



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

## IIC2513 – Tecnologías y Aplicaciones Web (II/2018)

Profesor: Raúl Montes

### Interrogación 2

24 de octubre de 2018

#### Pregunta 1: Conceptos

Responde en forma precisa y concisa las siguientes preguntas:

1. (1.2 puntos) Siendo `hoge` y `piyo` funciones, y `hoge` una promesa, ¿es lo mismo ejecutar `hoge.then(hoge).then(piyo)` versus `hoge.then(hoge); hoge.then(piyo)`? Justifica.
2. (1.2 puntos) Tienes definido en tu aplicación estilos generales para todos los “links”. Sin embargo, tienes una cierta ruta de tu aplicación en que quieres que los “links” se vean con un estilo diferente. ¿Cómo puedes lograr esto de una buena forma? Explica y justifica los cambios que harías (no es necesario escribir código).
3. (1.2 puntos) Te llega un e-mail de un príncipe nigeriano que te promete una millonaria herencia, pero para desbloquear esos dineros necesita que le envíes las *cookies* de tu *browser*. ¿Se las enviarías? ¿Por qué? De la forma en que el template del curso usa las *cookies* para el manejo de sesión, ¿hay parte de las *cookies* que le podrías enviar sin mayor riesgo?
4. (1.2 puntos) En el contexto de CSS, explica que es `margin` y `padding`, y luego explica un caso en que ambas propiedades te pueden servir de manera similar y un caso en que su funcionamiento diferente haga que sólo una de ellas te sirva.
5. (1.2 puntos) Si en una variable `foo` tenemos un objeto, y en una variable `bar` tenemos una función, y `foo.__proto__ === bar.prototype` es `true`, ¿qué puedes decir respecto de `foo` y `bar`? Explica.

#### Pregunta 2: a esta interrogación le falta estilo...

Te encuentras con el siguiente código HTML que no tiene ningún estilo aplicado (más allá de los por defecto del browser) y, ya pudiendo imaginarte lo horrible que se ve, decides escribir el código CSS necesario para que quede bonito (o así lo crees tú).

A continuación encontrarás el código HTML/EJS y una foto del resultado esperado. Pero antes, algunas aclaraciones y notas al respecto:

- ambos botones de los servicios son iguales, pero el estado en el que está el de arriba es cuando el mouse está sobre el botón
- no importa ser preciso con los espaciados ni colores, puedes confiar en tu “ojímetro”
- fíjate en las aprovechar los selectores de manera de evitar repetir estilos

- no se espera que seas exhaustivo(a) para tener el puntaje máximo de esta pregunta; evaluaremos con una rúbrica que considerará cierto grado de completitud, uso apropiado de propiedades y selectores, conceptos de CSS aplicados, etc.
- el elemento `hr` viene de “horizontal rule”. Por defecto (estilos del browser) es un elemento de tipo bloque (como un `div`) pero de alto 0 y con 1 pixel de borde.

Listing 1: index.html.ejs

```
<div id="signup">
  <h1>Crea tu cuenta</h1>
  <div class="signup-options">
    <div class="third-party-signup">
      <p>
        Elige uno de estos servicios
      </p>
      <a href="/gugl">Gugl</a>
      <a href="/feisbuc">Feisbuc</a>
    </div>
    <hr />
    <div class="local-signup">
      <%= include('form') %>
    </div>
  </div>
</div>
```

Listing 2: form.html.ejs

```
<form action="/users" method="post">
  <p>o llena el formulario</p>
  <label for="name">
    <span>nombre</span>
    <input type="text"
      name="name" id="name" />
  </label>
  <label for="email">
    <span>e-mail</span>
    <input type="text"
      name="email" id="email" />
  </label>
  <label for="pwd">
    <span>password</span>
    <input type="password"
      name="pwd" id="pwd" />
  </label>
  <input type="submit"
    value="Enviar" />
</form>
```

## Crea tu cuenta

Elige uno de estos servicios

Gugl

Feisbuc

o llena el formulario

nombre

e-mail

contraseña

Enviar

### Pregunta 3: Cyber Monday

El Departamento de Cyber Compras (DCC) de “una conocida multitienda” se quiere preparar para el siguiente Cyber Monday (como hay uno de esos a cada rato aquí...) con la siguiente táctica: si un usuario ve una página de producto 10 veces continuas (del mismo o diferente producto), con no más de 1 hora de separación entre cada una, y sin agregar nada al carrito de compras, entonces en la siguiente página que vea le mostrarán un cupón de descuento de 20% adicional (con la esperanza de darle ese empujón final para comprar algo que no necesita).

Ellos ya tienen implementado:

- un listado de productos (por búsqueda o por categoría) en que cada producto lleva a una URL del tipo `/products/:id`
- la página de producto mencionada en el punto anterior
- un sistema de carrito de compra que lo que hace es almacenar los productos agregados al carrito en un arreglo `cart` de la sesión del cliente
- una vista en el *layout* de la aplicación que, si el atributo `coupon` del `state` del contexto del *request* es `true` entonces mostrará el mensaje de que el usuario tiene un cupón de descuento

Por lo tanto, sólo necesitan que implementes un *middleware* que se encargue de ir llevando registro de manera tal de determinar si hay que mostrarle el cupón al usuario y, en caso afirmativo, setear el estado de manera tal que el *layout* lo muestre. **Una restricción importante**, en todo caso, es que no quieren usar la base de datos para esto (por lo tanto, tampoco hay modelos que te ayuden).

**Tip:** `Date.now()` entrega la cantidad de milisegundos desde el 1 de enero de 1970 a las 00:00:00 (UTC)

### Pregunta 4: Una pregunta de miedo

La compañía de arriendo de disfraces Daunting and Creepy Costumes (DCC), está preparando una gran fiesta de disfraces de Halloween para la próxima semana, y para organizarla mejor quieren utilizar la Web. Actualmente tienen una aplicación Web desarrollada con el mismo template del curso, y que cuenta ya con todo lo necesario para manejar las entidades de usuarios y de disfraces. Lo que quieren agregar ahora es la funcionalidad de que cada usuario pueda asociarse con N disfraces, a modo de una *wishlist* de los disfraces que podría querer arrendar para esta fiesta. ¿Podrías ayudarlos? ¿O te da miedito? Si eres capaz de sobrellevar lo espeluznante de estos cambios, responde lo siguiente:

1. (3 puntos) Detalla los comandos que ejecutarías y/o las modificaciones en el directorio del proyecto que realizarías de manera de tener un **modelo** compatible con lo que se requiere (sólo modelo y base de datos, no router ni vista).
2. (3 puntos) Escribe la acción del router que recibirá un request para asociar un nuevo disfraz al “usuario actualmente logueado” (suponiendo que ellos siguieron y tu seguirás las prácticas recomendadas que hemos estado siguiendo durante el semestre)