

INFORME DE EVALUACIÓN DE PRÁCTICAS DE EMPRESA

Proyecto OPTICON



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CÁTEDRA ENIA-UPV EN IA DESARROLLO SOSTENIBLE

Albert Asunción Benedito

Febrero – mayo 2026

ÍNDICE DE CONTENIDOS

Presentación	p 3
1. Descripción concreta y detallada de las tareas, trabajos desarrollados y departamentos de la entidad a los que ha estado asignado	p 3
a) Recogida de datos	p 3
b) Elaboración de la base de datos	p 3
c) Establecimiento de los Key Performance Indicator (KPI)	p 6
d) Visor	p 7
2. Valoración de las tareas desarrolladas con los conocimientos y competencias adquiridas en relación con los estudios universitarios	p 9
3. Relación de los problemas planteados y de los procedimientos seguidos para su resolución	p 10
4. Identificación de las aportaciones que, en materia de aprendizaje han supuesto las prácticas	p 11
5. Evaluación de las prácticas y sugerencias de mejora	p 12
ANEXO. Detalle de los Key Performance Indicator (KPI) elaborados	p 13
Referencias	p 18

PRESENTACIÓN

El presente informe recoge las actividades realizadas durante el periodo de prácticas en el marco de la **Cátedra ENIA-UPV en Inteligencia Artificial para el Desarrollo Sostenible**, desarrolladas en el Departamento de Ingeniería Civil de la Universitat Politècnica de València.

Estas prácticas han sido coordinadas por las profesoras María Laura Montalbán Domingo (Ingeniería de la Construcción y Proyectos de Ingeniería Civil) y Stella María Heras Barberá (Sistemas Informáticos y Computación), dentro del proyecto **OPTICON**, financiado por la Generalitat Valenciana.

El objetivo principal del proyecto es **optimizar los procesos de contratación pública**, mediante la recopilación, tratamiento y análisis de datos, así como la definición de indicadores clave (KPI) orientados a evaluar los contratos desde una perspectiva de **eficiencia y sostenibilidad**.

1. DESCRIPCIÓN CONCRETA Y DETALLADA DE LAS TAREAS, TRABAJOS DESARROLLADOS Y DEPARTAMENTOS DE LA ENTIDAD A LOS QUE HA ESTADO ASIGNADO

El trabajo realizado se estructura en cuatro fases principales:

a) Recogida de datos

En esta fase inicial se recopilaban datos públicos de licitaciones procedentes de la **Plataforma de Contratación del Sector Público**, correspondientes al período 2021–2025, excluyendo contratos menores.

Los datos fueron descargados en formato abierto mediante la herramienta **OpenPLACSP**¹, lo que permitió integrar múltiples archivos en documentos Excel únicos (.xlsx).

Cabe destacar el **gran volumen de información tratado**, con aproximadamente:

- 200.000 registros anuales por tabla
- Más de 2 millones de registros agregados
- Archivos de entre 150 y 200 MB

Cada conjunto incluye información tanto de licitaciones como de sus resultados, con más de 30 variables por registro.

b) Elaboración de la base de datos.

¹ https://contrataciondelestado.es/wps/portal/plataforma/datos_abiertos

Esta fase constituye el núcleo principal del trabajo. Tras la recopilación, los datos se volcaron en una **base de datos relacional en PostgreSQL**, elegida por su robustez, escalabilidad y la familiaridad previa con el sistema.

Se definieron inicialmente dos tablas principales:

- licitaciones
- resultados

Problemas de calidad e inconsistencia de los datos

Desde esta primera etapa se detectó el principal desafío del proyecto: la **inconsistencia de los datos**.

Durante el volcado se identificó un número significativo de duplicados:

- Licitaciones: de **915.062 registros iniciales a 749.040 tras depuración**
- Resultados: de **1.168.012 a 980.925 registros**

Se realizó una primera eliminación de duplicados completos y posteriormente se aplicaron criterios específicos:

- Licitaciones: clave única id + fecha de actualización (manteniendo la más reciente)
- Resultados: id + lote + adjudicatario + fecha de actualización

Mientras que en licitaciones el proceso fue relativamente directo, en resultados resultó considerablemente más complejo debido a:

- Falta de homogeneidad en criterios
- Inconsistencias en identificadores (especialmente en UTE)
- Datos imprecisos o incompletos

Estos problemas se atribuyen principalmente a errores humanos y decisiones ad hoc en el registro de datos en la plataforma.

Definición estratégica y ampliación de datos

Una vez obtenida una base consistente, se abordó una fase crítica de decisión: evaluar qué información estaba disponible, cuál era explotable y qué datos adicionales podían obtenerse de forma viable.

Se decidió acotar el análisis a **contratos de tipo “Obras”** y ampliar la base incorporando información no disponible directamente, centrada en las **modificaciones de contrato**.

Proceso de web scraping

Para ello, se desarrollaron varios **scripts en Python** encargados de rastrear páginas web y extraer información relevante.

El proceso consistió en:

1. Acceder a los enlaces de cada licitación

2. Identificar secciones relevantes (Anuncios, Otros documentos, Anuncios anulados)
3. Buscar términos específicos como “modificación de contrato”
4. Extraer enlaces y datos asociados
5. Exportar resultados a archivos .csv

El scraping se realizó en dos fases:

Fase 1: Identificación de modificaciones

Cada licitación tiene un enlace donde se puede acceder mediante enlaces a su anuncio, formalización y otros más, entre los cuales están las modificaciones de contrato (ver figura 1).

- Detección de número de modificaciones
- Obtención de enlaces asociados
- Clasificación según tipo (vigente, anulado, rectificación)

Los resultados se almacenaron en la tabla modificacion_contrato y en una ampliación de la tabla de licitaciones (licitacion_mod).

Fase 2: Extracción de detalles

- Acceso a cada enlace de modificación
- Extracción de campos específicos previamente definidos
- Volcado de información completa en la base de datos

Rendimiento del scraping

Tras un proceso iterativo de mejora:

- Velocidad alcanzada: ~20.000 páginas/hora
- Tasa de error: ~0,4%

</

Figura 1 Vista general de la web de la licitación, con enlaces a las modificaciones de contrato.

Debido al volumen de datos, el proceso se ejecutó en **lotes de aproximadamente 20.000 páginas**, con una duración de una hora por lote, lo que permitió optimizar el control de errores y la eficiencia.

Una vez capturadas las modificaciones junto a sus enlaces, se volcaban los resultados a una tabla diseñada para este propósito, “modificacion_contrato”, junto a unos campos añadidos a la tabla licitacion, que se amplió a licitacion_mod. Con este conjunto de enlaces a las modificaciones, se hace una segunda fase de scrapping para capturar los datos de estas modificaciones de contrato, que han sido estudiados previamente, y exportarlos a .csv (ver Figura 2).

Con los datos de esta fase se completan los campos de la tabla “modificacion_contrato” en PostgreSQL.

The image shows a screenshot of a public procurement modification announcement. At the top, it says "Anuncio de modificación de contrato" with a reference number "Número de Expediente 2017.0052 - 008 008" and a publication date of "07-09-2021 a las 12:25 horas". The entity is "Entidad Adjudicadora: Rectorado de la Universitat de València". The object of the contract is "Producción y sustitución de climatizadores de la Facultad de Farmacia del Campus de Burjassot". The modification details include "Número de contrato 1", "Fecha de modificación 23/07/2021", "Observaciones Ampliación de plazo", "Número de modificación 6", "Modificación del plazo 55 Día(s)", and "Ejeto total tras la modificación 945 Día(s)". The initial conditions include "Clasificación CPT 4521000 - Trabajos de instalación de calefacción, ventilación y aire acondicionado", "Plazo de ejecución 15 Mes(es)", "Lugar de ejecución Subestación Nacional Valencia/Valencia", and "Dirección Postal España". The adjudicator is "FULTON, S.A." with NIF A46035770.

Figura 2 Vista de anuncio de modificación de contrato con las modificaciones y las condiciones iniciales, entre ellas el plazo de ejecución.

En este proceso de automatización de la búsqueda web se consiguió después de un proceso de aprendizaje y mejora acelerar la búsqueda de información hasta unas 20000 webs por hora con tasas de información no encontrada entorno al 0.4%, pero aun así el volumen de la búsqueda convertía este proceso en un trabajo largo para obtener todos los resultados, siendo importante así organizar bien las distintas fases del trabajo (*scrapping*, volcado, mejora de los *scripts*, documentación del proceso) para optimizar el tiempo empleado. El proceso de *scrapping* se ejecutó así normalmente dividido en lotes de unas 20000 webs, en un proceso entorno a la hora de duración que minimizaba la cantidad de errores.

c) Definición de los Key Performance Indicator (KPI)

El establecimiento de KPI constituye una fase clave del proyecto.

Se definieron 8 indicadores, basados en el marco de la OCDE (2023), siguiendo un criterio de equilibrio entre:

- Relevancia analítica
- Disponibilidad de datos
- Viabilidad de implementación

Durante esta fase se detectaron carencias en los datos iniciales (por ejemplo, el plazo original del contrato), lo que motivó la ampliación del scraping para obtener dicha información.

Construcción de la base para KPI

Se creó la tabla consolidada: `tabla_licitaciones_resultados_total` con un total de 138.133 registros, que relaciona cada resultado con su licitación.

A partir de esta tabla se generaron vistas SQL específicas para cada indicador.

Tipología de los indicadores

Los KPI desarrollados incluyen:

- Indicadores basados en:
 - Modificaciones de contrato
 - Costes (PBL vs adjudicación)
 - Variables temporales
- Indicadores de distribución:
 - Tipo de procedimiento
 - Tipo de tramitación

El detalle completo se incluye en el anexo.

d) Desarrollo del Visor:

En esta etapa final, se implementó una herramienta de visualización web de los datos de la tabla que combina los datos completos de licitaciones y resultados junto a las vistas derivadas que construyen los indicadores. SE realizó un desarrollo en Django, un *framework* (un conjunto de herramientas, bibliotecas y estructuras de *software* que proporciona una base sobre la que desarrollar aplicaciones) de Python que permite desarrollar aplicaciones web de forma más o menos sencilla y estructurada sobre el cual ya tenía algo de experiencia.

Para construir el visor, mediante Django conectamos con nuestra base de datos. El grueso del código se realiza a través de los scripts:

- `models.py`: definición de modelos a partir de tablas y vistas
- `views.py`: selección y lógica de datos

- HTML (resultados_generales.htm): visualización

Así, a grandes rasgos el flujo de trabajo para nuestro visor es:

Vista de KPI en PostgreSQL -> Copia de la vista en “models.py” -> Importar las variables del modelo en “views.py” -> Mostrar los datos de la estructura deseada en el HTML.

El visor incluye una *Home* sencilla como *landing page* con un enlace a la página de nuestros datos, que llamamos “Resultados Generales”, y que será el visor propiamente dicho.

El visor incluye:

- KPI **estáticos** (sobre el total de datos), basado en el conjunto de los resultados de “tabla_licitaciones_resultados_total”.
- KPI **dinámicos** (dependientes de filtros aplicados), donde los datos de los KPI varían en función de los datos seleccionados.

Los datos pueden filtrarse por:

- Estado
- Tipo de procedimiento
- Tramitación
- Adjudicatario

campos todos ellos de la tabla general.

Se ha decidido apostar por implementar el visor dinámico ya que resulta muy interesante para establecer comparativas y permite al usuario una interactividad y un acceso potente a la comprensión de los datos, posibilitando su mejor uso y comprensión, que es lo que al final se pretende en un organismo público.

A nivel visual, respecto a KPIs que son dato de porcentaje sobre el total (1, 2 y 4), se ha decidido hacer una barra azul sobre gris muestre el % sobre el total, con el título arriba, y verde en el caso de los resultados filtrados (Figura 3).

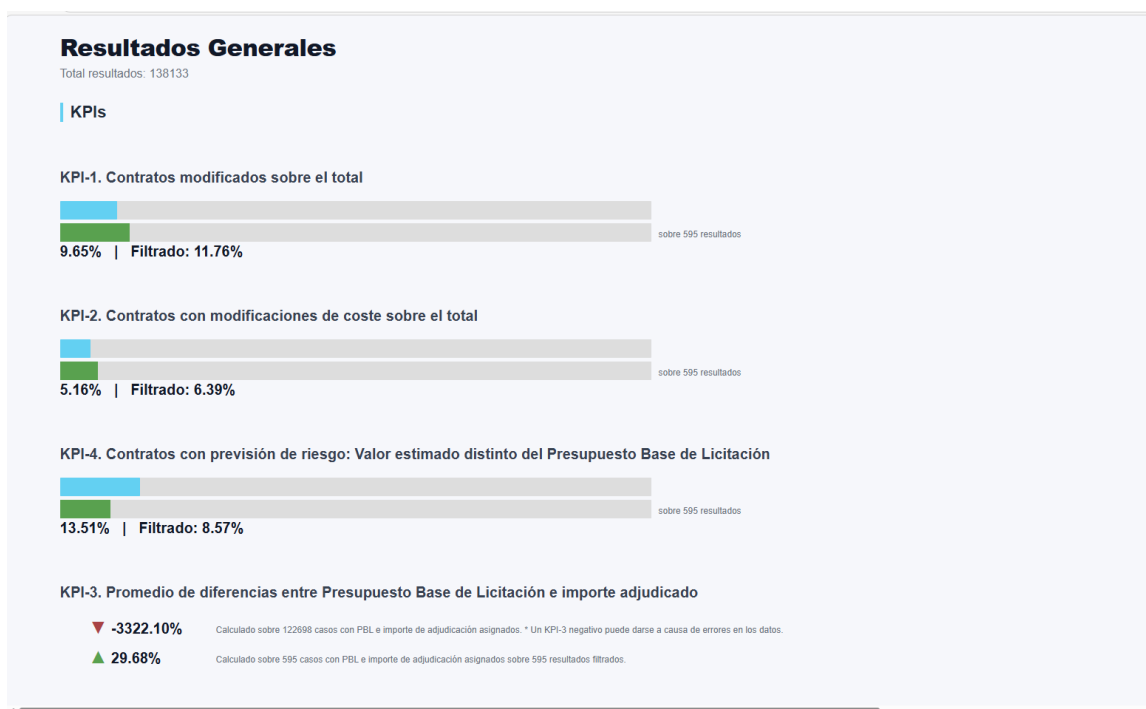


Figura 3 Vista de los KPI estáticos y dinámicos 1-4

Respecto al KPI3, al ser un porcentaje de variación se ha decidido mostrar el dato junto a una flecha de color variable según el valor, mientras que el KPI5 se mostrará con una barra vertical proporcional al valor en días del dato, ocupando 365 días el espacio asignado.

Para los KPI que muestran un gráfico de distribución sobre el total (6, 7, 8) solo se muestra un resultado, que se indicará en caso de ser un resultado filtrado (Figura 4).

2. VALORACIÓN DE LAS TAREAS DESARROLLADAS CON LOS CONOCIMIENTOS Y COMPETENCIAS ADQUIRIDAS EN RELACIÓN CON LOS ESTUDIOS UNIVERSITARIOS

Hay varios procesos durante las distintas etapas del trabajo que guardan estrecha relación con los estudios universitarios y en los cuales los conocimientos y competencias adquiridas me han resultado de mucha utilidad

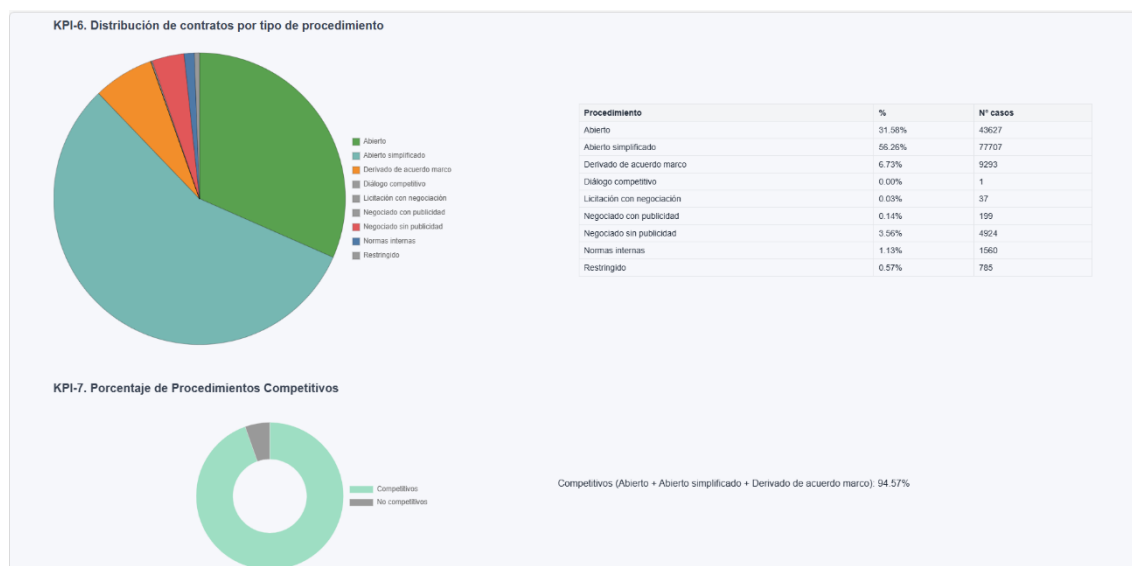


Figura 4 Vista de los resultados totales de los KPI 6 y 7 en gráfico de tarta

Tanto a la hora de pensar en los KPI como para las fases siguientes de la práctica, se han considerado técnicas de Inteligencia Artificial para elaborar modelos que predigan comportamientos, la estadística y las técnicas de optimización, así como materias específicas de la rama de computación como Aprendizaje Automático, Percepción y Agentes Inteligentes han resultado enriquecedoras.

También, toda la formación base en programación y lenguajes, con conceptos como las expresiones regulares, ha resultado fundamental para la elaboración de los distintos *scripts* en Python, tanto los *scrappers* como otros más sencillos de entrada y salida como un conversor de tiempo de plazo, de datos en lenguaje a número.

Finalmente, reseñar la vinculación especialmente en la etapa de desarrollo del visor con Django de conceptos desarrollados en materias como Concurrencia y Sistemas Distribuidos y Tecnología de Sistemas de información en la Red, como es el uso de javascript con *callbacks* o también el uso de hilos para paralelizar procesos en la etapa del *scraper*.

3. RELACIÓN DE LOS PROBLEMAS PLANTEADOS Y DE LOS PROCEDIMIENTOS SEGUIDOS PARA SU RESOLUCIÓN

Más allá de las dificultades técnicas encontradas, el principal problema encontrado ha sido la integridad y veracidad de los datos recogidos.

La plataforma de Contratación Pública se nutre de datos insertados por humanos, y como tal, es proclive a errores. Estos se deben no solamente a errores tipográficos claros (un error al poner una cifra menos en el Valor estimado del contrato por ejemplo puede romper toda lógica de un valor que debería ser mayor que el Presupuesto Base, además

de afectar a los cálculos de promedios, por ejemplo) sino también a diferencias de criterios sobre cuando insertarlo o dejarlo en blanco.

La falta de consistencia observada en los datos de la plataforma de contratación pública ha hecho que tengamos que tomar decisiones respecto a la limpieza de los datos asumiendo así un posible error.

Además, nuestra decisión de usar los datos de la última modificación en los contratos modificados nos ha dejado una importante limitación en los contratos con lotes, ya que hemos calculado modificaciones sin tener en cuenta a qué lote se aplicaba. Este es un punto de mejora inmediato en nuestra base de datos: en el futuro, será necesario *scrappear* las modificaciones (ya previamente guardadas) de los contratos con lotes e insertar un campo que diga a qué lote se aplica, y con esto, modificar los campos de la tabla “tabla_licitaciones_resultados_total” *modificacion_plazo* (junto a la conversión *modificacion_plazo_dias*) e *importe_acumulado_modificaciones*.

Los cálculos sobre las modificaciones de contrato son válidos hasta el momento solamente sobre contratos sin lote, por lo que por el momento se ha puesto el foco sobre los KPI en datos válidos sobre el total (KPI-1 y KPI-2). Los KPI se enriquecerían notablemente usando los datos de las modificaciones comparándolos con cantidades como el Presupuesto Base de Licitación y el Valor Esperado, y al tratarse solamente de 3000 casos de modificaciones de contratos con lotes sobre 13327, (22.1% del total de modificaciones), resulta un *scrappeo* relativamente rápido de hacer una vez comprobada la modificación lógica.

En cuanto a los problemas de índole más técnica, el proceso más complejo ha sido la elaboración y ejecución de los *scrappers*. Éste fue un proceso en el cual se dio una mejora constante, ya que mis conocimientos partían de una base bastante limitada. Este proceso supuso un aprendizaje y la modificación y elaboración de varias versiones de los *scrappers*, así como una atención especial a la hora de volcar, rellenar, completar y eliminar campos y resultados en la base de datos.

En un primer momento se desarrolló un *scraper* rápido (100 000 resultados en unos 15 minutos) que parecía dar buenos resultados, pero al comprobarlo descubrió múltiples fallos a partir de un número determinado de documentos procesados, en el que todos los documentos aparentemente carecían de modificaciones. Los servidores y páginas webs tienen una serie de estrategias de bloqueos y denegación de servicio precisamente para evitar *bots* y automatizaciones masivas, por lo que se producía esta pérdida de datos.

Las versiones siguientes diagnosticaban *Timeouts* y distintos tipos de error, y usando un sistema de hilos, se consiguió buenos resultados en tiempos razonables. Cabe destacar la importancia de encontrar un número equilibrado de *threads*, precisamente por estos bloqueos, y que consigan una velocidad aceptable (12 en el caso de esta herramienta).

Como se comentó en la descripción de las tareas, el proceso de *scrapping* se ejecutó en lotes de unas 20000 webs. Se hicieron algunos lotes más grandes dejando una noche computando, pero no valió especialmente la pena, no solamente por la cantidad de errores, sino por la posibilidad de manejarlo en cantidades más pequeñas parelizando el trabajo.

Este proceso de *scrapping* tuvo que lidiar también con la complejidad inherente a la burocracia administrativa, ya que estas modificaciones de contrato podían sufrir tanto rectificaciones como anulaciones, con sus anuncios correspondientes (y correspondientes duplicados), lo que se tuvo en cuenta para capturar el dato relevante. Esto requería el procesado de las distintas secciones de un *html*, y para eso resultó especialmente útil la librería *Beautiful Soup* para Python (<https://pypi.org/project/beautifulsoup4/>), diseñada para extraer datos de los documentos *html*.

Finalmente, pero no por ello más importante, tanto en la elaboración y limpieza como en la creación de las vistas de los KPI, el conocimiento de bases de datos y SQL resulta fundamental.

4. IDENTIFICACIÓN DE LAS APORTACIONES QUE, EN MATERIA DE APRENDIZAJE HAN SUPUESTO LAS PRÁCTICAS

En cuanto a las aportaciones en materias de aprendizaje, la mas destacable sería el *scrapper*, ya que ha supuesto un nuevo reto para mí, y me ha resultado fascinante, ya que de una operación en principio mecánica y aparentemente simple, he comprendido que en base a parámetros que tienen que ver con la seguridad, con la estructura de la web, con el uso de determinadas estructuras de datos o bibliotecas o la elección de procesos o hilos, pueden conseguirse resultados significativamente distintos en la automatización. La profundización en estas técnicas a la cual me ha llevado la necesidad de sacar buenos resultados en un tiempo aceptable es uno de los conocimientos más valiosos adquiridos durante las prácticas.

La creación y el mantenimiento de una base de datos de dimensiones reales para esto dominio de aplicación ha supuesto también una consolidación de mis conocimientos en Bases de datos y lenguaje SQL, con el que ya estaba familiarizado, pero con el que cada vez me siento más cómodo y con más herramientas para trabajar.

Algo similar ocurrió con el *framework* Django para elaborar el visor y del lenguaje HTML, que la elaboración del proyecto ha contribuido a consolidar.

5. EVALUACIÓN DE LAS PRÁCTICAS Y SUGERENCIAS DE MEJORA

La práctica me ha resultado excelente, tanto a nivel formativo como a nivel de interés personal y dinámicas de trabajo. El hecho de estar enmarcada en un proyecto de investigación mayor hace que la experiencia laboral suponga un estímulo intelectual en muchos aspectos y fuerza a buscar soluciones a los problemas que aparecen.

La dinámica de trabajo ha funcionado, ya que además del trabajo autónomo, durante el desarrollo de las prácticas se realizó una reunión semanal o bisemanal con las profesoras en las que se exponía el trabajo realizado y se identificaban y establecían los objetivos y el plan de trabajo para el próximo período. Este tipo de reuniones son muy enriquecedoras ya que aportan la parte fundamental del trabajo en equipo que es la visión e ideas de las otras personas del equipo del proyecto y también buscar soluciones y negociar el camino a seguir.

Una de las sugerencias, dado la estructura de este trabajo de investigación sería incorporar una sesión mensual de unas dos horas por ejemplo de trabajo conjunto *in situ*, planeada previamente tras identificar las necesidades del proyecto, para así avanzar más en cuanto a las decisiones que tomar frente a problemas que aparecen muchas veces de forma inesperada. Eso sí, las limitaciones logísticas y de disponibilidad hacen que este tipo de sesiones más allá de las reuniones sean difíciles de establecer.

ANEXO. Detalle de los Key Performance Indicator (KPI) elaborados.

Relación de los 8 KPI implementados en esta fase de desarrollo del proyecto. Estos KPIs se basan en el documento *Public procurement performance. A framework for measuring efficiency, compliance and strategic goal* de 2023 de la Organización para la Cooperación y el Desarrollo Económicos (OECD).

SE trabaja con la tabla de nuestra base de datos “tabla_licitaciones_resultados_total”, que relaciona cada resultado con su licitación, por lo cual cada lote adjudicado contará como un contrato distinto.

KPI-1. El primer KPI es el porcentaje de contratos modificados respecto del total.

Hay que tener en cuenta no contar las modificaciones de contratos anulados para calcularlo.

La consulta SQL es la siguiente:

```
CREATE OR REPLACE VIEW kpi1_contratos_modificados AS
SELECT
  ((SELECT count(*)
   FROM public.tabla_licitaciones_resultados_total
   WHERE num_mods_contrato_total_analizar>0 ) * 100.00)
  /
  (SELECT COUNT(*) from public.tabla_licitaciones_resultados_total )
AS porcentaje_contratos_modificados
```

Y el KPI-1 resultante es de **9.65%**.

KPI-2. El segundo KPI es el porcentaje de contratos con modificaciones de coste respecto al total, es decir, un subconjunto del que compone el KPI-1.

Con la view:

```
CREATE OR REPLACE VIEW kpi2_contratos_modificaciones_coste AS
SELECT
  ((SELECT count(*)
   FROM public.tabla_licitaciones_resultados_total
   WHERE num_mods_contrato_total_analizar>0 AND coste_modificacion is not
   null) * 100.00)
  /
  (SELECT COUNT(*) from public.tabla_licitaciones_resultados_total )
AS porcentaje_contratos_modificaciones_coste
```

El KPI-2 resultante es del **5.16%**.

KPI-3. El tercer KPI es la diferencia entre el presupuesto (sin impuestos) y el importe adjudicado.

Comprobamos que el 100% de los contratos tiene presupuesto, pero solamente 123267 del total de los 138133 casos (89.24%) tiene importe adjudicado al adjudicatario, el resto corresponde a licitaciones desiertas, desistimientos o renunciaciones. Sin embargo, hay 587 resultados donde el Presupuesto Base de Licitación es 0 (y no tiene por qué ser a causa de que el resultado de la adjudicación sea un desistimiento). El valor sobre el cual haremos la media será el de los casos que tengan **adjudicatario con PBL mayor que 0**, un total de **122698**.

Este KPI calcula la media del porcentaje respecto al Presupuesto Base de Licitación (PBL) que supone la diferencia entre el PBL y el importe adjudicado, es decir, la media del $(PBL - \text{Importe adjudicado}) * 100 / PBL$. Así, un importe de adjudicación más bajo que el PBL dará una diferencia positiva, mientras que un importe adjudicado mayor, negativa.

La fórmula general es el sumatorio de las diferencias entre el total de casos, así :

$$\sum (PBL - \text{Importe adjudicado}) * 100 / PBL / 123267$$

La view correspondiente es:

```
CREATE OR REPLACE VIEW
public.kpi3_media_diferencias_pbl_imp_adjudicado AS
SELECT
    AVG(
        (presupuesto_sin_impuestos - importe_adj_sin_imp_lic_lote) * 100.0
        / presupuesto_sin_impuestos
    ) FILTER (
        WHERE presupuesto_sin_impuestos > 0
        AND importe_adj_sin_imp_lic_lote IS NOT NULL
    ) AS media_diferencias_pbl_adjudicacion,

    COUNT(*) FILTER (
        WHERE presupuesto_sin_impuestos > 0
        AND importe_adj_sin_imp_lic_lote IS NOT NULL
    ) AS total_casos
FROM tabla_licitaciones_resultados_total;
```

El resultado de este KPI viene influido por datos donde el Presupuesto es muy inferior comparado con el importe adjudicado, que deberían filtrarse. Esos datos arrojan un porcentaje muy bajo, que, comparado con el 100% máximo en positivo, explica el valor tan bajo de la media (-3322.10%). Sin embargo, ya en el visor con resultados filtrados puede comprobarse la potencia de este indicador.

KPI-4. El cuarto va ligado también a estos costes y es el Porcentaje de licitaciones con una Aproximación Formal de Manejo de Riesgo, entendido por las licitaciones en las cuales hay una diferencia entre el Valor Estimado de Contrato (VEC) y el Presupuesto Base de Licitación (PBL).

Esta diferencia se calcula $VEC - PBL$, y debería ser siempre positiva, ya que teóricamente el Valor Estimado aumenta respecto al Presupuesto Base Licitado en base a previsiones de modificaciones, pero tenemos 1244 casos donde esto no ocurre. Hemos estudiado si la diferencia en los casos negativos corresponde a que se asuma que esa cifra de Valor Estimado sea el valor a sumar al PBL. Puede ser posible en los casos de Valor Estimado = 0, pero hay bastante variedad de casos.

Sí se ha observado que muchos de estos casos son errores de tipografía, como olvidarse una cifra que deberían corregirse al limpiar completamente la base de datos.

Sin embargo, al tratarse de un indicador de tipo Verdadero/Falso sobre un total, esta inconsistencia de los datos no tiene un impacto tan visible como en el KPI-3.

La consulta SQL es la siguiente:

```
CREATE OR REPLACE VIEW public.kpi4_pct_lic_riesgo_previsto
AS SELECT ((COUNT(*) filter (WHERE prevision_modificacion !='No')*
100.00) /
count(*)) AS pct_riesgo_previsto,

count(*) AS total_casos
FROM tabla_licitaciones_resultados_total
```

El KPI-4 resultante es del **13.51%**.

De este KPI-4 surgiría la propuesta de un 9º y 10º, que relacionaría estos datos de diferencia VEC-PBL con las modificaciones que tenemos recogidas, es decir, si esta previsión de modificaciones se ha ejecutado y en qué cantidad, es decir, el % de modificaciones efectivamente ejecutadas sobre las diferencias y su variación en %, cantidad que tenemos recogida en nuestro campo `coste_mod_pc_pbl` hasta el momento únicamente sobre los contratos sin lotes.

KPI-5. El quinto indicador es la media de la diferencia de tiempo entre la primera publicación y el acuerdo de licitación (“Average time for decisions” en el documento de la OECD).

Para calcularlo necesitamos en primer lugar que el campo “`primera_publicacion`” no sea NULL, que solo se da en 2 casos de contratos anulados.

La duda era si elegir como fecha de finalización la fecha de acuerdo o la de formalización. En principio parece resultar mejor opción la segunda, pero si comparamos los campos NULL de ambas, 46 frente a 29665, y vemos que la diferencia es de unos días, la primera nos daría una cantidad significativa de casos más a tener en cuenta. Sin embargo, entendemos que hay que estudiar los procesos completos, y finalmente elegimos solamente estudiar los casos donde sí haya fecha de formalización.

Así, la vista sería la siguiente:

```
CREATE OR REPLACE VIEW public.kpi5_tiempo_medio_formalizacion
AS
```

```

SELECT avg(fecha_form_contrato_lic_lote - primera_publicacion) AS
dias_medios_formalizacion,
count(*) AS total_casos
FROM tabla_licitaciones_resultados_total
WHERE primera_publicacion IS NOT NULL AND
fecha_form_contrato_lic_lote IS NOT NULL;

```

El KPI-5 resultante es de **91.49 días**.

KPI-6. El sexto es estudiar los resultados agrupados por tipo de procedimiento.

Observamos 9 tipos:

"Abierto"	31.58%	43627 casos
"Abierto simplificado"	56.26%	77707 casos
"Derivado de acuerdo marco"	6.73%	9293 casos
"Diálogo competitivo"	0.0007%	1 caso
"Licitación con negociación"	0.023%	37 casos
"Negociado con publicidad"	0.15%	199 casos
"Negociado sin publicidad"	3.57%	4924 casos
"Normas internas"	1.13%	1560 casos
"Restringido"	0.57%	785 casos

La consulta SQL sería:

```

CREATE OR REPLACE VIEW public.kpi6_pct_tipo_procedimiento AS
SELECT tipo_procedimiento,
count(*)::numeric * 100.0 / sum(count(*)) OVER () AS
porcentaje_tipo_procedimiento,
count(*) AS total_casos
FROM tabla_licitaciones_resultados_total
WHERE tipo_procedimiento IS NOT NULL GROUP BY
tipo_procedimiento;

```

Este KPI resulta sencillo de calcular, ya que el dato viene directamente de OpenPLACSP, y enlaza con el 7º KPI.

KPI-7. El séptimo KPI es una agrupación de los casos “Abierto”, “Abierto simplificado” y “Derivado de acuerdo marco” del KPI-6 sobre el total.

La consulta SQL corresponde a:

```

CREATE OR REPLACE VIEW kpi7_pct_procedimientos_competitivos AS
SELECT ( COUNT(*) FILTER ( WHERE tipo_procedimiento IN (
'Abierto', 'Abierto simplificado', 'Derivado de acuerdo marco')
) * 100.0 / COUNT(*)) AS pct_procedimientos_competitivos,
COUNT(*) AS total_casos
FROM tabla_licitaciones_resultados_total
WHERE tipo_procedimiento IS NOT NULL;

```

El KPI-7 resultante es del **94.57%**.

KPI-8. Finalmente, el octavo KPI agrupa los casos por el tipo de tramitación.

Hay 3 casos, “Emergencia”, “Ordinaria” y “Urgente”, siendo la ordinaria ampliamente mayoritaria (**93.86%**).

Así, de manera análoga al KPI6 del tipo de procedimiento, tendríamos la consulta SQL:

```
CREATE OR REPLACE VIEW public.kpi8_pct_tipo_tramitacion AS
SELECT tramitacion,
       count(*)::numeric * 100.0 / sum(count(*)) OVER () AS porcentaje,
       count(*) AS total_casos
FROM tabla_licitaciones_resultados_total
WHERE tramitacion IS NOT NULL GROUP BY tramitacion;
```

REFERENCIAS

1. ORGANISATION FOR ECONOMIC CO-OPERATION AND DEVELOPMENT (OECD). *Public procurement performance: A framework for measuring efficiency, compliance and strategic goals*. París: OECD Publishing, 2023. DOI: 10.1787/c7ce72c8-en.