



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERIA

Escuela Profesional de Ingeniería de Sistemas

Proyecto “EasyBeca”

Curso: Inteligencia de negocios

Docente: *Ing. Patrick Jose Cuadros Quiroga*

Integrantes:

Calizaya Ladera, Andy Michael	(2022074258)
Castillo Mamani, Diego Fernando	(2022073895)
Colque Ponce, Sergio Alberto	(2022073503)
Vargas Gutierrez, Angel Jose	(2022073504)

**Tacna – Perú
2025**



CONTROL DE VERSIONES

Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
3.0	DNFV	CNCM	CNCM	21/06/2025	Versión Original

Proyecto EASYBECA

Documento de Especificación de Requerimientos de Software

Versión 3.0



CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
3.0	DNFFV	CNCM	CNCM	21/06/2025	Versión Original

ÍNDICE GENERAL

1. INTRODUCCIÓN	4
1.1. Propósito (Diagrama 4+1)	4
1.2. Alcance	4
1.3. Definición, siglas y abreviaturas	5
2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS	6
2.1.1. Requerimientos Funcionales	6
2.1.2. Requerimientos No Funcionales – Atributos de Calidad	7
2.1. Restricciones	9
3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA	10
3.1. Vista de Caso de uso	10
1.1.1. Diagramas de Casos de uso	10
3.2. Vista Lógica	10
3.2.1. Diagrama de Subsistemas	10
3.2.2. Diagrama de Secuencia (vista de diseño)	11
3.2.3. Diagrama de Colaboración (vista de diseño)	14
3.2.4. Diagrama de Objetos	14
3.2.5. Diagrama de Clases	15
3.2.6. Diagrama de Base de datos (relacional o no relacional)	16
3.3. Vista de Implementación (vista de desarrollo)	16
3.3.1. Diagrama de arquitectura software (paquetes)	16
3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)	17
3.4. Vista de procesos	17
3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)	17
3.5. Vista de Despliegue (vista física)	18
3.5.1. Diagrama de despliegue	18
4. ATRIBUTOS DE CALIDAD DEL SOFTWARE	19



1. INTRODUCCIÓN

1.1. Propósito (Diagrama 4+1)

El propósito de este documento es definir y comunicar la arquitectura de software para el sistema "EasyBeca" utilizando el modelo 4+1. Este modelo permite abordar la solución de Inteligencia de Negocios (BI) desde múltiples perspectivas, asegurando que tanto los desarrolladores (encargados de los scripts ETL) como los usuarios finales (estudiantes, familias e instituciones) tengan una visión clara del sistema.

La arquitectura global se presenta como una solución de Inteligencia de Negocios (BI) basada en un flujo de datos (Pipeline de Datos), diseñada para separar la extracción y procesamiento de datos (Backend/ETL), el almacenamiento estructurado y la visualización interactiva (Frontend/Dashboard). Este enfoque tiene como objetivo principal facilitar la centralización de información dispersa, permitir la actualización periódica de datos y asegurar la portabilidad del análisis sin depender de infraestructura de servidores complejos.

Las decisiones de diseño han sido influenciadas por los requisitos funcionales y no funcionales. Por ejemplo, la necesidad de integrar fuentes heterogéneas (PDFs, portales web) justificó el uso de Python para la extracción y limpieza (ETL). La restricción de disponibilidad local y portabilidad llevó a elegir Excel como repositorio central y Power BI para la visualización, permitiendo que la solución se ejecute en equipos estándar con Windows sin dependencias externas complejas.

1.2. Alcance

El proyecto abarca el desarrollo de una solución integral que incluye los siguientes módulos y funcionalidades:

- Scripts en Python encargados de recolectar datos de fuentes públicas (PRONABEC, webs), limpiarlos y normalizarlos.



- Almacenamiento estructurado en archivos Excel que sirve como fuente única de verdad para el sistema.
- Conjunto de tableros interactivos en Power BI que permiten el análisis de becas por carrera, institución, tipo de beca, geografía y perfil socioeconómico.
- Funcionalidades para filtrar información histórica (2020-2025) y segmentar datos dinámicamente.

1.3. Definición, siglas y abreviaturas

Sigla/Abreviatura	Definición
Power BI	Herramienta de visualización de datos de Microsoft utilizada para la interfaz de usuario final.
Python	Lenguaje de programación utilizado para los scripts de automatización y limpieza de datos.
ETL	Extract, Transform, Load (Extraer, Transformar, Cargar). Proceso de mover datos desde diversas fuentes a un repositorio único.
Dashboard	Tablero de control visual que presenta las métricas y KPIs de forma gráfica e interactiva.

2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS

2.1.1. Requerimientos Funcionales



Id	Requerimiento	Descripción	Prioridad	Urgencia	Estado	Estabilidad
RF-001	Integración de fuentes de datos	El sistema debe recolectar datos de becas desde fuentes públicas (PRONABEC, PDFs, portales web, APIs).	Alta	Alta	Pendiente	Alta
RF-002	Extracción automática	El sistema debe extraer información relevante de PDFs y web usando Python.	Alta	Alta	Pendiente	Alta
RF-003	Limpieza y normalización	El sistema debe depurar datos: eliminar duplicados, corregir formatos, estandarizar nombres.	Alta	Alta	Pendiente	Alta
RF-004	Consolidación en Excel	El sistema debe almacenar la base final estructurada en archivos Excel.	Alta	Alta	Pendiente	Alta
RF-005	Cálculo de KPIs	Debe calcular KPIs: total becas por año, top carreras, top instituciones, distribución por género y estrato.	Alta	Alta	Pendiente	Alta
RF-006	Dashboard Becas por carrera	Debe mostrar ranking de carreras con más becas y filtros por año/tipo.	Alta	Media	Pendiente	Alta



RF-007	Dashboard Becas por institución	Debe mostrar distribución y ranking de instituciones con mayor cobertura.	Alta	Media	Pendiente	Alta
RF-008	Dashboard Becarios por beca	Debe comparar beneficiarios por programa (Beca 18, Permanencia, etc.).	Alta	Media	Pendiente	Alta
RF-009	Dashboard Mapa de becas	Debe visualizar distribución geográfica nacional e internacional.	Alta	Media	Pendiente	Alta
RF-010	Dashboard Perfil del becario	Debe mostrar distribución por género y estrato socioeconómico .	Alta	Media	Pendiente	Alta
RF-011	Filtros interactivos globales	Debe permitir filtros por año, carrera, institución, tipo de beca y ubicación.	Alta	Alta	Pendiente	Alta
RF-012	Actualización de datos	Debe permitir actualizar Excel consolidado y refrescar Power BI sin rediseñar.	Media	Media	Pendiente	Media
RF-013	Manejo de datos faltantes	Debe clasificar nulos como "No especificado" sin afectar KPIs.	Media	Media	Pendiente	Alta
RF-014	Consulta/Exportación visual	Debe permitir al usuario consultar gráficos y usarlos en informes.	Media	Baja	Pendiente	Media



2.1.2. Requerimientos No Funcionales – Atributos de Calidad

Id	Requerimiento	Descripción	Prioridad	Urgencia	Estado	Estabilidad
RNF-001	Usabilidad	El dashboard debe ser intuitivo, con etiquetas claras y fácil de interpretar para usuarios no técnicos.	Alta	Alta	Pendiente	Alta
RNF-002	Rendimiento	Las visualizaciones y filtros deben responder en ≤ 5 segundos en una laptop/PC estándar.	Alta	Media	Pendiente	Alta
RNF-003	Confiabilidad	Los KPIs y gráficos deben reflejar correctamente la base consolidada sin inconsistencias.	Alta	Alta	Pendiente	Alta



RNF-004	Integridad de datos	La estructura del Excel (columnas, tipos y formatos) debe mantenerse estable para no romper Power BI.	Alta	Alta	Pendiente	Alta
RNF-005	Seguridad y privacidad	El sistema no debe mostrar datos personales individuales; todo debe ser agregado y anonimizado.	Alta	Alta	Pendiente	Alta

2.1. Restricciones

El desarrollo y la funcionalidad de EasyBeca están sujetos a restricciones técnicas y operativas. El proyecto debe implementarse utilizando una pila tecnológica específica que consta de Python para el procesamiento de datos, Excel como base de datos ligera y Power BI para la visualización.

Críticamente, la funcionalidad depende de la disponibilidad y formato de las fuentes de datos públicas (PRONABEC, portales web); cambios drásticos en la estructura de los PDFs de origen pueden requerir mantenimiento en los scripts de extracción. El sistema está restringido a entornos Windows debido al uso de Power BI Desktop y Office. Además, el proyecto se desarrolla bajo un ciclo académico con fecha límite en 2025, lo que prioriza la funcionalidad esencial de visualización sobre características avanzadas como predicción compleja o bases de datos SQL dedicadas.

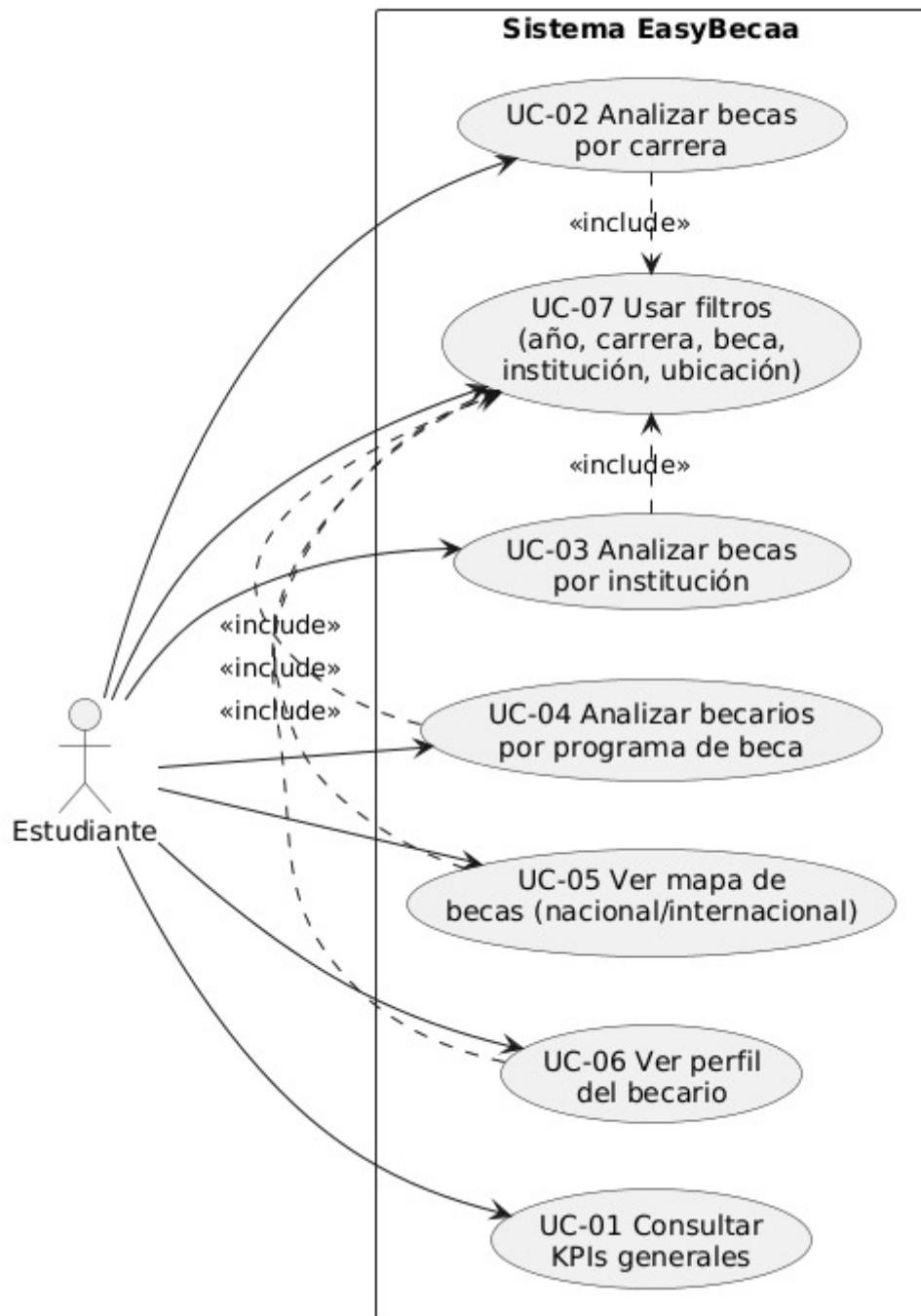


3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

3.1. Vista de Caso de uso

1.1.1. Diagramas de Casos de uso

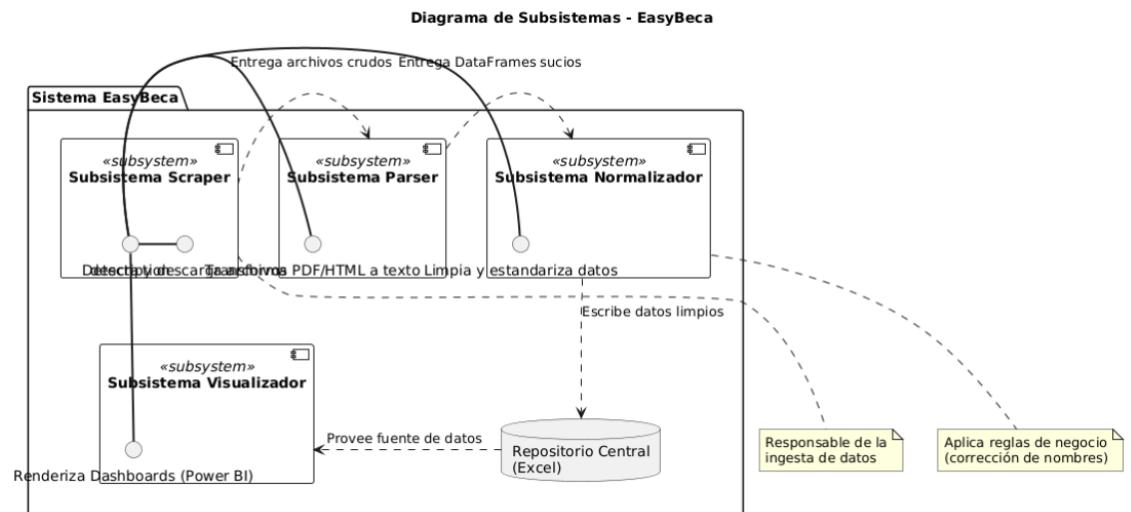
EasyBecaa - Diagrama de Casos de Uso (Solo Usuario)





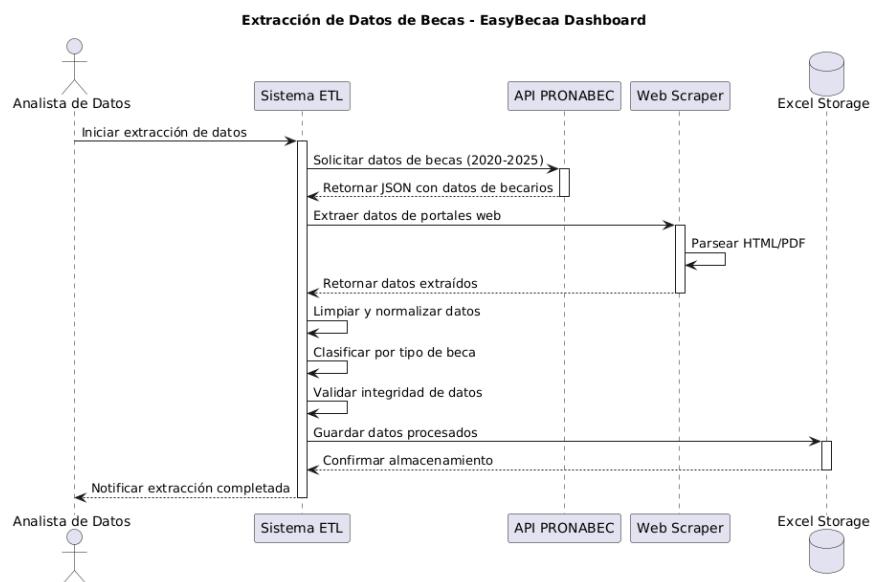
3.2. Vista Lógica

3.2.1. Diagrama de Subsistemas



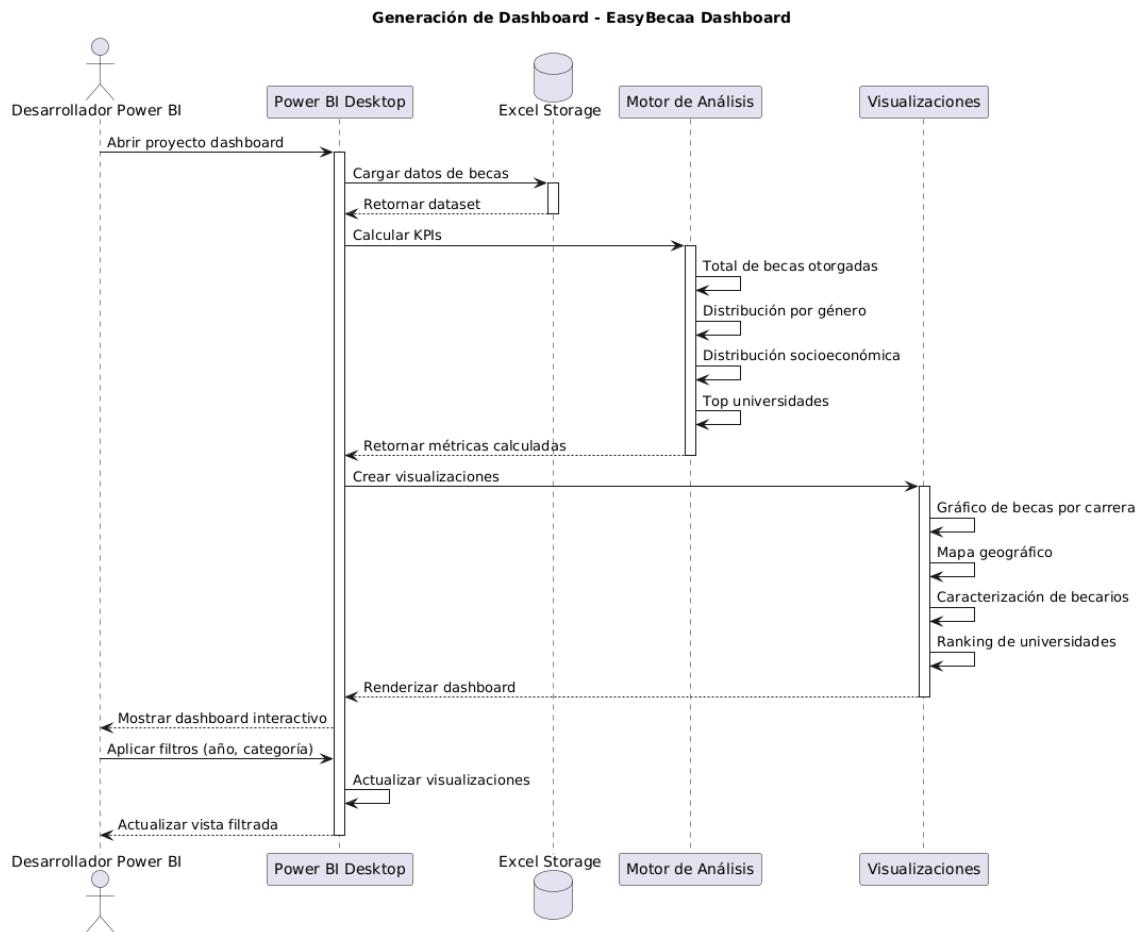
3.2.2. Diagrama de Secuencia (vista de diseño)

SD-UC-002 Extracción de Datos de Becas

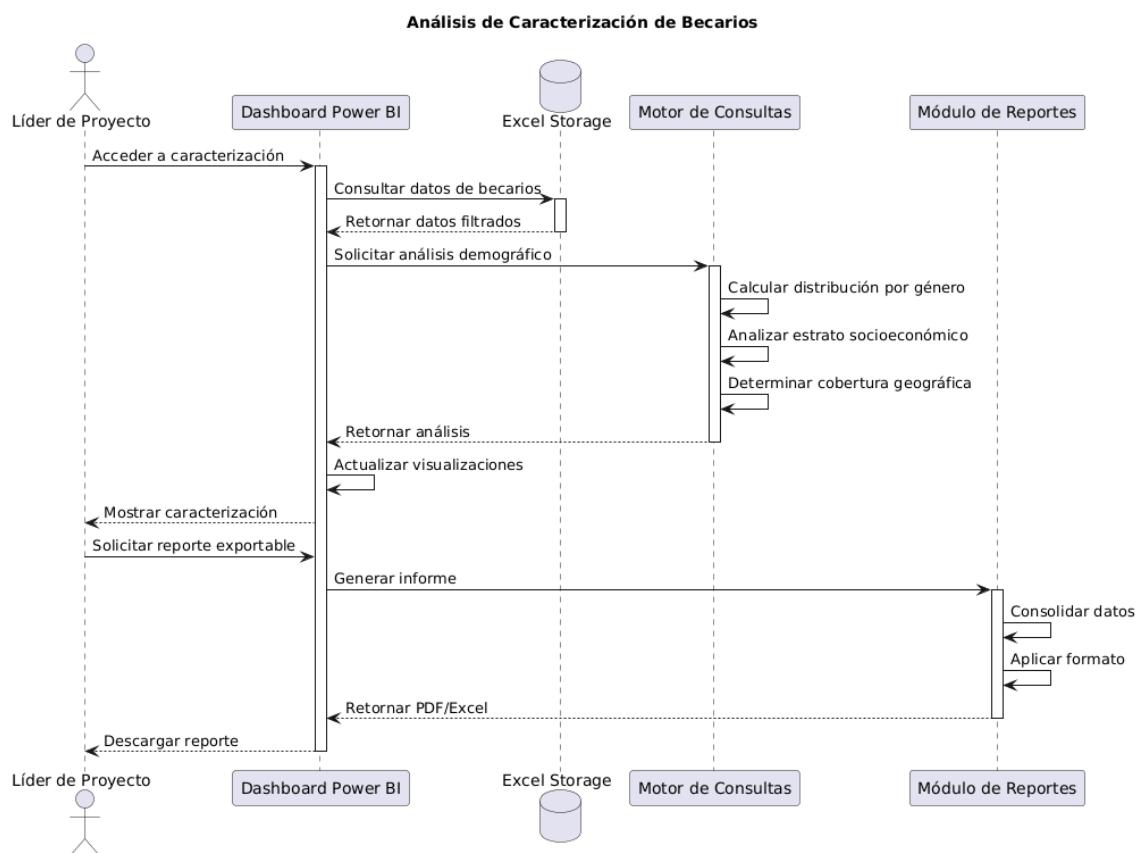




SD-UC-003 Generación de Dashboard en Power BI



SD-UC-004 Análisis de Datos de Becarios

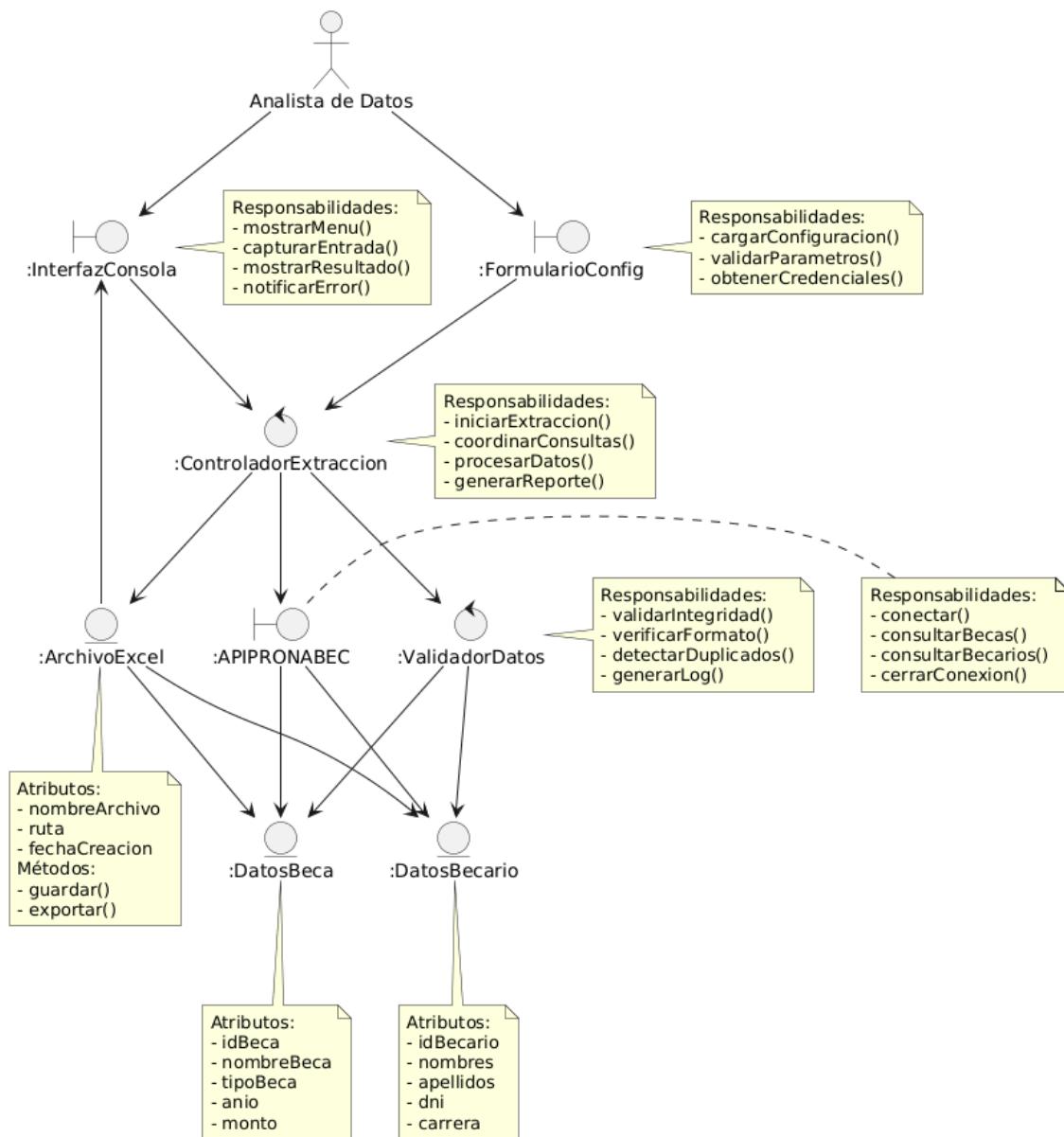


3.2.3. Diagrama de Objetos

UC-002: Extraer Datos de Becas desde PRONABEC



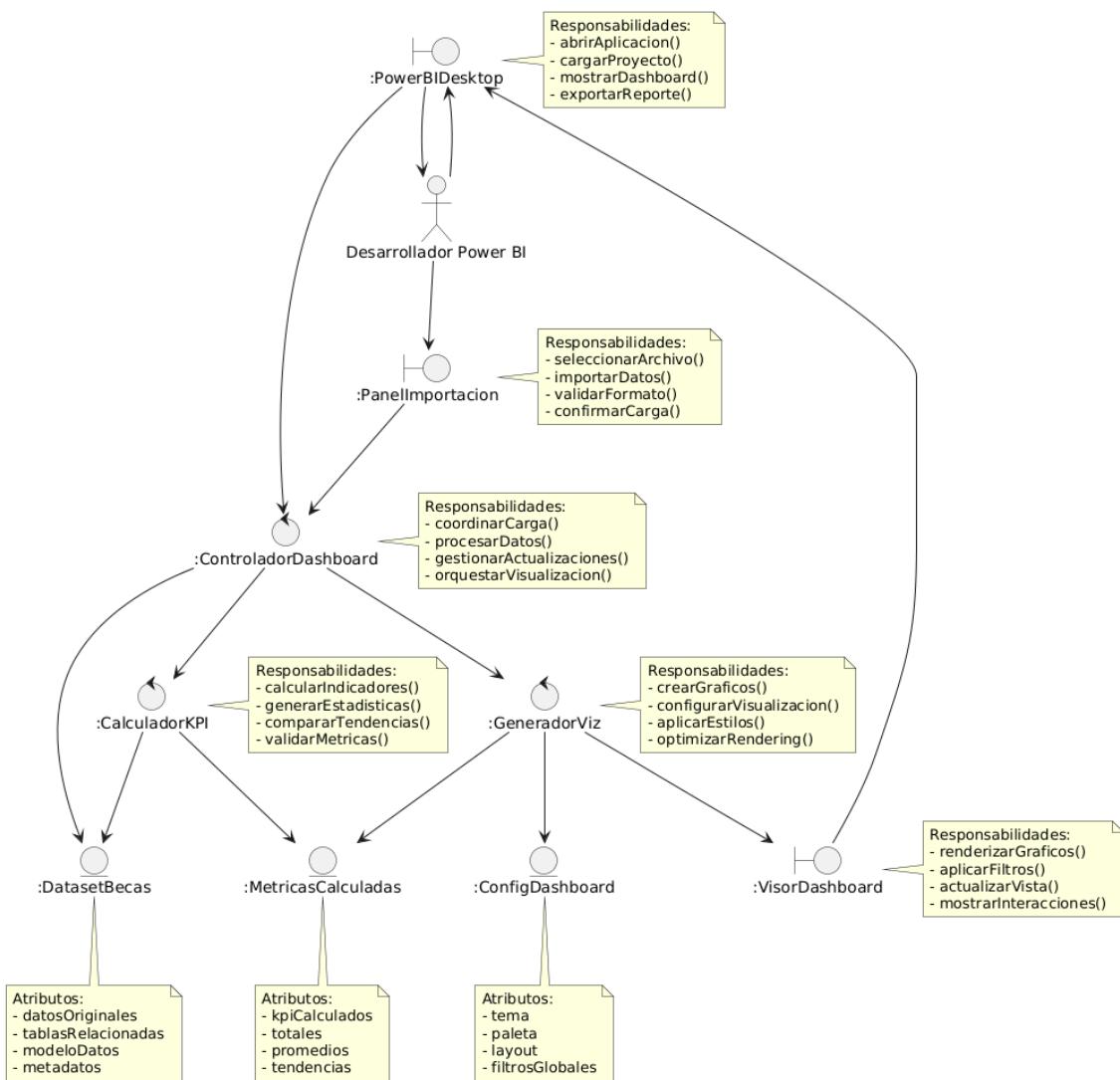
Análisis de Objetos: Extraer Datos de Becas desde PRONABEC



UC-002 Generar Dashboard Power BI



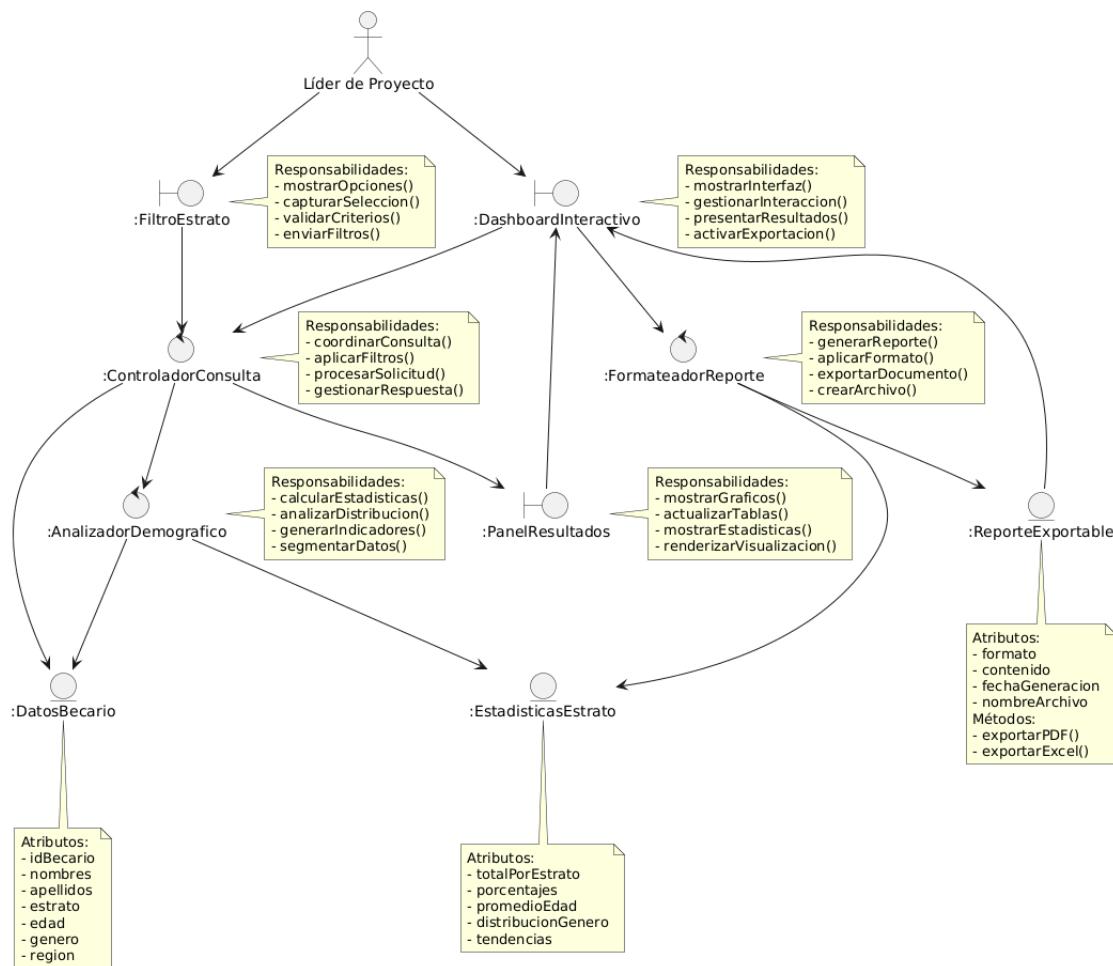
Análisis de Objetos: Generar Dashboard de Visualización en Power BI



UC-003 Consultar Caracterización



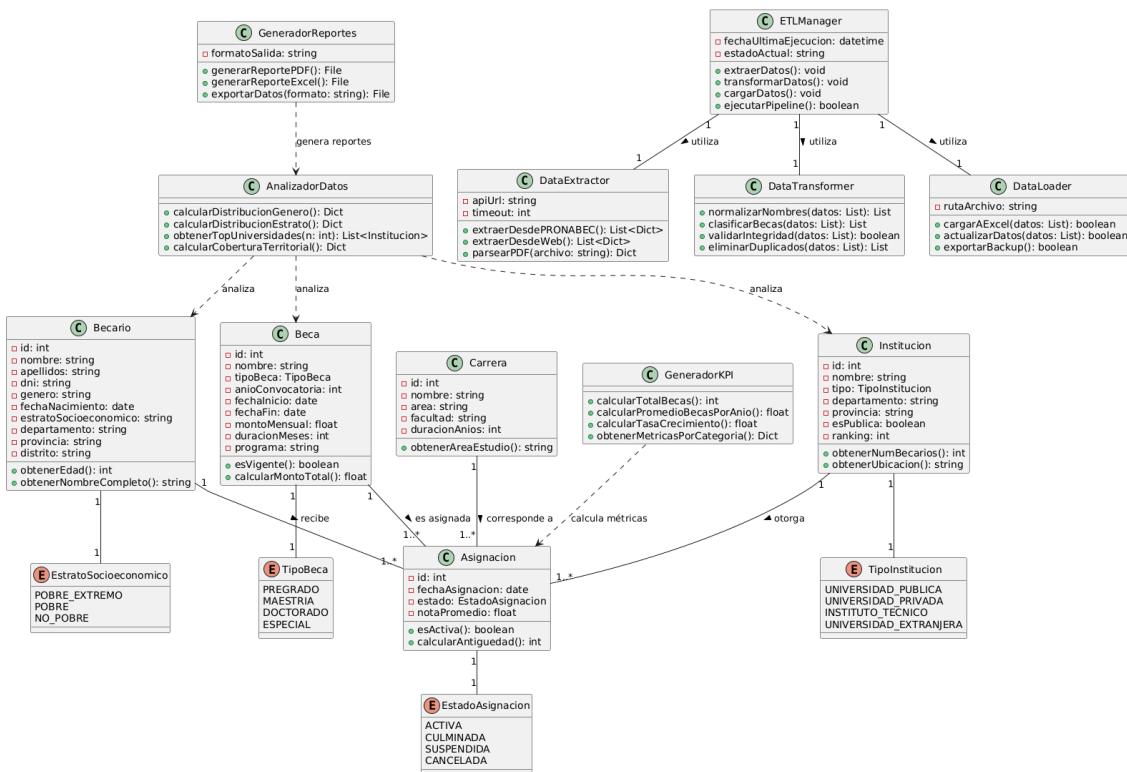
Análisis de Objetos: Consultar Caracterización de Becarios por Estrato



3.2.4. Diagrama de Clases



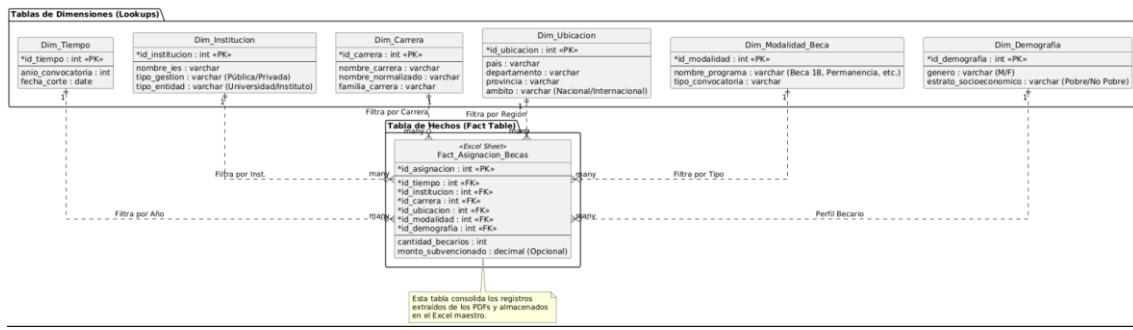
Diagrama de Clases - EasyBecaa Dashboard



3.2.5. Diagrama de Base de datos (relacional o no relacional)

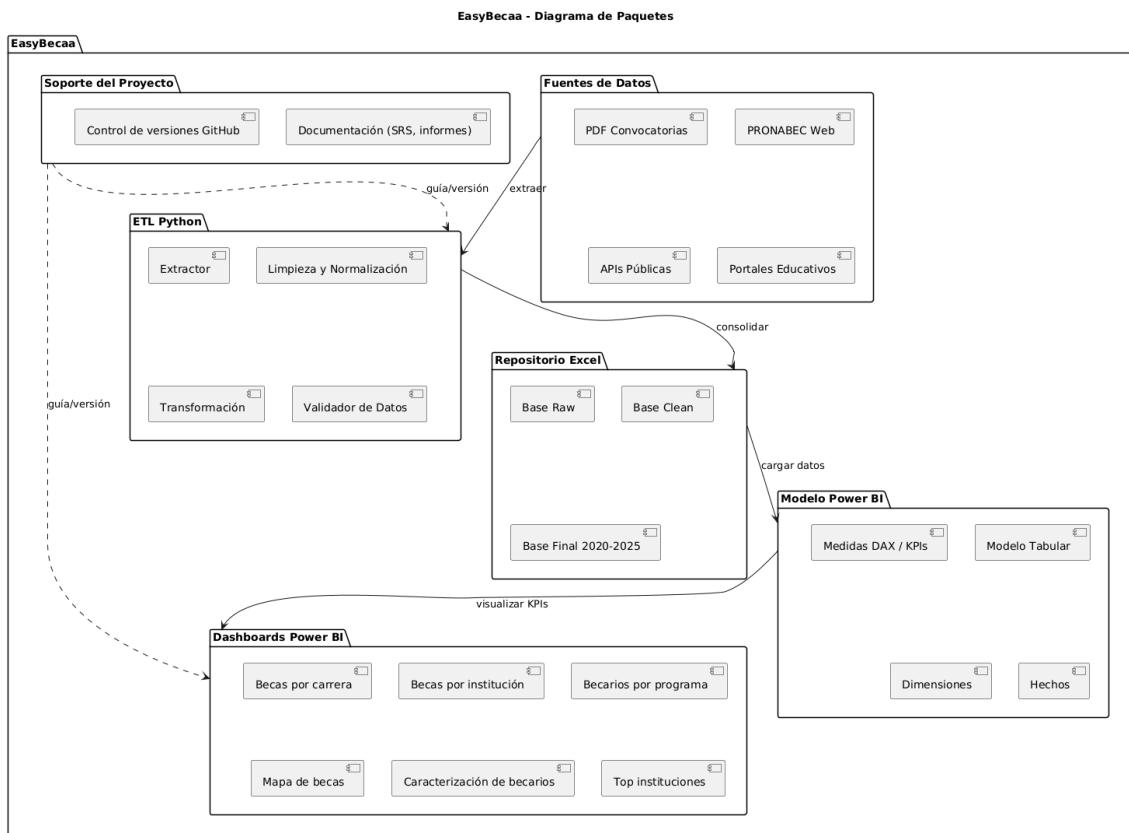


Diagrama de Modelo de Datos (Esquema Estrella) - EasyBeca

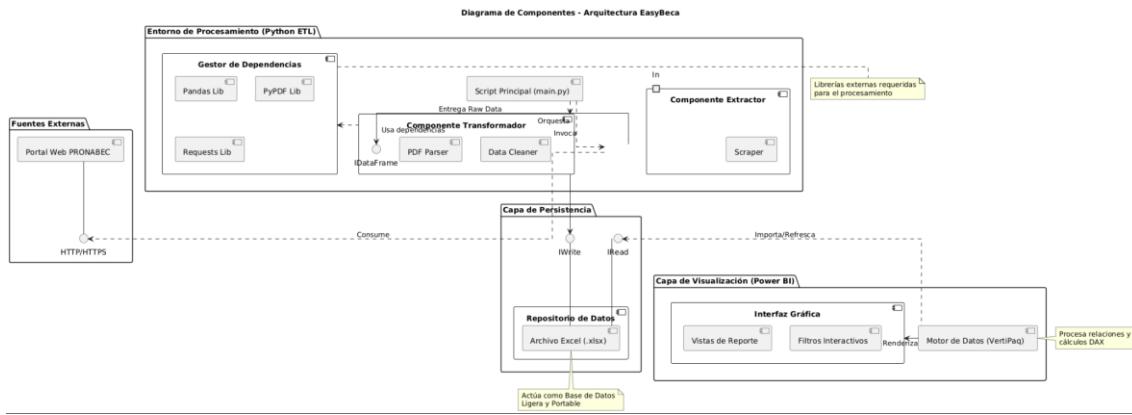


3.3. Vista de Implementación (vista de desarrollo)

3.3.1. Diagrama de arquitectura software (paquetes)



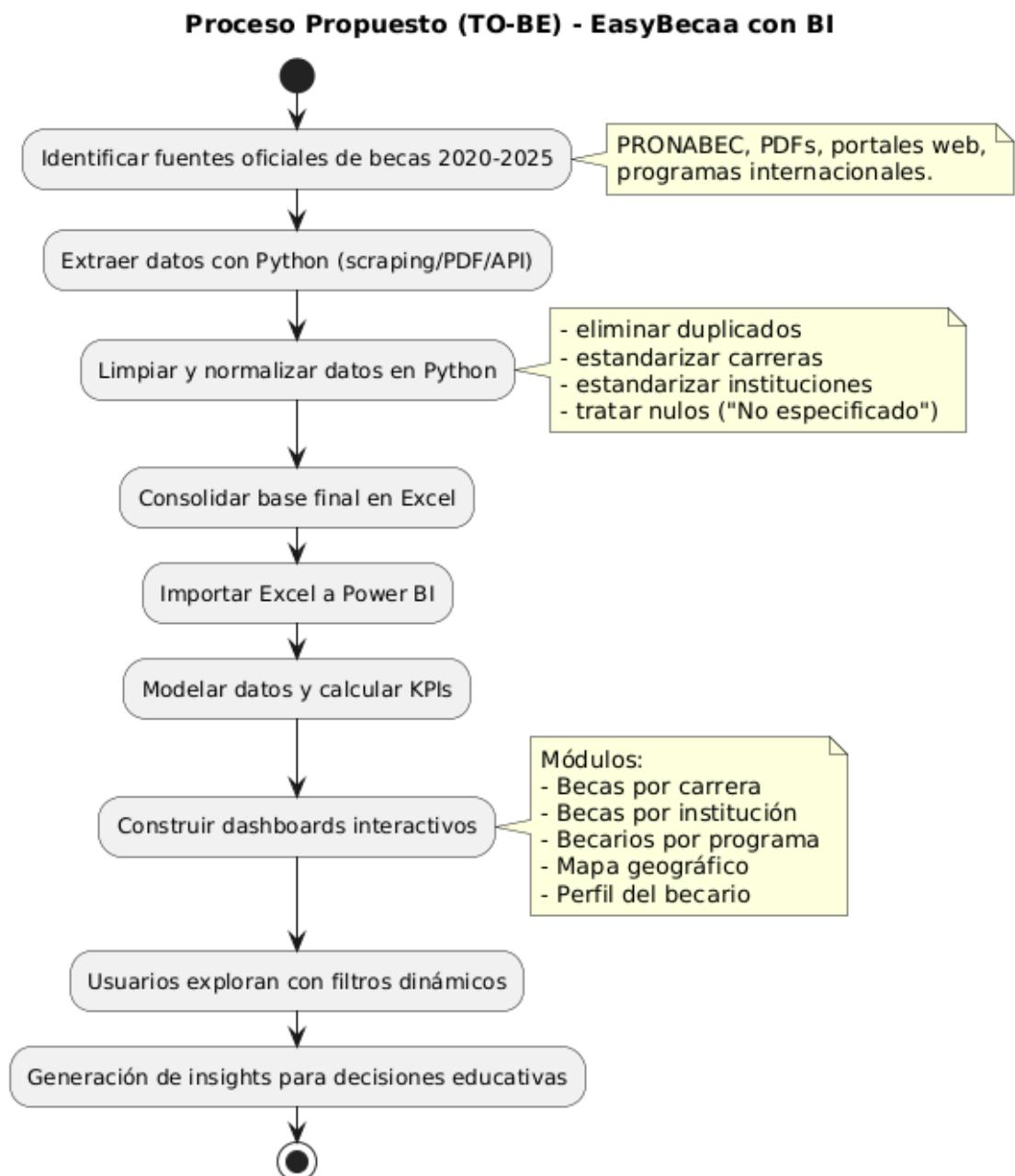
3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)





3.4. Vista de procesos

3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)

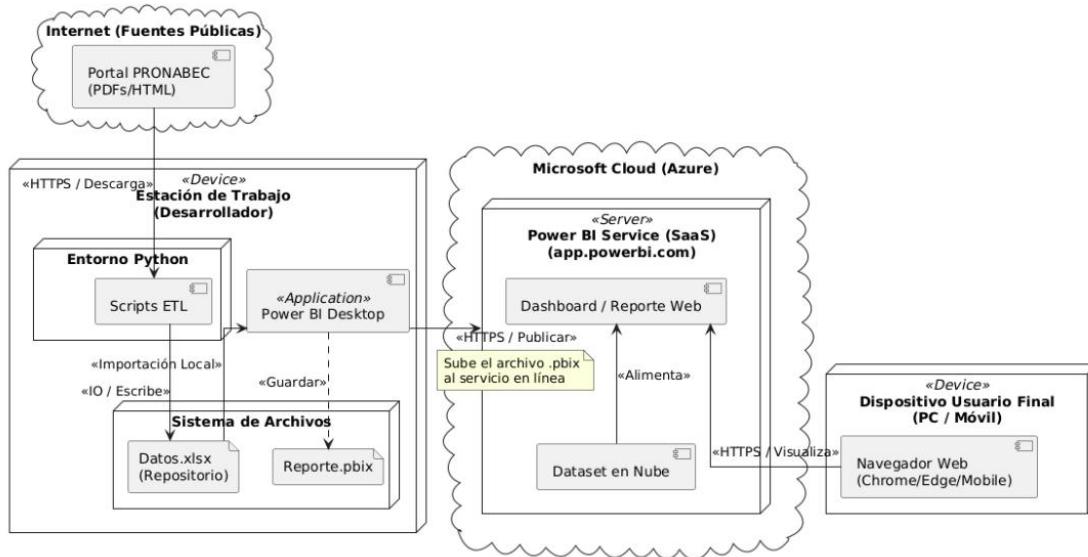


3.5. Vista de Despliegue (vista física)

3.5.1. Diagrama de despliegue



Diagrama de Despliegue - EasyBeca (Publicación Web)



4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

Escenario de Funcionalidad

Un usuario analista desea incorporar los datos de la convocatoria "Beca 18 - 2025". El sistema, mediante la ejecución del script ETL, debe ser capaz de procesar el nuevo PDF publicado por PRONABEC, extraer la lista de becarios, normalizar los nombres de las instituciones y agregar estos registros al archivo Excel histórico sin duplicar información existente. Finalmente, al abrir Power BI y pulsar "Actualizar", los nuevos datos deben reflejarse en todos los dashboards y KPIs automáticamente.

Escenario de Usabilidad

Un estudiante de secundaria sin conocimientos técnicos accede a EasyBeca para buscar oportunidades. Debe poder abrir el archivo de Power BI y, utilizando únicamente el ratón, filtrar el dashboard "Becas por carrera" seleccionando "Ingeniería de Sistemas". El sistema debe responder mostrando inmediatamente las instituciones que más becas han otorgado para esa carrera y la distribución geográfica, sin que el usuario tenga que escribir consultas ni configurar parámetros complejos.

Escenario de confiabilidad



Durante la ejecución del proceso de limpieza, el script de Python encuentra una institución con un nombre malformado o caracteres extraños en el archivo fuente. El sistema debe manejar este error clasificando el dato como "No especificado" o aplicando una regla de corrección difusa, en lugar de detener la ejecución abruptamente. Esto asegura que la generación del reporte final no se interrumpa por inconsistencias menores en los datos de entrada.

Escenario de rendimiento

El rendimiento del sistema se evalúa bajo condiciones de carga máxima. Se plantea un escenario donde 20 usuarios concurrentes inician sesión y acceden a su historial de presentaciones simultáneamente, simulando un pico de uso como el inicio de una clase. El servidor de aplicaciones y la base de datos deben procesar todas estas solicitudes de autenticación y consultas de manera eficiente. El rendimiento se considerará aceptable si el tiempo de carga de la página de historial para el 95% de los usuarios es inferior a 4 segundos

Escenario de mantenibilidad

Un usuario interactúa con el "Mapa de becas" que contiene miles de puntos de datos geográficos. Al hacer clic en el departamento de "Arequipa", el dashboard debe filtrar y actualizar los gráficos de "Top Instituciones" y "Tipos de Beca" correspondientes a esa región en un tiempo menor a 5 segundos, garantizando una experiencia de usuario fluida incluso con la carga completa de datos históricos (2020-2025).

Otros Escenarios

Finalmente, otro aspecto del rendimiento se mide en la capacidad de respuesta interactiva del dashboard durante el análisis de datos. Este escenario se enfoca en la experiencia de exploración: cuando un usuario aplica un filtro cruzado (por ejemplo, al hacer clic en una región específica en el mapa o seleccionar un año en el segmentador de tiempo), todos los demás objetos visuales (gráficos de barras, tarjetas de KPIs y tablas detalladas) deben actualizarse para reflejar el contexto seleccionado de manera fluida. La calidad de esta interacción se medirá por una latencia inferior a 1 segundo entre la selección del filtro y la actualización



visible completa de todos los elementos del reporte, garantizando que el flujo de análisis no se vea interrumpido por tiempos de carga perceptibles.