

# DOCUMENTACIÓ DEL CODI

Grup 14 - Subgrup 4

Pol Gay Hernández (pol.gay)

Mateus Grandolfi Albuquerque (mateus.grandolfi)

Wenqiang He (wenqiang.he)

Fardin Arafat Mia Akter (fardin.arafat.mia)

Lliurament 3.0 - Tardor 2025-26

## Índex

<b>1</b>	<b>Arquitectura del Sistema</b>	<b>3</b>
<b>2</b>	<b>Capa de Domini</b>	<b>3</b>
2.1	Paquet <code>app.controller</code> . . . . .	3
2.2	Paquet <code>kmeans</code> i <code>kselector</code> . . . . .	3
2.3	Paquet <code>Survey</code> i <code>Response</code> . . . . .	4
2.4	Paquet <code>distance</code> , <code>Encoder</code> i <code>validation</code> . . . . .	4
2.5	Altres components de Domini . . . . .	4
<b>3</b>	<b>Capa de Persistència</b>	<b>5</b>
<b>4</b>	<b>Capa de Presentació</b>	<b>5</b>
<b>5</b>	<b>Diagrama de Navegació</b>	<b>6</b>

# 1 Arquitectura del Sistema

El projecte s'ha dissenyat seguint una arquitectura clàssica per capes per garantir la separació de conceptes, la facilitat de manteniment i l'escalabilitat. Les tres capes principals són: la capa de **Presentació**, la capa de **Domini** (on resideix la lògica de negoci i els algorismes de clustering) i la capa de **Persistència**.

## 2 Capa de Domini

Aquesta és la capa central del sistema. Gestiona tota la lògica relacionada amb les enquestes, els usuaris, les respostes i, especialment, l'anàlisi de dades mitjançant algorismes de clustering.

### 2.1 Paquet `app.controller`

Aquest paquet conté els controladors de domini, que actuen com a intermediaris entre la capa de presentació i la lògica interna.

- **AnalyticsController**: Classe encarregada de coordinar l'anàlisi de les enquestes.
  - `analyzeSurvey(Survey survey, List<SurveyResponse> responses)`: És el mètode principal. Realitza el pretractament de dades mitjançant `OneHotEncoder`, executa l'algorisme de clustering configurat a l'enquesta i calcula el coeficient de *Silhouette* per validar la qualitat dels clústers. Finalment, genera una projecció 2D determinista per a la visualització en la interfície.
  - `projectTo2D(double[][] X, long seed)`: Implementa una projecció aleatòria Gaussiana per reduir la dimensionalitat de les dades d'entrada a dues dimensions, permetent la representació gràfica dels perfils.
- **SurveyController**: Gestiona el cicle de vida de les enquestes.
  - `createSurvey(...)`: Instancia noves enquestes validant els paràmetres inicials.
  - `saveSurvey(Survey survey)`: Aplica regles de negoci estrictes (títol <40 caràcters, descripció <150, mínim una pregunta obligatòria) abans de persistir-les.
  - `parseAnswers(String answersStr, Survey survey)`: Descompon una cadena de text amb format `id:valor` en una llista d'objectes `Answer` segons el tipus de pregunta.
- **UserController**: Administra les sessions i la identitat dels usuaris.
  - `createGuestSession()`: Genera usuaris temporals amb identificadors únics basats en el temps del sistema.
  - `login/register`: Connecta amb l'`AuthService` per gestionar credencials de `RegisteredUser`.
- **ResponseController**: Gestiona la recollida i recuperació de les respostes de les enquestes. Permet filtrar respostes per usuari o per enquesta específica.

### 2.2 Paquet `kmeans i kselector`

Conté la implementació del nucli algorítmic del projecte.

- **IClusteringAlgorithm**: Interfície que defineix el contracte per a qualsevol algorisme de clustering mitjançant el mètode `fit`.

- **KMeans**: Implementació de l'algorisme K-Means estàndard.
  - `fit(...)`: Minimitza la inèrcia (suma de distàncies al quadrat) de manera iterativa. Inclou lògica per gestionar clústers buits reassignant el punt més llunyà del seu centroides actual.
- **KMeansPlusPlus**: Estén **KMeans** amb una inicialització intel·ligent. Selecciona els centroides inicials amb una distribució de probabilitat basada en la distància al quadrat, accelerant la convergència i evitant mínims locals subòptims.
- **ElbowMethod**: Implementa la selecció automàtica de  $k$ . Calcula la inèrcia per a un rang de  $k$  i selecciona el colze de la corba buscant el punt de màxima distància perpendicular a la línia que uneix els extrems del rang (curvatura màxima).

## 2.3 Paquet Survey i Response

Defineix les estructures de dades fonamentals.

- **Survey**: Classe agregada que conté metadades (títol,  $k$ , mètode d'inicialització) i una llista de **Question**.
- **Question (i subclasses)**: Jerarquia que inclou **SingleChoiceQuestion**, **MultipleChoiceQuestion**, **OpenIntQuestion** i **OpenStringQuestion**. Cada tipus defineix les seves pròpies regles de validació.
- **SurveyResponse**: Representa un conjunt de respostes d'un usuari a una enquesta.
- **Answer (i subclasses)**: Classes que contenen el valor de la resposta (text, enter o opcions triades) vinculat a un identificador de pregunta.

## 2.4 Paquet distance, Encoder i validation

- **Distance**: Interfície per al càlcul de proximitat. Implementacions: **EuclideanDistance** (per a valors numèrics) i **CosineDistance** (ideal per a dades de text o dades d'alta dimensió normalitzades).
- **OneHotEncoder**: Transforma les respostes qualitatives (opcions de selecció) en vectors numèrics binaris. Això permet que els algorismes matemàtics com K-Means puguin processar dades no numèriques.
- **Silhouette**: Calcula la cohesió i la separació de cada punt respecte al seu clúster i als clústers veïns. Proporciona una mètrica objectiva entre -1 i 1 sobre la qualitat de l'agrupament.

## 2.5 Altres components de Domini

- **AuthService**: Gestiona la seguretat. Implementa un sistema de *hashing* per a les contrasenyes i manté el mapa de sessions actives (**Sesion**).
- **TxtSurveySerializer**: Implementa la importació i exportació en formats basats en text, parsejant les línies per reconstruir els objectes de l'enquesta.

### 3 Capa de Persistència

Aquesta capa s'encarrega de l'emmagatzematge permanent de les dades en fitxers, ja que no s'utilitzen bases de dades externes.

- **PersistenceDriver**: Actua com una *Facade* que centralitza l'accés a les diferents unitats de persistència (Enquestes, Usuaris i Respostes).
- **UserPersistence**: Gestiona el fitxer `userdata.json`.
  - Utilitza mètodes sincronitzats (`synchronized`) per evitar condicions de carrera en escriure al disc.
  - Implementa un motor de serialització JSON propi (`serialize/extractJsonObject`s) per mantenir el control total sobre l'estructura de dades sense dependències externes.
- **SurveyPersistence**: Emmagatzema les enquestes de forma independent. Cada enquesta es guarda amb les seves preguntes i configuració de clustering.
- **ResponsePersistence**: Gestiona l'històric de respostes. Permet fer *append* de noves respostes sense haver de reescriure tot el fitxer, optimitzant les operacions d'E/S.

### 4 Capa de Presentació

La interfície d'usuari està construïda amb tecnologies web modernes dins d'un entorn **Electron**, que permet l'execució del codi Java en el *backend*.

- **controller.ts (Bridge)**: És el component crític que permet la comunicació entre la interfície (React/TypeScript) i la lògica de negoci (Java).
  - `sendCommand(command: string)`: Codifica les accions de l'usuari en una cadena de text seguint un protocol definit (ex: `CREATE_SURVEY_FULL|...`) i l'envia al procés Java via Electron IPC.
  - `onResponse(cb)`: Escolta les respostes del backend, extreu els *chunks* JSON dels missatges de la consola i els transforma en objectes de l'estat de l'aplicació.
- **Components React**:
  - **SurveyEditor**: Interfície interactiva per a la creació de preguntes, configuració de *k* i mètodes d'inicialització.
  - **SurveyAnalyzer**: Visualitza els resultats del clustering mitjançant gràfics de dispersió 2D i mètriques de qualitat.
  - **SurveyList**: Component principal de navegació i gestió d'enquestes (Dashboard).
    - \* Implementa la lògica de filtratge dinàmic per presentar tres vistes diferents: repositori global d'enquestes, gestió de les enquestes pròpies (creades per l'usuari) i historial de participació (enquestes respostes).
    - \* Integra les operacions de persistència local interactuant amb el **controller** per a la importació i exportació de fitxers amb extensió pròpia (`.tbs`).
    - \* Gestiona la comunicació asíncrona amb el procés *Main* d'Electron per obrir diàlegs de sistema natius i capturar errors d'E/S en temps real.
  - **Auth**: Gestiona els formularis de registre i inici de sessió.

## 5 Diagrama de Navegació

La Figura 1 il·lustra el flux de navegació de l'aplicació, centrat en el component **SurveyList** com a distribuïdor principal (Dashboard). L'arquitectura de la interfície es basa en:

- **Navegació Centralitzada:** Totes les rutes principals (Respondre, Analitzar, Crear) neixen i retornen al llistat d'enquestes, simplificant la gestió de l'estat.
- **Protecció de Rutes (Auth Guard):** L'accés al **SurveyEditor** està protegit per una verificació d'autenticació intermèdia. Si l'usuari no té sessió activa, el sistema desvia el flux cap al component **Auth** i, només en cas d'èxit, redirigeix automàticament a l'editor.
- **Fluxos d'Edició Avançada:** S' destaca en vermell el cicle d'edició de respostes. A diferència del flux estàndard, aquest connecta directament l'**Analyzer** amb el **Responder** (carregant les dades prèvies) i retorna a l'anàlisi sense passar pel menú principal, oferint una experiència d'usuari més fluida.

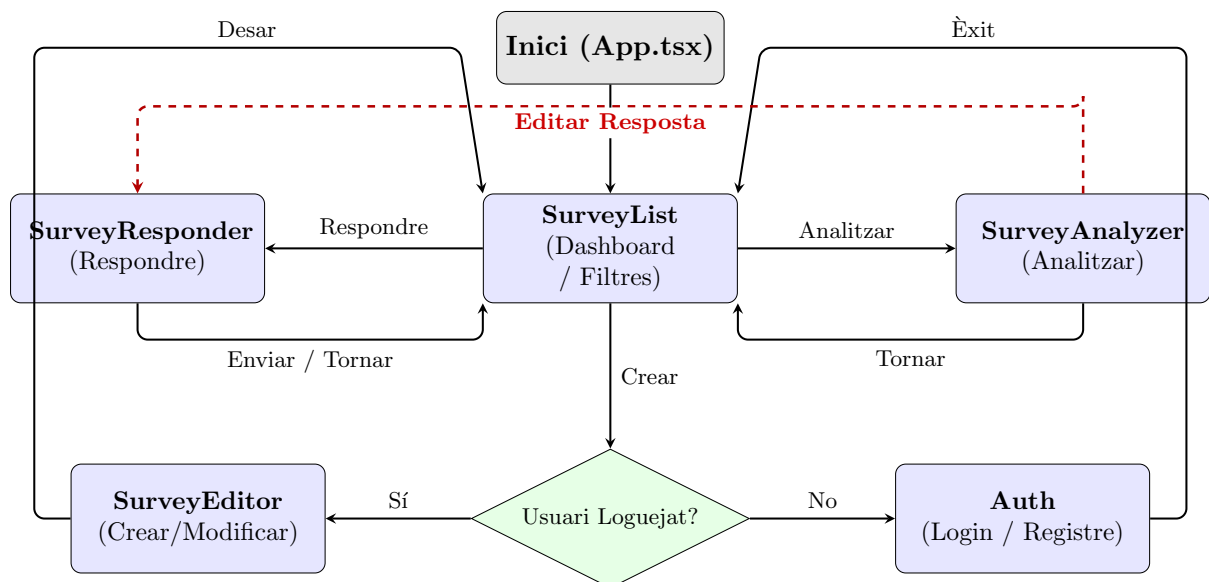


Figura 1: Diagrama de flux de navegació i estats de l'aplicació