

# TPE - Parte 1

## Consigna

Desarrollar un **sitio web dinámico** que permita visualizar un conjunto de **ítems** cargados dinámicamente por un usuario administrador. Estos *ítems* deben estar modelados en el sistema mediante una relación 1 a N. Por ejemplo, se puede pensar como **ítems** pertenecientes a **categorías**, **ítems** que tienen un conjunto de **componentes** o cualquier modelo de datos deseado que se adapte a esta relación.

En adelante, se usará **ítems** y **categoría** para describir los requerimientos funcionales, pero cada grupo lo adaptará a su modelo de datos.

## 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.

### **Noticias**

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

### **Autores y libros**

Un conjunto de autores con los libros que escribieron.

### **Rutas y ciudades**

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

### **Productos con especificaciones**

Debe contar con las tablas Producto y Especificación. Una especificación pertenece a un producto.

### **CVs**

Un listado de los potenciales empleados con sus distintas líneas en su currículum. Debe contar con las tablas Empleado y Línea.

### **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.

## **Requerimientos Funcionales**

### **Acceso público**

Deben existir **diferentes secciones** donde se muestre el contenido dinámicamente generado desde la base de datos. Estas secciones pueden ser accedidas de **manera pública**, no es necesario loguearse.

- **Listado de ítems:** Se debe poder visualizar todos los ítems cargados
- **Detalle de ítem:** Se debe poder navegar y visualizar cada ítem particularmente
- **Listado de categorías:** Se debe poder visualizar un listado con todas las categorías cargadas
- **Listado de ítems x categoría:** Se debe poder visualizar los ítems perteneciente a una categoría seleccionada

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

## Acceso administrador de datos

Debe existir una sección para la **administración** de datos, la cual es accedida solo a usuarios administradores del sitio.\*

- El usuario administrador debe loguearse con **usuario** y **contraseña**.
- El usuario debe poder **desloguearse** (logout)
- Solo los usuarios logueados pueden modificar las categorías y los ítems.

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

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

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

Importante:

*Al agregar Ítems (Noticias/Productos/...) se debe poder elegir la categoría a la que pertenecen utilizando un select 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.

## Requerimientos Técnicos (no funcionales)

- Todos los HTML deben mostrarse utilizando un motor de plantillas (Smarty o similar)\*.
- Todas las acciones realizadas en la página deben estar manejadas utilizando el patrón MVC.
- *Las URL deben ser semánticas.*
- El sitio debe incluir un archivo sql para instalar la base de datos.

## Entrega

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.

### Planilla compartida

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.

## 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 computadoras.

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).

### Notas

\* **Autenticación y seguridad:** Se recomienda comenzar con la sección “privada” sin autenticación. Una vez dada la clase de seguridad del 27/09 es muy sencillo protegerlo con usuario y contraseña.

\*\* **Vistas:** Se recomienda mantener una UI super sencilla hasta que se dé la clase de Templates Engine del 20/09