Desafío



Aclaraciones:

El siguiente desafío fue diseñado por el Staff de G&L Group con el fin de evaluar los conocimientos de los postulantes al curso de PHP.

Puede encontrar distintas formas de resolver el desafío, le recomendamos priorizar los métodos más óptimos que encuentre, priorice siempre la performance.

Este desafío no representa un examen de ingreso a la empresa, pero es obligatoria su aprobación para la obtención del certificado del curso.

Datos

### Generar una base de datos de nombre “desafio” con las siguientes tablas:

|  |  |  |
| --- | --- | --- |
| Tabla: materias | | |
| Nombre Campo | **Tipo dato** | **Atributos** |
| id | Int | obligatorio, clave primaria, autogenerado |
| carrera\_id | Int | Obligatorio, clave foranea de la columna id de la tabla Carreras |
| nombre | longitud máxima de 50 caracteres | obligatorio |
| descripcion | Longitud maxima de 255 caracteres | - |
| carga\_horaria | Int | Obligatorio, opciones entre 2,4,8 o 10 |

La tabla “Carreras” es dada a través del script: “carreras.sql”

Funcionalidades

### Se pide: desarrollar una página web que se conecte con la base de datos creada anteriormente y posea los siguientes controles y funcionalidades:

**Pantalla Principal**

**Obligatorio:**

* Se deben listar todas las materias creadas.
* Se debe poder acceder a la pantalla de Alta de materias.
* Se debe poder dar de baja una materia ya creada (baja física).

**Opcional:**

* Contar con un filtro de las materias en el listado (se aconseja utilizar el comparador LIKE para implementar, también se aconseja usar el evento keyup de jQuery).
* Contar con una pantalla de modificación, que permita editar los datos de materias ya creadas.

**Pantalla de Alta**  
  
**Obligatorio:**

* La forma de asignar la carrera con la materia, debe ser a través de una lista desplegable, en el cual los datos de la carrera, deben ser obtenidos de la tabla Carreras.
* Debe validarse que en la descripción, solo puedan ingresarse letras.
* En la carga horaria, solo debe poder seleccionarse los valores: 2, 4, 6, 8 o 10.
* No se puede registrar una Materia que no cumpla con las condiciones mencionadas anteriormente.

**Opcional:**

* Informar de los errores en la entrada de datos, en tiempo real (antes de presionar el botón Agregar); por ejemplo, si ingreso en la descripción un “#”, el borde del input se ponga en rojo.

**Pantalla de modificación (opcional)**

Similar a la pantalla de alta (cambiar los nombres de los botones).  
Sería interesante poder ver los datos previamente guardados dentro de los inputs a modificar (por ejemplo que los datos actuales se vean como un placeholder).   
Tener en cuenta que al modificar, también deben validarse los datos!!

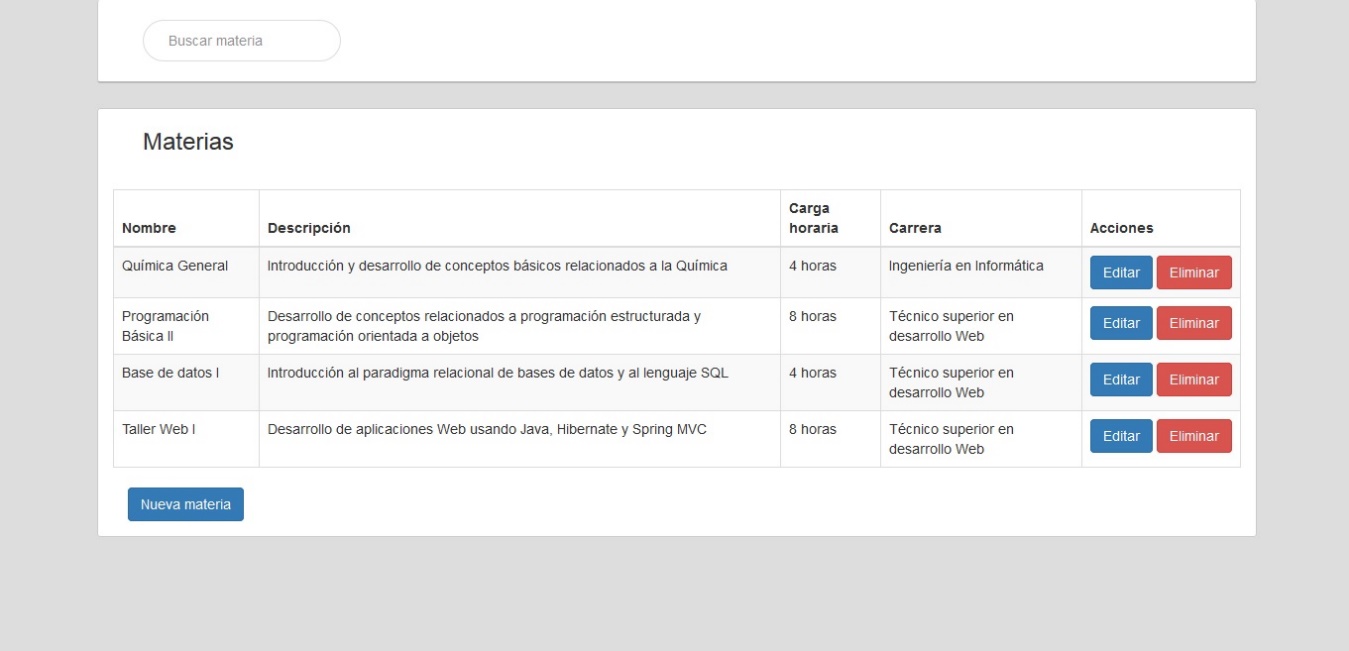
**Estilos**

**Obligatorio:**

* No se debe utilizar la etiqueta “style” en el HTML, debe realizarse todo mediante el uso de uno o varios archivos de extensión “.css”

**Opcional:**

* Implementar Boostrap para la vistas de base y usar un archivo propio de “.css” para mejorar la vista.
* Que las vistas sean responsive al achicar o agrandar la pantalla.



\*La imagen es sólo a **MODO DE EJEMPLO**, no es necesario que esté 100% igual.

**Aclaraciones**

* El código debe estar comentado e indentado (tabulado).
* El sistema será probado en Google Chrome.
* Se recomienda poner separados los archivos de JavaScript, es decir, un archivo JS para cada vista.
* La acción de “Agregar Materia” debe llevar a una nueva vista, donde este el formulario (luego de agregarse, debe mostrar nuevamente el listado actualizado).
* La acción de “Borrar Materia” debe lanzar una confirmación ejemplo: “Esta seguro que desea eliminar?, con opciones de Aceptar o Cancelar (Opcional: utilizar un modal) en caso de Aceptar, se debe eliminar la materia y actualizar la vista correspondiente, donde se pueda visualizar que ya no existe más.
* La acción de modificar (opcional) se aconseja que sea similar al Alta.

**Formato de entrega:**

Un zip con el nombre y apellido de los integrantes (Por ej. si son Ramiro Cerratto y Sergio Raggio, se debe llamar “**RamiroCerratto-SergioRaggio.zip**”) protegido con contraseña, que debe ser enviada por email.

El mismo debe contener:

* Código fuente
* Scripts de la base de datos ( 2 archivos):
  + “estructura.sql” que contenga la estructura de la base de datos
  + “datos.sql” que contenga los datos brindados, y datos de ejemplo
* Documentación (un archivo en Word que contenga explicaciones que crean necesarias, posibles mejoras, extras que hicieron, en caso de hacerlos)

# 