**Universidad Tecnológica Nacional**

**Facultad Regional Gral. Pacheco**



TÉCNICO UNIVERSITARIO EN PROGRAMACIÓN

LABORATORIO III – T.N

EXAMEN FINAL

Docente: Angel Simon

Alumno: Pedro Tonello

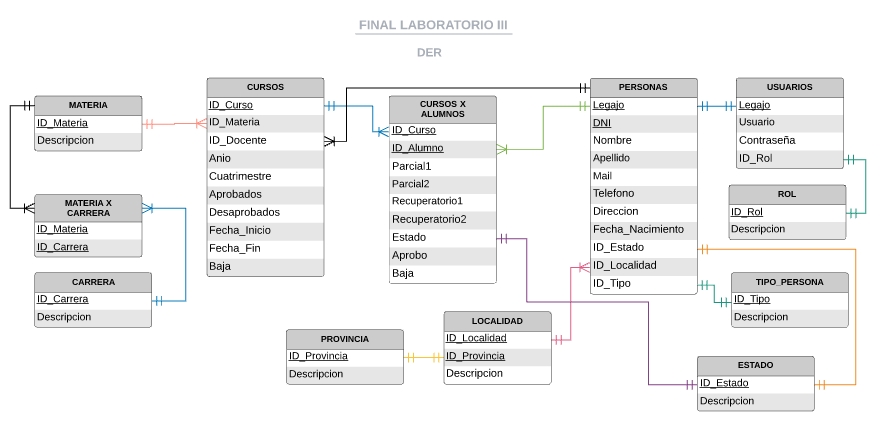
El sistema presentado es de gestión educativa con dos perfiles distintos (Administrador y profesor).

1. Diagrama entidad relación del sistema:

Link para acceder:

<https://app.lucidchart.com/invitations/accept/e379072a-ac3d-4623-b7e7-5612fba460b3>

Imagen:



1. En el perfil del Administrador, a través de una vista hice un listado con todos los datos de los profesores. La idea de hacer esta vista es poder simplificar la complejidad y longitud de la consulta. Se puede ver la invocación desde la app en “**servletListarDatosProfesores.java”** esta el back, el front esta en “**datosProfesores.jsp**”.

El sistema esta dividido en capas, por lo tanto los métodos se encuentran en **DaoImpl / PersonaDaoImpl.java** y las querys se encuentran en **dao.querys/queriesPersona.java**. Las querys están en enum para tener separados y prolijos los metodos. A continuación dejo el código de la vista, de todas formas se puede ver desde la base de datos también.

Código de la vista:

CREATE

ALGORITHM = UNDEFINED

DEFINER = `root`@`localhost`

SQL SECURITY DEFINER

VIEW `vw\_profesores` AS

SELECT

`p`.`LEGAJO` AS `LEGAJO`,

`p`.`NOMBRE` AS `NOMBRE`,

`p`.`APELLIDO` AS `APELLIDO`,

`p`.`MAIL` AS `MAIL`,

`p`.`TELEFONO` AS `TELEFONO`,

`p`.`DNI` AS `DNI`,

`p`.`DIRECCION` AS `DIRECCION`,

`l`.`DESCRIPCION` AS `LOCALIDAD`,

`pr`.`DESCRIPCION` AS `PROVINCIA`,

(YEAR(CURDATE()) - YEAR(`p`.`FECHA\_NACIMIENTO`)) AS `EDAD`,

`u`.`USUARIO` AS `USUARIO`,

`u`.`CONTRASEÑA` AS `CONTRASEÑA`,

(SELECT

COUNT(`c`.`ID\_CURSO`)

FROM

`cursos` `c`

WHERE

(`c`.`ID\_DOCENTE` = `p`.`LEGAJO`)) AS `CURSOS`

FROM

(((`personas` `p`

JOIN `usuario` `u` ON ((`p`.`LEGAJO` = `u`.`LEGAJO`)))

JOIN `localidad` `l` ON ((`l`.`ID\_LOCALIDAD` = `p`.`ID\_LOCALIDAD`)))

JOIN `provincia` `pr` ON ((`l`.`ID\_PROVINCIA` = `pr`.`ID\_PROVINCIA`)))

WHERE

((`p`.`ID\_TIPO` = 1)

AND (`p`.`ID\_ESTADO` = 1))

1. También en el perfil del Administrador, realice un procedimiento almacenado que va ser el encargado de traer una consulta donde en ella se va generar un reporte.

El reporte es filtrado por dos fechas, fecha de inicio y fecha de fin.

En el rango de esas fechas se va obtener los cursos asociados con el nombre de la materia, nombre del profesor a cargo, cantidad de alumnos, porcentaje de aprobados y desaprobados.

Se puede ver la invocación desde la app en “**servletReporte.java**” esta el back, el front esta en “**AdministradorReporte.jsp**”.

El sistema esta dividido en capas, por lo tanto los métodos se encuentran en **DaoImpl/ReporteDaoImpl.java** y las querys se encuentran en **dao.querys/queriesCurso.java**. Las querys están en enum para tener separados y prolijos los metodos. A continuación dejo el código de la vista, de todas formas se puede ver desde la base de datos también.

Código del procedimiento almacenado:

CREATE DEFINER=`root`@`localhost` PROCEDURE `sp\_reporte`(IN FECHA\_I DATE,IN FECHA\_F DATE)

SELECT COUNT(CU.ID\_ALUMNO) AS CANTIDAD, C.APROBADOS, C.DESAPROBADOS, C.ANIO, C.CUATRIMESTRE, P.APELLIDO, M.DESCRIPCION

FROM CURSOS AS C

INNER JOIN MATERIA AS M ON M.ID\_MATERIA = C.ID\_MATERIA

INNER JOIN PERSONAS AS P ON P.LEGAJO = C.ID\_DOCENTE

INNER JOIN CURSOSXALUMNOS AS CU ON C.ID\_CURSO = CU.ID\_CURSO

WHERE C.FECHA\_INICIO >= FECHA\_I

AND C.FECHA\_FIN <= FECHA\_F

GROUP BY C.ID\_CURSO

1. Realice otro procedimiento almacenado pero este realiza una acción en la base de datos.

En el sistema se puede dar de alta profesores y alumnos desde el perfil del Administrador.

Solamente si es profesor a la hora de darlo de alta, se le debe generar un usuario y contraseña para que luego el profesor pueda loguearse en el sistema. Por lo tanto, el procedimiento almacenado que hice, inserta a la persona sin importar el tipo de persona que sea (profesor o alumno) y luego con un IF verifico el tipo, y le genero un usuario y contraseña. Anteriormente, en el sistema esto lo hacia con diferentes métodos pero de esta manera quedo automatizado con el insert del profesor.

Se puede ver la invocación desde la app en “**servletPersonas.java**” esta el back, el front esta en “**personas.jsp**”.

El sistema esta dividido en capas, por lo tanto los métodos se encuentran en **DaoImpl / PersonaDaoImpl.java** y las querys se encuentran en **dao.querys/queriesPersona.java**. Las querys están en enum para tener separados y prolijos los metodos. A continuación dejo el código de la vista, de todas formas se puede ver desde la base de datos también.

Código del procedimiento almacenado:

CREATE DEFINER=`root`@`localhost` PROCEDURE `pr\_insert\_user\_profesor`(IN DNIS INT(11) , IN NOMBRES VARCHAR(50), IN APELLIDOS VARCHAR(50), IN MAILS VARCHAR(50), IN TELEFONOS INT(11), IN ID\_TIPOS INT(11), IN DIRECCIONS VARCHAR(50), IN ID\_LOCALIDADS INT(11), IN FECHA\_NACIMIENTOS DATE)

begin

DECLARE USERNAME VARCHAR(50);

DECLARE LEGAJOS INT;

set USERNAME = DNIS;

INSERT INTO PERSONAS(DNI, NOMBRE, APELLIDO, MAIL, TELEFONO, ID\_TIPO, DIRECCION, ID\_LOCALIDAD, FECHA\_NACIMIENTO, ID\_ESTADO) VALUES

(DNIS,NOMBRES,APELLIDOS,MAILS,TELEFONOS,ID\_TIPOS,DIRECCIONS,ID\_LOCALIDADS,FECHA\_NACIMIENTOS,1);

IF ID\_TIPOS = 1 then

SET LEGAJOS = (SELECT LEGAJO FROM PERSONAS ORDER BY LEGAJO DESC LIMIT 1);

INSERT INTO USUARIO (LEGAJO,USUARIO, CONTRASEÑA, ID\_ROL) VALUES (LEGAJOS,USERNAME,DNIS,2);

end IF;

end

1. Por ultimo, realice un trigger que la función que cumple es que cada vez que alguien haga un update de su usuario o contraseña voy a estar guardándome sus credenciales antiguas en otra tabla automáticamente para llevar un control.

El trigger se encuentra en la tabla Usuario.

Código del trigger:

delimiter //

create trigger tr\_before\_update\_user\_pass

before update

on usuario

for each row

begin

insert into user\_pass\_old (usuario, contraseña, legajo, id\_rol,fecha) values (old.usuario, old.contraseña, old.legajo, old.id\_rol, NOW());

end //

delimiter ;