



IIC2513 – Tecnologías y Aplicaciones Web

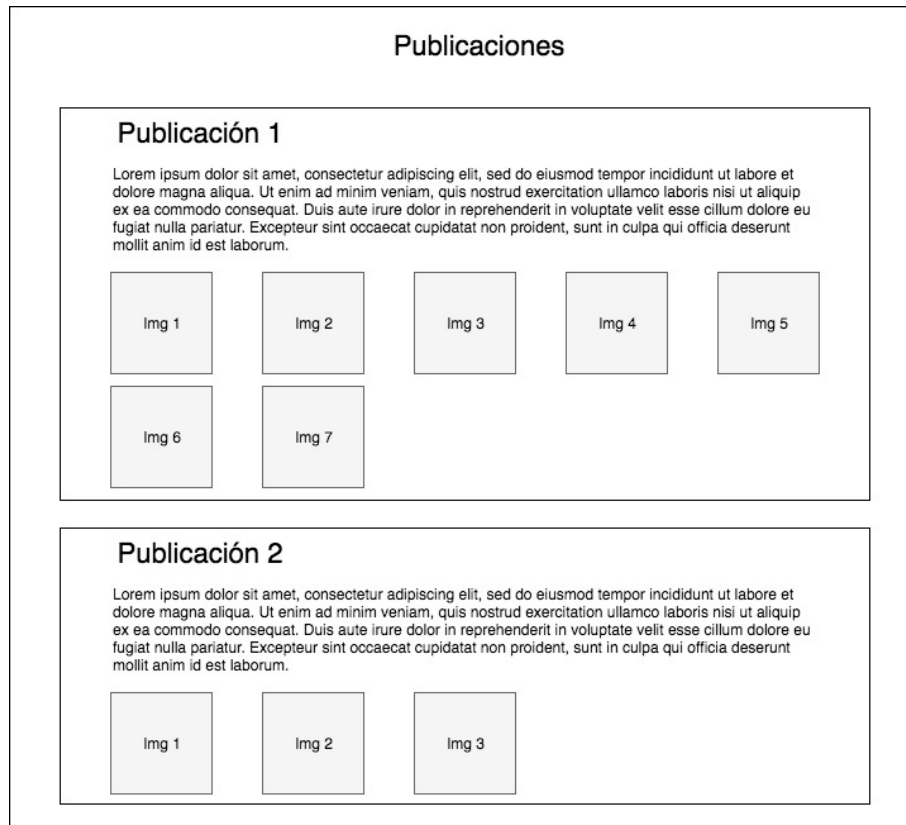
Interrogación 3

Instrucciones: Sea preciso: no es necesario escribir extensamente pero sí ser preciso. En caso de ambigüedad, utilice su criterio y explicita los supuestos que considere convenientes. Esta interrogación fue diseñada para durar 120 minutos. Escriba su nombre en cada una de las hojas que entregue.

1. (0.6 pts) Suponga que usted es un usuario registrado de una aplicación web, pero no puede iniciar sesión porque olvidó su contraseña. Al solicitar ayuda con su problema, se le envía un correo con la contraseña que había perdido.
 - a. ¿Qué información sobre la aplicación web puede deducir en base al escenario anterior?
 - b. Explique por qué esta es una mala política a partir de 2 vulnerabilidades a las que se ven expuestos los usuarios de la aplicación.

2. (0.8 pts) Explique cuáles son los 4 propósitos del *Asset Pipeline* de *Ruby on Rails* y los beneficios de cada uno.

3. (1.2 pts) Considere la siguiente imagen:



Se le pide que implemente el *HTML* y *CSS* necesarios para lograr ese resultado.

Notas:

- No considere el borde externo de la imagen, no es parte de la página web.
- Puede usar *SCSS* o *CSS*.
- No necesita escribir todo el contenido que se ve en la imagen en su *HTML*. Concéntrese en la estructura y utilice *placeholders*.
- Asigne tamaños y márgenes proporcionales a la representación de los elementos como se ven en la imagen.

4. (0.6 pts)

- a. Al utilizar *jQuery* para manipular el *DOM* se acostumbra que todo el código se encuentre dentro de una función anónima:

```
$(function() {...})
```

Explique qué implica esto.

- b. Por otra parte, al utilizar *Ruby on Rails* con *jQuery* se recomienda reemplazar el código anterior por:

```
$(document).on('turbolinks:load', function() {...})
```

Explique por qué este cambio, detallando qué hace la librería *turbolinks*.

5. (0.8 pts) Escriba el *output* resultante al ejecutar el siguiente código *JavaScript*:

```
1  var v1 = 0;
2  function f1(v2) {
3    console.log(v1);
4    v1 += v2;
5    var f2 = function () {
6      v1 += v2;
7      console.log(v1);
8    };
9
10   console.log(v1);
11   return function (v3) {
12     v2 += v3 || 0;
13     f2();
14     v1 += v2;
15     console.log(v1);
16   };
17 }
18
19 v1 = 1;
20 var ff = f1(1);
21 ff(1);
22 ff();
```

6. (1.2 pts) Considere la siguiente estructura *HTML*:

```
<h2>Consultar restricción:</h2>
<div>
  <label for="digit-input">Último dígito</label>
  <input type="number" id="digit-input" name="digit-input">
  <span id="error"></span>
</div>
<a id="validate" href="#">Validar</a>
<span id="result"></span>
<h2>Calidad del Aire</h2>
<p>
  "Nivel de contaminación: "
  <span id="pollution-index"></span>
</p>
<div id="comunidades">
  <a id="macul" href="#">Macul</a>
  <a id="la-florida" href="#">La Florida</a>
  <!-- más anchors para cada una de las comunas -->
</div>
```

Asuma que existen 2 constantes:

- **BANNED_DIGITS**: arreglo con *integers* que representan los dígitos con restricción
- **POLLUTION**: *hash* que contiene los niveles de contaminación de cada comuna. Las llaves son el identificador para cada comuna y los valores son *integers* que representan la contaminación correspondiente.

Se le solicita que con la ayuda de *jQuery* implemente las siguientes funcionalidades en *JavaScript*:

- Al hacer *click* en el *anchor* *validate*:
 - Si el campo *'digit-input'* está vacío, se agrega el texto *'Debe ingresar un número'* al elemento *'error'*, con la clase *'red'*. Además, se elimina cualquier texto y clase del elemento *'result'*.
 - Si el campo *'digit-input'* tiene un valor, se elimina cualquier texto y clase del elemento *'error'*. Por otra parte, si el arreglo **BANNED_DIGITS** no contiene elementos, el texto de *'result'* debe decir *'No hay restricción hoy'*, y el elemento no debe tener ninguna clase. De lo contrario se verifica si el valor de *'digit-input'* está contenido en el arreglo:
 - De ser así, se reemplaza el texto del elemento *'result'* por *'Su auto tiene restricción'* y se le asigna solamente la clase *'red'*.
 - De lo contrario el texto debe decir *'Su auto no tiene restricción'* y asignar solamente la clase *'green'*.
- Al hacer *click* sobre el *anchor* de una comuna se debe actualizar el texto del elemento *'pollution-index'* por el nivel de contaminación correspondiente a esa comuna.

NOTAS:

- Puede usar la función `Array.indexOf(value)`, que retorna el índice del primer `value` contenido en `Array`. Si el valor no se encuentra en el arreglo retorna -1.
- Algunas funciones de *jQuery* son:
 - `.on(events,[selector],[data],handler)`
 - `.val([value])`
 - `.text([text])`
 - `.removeClass([className])`
 - `.addClass([className])`
 - `.attr(attributeName,[value])`
 - `.find(selector)`

7. (0.8 pts)

- a. Defina y explique qué es *AJAX*, especificando en qué se diferencia de una petición “tradicional”.
- b. Explique 3 políticas claramente distintas con las que un servidor puede responder a una petición *AJAX* realizada por un navegador. Describa qué es necesario implementar a nivel de *Backend* y *Frontend* para cada una.