado: los patrones identificados lenguaje técnico poco accesible para un postulante primerizo, errores de lectura o de interpretación de las bases, falta de detalle en las respuestas, brechas ya documentadas también por la propia CORFO en su caracterización del ecosistema postulante (sección 2.1.1) se tradujeron directamente en el diseño de los *prompts* que sostienen el motor de admisibilidad y el Taller IA (Capítulo 3): el sistema no solo detecta estos problemas, sino que fue construido sabiendo de antemano dónde y por qué suelen aparecer.

El objetivo específico 2 diseñar una plataforma que facilite la estructuración, revisión y mejora de postulaciones de manera escalable y sistemática se cumple de forma sistemática en su totalidad: la separación de responsabilidades del *backend* en routers, servicios de IA independientes y un cron dedicado al scraping (Capítulo 3) constituye, por diseño, una estructura ordenada y mantenible para procesar postulaciones. En su dimensión de escalabilidad, en cambio, el cumplimiento es parcial: el servicio de IA (*ai_service.py*) opera hoy sobre un *ThreadPoolExecutor* síncrono, adecuado para el volumen de uso actual, pero no para un escenario de múltiples llamadas concurrentes a gran escala, donde un entorno de ejecución asíncrona dedicado ofrecería mejor rendimiento. Este punto queda declarado como trabajo futuro en la sección 5.5.

El objetivo específico 3 desarrollar un módulo de búsqueda inteligente que centralice y clasifique las convocatorias vigentes se responde directamente a través del módulo de Análisis Rápido (sección 3.4): en vez de exigir al postulante revisar convocatoria por convocatoria, el sistema centraliza la búsqueda mediante una combinación de filtros duros/blandos y la comprensión que la IA construye tanto del proyecto como de cada convocatoria. Vale la pena matizar la formulación original del objetivo: más que evaluar el perfil y las necesidades de cada empresa contra el catálogo, lo que el sistema efectivamente determina es si un proyecto dado satisface la necesidad específica que cada convocatoria busca cubrir una inversión de énfasis que refleja mejor cómo opera en la práctica un proceso de *matching* contra bases concursales, y que se considera un ajuste conceptual, no un incumplimiento del objetivo.

El objetivo específico 4 implementar un motor de asistencia basado en IA que genere retroalimentación en tiempo real para mejorar la coherencia y el cumplimiento de los puntos clave de la postulación se materializa en el Taller IA (sección 3.7). Su componente

de cumplimiento es objetivamente verificable, ya que el porcentaje de criterios de admisibilidad satisfechos según el *checklist* (sección 3.5.1) sube de forma trazable a medida que el usuario aplica sugerencias. Su componente de coherencia, en cambio, queda como una deuda metodológica: a diferencia del cumplimiento, este trabajo no desarrolló una métrica objetiva de coherencia textual que permita afirmar que las sugerencias del Taller IA la mejoran, más allá de la revisión cualitativa de casos individuales. Definir y medir esa dimensión de forma operacionalizable queda como una pregunta abierta para trabajo futuro (sección 5.5).

El objetivo específico 5 validar la precisión técnica y la fiabilidad de la solución mediante juicio de experto se cumple parcialmente: el motor de admisibilidad fue validado con rigor metodológico sobre 20 casos de prueba controlados (5N), con resultados consistentes (sección 4.3), y contrastado adicionalmente contra el juicio efectivo de 279 evaluadores de Start-Up Chile en el caso real de MunIA (sección 4.3.4). Lo que no se completó dentro del plazo disponible fue el panel de juicio experto ciego sobre el mismo set de 20 casos, que queda declarado como trabajo futuro inmediato (secciones 4.3.5 y 5.5).

Respecto al objetivo general implementar y evaluar un sistema de apoyo computacional que permita a las PYMEs optimizar el cumplimiento de criterios de admisibilidad, democratizando el acceso a información estratégica y ofreciendo asistencia experta, la evidencia reunida en esta memoria respalda que Admisivo efectivamente responde a los dolores identificados y mejora el acceso a información estratégica del ecosistema CORFO. No es posible, sin embargo, afirmar que el sistema entregue 'asistencia experta' en un sentido estricto: el modelo de lenguaje subyacente no fue entrenado específicamente para este dominio, y como se declara en la sección 5.4, sus respuestas siguen siendo de naturaleza probabilística. El objetivo general se considera, en consecuencia, cumplido con matices: el sistema ayuda, orienta y mejora la calidad del proceso de postulación de forma medible, pero no sustituye ni pretende sustituir el juicio de un experto humano.

5.2 Validación Temprana de Requerimientos

El alcance final de este trabajo un motor de admisibilidad y un taller de redacción asistida acotados al ecosistema CORFO (sección 1.5.3) es el resultado de un proceso de levantamiento empírico de requerimientos con actores del ecosistema de emprendimiento e innovación, y difiere sustancialmente de la propuesta inicial del proyecto.

La propuesta inicial no fijaba la admisibilidad como indicador de éxito, sino un objetivo más amplio y menos verificable: mejorar de forma general la calidad de las postulaciones mediante un modelo entrenado específicamente para ese propósito, sin un criterio cuantitativo de éxito definido. El levantamiento de requerimientos descrito a continuación evidenció que sostener ese alcance con los tiempos y recursos disponibles no era viable, lo que motivó su acotación hacia la admisibilidad como indicador cuantitativo y verificable una definición de éxito más restringida, pero medible, que determinó el diseño descrito en el Capítulo 3.

El levantamiento de requerimientos se realizó mediante iteración directa con dos perfiles de *stakeholder* complementarios, en el marco de un programa formal de mentoría a organizaciones vinculadas al ecosistema de innovación. El primero, una fuente institucional con experiencia directa en la gestión de alianzas estratégicas de CORFO, permitió contrastar los dolores identificados en el Capítulo 1 contra la perspectiva de la propia institución evaluadora. El segundo, un actor académico responsable de la formulación de proyectos bajo restricciones de tiempo similares a las de una PYME típica, permitió identificar limitaciones funcionales concretas de la plataforma oficial de CORFO no documentadas en las bases de postulación: ausencia de guardado automático del formulario en curso, falta de trazabilidad sobre los requisitos ya cumplidos durante el llenado, y ausencia de visibilidad centralizada de fechas de convocatoria.

Este levantamiento permitió distinguir requerimientos funcionales prioritarios que determinaron el diseño final del motor de admisibilidad y del Taller IA (Capítulo 3) de funcionalidades identificadas pero descartadas del alcance por restricciones de tiempo: un indicador consolidado de todos los ítems necesarios para completar una postulación, la gestión de antecedentes tributarios (SII), y la construcción asistida de presupuestos. Estas quedan documentadas como trabajo futuro en la sección 5.5.

5.3 Decisiones de Infraestructura Técnica

La configuración de infraestructura utilizada en producción (Capítulo 3) fue el resultado de una restricción técnica no anticipada en el diseño inicial. La primera decisión de despliegue consideró Google Cloud Platform, buscando consistencia con el uso de Gemini como proveedor de IA. Esa decisión resultó incompatible con un requisito crítico del sistema: el *scraper* de ClaveÚnica y del sistema Charly.io de CORFO (sección 3.1.4) requiere iniciar sesión de forma automatizada, y el gobierno de Chile bloquea por rango de IP los intentos de inicio de sesión que provienen de proveedores *cloud* identificados como tráfico masivo, Google Cloud entre ellos.

Se evaluaron mitigaciones alternativas, incluyendo el uso de undetected-chromedriver y la contratación de un servicio de proxy, sin resultados viables dentro de las restricciones de costo del proyecto. La solución adoptada fue migrar a Railway, un proveedor de menor escala cuyos rangos de IP no están identificados como tráfico masivo y, por lo tanto, no están sujetos al mismo bloqueo.

5.4 Limitaciones

La limitación conceptual más importante de este trabajo, y la que el autor considera necesario declarar sin matices, es que Admisivo no puede ni podrá, en ninguna versión futura razonable garantizar la admisibilidad de una postulación. Esto no es una falla de diseño ni una promesa incumplida: es una consecuencia directa de construir el motor de evaluación sobre modelos de lenguaje de naturaleza probabilística (sección 2.2), que por definición pueden fallar, y que en el mejor de los casos permiten acercarse a un óptimo, no garantizarlo. Declarar esto de forma explícita no debilita el aporte del sistema: al contrario, es lo que permite comunicarlo honestamente como una herramienta de apoyo a la decisión el alcance funcional definido desde la sección 1.5.3 y no como un oráculo infalible. Las limitaciones técnicas específicas de esta afirmación la ausencia de un panel de juicio experto, la dependencia de una verdad base propia para el set 5N, y el caso único de validación externa disponible (MunIA) ya fueron declaradas en detalle en la sección 4.3.5 y no se repiten aquí.

Una segunda limitación, de naturaleza distinta, es la ausencia de un *partnership* real con usuarios activos que permitiera una validación en condiciones de producción. Esta no fue una omisión pasiva: durante el desarrollo de este trabajo se sostuvieron conversaciones concretas con ValpoEmprende para pilotear Admisivo con postulantes reales a su línea Semilla Inicia. Ese piloto no se concretó porque, este año, ValpoEmprende no abrió convocatoria para Semilla Inicia, redirigiendo en cambio sus recursos a consolidar empresas ya formadas a través de Semilla Expande. La validación con usuarios reales en producción, por lo tanto, no se descartó por falta de gestión, sino por un cambio de prioridades de la contraparte externo al control del autor, y queda como la primera prioridad de trabajo futuro (sección 5.5).

5.5 Trabajo Futuro

A partir de las limitaciones declaradas en la sección anterior y de los hallazgos técnicos documentados a lo largo del Capítulo 4, se identifican las siguientes líneas de trabajo futuro, ordenadas de mayor a menor prioridad para el autor.

1. Validación con postulantes reales en profundidad (prioridad principal). Reemplazar o complementar el set sintético 5N con un piloto real: acompañar a múltiples postulantes que compitan por una misma convocatoria durante todo su ciclo real de postulación que, según la experiencia recogida durante este trabajo, puede tomar entre 2 y 4 meses considerando la preparación previa de la solicitud, no solo el plazo formal de postulación. Esta es la línea de trabajo que el autor priorizaría por sobre cualquier otra, incluyendo el panel de juicio experto descrito a continuación, porque permitiría validar el sistema completo (admisibilidad y Taller IA) contra resultados reales de adjudicación, no solo contra un *checklist* estructural.
2. Panel de juicio experto ciego sobre el set 5N y verificación de alucinaciones del Taller IA: un experto en formulación y/o evaluación de proyectos CORFO evaluaría, sin conocimiento

previo del veredicto del sistema, los 20 casos de prueba (sección 4.3.1) y clasificaría como Correcta, Errónea o Incompleta una muestra de sugerencias reales del Taller IA ya disponibles en el registro de auditoría de las sesiones SUS (sección 4.2). Las métricas de salida serían la matriz de confusión, el coeficiente Kappa de Cohen y la tasa de falsos positivos críticos casos ADMISIBLE del sistema que el experto determina INADMISIBLE, el error de mayor costo para el usuario final.

3. Refinamiento del motor de IA una vez disponible una métrica de fiabilidad basada en juicio experto: ajuste de *prompts* o evaluación de modelos alternativos al actualmente utilizado (sección 3.6), optimizando específicamente contra los desacuerdos que el panel de juicio experto logre identificar.

4. Expansión a otros fondos públicos más allá de CORFO por ejemplo, Sercotec o FOSIS, excluidos del alcance original por la sección 1.5.3. La arquitectura del sistema no requiere cambios estructurales para esto: la extensión implica principalmente construir un nuevo *scraper* adaptado al formulario de cada nuevo instrumento, ya que el resto de la plataforma (motor de admisibilidad, Taller IA, *matching*) está diseñado para operar sobre cualquier *checklist* de criterios, no solo sobre los de CORFO.

5. Funcionalidades identificadas durante el levantamiento de requerimientos, pero descartadas del alcance por restricciones de tiempo (sección 5.2): un indicador consolidado de todos los ítems requeridos para completar una postulación, gestión de antecedentes tributarios (SII) y apoyo en la construcción de presupuestos.

6. Definición y medición operacional de la coherencia de una postulación (objetivo específico 4, sección 5.1), dimensión que este trabajo no logró convertir en una métrica objetiva más allá de la revisión cualitativa de casos.

7. Migración del servicio de IA a un modelo de ejecución asíncrono dedicado, para escalar el objetivo específico 2 (sección 5.1) de un `ThreadPoolExecutor` síncrono a un entorno capaz de sostener múltiples llamadas concurrentes a mayor volumen.

8. Comparación cuantitativa entre el motor de *matching* de Admisivo y el buscador oficial de CORFO, propuesta originalmente como un cuarto eje de validación y no ejecutada en esta versión (sección 4.3.5).

9. Evolución hacia una plataforma de gestión y seguimiento de postulaciones para terceros incubadoras, aceleradoras o programas de mentoría como el descrito en la sección 5.2 que permita *trackear* el progreso de múltiples clientes o postulantes de forma centralizada, extendiendo el caso de uso individual actual hacia un caso de uso institucional.

ANEXOS

Anexo A: Recorrido visual de la plataforma

Este anexo documenta visualmente las tres funcionalidades núcleo descritas en el Capítulo 3, capturadas sobre el mismo caso ficticio “StockPro Inteligente” utilizado en la prueba de usabilidad (sección 4.2). Las capturas siguen el flujo real de un postulante: descubrimiento de la convocatoria, formulario dinámico, diagnóstico de admisibilidad y edición asistida en el Taller IA.

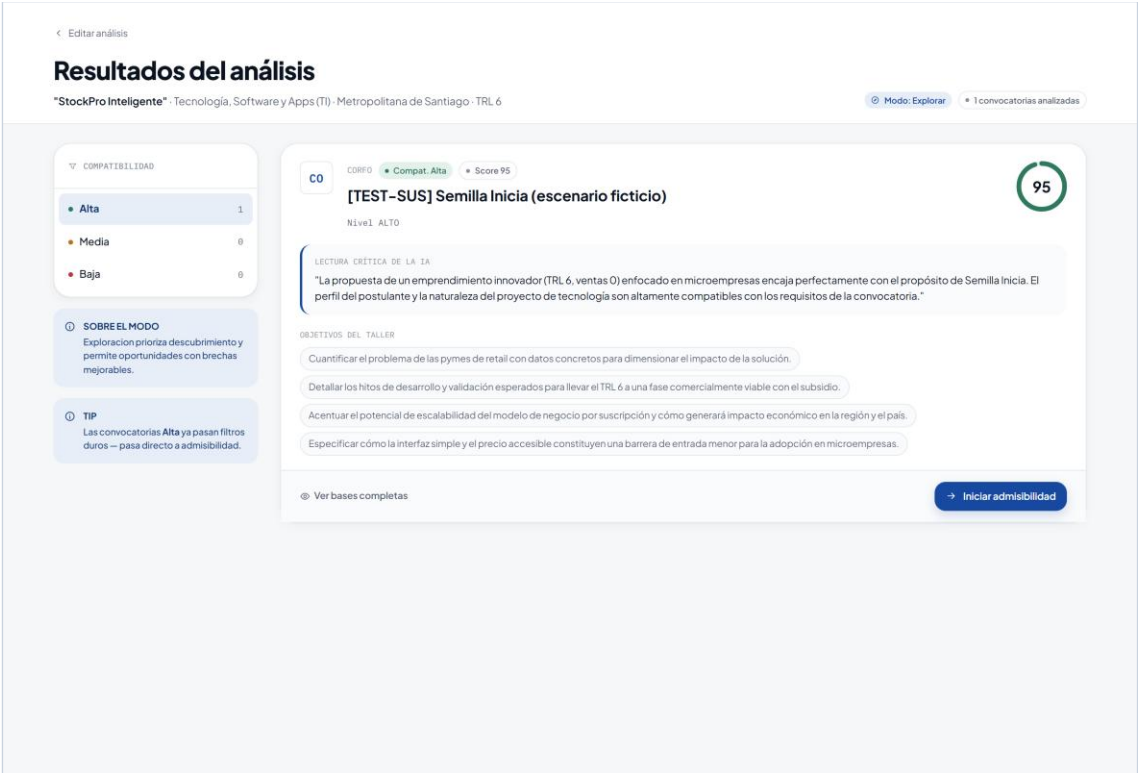


Figura 9: Módulo de Análisis Rápido
Fuente: Elaboración propia

< Cambiar convocatoria

Convocatorias > [TEST-SUS] Semilla Inicia (escenario ficticio)

Formulario · [TEST-SUS] Semilla Inicia (escenario ficticio)

→ Al terminar verás tu análisis de admisibilidad.

Formulario oficial CORFO

Extraído automáticamente

Formulario extraído directamente del sistema de postulación de CORFO

Sección 01 / 04

Datos de la Postulación

El Problema y la Solución

Equipo y Escalabilidad

Datos Críticos Financieros

01 DATOS DE LA POSTULACIÓN

1 / 4 · 25%

1. Título del Proyecto *

Ej: StockPro Inteligente

4. Monto Solicitado a CORFO (CLP) *

Monto máximo investigado en las bases...

2. Región de Ejecución *

Seleccione una opción

3. Tipo de Postulante *

Seleccione una opción

< Anterior

Guardar progreso

> Continuar a El Problema y la Solución

Fuente: Elaboración propia

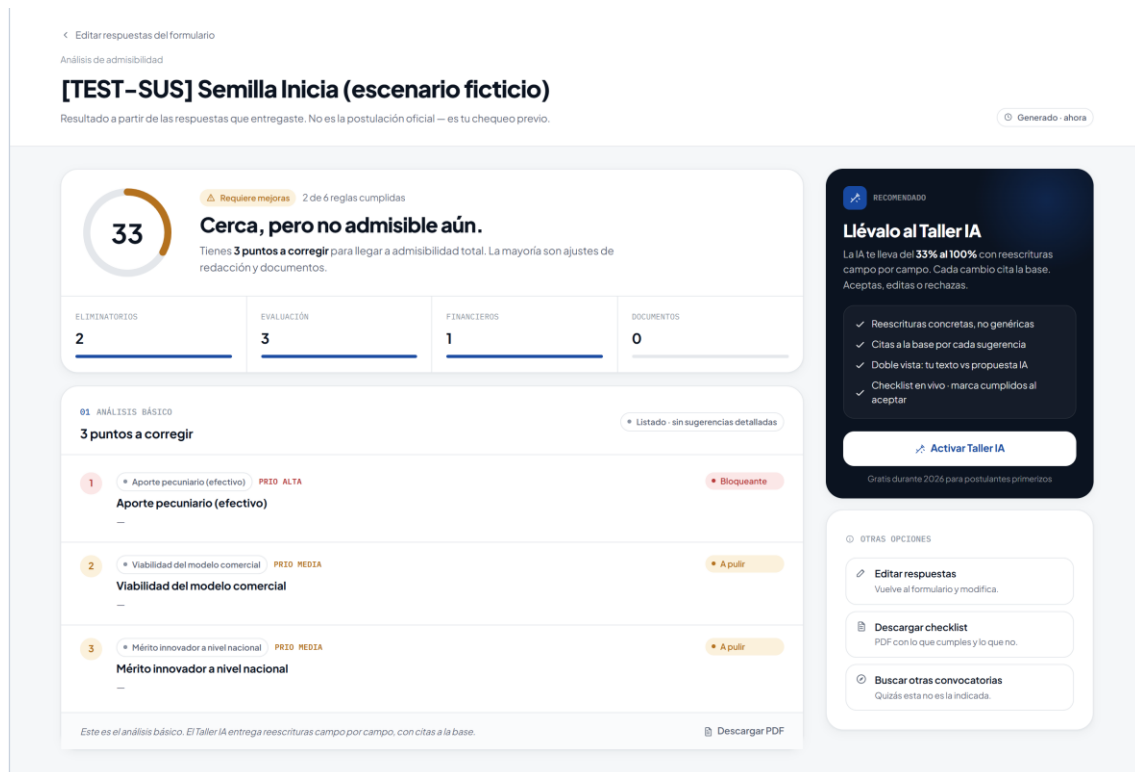


Figura 11: Diagnóstico de admisibilidad
Fuente: Elaboración propia

SISTEMA DE APOYO BASADO EN INTELIGENCIA ARTIFICIAL GENERATIVA PARA MEJORAR LA ADMISIBILIDAD DE POSTULACIONES DE PYMES A FONDOS CORFO

Taller IA (TEST-SUS) Semilla Inicia (escenario ficticio)

ADMISIBILIDAD 33% 33 Reanudar

2 / 6 reglas cumplidas

Pulamos tu propuesta, sección por sección

PUNTOS CRÍTICOS

Filtros eliminatorios 2/2

- Tipo de postulante admisible
- Limite de ventas previas

Filtros de evaluación 6/3

- Dedicación del equipo emprendedor
- Mérito innovador a nivel nacional
- Viabilidad del modelo comercial

Filtros financieros 6/3

- Cofinanciamiento mínimo (25%)

81 bloque 02 / 04 < Datos de la Postulación 2 El Problema y la Solución Equipo y Enfoque >

TU BORRADOR

Tu respuesta original

5. Describe el problema u oportunidad

Muchas pymes de retail no tienen visibilidad de su stock en tiempo real y pierden ventas por quiebres o sobre-stock.

6. Describe su solución y por qué es innovadora a nivel nacional

Una app de inventario que se controla desde el celular, permitiendo registrar entradas y salidas de stock al instante.

ADMISIVO 1A

Edita para mejorar

5. Describe el problema u oportunidad

Muchas pymes de retail no tienen visibilidad de su stock en tiempo real y pierden ventas por quiebres o sobre-stock.

1A MARCA ESTE CAMPO - PARCIAL

La descripción de la solución (Una app de inventario que se controla desde el celular...) es genérica y no detalla la novedad tecnológica significativa a nivel nacional, ni explica por qué no es una solución ampliamente disponible en el mercado.

PROPUESTA DE LA 1A

Aplicar el campo

Pedir otra versión (3) Copiar

Nuestra solución, StockPro Inteligente, es una aplicación móvil de gestión de inventario que integra algoritmos de IA para la predicción de demanda y optimización de stock en tiempo real, algo no disponible de forma accesible para pymes en el mercado chileno. A diferencia de sistemas ERP complejos, StockPro ofrece una interfaz intuitiva y móvil que permite a los pequeños retailers registrar entradas y salidas al instante, minimizando quiebres de stock y sobre-inventario de manera proactiva, lo que representa una innovación significativa a nivel nacional para este segmento.

6. Describe su solución y por qué es innovadora a nivel nacional

Una app de inventario que se controla desde el celular, permitiendo registrar entradas y salidas de stock al instante.

ÚLTIMA ADMISIBILIDAD EVALUADA

33% 2 de 6 reglas

NO CUMPLE

- Tipo de postulante admisible
- Limite de ventas previas
- Dedicación del equipo emprendedor
- Mérito innovador a nivel nacional
- Viabilidad del modelo comercial

Figura 12: Taller IA
Fuente: Elaboración propia

SISTEMA DE APOYO BASADO EN INTELIGENCIA ARTIFICIAL GENERATIVA PARA MEJORAR LA ADMISIBILIDAD DE POSTULACIONES DE PYMES A FONDOS CORFO

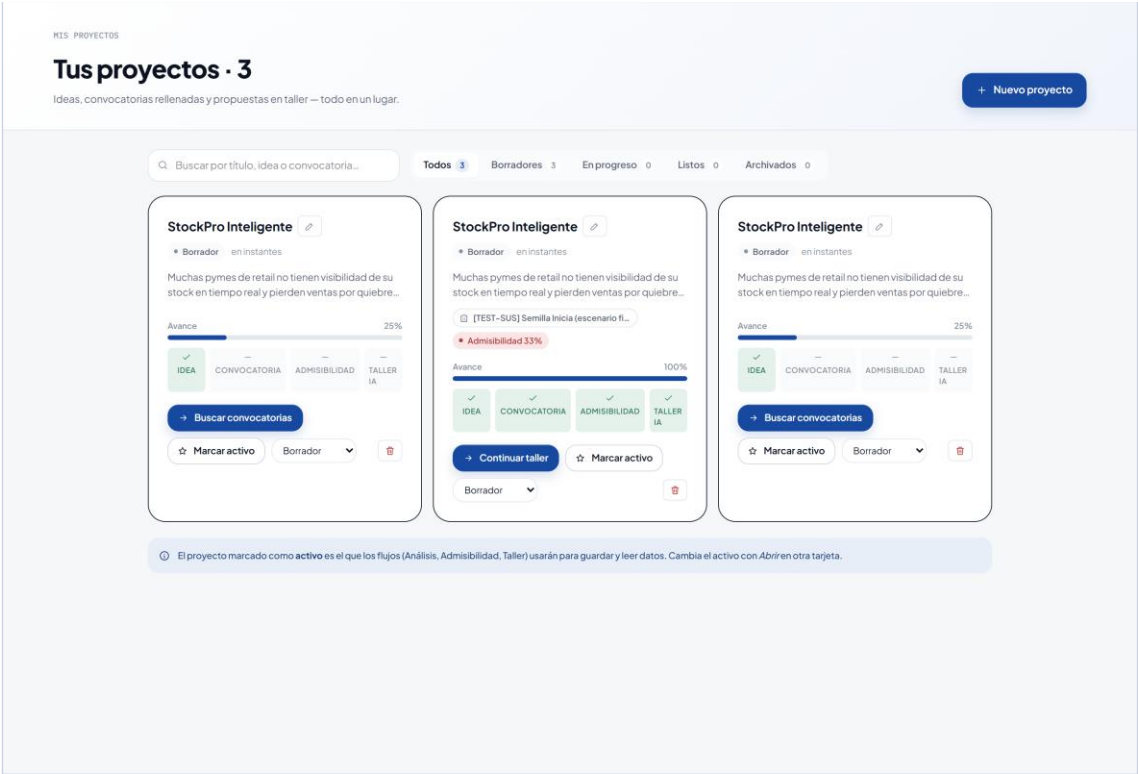


Figura 13: Panel de proyectos
Fuente: Elaboración propia

REFERENCIAS BIBLIOGRÁFICAS

- [1] Corfo, *Corfo | PostulaBien! | Charla 2 "Errores comunes al momento de postular"*, (17 de agosto de 2023). Accedido: 12 de agosto de 2025. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=KDXZ19bX6kY>
- [2] «Las incubadoras de negocios dan el salto: preparan fondos privados y se expanden para apoyar a las pymes». Accedido: 12 de agosto de 2025. [En línea]. Disponible en: <https://www.fintechile.org/noticias/las-incubadoras-de-negocios-dan-el-salto-preparan-fondos-privados-y-se-expanden-para-apoyar-a-las-pymes>
- [3] «Asesorías CORFO – Asesorías CORFO». Accedido: 12 de agosto de 2025. [En línea]. Disponible en: <https://asesoriascorfo.cl/>
- [4] Corfo, *Corfo | Postula Bien Semilla Inicia: "Orientación sobre errores comunes al momento de postular"*, (27 de marzo de 2024). Accedido: 22 de enero de 2026. [En línea Video]. Disponible en: <https://www.youtube.com/watch?v=3Q4KtsHFAyI>
- [5] «DataInnovación - InnovaChile Corfo». Accedido: 29 de junio de 2025. [En línea]. Disponible en: <https://datainnovacion.cl/convocatorias>
- [6] «2025_BASES_Scale-ME.pdf». Accedido: 19 de agosto de 2025. [En línea]. Disponible en: https://me.cl/wp-content/uploads/2025/01/2025_BASES_Scale-ME.pdf
- [7] «CORFO». Accedido: 27 de agosto de 2025. [En línea]. Disponible en: <https://corfo.cl/sites/cpp/>
- [8] A. Vaswani et al., «Attention Is All You Need», en *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, Long Beach, CA, EE. UU., 2017, pp. 5998-6008.
- [9] «OpenAI para empresas». Accedido: 19 de agosto de 2025. [En línea]. Disponible en: <https://openai.com/es-419/business/>
- [10] «Ayuda de Aplicaciones con Gemini». Accedido: 27 de agosto de 2025. [En línea]. Disponible en: <https://support.google.com/gemini?hl=es-419&jid=5288930114617748420-NC#topic=15280100>
- [11] J. Wei et al., "Chain-of-Thought Prompting Elicits Reasoning in Large Language Models," en *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 24824-24837, 2022.
- [12] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," en *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 9459-9474, 2020.
- [13] J. Brooke, "SUS: A quick and dirty usability scale," en *Usability Evaluation in Industry*, P. W. Jordan, B. Thomas, B. A. Weerdmeester, y A. L. McClelland, Eds. Londres, Reino Unido: Taylor & Francis, 1996.
- [14] A. Bangor, P. Kortum, y J. Miller, "An Empirical Evaluation of the System Usability Scale," *International Journal of Human-Computer Interaction*, vol. 24, n.º 6, pp. 574-594, 2008.
- [15] J. Nielsen y T. K. Landauer, "A mathematical model of the finding of usability problems," en *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, Ámsterdam, Países Bajos, 1993, pp. 206-213.
- [16] J. R. Lewis y J. Sauro, "The Factor Structure of the System Usability Scale," en *Human Centered Design*. Berlín, Heidelberg: Springer, 2009, pp. 94-103.

[17] J. Nielsen, "Severity Ratings for Usability Problems," *Nielsen Norman Group*, 1994. [En línea]. Disponible en: <https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/> [Accedido: 10 de julio de 2026].