

# WEB 2

## TPE - Parte 2

### Consigna

Para la segunda entrega, se debe continuar el trabajo de la primera etapa. El objetivo es agregar nueva funcionalidad detallada en forma de *user stories* abajo. Las *stories* se agrupan por tema sólo para facilitar la organización.

#### Roles de usuario:

1. Como usuario quiero poder registrarme en el sitio generando nombre de usuario/mail y contraseña.  
*Al registrarse el usuario se loguea automáticamente. Este usuario no tiene privilegios de administración.*
2. Como administrador del sitio, quiero poder asignar o quitar permisos de administración a cualquier usuario.
3. Como administrador del sitio, quiero poder eliminar usuarios.

#### Comentarios (todo por API REST):

4. Como usuario registrado, quiero poder postear comentarios en los ítems del sitio asignándoles un puntaje de 1 a 5.  
*Cada ítem del sitio tendrá la posibilidad de recibir comentarios y puntuaciones **solamente** de usuarios logueados.*
5. Como administrador del sitio, quiero poder borrar comentarios.

### USER STORIES OPTATIVAS

Las User Stories optativas suman un **punto cada uno**. Se debe completar al **menos una para acceder a la promoción ya que el resto del trabajo sin opcionales suma 6 puntos**.

6. Como administrador del sitio, quiero poder asociar una imagen a un ítem.  
*Las imágenes de los “ítems” se deben poder subir y eliminar desde el ABM de los mismos.*
7. Como usuario quiero poder navegar los listados de ítems en forma paginada.  
*Se debe generar una paginación del lado del servidor para recorrer los listados en forma paginada. (Se recomienda empezar con “anterior” y “siguiente”)*

8. Como usuario quiero poder realizar búsquedas avanzadas de los ítems.  
*Se debe incluir un formulario de búsquedas avanzadas donde se filtren los ítems dependiendo de los atributos internos. Esta búsqueda se debe resolver del lado del servidor.*
9. Como usuario quiero poder ordenar los comentarios por antigüedad o puntaje, ascendente o descendente. (Via API REST)  
*Se debe ordenar del lado del servidor.*
10. Como usuario quiero poder filtrar los comentarios por cantidad de puntos. (Via API REST)  
*Se debe filtrar del lado del servidor.*

## Aclaraciones

### Respecto a los comentarios:

- Todo el sistema de comentarios debe funcionar por medio de una **API REST**. Por ejemplo, cuando un usuario ingresa un comentario, el sitio no se debe recargar en su totalidad, solo el listado de comentarios.
- Se debe renderizar todo lo relacionado a comentarios utilizando **Client Side Render JS** mediante la API REST.
- Los comentarios se pueden ver siempre, pero sólo agregar por usuarios registrados y sólo borrar por administradores.
- Los comentarios se deben poder crear. No es necesario poder modificarlos.

### Respecto a los usuarios:

- Existirán al menos dos roles de usuarios registrados. (administradores y no-administradores)
- Los usuarios registrados no son administradores (a menos que se les dé el permiso luego)

## Entrega

Se debe entregar en el mismo repositorio GIT del grupo.

Fecha:

**Entrega: Ver cronograma**

## Defensa: Ver cronograma

### Criterios de corrección

Se evaluará la correcta división de responsabilidades en las clases, no repetición de código, identificadores (nombres de clases, variables, etc) descriptivos, etc.

Los trabajos deben implementar la totalidad de la funcionalidad (ambas entregas) funcionando correctamente.

- User Story sin implementar: -2/-4 (según complejidad de la story).
- User Story que no anda: -2.
- User Story con bug menor: -1.
- Poca Prolijidad General (código difícil de leer, mala división de clases, pero respetando MVC): hasta -3.
- Sin chequeo de entradas en el servidor (`isset && != ""`): -1
- No respeta MVC: desaprobado.
- PDO inseguro: desaprobado.

Nota máxima: 10