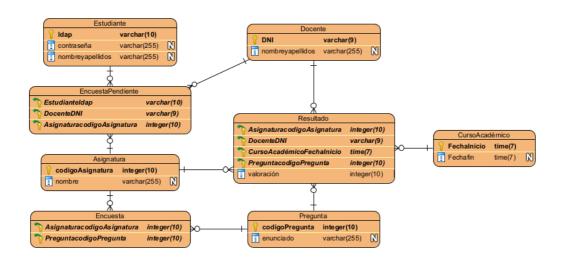
Criterios de corrección

Criterio		Ptos.
Esquema relacional de la base de datos	\rightarrow	0,4
Relaciones del modelo de dominio en el diagrama de clases	\rightarrow	0,4
Gestores y atributos necesarios en el diagrama de clases	\rightarrow	0,2
Identificación	\rightarrow	0,5
Comprobar si encuestas pendientes	\rightarrow	0,6
Mostrar encuestas pendientes	\rightarrow	0,8
Mostrar docentes asignatura seleccionada	\rightarrow	0,6
Mostrar preguntas asignatura seleccionada	\rightarrow	0,7
Comprobar si existe resultado previo	\rightarrow	0,8
Crear nuevo resultado (Objetos)	\rightarrow	1
Crear nuevo resultado (Base de datos)	\rightarrow	0,5
Actualizar resultado existente (Objetos)	\rightarrow	1
Actualizar resultado existente (Bases de Datos)	\rightarrow	0,5
Eliminar encuesta pendiente (Objetos)	\rightarrow	0,6
Eliminar encuesta pendiente (Base de datos)	\rightarrow	0,4
Bucles	\rightarrow	0,2
Condiciones	\rightarrow	0,3
Párametros	\rightarrow	0,25
JSON	\rightarrow	0,25

El diagrama de clases, la estructura de los JSON y los diagramas de secuencia que se presentan en esta solución son una de las múltiples posibilidades correctas de implementación del caso de uso.

Esquema Relacional



Estructura JSON (una de las muchas posibilidades correctas)

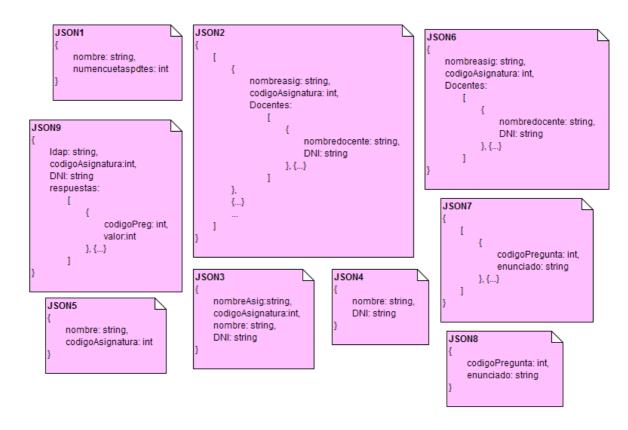


Diagrama de Clases (una de las muchas posibilidades correctas)

Estudiante

-Idap: string
-contraseña: string
-nombreyapellidos: string
-pendientes: Encuesta Pendiente[]

+comprobar(ldap : string, pwd : string) : boolean

+getnumPdtes(): int +getnombre(): string +getldap(): string +obtenerEncPdtes(): JSON2

+eliminarPdte(laAsig: Asignatura, elDoc: Docente): void

+eliminardeLista(EncPdte : EncuestaPendiente)

EncuestaPendiente

-sobreQuien : Docente -deQue : Asignatura +obtenerInfo(): JSON3

+comprobar(laAsig : Asignatura, elDoc : Docente) : boolean

Pregunta

-codigoPregunta : int -enunciado : string +getInfor() : JSON8 +getCodigo() : int

Docente -DNI : string

-DNI: string -nombreyapellidos: string +getDatosDoc(): JSON4 +getDNI(): string

Asignatura

-codAsignatura : int -nombre : string -suspreguntas : Pregunta[]

+getInfoAsig(): JSON5 +getCod(): int +getPreguntas(): JSON7

+obtenerPreg(codPreg : int) : Pregunta

CursoAcadémico

-Fechalnicio : date -FechaFin : date

+comprobarFecha(now()): boolean

Resultado

-deQue: Asignatura -sobreQuien: Docente -cuando: CursoAcadémico -preg: Pregunta -valoración: int

+comprobar(laAsig : Asignatura, elCurso : CursoAcadémico, elDoc : Docente, laPreg : Pregunta) : boolean

+sumarValor(valor: int)

+Resultado(laAsig: Asignatura, elCurso: CursoAcadémico, elDoc: Docente, laPreg: Pregunta, valor: int): Resultado

Inkesta

+identificarypdtes(idap: string, pwd): JSON1 +obtenerEncPdtes(idap: string): JSON2 +obtenerPreguntas(codAsig: int): JSON7 +almacenarRespuestas(respuestas: JSON9): void

GestorAsignaturas

-todas : Asignatura[]

+buscarAsignatura(codAsig : int) : Asignatura +obtenerPreguntas(laAsig : Asignatura) : JSON7

+obtenerPregunta(laAsig: Asignatura, codPreg:int): Pregunta

GestorResultados

-todos : Resultado[]

+almacenarResultado(laAsig: Asignatura, codAsig: int, elCurso: CursoAcadémico, elDoc: Docente, DNI: string, laPreg: Pregunta, codPreg: int, valor: int)

GestorEstudiantes

-estudiantado : Estudiante[]

+identificarypdtes(ldap: string, pwd: string): JSON1 +buscarEstudiante(ldap: string): Estudiante +obtenerEncPdtes(elEst: Estudiante): JSON2

+eliminarPdte((ldap : string, laAsig : Asignatura, codAsig : int, elDoc : Docente, DNI : string)

GestorCursos

todos : CursoAcadémico[]

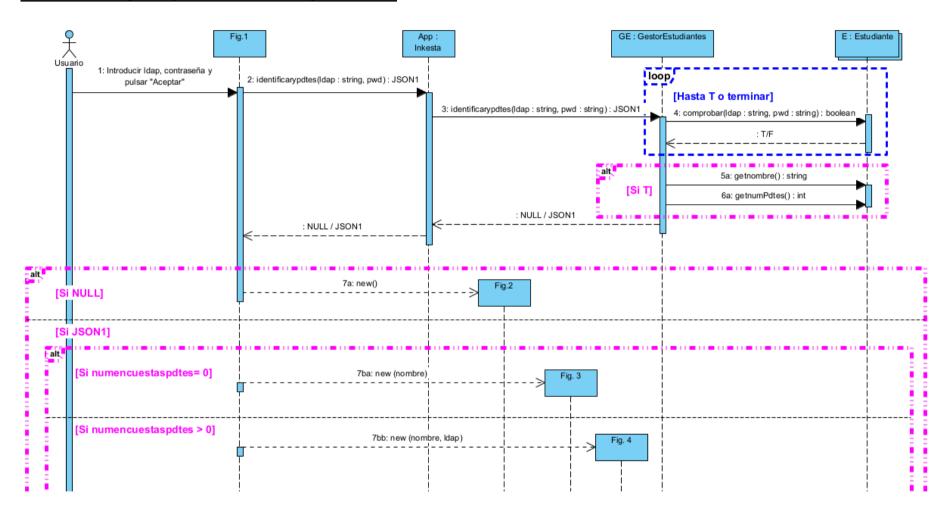
+buscarCurso(fecha : date) : CursoAcadémico

GestorDocentes

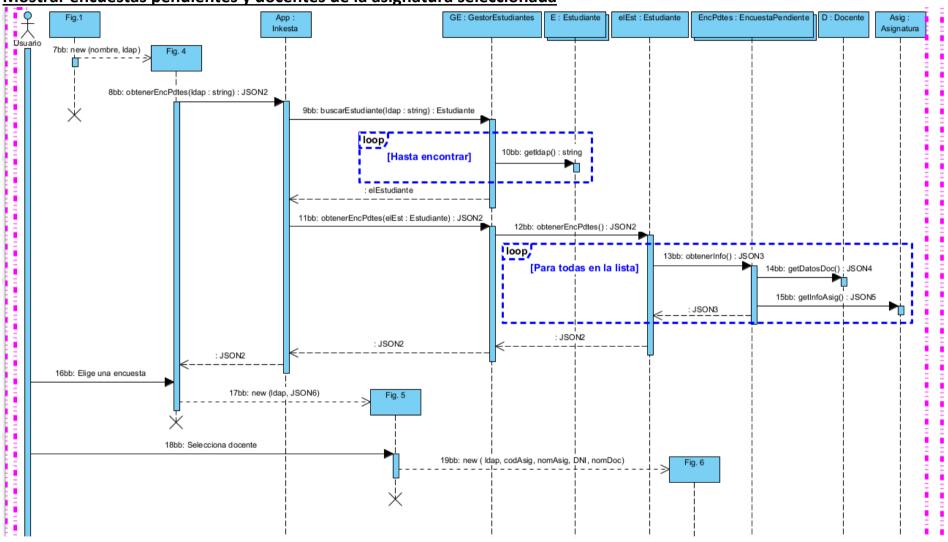
-profesorado : Docente[]

+buscarDocente(DNI: string): Docente

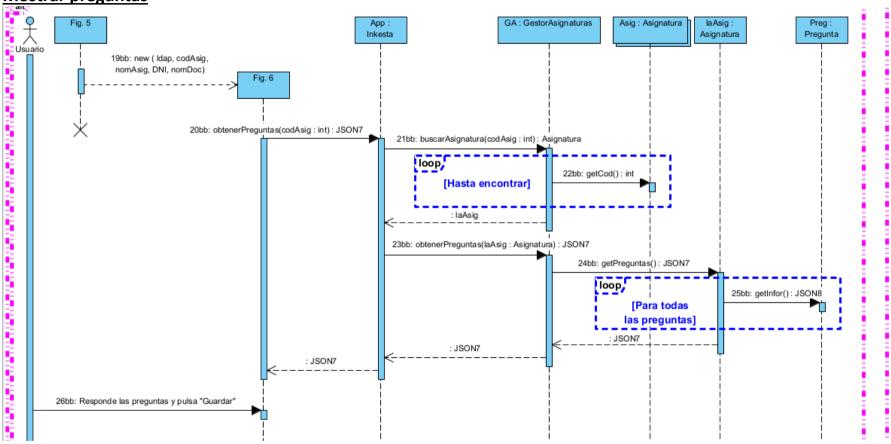
Iniciar sesión y comprobar encuestas pendientes



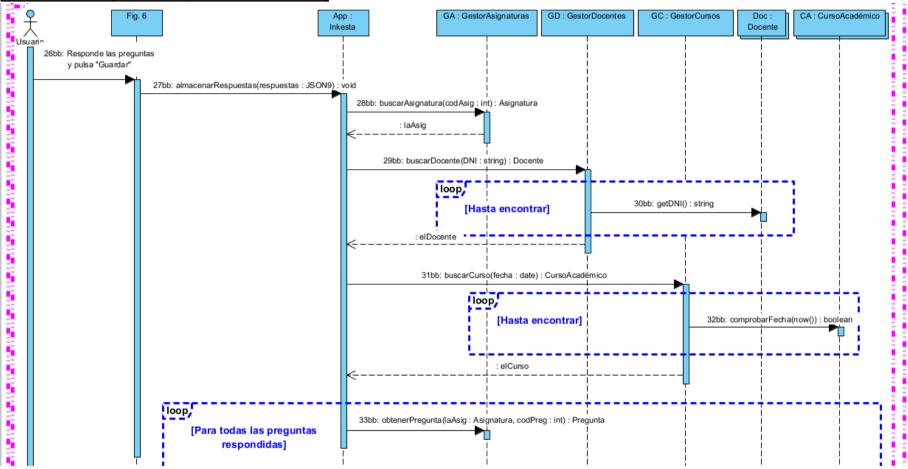
Mostrar encuestas pendientes y docentes de la asignatura seleccionada



Mostrar preguntas



Almacenar respuestas I (buscar instancias)



Almacenar Respuestas II (crear/actualizar resultado) Usuario Ioop, GE : GestorEstudiantes GA : GestorAsignaturas GR : GestorResultados la Asig: Asignatura SGRD Res · Resultado Inkesta UPDATE Resultado set Valoración = (SELECT Valoración FROM Resultado WHERE AsignaturacodigoAsignatura = %codAsig% AND 33bb: obtenerPregunta(laAsig: Asignatura, codPreg: int): Pregunta [Para todas las preguntas DocenteDNI= %DNI% AND respondidas] 34bb: obtenerPreg(codPreg:int): Pregunta PreguntacodigoPregunta= %CodPreg% AND CursoAcademicoFechalnicio = (SELECT Fechalnicio FROM CursoAcademico loop. WHERE %now()% BETWEEN Fechalnicio AND FechaFin)) 35bb: getCodigo(): int %valor% [Hasta encontrar] NHERE AsignaturacodigoAsignatura = %codAsig% AND : laPreg DocenteDNI= %DNI% AND laPreg PreguntacodigoPregunta= %CodPreg% AND CursoAcademicoFechalnicio = (SELECT Fechalnicio FROM CursoAcademico 36bb: almacenarResultado(laAsig: Asignatura, codAsig: int, elCurso: CursoAcadémico, elDoc: Docente, DNI: string, WHERE %now()% BETWEEN Fechalnicio AND FechaFin) la Preg: Pregunta, codPreg: int, valor: int) loop, 37bb: comprobar(laAsig : Asignatura, elCurso CursoAcadémico, elDoc : Docente, laPreg : Pregunta) : boolean [Hasta T o terminar] [Si T] 38bba: sumarValor(valor: int) 39bba: execSQL (sql1) [Si no existe] 38bbb; Resultado(laAsig : Asignatura, elCurso : CursoAcadémico, elDoc : Docente, la Preg: Pregunta, valor: int): Resultado 39bbb: execSQL (sql2) INSERT INTO Resultado (AsignaturacodigoAsignatura, DocenteDNI, PreguntacodigoPregunta, CursoAcademicoFechalnicio, Valoración) VALUES

%codAsig%, %DNI%, %CodPreg%, (SELECT Fechalnicio FROM CursoAcademico WHERE %now()% BETWEEN Fechalnicio AND FechaFin), %valor%)

40bb: eliminarPdte(ldap: string, laAsig: Asignatura, codAsig: int, elDoc: Docente, DNI: string)

Almacenar Respuestas III (eliminar encuesta de las pendientes)

