

Documentació

Entrega v1.0

Projectes de Programació QT 2025-26



Grup 33-5

Arnau Sánchez Coll *id:arnau.sanchez.coll*

Roger Corcoles *id:roger.corcoles*

Adrià Aguilar *id:adria.aguilar*

Joaquim Fuentes *id:joaquim.fuentes*

Igor Bolige Cubas *id:igor.bolige*

ÍNDEX

1. Descripció dels Casos d'ús	2
2. Descripció d'Atributs i Mètodes	5
2.1 Classe Pregunta (abstracta)	5
2.2 Classe PreguntaInt	5
2.3 Classe PreguntaOrdinal	5
2.4 Classe PreguntaNominalUnica	6
2.5 Classe PreguntaNominalMult	6
2.6 Classe PreguntaString	6
2.7 Classe Enquesta	6
2.8 Classe Resposta	7
2.9 Classe Respostes	7
2.10 Classe Usuari	8
2.11 Classe K-means	8
2.12 Classe K-medoids	
2.13 Classe Anàlisi	8
3. Classes Implementades per Cada Membre	8
4. Estructures de dades i algorismes utilitzats	9

1. Descripció dels Casos d'ús

Els casos d'ús descrits a continuació representen les principals funcionalitats que ofereix el sistema d'enquestes, tant des del punt de vista del creador o administrador com del participant o analista.

Cada cas d'ús especifica breument l'objectiu de la interacció, l'actor principal implicat, les precondicions necessàries, el flux principal d'accions i les possibles alternatives o excepcions.

L'objectiu d'aquesta secció és complementar el diagrama de casos d'ús amb una descripció textual que permeti entendre clarament el comportament del sistema i el paper de cada actor dins del procés general de gestió, resposta i anàlisi d'enquestes.

Gestió d'Usuaris

- Objectiu: agrupar les operacions d'alta, consulta i manteniment d'usuaris.
- Actor: Creador/Administrador.
- Notes: Paraigües dels UC: Crear usuari, Llistar usuaris, Consultar usuari, Modificar usuari, Crear enquesta per usuari, Llistar enquestes d'un usuari.

Crear usuari

- Objectiu: donar d'alta un nou usuari al sistema.
- Actor: Creador/Administrador.
- Precondicions: actor amb permisos; nom i email facilitats.
- Flux principal: l'actor introdueix nom i email; el sistema valida (no buits, format d'email vàlid, email no duplicat); el sistema assigna un identificador únic (idU) i crea l'usuari; confirma l'alta mostrant idU.
- Alternatives: nom/email nuls o buits → error de validació; email amb format invàlid → error; email duplicat → operació rebutjada.
- Postcondicions: usuari creat i disponible per a consultes i per crear-hi enquestes.

Llistar usuaris

- Objectiu: obtenir el llistat d'usuaris registrats.
- Actor: Creador/Administrador.
- Precondicions: cap.
- Flux principal: l'actor sol·licita el llistat; el sistema retorna la col·lecció d'usuaris (idU, nom, email) ordenada (p. ex. per idU).
- Alternatives: no hi ha usuaris → es mostra llista buida o missatge informatiu.
- Postcondicions: cap canvi d'estat.

Consultar usuari

- Objectiu: visualitzar les dades d'un usuari concret.
- Actor: Creador/Administrador.
- Precondicions: idU vàlid i existent.
- Flux principal: l'actor indica idU; el sistema recupera i mostra nom, email i metadades associades (p. ex. nombre d'enquestes creades).
- Alternatives: idU inexistent → "usuari no trobat"; falta de permisos → accés denegat.
- Postcondicions: cap canvi d'estat.

Modificar usuari

- Objectiu: actualitzar dades bàsiques de l'usuari (nom, email).
- Actor: Creador/Administrador.
- Precondicions: idU vàlid; dades noves facilitades.
- Flux principal: l'actor edita nom i/o email; el sistema valida (no buits, email amb format vàlid, email no duplicat); el sistema aplica els canvis i confirma.
- Alternatives: validacions fallides (nom/email buits, email invàlid, email ja existent) → operació rebutjada; cancel·lació de l'actor.
- Postcondicions: l'usuari queda actualitzat en memòria (i en persistència si després s'executa "Guardar").

Crear encuesta per usuari

- Objectiu: associar una nova encuesta a un usuari existent.
- Actor: Creador/Administrador.
- Precondicions: idU existent; títol d'enquesta no buit.
- Flux principal: l'actor indica idU i títol; el sistema valida; crea l'enquesta amb idE únic i la vincula a idU; retorna confirmació amb idE.
- Alternatives: usuari inexistent → error; títol nul/buit → error de validació.
- Postcondicions: encuesta creada i associada a l'usuari; disponible per ser modificada/guardada.

Llistar les encuestes d'un usuari

- Objectiu: obtenir totes les encuestes creades per un usuari.
- Actor: Creador/Administrador.
- Precondicions: idU vàlid i existent.
- Flux principal: l'actor indica idU; el sistema retorna la llista d'enquestes (idE, títol, estat si escau) associades a l'usuari.
- Alternatives: usuari sense encuestes → es retorna llista buida o missatge informatiu; idU inexistent → error.
- Postcondicions: cap canvi d'estat.

Gestió d'Encuestes

- Objectiu: agrupar les operacions d'administració d'enquestes.
- Actor: Creador/Administrador.
- Notes: Paraigües dels UC: Guardar Encuesta, Modificar Encuesta (Modificar títol encuesta, Afegir pregunta a encuesta, Eliminar pregunta de encuesta), Eliminar Encuesta, Crear encuesta, Consultar encuesta.

Guardar Encuesta

- Objectiu: persistir en disc/BD l'estat actual de l'enquesta.
- Actor: Creador/Administrador.
- Precondicions: Encuesta carregada i vàlida.
- Flux principal: el sistema serialitza i desa; confirma operació.
- Alternatives: error d'IO; encuesta invàlida → no es desa.
- Postcondicions: versió guardada disponible per reobrir/importar.

Modificar Encuesta

- Objectiu: canviar metadades i/o conjunt de preguntes d'una encuesta existent.
- Actor: Creador/Administrador.

- Precondicions: Enquesta existent carregada.
- Flux principal: l'actor edita títol, descripció, obligatòries, afegeix/elimina/modifica preguntes; el sistema valida la definició després de cada canvi.
- Alternatives: cancel·lar canvis; errors de validació (camp buit, opcions buides...).
- Postcondicions: l'estat editat queda en memòria de treball; (persistència si després es fa Guardar Enquesta).

Crear Enquesta

- Objectiu: donar d'alta una nova enquesta al sistema.
- Actor: Creador/Administrador.
- Precondicions: usuari autenticat i existent al sistema, títol d'enquesta disponible i no buit.
- Flux principal: l'actor indica el títol (i opcionalment una descripció), el sistema valida camps obligatoris, assigna un identificador únic (idE) i crea l'enquesta buida (sense preguntes), el sistema associa l'enquesta a l'actor creador i el sistema mostra confirmació amb l'idE.
- Alternatives: títol nul o buit → error de validació, usuari inexistent o sense permisos → operació rebutjada.
- Postcondicions: l'enquesta queda creada en memòria de treball i disponible per ser consultada o modificada; la persistència en disc/BD es fa només si posteriorment s'executa "Guardar Enquesta".

Consultar Enquesta

- Objectiu: visualitzar la informació d'una enquesta existent.
- Actor: Creador/Administrador.
- Precondicions: enquesta existent identificador d'enquesta (idE) vàlid, permisos suficients.
- Flux principal: l'actor selecciona l'enquesta pel seu idE; el sistema recupera i mostra el títol, l'id del creador, i la llista de preguntes (idP, tipus i enunciat); el sistema pot mostrar metadades addicionals (nombre de preguntes, estat, etc.).
- Alternatives: idE inexistent → error "enquesta no trobada"; enquesta sense preguntes → es mostra buida; falta de permisos → accés denegat.
- Postcondicions: cap canvi d'estat; l'enquesta continua disponible per a modificació o guardat.

Eliminar Enquesta

- Objectiu: suprimir una enquesta del repositori/persistència.
- Actor: Creador/Administrador.
- Precondicions: enquesta existent seleccionada; confirmació de l'usuari.
- Flux principal: el sistema elimina i informa.
- Alternatives: cancel·lació; dependències que impedeixen eliminar.
- Postcondicions: l'enquesta ja no és disponible a llistats ni per a resposta.

Gestió de Preguntes

- Objectiu: agrupar totes les operacions de definició i consulta del banc de preguntes.
- Actor: Creador/Administrador.
- Notes: Paraigües dels UC: Crear pregunta (text / numèrica / opció única / opció múltiple / ordinal) i Imprimir totes les preguntes. La vinculació d'una pregunta a una enquesta es fa des de Modificar Enquesta (afegir/eliminar pregunta) i dona accés al catàleg de preguntes perquè després puguin ser associades a enquestes o modificades en altres casos d'ús d'administració.

Crear pregunta (text / numèrica / opció única / opció múltiple / ordinal)

- Objectiu: donar d'alta una nova pregunta del tipus desitjat per utilitzar-la a les enquestes.
- Actor: Creador/Administrador.
- Precondicions: sistema en execució, gestor de preguntes accessible, actor amb permisos; enunciat no buit; tipus seleccionat.
- Flux principal: L'actor tria el tipus de pregunta. Introdueix enunciat i marca si és obligatòria. Segons el tipus, el sistema demana rang numèric o llistat d'opcions/ordre. El sistema valida: enunciat no buit; rangs coherents; opcions no buides ni duplicades; paràmetres específics. Si tot és correcte, s'assigna idP únic i s'afegeix al catàleg. Es mostra confirmació.
- Alternatives: cancel·lació de la creació; errors de validació (enunciat buit, $\min \geq \max$, opcions nul·les/buides/insuficients, qMax incoherent). Es mostra missatge i es permet reintroduir dades sense tancar el gestor.
- Postcondicions: la pregunta queda registrada amb idP únic i disponible per vincular a enquestes; la persistència dependrà del cas d'ús Guardar Enquesta.

Tipus i validacions específiques:**Text**

- Paràmetres: enunciat, obligatòria (opcionalment longitud màxima).
- Validacions: enunciat no buit.
- Errors típics: enunciat buit o nul.

Numèrica (integer)

- Paràmetres: enunciat, obligatòria, min, max.
- Validacions: $\min < \max$ i valors vàlids.
- Errors típics: $\min \geq \max$, valors no numèrics.

Opció única (nominal_única)

- Paràmetres: enunciat, obligatòria, opcions.
- Validacions: ≥ 2 opcions, sense duplicats ni buides.
- Errors típics: una sola opció, duplicats, cadenes buides.

Opció múltiple (nominal_múltiple)

- Paràmetres: enunciat, obligatòria, opcions, qMax (màxim seleccions).
- Validacions: ≥ 2 opcions, sense duplicats ni buides; $1 \leq qMax \leq \text{nombre d'opcions}$.
- Errors típics: qMax fora de rang; opcions invàlides.

Ordinal

- Paràmetres: enunciat, obligatòria, llista ordenada d'opcions (p. ex. [Baix, Mitjà, Alt]).
- Validacions: ≥ 2 opcions, sense duplicats ni buides; ordre explícit.
- Errors típics: menys de 2 opcions, duplicats, cadenes buides.

Imprimir totes les preguntes

- Objectiu: llistar en pantalla totes les preguntes definides al sistema, amb informació bàsica.
- Actor: Creador/Administrador.
- Precondicions: cap (si el catàleg és buit, es mostra un missatge).

- Flux principal: l'actor selecciona l'opció d'imprimir; el sistema recorre el catàleg i mostra per cada pregunta: idP, tipus (text, integer, nominal única, nominal múltiple, ordinal), enunciat, obligatorietat i, si escau, rang min/max, opcions i qMax.
- Alternatives: catàleg buit → missatge “No hi ha preguntes registrades”; si hi ha errors d'E/S de consola, es pot repetir l'operació o tornar al menú.
- Postcondicions: cap canvi d'estat; només es mostra l'estat actual del banc per facilitar-ne l'edició o la vinculació a enquestes en altres UC.

Gestió de Respostes

- Objectiu: agrupar les operacions per crear, consultar i gestionar respostes d'enquestes.
- Actor: Respondent/Usuari (quan contesta); Creador/Administrador (quan consulta o elimina).
- Notes: Paraigües dels UC: Mostrar Respostes, Eliminar Respostes, Contestar Enquesta, Get Resposta, Eliminar Resposta.

Mostrar Respostes

- Objectiu: visualitzar totes les respostes d'un usuari a una enquesta.
- Actor: Creador/Administrador.
- Precondicions: existeixen respostes per a la parella (idE, idU).
- Flux principal: L'actor indica idE (enquesta) i idU (usuari). El sistema recupera el conjunt Respostes amb `getRespostes(idE, idU)` i llista, per cada idP, el tipus i el contingut. El sistema pot calcular i mostrar metadades (p. ex., nombre de respostes vs. preguntes totals).
- Alternatives: (idE, idU) inexistent → missatge “No existeixen respostes per a (idE, idU)”. Enquesta sense cap resposta d'aquest usuari → llista buida o missatge informatiu.
- Postcondicions: cap canvi d'estat; només visualització.

Eliminar Respostes

- Objectiu: suprimir totes les respostes d'un usuari a una enquesta.
- Actor: Creador/Administrador.
- Precondicions: (idE, idU) vàlids; confirmació de l'actor.
- Flux principal: L'actor indica idE i idU i confirma l'operació. El sistema elimina el bloc de respostes amb `EliminarRespostes(idE, idU)`. El sistema informa que les respostes s'han eliminat correctament.
- Alternatives: No existeixen respostes per (idE, idU) → missatge d'error i no es fa res. Cancel·lació de l'actor → no es fa res.
- Postcondicions: el conjunt Respostes(idE, idU) deixa d'existir; l'usuari queda “sense respondre” aquesta enquesta.

Contestar Enquesta

- Objectiu: permetre que un usuari respongui totes o part de les preguntes de una enquesta (crear bloc de respostes i afegir-ne una per pregunta).
- Actor: Respondent/Usuari (o Creador/Administrador actuant en nom seu).
- Precondicions: Enquesta publicada i accessible. idE existent; idU existent. L'usuari no té ja creat un bloc Respostes(idE, idU) (si ja existeix, caldria esborrar o disposar d'un “modificar resposta”).
- Flux principal: El sistema crea el contenidor buit de respostes amb `creaRespostes(idE, idU)`. Per a cada pregunta (idP) de l'enquesta: L'actor introdueix el contingut segons el tipus de pregunta. El sistema valida:
 - Text: cadenes vàlides (i obligatòries si escau). Integer: dins del rang [min, max]. Nominal única: valor dins de les opcions definides. Nominal múltiple: subconjunt d'opcions, sense duplicats i $i \leq qMax$.

Ordinal: índex/opció dins de la llista ordenada. Si és vàlid, el sistema desa la resposta amb `AfegeixResposta(idE, idU, idP, contingut)`. En acabar, el sistema confirma que l'enquesta s'ha contestat.

- Alternatives: Bloc ja existent per (idE, idU) → error: ja hi ha respostes; es pot cancel·lar o eliminar prèviament i tornar a començar. Validació fallida d'una resposta → es mostra l'error (fora de rang, opció inexistent, massa seleccions, camp obligatori buit, duplicats, etc.) i es demana reintroduir el valor. Pregunta no pertany a l'enquesta idE → error i no es desa. Sortida sense guardar.
- Postcondicions: totes les respostes vàlides queden registrades per (idE, idU); si hi ha preguntes no contestades i eren obligatòries, el procés es considera incomplet fins que es validin.

Contestar Pregunta

- Objectiu: capturar una resposta per a una pregunta concreta.
- Actor: Enquestat.
- Precondicions: pregunta visible; si és obligatòria, s'ha de completar abans d'enviar.
- Flux principal: l'usuari introdueix valor (numèric, opció/s, text); el sistema comprova coherència bàsica (tipus).
- Alternatives: deixar buida si no és obligatòria; format invàlid → missatge.
- Postcondicions: la resposta queda en l'esborrany/sessió.

Save

- Objectiu: desar l'estat parcial de les respostes (continuar més tard).
- Actor: Enquestat.
- Precondicions: hi ha almenys una resposta editada.
- Flux principal: el sistema desa parcialment (sessió/esborrany).
- Alternatives: error d'IO.
- Postcondicions: es pot reprendre la resposta des del mateix punt.

Eliminar Resposta

- Objectiu: esborrar la resposta d'una pregunta dins de la sessió de resposta.
- Actor: Enquestat.
- Precondicions: existeix una resposta prèviament guardada/introduïda.
- Flux principal: l'actor demana eliminar; el sistema la treu de l'esborrany.
- Alternatives: cancel·lació.
- Postcondicions: la pregunta torna a estat "sense resposta".

Get Resposta

- Objectiu: obtenir la resposta d'un usuari a una pregunta concreta d'una enquesta.
- Actor: Creador/Administrador.
- Precondicions: idE, idU, idP vàlids; la resposta existeix.
- Flux principal: L'actor indica idE, idU, idP. El sistema recupera la resposta amb `GetResposta(idE, idU, idP)` i en mostra el contingut i metadades (tipus, validacions si es vol).
- Alternatives: No existeix el bloc (idE, idU) → error: no hi ha respostes. No existeix la resposta per idP → error: resposta inexistent.
- Postcondicions: cap canvi d'estat; només consulta puntual.

Eliminar Resposta

- Objectiu: eliminar la resposta d'un usuari per a una pregunta concreta.

- Actor: Creador/Administrador.
- Precondicions: idE, idU, idP vàlids; la resposta existeix.
- Flux principal: L'actor indica idE, idU, idP i confirma l'operació. El sistema elimina la resposta amb EliminarResposta(idE, idU, idP). El sistema informa de l'eliminació correcta.
- Alternatives: Resposta inexistent per idP → error i no es fa res. Si en eliminar l'última resposta d'un bloc (idE, idU), el sistema pot (segons disseny) deixar el bloc buit o esborrar el bloc sencer — en la vostra implementació, el bloc es manté fins que es cridi EliminarRespostes.
- Postcondicions: la resposta esdevé inexistent; si era obligatòria, l'enquesta queda incompleta per aquest usuari fins que torni a respondre.

Gestió d'Anàlisi

- Objectiu: agrupar les operacions de configuració i execució d'anàlisi de respostes (clustering).
- Actor: Creador/Administrador.
- Notes: Paraigües dels UC: Carregar Anàlisi, Executar K-Means, Executar K-Medoids, Visualitzar Resultat d'Anàlisi (silhouette, inèrcia/cost, assignacions i llistat de clústers).

Carregar Anàlisi

- Objectiu: preparar i llançar una anàlisi de clustering sobre una enquesta existent.
- Actor: Creador/Administrador/Analista.
- Precondicions: Existeix una enquesta (idE) vàlida al sistema. Hi ha respostes registrades per a aquesta enquesta ($n \geq 1$). Paràmetres d'execució disponibles: algorisme (K-Means o K-Medoids), k ($1 < k \leq n$) i maxIter (> 0).
- Flux principal: L'actor obre el menú d'Anàlisi. L'actor selecciona l'algorisme: K-Means o K-Medoids. L'actor introdueix idE, k i maxIter. El sistema valida: idE existent; n de respostes suficient; $1 < k \leq n$; maxIter > 0 . El sistema carrega el dataset de (idE) i prepara la representació: Si és K-Means: vectoritza totes les preguntes (one-hot/multi-hot per nominals, normalització per numèriques, encoder de text) i empra distància euclidiana. Si és K-Medoids: utilitza la distància de domini entre Respostes. El sistema executa l'algorisme amb els paràmetres indicats (maxIter) i obté assignacions/clústers i mètrica principal (inèrcia per K-Means o cost per K-Medoids). El sistema calcula la silhouette: Euclidiana sobre els vectors (K-Means). De domini sobre la matriu de distàncies (K-Medoids). El sistema mostra el resultat resum: Algorisme, k , temps (ms), silhouette, cost/inèrcia. Assignació de cada mostra al seu clúster i , opcionalment, el contingut de cada clúster (llista d'índexs/ids).
- Alternatives: idE inexistent → error "enquesta no trobada". Sense respostes o k fora de rang → error de validació (indicar condicions vàlides). maxIter no vàlid → error (ha de ser > 0). Falla la vectorització (K-Means) o la distància de domini (K-Medoids) → es mostra error i es cancel·la l'execució. Execució sense convergència dins de maxIter → es retorna l'últim estat amb avís.
- Postcondicions: No es modifica cap dada persistent; s'ha generat un resultat d'anàlisi disponible a pantalla (i opcionalment exportable). L'actor pot repetir l'anàlisi amb altres k /maxIter o canviar d'algorisme.

K-means (inclou → Guardar, Desplegar Resultats)

- Objectiu: aplicar el clúster K-means sobre les respostes importades.
- Actor: Creador/Analista.
- Precondicions: dades carregades (Importar Respostes) o Carregar Anàlisi.
- Flux principal: l'actor tria K i paràmetres; el sistema executa l'algorisme; inclou Guardar del model/resultats i inclou Desplegar Resultats (taules/gràfics).
- Alternatives: K invàlid; dades incompletes.

- Postcondicions: resultats calculats, mostrats i guardats si s'escull.

K-medoids (inclou → Guardar, Desplegar Resultats)

- Objectiu: aplicar el clúster K-medoids sobre les respostes importades.
- Actor: Creador/Analista.
- Precondicions: dades carregades (Importar Respostes) o Carregar Anàlisi.
- Flux principal: l'actor tria K i paràmetres; el sistema executa l'algorisme; inclou Guardar del model/resultats i inclou Desplegar Resultats (taules/gràfics).
- Alternatives: K invàlid; dades incompletes.
- Postcondicions: resultats calculats, mostrats i guardats si s'escull.

Guardar (resultats d'anàlisi)

- Objectiu: persistir models/resultats d'anàlisi (centroïdes, assignacions, mètriques).
- Actor: Creador/Analista.
- Precondicions: existeix un resultat recent (de K-means o K-medoids).
- Flux principal: el sistema serialitza resultats/metadades.
- Alternatives: error d'IO.
- Postcondicions: anàlisi disponible per a càrrega futura.

Desplegar Resultats

- Objectiu: mostrar resultats del clúster (taules, mètriques, gràfics bàsics).
- Actor: Creador/Analista.
- Precondicions: s'han generat resultats d'un algorisme.
- Flux principal: el sistema calcula i presenta resums (p. ex., mida de clústers, distàncies, centrals).
- Alternatives: dades insuficients.
- Postcondicions: resultats visibles; sense canvis en persistència.

NouImportCSV

- Objectiu: agrupar les operacions d'importació d'enquestes a partir d'arxius CSV.
- Actor: Creador/Administrador.
- Notes: Paraigües dels UC: Importar enquesta des de CSV, Veure enquestes importades.

Importar enquesta des de CSV

- Objectiu: crear una nova enquesta al sistema a partir de carregar un fitxer CSV resolt (capçalera = camps/preguntes; files = usuaris/respostes).
- Actor: Creador/Administrador.
- Precondicions: fitxer CSV accessible; format vàlid (columna d'identificador d'usuari i columnes de preguntes); usuari creador existent.
- Flux principal: l'actor indica ruta CSV, títol i id del creador; el sistema llegeix la capçalera, infereix tipus de preguntes (numèriques/nominals/ordinal... segons regles), crea la nova enquesta (idE) amb les preguntes, dona d'alta les respostes per cada usuari detectat i confirma.
- Alternatives: fitxer inexistent o danyat → error d'I/O; capçalera invàlida/columnes obligatòries absents → error de validació; valors no parsables → es normalitzen a "MISSING" o es reporta fila descartada.
- Postcondicions: enquesta creada amb idE nou i respostes associades; queda disponible per consulta i anàlisi.

Veure enquestes importades

- Objectiu: Llistar totes les enquestes que han estat creades via importació CSV.
- Actor: Creador/Administrador.
- Precondicions: almenys una importació prèvia (en cas contrari, llista buida).
- Flux principal: l'actor sol·licita el llistat; el sistema mostra idE, títol, creador, nombre de preguntes i nombre de respostes (i, si escau, la ruta o etiqueta del CSV d'origen).
- Alternatives: cap importació registrada → missatge informatiu “No hi ha enquestes importades”.
- Postcondicions: cap canvi d'estat; només consulta.

2. Descripció breu d'Atributs i Mètodes

En aquest apartat es descriuen les classes principals del model conceptual en la seva versió de disseny, juntament amb els seus atributs i mètodes més rellevants.

L'objectiu d'aquesta secció és detallar l'estructura i el comportament de cada classe, ressaltant-ne la seva funció dins del sistema i les relacions que mantenen entre elles.

Els *setters* i *getters* s'han omès per simplicitat, ja que no aporten informació addicional sobre la lògica de domini.

2.1 Classe Pregunta (abstracta)

Propòsit:

Defineix l'estructura comuna de totes les preguntes del sistema, incloent l'enunciat, la obligatorietat i la validació bàsica.

Atributs:

- ❖ idP: Integer, identificador intern assignat pel sistema.
- ❖ Enunciat: String, text que formula la pregunta.
- ❖ Obligatoria: Bool, indica si cal respondre la pregunta obligatòriament.

Mètodes:

- ❖ validarDefinició() comprova que l'enunciat no sigui buit.
- ❖ modificarEnunciat(String) actualitza el text de la pregunta.
- ❖ modificarObligatoria(Bool) canvia el valor de l'obligatorietat.
- ❖ getTipus() retorna el tipus de pregunta.

2.2 Classe PreguntaInteger

Propòsit:

Representa una pregunta numèrica, que emmagatzema un valor enter dins d'un rang.

Atributs:

- ❖ maxValue: Integer, valor màxim permès.
- ❖ minValue: Integer, valor mínim permès.

Mètodes:

- ❖ validarDefinició() comprova que el rang sigui coherent ($\text{MinValue} \leq \text{MaxValue}$).

- ❖ `modificarMinValue(Integer)` modifica el valor del `MinValue`.
- ❖ `modificarMaxValue(Integer)` modifica el valor del `MaxValue`.
- ❖ `getTipus()` retorna el tipus de pregunta “integer”.

2.3 Classe PreguntaOrdinal

Propòsit:

Modela una pregunta amb opcions ordenades (p. ex. “poc / normal / molt”).

Atributs:

- ❖ `opcions`: `Vector<String>`, llista d’opcions definides pel creador en ordre ascendent.

Mètodes:

- ❖ `validarDefinicio()` verifica que la llista no sigui buida i que cap opció sigui nul·la.
- ❖ `modificarOpcions(Vector<String>)` substitueix les opcions actuals per unes noves.
- ❖ `getTipus()` retorna el tipus de pregunta “ordinal”.

2.4 Classe PreguntaNominalUnica

Propòsit:

Pregunta no ordenada on l’usuari pot triar una sola opció.

Atributs:

- ❖ `opcions`: `Set<String>`, conjunt d’opcions sense duplicats.

Mètodes:

- ❖ `validarDefinicio()` comprova que hi hagi almenys una opció i que totes siguin vàlides.
- ❖ `modificarOpcions(Set<String>)` actualitza el conjunt d’opcions disponibles.
- ❖ `getTipus()` retorna el tipus de pregunta “nominal_unica”.

2.5 Classe PreguntaNominalMult

Propòsit:

Pregunta no ordenada que permet seleccionar diverses opcions.

Atributs:

- ❖ `opcions`: `Set<String>`, conjunt d’opcions sense duplicats.
- ❖ `qMax`: `Integer`, màxim d’opcions que es poden triar.

Mètodes:

- ❖ `validarDefinicio()` comprova que $qMax \geq 1$ i que no superi el nombre d’opcions disponibles i que les opcions siguin vàlides.
- ❖ `modificarOpcions(Set<String>)` actualitza les opcions.
- ❖ `modificarQMax(Integer)` actualitza el valor màxim de seleccions.
- ❖ `getTipus()` retorna el tipus de pregunta “nominal_multiple”.

2.6 Classe PreguntaText

Propòsit:

Pregunta oberta que permet una resposta lliure en format text.

Atributs:

No té atributs addicionals.

Mètodes:

- ❖ validarDefinicio() reusa la validació bàsica d'enunciat.
- ❖ getTipus() retorna el tipus de pregunta "string".

2.7 Classe Enquesta

La classe Enquesta representa una enquesta concreta creada per un usuari, identificada per un id, amb un títol i un conjunt ordenat de preguntes associades.

Centralitza la gestió de les preguntes d'aquesta enquesta (alta, baixa, modificació i consulta), delegant la creació i obtenció de les preguntes reals al controlador de domini.

Atributs:

- ❖ idE: Integer. Identificador únic de l'enquesta
- ❖ titol: String. Títol descriptiu de l'enquesta
- ❖ idsPreguntes: ArrayList<Integer>. Llista ordenada dels identificadors de totes les preguntes que formen part de l'enquesta
- ❖ idCreador: Integer. Identificador de l'usuari que ha creat l'enquesta
- ❖ ctrlDomini: CtrlDomini. Referència al controlador de domini, utilitzada per registrar i obtenir les preguntes i l'usuari creador

Mètodes principals (sense detallar getters/setters trivials):

- ❖ Enquesta(Integer idE, String titol, Integer idCreador, CtrlDomini ctrlDomini): constructora que inicialitza una enquesta buida i valida que idE, titol, idCreador i ctrlDomini siguin vàlids.
- ❖ getIdsPreguntes(): retorna una llista no modificable amb els ids de totes les preguntes de l'enquesta, preservant l'ordre intern.
- ❖ getNumPreguntes(): retorna el nombre de preguntes actuals de l'enquesta a partir de la mida d'idsPreguntes.
- ❖ getPreguntes(): construeix i retorna una llista d'objectes Pregunta recuperats del CtrlDomini a partir de cada id emmagatzemat a idsPreguntes.
- ❖ getCreador(): retorna l'objecte Usuari corresponent a idCreador, fent la consulta al CtrlDomini.
- ❖ setTitol(String nouTitol) i modificarTitol(String nouTitol): actualitzen el títol de l'enquesta validant que el nou títol no sigui nul ni en blanc.
- ❖ afegeixPregunta(Integer idPregunta): afegeix al final de l'enquesta una pregunta ja existent (identificada pel seu id); comprova que l'id no sigui nul i que no estigui duplicat dins d'idsPreguntes.
- ❖ creaPregunta(String tipus, String enunciat, boolean obligatoria, Object[] params, Integer index): crea una nova Pregunta del tipus indicat (integer, ordinal, nominal_unica, nominal_multiple, text) mitjançant els mètodes de CtrlDomini, validant els paràmetres específics de cada tipus; registra la pregunta al domini, n'obté l'id i l'afegeix a idsPreguntes a la posició indicada (o al final si index és nul).

- ❖ `eliminaPregunta(Integer idPregunta)`: elimina de l'enquesta la pregunta identificada per `idPregunta` si existeix. Retorna `true` si s'ha eliminat correctament i `false` si no.
- ❖ `eliminaPregunta(int index)`: elimina i retorna l'id de la pregunta situada a la posició indicada.
- ❖ `modificaPregunta(int index, Integer nouIdPregunta)`: substitueix la pregunta que hi ha a la posició `index` per una nova pregunta identificada per `nouIdPregunta`, validant que l'índex sigui vàlid, que el nou id no sigui nul i que no provoqui duplicats a la llista.

2.8 Classe Resposta

Propòsit:

Conté la informació d'una resposta individual a una pregunta determinada.

Atributs:

- ❖ `contingut`: `Vector<String>`, valor o conjunt de valors introduïts.

Mètodes:

- ❖ `calcula_distancia(Resposta)` calcula la distància entre dues respostes per comparacions o anàlisi:

Integer (“integer”)

Distància relativa al rang:

$$d = |v1 - v2| / (\max - \min)$$

Fent servir `max` i `min` per normalitzar

Ordinal (“ordinal”)

Les respostes són índexs dins d'una llista ordenada.

$$d = |i1 - i2| / (nOpcions - 1)$$

Nominal única (“nominal_unica”)

$d = 0$ si `resposta1 = resposta2`, altrament $d = 1$.

Nominal múltiple (“nominal_multiple”)

Distància de Jaccard entre conjunts d'opcions triades `A` i `B`:

$$d = 1 - (|A \cap B| / |A \cup B|)$$

Text lliure (“text”)

Levenshtein normalitzat per la longitud màxima:

$$d = \text{Levenshtein}(s1, s2) / \max(|s1|, |s2|)$$

2.9 Classe Respostes

Propòsit:

Agrupa un conjunt de respostes individuals provinents d'un mateix enquestat o sessió.

Atributs:

No té atributs addicionals.

Mètodes:

- ❖ `calcula_distanciaRespostes(Resposta, Resposta)` calcula la distància global entre dues col·leccions de respostes.

2.10 Classe Usuari

Propòsit:

Representa un usuari del sistema (creador, enquestat o analista)

Atributs:

- ❖ `idU`: integer, identificador de l'usuari.
- ❖ `Email`: String, correu electrònic de contacte (ha de contenir el caracter `@` i és únic)
- ❖ `Nom`: String, nom identificador.

Mètodes:

- ❖ `getEnquestesUsuari()` : Retorna un llistat de les Enquestes que ha creat l'usuari.
- ❖ `creaEnquesta(idE, títol)` crea una nova enquesta com a creador.

2.11 Classe K-means

Base

- Hereta d'Algorisme (disposa de `ctrlDomini`, `k` i el dataset respostes).

Mètodes principals

- `executa(maxIter, unusedFlag)` : `KMeansResultat`
 1. **Vectoritza** totes les Respostes a un únic espai `double[]` consistent (one-hot/multi-hot per nominals, normalització per integers/ordinals, bossa de caràcters + longitud per text).
 2. **Inicialitza** centroides (RANDOM o K-Means++).
 3. Alterna **assignació** (punt→centroide) i **recomputació** de centroides fins a convergència o `maxIter`.
 4. Calcula **inèrcia** (suma de distàncies quadràtiques al centroide) i construeix els **clústers**.

Tipus de retorn

- `KMeansResultat` amb:

- centroids: List<double[]> — k centroides a \mathbb{R}^{DRD} .
- clusters: Map<Integer, List<Integer>> — idCluster → índexs de mostres.
- assignacio: int[] — per a cada mostra, el cluster assignat.
- inertia: double — suma d'errors quadràtics intra-cluster.

Helpers interns típics

- initRandom(...) / initKMeansPlusPlus(...) — estratègies d'inicialització.
- assignPoints(...) — pas d'assignació (minimitza distància L2 al centroide).
- recomputeCentroids(...) — mitjana per dimensió; resinsebrat si un clúster queda buit.
- sqDist(a,b) — distància quadràtica (sense arrel) per eficiència.

Notes de disseny

- Es desacobla la **vectorització** (via un component Vectoritzador) de l'algorisme K-Means, per tal que:
 - l'espai sigui **fix** i coherent per a tot el dataset,
 - es puguin afegir fàcilment noves codificacions (p. ex. TF-IDF a futur),
 - i la **silhouette euclidiana** de Anàlisi sigui comparable amb el mateix embedding.

2.12 Classe Algorisme (abstracta)

Propòsit:

Centralitza el comportament comú dels algorismes de clustering del domini (K-means i K-medoids), gestionant el conjunt de respostes a una enquesta, el nombre de clústers k i la validació estructural de les dades abans d'executar l'algorisme.

Atributs:

ctrlDomini: CtrlDomini

Referència al controlador de domini, utilitzada per accedir a altres entitats de domini si cal.

Es valida que no sigui nul al constructor.

idAlgorisme: int

Identificador intern de l'algorisme, no negatiu, que permet distingir diferents configuracions o execucions dins del domini.

k: int

Nombre de clústers a generar. Ha de ser estrictament positiu; qualsevol valor ≤ 0 provoca una `IllegalArgumentException`.

respostes: List<Respostes>

Conjunt de Respostes sobre el qual s'executarà l'algorisme. Es guarda com a còpia interna i es valida que no sigui nul ni buit i que tingui estructura coherent.

Constructors principals:

Algorisme(CtrlDomini ctrlDomini, int idAlgorisme, int k)

Inicialitza l'algorisme amb el controlador de domini, l'identificador i el nombre de clústers.

Comprova que ctrlDomini no sigui nul, que idAlgorisme sigui no negatiu i que k sigui positiu.

Algorisme(CtrlDomini ctrlDomini, int idAlgorisme, int k, List<Respostes> respostes)

Reutilitza el constructor anterior i, a més, estableix la llista de Respostes cridant `setRespostes(respostes)`, fent totes les comprovacions estructurals necessàries.

Mètodes públics principals:

int getIdAlgorisme()

Retorna l'identificador intern associat a aquest algorisme.

int getK()

Retorna el valor actual de k.

void setK(int k)

Modifica el nombre de clústers, comprovant que $k > 0$.

void setRespostes(List<Respostes> respostes)

Estableix el conjunt de Respostes sobre el qual treballarà l'algorisme.

Comprova que la llista no sigui nul·la ni buida.

protected List<Respostes> getRespostes()

Retorna la llista interna de Respostes.

abstract Object executa(int maxIter, boolean useAllPairsSwap)

Operació abstracta que defineix el contracte d'execució dels algorismes de clustering.

Mètodes protegits de validació:

protected final void validaEstructura(List<Respostes> respostes)

Verifica que totes les instàncies de Respostes dins la llista siguin compatibles entre si abans d'executar qualsevol algorisme.

2.12 Classe K-medoids

Propòsit:

La classe KMedoids implementa l'algorisme de clustering k-medoids (PAM) sobre un conjunt de Respostes, utilitzant la distància definida al domini per agrupar usuaris amb patrons de resposta similars.

Hereta de la classe abstracta Algorisme.

Atributs principals:

La classe no afegeix atributs d'instància propis, sinó que reutilitza els de la classe base Algorisme.

Disposa d'una classe interna estàtica Resultat que encapsula el resultat final del clustering: la llista de medoids, el mapa de clusters i el cost total de la solució.

Classe interna Resultat:

- medoids: List<Integer>. Índexs dins del dataset (llista de Respostes) que han estat seleccionats com a medoids
- clusters: Map<Integer, List<Integer>>. Per a cada medoid (índex del dataset), guarda la llista d'índexs de les Respostes assignades a aquest clúster.
- costTotal: double. Suma de les distàncies de cada element al seu medoid, d'acord amb la matriu de distàncies precomputada.

Constructors:

- KMedoids(CtrlDomini ctrlDomini, int idAlgorisme, int k, List<Respostes> respostes): inicialitza un algorisme k-medoids amb un controlador de domini, un identificador d'algorisme, el nombre de clusters k i el dataset complet de Respostes sobre el qual s'ha de clusteritzar.
- KMedoids(CtrlDomini ctrlDomini, int idAlgorisme, int k): inicialitza l'algorisme amb el mateix conjunt de paràmetres però deixant el dataset per establir posteriorment a la classe base.

Mètode principal:

- `Object executa(int maxIter, boolean useAllPairsSwap)`: és el punt d'entrada de l'algorisme. Comprova que `k` sigui positiu i no superi el nombre de respostes. A cada iteració calcula el millor intercanvi possible entre un medoid i un element no medoid, actualitza els medoids si el cost millora i recalcula les assignacions fins que no hi ha millora o s'arriba al màxim d'iteracions. Al final construeix el mapa de clusters i retorna un objecte `Resultat` amb medoids, clusters i cost total.

Mètodes privats més rellevants:

- `precomputaMatriuDistancies(List<Respostes> respostes)`: construeix una matriu simètrica `D` on `D[i][j]` és la distància mitjana entre les Respostes `i` i `j`; reutilitza la funció `distanciaEntreRespostes` i evita recomputacions futures.
- `distanciaEntreRespostes(Respostes a, Respostes b)`: recorre totes les preguntes `i`, per a cada parell de Resposta, crida `calcula_distanciaRespostes` a la classe `Resposta`, normalitza cada distància a l'interval `[0,1]` i retorna la mitjana de totes les distàncies.
- `inicialitzaMedoids(int n, int k)`: selecciona els índexs inicials dels medoids; per `k = 1` tria el primer element, i per `k > 1` reparteix els medoids de manera aproximadament uniforme sobre els `n` elements, afegint elements addicionals si cal per garantir que es tenen `k` medoids diferents.
- `assignaTots(double[][] D, List<Integer> medoids)`: per a cada element del dataset selecciona el medoid més proper segons la matriu `D` i construeix un mapa que assigna cada índex al seu medoid.
- `costTotal(double[][] D, List<Integer> medoids, Map<Integer, Integer> assignacio)`: calcula el cost total de la solució sumant `D[i][m]` per a cada element `i` assignat al seu medoid `m`.
- `deltaSwap(double[][] D, List<Integer> medoids, Map<Integer, Integer> assignacio, int mOut, int hIn)`: estima la variació de cost que es produiria en intercanviar un medoid existent `mOut` per un element no medoid `hIn`, utilitzant per a cada punt la millor i segona millor distància actual.
- `construeixClusters(List<Integer> medoids, Map<Integer, Integer> assignacio)`: crea, a partir de l'assignació d'elements a medoids, un mapa des de medoid a llista d'índexs de membres del clúster, emprat finalment per generar l'objecte `Resultat`.

2.13 Classe Anàlisi

Atributs clau

- `ctrl`: `CtrlDomini` — façana per obtenir metadades (tipus de preguntes, rangs, etc.) i accedir a respostes quan cal calcular distàncies.

Mètodes principals

- `executaKMeans(dades, k, maxIter)` : `ResultatAnalisi`
Orquestra una execució de K-Means sobre el conjunt `dades` (llista de Respostes). Crea l'algorisme, injecta el dataset, mesura el temps, recupera assignacions/clústers i, si escau, calcula **silhouette euclidiana** al mateix espai vectoritzat que usa KMeans. Emplena `ResultatAnalisi` amb `algorisme="KMeans"`, `k`, `millis`, `assignacio`, `clusters`, `qualitatSilhouette` i `inerciaOCost` (inèrcia).

- `executaKMedoids(dades, k, useAllPairsSwap, maxIter)` : `ResultatAnalisi`
Idèntic flux per a K-Medoids, recuperant cost i calculant **silhouette de domini** amb la distància pròpia de les respostes (la teva `calcula_distanciaRespostes`).
- `calculaSilhouette(dades, assign)` : `double`
Calcula la silhouette mitjana $[-1,1][0,1][0,1]$ utilitzant **distància de domini** (agregant la distància pregunta a pregunta, normalitzada $[0,1][0,1][0,1]$).
- `calculaSilhouetteEuclidiana(X, assign)` : `double`
Silhouette sobre vectors `double[]` (espai euclidià) — s'utilitza per validar K-Means quan el pipeline treballa amb one-hot/multi-hot/text-bag-of-chars, etc.

Notes de disseny

- `ResultatAnalisi` és un DTO homogeni per a qualsevol algorisme: encapsula cronometratge, k, assignació, clústers, mètrica de qualitat i inèrcia/cost. El camp `rawResult` conserva el resultat específic (p. ex. `KMeansResultat`) per si es vol inspeccionar després.

3. Relació de les classes implementades per cada membre de l'equip. Classes Implementades per Cada Membre

A continuació es presenta la relació de classes implementades per cada membre de l'equip durant el primer lliurament del projecte.

Cada membre ha estat responsable de la implementació, validació i prova de les classes que se li han assignat.

Membre de l'equip	Classes implementades	Responsabilitats principals
Igor Bolige	Pregunta, PreguntaInteger, PreguntaOrdinal, PreguntaNominalUnica, PreguntaNominalMult, PreguntaText, Driver Pregunta, TestPregunta i Test de totes les subclasses de Pregunta	Implementació del model de preguntes i validació de definicions.

Adrià Aguilar	CtrlDomini, DriverCsvImporter.java, DriverUsuari,Usuari, DriverDomain, Main, TestUsuari, DriverDomain	Gestió d'usuaris, creació d'enquestes a través d'usuari i interacció amb preguntes. Importar enquestes en csv i mantenir les instàncies de les classes al CtrlDomini. Guardar les instàncies dels drivers específics i fer el menú principal o es crida a cada driver.
Arnau	Enquesta, K-Medoids, Algorisme, Test Enquesta, Test K-Medoids	Gestió de les enquestes. Desenvolupament de k-medoid. Desenvolupament de la classe Algorisme
Roger Corcoles	Resposta, Respostes, LoanCSVImporter, DriverCSVClustering, DriverResposta, TestResposta, TestRespostes CsvImporter	Estructuració i tractament de les respostes individuals i conjunes. Encarregat de llegir arxius csv i passar a format intern
Joaquim Fuentes	Anàlisi, K-Means, Vectoritzador, Driver Anàlisi, Driver K-Means, Test Anàlisi, Test K-Means, Driver	Desenvolupament dels algorismes d'anàlisi i clustering.

4. Estructures de dades i algorismes utilitzats

En aquest apartat es descriuen les estructures de dades i els algorismes que s'han utilitzat per implementar les funcionalitats principals del model.

L'objectiu és justificar les decisions de disseny preses, mostrant com aquestes estructures permeten una gestió eficient de les preguntes, respostes i operacions d'anàlisi dins del sistema.

Estructures de dades

Element del sistema	Estructura utilitzada	Justificació
Preguntes ordinals	Vector<String>	Manté l'ordre definit pel creador i permet accés per índex.
Preguntes nominals	Set<String>	Evita duplicats i no imposa ordre en les opcions.

Preguntes (enquesta)	List<Pregunta>	Emmagatzema totes les preguntes en ordre de creació; fàcil recorregut seqüencial.
Respostes	Vector<String>	Guarda un o més valors textuais segons el tipus de pregunta.
Grups de respostes	*Resposta[] respostes	Representa les respostes d'un enquestat; base per a càlculs de distància.
Algorismes (K-means / K-medoids)	K-means: List<double[]> dades, List<double[]> centroides, int[] assignació, Map<Integer, List<Integer>> clústers K-medoids: double[][] matriuDistàncies, List<Integer> medoids, Map<Integer, List<Integer>> clústers	<p>K-means treballa sobre vectors numèrics i necessita calcular mitjanes i distàncies euclidianes de forma eficient, cosa que s'aconsegueix amb double[] i estructures lineals simples.</p> <p>K-medoids reutilitza una matriu de distàncies simètrica i només manipula índexs de punts com a medoids, fet que redueix càlcul redundant i simplifica l'actualització de clústers.</p>

Descripció d'algorismes

K-medoids:

K-Medoids és un algorisme de particionat que vol agrupar n observacions en k clusters minimitzant el cost total: la suma de distàncies de cada punt al seu representant, anomenat medoide. A diferència de K-Means, que utilitza centroides (mitjanes que no necessàriament pertanyen al conjunt de dades) i acostuma a dependre de la distància euclidiana, K-Medoids sempre tria com a centres punts reals del dataset i admet mètriques arbitràries; per això és més robust a valors atípics i especialment útil amb dades categòriques o mixtes.

El codi comença precomputant una matriu de distàncies entre totes les parelles de respostes. Tenim que la distància entre dos usuaris és el promig de distàncies pregunta a pregunta, acotades entre 0 i 1. Després s'inicialitzen els medoids triant k índexs repartits al rang de $[0..n-1]$. Amb aquests medoids inicials, s'assigna cada punt al medoid més proper i es calcula el cost inicial com la suma $D[i][m]$ de totes les assignacions.

El nucli d'optimització és un bucle d'intercanvis. Per a cada medoid m i cada no-medoid h es calcula el guany potencial de substituir m per h . Si la suma de canvis és negativa, el cost baixa, es fa

l'intercanvi, es reassignen tots els punts i s'actualitza el cost. Aquest procés es repeteix fins que no hi ha millora o s'assoleix el màxim d'iteracions.

La conté la llista de medoids finals, el mapa de clústers (cada medoid amb els índexs dels seus membres) i el cost total. L'algorisme pressuposa que totes les instàncies tenen el mateix nombre i ordre de característiques (mateixa enquesta i mateixes preguntes alineades), i que la mètrica per pregunta està ben normalitzada.