|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | MainWindowForm |
| **Operación** | **RevisarEvaluacion()** |

Pseudocódigo

void ActualizarEvaluacion(){

esSup = unControladorSesiones.ValidarEsSupervisor();

if(esSup){

unActualizarEvaluacionForm.Mostrar();

this.Ocultar();

}

else{

MostrarError("No Autorizado");

}

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | MainWindowForm |
| **Operación** | Ocultar () |

Pseudocódigo

void Ocultar(){

setVisible(false);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ActualizarEvaluacionForm |
| **Operación** | **Mostrar()** |

Pseudocódigo

void Mostrar(){

listaEvs = unContEvaluacion.ObtenerEvaluacionesReclamadas();

MostrarListaEvaluaciones(listaEvs);

setVisible(true);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ActualizarEvaluacionForm |
| **Operación** | **MostrarListaEvaluaciones(listaEvaluaciones)** |

Pseudocódigo

void MostrarListaEvaluaciones(listaEvaluaciones){

foreach(ev in listaEvaluaciones){

comboEvs.add(ev) //Componente Evaluaciones

}

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ActualizarEvaluacionForm |
| **Operación** | **ObtenerEvaluacion(idEvaluacion)** |

Pseudocódigo

void ObtenerEvaluacion(idEv){

// PuntosEv será un arrayList< HashMap<String, String> >

// que te enviará la controladora

puntosEv = unControladorEvaluacion.ObtenerEvaluacionReclamada(idEv);

MostrarEvaluacion(puntosEv);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ActualizarEvaluacionForm |
| **Operación** | **MostrarEvaluacion(PuntosEv)** |

Pseudocódigo

void MostrarEvaluacion(ptos){

// ptos será un arrayList< HashMap<String, String> >

for(i = 0; i < ptos.size(); i++){

id = ptos[i].get("id");

pto = ptos[i].get("pto");

resp = ptos[i].get("resp");

coment = ptos[i].get("coment");

// Aquí agregas un punto de evaluación al panel.

pnlPuntos.addPunto(id, preg, resp, coment);

}

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | RevisarEvaluacionForm |
| **Operación** | **ActualizarEvaluacion(idEvaluacion, recProcedentes)** |

Pseudocódigo

void GuardarRevision(idEval, recProcedentes){

// recProcedentes serán aquellas respuestas que el supervisor

// marcó como procedentes, con su correspondiente respuesta:

// HashMap<Integer, boolean> (IdPunto, cumplePunto).

// Las respuestas que no sean marcadas como procedentes no se incluyen

// en recProcedentes.

// idEval será el id que seleccionó el usuario en el comboBox

unControladorEvaluacion.ActualizarEvaluacion(idEv, recProcedentes);

mostrar("Evaluación Actualizada");

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ControladorEvaluacion |
| **Operación** | **ObtenerEvaluacionesReclamadas() : HashMap<Integer, String>** |

Pseudocódigo

HashMap ObtenerEvaluacionesReclamadas(){

usuarioActual = unControladorSesiones.getUsuarioActual();

unSup.copiar(usuarioActual);

listaEv = unSup.ObtenerEvaluacionesReclamadas();

foreach(ev in listaEv){

id = ev.getId();

// La evaluación se formateará: "id [ dd-mm-yyyy ]". EJ:

// 1 [ 01-11-2013 ]

// En java busca la manera de formatear la fecha.

evFormateada = ev.getId + " [ " + ev.getFechaCreada() +

" | " + ev.getRepresentante() + " ]";

evaluaciones.put(id, evFormateada);

}

return evaluaciones;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ControladorEvaluacion |
| **Operación** | **ObtenerEvaluacionReclamada(idEvaluacion) : ArrayList** |

Pseudocódigo

ArrayList ObtenerEvaluacionReclamada(idEvaluacion){

ptosEv = ArrayList< HashMap<String, String> >

unaEv = unEvaluacionDA.LeerEvaluacionReclamada(idEvaluacion);

cantResp = unaEv.getContarRespuestas();

for(i = 0; i < cantResp; ++i){

id = unaEv.getIdRespuesta(i);

resp = unaEv.getRespuesta(id) ? "Si" : "No";

pto = unaEv.getPuntoEvaluacion(id);

coment = unaEv.getComentario(id);

unPtoEv = HashMap<String, String>

unPtoEv.put("id", id.toString());

unPtoEv.put("pto", pto);

unPtoEv.put("resp", resp);

unPtoEv.put("coment", coment);

ptosEv.add(unPtoEv);

}

return ptosEv;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ControladorEvaluacion |
| **Operación** | **ActualizarEvaluacion(idEvaluacion, recProcedentes)** |

Pseudocódigo

String ActualizarEvaluacion(idEvaluacion, recProcedentes){

unaPlant = PlantillaDA.LeerSoloPlantilla(idEvaluacion);

unaEv = Evaluacion(id, unaPlant);

foreach(resp in recProcedentes){

num = resp.getKey();

cumple = resp.getValue(); // booleano

unaEv.CrearRespuesta(num, cumple);

}

unaEvaluacionDA.ActualizarEvaluacion(unaEv);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ControladorSesiones |
| **Operación** | ValidarEsSupervisor (): boolean |

Pseudocódigo

boolean ValidarEsSupervisor(){

return usuarioActual.getRol() == "Supervisor";

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | ControladorSesiones |
| **Operación** | getUsuarioActual(): Usuario |

Pseudocódigo

Usuario getUsuarioActual(){

return usuarioActual();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Supervisor |
| **Operación** | **ObtenerEvaluacionesReclamadas(): ArrayList** |

Pseudocódigo

ArrayList ObtenerListaEvaluaciones(){

listEvals = EvaluacionDA.LeerEvaluacionesReclamadas(idSupervisor);

return listEvals;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | EvaluacionDA |
| **Operación** | **LeerEvaluacionesReclamadas(idSupervisor): ArrayList** |

Pseudocódigo

ArrayList LeerEvaluacionesReclamadas(idSupervisor){

commSQL.Texto = "SELECT id\_evaluacion, fecha\_creada, " +

" id\_representante, nombre " +

" FROM evaluacion ev, usuario usr " +

" WHERE id\_usuario = id\_representante " +

" AND id\_estado\_evaluacion = 2 " +

" AND id\_representante in " +

" (SELECT id\_usuario " +

" FROM usuario " +

" WHERE id\_supervisor = ?) ";

parametros = [idSupervisor];

commSQL.setParametros(parametros);

conexionDB.Abrir();

resultados = commSQL.Ejecutar();

while(resultados.next()){

id = resultados["id\_evaluacion"];

fecha = resultados["fecha\_creada"];

idRep = resultados["id\_representante"];

nomRep = resultados["nombre"];

unRep = Representante(idRep, nomRep);

unaEv = Evaluacion(id, fecha, unRep);

evals.add(unaEv);

}

conexionDB.Cerrar();

return evals;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | EvaluacionDA |
| **Operación** | **LeerEvaluacionReclamada(idEvaluacion): Evaluacion** |

Pseudocódigo

Evaluacion LeerEvaluacionReclamada(idEvaluacion){

unaPlant = unPlantillaDA.LeerPlantilla(idEvaluacion);

unaEv = Evaluacion(idEvaluacion, unaPlant);

unRespuestaDA.CargarRespuestasReclamadas(unaEv);

return unaEv;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | EvaluacionDA |
| **Operación** | **ActualizarEvaluacion (unaEv)** |

Pseudocódigo

void ActualizarEvaluacion(unaEv){

cantResp = unaEv.contarRespuestas();

idEv = unaEv.getId();

idPlant = unaEv.getIdPlantilla();

for(i = 0; i < cantResp; i++){

idResp = unaEv.getIdRespuesta(i);

cumple = unaEv.getCumplePunto(idResp);

unRespuestaDA.cambiarRespuesta(idEv, idPlant, idResp, cumplePunto);

}

parametros = [idEv];

commSQL.setParametros(parametros);

conexionDB.Abrir();

commSQL.Texto = "Update ... " // Evaluacion

+ "Set id\_estado\_evaluacion = 3 "

+ "Where id\_evaluacion = ?";

conexionDB.Cerrar();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | RespuestaDA |
| **Operación** | **CargarRespuestasReclamadas(unaEv)** |

Pseudocódigo

idEv = unaEv.getId();

idPlant = unaEv.getIdPlantilla();

commSQL.Texto = "Select ... " // respuestas

+"where id\_evaluacion = ? "

+"and id\_plantilla = ? "

+"and es\_reclamada = 1 ";

parametros = [idEv, idPlant];

commSQL.setParametros(parametros);

conexionDB.Abrir();

resultados = commSQL.Ejecutar();

while(resultados.next()){

num = resultados["id\_punto\_evaluacion"];

// cumple es true si es igual a 1

cumple = resultados["cumple\_punto"] == 1;

comentario = resultados["comentario"];

unaEv.CrearRespuesta(num, cumple, comentario);

}

conexionDB.Cerrar();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | RespuestaDA |
| **Operación** | CambiarRespuesta(idEvaluacion, idPlantilla, idResp, cumplePunto) |

Pseudocódigo

void CambiarRespuesta (idEv, idPlant, idResp, cumplePunto) {

commSQL.Texto = "Update ... " // Respuesta

+ "Set es\_reclamada = 0, " // 1 = true

+ "cumple\_punto = ? "

+ "where "

+ "id\_evaluacion = ?";

+ "and id\_plantilla = ?";

+ "and id\_punto\_evaluacion = ?";

cumple = !cumplePunto;

parametros = [cumple, idEv, idPlant, idResp]

commSQL.setParametros(parametros);

conexionDB.Abrir();

commSQL.Ejecutar();

conexionDB.Cerrar();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | **Evaluacion (id, fechaCreada, unRep)** |

Pseudocódigo

Evaluacion (id, fechaCreada){

this.id = id;

this.fechaCreada = fechaCreada;

this.representante = unRep;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | Evaluacion (id, unaPlant) |

Pseudocódigo

Evaluacion (id, unaPlant){

this.id = id;

this.plantilla = unaPlant;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | getId(): int |

Pseudocódigo

int getId(){

return idEvaluacion;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | getIdPlantilla(): int |

Pseudocódigo

int getIdPlantilla(){

return this.plantilla.getId();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | **CrearRespuesta(num: Int, cumple: bool, comentario: String)** |

Pseudocódigo

void CrearRespuesta(num, cumple){

resp = Respuesta(num, cumple, comentario);

respuestas.add(resp);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | **CrearRespuesta(num: Int, cumple: bool)** |

Pseudocódigo

void CrearRespuesta(num, cumple){

resp = Respuesta(num, cumple);

respuestas.add(resp);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | **contarRespuestas(): int** |

Pseudocódigo

int contarRespuestas(){

return this.respuestas.size();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | getIdRespuesta(index): int |

Pseudocódigo

int getIdRespuesta(index){

return this.respuestas.at(index).getId();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Evaluacion |
| **Operación** | getRespuesta(idResp):boolean |

Pseudocódigo

boolean getRespuesta(idResp){

foreach(resp in this.respuestas){

if(resp.getId){

cumple = resp.getCumplePunto();

}

}

return cumple;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Representante |
| **Operación** | Representante (id, nombre) |

Pseudocódigo

Representante(id, nombre){

base(id, nombre);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Respuesta |
| **Operación** | **Respuesta (num, cumple, comentario)** |

Pseudocódigo

Respuesta (num, cumple, comentario){

this.idPuntoEvaluacion = num;

this.cumplePunto = cumple; // booleano

this.comentario = comentario;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Respuesta |
| **Operación** | Respuesta(num: Int, cumple: bool) |

Pseudocódigo

Respuesta(num, cumple){

this.idPuntoEvaluacion = num;

this.cumplePunto = cumple; // booleano

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Respuesta |
| **Operación** | getId(): int |

Pseudocódigo

int getId(){

return this.idPuntoEvaluacion;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Respuesta |
| **Operación** | getCumplePunto(): boolean |

Pseudocódigo

String getComentario (){

return this.cumplePunto;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | PlantillaDA |
| **Operación** | LeerPlantilla(idEvaluacion) |

Pseudocódigo

Plantilla LeerPlantilla(idEv){

commSQL.Texto = "Select id\_plantilla from evaluacion "

+ "where id\_evaluacion = ?"

parametros = [idEv];

commSQL.setParametros(parametros);

conexionDB.Abrir();

resultados = commSQL.Ejecutar();

id = resultados["id\_plantilla"];

unaPlant = Plantilla(id);

unPuntoEvaluacionDA.LeerPuntosEvaluacion(unaPlant);

conexionDB.Cerrar();

return unaPlant;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | PlantillaDA |
| **Operación** | LeerSoloPlantilla(idEvaluacion) |

Pseudocódigo

Plantilla LeerSoloPlantilla(idEvaluacion)

commSQL.Texto = "Select id\_plantilla from evaluacion "

+ "where id\_evaluacion = ?"

parametros = [idEv];

commSQL.setParametros(parametros);

conexionDB.Abrir();

resultados = commSQL.Ejecutar();

id = resultados["id\_plantilla"];

unaPlant = Plantilla(id);

conexionDB.Cerrar();

return unaPlant;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | PuntoEvaluacionDA |
| **Operación** | LeerPuntosEvaluacion(unaPlant) |

Pseudocódigo

LeerPuntosEvaluacion(unaPlant){

commSQL.Texto = "Select ... " // puntos\_evaluacion

+"where id\_plantilla= ?";

parametros = [plantillaActual.getId()];

commSQL.setParametros(parámetros);

conexionDB.Abrir();

resultados = commSQL.Ejecutar();

while(resultados.next()){

num = resultados["id"];

ptoEv = resultados["descripcion"];

unaPlant.CrearPuntoEvaluacion(num, ptoEv)

}

conexionDB.Cerrar();

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Plantilla |
| **Operación** | Plantilla(idPlantilla) |

Pseudocódigo

Plantilla(idPlantilla){

this.plantilaId = idPlantilla;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Plantilla |
| **Operación** | getId(): int |

Pseudocódigo

int getId(){

return this.plantilaId;

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | Plantilla |
| **Operación** | CrearPuntoEvaluacion(num, ptoEv) |

Pseudocódigo

void CrearPuntoEvaluacion(num, ptoEv){

unPtoEv = PuntoEvaluacion(num, ptoEv);

this.puntosEvaluacion.add(unPtoEv);

}

|  |  |
| --- | --- |
| **Caso de Uso** | CU005: Actualizar Evaluación |
| **Clase** | PuntoEvaluacion |
| **Operación** | PuntoEvaluacion(num, ptoEv) |

Pseudocódigo

PuntoEvaluacion(num, ptoEv){

this.puntoEvaluacionId = num;

this.descripcion = ptoEv;

}