



UNIVERSIDAD PRIVADA DE TACNA

FACULTAD DE INGENIERÍA

Escuela Profesional de Ingeniería de Sistemas

**Plataforma de Análisis del Mercado Laboral
Tecnológico**

Curso: Inteligencia de Negocios

Docente: Mag. Patrick Cuadros Quiroga

Integrantes:

Daleska Nicolle Fernandez Villanueva	(2021070308)
Andree Sebastian Flores Melendez	(2017057494)
Mario Antonio Flores Ramos	(2018000597)

**Tacna – Perú
2025**

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
2.0	AFM	AFM	AFM	10/10/2025	Versión Original

**Plataforma de Análisis del Mercado Laboral
Tecnológico
Documento de Arquitectura de Software**

Versión 2.0

CONTROL DE VERSIONES					
Versión	Hecha por	Revisada por	Aprobada por	Fecha	Motivo
2.0	AFM	AFM	AFM	09/06/2025	Versión Original

Contenido

1. INTRODUCCIÓN	5
1.1. Propósito (Diagrama 4+1)	5
1.2. Alcance	5
1.3. Definición, siglas y abreviaturas	5
1.4. Organización del documento	5
2. OBJETIVOS Y RESTRICCIONES ARQUITECTONICAS	5
2.1.1. Requerimientos Funcionales	5
2.1.2. Requerimientos No Funcionales – Atributos de Calidad	5
3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA	6
3.1. Vista de Caso de uso	6
3.1.1. Diagramas de Casos de uso	6
3.2. Vista Lógica	6
3.2.1. Diagrama de Subsistemas (paquetes)	7
3.2.2. Diagrama de Secuencia (vista de diseño)	7
3.2.3. Diagrama de Colaboración (vista de diseño)	7
3.2.4. Diagrama de Objetos	7
3.2.5. Diagrama de Clases	7
3.2.6. Diagrama de Base de datos (relacional o no relacional)	7
3.3. Vista de Implementación (vista de desarrollo)	7
3.3.1. Diagrama de arquitectura software (paquetes)	7
3.3.2. Diagrama de arquitectura del sistema (Diagrama de componentes)	7
3.4. Vista de procesos	7
3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)	8
3.5. Vista de Despliegue (vista física)	8
3.5.1. Diagrama de despliegue	8
4. ATRIBUTOS DE CALIDAD DEL SOFTWARE	8
Escenario de Funcionalidad	8
Escenario de Usabilidad	8
Escenario de confiabilidad	9
Escenario de rendimiento	9
Escenario de mantenibilidad	9
Otros Escenarios	9

1. INTRODUCCIÓN

1.1. Propósito (Diagrama 4+1)

El enfoque arquitectónico se basa en el modelo 4+1 vistas de Kruchten, adaptado al contexto de un dashboard analítico desarrollado con Power BI. El objetivo es ofrecer una visión integral que incluya las distintas perspectivas técnicas, de usuario y de implementación.

Vista Lógica (Modelo de Datos):

Define la estructura de los datos, relaciones entre tablas (dimensiones y hechos), y medidas calculadas. En Power BI, esto se implementa mediante modelos tabulares y lenguaje DAX para análisis dinámico. Ejemplo: relaciones entre la tabla de ofertas laborales, tecnologías, regiones y perfiles de usuarios.

Vista de Procesos (Flujos de Interacción):

Describe cómo los usuarios interactúan con el dashboard. Incluye filtros por región, años, tecnologías, experiencia, etc. También cubre el comportamiento de los visuales al seleccionar datos. Ejemplo: un usuario selecciona "Python" y visualiza automáticamente los salarios promedio, demanda por región y crecimiento de empleos.

Vista Física (Infraestructura y Datos):

Indica cómo se obtienen, transforman y actualizan los datos. Los datos pueden provenir de APIs de empleo (Computrabajo, LinkedIn), archivos CSV, web scraping, o bases de datos en la nube (Amazon S3). Se definen las fuentes, frecuencia de actualización y almacenamiento temporal.

Vista de Escenarios (Casos de uso):

Describe cómo diferentes tipos de usuarios acceden y utilizan el dashboard. Ejemplo:

- Estudiante consulta habilidades más demandadas en su región.
- Universidad revisa estadísticas para actualizar su plan curricular.
- Profesional compara su perfil con el mercado para mejorar su empleabilidad.

Vista de Implementación (Componentes de Power BI):

Incluye las páginas del dashboard, visuales utilizados (gráficos de barras, líneas, mapas), y filtros avanzados. También se considera el uso de bookmarks, tooltips y paneles de navegación.

Este enfoque arquitectónico permite que el diseño del dashboard sea modular, escalable y alineado con las necesidades de los usuarios, priorizando la eficiencia de análisis sobre la portabilidad de plataforma, ya que Power BI es la herramienta base de desarrollo.

1.2. Alcance

Este proyecto contempla el desarrollo de un dashboard web interactivo que permitirá a estudiantes, egresados y profesionales de carreras de ingeniería de sistemas, software y ciencias de la computación explorar y analizar datos del mercado laboral relacionados con su campo profesional. El sistema se centrará en la recopilación, visualización y análisis de ofertas laborales con el fin de apoyar la toma de decisiones académicas y profesionales.

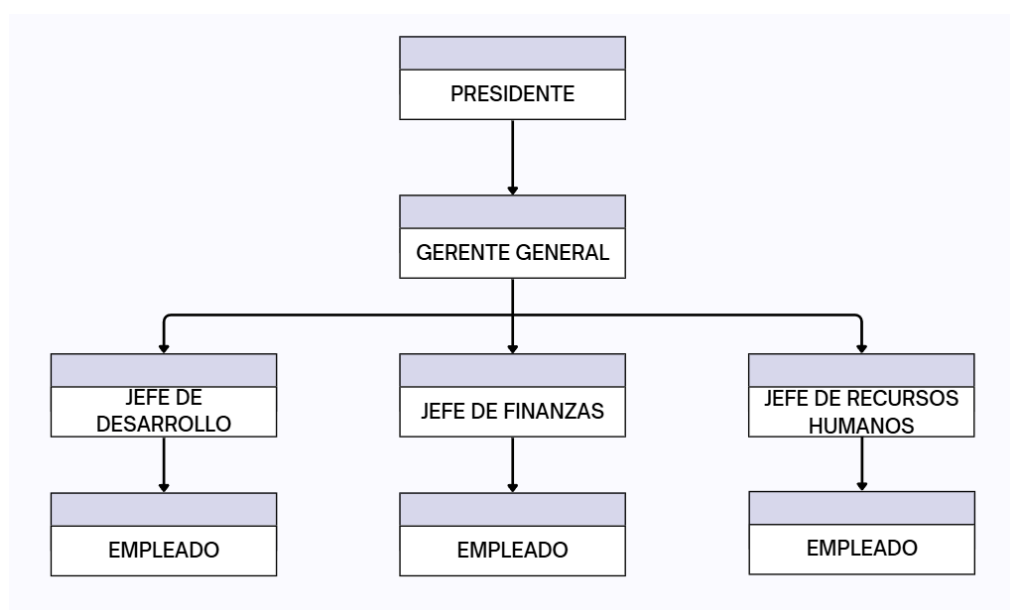
1.3. Definición, siglas y abreviaturas

En este apartado se explican los términos, acrónimos y abreviaturas que se utilizan en este documento para facilitar su comprensión. Se incluyen tanto términos técnicos como específicos del contexto del dashboard para análisis del mercado laboral tecnológico.

- Dashboard: Interfaz visual que muestra datos e indicadores clave para facilitar la interpretación y toma de decisiones.

- Fuente de Datos: Origen desde donde se obtienen los datos, como portales de empleo, bases de datos públicas, APIs, etc.
- KPIs (Key Performance Indicators): Indicadores clave de rendimiento que se muestran en el dashboard para medir tendencias y desempeño.
- Power BI: Herramienta de Microsoft para la visualización de datos y creación de dashboards interactivos.
- SaaS (Software as a Service): Modelo de distribución de software en el que las aplicaciones se alojan en la nube y se accede a ellas vía internet.
- Tendencias Laborales: Cambios y patrones observados en la demanda de empleo, habilidades y tecnologías dentro del mercado laboral.
- Usuarios: Personas que interactúan con el dashboard, como estudiantes, profesionales, docentes y reclutadores.

1.4. Organización del documento



2. OBJETIVOS Y RESTRICCIONES ARQUITECTÓNICAS

1.1.1. Requerimientos Funcionales

ID	Descripción	Prioridad
RF001	El administrador sube un archivo CSV con nuevas ofertas laborales a la aplicación de limpieza, la cual procesa y estandariza los datos, y los sube al data lake (S3) para su posterior catalogación y consumo por el dashboard.	Alta
RF002	El sistema debe permitir al usuario visualizar una lista de ofertas laborales detalladas, mostrando al menos: título, empresa, ubicación, fecha de publicación, tipo de contrato y descripción.	Alta
RF003	El sistema debe permitir al usuario filtrar la lista de ofertas laborales por ubicación geográfica (región, ciudad).	Alta
RF004	El sistema debe permitir al usuario filtrar la lista de ofertas laborales por tipo de contrato (ej. Plazo Indeterminado, Por Inicio de Actividad, etc.).	Alta
RF005	El sistema debe permitir al usuario filtrar la lista de ofertas laborales por nivel de experiencia requerido (ej. Junior, Semi-Senior, Senior, años de experiencia).	Alta
RF006	El sistema debe permitir al usuario filtrar la lista de ofertas laborales por lenguajes de programación específicos demandados (ej. Python, Java, C#, JavaScript, etc.).	Alta
RF007	El sistema debe permitir al usuario filtrar la lista de	Alta

	ofertas laborales por herramientas o frameworks específicos demandados (ej. React, Django, Docker, SQL Server, etc.).	
RF008	El sistema debe mostrar un resumen visual (ej. gráfico de barras, nube de palabras) de las habilidades y tecnologías (lenguajes, frameworks, herramientas, BBDD) más demandadas en el conjunto de ofertas.	Alta
RF009	El sistema debe permitir visualizar la distribución geográfica de las ofertas laborales en un mapa interactivo o gráfico de barras por región/ciudad.	Alta
RF0010	El sistema debe permitir al usuario filtrar o segmentar el análisis geográfico por tipo de modalidad de trabajo (presencial, remoto, híbrido).	Alta
RF011	El sistema debe mostrar gráficos interactivos (ej. boxplot, barras) que presenten el rango o promedio salarial por especialización (basada en combinaciones de tecnologías/roles).	Alta
RF012	El sistema debe permitir filtrar o segmentar los gráficos de tendencias salariales por nivel de experiencia.	Alta
RF013	El sistema debe permitir filtrar o segmentar los gráficos de tendencias salariales por ubicación geográfica.	
RF014	El sistema debe mostrar un resumen visual (ej. gráfico de pastel, barras) de la distribución de ofertas laborales por nivel educativo requerido	Alta

	(Técnico, Universitario, Maestría, etc.).	
RF015	El sistema debe permitir al usuario ver el detalle completo de una oferta laboral seleccionada, incluyendo enlace a la fuente original si está disponible.	Alta

1.1.2. Requerimientos No Funcionales – Atributos de Calidad

[Definir la prioridad de los requerimientos NO funcionales.]

ID	Descripcion	Prioridad

[Los Atributos de Calidad (QAs) son propiedades medibles y evaluables de un sistema, estas propiedades son usadas para indicar el grado en que el sistema satisface las necesidades de los stakeholders [Wojcik 2013].

Los QAs además son concebidos como aquellos requerimientos que no son funcionales. De hecho, la funcionalidad es mayormente ortogonal a los QAs; un diseño puede cumplir con la funcionalidad deseada y fallar a la hora de satisfacer sus requerimientos de calidad. De esta manera, se entiende a la funcionalidad como la capacidad del sistema para hacer el trabajo para el cual fue pensado, independientemente de la estructura. Existen QAs mayormente usados que se suelen identificar en numerosos sistemas y se tienen que describir, aunque la lista no es fina ya que muy a menudo hay situaciones en que podrían identificarse y proponerse nuevas propiedades para las diversas necesidades de stakeholders.]

2.1. Restricciones

[Aquí van las restricciones del proyecto]

3. REPRESENTACIÓN DE LA ARQUITECTURA DEL SISTEMA

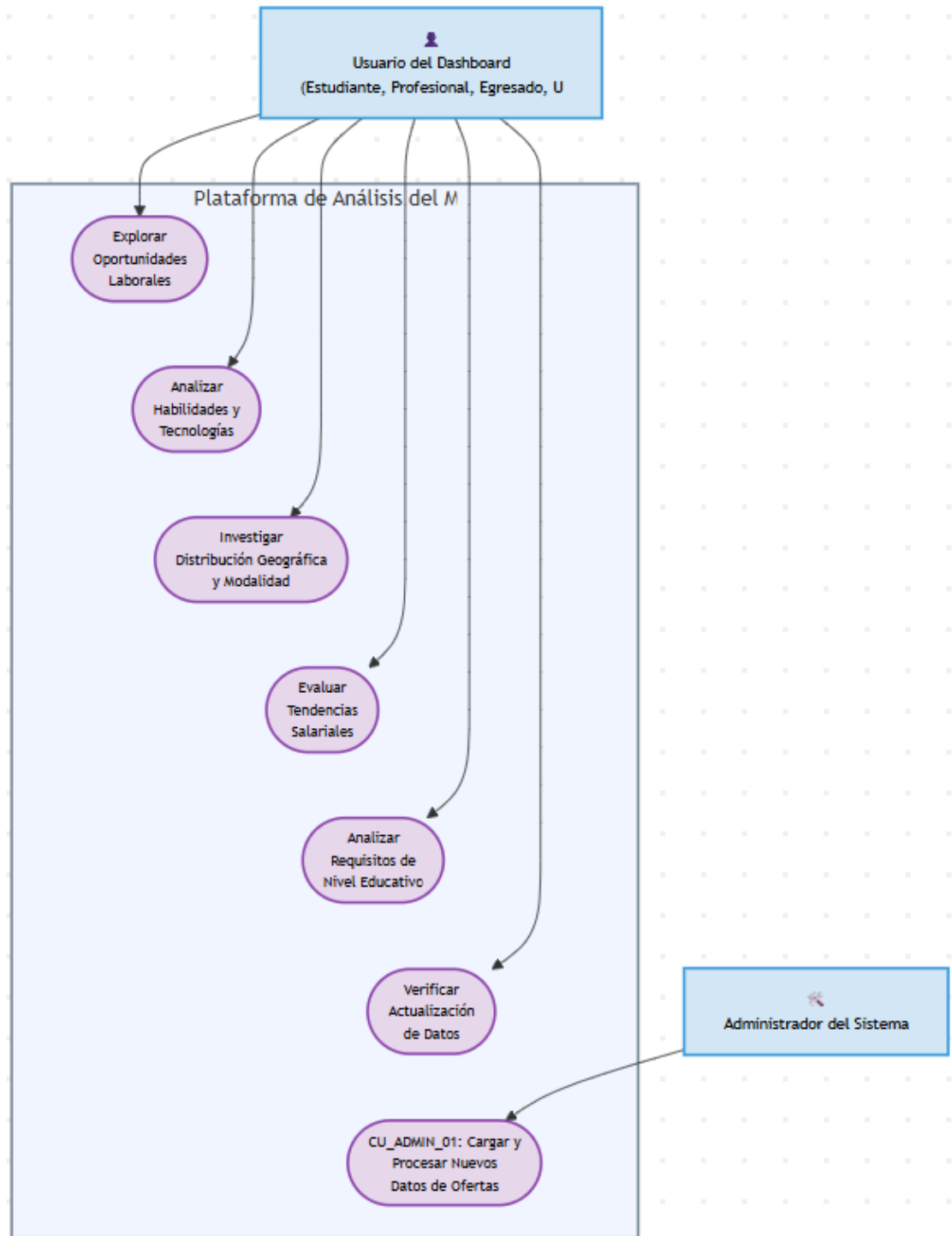
3.1. Vista de Caso de uso

[En esta sección se describen los casos de uso del sistema (nombre de la aplicación), donde se abarcan todas las funcionalidades del sistema, se muestran los actores que interactúan en el sistema y las funcionalidades asociadas; asimismo se listará los casos de uso o escenarios del modelo de casos de uso que representen funcionalidades centrales del sistema final, que requieran una gran cobertura arquitectónica o aquellos que impliquen algún punto especialmente delicado de la arquitectura.

La documentación a incluir en esta sección corresponde a la obtenida como consecuencia de la actividad “Realización de casos de uso”:


- Flujos de eventos- Diseño: descripción textual de cómo se realiza el caso de uso en términos de los objetos que colaboran. Resumen de los diagramas conectados con el caso de uso y explicación de sus relaciones.
- Diagramas de interacción: Diagramas de secuencia, Diagramas de colaboración, objetos participantes, Diagramas de clases.
- Requisitos derivados: Descripción textual que recoge todos los requisitos, normalmente los no funcionales, de la realización del caso de uso no que han de tenerse en cuenta durante la implementación]

1.1.3. Diagramas de Casos de uso



graph TD

%% Actores Externos %%

ActorUsuarioDashboard[" Usuario del Dashboard (Estudiante, Profesional, Egresado, Universidad)"]

```
ActorAdmin["🔧<br>Administrador del Sistema"]
```

```
%% Límite de la Plataforma %%
```

```
subgraph PlataformaAnalisis["Plataforma de Análisis del Mercado Laboral Tecnológico"]
```

```
direction TB %% Forzar dirección Top-Bottom para los elementos internos del subgraph
```

```
%% Casos de Uso del Dashboard (Usuarios Finales) - Forma de Estadio/Óvalo %%
```

```
CU01(["Explorar<br>Oportunidades<br>Laborales"])
```

```
CU02(["Analizar<br>Habilidades y<br>Tecnologías"])
```

```
CU03(["Investigar<br>Distribución Geográfica<br>y Modalidad"])
```

```
CU04(["Evaluar<br>Tendencias<br>Salariales"])
```

```
CU05(["Analizar<br>Requisitos de<br>Nivel Educativo"])
```

```
CU06(["Verificar<br>Actualización<br>de Datos"])
```

```
%% Caso de Uso de Administración - Forma de Estadio/Óvalo %%
```

```
CU_ADMIN_01(["CU_ADMIN_01: Cargar y<br>Procesar Nuevos<br>Datos de Ofertas"])
```

```
%% Líneas invisibles para guiar el layout vertical DENTRO del subgraph
```

```
CU01 ~~~ CU02
```

```
CU02 ~~~ CU03
```

```
CU03 ~~~ CU04
```

```
CU04 ~~~ CU05
```

```
CU05 ~~~ CU06
```

```
CU06 ~~~ CU_ADMIN_01
```

```
%% El comentario anterior estaba aquí, causando el error.
```

```
%% La línea anterior (CU06 ~~~ CU_ADMIN_01) ahora está limpia.
```

```
end
```

%% Relaciones Actor -> Caso de Uso %%

ActorUsuarioDashboard --> CU01

ActorUsuarioDashboard --> CU02

ActorUsuarioDashboard --> CU03

ActorUsuarioDashboard --> CU04

ActorUsuarioDashboard --> CU05

ActorUsuarioDashboard --> CU06

ActorAdmin --> CU_ADMIN_01

%% Estilos %%

```
classDef actor
fill:#D6EAF8,stroke:#3498DB,stroke-width:2px,color:#000,font-size:12px
,text-align:center;
```

```
classDef usecaseStadium
fill:#E8DAEF,stroke:#8E44AD,stroke-width:2px,color:#000,font-size:11px
,text-align:center,shape:stadium;
```

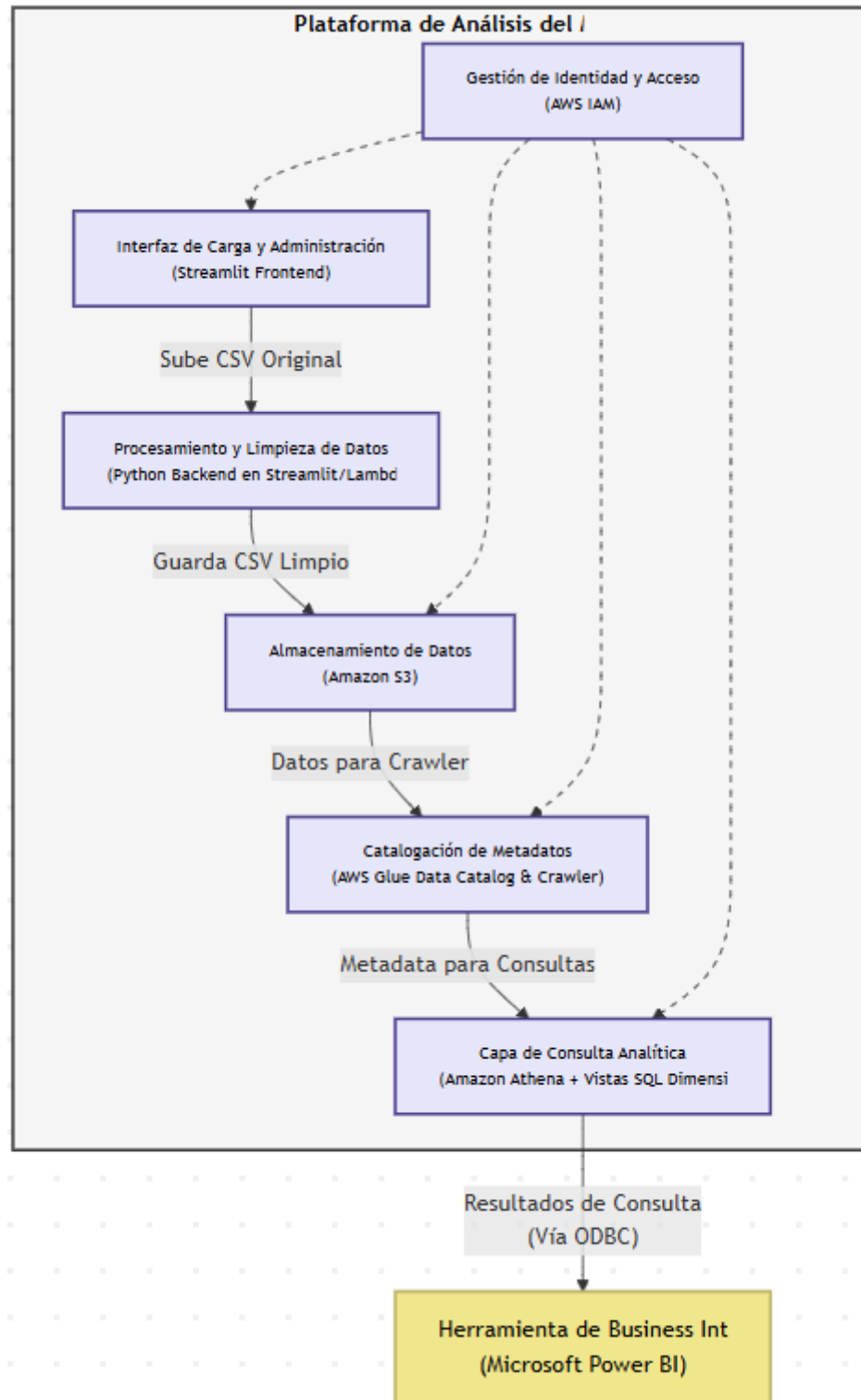
```
classDef platformBoundary
fill:#f0f8ff,stroke:#778899,stroke-width:2px,color:#333;
```

```
class ActorUsuarioDashboard,ActorAdmin actor;
class CU01,CU02,CU03,CU04,CU05,CU06,CU_ADMIN_01
usecaseStadium;
class PlataformaAnalisis platformBoundary;
```

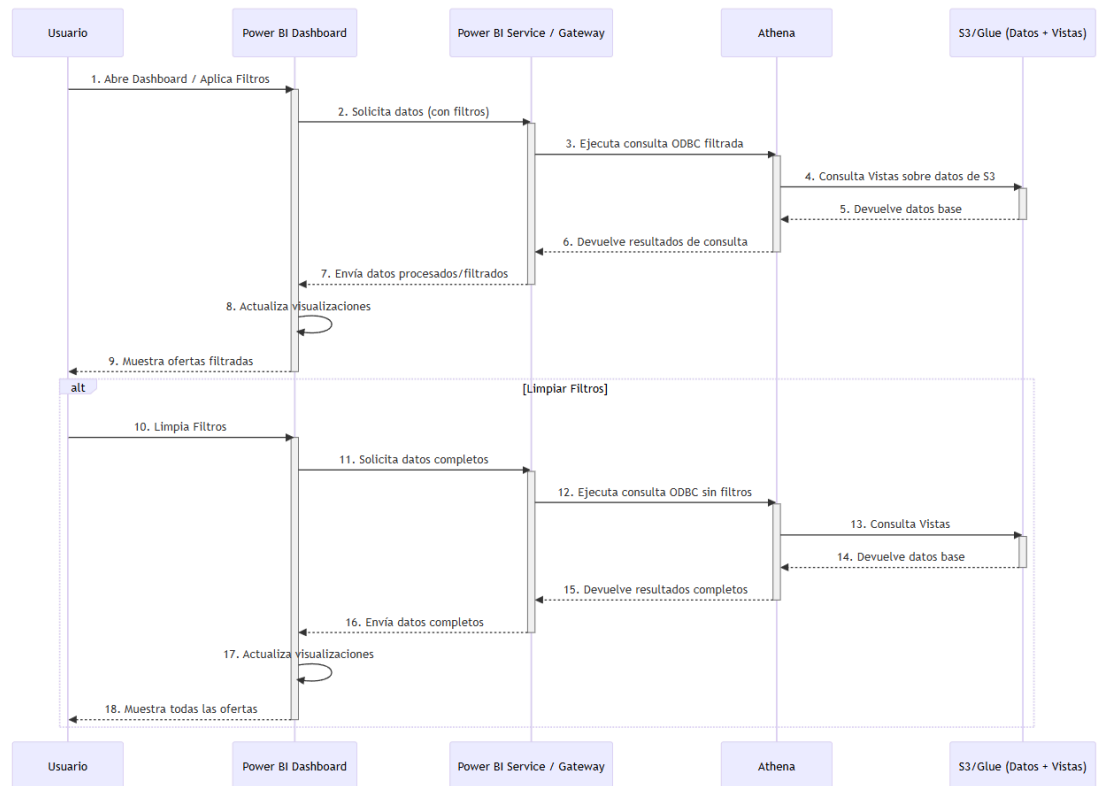
3.2. Vista Lógica

[La vista lógica se encarga de representar los requerimientos funcionales del sistema. Esta sección describe las partes del diseño del modelo significativas para la arquitectura, tales como subsistemas y paquetes.]

3.2.1. Diagrama de Subsistemas (paquetes)



3.2.2. Diagrama de Secuencia (vista de diseño)



3.2.3. Diagrama de Colaboración (vista de diseño)

3.2.4. Diagrama de Objetos

3.2.5. Diagrama de Clases

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

3.3. Vista de Implementación (vista de desarrollo)

[Se detalla la estructura general del Modelo de Implementación y el mapeo de los subsistemas, paquetes y clases de la Vista Lógica a subsistemas y componentes de implementación de manera más detallada]

3.3.1. Diagrama de arquitectura software (paquetes)

[Se detalla la manera como fue implementado el sistema propuesto, se describe visualmente las capas que tiene el sistema, como están distribuidas y sus principales funciones]

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

[Se detalla la manera como fue implementado el sistema propuesto, se describe visualmente las capas que tiene el sistema, como están distribuidas y sus principales funciones]

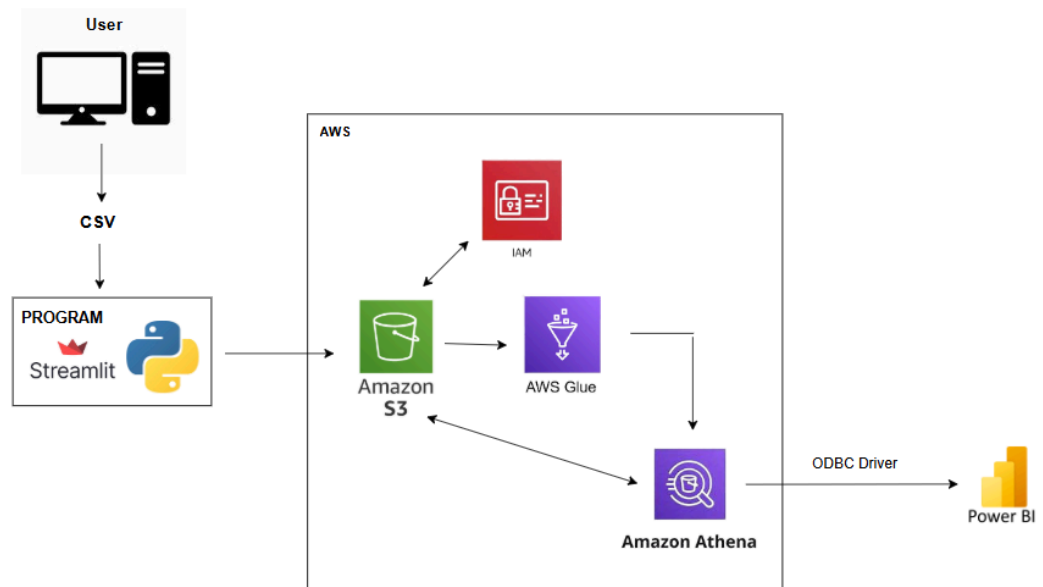
3.4. Vista de procesos

[Describe la descomposición del sistema procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican.]

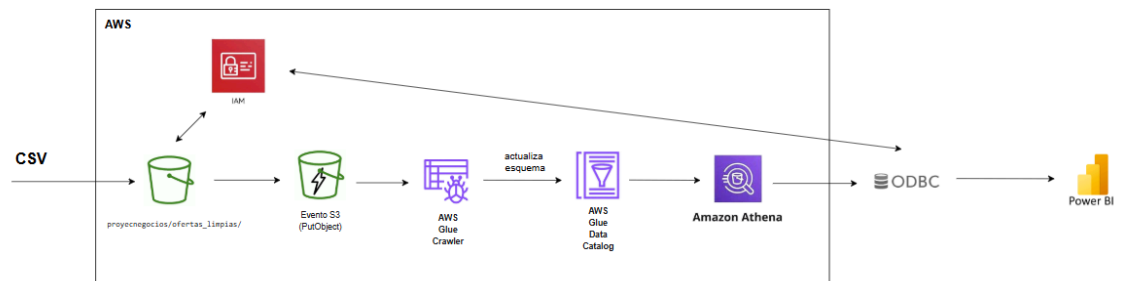
3.4.1. Diagrama de Procesos del sistema (diagrama de actividad)

[Se realizará un diagrama del o los procesos del sistema donde se exponga las actividades donde interviene el sistema propuesto, adicionando diagramas que definan el detalle la descomposición del sistema en procesos pesados. Indica que procesos o grupos de procesos se comunican o interactúan entre sí y los modos en que estos se comunican]

3.5. Vista de Arquitectura en nube alineada a las fuentes de datos extraídas



3.5.1. Arquitectura de Procesamiento de datos automatizada



4. ATRIBUTOS DE CALIDAD DEL SOFTWARE

[Los Atributos de Calidad (QAs) son propiedades medibles y evaluables de un sistema, estas propiedades son usadas para indicar el grado en que el sistema satisface las necesidades de los stakeholders [Wojcik 2013].

Los QAs además son concebidos como aquellos requerimientos que no son funcionales. De hecho, la funcionalidad es mayormente ortogonal a los QAs; un diseño puede cumplir con la funcionalidad deseada y fallar a la hora de satisfacer

sus requerimientos de calidad. De esta manera, se entiende a la funcionalidad como la capacidad del sistema para hacer el trabajo para el cual fue pensado, independientemente de la estructura. Existen QAs mayormente usados que se suelen identificar en numerosos sistemas y se tienen que describir, aunque la lista no es fina ya que muy a menudo hay situaciones en que podrían identificarse y proponerse nuevas propiedades para las diversas necesidades de stakeholders.]

Escenario de Funcionalidad

[se califica de acuerdo con el conjunto de características y capacidades del programa, la generalidad de las funciones que se entregan y la seguridad general del sistema.]

Escenario de Usabilidad

[Este atributo de calidad se refiere a la facilidad con la que un usuario puede aprender a utilizar e interpretar los resultados producidos por un sistema [Barbacci 1995]. Para este atributo de calidad, se suelen considerar diversos aspectos de la interacción humano computadora, tales como: aprendizaje del sistema, utilización eficiente del sistema, minimización del impacto de errores, adaptación del sistema a las necesidades del usuario, confianza y satisfacción, entre otros.]

Escenario de confiabilidad

[Es el equilibrio entre la confidencialidad, la integridad, la irrefutabilidad y la disponibilidad de la información y datos manipulados por el sistema. Se trata del estado de un sistema, el cual puede ser transitorio y volátil. La seguridad de un sistema se caracteriza por mecanismos y técnicas empleados para intentar reducir lo más posible el impacto provocado por un ataque, y las amenazas (entendidas como los caminos mediante los cuales se pueden provocar un ataque).

Abarca los planos de observación físico, lógico y humanos. Posee tres tipos de enfoque: prevención, precaución y reacción.]

Escenario de rendimiento

[Se mide con base en la velocidad de procesamiento, el tiempo de respuesta, el uso de recursos, el conjunto y la eficiencia.] (Pressman 2010, pág. 187)

Escenario de mantenibilidad

[Combina la capacidad del programa para ser ampliable (extensibilidad), adaptable y servicial. (Pressman 2010, pág. 187)

Otros Escenarios

[“Otros escenarios como por ejemplo: Performance”

Performance: El atributo de calidad Performance se refiere a la capacidad de responder, ya sea el tiempo requerido para responder a eventos determinados, o bien, la cantidad de eventos procesados en un intervalo de tiempo dado. La Performance caracteriza la proyección en el tiempo de los servicios entregados por el sistema.]