

## Redictado Proyecto de Software Trabajo para la Facultad de Informática - Proceso de Tesinas de Grado

En el presente ciclo lectivo trabajaremos en el desarrollo de un prototipo para la Facultad de Informática. La propuesta de trabajo surge de una necesidad de actualizar el sistema actual para la presentación y seguimiento de tesinas de grado. A lo largo de la cursada se desarrollará un prototipo funcional, que es un modelo a escala de la realidad (sin la funcionalidad total), el cual se irá enriqueciendo hasta llegar al producto final deseado.

El trabajo se llevará a cabo en **dos etapas**, cada etapa finalizará con la entrega de un prototipo funcional que deberá presentarse en funcionamiento desde el servidor provisto por la cátedra.

### Etapa 1 - v 0.1.0

Fecha de entrega: 03/05 a las 8:00 (cierre del servidor)

Fecha de reentrega: 17/05 a las 8:00 (cierre del servidor)

### 1.1 Manejo de sesiones

Deberá implementarse un manejo de sesiones adecuado, verificando la sesión y roles cuando corresponda. Para cada módulo se indicará si requiere autenticación o no y roles necesarios.

### 1.2 Módulo de usuarios

#### Parte pública

La idea es que cada alumno pueda registrarse para luego poder iniciar el proceso de presentación de su propuesta de tesina de grado. Para ello es necesario tener un módulo de registro de manera pública, medio por el cual el alumno solicita el alta de su usuario.

**Importante: por el momento esta acción no hace falta que se implemente, pero es bueno tener en cuenta que va a existir y que es aquí en donde arranca el proceso para los alumnos.**

#### Parte administrativa

Desarrollar el **módulo usuarios** que deberá contemplar al menos la siguiente funcionalidad:

- CRUD<sup>1</sup> de usuarios, validar la existencia de un usuario, es decir, no pueden existir dos usuarios con el mismo nombre de usuario.

---

<sup>1</sup> Create Read Update Delete: creación, lectura (listado), actualización y borrado de usuarios (ABM).

El resultado de la búsqueda debe estar paginado en base a la configuración del sistema.

- Activar/Bloquear usuario.
- Asignar o desasignar roles de un usuario (pueden ser varios).

Esta funcionalidad debe estar disponible para usuarios que cuenten con el rol de: “Personal administrativo” y “Administrador”.

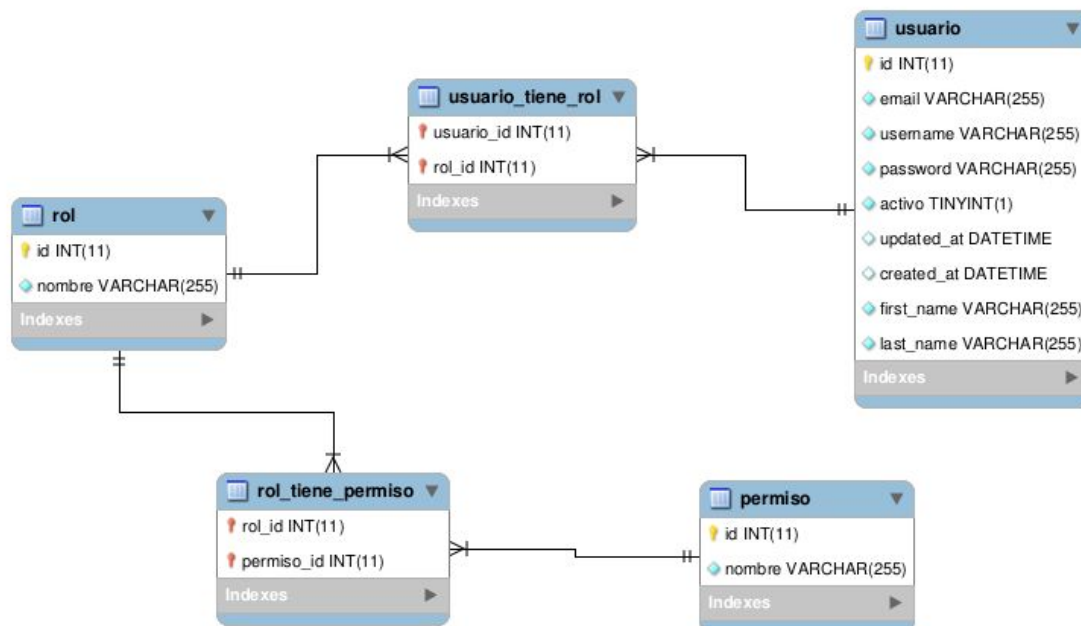
**Para esta etapa, tampoco será obligatorio desarrollar el CRUD de los roles y los permisos**, podrán administrarse desde la base de datos.

Considerar que un usuario podrá tener más de un rol, y para cada rol se pueden configurar varios permisos. Los permisos necesarios asociados a cada rol deberán deducirse del enunciado, ante la duda **consulte a su ayudante**.

El nombre de los permisos deberá respetar el patrón **modulo\_accion**, por ejemplo, el módulo de tesinas deberá contemplar los siguientes permisos:

- tesina\_index: permite acceder al index (listado) del módulo.
- tesina\_new: permite cargar un paciente.
- tesina\_destroy: permite borrar un paciente.
- tesina\_update: permite actualizar un paciente.
- tesina\_show: permite visualizar un paciente.

## Modelo de Usuarios, roles y permisos



## 1.3 Módulo de Validaciones de Usuario

Este módulo permitirá al **Personal administrativo** de la facultad realizar una serie de validaciones previas que los **usuarios** (alumnos) deben pasar satisfactoriamente para poder arrancar el proceso de presentación de la propuesta de tesina.

Se debe visualizar un listado con los usuarios que tengan pendiente las validaciones necesarias antes de poder presentar la propuesta de tesina.

El Personal Administrativo podrá validar los datos correspondientes a cada alumno y cambiar su estado.

El sistema debe permitir al **Personal Administrativo** autorizar o rechazar la validación, permitiendo también dejar un comentario que indique los motivos de cualquiera de los resultados.

Las validaciones que se deberán tener en cuenta para esta entrega son las siguientes:

- El usuario es un alumno de la facultad
- El usuario cumple las condiciones para poder realizar la presentación de la tesis

## 1.4 Módulo de carga de Tesina

Una vez que el **usuario** (alumno) haya pasado las validaciones ya mencionadas, tendrá la opción de completar el formulario de carga para la presentación de la tesis.

Los datos necesarios la presentación son los enunciados en el documento: <http://info.unlp.edu.ar/wp-content/uploads/2019/02/Ordenanza-307-Tesinas-de-Grado-definitiva.pdf>

Todos los datos que se solicitan son obligatorios. Para el caso de tipo de clasificación de la tesina, se deben dar sólo las opciones que se visualizan en el formulario (Form1) de la Ordenanza 307.

**Formulario 1. Presentación de la Propuesta**

Director					
Codirector					
Asesor Profesional					
Alumnos	Nro. Alumno	Apellido y Nombres		Dirección, teléfono y e-mail	
Clasificación	Investigación Teórica	Investigación Aplicada	Desarrollo Tecnológico	Transferencia Tecnológica	Temática de RS
Plazo de Ejecución					

De esta forma los datos necesarios para este módulo se podrían dividir en dos partes:

- Título de la tesina (string)

- Fecha (date)
- Objetivo (string)
- Motivación (string)
- Desarrollos propuestos (string)
- Resultados esperados (string)

Según el formulario presentado:

- Director (string)
- Codirector (string)
- Asesor Profesional (string)
- Alumnos (debería estar ligado a alguna entidad Alumno)
  - Número de alumno (integer)
  - Nombre (string)
  - Apellido (string)
  - Direccion (string)
  - Telefono (integer)
  - Mail (string)
- Clasificación (select de Investigación teórica, Investigación aplicada, Desarrollo Tecnológico, Transferencia tecnológica y Temática de RS)
- Plazo de ejecución (en meses, integer)

Validar correctamente todos estos datos de entrada a la hora de guardarlos en la base de datos. De la misma forma limpiar las entradas de valores no permitidos como tags html.

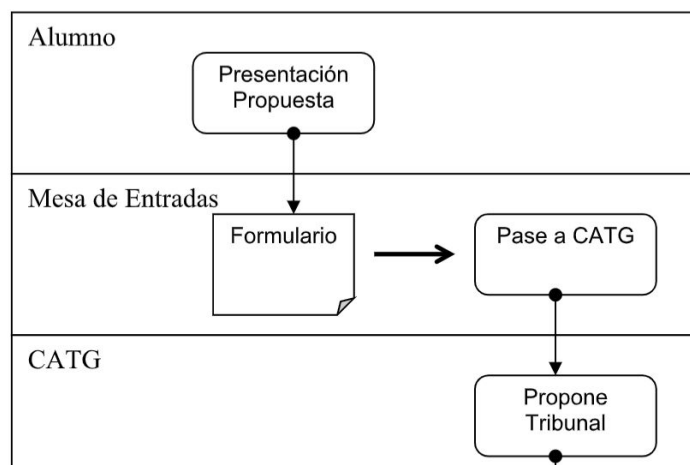
Personas con el rol de **Personal Administrativo** podran ver todas las presentaciones en curso y aceptarlas o rechazarlas según considere, aclarando el motivo correspondiente en caso de rechazo.

El **alumno** siempre podrá ver el estado de su solicitud y en caso de ser rechazada podrá visualizar los motivos del rechazo incluso volver a mandar la propuesta si por ejemplo el motivo del rechazo tiene que ver con algún dato incorrecto en la presentación.

Estados posibles de la presentación:

- **Propuesta entregada:** cuando el alumno completa el formulario y lo manda. Es el estado inicial.
- **Propuesta rechazada:** la solicitud es rechazada. Recordar que tiene que existir un comentario que describa los motivos del rechazo. Esto da la posibilidad al alumno que realice la presentación nuevamente con nuevos datos, por ejemplo, si es que ese fue el motivo del rechazo.
- **Propuesta aceptada:** la presentación de la tesina es aceptada y el proceso continúa.

El siguiente diagrama de flujos muestra el proceso mencionado hasta el momento:



## Consideraciones generales

- El prototipo debe ser desarrollado utilizando PHP, HTML5, CSS3 y MySQL, y **respetando el modelo en capas MVC**. Debe utilizar PDO como mecanismo de abstracción de bases de datos, teniendo la posibilidad de utilizar Doctrine como ORM<sup>2</sup>. Debe tener en cuenta los conceptos de semántica web proporcionada por HTML5 siempre y cuando sea posible con una correcta utilización de las etiquetas del lenguaje.
- El trabajo será evaluado **desde el servidor de la cátedra** que cada grupo deberá gestionar mediante Git. **NO se aceptarán entregas que no estén realizadas en tiempo y forma en el servidor provisto por la cátedra.**
- Cada grupo tendrá un ayudante a cargo, quien evaluará el progreso y la participación de cada integrante mediante las consultas presenciales y el seguimiento mediante GitLab.
- Toda vista (**HTML5** y **CSS3**) debe validar contra las especificaciones de la W3C (<http://validator.w3.org/> y <https://jigsaw.w3.org/css-validator/> respectivamente). En esta oportunidad puede utilizar **Bootstrap** u otro framework similar para realizar el maquetado.
- **El uso de Twig como motor de plantillas es obligatorio.**
- **NO** puede utilizar un framework PHP para el desarrollo.
- Para esta etapa, la entrega y aprobación son obligatorias. Todos los integrantes deben presentarse a la defensa.
- Tener en cuenta que el servidor de la cátedra utiliza los siguientes servicios y versiones:
  - Servidor Web: Apache 2.4.29
  - Servidor de Base de Datos: MariaDB 10.1.34
  - Intérprete PHP: Php 7.2
- **Importante:**
  - El proyecto podrá ser realizado de modo individual o en grupos de dos integrantes (será responsabilidad de los estudiantes la conformación de los

<sup>2</sup> Object-Relational Mapping

equipos de trabajo). Todos los estudiantes cumplirán con la totalidad de la consigna, sin excepciones.

- La repetición de código podría implicar la desaprobación de la entrega.