



UNIVERSIDAD POLITÉCNICA DE CATALUÑA

FACULTAD DE INFORMÁTICA DE BARCELONA (FIB)

Especificación de Requisitos y Diseño

Parte I: Descripción de Casos de Uso

Parte II: Descripción del Diagrama de Clases

Parte III: Funcionalidades Extras y Algoritmia

Parte IV: Drivers, Tests y Manual de Ejecución

Parte V: Planificación y Gestión

Parte VI: Arquitectura de tres capas

Parte VII: Interfaz gráfica (GUI) y Manual de usuario

Asignatura:

Proyecto de Programación (PROP)

Integrantes del grupo:

Ramón García

Albert Porta

Dani del Rio

Jesús Peñaranda

Cuatrimestre Q1 – Curso 2025-2026

Índice

1. Descripción de Casos de Uso	5
1.1. Gestión de Encuestas	5
1.1.1. CU-01: Alta encuesta	5
1.1.2. CU-02: Importar encuesta	5
1.1.3. CU-03: Guardar encuesta	5
1.1.4. CU-04: Consultar encuesta	5
1.1.5. CU-05: Editar encuesta	6
1.1.6. CU-06: Baja encuesta	6
1.2. Gestión de Respuestas y Análisis	6
1.2.1. CU-07: Responder encuesta	6
1.2.2. CU-08: Importar respuesta	6
1.2.3. CU-09: Guardar respuesta	7
1.2.4. CU-10: Importar pregunta	7
1.2.5. CU-11: Análisis encuesta	7
1.3. Gestión de Perfiles y Clustering	7
1.3.1. CU-12: Obtener perfiles	7
1.3.2. CU-13: Obtener calidad clustering	7
1.3.3. CU-14: Importar perfiles	8
1.3.4. CU-15: Exportar perfiles	8
1.3.5. CU-16: Obtener representante	8
1.3.6. CU-17: Listar individuos	8
1.3.7. CU-18: Salir	8
2. Descripción del Diagrama de Clases	11
2.1. Modelo de Dominio	11
2.1.1. Clase: Encuesta	11
2.1.2. Clase: Persona	11
2.1.3. Clase: Pregunta (Abstracta)	11
2.1.4. Clase: Única (Hereda de Pregunta)	12
2.1.5. Clase: Múltiple (Hereda de Pregunta)	12
2.1.6. Clase: Libre (Hereda de Pregunta)	12
2.1.7. Clase: Cjt_Respuestas	12
2.1.8. Clase: Respuesta (Abstracta)	13
2.1.9. Clase: Numérica (Hereda de Respuesta)	13
2.1.10. Clase: Textual (Hereda de Respuesta)	13
2.1.11. Clase: Perfil	13
2.2. Módulo de Clustering y Algoritmia	13
2.2.1. Clase: IEstrategiaAlgoritmo (Interfaz)	13
2.2.2. Clase: Kmeans (Implementa IEstrategiaAlgoritmo)	14
2.2.3. Clase: Kmedoides (Implementa IEstrategiaAlgoritmo)	14
2.2.4. Clase: ClusteringSub (Implementa IEstrategiaAlgoritmo)	14
2.2.5. Clase: IEstrategiaIni (Interfaz)	14
2.2.6. Clase: IniKmeansPP (Implementa IEstrategiaIni)	15
2.2.7. Clase: IniTrivial (Implementa IEstrategiaIni)	15
3. Funcionalidades Extras y Algoritmia Avanzada	16
3.1. Módulo de Análisis de Datos (Clustering)	16
3.1.1. Algoritmo K-Means	16
3.1.2. Algoritmo K-Medoides	16

3.1.3.	Estrategias de Inicialización	16
3.2.	Funcionalidades Extras	17
3.2.1.	Algoritmos de Optimización y Calidad	17
3.2.2.	Simulación Gráfica 2D	17
3.2.3.	Algoritmo K-Medoides	18
3.2.4.	Filtrado Selectivo de Preguntas	18
3.2.5.	Ayuda Interactiva y Manual de Usuario	18
3.2.6.	Sistema de Login y Persistencia de Sesión	20
4.	Pruebas y Ejecución del Sistema	21
4.1.	Drivers del Sistema	21
4.2.	Tests de Integración	22
4.2.1.	Test: IntegracionTest_Edades	22
4.2.2.	Test: IntegracionTest_Visual	23
4.2.3.	Test: IntegracionTest_Persistencia	23
4.2.4.	Test: IntegracionTest_Frontera	24
4.3.	Plan de Verificación Funcional y Pruebas de Robustez (GUI)	25
4.4.	Manual de Ejecución	26
5.	Planificación y Gestión del Proyecto	27
5.1.	Cronograma de Trabajo (Diagrama de <i>Gantt</i>)	27
5.2.	Gestión de Tareas y Roles	27
5.3.	Dificultades y Estado de las Ampliaciones	27
5.4.	Asignación Clases	28
6.	Arquitectura de tres capas	29
6.1.	Capa de Dominio	29
6.1.1.	Controlador de Dominio (ctrlDominio)	29
6.2.	Capa de Presentación	29
6.2.1.	Controlador de Presentación (ctrlPresentacion)	29
6.2.2.	Vistas del Sistema	30
6.3.	Capa de Persistencia	30
6.3.1.	Controlador de Persistencia (ctrlPersistencia)	30
6.3.2.	Estrategia de Almacenamiento	31
7.	Interfaz gráfica de usuario (GUI)	35
7.1.	Manual de Usuario y Guía de Uso	36

Índice de figuras

1.	Diagrama UML de Casos de Uso del Sistema	4
2.	Diagrama de Clases UML del Proyecto	10
3.	Diagrama de Clases Detallado: Capa de Presentación	32
4.	Diagrama de Clases Detallado: Capa de Dominio	33
5.	Diagrama de Clases Detallado: Capa de Persistencia	34
6.	Vista de Login	39
7.	Vista Principal	39
8.	Vista Creación de Encuesta (vacía)	40
9.	Encuesta creada	40
10.	Vista para Responder Encuestas	41
11.	Vista Edición Encuestas	41

12.	Vista Análisis	42
13.	Vista Análisis con elección automática número clusters (k)	42
14.	Vista de análisis extendido del clustering	43
15.	Gráfica que simula los puntos 2D del clustering	43

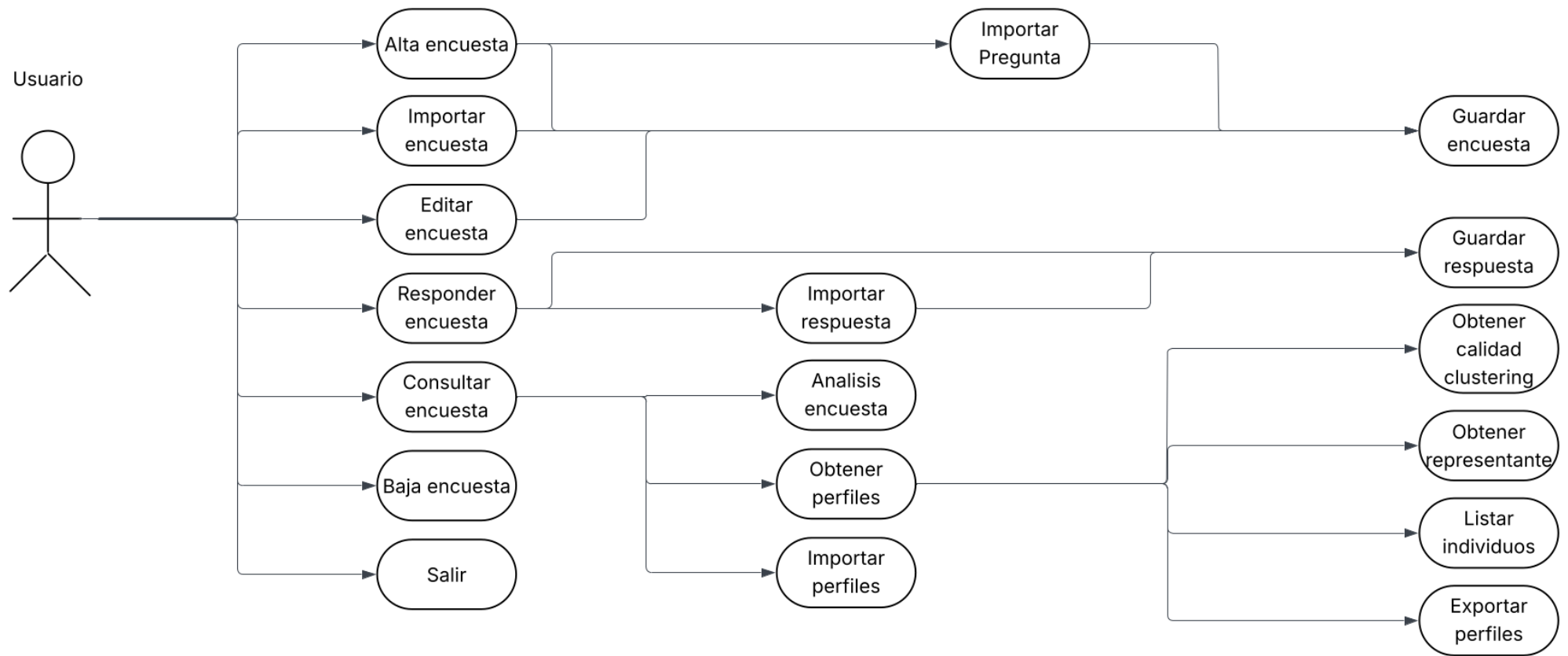


Figura 1: Diagrama UML de Casos de Uso del Sistema

Descripción de Casos de Uso

1.1 Gestión de Encuestas

1.1.1 CU-01: Alta encuesta

Actor	Usuario
Comportamiento	<ol style="list-style-type: none"> 1. El usuario indica al sistema que desea crear una encuesta. 2. El usuario proporciona el enunciado de una pregunta, el tipo de la pregunta, el tipo de las respuestas y las opciones (si aplica). 3. Si el usuario quiere seguir añadiendo preguntas, vuelve al paso 2. 4. El sistema crea una encuesta nueva con las preguntas proporcionadas. 5. El sistema informa al usuario del identificador de la encuesta creada.
Cursos Alternativos	<ul style="list-style-type: none"> – <i>Error de validación:</i> Si el enunciado, tipo o las opciones no son válidas, el sistema informa del error y vuelve al paso 2. – <i>Cancelación:</i> En cualquier momento, si el usuario decide acabar la interacción, la encuesta no se crea.

1.1.2 CU-02: Importar encuesta

Actor	Usuario
Comportamiento	<ol style="list-style-type: none"> 1. El usuario indica al sistema que desea importar una encuesta. 2. El sistema pide el identificador de la encuesta a importar. 3. El usuario proporciona el identificador. 4. El sistema importa la encuesta.
Cursos Alternativos	<ul style="list-style-type: none"> – <i>Fichero inexistente:</i> Si no existe un fichero con el nombre especificado, el sistema informa del error y vuelve al paso 2. – <i>Duplicidad:</i> Si ya existe una encuesta con ese identificador, el sistema informa del error y vuelve al paso 2. – <i>Cancelación:</i> El usuario puede cancelar la interacción en cualquier momento.

1.1.3 CU-03: Guardar encuesta

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	<ol style="list-style-type: none"> 1. El usuario indica que quiere guardar la encuesta. 2. El sistema guarda la encuesta en un fichero.

1.1.4 CU-04: Consultar encuesta

Actor	Usuario
Comportamiento	<ol style="list-style-type: none"> 1. El usuario indica que desea consultar una encuesta. 2. El sistema pide el identificador de la encuesta.

	3. El usuario lo proporciona.
	4. El sistema muestra las posibles acciones que puede realizar el usuario.
Cursos Alternativos	– <i>ID incorrecto</i> : Si el identificador no corresponde a ninguna encuesta, el sistema informa y vuelve al paso 2.

1.1.5 CU-05: Editar encuesta

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	1. El usuario indica que quiere editar la encuesta. 2. El sistema pregunta al usuario si quiere añadir o eliminar preguntas.

1.1.6 CU-06: Baja encuesta

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	1. El usuario indica su voluntad de eliminar una encuesta. 2. El sistema pide confirmación. 3. Si el usuario confirma, la encuesta se elimina definitivamente.

1.2 Gestión de Respuestas y Análisis

1.2.1 CU-07: Responder encuesta

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	1. El usuario indica que desea responder una encuesta. 2. El sistema muestra el enunciado de una pregunta y pide respuesta. 3. El usuario indica su respuesta o salta la pregunta. 4. Si quedan preguntas sin tratar, vuelve al paso 2. 5. El sistema añade las respuestas del usuario.
Cursos Alternativos	– Si el usuario salta una pregunta, esta quedará como no contestada.

1.2.2 CU-08: Importar respuesta

Actor	Usuario
Precondición	El usuario ha indicado que quiere contestar una encuesta.
Comportamiento	1. El usuario indica que quiere importar respuestas. 2. El sistema importa las respuestas.
Cursos Alternativos	– Si el fichero no existe, el sistema informa al usuario del error.

1.2.3 CU-09: Guardar respuesta

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	<ol style="list-style-type: none"> 1. El usuario indica que quiere guardar las respuestas. 2. El sistema guarda las respuestas de la encuesta.

1.2.4 CU-10: Importar pregunta

Actor	Usuario
Precondición	El usuario ha decidido crear una encuesta.
Comportamiento	<ol style="list-style-type: none"> 1. El usuario solicita importar una o varias preguntas ya generadas en disco local. 2. El sistema importa y guarda las preguntas en la encuesta actual.

1.2.5 CU-11: Análisis encuesta

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	<ol style="list-style-type: none"> 1. El usuario solicita realizar un análisis sobre las respuestas. 2. El sistema pide: número de clusters, algoritmo y tipo de inicialización. 3. El usuario proporciona los datos. 4. El sistema ejecuta el algoritmo.
Cursos Alternativos	<ul style="list-style-type: none"> – Si no hay respuestas recopiladas, el sistema informa del error. – Si los parámetros son incorrectos, el sistema informa del error y vuelve al paso 2.

1.3 Gestión de Perfiles y Clustering

1.3.1 CU-12: Obtener perfiles

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	<ol style="list-style-type: none"> 1. El usuario indica que quiere gestionar perfiles. 2. El sistema muestra las acciones disponibles.

1.3.2 CU-13: Obtener calidad clustering

Actor	Usuario
Precondición	El usuario ha ejecutado "Obtener perfiles".
Comportamiento	<ol style="list-style-type: none"> 1. El usuario indica que desea obtener la calidad del clustering.

Cursos Alternativos	2. El sistema muestra los índices de calidad disponibles. – Si no se ha ejecutado ningún algoritmo, el sistema informa del error y vuelve a "Obtener Perfiles".
----------------------------	---

1.3.3 CU-14: Importar perfiles

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	1. El usuario selecciona "Importar perfiles". 2. El sistema importa los perfiles relacionados.

1.3.4 CU-15: Exportar perfiles

Actor	Usuario
Precondición	El usuario ha decidido consultar una encuesta.
Comportamiento	1. El usuario solicita exportar los perfiles generados. 2. El sistema genera y guarda el archivo de perfiles.

1.3.5 CU-16: Obtener representante

Actor	Usuario
Precondición	El usuario ha ejecutado "Obtener perfiles".
Comportamiento	1. El usuario indica que quiere conocer el representante de un perfil. 2. El sistema pide el perfil. 3. El usuario lo indica. 4. El sistema muestra el representante.
Cursos Alternativos	– Si el perfil no existe, el sistema informa y vuelve al paso 2.

1.3.6 CU-17: Listar individuos

Actor	Usuario
Precondición	El usuario ha ejecutado "Obtener perfiles".
Comportamiento	1. El usuario desea listar individuos similares a un perfil. 2. El sistema pide el perfil. 3. El usuario lo indica. 4. El sistema muestra la lista de individuos.
Cursos Alternativos	– Si el perfil no existe, el sistema informa y vuelve al paso 2.

1.3.7 CU-18: Salir

Actor	Usuario
Precondición	El usuario ha ejecutado "Salir".

- Comportamiento**
1. El usuario desea salir de la aplicación.
 2. El sistema alerta del cierre de la aplicación con posible cancelación.
 3. El usuario confirma la voluntad de salir de la aplicación.
 4. La aplicación se cierra exitosamente.

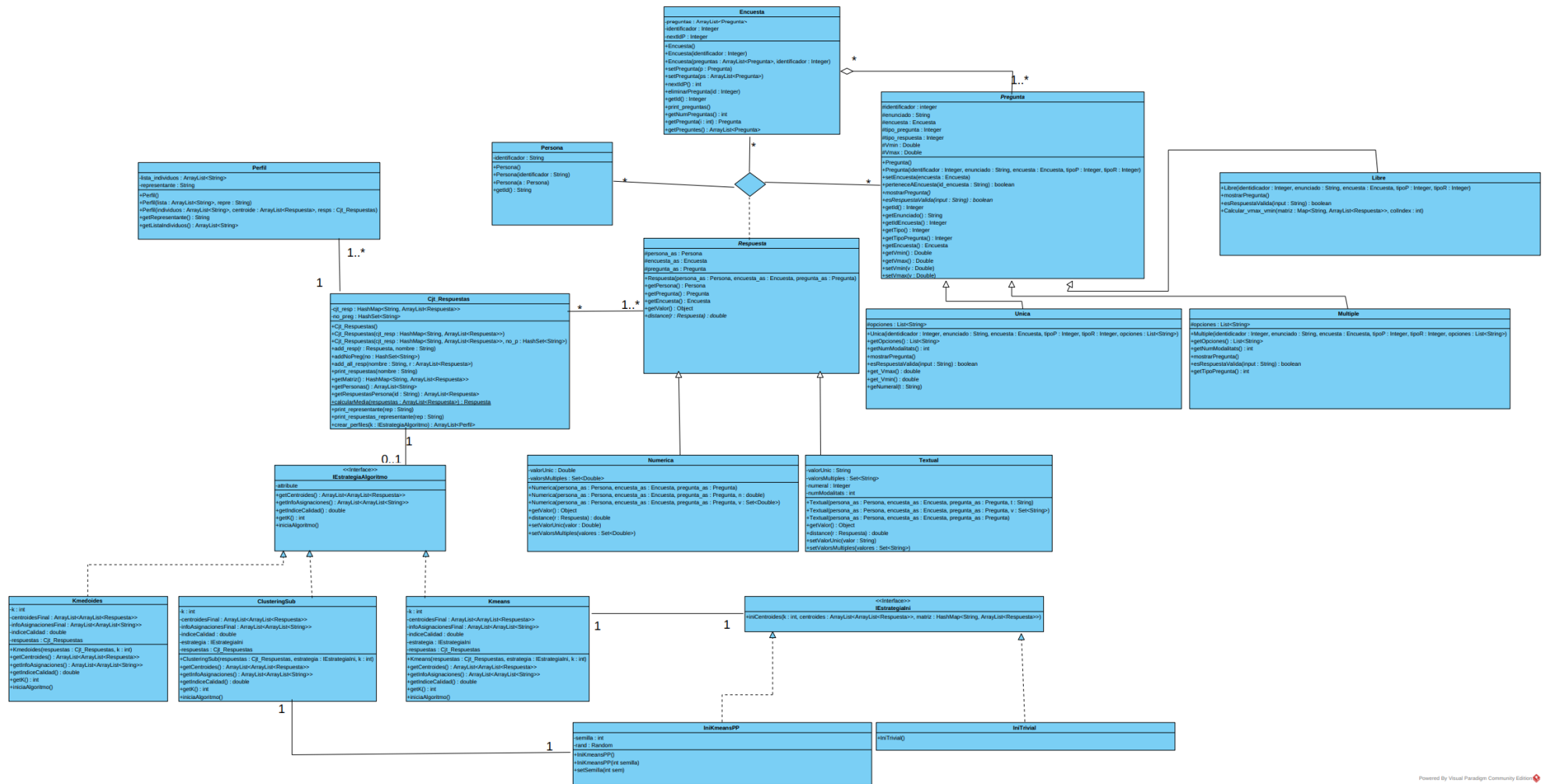


Figura 2: Diagrama de Clases UML del Proyecto

Descripción del Diagrama de Clases

2.1 Modelo de Dominio

2.1.1 Clase: Encuesta

Definición: Representa un cuestionario, creado o importado, que contiene una colección ordenada de preguntas.

Atributos:

- **identificador** (Integer): ID global único, fundamental para la gestión y unicidad de la encuesta. Permite valor nulo.
- **nextIdP** (Integer): Contador que genera IDs locales únicos y secuenciales para las preguntas internas.
- **preguntas** (ArrayList<Pregunta>): Colección ordenada de preguntas. Se utiliza ArrayList para acceso eficiente por índice y mantenimiento del orden.

Métodos clave:

- Constructoras para inicialización vacía, con ID, o con lista de preguntas (copia defensiva).
- **nextIdP()**: Genera y devuelve el siguiente ID disponible.
- **eliminarPregunta(Integer id)**: Elimina una pregunta por su ID local.

Relaciones:

- **Agregación** con Pregunta.
- **Asociación** con Persona.
- **Asociación** con Respuesta.

2.1.2 Clase: Persona

Definición: Representa a un individuo que responde encuestas en el sistema.

Atributos:

- **identificador** (String): ID único para rastreo flexible (nombres, alias o códigos).

Relaciones:

- **Asociación** con Pregunta.
- **Asociación** con Respuesta.
- **Asociación** con Encuesta.

2.1.3 Clase: Pregunta (Abstracta)

Definición: Clase base que representa un ítem de la encuesta. Se especializa en tipos Única, Múltiple o Libre.

Atributos:

- **identificador** (Integer): ID local único.
- **enunciado** (String): Texto visible para el usuario.
- **encuesta** (Encuesta): Referencia a la encuesta contenedora.
- **tipo_pregunta** (Integer): Clasificación de entrada (1: Libre, 2: Múltiple, 3: Única).
- **tipo_respuesta** (Integer): Clasificación para métricas (Cuantitativa, Cualitativa Ordenada/No Ordenada).
- **Vmax / Vmin** (Double): Valores extremos para normalización numérica en el clustering.

Métodos abstractos:

- `mostrarPregunta()`: Define la visualización según el tipo.
- `esRespuestaValida(String input)`: Valida la entrada del usuario.

Relaciones:

- **Asociación** con Pregunta.
- **Asociación** con Respuesta.
- **Agregación** con Encuesta.

2.1.4 Clase: Única (Hereda de Pregunta)

Definición: Especialización donde se selecciona una sola opción predefinida.

Atributos adicionales:

- `opciones (List<String>)`: Lista ordenada de opciones. Permite acceso por índice para validación.

Funcionalidad: Calcula automáticamente V_{max}/V_{min} si las opciones son convertibles a números. Valida que la selección esté en rango.

2.1.5 Clase: Múltiple (Hereda de Pregunta)

Definición: Especialización donde se seleccionan hasta n opciones.

Atributos adicionales:

- `opciones (List<String>)`: Lista ordenada de opciones.

Funcionalidad: Valida entradas separadas por comas, asegurando que cada selección esté dentro del rango de opciones disponibles.

2.1.6 Clase: Libre (Hereda de Pregunta)

Definición: Especialización para respuestas de texto libre o numéricas sin opciones predefinidas.

Funcionalidad: Validación básica (no vacío). Implementa `Calcular_vmax_vmin` leyendo directamente de la matriz de respuestas.

2.1.7 Clase: Cjt_Respuestas

Definición: Agrupa todas las respuestas de una Persona a una Encuesta. Es el objeto principal sometido a análisis.

Atributos:

- `cjt_resp (HashMap<String, ArrayList<String, Respuesta>>)`: Matriz de respuestas indexada por ID de Persona.
- `no_preg (HashSet<String>)`: Preguntas excluidas del análisis de perfiles.

Métodos clave:

- `getMatriz()`: Devuelve la matriz filtrada para el algoritmo.
- `calcularMedia()`: Método polimórfico para hallar el centroide (usa media numérica o moda según el tipo).

Relaciones:

- **Asociación** con Perfil.
- **Asociación** con IEstrategiaAlgoritmo.
- **Asociación** con Respuesta.

2.1.8 Clase: Respuesta (Abstracta)

Definición: Representa la respuesta concreta de un individuo a una pregunta.

Atributos: Referencias a *Persona*, *Encuesta* y *Pregunta* para formar una identificación única.

Métodos clave:

- `getValor()`: Abstracto, devuelve el valor real (Object).
- `distance(Respuesta r)`: Abstracto, calcula la distancia entre dos respuestas. Fundamental para el clustering.

Relaciones:

- **Asociación** con *Cjt_Respuestas*.
- **Asociación** con *Encuesta*.
- **Asociación** con *Pregunta*.
- **Asociación** con *Persona*.

2.1.9 Clase: Numérica (Hereda de Respuesta)

Definición: Almacena cantidades numéricas (simples o múltiples).

Cálculo de Distancia:

- **Valor Único:** Distancia Euclidiana Normalizada.
- **Valor Múltiple:** $(1 - \text{Índice de Jaccard})$.

2.1.10 Clase: Textual (Hereda de Respuesta)

Definición: Almacena respuestas tipo *String*.

Atributos: Incluye `numeral` para manejar categorías ordenadas como pseudo-números.

Cálculo de Distancia: Utiliza lógica de conjuntos (Jaccard) para respuestas múltiples o categóricas no ordenadas, y distancia proporcional para ordenadas.

2.1.11 Clase: Perfil

Definición: Representa un cluster resultante, agrupando individuos similares.

Atributos: Lista de individuos (`lista_individuos`) y el identificador del representante/centroide (`representante`).

Relaciones:

- **Asociación** con *Cjt_Respuestas*.

2.2 Módulo de Clustering y Algoritmia

2.2.1 Clase: IEstrategiaAlgoritmo (Interfaz)

Definición: Define el contrato para algoritmos de clustering, permitiendo intercambiar implementaciones.

Relaciones:

- **Asociación** con *Cjt_Respuestas*.

2.2.2 Clase: Kmeans (Implementa IEstrategiaAlgoritmo)

Definición: Implementación del algoritmo K-means estándar.

Atributos:

- MAX_ITERATIONS (300): Límite de seguridad.
- CONVERGENCE_TOLERANCE (1e-4): Umbral de parada.
- estrategia (IEstrategiaIni): Estrategia de inicialización inyectada.

Métodos clave:

- iniciaAlgoritmo(): Ciclo principal (asignación ↔ actualización) hasta convergencia.
- asignaCentroide(): Asigna cada persona al cluster más cercano.
- recalculaCentroides(): Calcula nuevos prototipos usando la media/moda del grupo.
- evaluaCalidad(): Calcula el Coeficiente de *Silhouette*.

Relaciones:

- **Asociación** con IEstrategiaIni.

2.2.3 Clase: Kmedoides (Implementa IEstrategiaAlgoritmo)

Definición: Implementación del algoritmo K-Medoides. A diferencia de K-Means, los centros (*medoides*) son puntos existentes en el conjunto de datos, lo que aporta robustez frente a *outliers*.

Atributos:

- MAX_ITERATIONS (300): Límite de seguridad para evitar bucles infinitos.
- CONVERGENCE_TOLERANCE (0.0): Umbral estricto de parada (se detiene cuando los medoides dejan de cambiar).
- respuestas (Cjt_Respuestas): Referencia a la matriz de datos.

Métodos clave:

- iniciaAlgoritmo(): Controla el ciclo principal de asignación y actualización.
- inicializaMedoidesTrivial(): Inicialización interna que selecciona los primeros k puntos.
- recalculaMedoides(): Busca el nuevo centro calculando el punto que minimiza la suma de distancias dentro del cluster.
- evaluaCalidad(): Calcula el Coeficiente de *Silhouette* global.

2.2.4 Clase: ClusteringSub (Implementa IEstrategiaAlgoritmo)

Definición: Variante del algoritmo que utiliza obligatoriamente la inicialización K-means++.

Funcionalidad: Ejecuta el clustering y evalúa la calidad mediante el Coeficiente de *Silhouette* global.

Relaciones:

- **Asociación** con IniKmeansPP.

2.2.5 Clase: IEstrategiaIni (Interfaz)

Definición: Interfaz para definir la estrategia de selección de centroides iniciales, permitiendo variar entre métodos triviales o inteligentes.

Métodos: iniCentroides(int k, ArrayList<ArrayList<Respuesta>> centroides, HashMap<String, ArrayList<Respuesta>> matriz): Este método se encarga de inicializar la lista de centroides pero la implementación depende de las subclases.

- **Asociación** con `Kmeans`.

2.2.6 Clase: `IniKmeansPP` (Implementa `IEstrategiaIni`)

Definición: Estrategia inteligente (K-means++). Selecciona el primer centroide al azar y los siguientes con probabilidad proporcional al cuadrado de la distancia a los ya existentes.

Atributos: Usa `semilla` y `Random` para reproducibilidad.

2.2.7 Clase: `IniTrivial` (Implementa `IEstrategiaIni`)

Definición: Estrategia básica que selecciona los primeros k elementos de la matriz de datos como centroides iniciales.

Funcionalidades Extras y Algoritmia Avanzada

3.1 Módulo de Análisis de Datos (Clustering)

El sistema de análisis se estructura bajo el patrón de diseño Estrategia, definido por la interfaz `IEstrategiaAlgoritmo`. Esto permite que el controlador de dominio ejecute indistintamente diferentes algoritmos de agrupamiento llamando a un método común `iniciaAlgoritmo()` y recuperando los resultados mediante `getCentroides()` y `getInfoAsignaciones()`.

A continuación se detalla la lógica interna implementada por cada componente.

3.1.1 Algoritmo K-Means

La clase `Kmeans` implementa la versión estándar del algoritmo, trabajando con centroides calculados matemáticamente.

- **Flujo de Ejecución:** El algoritmo itera un máximo de **300 veces** (`MAX_ITERATIONS`) o hasta que la diferencia entre los centroides antiguos y los nuevos sea menor a **0.0001** (`CONVERGENCE_TOLERANCE`).
- **Asignación:** En cada iteración, cada respuesta de la matriz se asigna al centroide con el que tiene menor distancia promedio.
- **Recálculo de Centroides:** Para calcular el nuevo centro de un cluster, el algoritmo no se limita tan solo a medias numéricas. Delega en la clase `Cjt_Respuestas` mediante el método `calcularMedia`, el cual gestiona los tipos de datos:
 - Para respuestas **Numéricas**: Calcula la media aritmética estándar.
 - Para respuestas **Textuales (Única/Libre)**: Calcula la **Moda Simple** (el valor más frecuente).
 - Para respuestas **Múltiples**: Calcula la **Moda Múltiple** (el conjunto de opciones más frecuente).

3.1.2 Algoritmo K-Medoides

La clase `Kmedoides` implementa una variante donde el centro del cluster (*medoide*) debe ser obligatoriamente una respuesta existente en el conjunto de datos, no un valor inventado.

- **Inicialización:** A diferencia de K-Means, esta implementación utiliza internamente una inicialización trivial fija, seleccionando los primeros *k* elementos como medoides iniciales.
- **Criterio de Convergencia:** La tolerancia de convergencia es estricta (**0.0**), lo que significa que el algoritmo solo se detiene cuando los medoides dejan de cambiar completamente o se llega a las 300 iteraciones.
- **Recálculo del Medoide:** En el método `recalculaMedoides`, el algoritmo evalúa cada punto candidato dentro de un cluster. Suma las distancias de ese candidato contra todos los demás miembros del grupo. El punto que obtiene la **menor suma de distancias** se convierte en el nuevo medoide.

3.1.3 Estrategias de Inicialización

Para el algoritmo K-Means (y la variante `ClusteringSub`), el sistema permite inyectar diferentes estrategias de inicialización mediante la interfaz `IEstrategiaIni`.

1. **Inicialización Trivial (IniTrivial):** Simplemente recorre la matriz de respuestas y selecciona los primeros k elementos encontrados como centroides iniciales.
2. **Inicialización K-Means++ (IniKmeansPP):** Implementa una selección probabilística para dispersar de manera estocástica los centroides:
 - Usa una semilla fija (1234) para garantizar reproducibilidad en los juegos de prueba.
 - El primer centroide se elige aleatoriamente.
 - Los siguientes se eligen calculando la distancia cuadrada (D^2) de cada punto al centroide más cercano. La probabilidad de elegir un punto es proporcional a esta distancia, favoreciendo a los puntos más lejanos.

3.2 Funcionalidades Extras

En esta sección se detallan las funcionalidades adicionales implementadas para enriquecer la experiencia de usuario y el análisis del sistema. Estas extensiones incluyen mejoras algorítmicas, herramientas de visualización y gestión de sesiones (*login*).

3.2.1 Algoritmos de Optimización y Calidad

Para asistir al usuario en la toma de decisiones sobre el número de grupos (k) y validar la coherencia de los resultados, se han integrado dos métodos estadísticos avanzados.

Selección del número de clusters (k) maximizando el índice Silhouette:

Técnica heurística para determinar el número óptimo de clusters.

Funcionamiento:

- El sistema ejecuta el algoritmo de clustering iterativamente para un rango de valores de k .
- En cada iteración, se calcula el Coeficiente de "Silhouette" resultante.
- Finalmente, el sistema compara todos los coeficientes y selecciona automáticamente el valor de k que maximiza este indicador.

Coeficiente de 'Silhouette':

Métrica de validación interna utilizada tanto para el cálculo del k óptimo como para evaluar la calidad de cualquier ejecución manual. Para cada punto i , se calcula:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

Donde $a(i)$ es la distancia media intra-cluster y $b(i)$ es la distancia media al cluster vecino más cercano.

Interpretación en el Sistema:

- El sistema devuelve un valor global entre -1 y 1.
- Valores cercanos a 1 indican clusters bien definidos y separados.
- Valores cercanos a 0 indican clusters solapados.
- Valores negativos indican asignaciones erróneas.

3.2.2 Simulación Gráfica 2D

Dado que los datos de las encuestas son multidimensionales (cada pregunta añade una dimensión), su visualización directa es compleja. Se ha implementado un módulo de simulación gráfica en 2D

para ofrecer una representación visual intuitiva del proceso de clustering.

- **Proyección:** El sistema realiza una proyección de los puntos multidimensionales sobre un plano 2D, permitiendo visualizar la distribución espacial relativa de las respuestas.
- **Dinámica:** La gráfica muestra en tiempo real la posición de los centroides (o medoides) y cómo estos se desplazan en cada iteración del algoritmo hasta alcanzar la convergencia.
- **Codificación de color:** Cada punto se colorea según el cluster al que ha sido asignado, facilitando la identificación visual de los grupos y los outliers.

3.2.3 Algoritmo K-Medoides

Como alternativa robusta al K-Means, se ha añadido el algoritmo **K-Medoides**.

Diferencias clave con K-Means:

- **Centroides Reales:** En lugar de calcular un promedio matemático (que puede no corresponder a ninguna respuesta real), K-Medoides selecciona un individuo real del conjunto de datos como representante del cluster (el medoide).
- **Robustez al Ruido:** Es menos sensible a valores atípicos (*outliers*) que K-Means, ya que minimiza la suma de disimilaridades en lugar de la suma de distancias euclidianas cuadradas.
- **Aplicación:** Ideal para encuestas con muchas variables categóricas o cualitativas donde la "media" aritmética no tiene una interpretación directa.

3.2.4 Filtrado Selectivo de Preguntas

Para dotar de mayor flexibilidad al análisis, se ha implementado una funcionalidad que permite al usuario seleccionar manualmente qué preguntas no quiere incluir en el proceso de clustering. Esto es fundamental en escenarios reales donde ciertas respuestas podrían introducir ruido y distorsionar los grupos resultantes.

- **Selección Dinámica:** Antes de ejecutar el algoritmo, el usuario dispone de una interfaz para marcar o desmarcar las preguntas que desea ignorar en el clustering actual.
- **Procesamiento Interno:** El sistema (clase `Cjt_Respuestas`) filtra la matriz de datos en tiempo real, generando una proyección del espacio de respuestas que considera únicamente las dimensiones relevantes seleccionadas por el usuario.
- **Análisis Comparativo:** Esta característica permite realizar múltiples análisis sobre la misma encuesta enfocándose en diferentes aspectos (por ejemplo, agrupar usuarios solo por sus "Datos Demográficos" y luego solo por sus "Intereses" etc.) sin necesidad de modificar los datos originales.

3.2.5 Ayuda Interactiva y Manual de Usuario

Para mejorar la usabilidad del sistema y asistir a usuarios no expertos en conceptos estadísticos, se ha integrado un manual de ayuda visual dentro de la propia interfaz de creación de encuestas.

Manual para el Diseño de Encuestas

Guía rápida para elegir la configuración correcta según los datos que quieras recoger.

1. Tipo de Respuesta: Cuantitativa

¿Cuándo usarla? Cuando la respuesta esperada es un número real con valor matemático.

Ejemplos: Edad, Sueldo, Altura, Número de hijos.

- El sistema calculará la diferencia real (distancia) entre valores.
- (Por ejemplo: Tener 20 años se parece más a tener 25 que a tener 60).

2. Tipo de Respuesta: Cualitativa

Úsala cuando quieras que el usuario seleccione **una única opción** de una lista predefinida.

A. No Ordenada (Nominal)

¿Cuándo usarla? Cuando las opciones son distintas pero no tienen jerarquía.

Ejemplos:

- **Color favorito:** (Rojo, Verde, Azul).
- **Ciudad:** (Madrid, Barcelona, Valencia).

¿Cómo funciona? El sistema solo distingue igualdad o desigualdad. No calcula cercanía.

B. Ordenada (Ordinal)

¿Cuándo usarla? Cuando las opciones tienen un rango, nivel o intensidad lógica.

Ejemplos:

- **Nivel de satisfacción:** (Bajo, Medio, Alto).
- **Frecuencia:** (Nunca, A veces, Siempre).

¿Cómo funciona? El sistema entiende la distancia lógica ("Bajo" está más cerca de "Medio" que de "Alto").

3. Tipo de Pregunta: Texto Libre

¿Cuándo usarla? Cuando no hay opciones y el usuario escribe libremente.

Ejemplo: Nombre de pila, Comentarios breves.

¿Cómo funciona? El sistema busca similitud gramatical (Distancia de Levenshtein).

4. Tipo de Pregunta: Múltiple (Conjuntos)

Cuándo usarla? Cuando el usuario puede marcar **varias casillas** a la vez.

Ejemplo: ¿Qué aficiones tienes? (Deporte, Cine, Lectura...).

Cómo funciona? El sistema usa el Índice de Jaccard para agrupar personas con conjuntos de intereses parecidos.

5. Tipo de Pregunta: Única

Cuándo usarla? Cuando el usuario puede marcar **una sola casilla** a la vez.

Ejemplo: ¿Qué color te gusta más? (Rojo, Azul, Amarillo etc.).

¿Cómo funciona? El sistema mapea las opciones a una **escala numérica normalizada** (de 0 a 1) para calcular matemáticamente la distancia proporcional entre la opción elegida y las demás.

3.2.6 Sistema de Login y Persistencia de Sesión

Para soportar un entorno multi-usuario y garantizar la continuidad del trabajo, se ha implementado un sistema de autenticación y persistencia de estado.

- **Login Funcional:** Al iniciar la aplicación, el usuario debe identificarse. El sistema carga el perfil asociado a dichas credenciales.
- **Persistencia de Encuestas Activas:** El sistema guarda automáticamente el estado de las encuestas en curso asociadas a cada usuario. Esto permite:
 - Cerrar la aplicación y retomarla en el mismo punto.
 - Mantener separadas las respuestas y configuraciones de distintos usuarios en la misma máquina.
 - Evitar la pérdida de datos ante cierres inesperados.

Pruebas y Ejecución del Sistema

Esta sección describe los controladores (drivers) implementados para la ejecución del sistema, los tests de integración diseñados para validar la lógica del dominio y el manual de uso del script de ejecución.

4.1 Drivers del Sistema

Se han implementado dos drivers principales para interactuar con la capa de dominio sin necesidad de la interfaz gráfica completa.

Driver Visual (DriverVisual.java):

Este driver proporciona una interfaz de consola interactiva. Permite al usuario crear encuestas, responderlas y ejecutar el clustering mediante menús.

```
Menú
  1- Crear Encuesta de Prueba
  2- Responder Encuesta Manualmente
  3- Ejecutar Algoritmo (K-Means)
  4- Visualizar Informe de Resultados
  5- Salir
....
Seleccione una opción: 1
Encuesta creada con ID: 1
Se han configurado 3 preguntas...

Seleccione una opción: 3
Introduce el número de clusters deseado (k): 2
Ejecutando K-Means.....
Clustering finalizado.
```

Driver Ejemplo Completo (DriverEjemploCompleto.java):

Driver autoejecutable que simula un flujo completo. Crea una encuesta, genera usuarios simulados, prueba la persistencia y ejecuta el algoritmo automáticamente.

```
DRIVER AUTOMÁTICO: DEMOSTRACIÓN DE INTEGRACIÓN

Paso 1: Creando Encuesta 'Estilo de Vida'...
Encuesta creada con ID: 1

Paso 2: Generando 4 usuarios simulados...
- Usuarios registrados: A y B (Jóvenes), C y D (Mayores).

Paso 3: Probando Persistencia...
- Datos recuperados del disco correctamente.

Paso 4: Ejecutando K-Means (k = 2) Ignorando 'Ciudad'...

Paso 5: Resultados Obtenidos:
Cluster 1: [UserC, UserD]
Cluster 2: [UserA, UserB]
Calidad del Clustering 0.8521
Resultado: ÉXITO
```

Driver Avanzado Comparativo (DriverComparativa.java):

Este driver valida las funcionalidades avanzadas (Extras). Simula una encuesta compleja con datos mixtos para probar el cálculo automático de k y realizar una comparativa de rendimiento entre K-Means y K-Medoides.

```
DRIVER COMPARATIVO: BÚSQUEDA DE K ÓPTIMA Y COMPARACIÓN K-MEANS VS K-Medoides
```

```
[PASO 1] Inicializando Entorno y Login...
```

```
PASO 2
```

```
Creando Encuesta 'Perfil Tecnológico'...
```

```
->Encuesta creada con ID: 1
```

```
[PASO 3] Inyectando datos simulados (3 Perfiles Claros)...
```

```
->8 Usuarios registrados en 3 grupos teóricos.
```

```
[PASO 4] Calculando K Óptima...
```

```
->El sistema sugiere que la k óptima es: 5
```

```
[PASO 5] Batalla de Algoritmos con k = 5
```

```
-----
```

```
Ejecutando K-MEANS... Hecho. Calidad (Silhouette): 0,6724
```

```
Cluster 1: [Jefe_1] [Rep: Boss_1]
```

```
Cluster 2: [Data_Jr_2, Data_Jr_3] [Rep: Data_Jr_2]
```

```
Cluster 3: [Dev_Sr_1, Dev_Sr_3, Dev_Sr_2] [Rep: Dev_Sr_3]
```

```
Cluster 4: [Jefe_2] [Rep: Boss_2]
```

```
Cluster 5: [Data_Jr_1] [Rep: Data_Jr_1]
```

```
-----
```

```
Ejecutando K-MEDOIDES... Hecho. Calidad (Silhouette): 0,8049
```

```
Cluster 1: [Boss_2, Boss_1] [Rep: Boss_2]
```

```
Cluster 2: [Data_Jr_2] [Rep: Data_Jr_2]
```

```
Cluster 3: [Data_Jr_3] [Rep: Data_Jr_3]
```

```
Cluster 4: [Dev_Sr_1, Dev_Sr_3, Dev_Sr_2] [Rep: Dev_Sr_1]
```

```
Cluster 5: [Data_Jr_1] [Rep: Data_Jr_1]
```

```
=====
```

```
CONCLUSIÓN FINAL
```

```
=====
```

```
K-Means: 0,6724
```

```
K-Medoides: 0,8049
```

```
-----
```

```
GANADOR: K-MEDOIDES (Mejor adaptación a variables mixtas/outliers)
```

4.2 Tests de Integración

Tests JUnit diseñados para verificar la corrección del sistema en escenarios complejos.

4.2.1 Test: IntegracionTest_Edades

Descripción: Verifica la agrupación basada en una sola variable numérica.

Escenario: 3 usuarios jóvenes (18-27) y 3 mayores (67-81).

Objetivo: Validar que el algoritmo separa perfectamente los dos grupos de edad.

```

ENCUESTA
ID_Pregunta,Enunciado,Tipo_Pregunta...
1,Edad,1,,

RESPUESTAS
Persona_ID,Edad
A,18.0
B,27.0
C,21.0
D,78.0
E,81.0
F,67.0

EXPECTATIVA DEL TEST: se esperan dos clusters (k = 2)
=== Clusters Esperados ===
Expected 1: [A, B, C]
Expected 2: [D, E, F]
=== Clusters Calculados ===
Actual 1: [A, B, C]
Actual 2: [D, E, F]

Verificación: todos los clusters esperados encontrados - TEST PASADO

```

4.2.2 Test: IntegracionTest_Visual

Descripción: Test exhaustivo con datos heterogéneos.

Escenario: Grupo A (Jóvenes/Madrid/Deporte) vs Grupo B (Adultos/BCN/Cultura).

Objetivo: Verificar métricas combinadas (Euclídea + Jaccard).

```

INICIO DEL TEST DE INTEGRACIÓN:
Paso 1: Configuración de la encuesta (ID: 1)
  1  Edad      (Libre)
  2  Ciudad    (Única)
  3  Hobbies   (Multiple)

Paso 2: Generación de perfiles (Datos inyectados)
PERSONA EDAD CIUDAD HOBBIES
A1 20.0 Madrid [Futbol, Tennis]
B1 40.0 Barcelona [Pintura, Leer]

Paso 3: Ejecución K-means (k = 2)
  - Algoritmo finalizado sin errores.

Paso 4: Verificación de resultados
CLUSTER 1 OBTENIDO: [A1, A2, A3] -> CORRECTO (Grupo A)
CLUSTER 2 OBTENIDO: [B2, B3, B1] -> CORRECTO (Grupo B)

Paso 5: Resumen assertions JUnit
[OK] TEST 1: Número de clusters es 2
[OK] TEST 2: Grupo A encontrado completo
[OK] TEST 3: Grupo B encontrado completo
[OK] TEST 4: Calidad del Clustering (0,7432) > 0.5
Resultado: ÉXITO

```


4.2.3 Test: IntegracionTest_Persistencia

Descripción: Evalúa la capa de persistencia y su integración con el dominio.

Escenario: Guardado en disco → Reinicio del controlador → Carga.

Objetivo: Asegurar la integridad de los datos tras el almacenamiento.

```
TEST DE PERSISTENCIA: Guardar - Reiniciar - Cargar - Ejecutar
Fase 1: Generando datos y guardando en 'Datos/'...
Guardando encuesta ID: 1 a CSV...
Guardando respuestas a CSV...
Fase 2: Simulando reinicio del sistema (Nuevo Controlador)
Controlador nuevo esta vacío? Si
Fase 3: Importando desde CSV y Ejecutando
Importando encuesta ID: 1...
Personas recuperadas del disco: 2
Ejecutando K-Means (2 Clusters)...
Clustering ejecutado con éxito sobre datos persistidos.
Fase 4: Exportando Perfiles Resultantes
Verificando fichero de perfiles: Fichero creado en Datos/Perfiles/1.csv
TEST COMPLETADO CORRECTAMENTE
```

4.2.4 Test: IntegracionTest_Frontera

Descripción: Análisis de comportamiento límite.

Escenario: Polos fijos [0.0, 0.1] y [0.9, 1.0]. Se introduce un valor variable (0.4, 0.5, 0.6) para verificar el salto de cluster.

Objetivo: Verificar la sensibilidad del algoritmo K-Means ante pequeños cambios en valores intermedios.

```
INICIO TEST DE FRONTERA VISUAL (Valores como IDs)
=====
Extremos fijos: [0.0, 0.1] y [0.9, 1.0]
=====
ESCENARIO: Añadimos valor 0,5
->Calculando K-Means (k = 2)... Hecho.
->Calidad del Clustering: 0,6244
->OBSERVACIÓN (0.5): Se agrupó con los ALTOS.
ESCENARIO: Añadimos valor 0,4
->Calculando K-Means (k = 2)... Hecho.
->Calidad del Clustering: 0,4278
->VERIFICACIÓN (0.4): CORRECTO. 0.4 está en el grupo bajo [0.0, 0.1, 0.4]
ESCENARIO: Añadimos valor 0,6
->Calculando K-Means (k = 2)... Hecho.
->Calidad del Clustering: 0,4257
->VERIFICACIÓN (0.6): CORRECTO. 0.6 está en el grupo alto [1.0, 0.6, 0.9]
TEST FINALIZADO
```

4.3 Plan de Verificación Funcional y Pruebas de Robustez (GUI)

Para garantizar la integridad del sistema a través de la interfaz gráfica, se ha ejecutado un plan de verificación exhaustivo. Estas pruebas validan tanto el flujo de trabajo estándar (gestión de encuestas y usuarios) como la robustez del sistema ante situaciones de error o límites.

A continuación se detallan los resultados obtenidos:

ID	Escenario de Prueba	Comportamiento del Sistema	Estado
TS-01	Persistencia de Datos por Usuario El Usuario 1 tiene las encuestas {1,2,3}. Se cierra la aplicación completamente y se vuelve a iniciar sesión con Usuario 1.	Restauración Correcta: El sistema carga el perfil y la Vista Principal muestra exactamente las encuestas {1,2,3} sin pérdida de datos.	OK
TS-02	Feedback de Login (Seguridad) a) Introducir usuario inexistente. b) Introducir usuario real con contraseña mal.	a) Mensaje: "El usuario no existe". b) Mensaje: "Contraseña incorrecta". El sistema distingue el error para informar correctamente.	OK
TS-03	Edición de Encuesta Abrir una encuesta existente, modificar el enunciado de una pregunta y guardar cambios.	El archivo se sobrescribe correctamente. Al volver a abrir la encuesta, los cambios se reflejan en la interfaz.	OK
TS-04	Configuración Algoritmos <i>Algoritmo:</i> K-Means, K-Medoides, ClusteringSub. <i>Inicialización:</i> Trivial, K-MEANSPP.	Efecto: Ejecución visual y generación de resultados (Clusters/Silhouette).	OK
TS-05	Eliminación de Encuesta Seleccionar una encuesta en la Vista Principal y pulsar el botón "Papelera".	La encuesta desaparece de la lista inmediatamente y el archivo asociado se desvincula del usuario.	OK
TS-06	Validación de Tipos (Input) Intentar responder una pregunta numérica introduciendo texto (por ejemplo: "hola").	Bloqueo: La interfaz impide la escritura hasta que el dato sea numérico válido.	OK
TS-07	Clustering sin Datos Intentar calcular K óptimo o analizar una encuesta que tiene 0 respuestas.	Control de Excepción: Alerta "No hay suficientes respuestas para analizar". El sistema no se cierra inesperadamente.	OK
TS-08	Importación Inexistente Intentar importar un archivo .csv indicando un ID que no existe en disco.	Mensaje de error controlado: "Archivo no encontrado".	OK
TS-09	Cálculo de K Óptimo a) Ejecutar con datos válidos. b) Ejecutar sin respuestas cargadas.	a) Muestra la gráfica y sugiere el valor de K . b) Muestra alerta: "No hay respuestas suficientes para calcular".	OK
TS-10	Filtrado Preguntas <i>Valores:</i> Pregunta desmarcada (por ejemplo: "Edad"). <i>Objetivo:</i> Exclusión de preguntas específicas para el cálculo del clustering.	Efecto: Resultados generados ignorando esa columna de datos.	OK

Cuadro 1: Resultados de las pruebas funcionales y de robustez (Caja Negra)

4.4 Manual de Ejecución

Para facilitar la ejecución del proyecto, los tests y la interfaz gráfica, se proporciona el script `Ejecutar_script.sh`.

Instrucciones de uso

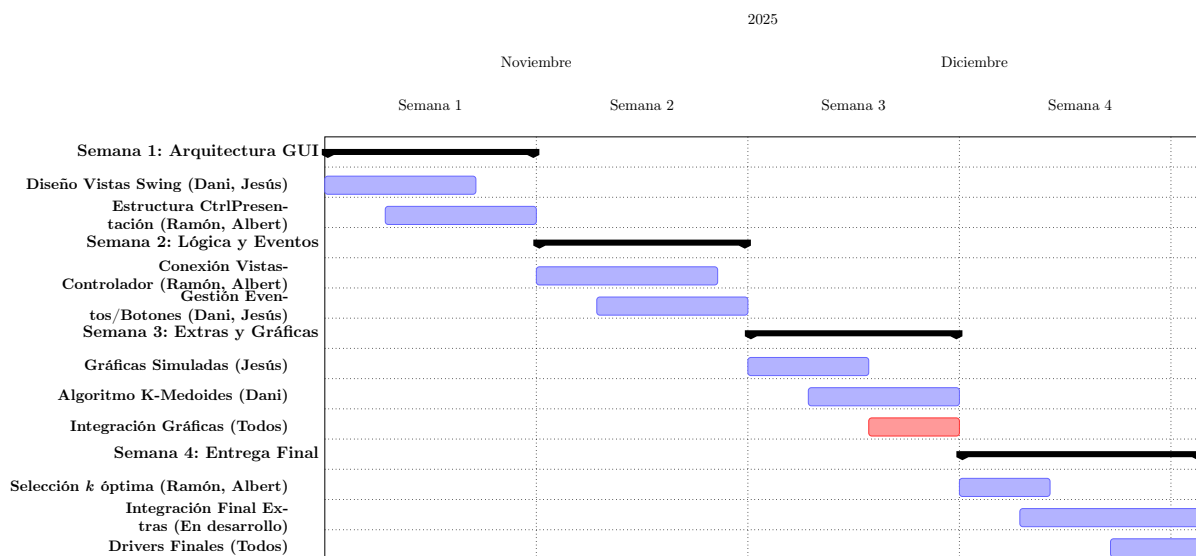
1. Abra una terminal en el directorio raíz del proyecto.
 2. Ejecute el comando: `./Ejecutar_script.sh`
 3. Aparecerá un menú interactivo con las siguientes opciones:
- **Opción 0) Ejecutar Interfaz Gráfica:** Compila el proyecto y lanza la aplicación principal con GUI.
 - **Opción 1) Ejecutar Driver AUTOMÁTICO-1:** Ejecuta el `DriverEjemploCompleto` para ver una demostración rápida por consola.
 - **Opción 2) Ejecutar Driver AUTOMÁTICO-2:** Ejecuta el `DriverComparativa` para ver una demostración rápida por consola.
 - **Opción 3) Ejecutar Driver INTERACTIVO:** Lanza el `DriverVisual`, permitiendo al usuario interactuar manualmente con el sistema desde la terminal.
 - **Opción 4) Ejecutar TESTS (Reporte web):** Ejecuta todos los tests unitarios y de integración mediante Gradle. Al finalizar, intenta abrir automáticamente el reporte HTML de resultados en el navegador predeterminado. En la sección *output* en cada test se observan los resultados obtenidos.
 - **Opción 5) Salir:** Cierra el menú.

Planificación y Gestión del Proyecto

En esta sección se detalla el cronograma de trabajo seguido por el equipo durante la segunda fase del proyecto (17 de noviembre - 15 de diciembre de 2025), enfocada en la Capa de Presentación y las funcionalidades extras.

5.1 Cronograma de Trabajo (Diagrama de *Gantt*)

La planificación se ha dividido en cuatro semanas intensivas para desarrollar la Interfaz Gráfica (GUI) y los algoritmos extra (K-Medoides y selección óptima de k), integrándolos con el Dominio ya existente.



5.2 Gestión de Tareas y Roles

- **Dani del Rio y Jesús Peñaranda:** Encargados principales de la **Visualización y Algoritmia Avanzada**. Diseñaron las vistas en Swing y desarrollaron la lógica matemática para las extensiones.
- **Ramón García y Albert Porta:** Responsables de la **Arquitectura de Control**. Implementaron el `CtrlPresentacion.java` para asegurar que las acciones del usuario en la interfaz se comunicaran correctamente con el dominio ya existente.

5.3 Dificultades y Estado de las Ampliaciones

Durante esta fase de desarrollo centrada en la interfaz y las funcionalidades extra, surgieron complicaciones técnicas que afectaron la integración final:

1. **Validación de Datos y Tipado Dinámico:** Uno de los principales desafíos técnicos surgió al manejar la entrada de datos por parte del usuario. Inicialmente, el diseño del dominio asumía que los valores numéricos llegarían siempre limpios y en el formato correcto. Sin embargo, al conectar la capa de presentación, nos encontramos con que los campos de texto (TextFields) devuelven cadenas de caracteres (*String*), lo que provocaba errores de ejecución (`NumberFormatException`) si el usuario introducía texto en preguntas numéricas.

o dejaba espacios en blanco. Esto obligó a implementar una capa intermedia de validación y parsing robusta en el controlador de presentación para asegurar que los datos fuesen convertidos y saneados correctamente antes de ser procesados por la lógica del dominio.

2. **Complejidad en la Integración Global con la Interfaz:** La integración de los distintos módulos (gestión de encuestas, algoritmo K-Means y persistencia) con la Interfaz Gráfica de Usuario (GUI) presentó problemas no previstos durante las pruebas unitarias. Al unificar todo el sistema, surgieron conflictos en el flujo de datos entre las vistas y el controlador, especialmente al intentar actualizar la visualización de los resultados en tiempo real tras la ejecución del algoritmo. Además, al añadir nuevas funcionalidades surgen actualizaciones necesarias de tests anteriores.

5.4 Asignación Clases

Clase / Fichero	Desarrollador(es) Asignado(s)
Cjt_Respuestas.java	Jesús Peñaranda
ClusteringSub.java	Dani del Rio
Encuesta.java	Albert Porta
Kmeans.java	Dani del Rio
IEstrategiaAlgoritmo.java	Dani del Rio, Jesús Peñaranda
IEstrategiaIni.java	Dani del Rio, Jesús Peñaranda
IniTrivial.java	Dani del Rio, Albert Porta
Kmedoides.java	Dani del Rio, Jesús Peñaranda
IniKmeansPP.java	Dani del Rio
Libre.java	Ramón García
Multiple.java	Ramón García
Numerica.java	Jesús Peñaranda
Persona.java	Albert Porta
Pregunta.java	Ramón García
Respuesta.java	Jesús Peñaranda, Albert Porta
Textual.java	Jesús Peñaranda
Unica.java	Ramón García
Perfil.java	Albert Porta

Arquitectura de tres capas

6.1 Capa de Dominio

La Capa de Dominio encapsula toda la lógica matemática del programa, los algoritmos y la gestión de datos del sistema, manteniéndose totalmente independiente de la interfaz gráfica. Su punto de acceso principal es la clase `ctrlDominio`.

6.1.1 Controlador de Dominio (`ctrlDominio`)

Esta clase actúa como una "fachada" del dominio. Su responsabilidad es recibir las peticiones de la Capa de Presentación, para ejecutar la lógica del programa necesaria y coordinar la persistencia de los datos.

Sus funciones principales se dividen en cuatro bloques:

Gestión de Sesión y Usuarios:

Gestiona el ciclo de vida de los usuarios mediante los métodos `login` y `registrar`. Mantiene en memoria (`mapaUsuarios` y `mapaCredenciales`) la información de acceso y la lista de encuestas asociadas al usuario logueado, asegurando que cada usuario solo acceda a su propia información.

Gestión de Encuestas:

Permite crear, importar, editar y eliminar encuestas. Destaca el uso de una clase interna `DatosPregunta` para recibir datos crudos de la vista y convertirlos en objetos complejos (`Pregunta`, `Libre`, `Multiple`, etc.) dentro del sistema.

Lógica de Respuesta y Validación:

Es responsable de interpretar las respuestas de los usuarios. El método `responderEncuesta` contiene la lógica para distinguir si una respuesta es numérica o textual y si es simple o múltiple, instanciando los objetos correspondientes y validando la coherencia con la encuesta.

Algoritmos de Análisis (Clustering):

Es el núcleo computacional del proyecto. Coordina la ejecución de algoritmos de agrupamiento (`K-Means`, `K-Medoides`).

- **Pre-procesamiento:** Normaliza los valores y calcula mínimos/máximos (`analizarEncuesta`).
- **Cálculo del K-Óptimo:** Implementa una búsqueda para maximizar la calidad del clustering (*Silhouette*) ejecutando iterativamente los algoritmos para sugerir al usuario el mejor número de clusters (k).

6.2 Capa de Presentación

La Capa de Presentación es la encargada de la interacción con el usuario. Sigue el patrón *MVC* (Modelo-Vista-Controlador) donde las "Vistas" son los paneles gráficos y el `ctrlPresentacion` actúa como intermediario.

6.2.1 Controlador de Presentación (`ctrlPresentacion`)

Este controlador orquesta la navegación y el flujo de la aplicación. Inicializa el marco principal (`JFrame`) y gestiona las diferentes pantallas.

Gestión de Vistas (`CardLayout`):

Utilizamos un diseño de tarjetas (`CardLayout`) para intercambiar dinámicamente entre las

distintas pantallas (Login, Principal, Creación, etc.) sin cerrar la ventana principal. Esto centraliza la navegación en el método `mostrarVista`.

Comunicación con Dominio:

Cualquier acción del usuario (como "Guardar Encuesta" o "Analizar") se captura aquí y se reenvía al `ctrlDominio`. Esto mantiene a las vistas de manera independiente, encargándose solo de mostrar datos y capturar eventos.

Inicialización:

Al arrancar, configura el sistema y carga la vista de Login por defecto.

6.2.2 Vistas del Sistema

Las vistas están organizadas por funcionalidad y son gestionadas por el controlador:

- **VistaLogin:** Punto de entrada para autenticación y registro.
- **VistaPrincipal:** Panel de control ("Vista Principal") donde se listan las encuestas del usuario.
- **VistaCreacion / VistaEdicion:** Formularios dinámicos para diseñar las preguntas de la encuesta.
- **VistaResponder:** Interfaz para que los usuarios contesten las preguntas.
- **VistaAnálisis:** Pantalla de configuración para ejecutar los algoritmos de clustering (selección de k , algoritmo, etc.).
- **VistaInfoPerfiles:** Visualización de los resultados, mostrando los clusters generados, representantes y calidades.

6.3 Capa de Persistencia

La Capa de Persistencia es la responsable de almacenar y recuperar los datos del sistema de forma permanente, asegurando que la información sobreviva cuando se cierra la aplicación. Su diseño aísla completamente a la Capa de Dominio de los detalles de bajo nivel sobre el manejo de archivos.

6.3.1 Controlador de Persistencia (`ctrlPersistencia`)

Esta clase actúa como el único punto de entrada a la capa de datos. Su función principal es ofrecer una interfaz limpia y sencilla al Dominio para las operaciones de Guardado y Carga.

Implementa un patrón de diseño que delega la complejidad en clases especializadas llamadas "Gestores". El controlador no escribe los archivos directamente, sino que redirige la petición al gestor adecuado según el tipo de dato:

- **Gestión de Encuestas:** Utiliza `GestorEncuestas.java` para guardar la definición estructural de las encuestas (preguntas, opciones, enunciados).
- **Gestión de Respuestas:** Delega en `GestorRespuestas.java` el almacenamiento de las contestaciones de los usuarios, vinculándolas a su encuesta correspondiente.
- **Gestión de Perfiles :** Utiliza `GestorPerfiles.java` para persistir los resultados de los análisis de clustering (los grupos formados y sus representantes).

- **Gestión de Preguntas :** Utiliza `GestorPreguntas.java` para la importación de una o varias preguntas en una encuesta determinada.

6.3.2 Estrategia de Almacenamiento

Aunque el controlador oculta los detalles, el sistema está diseñado para manejar la persistencia mediante archivos (*.csv*, *.txt*). Esto permite una gestión donde cada clase (*Encuesta*, *Respuesta*, *Perfil*, *Pregunta*) se gestiona de manera independiente, facilitando la escalabilidad y el mantenimiento del código sin afectar a la lógica o cálculo matemático.

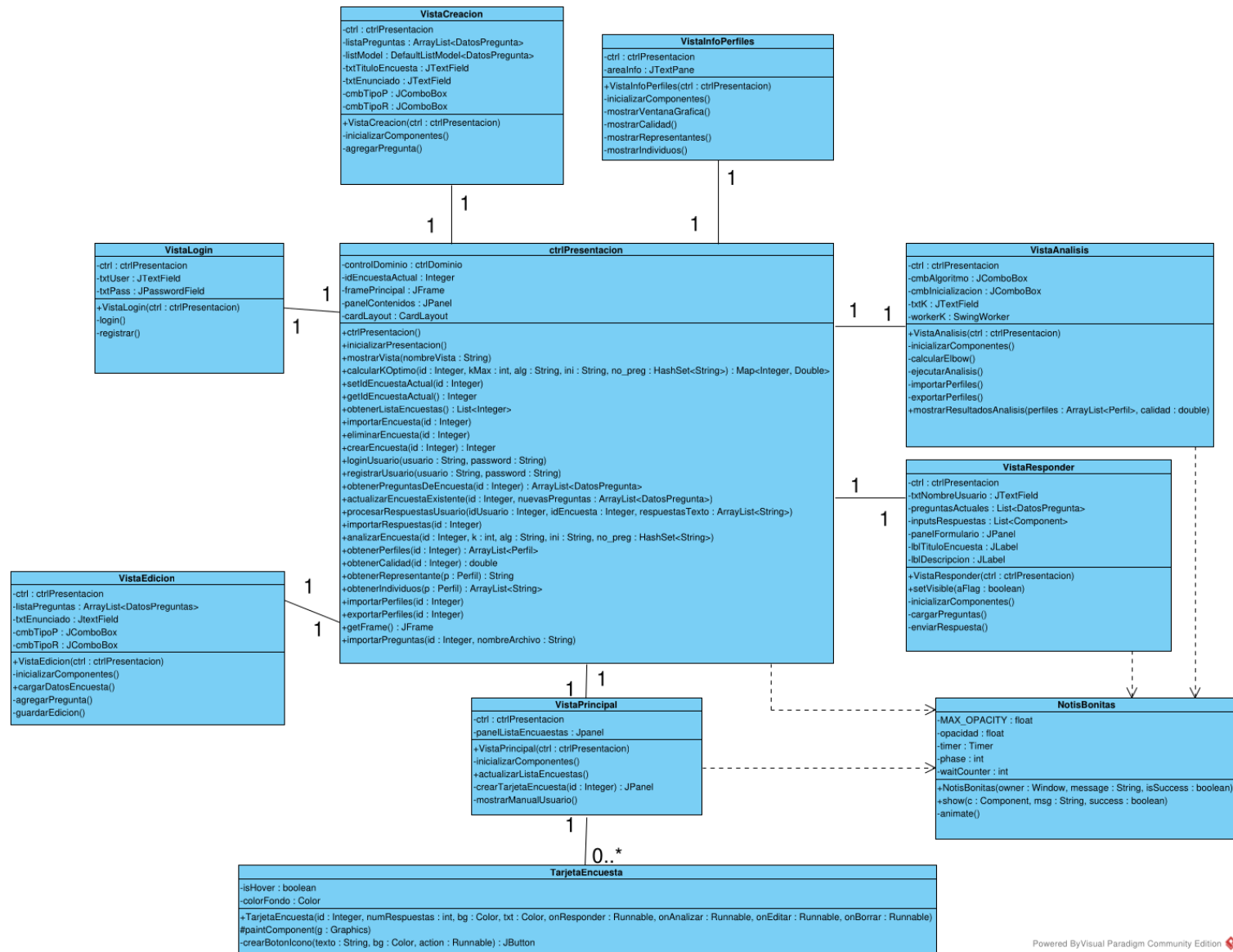


Figura 3: Diagrama de Clases Detallado: Capa de Presentación

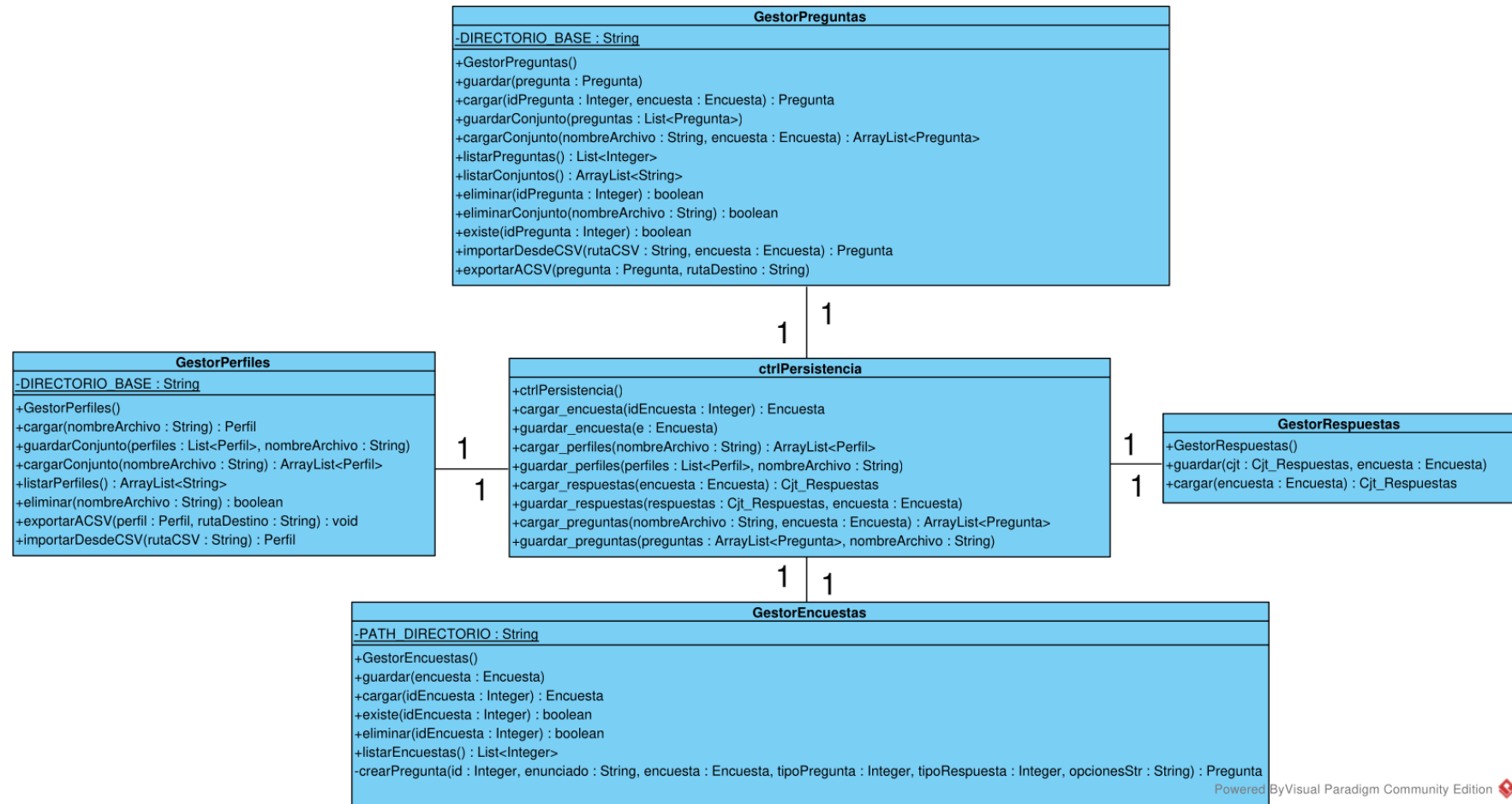


Figura 5: Diagrama de Clases Detallado: Capa de Persistencia

Interfaz gráfica de usuario (GUI)

La interfaz gráfica del sistema ha sido desarrollada utilizando la biblioteca **Java Swing**, siguiendo estrictamente el patrón de arquitectura de tres capas. El objetivo principal de esta parte es proporcionar una capa de abstracción visual que permita al usuario interactuar con la lógica de dominio de forma intuitiva, ocultando la complejidad de los algoritmos internos.

El diseño se ha estructurado como una aplicación de ventana única gestionada mediante un **CardLayout**. Este enfoque permite una navegación fluida entre las distintas secciones (Login, Vista Principal, Edición y Análisis) sin la necesidad de abrir múltiples ventanas emergentes, mejorando así la experiencia de usuario.

La interfaz se divide funcionalmente en cuatro bloques principales que se detallan visualmente a continuación:

1. **Control de Acceso:** Gestión de autenticación y registro de usuarios para garantizar la seguridad y persistencia de los datos.
2. **Gestión de Encuestas (Vista Principal):** Panel centralizado para la administración de las encuestas disponibles.
3. **Diseño y Recogida de Datos:** Entornos dinámicos para la creación de preguntas (con validación de tipos en tiempo real) y la admisión de respuestas.
4. **Análisis y Visualización:** Sección avanzada que permite configurar los parámetros de los algoritmos de clustering y visualizar gráficamente los perfiles resultantes.

7.1 Manual de Usuario y Guía de Uso

A continuación se detalla el funcionamiento de la interfaz gráfica, paso a paso, cubriendo desde el acceso hasta el análisis avanzado.

1 Inicio de la Ejecución y Login

Al abrir la aplicación, se presenta la vista de bienvenida. Este paso es fundamental para asociar las encuestas a un propietario y agilizar el relleno de datos.

- **Iniciar Sesión:** Introduzca sus credenciales para cargar sus encuestas.
- **Crear Cuenta:** Opción para nuevos usuarios.

Beneficio: Al responder encuestas, el sistema rellenará automáticamente su nombre de usuario, ahorrando tiempo al usuario.

2 Menú Principal (Vista Principal)

El panel de control se divide en dos áreas claras:

1. Barra Superior de Acciones:

- **Crear:** Permite diseñar una encuesta nueva.
- **Importar:** Carga encuestas externas desde la carpeta /Encuestas.
- **Ayuda:** Abre un *pop-up* con la guía de tipos de preguntas.
- **Salir:** Cierra la aplicación solicitando confirmación.

2. Área de Encuestas:

Muestra sus encuestas activas con cuatro botones de acción rápida:

- **Responder:** Para introducir datos.
- **Analizar:** Para ejecutar el clustering.
- **Editar:** Para modificar preguntas.
- **Eliminar (papelera):** Borra la encuesta definitivamente.

3 Creación de Encuestas

El diseñador de encuestas se divide en izquierda (lista actual) y derecha (formulario).

1. **Título:** Asigne un nombre descriptivo (por ejemplo: "Estudio de Mercado").
2. **Configurar Pregunta:** Escriba el enunciado y seleccione:
 - *Tipo:* Libre, Múltiple o Única.
 - *Respuesta:* Cuantitativa (Numérica) o Cualitativa (Ordenada/No Ordenada).
3. **Importar Pregunta:** Escriba el nombre del fichero *.csv* en la que guarda la pregunta/s que quiere importar el usuario.
4. **Opciones:** Para preguntas cerradas, separe las opciones con punto y coma (por ejemplo: Rojo;Verde;Azul).

Al finalizar, pulse **Guardar Encuesta** para almacenarla en Datos/Encuestas/encuesta_X.csv.

4 Importación de Encuestas

Si dispone de un fichero `encuesta_X.csv` en la carpeta de datos, puede cargarlo pulsando "Importar". El sistema solicitará el *ID* (número X) y, si es válido, la encuesta aparecerá inmediatamente en su lista principal lista para ser usada.

5 Responder Encuesta

Interfaz para la recogida de datos.

- **Identificación:** El campo "Nombre" aparece pre-rellenado con su usuario.
- **Enviar Respuestas:** Guarda las respuestas del formulario en el archivo `Datos/Respuestas/respuestas_encuesta_X.csv`.
- **Importar Respuestas:** Permite cargar masivamente un fichero de respuestas externo a la memoria.
- **Cancelar:** Aborta la operación y vuelve al menú principal sin guardar cambios.

6 Análisis: Configuración y Ejecución

En la pantalla de configuración, defina los parámetros del algoritmo antes de procesar los datos.

1. **Parámetros:** Seleccione el *Algoritmo* (K-MEANS, K-MEDOIDES, CLUSTERING-SUB) y la *Inicialización* (*K-MEANSPP*, *TRIVIAL*) deseada.
2. **Número de Clusters (K):**
 - **Calcular K Óptimo:** Si tiene dudas, use este botón. El sistema aplicará una búsqueda del número de clusters óptimos (k) y rellenará el campo k automáticamente con el valor sugerido.
3. **Gestión de Perfiles:**
 - **Importar Perfiles:** Carga una configuración guardada.
 - **Exportar Perfiles:** Guarda la configuración actual.
4. **Lanzamiento:** Pulse **Ejecutar Análisis** para procesar los datos y abrir el la vista de resultados.

7 Análisis: Visualización de Resultados

Tras ejecutar el análisis, se abre una nueva ventana con un menú lateral para navegar por los hallazgos:

- **Ver Calidad:** Muestra el índice de *Silhouette* global (valor numérico) para evaluar la cohesión de los grupos.
- **Representantes:** Visualiza los usuarios "prototipos" más cercanos al centroide (la respuesta promedio del grupo) de cada cluster.
- **Individuos:** Muestra la lista detallada de usuarios asignados a cada grupo.
- **Ver Gráfica Kmeans:** Abre la proyección 2D simulada para ver la distribución espacial de los clusters.
- **Exportar CSV:** Descarga los resultados detallados en un archivo *.csv*.
- **Volver:** Regresa a la pantalla de configuración anterior.

8 Editar Encuesta

Permite corregir o ampliar una encuesta existente desde el menú principal.

- **Borrar Preguntas:** Seleccione una pregunta de la lista izquierda y pulse "Borrar Seleccionada".
- **Añadir Preguntas:** Use el formulario derecho para agregar nuevas preguntas al final de la encuesta.

Pulse "Guardar Cambios" para sobrescribir el archivo original o "Cancelar" para descartar las modificaciones.

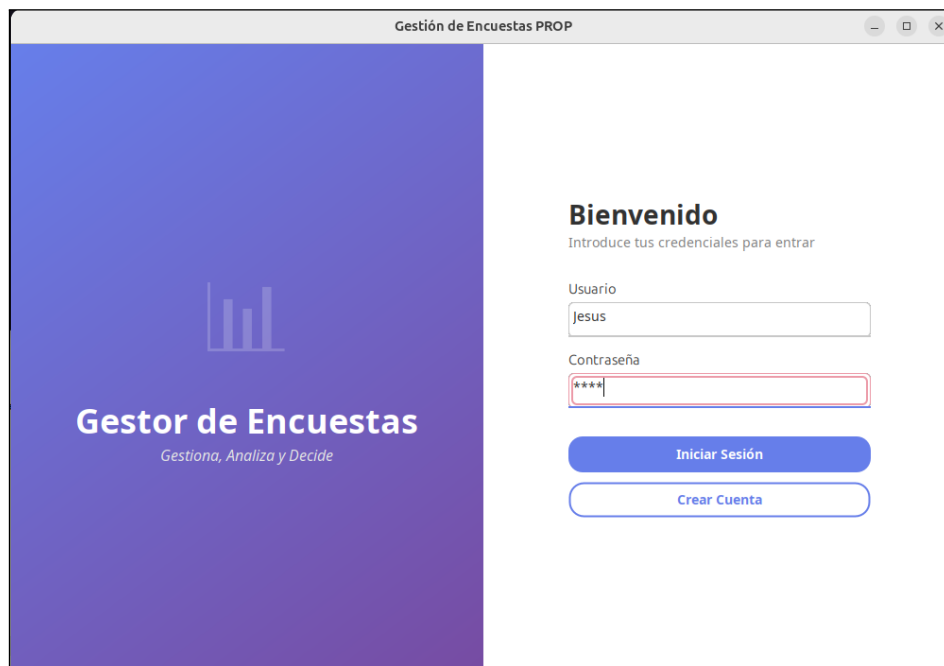


Figura 6: Vista de Login



Figura 7: Vista Principal

Gestión de Encuestas PROP

Crear Nueva Encuesta

Título de la Encuesta:

Preguntas Añadidas

Añadir Pregunta

Enunciado:

Tipo Pregunta:

Libre

Tipo Respuesta:

Cuantitativa

Añadir Pregunta

Importar Preguntas

Cancelar GUARDAR ENCUESTA

Figura 8: Vista Creación de Encuesta (vacía)

Gestión de Encuestas PROP

Crear Nueva Encuesta

Título de la Encuesta:

Preguntas Añadidas

Añadir Pregunta

Enunciado:

Tipo Pregunta:

Libre

Tipo Respuesta:

Cuantitativa

Añadir Pregunta

Importar Preguntas

Cancelar GUARDAR ENCUESTA

1. ¿Cuál es tu edad?
Tipo: 1
2. ¿Cuál es tu género?
Tipo: 3
3. ¿Qué deportes practicas?
Tipo: 2
4. ¿Cuántas horas duermes al día?
Tipo: 1
5. ¿Cómo calificarías tu nivel de inglés?
Tipo: 3
6. ¿Qué redes sociales usas?
Tipo: 2
7. ¿Cuál es tu color favorito?
Tipo: 3
8. ¿Cuántos libros lees al mes?
Tipo: 1

Figura 9: Encuesta creada

Gestión de Encuestas PROP

Encuesta #2

Completa las 8 preguntas para enviar tus respuestas

Tu Nombre:

1/8 Pregunta 1

¿Cuál es tu edad?

2/8 Pregunta 2

¿Cuál es tu género?

☐ Masculino
☐ Femenino
☐ Otro
☐ Prefiero no decir

3/8 Pregunta 3

¿Qué deportes practicas?

Figura 10: Vista para Responder Encuestas

Gestión de Encuestas PROP

Editando Encuesta #2

Preguntas Actuales

1. ¿Cuál es tu edad?
Tipo: 1
2. ¿Cuál es tu género?
Tipo: 3
3. ¿Qué deportes practicas?
Tipo: 2
4. ¿Cuántas horas duermes al día?
Tipo: 1
5. ¿Cómo calificarías tu nivel de inglés?
Tipo: 3
6. ¿Qué redes sociales usas?
Tipo: 2
7. ¿Cuál es tu color favorito?
Tipo: 3
8. ¿Cuántos libros lees al mes?
Tipo: 1

Añadir / Modificar

Enunciado:

Tipo:

Respuesta:

Figura 11: Vista Edición Encuestas

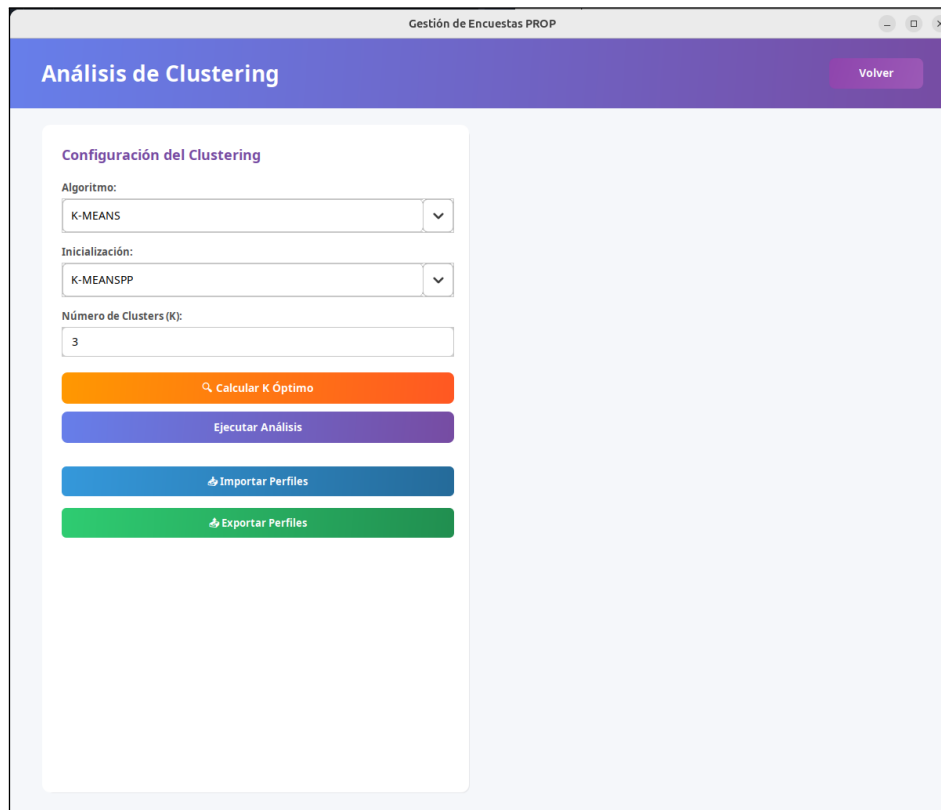


Figura 12: Vista Análisis

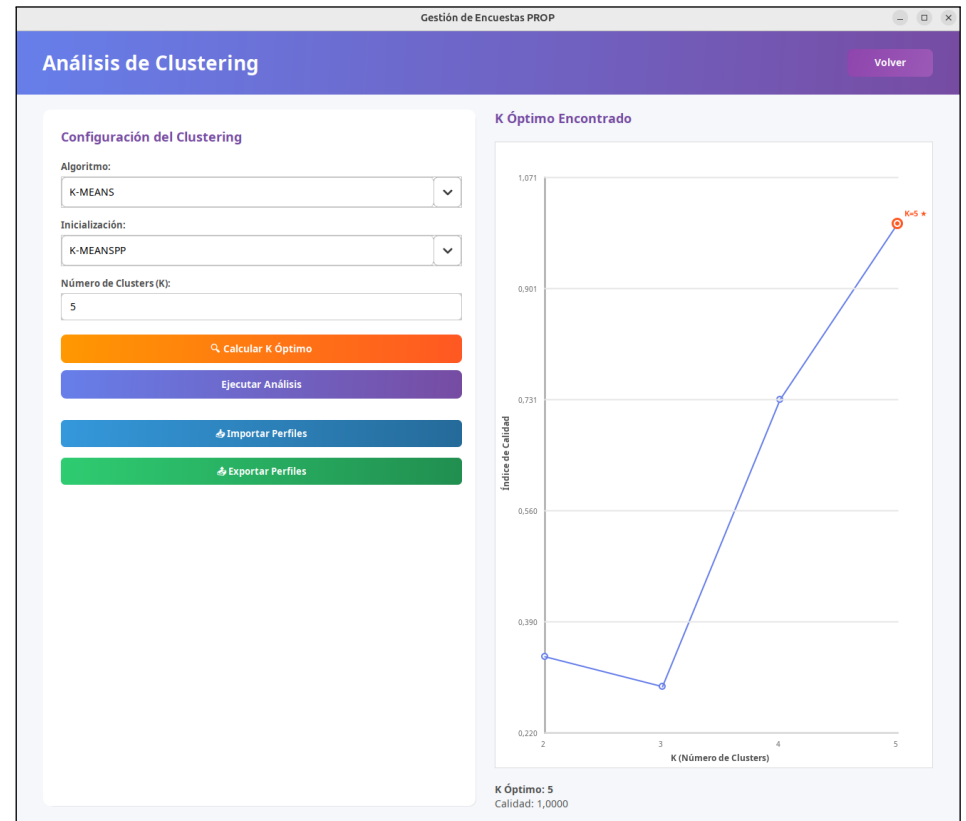


Figura 13: Vista Análisis con elección automática número clusters (k)

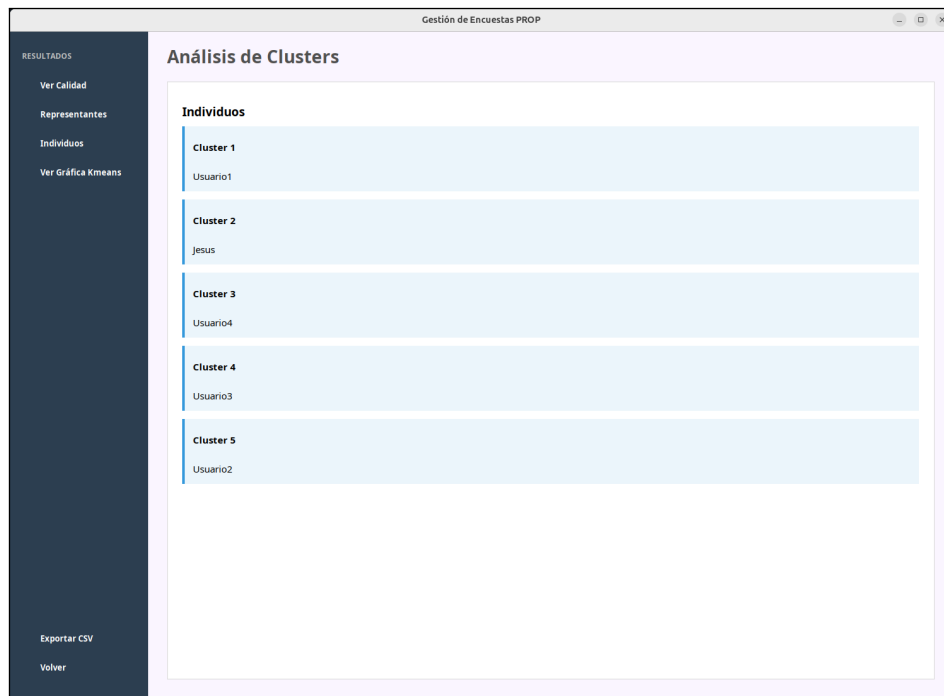


Figura 14: Vista de análisis extendido del clustering

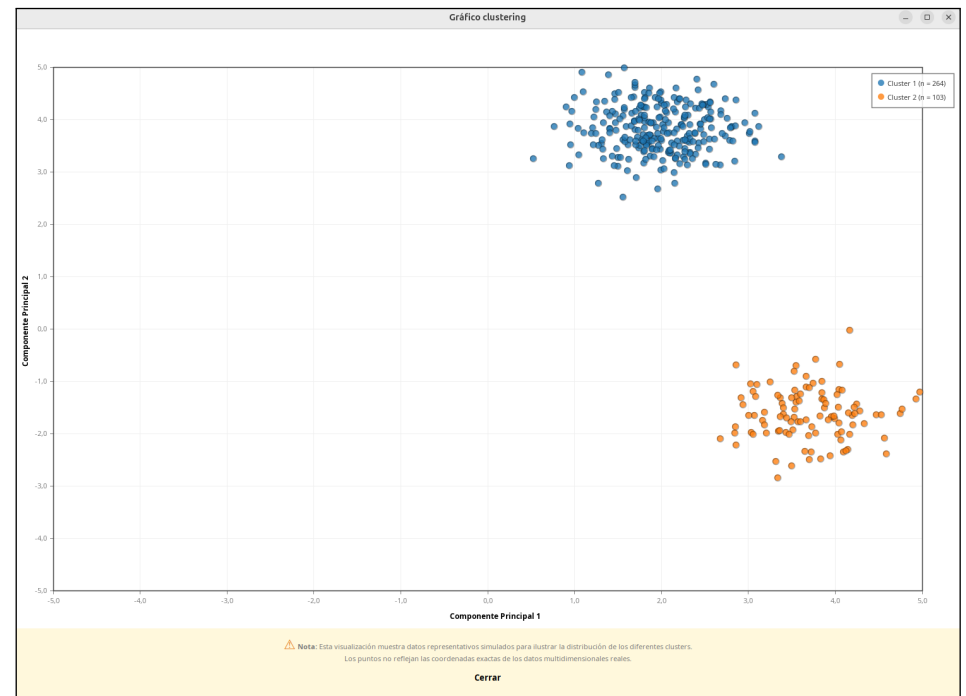


Figura 15: Gráfica que simula los puntos 2D del clustering