

Programación Web 3

UNLaM - Tecnicatura en Desarrollo Web

Trabajo Práctico

1. Objetivo

Este documento describe el alcance funcional y los requisitos técnicos del trabajo práctico que los alumnos de la materia Programación Web III deberán aprobar a fin de estar en condición de aprobar/promocionar la materia.

2. Equipo

El equipo para realizar el trabajo práctico deberá ser de **4 alumnos**.

3. Objetivo del Proyecto

El proyecto a realizar consiste en un aplicación para organización de tareas. Este tipo de aplicaciones es conocido como **organizadores de tareas** o **administradores de tareas**. Ejemplos:

<https://www.wunderlist.com/en/>

<https://app.asana.com>

Las principales funcionalidades son:

- Registro de Usuario.
- Login de Usuario.
- Creación, listado y detalle de Tareas.
- Creación y listado de Carpetas.
- Creación de comentarios en Tarea.
- Creación de adjunto en Tarea.

4. Especificación Funcional

Listado de Páginas:

Sin estar logueado

1. [Página de inicio \(/home/index y /\)](/home/index).
 - a. Describe el objetivo del sitio incentivando a registrarse.
 - b. Contiene un cuadro de **login** y un link de **registración**.
2. [Página de login \(/home/login\)](/home/login).
3. [Página de registraci3n \(/home/registracion\)](/home/registracion).

Es necesario estar **logueado (*)**

5. [Página de inicio \(/home/index y /\)](/home/index).
 - a. [Mis carpetas](#)
 - b. [Mis tareas](#)
 - c. [Menú](#)
6. Carpetas
 - a. [Mis Carpetas](#)
 - i. Listado de carpetas del usuario (</carpetas/> y </carpetas/index>).
 - b. [Crear Carpetas \(/carpetas/crear\)](#)
 - c. [Tareas en Carpeta \(/carpetas/tareas/{id}\)](#)
7. Tareas
 - a. [Mis Tareas](#)
 - i. Listado de tareas del usuario (</tareas/> y </tareas/index>).
 - b. [Crear Tareas \(/tareas/crear\)](#)
 - c. [Detalle Tarea - Comentarios y Archivos \(/tareas/detalle/{id}\)](#)
8. Página de login redirige a Página de inicio.
9. Página de registraci3n redirige a Página de inicio.

(*) en caso de querer acceder a una página donde es necesario estar logueado, se redigirá al login y luego se volverá a redirigir a la página donde quería ir inicialmente.

4.0 Menú

- Inicio (/)
- Carpetas
 - [Mis Carpetas \(/carpetas/ y /carpetas/index\)](#)
 - [Crear Carpetas \(/carpetas/crear\)](#)
- Tareas
 - [Mis Tareas \(/tareas/ y /tareas/index\)](#)
 - [Crear Tareas \(/tareas/crear\)](#).
- [Salir \(/home/logout\)](#). Luego debe redirigir a la home.

4.1 Registración de usuario

Al ingresar a la aplicación, estará la opción para que un nuevo usuario pueda registrarse. En esta opción se le solicitarán varios datos. Todos son obligatorios. En caso de estar logueado, deberá redirigir a la Página de inicio.

Los datos solicitados serán:

- Nombre. Máximo de 50 caracteres. Obligatorio.
- Apellido. Máximo de 50 caracteres. Obligatorio.
- Email. Máximo de 200 caracteres. Obligatorio.
 - Se deberá validar que el formato del dato ingresado sea un email.
- Contraseña. Máximo 20 caracteres. Obligatorio.
 - La contraseña al menos deberá contener al menos 1 número, 1 letra mayúscula y 1 letra minúscula.
 - Deberá existir otro campo donde el usuario deba volver a ingresar la contraseña para validar su correcta escritura.
 - Ambos campos de contraseña no deben ser visibles para el usuario sino que deben enmascarar el valor correspondiente.
- Captcha. El usuario deberá ingresar el valor correcto para poder registrarse.

Validaciones:

- En caso de que ya exista un usuario registrado activo con el mismo email, se deberá validar y mostrar un mensaje amigable que indique que el email ya existe.
- En caso de que ya exista un usuario registrado inactivo con el mismo email, se deberá permitir la registración del usuario. Activando la registración ya existente y no duplicando la registración del mismo. Se deberá actualizar los datos Nombre, Apellido y Contraseña.

Al momento de activarse un usuario, se deberá crear una carpeta "General" que será utilizada por defecto en aquellas tareas que no se les asocie ninguna carpeta.

4.2 Login de usuario

El usuario podrá ingresar a la aplicación con su email y contraseña definidos en la registración.

Los datos solicitados serán:

- Email. Máximo de 50 caracteres. Obligatorio.

- Se deberá validar que el formato del dato ingresado sea un email.
- Contraseña. Máximo 20 caracteres. Obligatorio.
 - El campo de contraseña no debe ser visible para el usuario sino que debe estar enmascarado.
- Recordarme. Checkbox que al hacer login y estar marcado, creará una cookie (encriptada) que será utilizada para que en un futuro no sea requerido el login nuevamente para ese usuario en ese navegador.

Validaciones:

- El usuario debe estar registrado en el sistema. Caso contrario, se deberá mostrar un mensaje amigable "Verifique usuario y/o contraseña."
- El usuario debe estar activo. Caso contrario, se deberá mostrar un mensaje amigable "Usuario inactivo."
- La contraseña debe coincidir con la que está registrada el usuario. Caso contrario, se deberá mostrar un mensaje amigable "Verifique usuario y/o contraseña."

Al ingresar al sitio el usuario será dirigido a la pantalla de Home donde se ve un menú con las opciones posibles y, además, verá todas las tareas que se encuentran asociadas a su usuario. Adicionalmente, en la parte superior derecha del sitio, aparecerá un link/boton de logout que al clickearlo borrará los datos de sesión y redirigirá a la página de Login.

4.3 Home

La Home de la aplicación tendrá el menú con todas las opciones que se deben soportar de creación, modificación y eliminación. Junto a este menú se deberá mostrar el listado de tareas no completadas asignadas al usuario actual ordenadas por prioridad y luego por fecha de fin ascendente. En caso de que no tengan fecha de fin asignada, la prioridad será el factor por el cual serán ordenadas las tareas.

En el lado izquierdo se deberá mostrar el listado de las carpetas creadas por el usuario. Al hacer click en cada una debe redirigir a Tareas de Carpeta, donde se verán sólo las tareas vinculadas a esa carpeta, asignadas al usuario actual ordenadas por prioridad y luego por fecha de fin ascendente.

El listado de tareas tendrá las siguientes columnas:

- 1) Fecha Fin
- 2) Nombre
- 3) Prioridad
- 4) Carpeta

- 5) Estimado
- 6) Completada
- 7) Botón/link “Ver” redirige a pantalla de detalle de tarea (/tareas/detalle)
- 8) Botón/link/checkbox “Completar” (solo para las tareas no completadas), el mismo marca la tarea como Completada.

4.4 Mis Carpetas (*1)

Se visualizarán todas las carpetas del usuario actual ordenadas por su Nombre de manera ascendente, al hacer click en alguna carpeta debe redirigir a Tareas en Carpeta.

4.5 Tareas en Carpeta (*1)

Se visualizarán todas las tareas del usuario actual pertenecientes a la carpeta elegida (parámetro en la url).

4.6 Crear Carpeta (*1)

Las carpetas se crearán para agrupar tareas. El uso de las mismas, al momento de crear una tarea es opcional. Luego de crear una Carpeta debe redirigir a Mis Carpetas.

Los datos solicitados son:

- Nombre. Máximo de 50 caracteres. Obligatorio.
- Descripción. Máximo de 200 caracteres. Opcional.

4.7 Crear Tarea (*1)

Una vez logueado en la aplicación, el usuario podrá crear nuevas tareas dentro de la aplicación. Luego de crear una Tarea, debe redirigir a Mis Tareas.

Los datos solicitados serán:

- Nombre. Máximo de 50 caracteres. Obligatorio.
- Descripción. Máximo de 200 caracteres. Opcional.
- Estimado Horas. Número con hasta 2 Decimales. Opcional
- Fecha Fin. Opcional.
- Prioridad. Los elementos posibles son: Urgente, Alta, Media, Baja. Opcional.
- Carpeta. Las carpetas posibles se obtendrán dinámicamente y ordenadas en orden alfabético ascendente (de la A-Z). Opcional.

Por defecto todas las tareas tendrán una prioridad Baja.

Se permitirán crear tareas con igual nombre.

En caso de que el usuario no haya creado carpetas o no haya seleccionado ninguna, las tareas se deberán asociar a la carpeta "General". A nivel Base de datos, deben usar la misma carpeta compartida (posee IdUsuario = NULL).

4.8 Mis Tareas (*1)

Se visualizarán todas las tareas del usuario actual ordenadas por su fecha de creación de manera descendiente, al hacer click en alguna tarea deberá redirigir a Tareas en Carpeta.

Se deberá incluir una opción que permite filtrar las tareas aún no completadas de las que sí lo están.

4.9 Detalle de Tarea - Comentarios y archivos adjuntos (*1)

Se visualizará toda la información de la tarea elegida sin poder editarla, visualizando debajo listado todos los comentarios ordenados por fecha de creación descendente y archivos adjuntos (nombre y link de descarga) ordenados por fecha de creación descendente .

El usuario podrá ingresar un nuevo comentario.

El usuario podrá adjuntar un nuevo archivo. El archivo subido se guardará en (/archivos/tareas/{idtarea}/{nombre-archivo}_{valor aleatorio}) ejemplo:

<https://github.com/unlam-pw3/subir-imagen>

(*1) En caso de querer acceder a una página donde es necesario estar logueado, se redigirá al login y luego se volverá a redirigir a la página donde quería ir inicialmente.

4.10 Logout

Desloguea al usuario y redirige a la Página de inicio.

5. Requisitos Técnicos

5.1 Proyecto .NET

1. El trabajo práctico deberá ser realizado utilizando ASP.NET MVC y Entity Framework. El tipo de proyecto a utilizar es una aplicación web MVC.

5.2 Estilos

1. No se permitirán que se utilicen los estilos ya provistos por Microsoft en la aplicación de ejemplo que provee Visual Studio.
2. Todos los archivos .css deberán estar dentro de una carpeta.
3. No utilizar estilos inline (atributo style="") ni definir estilos dentro de una pagina (tags <style>).
4. Debe de utilizarse algún framework/biblioteca de hojas de estilo. Algunos ejemplos:
 - a. Twitter Bootstrap (<http://getbootstrap.com/> , temas <http://bootswatch.com/>)
 - b. Foundation (<http://foundation.zurb.com/docs/>)
 - c. KickStart (<http://www.99lime.com/elements/>)
 - d. Bulma (<http://bulma.io/>)
 - e. Otro definido por los alumnos y validado con el cuerpo docente.

5.3 JavaScript

1. No utilizar JavaScript inline dentro de una página, se deberá referenciar a archivos js.
2. Todos los archivos .js deberán estar dentro de una carpeta.
 - a. Si se decide utilizar algún js que no es propio, el mismo deberá estar dentro de una subcarpeta.
3. **Se deberá utilizar Bundle and minification** (ejemplo:
<https://geeks.ms/etomas/2012/07/30/bundles-en-asp-net-mvc4/>)

Recomendaciones

Las funciones específicas de una página, deberían estar en un archivo .js con el mismo nombre de la página.

Aquellas funciones utilizadas en más de una página, deberían de estar dentro de otro archivo .js de uso común.

5.4 HTML

1. No utilizar tags table para organizar el contenido de una página en columnas, los tags

table solo están permitidos para representar una grilla/listado de información.

2. Todo el contenido debe ser SEO friendly.
3. Se requiere el uso de Layouts para estructura los formularios web de la aplicación. Dentro del BaseLayout deberán referenciarse las hojas de estilo y archivos de javascript de uso común por toda la aplicación.
4. Debe utilizarse xHtml.

3.5 Validación

1. Utilizar validaciones tanto del lado del cliente (JavaScript) como del lado del servidor utilizando DataAnnotations.
2. Se puede utilizar una lista que detalle todos los campos que no cumplieron con las validaciones.

5.6 Arquitectura y Consideraciones de Desarrollo

1. La capa de acceso a datos deberá ser realizada con Entity Framework. Este componente de .NET será explicado en clases a fin de que los alumnos comprendan cómo utilizarlo.
2. La capa Web deberá ser realizada utilizando MVC.
3. Deberán crear un repositorio privado en Github que deberá ser compartido con los profesores y donde irán subiendo los avances
4. Utilizar la menor cantidad posible de código en los /Controllers/[Entidad]Controller, e intentar que en los mismos haya llamadas a métodos dentro de otro proyecto que contenga las reglas de negocio.
5. Compatibilidad con exploradores.
 - a. Google Chrome (la última versión para Windows).

5.6 Repositorio de Código

El código fuente deberá ser administrado utilizando un repositorio de código en GitHub. Al momento de entregar el trabajo práctico, los alumnos deberán compartir el link para que los profesores puedan descargar el código del proyecto.

Todo lo necesario para ejecutar el proyecto deberá estar incluido y la descripción de los pasos necesarios para correr la aplicación (caso que haya adicionales a la ejecución normal de un proyecto ASP.NET MVC) deberán estar detallados en el archivo readme.md ubicado en la raíz del proyecto.

6. Forma de Entrega

El TP deberá ser enviado a los siguientes destinatarios, copiando a todos los integrantes del equipo en el email enviado.

- a) mpazwasiuchnik@unlam.edu.ar
- b) pnsanchez@unlam.edu.ar
- c) el resto de los integrantes que participaron del tp (como CC)

El email deberá contener los siguientes puntos:

- a. El título del email será 2018-1C-TP-[Integrantes]
- b. Respuestas a las siguientes preguntas:
 - i. ¿Qué nota creen que deberían sacar en el tp? (1-10, donde para 7 debe estar toda la funcionalidad pedida) y ¿por qué?
 - ii. ¿Qué cosas creen que podrían mejorarse?
 - iii. ¿Qué les resultó más complicado?
- c. Link público del repositorio de github para que el proyecto pueda ser descargado. El repositorio deberá incluir el código fuente de la solución de punto net y los scripts de base de datos adicionales de datos que crean necesarios para probar la aplicación. No deberán agregar el script de la estructura de la base de datos compartida por la cátedra.
 - i. incluir Script de sql con datos de prueba suyos.
 - ii. incluir credenciales de usuarios registrados.

7. Condiciones de Aprobación

1. El TP deberá cumplir con todos los requisitos técnicos y funcionales definidos. No se aceptará que alguna funcionalidad o requisito técnicos no se encuentren plasmados en el trabajo práctico.
2. Al momento de la entrega del trabajo práctico, todos los alumnos de cada equipo deberán estar presentes para la defensa del trabajo práctico.

3. En la defensa del trabajo, se evaluará el grado de conocimiento y participación en el desarrollo del trabajo práctico de cada alumno.