

IMPORTANTE:

Se pueden bajar templates de internet o traer código hecho, pero en ningún caso se debe incluir código obsoleto o que no cumpla ninguna función dentro del parcial.

1- **punto_01.html**: diseñar un documento HTML 5 que posea un botón y un control para seleccionar un color. Al pulsar el botón, se cambiará el color de fondo del cuerpo del documento.

2- **punto_02.html**: diseñar un documento HTML 5, que permita enlazar un control de tipo rango (de 0 a 100) con una barra de progreso (de BOOTSTRAP) activa, verde y rallada. El valor inicial de ambos controles será 10%. Cada cambio también debe mostrarse por consola.

3- **punto_03.html**: tomando como punto de partida el ejercicio anterior se pide: reemplazar el control de tipo rango, por un botón (azul y como texto RANDOM) que al pulsarse invoque a una función, que por AJAX, retornará un número aleatorio que actualizará la barra de progreso. La página **backend/delay.php** retornará dicho valor luego de dos segundos.

4- **punto_04.html**. tomando como punto de partida el ejercicio anterior se pide: agregar un elemento Alert para mostrar mensajes (*alert-info*) y un spinner (de color rojo), ambos de BOOTSTRAP. Se debe actualizar la barra de progreso e informar el porcentaje en el Alert. En el tiempo de espera, el spinner se debe de activar y cuando se actualice se debe detener.

5- **punto_05.html**: generar un documento HTML 5 que utilice BOOTSTRAP, que posea un botón (azul con el texto Tabla Random), un elemento Alert (*alert-info*) que cuando se pulse el botón se invoque a **backend/array.php** por AJAX (que retornará un array de string con valores aleatorios "F" ó "M"). Esto permitirá generar una tabla dinámica cuyas filas serán de color rosa (*danger*) si es "F" o de color celeste (*info*) si son "M".

6- **punto_06.html**: diseñar un documento HTML 5, que utilice BOOTSTRAP, y que permita mostrar cuatro imágenes (dos imágenes por fila): **img/Jellyfish.jpg** (con vértices redondeados), **img/Koala.jpg** (redonda), **img/Lighthouse.jpg** (con borde) y **img/Penguins.jpg** (sin formatos especiales). Todas las imágenes deben ser de 64 x 64 píxeles.


El diseño de la página debe tener el siguiente comportamiento responsive: si se visualiza en una PC la primer fila tendrá las columnas de un 40% - 60%. La segunda fila 80% - 20%.

Si se visualiza desde una tablet: fila uno con 80% - 20% y fila dos con 40% - 60%.

Si se visualiza desde un teléfono: fila uno con 50% - 50% y la fila dos con 20% - 80%.

Universidad Tecnológica Nacional											
Facultad Regional Avellaneda											
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos											
Materia: LABORATORIO III											
Apellido:							Fecha:				
Nombre:							Docente ⁽²⁾ :				
División:							Nota ⁽²⁾ :				
Legajo:							Firma ⁽²⁾ :				
Instancia ⁽¹⁾ :	PP	X	RPP		SP		RSP		FIN		

Random

Random 

22%

Tabla Random

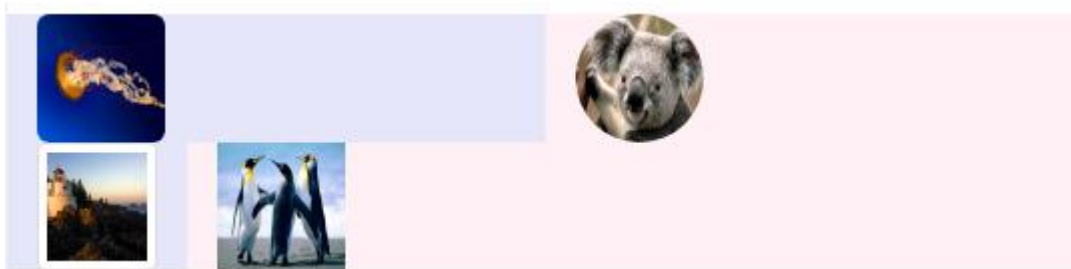
F - M - F - M - M - F - F - M - M - F

SEXO

F

M

F



7- **punto_07.html**. diseñar una página que posea un formulario para el ingreso de datos de un usuario. Se debe tener en cuenta que: todos los campos son requeridos, el control de tipo fecha debe tener como límites el 25-08-1944 y 31-12-1999, que el e-mail deberá tener un formato correcto y que la contraseña no sea visible. Los datos se enviarán a **backend/login.php** que recibirá los siguientes parámetros por POST: usuario, fechaNac, email y password. Dicha página retornará el nombre del usuario, la fecha de nacimiento y el e-mail, siempre y cuando el usuario sea "pepe" y la contraseña "123456", caso contrario se mostrará "ERROR EN LOGIN".

8- **punto_08.html**: Generar una página que posea un menú, un cuerpo y un pie de página (con el texto UTN fra). El menú tendrá una opción de Home (no hace nada), una opción por cada punto realizado (se deberá enlazar cada opción con cada página **.html**) y por último un ícono con su apellido y nombre.

Generar una regla CSS sólo para el pie de página que establezca: color de fondo **#555**, color de fuente **#fff** y un padding de 15 píxeles.

Crear otra regla CSS, con un identificador único para el cuerpo, con lo siguiente: altura de 500 píxeles y con un color de fondo **#5f9ea0**.

Nombre de Usuario

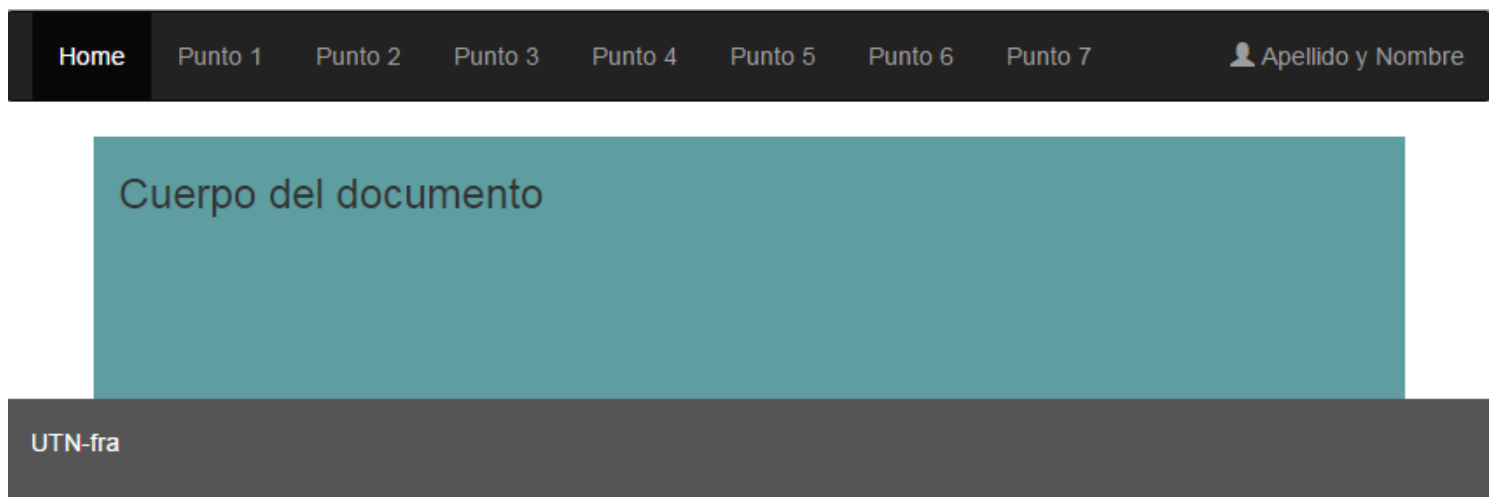
dd/mm/aaaa

Email

Contraseña

Aceptar

Limpiar



NOTA: LOS PUNTOS ANTERIORES DEBEN ESTAR CORRECTOS PARA OBTENER COMO NOTA UN **4 (CUATRO)**. A PARTIR DE LA RESOLUCION DE LOS PUNTOS ANTERIORES SE HABILITARA LA RESOLUCION DE LOS SIGUIENTES PUNTOS:

9.- (2pts.) Realizar una aplicación web que muestre un listado de remeras, tomando como origen de datos el archivo **remeras.json**. La aplicación tendrá un `<input type="text">` y un `<input type="button">` que al pulsarlo, generará dinámicamente un listado de las remeras filtradas por país de fabricante (armar una tabla html en javascript). Como página nexa, utilice **administrarRemeras.php**, pasándole la opción "traerRemeras".

10.- (2pts.) Realizar una aplicación web que permita agregar un nuevo JSON al archivo **remeras.json**. La aplicación deberá tener tantos elementos de tipo `<input type="text">` como atributos tenga el JSON. Como página nexa, utilice **administrarRemeras.php**, pasándole la opción "agregarRemera" y el JSON a ser agregado.

Nota: Para simplificar el ejercicio, *'hardcodee'* el valor del campo **logo** con el siguiente valor:
"<https://robohash.org/authicperferendis.bmp?size=50x50&set=set1>".

11.- (2pts.) A la aplicación del punto anterior, agregarle al listado una columna extra (ACCION) que posea un elemento `<a>` (cuyo texto sea Eliminar) que permita quitar el elemento seleccionado del archivo **remeras.json**, previa confirmación por parte del usuario. Como página nexa, utilice **administrarRemeras.php**, pasándole la opción "quitarRemera" y el ID de la remera (objJSON.id) a ser eliminado.