# Propuesta Técnica: Sistema Inteligente de Consulta de Subvenciones

## Sistema RAG (Retrieval-Augmented Generation) Integrado a la Plataforma Artisting

**Para:** Artisting

**De:** Cristian Rojas

**Fecha:** 28 de Noviembre de 2025

**Asunto:** Implementación de sistema de IA para búsqueda y consulta inteligente de convocatorias de subvenciones, integrado al sitio web existente de Artisting

---

## Resumen

Esta propuesta detalla la implementación de un sistema de Inteligencia Artificial que transformará aproximadamente **143.000 PDFs** de convocatorias de subvenciones en una base de datos consultable e inteligente, **integrado directamente al sitio web existente de Artisting**. El sistema reemplazará el motor RAG legal actual con una arquitectura especializada para subvenciones, manteniendo la infraestructura de autenticación, billing (Stripe) y diseño de la plataforma.

A diferencia de una búsqueda tradicional por palabras clave, el sistema será capaz de **entender el contexto y el significado** de cada convocatoria. Lo que permitirá responder **preguntas complejas y específicas**, siempre citando las bases y artículos concretos de los documentos originales.

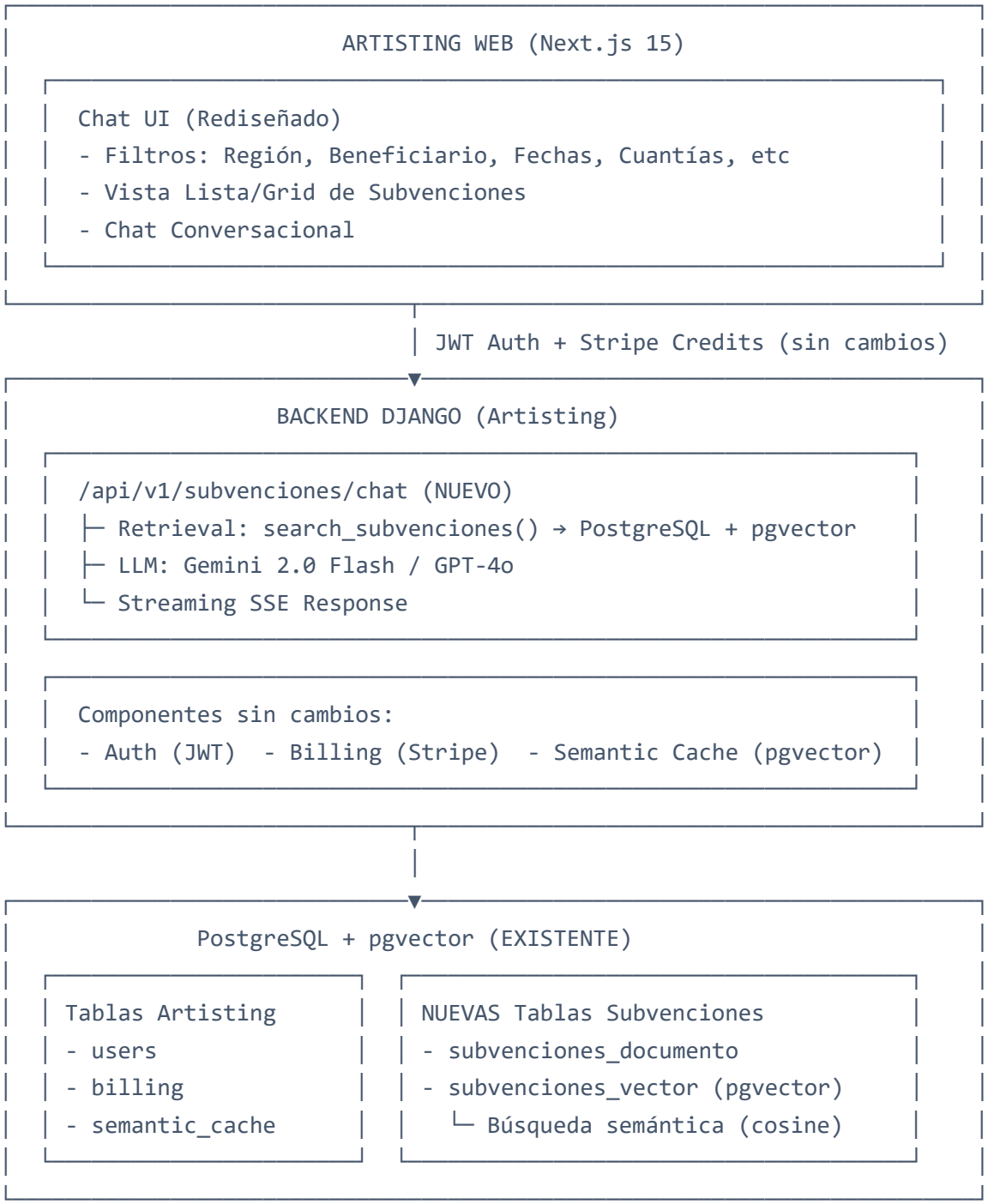
Un "GPT experto" alimentado directamente con los **90 GB** de texto completo de todas las convocatorias supondría un coste inicial y mensual muy elevado. Con esta arquitectura a medida, en cambio, entrenamos al modelo para identificar la información clave mediante un **resumen experto** de cada convocatoria (una pequeña porción del documento original). Solo ese resumen se vectoriza para la búsqueda semántica y, una vez localizadas las convocatorias relevantes, otro módulo recupera la totalidad de la información necesaria del PDF y genera la respuesta final con un modelo avanzado. Este enfoque nos permite **reducir el coste inicial en torno a un 90%** y disminuir de forma muy significativa el coste mensual, ya que la búsqueda se realiza sobre una base de datos optimizada de unos **9 GB**, y no sobre los 90 GB de texto bruto.

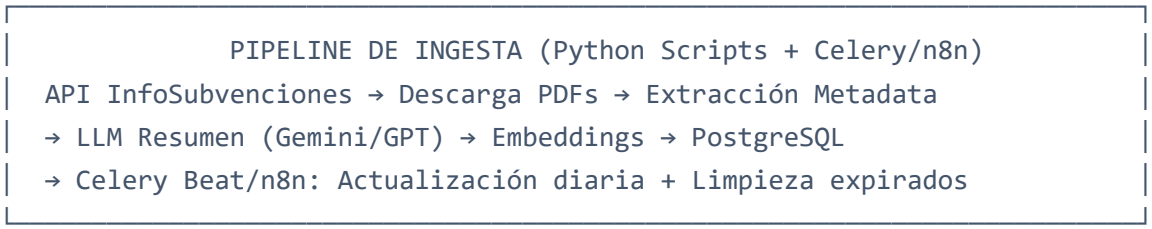
**Enfoque de desarrollo:** Comenzaremos con un prototipo funcional (1,000 PDFs) para validar la precisión del sistema antes de procesar el volumen completo de información.

---

# 1. Arquitectura del Sistema

El sistema se compone de los siguientes módulos principales, integrados con la plataforma Django/Next.js existente de Artisting:






## 1.1. Sistema de Ingesta Inteligente

**Objetivo:** Descargar, procesar y almacenar las convocatorias de subvenciones.

**Proceso:**

1. Conexión a la API de InfoSubvenciones
2. Descarga de PDFs de convocatorias
3. Extracción de metadatos estructurados (región, beneficiarios, plazos, cuantías)
4. Generación de resúmenes de alta calidad usando **Gemini 2.5 Flash-Lite**
5. Deduplicación (evitar procesar documentos repetidos)

**Tecnologías:**

- Python para scripts de automatización
- API oficial de InfoSubvenciones
- Gemini 2.5 Flash-Lite (Google AI)
- Checksum MD5 para deduplicación

## 1.2. Sistema de Vectorización

**Objetivo:** Convertir el texto en representaciones matemáticas que permitan búsquedas por significado.

**¿Qué son los embeddings?** Los embeddings convierten palabras y frases en coordenadas numéricas en un espacio matemático multidimensional. Esto permite que el sistema entienda el *significado* del texto, no solo las palabras exactas:

- **Búsqueda tradicional:** Solo encuentra documentos con las palabras exactas ("ayuda para contratar personal")
- **Búsqueda semántica:** Encuentra documentos con significados similares, aunque usen palabras diferentes ("subvención para staff", "apoyo para empleados", "financiación para contratación")

Esta capacidad permite que el sistema comprenda las matices del lenguaje y encuentre información relevante incluso cuando la pregunta del usuario no coincide palabra por palabra con el documento.

**Estrategia de optimización:** En lugar de vectorizar los 90GB completos de PDFs (extremadamente costoso), solo vectorizamos los resúmenes de 200 palabras generados por el LLM. Esto reduce los costes de procesamiento en más del 90% manteniendo la precisión.

**Tecnología:**

- Gemini Embedding 001 (Google AI), o similar

## 1.3. Base de Datos Híbrida (PostgreSQL + pgvector)

**Objetivo:** Almacenar información estructurada y vectorial en un solo lugar, aprovechando la infraestructura existente de Artisting.

La base de datos combina dos tipos de búsqueda:

| Tipo de Dato | Columnas | Función |
|---|---|---|
| **Datos estructurados (SQL)** | región, fecha_límite, tipo_beneficiario, cuantía, número_convocatoria | Filtros precisos (ej: "solo Madrid", "solo PYMES") |
| **Datos vectoriales** | embedding del resumen | Búsqueda por significado (ej: "ayudas para teatro") |
| **Documento fuente** | PDF completo o enlace | Fuente de verdad para la respuesta final |

**Integración con infraestructura existente:**

- **PostgreSQL con extensión pgvector** (ya presente en Artisting para semantic cache)
- Reutilización de la base de datos existente, añadiendo nuevas tablas para subvenciones
- Compatible con el stack actual: Django REST, Celery, Redis

## 1.4. Sistema de Consulta Inteligente ("El Bibliotecario")

**Objetivo:** Responder preguntas del usuario con precisión y citar las fuentes.

**Proceso en 4 pasos:**

1. **Filtro SQL:** Aplicar filtros duros según los criterios del usuario (región, tipo de beneficiario, fechas)
2. **Búsqueda semántica:** Entre los documentos filtrados, buscar los 3-5 más relevantes usando vectores
3. **Recuperación del documento original:** Obtener el PDF completo o las secciones relevantes directamente de la página oficial
4. **Generación de respuesta:** Se usa un modelo más potente, se lee el documento original y genera una respuesta precisa con citación de artículos específicos

**Ejemplo de consulta:**

Usuario: "¿Qué ayudas hay para autónomos de artes escénicas en Madrid que cierren en 2026?"

Sistema:

1. Filtra: región = Madrid, beneficiario = autónomos, deadline > 2026
2. Busca semánticamente: "artes escénicas"
3. Identifica las convocatorias que aplican, Recupera los 3 PDFs más relevantes
4. Responde: "La convocatoria 'Cultura Madrid 2026' (ID: 12345) cubre proyectos de artes escénicas. Según el Artículo 5.2, autónomos pueden solicitar hasta 15.000€. Plazo: 31/03/2026. [enlace al PDF]"

## 1.5. Sistema de Actualización Continua

**Objetivo:** Mantener la base de datos actualizada automáticamente.

**Funciones:**

- Consulta diaria a la API de InfoSubvenciones para detectar nuevas convocatorias
- Procesamiento automático de nuevas subvenciones (sumariza, ingesta + vectorización)
- Detección y eliminación de convocatorias expiradas (basadas en fechas límite). Remueve todos los vectores de la base de datos para mantener la calidad de la información

## 1.6. Interfaz de Usuario (Integrada a Artisting)

**Objetivo:** Reemplazar la interfaz de chat legal actual con una experiencia especializada para subvenciones.

**Implementación:**

- **Rediseño completo del frontend de chat** (manteniendo estilos consistentes con Artisting)
- **Nuevas funcionalidades específicas para subvenciones:**
  - Selectores/filtros interactivos (región, tipo de beneficiario, fechas límite, cuantías)
  - Vista de listado de subvenciones relevantes
  - Chat conversacional para consultas en lenguaje natural
  - Citación directa con enlaces a PDFs oficiales
- **Integración con backend Django:**
  - Reutilización de autenticación JWT existente
  - Sistema de créditos Stripe sin modificaciones
  - API REST endpoints nuevos bajo `/api/v1/subvenciones/`
- **Next.js 15 App Router** (tecnología actual del frontend)

---

# 2. Modelos de Lenguaje y Alternativas

## Opciones de Modelos de IA

El sistema soporta dos proveedores principales de modelos, con flexibilidad para ajustar según preferencias de coste/calidad:

**Opción A: Google AI (Gemini)**

| Modelo | Uso | Coste por 1M tokens |
|---|---|---|
| **Gemini 2.0 Flash** | Generación de resúmenes (ingesta masiva) | Input: $0.10 / Output: $0.40 |
| **Gemini Embedding 004** | Vectorización de texto | $0.02 |
| **Gemini 2.0 Flash** o **Pro** | Respuestas finales al usuario | Flash: $0.30/$2.50 - Pro: $1.25/$10.00 |

**Opción B: OpenAI**

| Modelo | Uso | Coste por 1M tokens |
|---|---|---|
| **GPT-4o-mini** | Generación de resúmenes (ingesta masiva) | Input: $0.15 / Output: $0.60 |
| **text-embedding-3-small** | Vectorización de texto | $0.02 |
| **GPT-4o** | Respuestas finales al usuario | Input: $2.50 / Output: $10.00 |

**Recomendación:** Se propone utilizar **Gemini 2.0 Flash** para optimizar costes, con la posibilidad de cambiar a GPT-4o si se requiere mayor precisión en dominios específicos.

## Alternativas de Embeddings

- **Google:** Gemini Embedding 004 (multilingual, optimizado para RAG)
- **OpenAI:** text-embedding-3-small/large (probados en producción)
- **Cohere:** embed-multilingual-v3.0 (excelente para español)

**Nota sobre configurabilidad:** La arquitectura permite cambiar modelos fácilmente mediante variables de entorno, sin modificar código. Los costes finales dependerán del volumen de consultas y el modelo elegido.

---

# 3. Estimación de Costes e Inversión

## 3.1. Costes Únicos de Puesta en Marcha (Setup)

Esta sección detalla la inversión inicial necesaria para la ingesta masiva y el procesamiento de los **143.000 documentos** históricos para construir la base de datos inteligente.

**Cálculo de Inversión Inicial (Pago Único)**

- **Procesamiento IA Inicial:** Coste de API (LLM + Embeddings) para generar los resúmenes y vectores de todo el corpus.
  - **200 € - 500 €**
- **Infraestructura Cloud:** Capacidad de cómputo para descargar, *parsear* y procesar los 143.000 PDFs.
  - **50 € - 100 €**
- **Total Setup Tecnológico Estimado:**
  - **250 € - 600 €**

## 3.2. Costes Operativos Recurrentes (Mensuales)

Esta estimación cubre el mantenimiento continuo del sistema, la gestión de nuevas convocatorias y el uso estándar por parte de los usuarios.

**Cálculo de Costes Operativos (Mensuales)**

- **Base de Datos (PostgreSQL + pgvector):** Ya incluida en la infraestructura existente de Artisting.
  - **Sin coste adicional** (almacenamiento marginal: ~10-15 GB para subvenciones)
- **APIs de IA:** Uso variable para la generación de respuestas y la ingesta continua de nuevos documentos.
  - **~15 € - 25 € / mes** (dependiendo del volumen de consultas y modelo elegido)
- **Total Mensual Estimado:**
  - **15 € - 25 € / mes**

**Nota:** El coste mensual se reduce significativamente al aprovechar la infraestructura PostgreSQL existente de Artisting, eliminando la necesidad de un servicio adicional de base de datos.

---

# 4. Integración con Artisting y Arquitectura Técnica

## 4.1. Componentes a Reemplazar en el Backend Django

El sistema reemplazará los siguientes componentes del backend actual:

**Módulos a modificar:**

- `backend/chat/views.py`: Reemplazo de la lógica de chat legal (`chat_api`) por consultas de subvenciones

- **`backend/boe/retrieval.py`:** Nueva función `search_subvenciones()` que reemplaza `search_boe()` para el contexto de subvenciones
- **`backend/apps/agent/services/llm_client.py`:** Actualización para soportar Gemini/GPT-4o en lugar de DeepSeek

**Componentes que se mantienen sin cambios:**

- Sistema de autenticación JWT (`JWTAuthentication`)
- Billing y créditos Stripe (`billing/`, `users/models.py::UserProfile`)
- Caché semántico (`semantic_cache/`)
- Infraestructura Celery y Redis
- Base de datos PostgreSQL con pgvector

**Nuevas tablas PostgreSQL:**

```
-- Tabla principal de subvenciones
CREATE TABLE subvenciones_documento (
    id UUID PRIMARY KEY,
    numero_convocatoria VARCHAR(255),
    titulo TEXT,
    region VARCHAR(100),
    tipo_beneficiario VARCHAR(100),
    cuantia_min DECIMAL(12,2),
    cuantia_max DECIMAL(12,2),
    fecha_limite DATE,
    fecha_publicacion DATE,
    url_pdf TEXT,
    pdf_hash VARCHAR(64) UNIQUE,
    resumen_experto TEXT,
    metadata JSONB,
    created_at TIMESTAMP DEFAULT NOW()
);

-- Tabla de vectores para búsqueda semántica
CREATE TABLE subvenciones_vector (
    id SERIAL PRIMARY KEY,
    documento_id UUID REFERENCES subvenciones_documento(id) ON DELETE CASCADE,
    embedding vector(768),  -- dimensión depende del modelo de embeddings
    created_at TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_subvenciones_vector_embedding ON subvenciones_vector
USING ivfflat (embedding vector_cosine_ops);
```

## 4.2. Cambios en el Frontend (Next.js)

**Archivos a modificar:**

- `frontend/app/chat/page.tsx`**:** Rediseño completo con filtros y vista de listado
- `frontend/lib/services/chat.ts`**:** Nuevos endpoints para `/api/v1/subvenciones/chat`
- Componentes nuevos:
  - `SubvencioneFilters.tsx` (región, beneficiario, fechas, cuantías, otro a especificar al comienzo del projecto)
  - `SubvencionesListView.tsx` (tabla/grid de resultados)
  - `SubvencionesChat.tsx` (interfaz conversacional)

**Estilos:** Se mantendrá la paleta de colores y diseño de Artisting, adaptando únicamente la estructura de la interfaz.

## 4.3. Arquitectura Reutilizable para Chat Legal

Aunque el sistema de subvenciones y Chat Legal tienen diferentes estructuras de datos, algunos componentes serán diseñados de forma modular para facilitar la futura implementación de Chat Legal:

**Componentes potencialmente reutilizables:**

1. **Pipeline de ingesta genérico (Python):**
   - Descarga y procesamiento de documentos
   - Generación de resúmenes con LLM
   - Deduplicación por hash
   - *Diferencias:* Fuente de datos (InfoSubvenciones API vs BOE/jurisprudencia), estructura de metadata
2. **Módulo de vectorización:**
   - Generación de embeddings
   - Inserción en PostgreSQL/pgvector
   - *Reutilizable:* 80-90% del código
3. **Motor RAG (Retrieval):**
   - Búsqueda híbrida (SQL + vectorial)
   - Ranking y reranking
   - *Reutilizable:* 70-80% con ajustes en filtros SQL
4. **Sistema de actualización continua:**
   - Detección de nuevos documentos
   - Procesamiento incremental
   - Limpieza de documentos obsoletos
   - *Diferencias:* Criterios de obsolescencia (fecha límite vs vigencia legal)

**Componentes específicos no reutilizables:**

- Extracción de metadata (campos completamente distintos)
- Filtros de interfaz (subvenciones: región/cuantía vs legal: tipo de norma/jurisdicción)
- Lógica de citación (artículos de convocatorias vs artículos legales)

**Estrategia de desarrollo:** Se priorizará la entrega funcional del sistema de subvenciones, documentando las secciones de código que puedan servir como plantilla para Chat Legal, pero sin agregar complejidad innecesaria en esta fase.

---

# 5. Propuesta de Desarrollo en Dos Fases

## 5.1. Fase 1: Prototipo Funcional (MVP)

El enfoque en esta primera fase es la validación técnica y la prueba de concepto del modelo RAG (Retrieval-Augmented Generation) propuesto.

- **Objetivo Principal:** Validar la precisión y viabilidad de la arquitectura con una muestra controlada de datos.
- **Alcance Clave:**
    - **Ingesta Inicial:** Procesamiento, resumen y vectorización de **1.000 PDFs** para establecer la base de datos de prueba.
    - **Motor RAG:** Implementación del *core* funcional: Módulos de Ingesta, Vectorización y el Motor de Recuperación ("El Bibliotecario").
    - **Integración Backend:** Modificación de `backend/chat/views.py`, nuevo módulo `backend/subvenciones/`, integración con PostgreSQL/pgvector existente.
    - **Prototipo Frontend:** Interfaz básica de chat con filtros simples, sin diseño final pulido.
    - **Entrega:** Un **prototipo funcional integrado** en el entorno de desarrollo de Artisting, accesible mediante autenticación, para que el equipo pueda validar la calidad, veracidad de las respuestas y correcta citación de fuentes.
- **Métrica de Éxito:** El prototipo se considerará exitoso al demostrar una tasa de precisión superior al 95% en la recuperación y citación de fuentes para un conjunto predefinido de preguntas complejas.

**Inversión Fase 1: €1,500 + IVA**

**Términos de pago:**

- 30% al inicio (€450 + IVA)
- 70% a la entrega (€1,050 + IVA)

**Duración estimada:** 2-3 semanas

---

## 4.2. Fase 2: Producción y Escalado

Una vez validada la precisión en la Fase 1, esta etapa se centra en la integración del volumen total de datos y la entrega del sistema final optimizado.

- **Objetivo Principal:** Integrar el volumen total de datos, optimizar el rendimiento y entregar el sistema listo para el usuario final.
- **Duración Estimada:** 3 Semanas.
- **Alcance Clave:**
    - **Escalado de Datos:** Procesamiento masivo e ingesta de los **143.000 documentos** restantes, utilizando los costes de cómputo y API presupuestados en la Fase Setup.
    - **Interfaz de Usuario Completa:**
        - Rediseño frontend completo siguiendo los estilos de Artisting
        - Filtros avanzados interactivos (región, beneficiario, fechas, cuantías)
        - Vista de listado/grid de subvenciones
        - Historial de consultas
        - Presentación formal de respuestas con citaciones
    - **Optimización:**
        - Ajuste fino de parámetros del modelo RAG
        - Reducción de latencia de respuesta
        - Configuración de actualización automática diaria (Celery tasks)
        - Sistema de limpieza de convocatorias expiradas
    - **Testing y QA:** Pruebas de integración, pruebas de carga, validación de precisión
- **Entrega: Sistema listo para producción** integrado en Artisting, con documentación técnica completa y sistema de mantenimiento automatizado activo.

**Inversión Fase 2: €2,000 + IVA**

**Términos de pago:**

- 100% al finalizar el proyecto

**Duración estimada:** 2-3 semanas

---

# 6. Coste Total del Proyecto

A continuación, se presenta la inversión total requerida para el proyecto, integrando los honorarios de desarrollo y el coste tecnológico de la infraestructura.

## 6.1. Inversión en Desarrollo

- **Total Honorarios de Desarrollo (Fase 1 + Fase 2):**
    - **3.500 € + IVA**

## 6.2. Costes Tecnológicos de Setup

- **Coste Setup Tecnológico (Pago por Uso):**
    - Este coste será incurrido y facturado directamente a las cuentas del Cliente por los proveedores de Cloud y APIs (según el detalle de consumo del Punto 3.1), garantizando la titularidad de todos los recursos.
    - **Estimado: 200 € - 500 €**
    - Incluye: Procesamiento inicial de 143.000 PDFs (LLM + embeddings), capacidad de cómputo para ingesta masiva

## 6.3. Coste Total General

- **Inversión Inicial Total:**
    - **3.700 € - 4.000 € (aproximado)**
- **Coste Operativo Mensual Recurrente:**
    - **15 € - 25 €/mes** (significativamente reducido vs propuesta original gracias a reutilización de infraestructura PostgreSQL)

## 6.4. Ventajas Económicas de la Integración con Artisting

✓ **Ahorro de ~25 €/mes** en base de datos (reutilización de PostgreSQL existente) ✓ **Sin costes de infraestructura adicional** (Celery, Redis ya implementados) ✓ **Sin costes de desarrollo de autenticación/billing** (stack completo ya funcional) ✓ **Arquitectura base reutilizable** para el futuro proyecto Chat Legal

**Nota:** Este total general se refiere a la inversión inicial de desarrollo y setup. El coste operativo mensual iniciará una vez que el sistema entre en producción.

---

# 7. Ventajas del Enfoque Propuesto

## Validación Temprana

El prototipo con 1.000 PDFs permite validar la calidad del sistema antes de invertir en el procesamiento completo.

## Eficiencia de Costes

- Vectorización solo de resúmenes (no texto completo) **reduce costes >90%**
- **Reutilización de infraestructura existente** (PostgreSQL, Celery, Redis, Auth, Billing)
- Uso de modelos optimizados para cada tarea dentro de un presupuesto razonable
- **Ahorro mensual de ~25-30 €** vs implementación standalone

## Precisión Garantizada

- El sistema siempre cita fuentes específicas
- Respuestas basadas en el documento original (no en resúmenes)
- Combinación de filtros SQL + búsqueda semántica para máxima precisión

## Integración Nativa con Artisting

- **Sin fricción para usuarios existentes:** Misma autenticación, mismo sistema de créditos
- **Diseño coherente:** Mantiene el look & feel de la plataforma
- **Infraestructura probada:** Aprovecha stack Django/Next.js ya en producción
- **Desarrollo más rápido:** Menos código nuevo, más reutilización

## Flexibilidad Técnica

- Posibilidad de cambiar modelos de IA según necesidades (Gemini ↔ GPT-4o)
- Base de datos que permite consultas SQL tradicionales y búsquedas semánticas
- Arquitectura modular y mantenible
- **Código base reutilizable** para el futuro proyecto Chat Legal

## Automatización

- Sistema de actualización automática (Celery Beat/n8n)
- Limpieza de convocatorias expiradas
- Bajo mantenimiento operativo
- Integración con el sistema de monitoreo existente de Artisting
- Automatización de reportes a usuarios - (Inicial / Semanal)