



TPE - Parte 1

Consigna

Desarrollar un **sitio web dinámico** que permita visualizar items con sus categorías, los cuales pueden ser administrados por un usuario administrador.

La presentación es de acuerdo a los grupos definidos en la planilla correspondiente. La entrega se realizará en un repositorio público de GIT a elección del grupo. La URL debe estar subida a la planilla en la fecha de entrega pautada.

Descripción de datos:

Debe contar con una base de datos (acorde al tipo de página a implementar). La base de datos debe tener al menos una relación 1-N. Por ejemplo, debe haber una *categoría*, y un nivel de *ítem* que es agrupado en la categoría. Las columnas de cada tabla deben ser acordes a la lógica de su página.

A continuación se dan ejemplos sugeridos que se pueden aplicar a muchas páginas, otras opciones son bienvenidas.

Comercial

Debe contar con las tablas Producto y Categoría. Un producto puede pertenecer sólo a una categoría.

Novedades

Debe contar con las tablas Noticia y Sección. Una noticia puede pertenecer sólo a una sección.

Rutas y ciudades

Debe contar con las tablas Ciudad y Ruta. Una ruta puede llegar a varias ciudades.

Otras

Funciones en salas de cine, animales en reinos animales, películas según el estudio de filmación (o género), materias con su profesor a cargo (uno solo), temporadas y episodios, etc

Planilla

En la planilla de consigna se debe indicar las dos tablas (mínimo) y todos sus campos. Esto debe ser revisado con los ayudantes antes de comenzar a implementar.



Requerimientos Funcionales

Usuario visitante/anónimo

Debe tener 2 paginas donde se muestre el contenido dinámicamente generado de la base de datos.

- Se debe poder visualizar todos los items (Listado de Items)
- Se debe poder navegar y visualizar cada ítem particularmente (detalle de ítem)
- Se debe poder visualizar todas las categorías (listado de categorías)
- Se debe poder visualizar los items ordenados por categorías.

Importante: En cada ítem siempre se debe mostrar la categoría a la que pertenece (el nombre de la misma).

Administrador de Datos

Debe haber una página de administración, la cual se accede mediante “nombre de usuario” y “contraseña”. El usuario puede salir del modo “administrador” (logout).

- Solo los usuarios logueados pueden modificar las categorías y los items. Los usuarios visitantes pueden ver el sitio sin loguearse .

Se debe crear servicios de ABM (alta/baja/modificación) para administrar los datos:

Administrar Items (entidad del lado N de la relación).

- Lista de Items (Noticias/Productos/...)
- Agregar Items (Noticias/Productos/...)
- Eliminar Items (Noticias/Productos/...)
- Editar Items (Noticias/Productos/...)

Importante:

Al agregar Items (Noticias/Productos/...) se debe poder elegir la categoría a la que pertenecen utilizando un dropdown que muestre el nombre de la misma.

Administrar Categorías (entidad del lado 1 de la relación)

- Lista de Categorías
 - Agregar Categorías
 - Eliminar Categorías
 - Editar Categorías.
-
- Como dueño del sitio, quiero poder loguearme para administrar el sitio.



Requerimientos Técnicos (no funcionales)

- Todos los HTML deben mostrarse utilizando una plantilla (Smarty).
- Todas las acciones realizadas en la página deben estar manejadas utilizando el patrón MVC.
- *Las URL deben ser amigables.*
- El sitio debe generar su base de datos con datos iniciales al momento de cargarlo por primera vez. ****opcional****

Comentarios:

- Para el macro se necesita TP04, pero muchas partes de esas tareas se pueden empezar antes y unir después.
- Las consultas SQL de todo se pueden hacer luego de T02.
- Los HTML y las plantillas Smarty se pueden hacer luego de TP05.

Pautas de corrección

Se considera fundamental la aplicación de buenas prácticas, y la elección apropiada de cada tecnología para cada punto a resolver. Hacer prácticas marcadas como malas se considera motivo de pérdida de promoción y de puntaje de cursada.

En caso de no tener la planilla completa para la fecha fijada se perderá la **promoción**.

La exposición/defensa se realizará en una computadora dada por la cátedra, por lo que se recomienda probar la página web en diferentes monitores y sistemas operativos.

En el caso de trabajos grupales, la defensa es de cada miembro por separado. **Todos los integrantes deben poder responder a cualquier pregunta de cualquier parte del trabajo.** En caso de errores graves se puede desaprobado el trabajo (lo que implica desaprobado la cursada).

Se debe entregar el link repositorio GIT del grupo con el código fuente

- Planilla completa (Tema del Sitio y grupo): Jueves 30 de Septiembre.
- Entrega:
 - Se debe completar el repositorio GIT en la planilla
- **TANDIL**
 - **Entrega: 31 octubre 23.59hs**
 - **Defensa: 1 noviembre de 11 a 13 y 15 a 17**
- **BOLIVAR**
 - **Entrega:**
 - **Defensa:**