Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2513 Tecnologías y Aplicaciones Web (I/2018) - Sección 1

Profesor: Gabriel Vidal Salazar

Interrogación 2

Fecha: Miércoles 16 de Mayo de 2018

Parte 1 (20%): Preguntas teóricas

Responde, de forma precisa, las siguientes preguntas:

1. (1 pt) ¿Cómo es posible tener estado en las aplicaciones web en un protocolo *stateless* como `HTTP`?
2. (1 pt) Si necesitaramos que las descargas de los archivos de nuestro proyecto fueran privadas, es decir, que el link que el usuario recibe sea sólo válido para él ¿Cómo debería implementarse tanto la carga como la descarga de los recursos?
3. (1 pt) ¿Cuál es el problema de guardar el id del usuario en la sesión?¿Cómo se puede solucionar?
4. (1 pt) ¿Por qué, al agregar una función al prototipo de una clase, no estamos creando una función por cada objeto creado?
5. (1 pt) ¿En qué se traducen las clases en JavaScript que han sido definidas con `class`?
6. (1 pt) ¿Cuáles son los pasos y elementos necesarios para poder enviar un correo desde el template de su proyecto?

Parte 2 (80%): Preguntas prácticas

NOTA: Parte A, B y C en hojas separadas.

Parte A (20%): Hoisting

Indica correctamente la salida de este programa. La funcion `printWithoutError` tiene el mismo funcionamiento que `console.log` salvo que, cuando algo lanzaría una excepción, la controla, coloca en consola `-- NOT DEFINED --` y el programa continúa su ejecución.

```
printWithoutError('Iniciando el programa');
printWithoutError(x);

x = 2;
cuak(x);

var y = 3;
printWithoutError(bark(y));

var test = (testInput) => printWithoutError(`Testing... ${testInput}`);
test(x + 5);

function cuak(times) {
  while(times-- > 0){
    printWithoutError('cuak!');
  }
}

test(x + y);

var bark = function(times) {
  while(times-- > 0){
    printWithoutError('bark!');
  }
}

bark(3);

printWithoutError(x);
var x;
```

Puntajes

- 6 pts por el correcto output
- En caso de omisión de una salida se descuenta puntaje proporcional
- En caso de agregar salidas que no corresponden, se descuenta una correcta por cada adicional

Parte B (40%): Funciones Síncronas y Asíncronas

Escribe el código del router que te permita crear una orden de compra(`order`) con los productos(`product`) asociadas a ella.

Para esto debes suponer lo siguiente:

- Hay una relación ya configurada de tipo N:N entre `order` y `product`.
- Los parámetros vienen en el body y corresponden a la información de la orden, en la propiedad `orderInfo`, además de un arreglo en la propiedad `productsIds` con los identificadores de los productos. Estos ya se han creado con antelación.

- En caso de cualquier error, debes incluir un mensaje de error en `ctx.messages.error` y redireccionar a la página `orders.list`.
- Si todo sale bien, debes incluir un mensaje al usuario en `ctx.messages.info`, enviar un correo (utilizando la función `sendOrderAlertEmail` que recibe el `ctx` y la orden) y redireccionar a `orders.list`

En resumen debes realizar lo siguiente:

1. Recibir los parámetros y crear la orden de compra (incluyendo los productos)
2. En caso de éxito se debe incluir el mensaje al usuario, enviar un correo y redireccionar al sitio antes indicado
3. En caso de error se debe incluir un mensaje al usuario y redireccionar al sitio

Debes utilizar las herramientas entregadas por el template del curso. Si necesitas crear un nuevo archivo indica bien su comienzo y término, colocando al inicio su ubicación.

Además, recuerda que a la herramienta de revisión de sintaxis no le gustan los `await` dentro de los ciclos. Debes considerar esto en tu solución.

A continuación el código que debes completar.

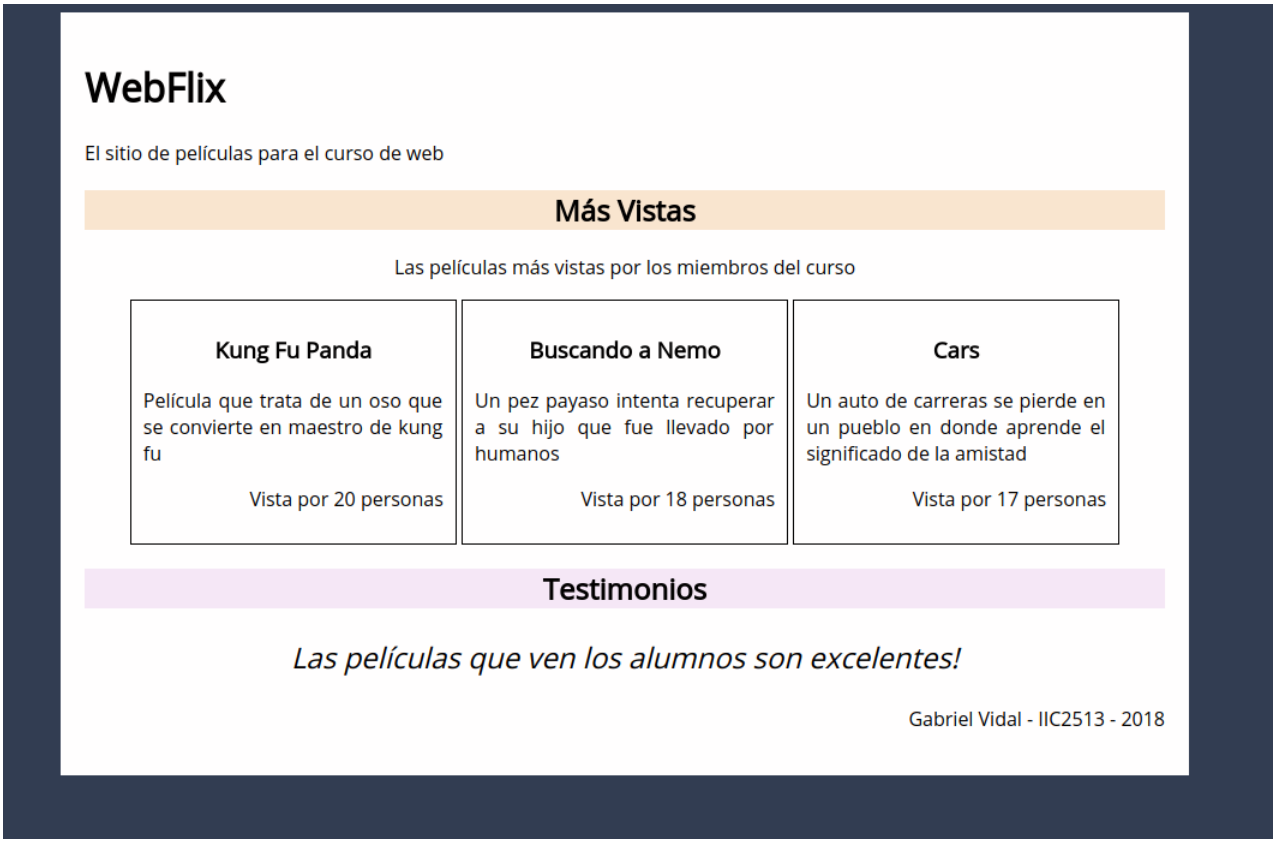
```
router.post('orders.new', '/', async (ctx) => {
  [...]
});
```

Puntajes:

- Punto 1 anterior:
  - 1 pt por crear la orden
  - 3 pts por asociar correctamente los productos a la orden
- Punto 2: 1pt
- Punto 3: 1pt

Parte C (40%): Javascript del lado del cliente

¿Te acuerdas de este sitio web?



Deberás agregar un selector, bajo el texto "El sitio de películas...", que te permita seleccionar entre distintos estilos para la página. Dentro de los estilos están:

- Ubuntu
  - Texto "Más vistas" y 'Testimonios" con color de fondo `#E95420`, color letra blanco
  - Películas más vistas con color de fondo `#f7f7f7`, color de letra `#667` y borde color `#cdcdcd`
- Rails
  - Texto "Más vistas" con color de fondo `#c52f24`, color letra blanco
  - Texto "Testimonios" con color de fondo `#fff9d8`, color letra negro
  - Películas más vistas con color de fondo `#d5e9f6`, color de letra negro y sin borde
- Node
  - Texto "Más vistas" y "Testimonios" con color de fondo `#eaf5e9`, color letra `#026e00`
  - Películas más vistas con color de fondo `#43853d`, color de letra blanco y sin borde

Todos los demás estilos relacionados con posición de texto y otros se deben mantener. Sólo se debe cambiar los estilos mencionados.

Si necesitas JavaScript, puedes utilizarlo o usar `jQuery`. Puedes suponer que ya se encuentra cargado en el sitio.

Puedes incluir otros archivos, indicando su nombre y su contenido; estos archivos se cargarán automáticamente. Además, puedes indicar los cambios de forma precisa, mencionando claramente el lugar donde va el cambio que quieres introducir.

A continuación te dejamos el código de la página de la imagen:

```
// index.html

<!DOCTYPE html>
<html>
  <head>
    <meta charset="UTF-8">
    <title>CSS Example</title>
    <link href="https://fonts.googleapis.com/css?family=Open+Sans" rel="stylesheet">
    <link rel="stylesheet" href="app.css">
  </head>
  <body>
    <div id="container">
      <header>
        <h1>WebFlix</h1>
<p>El sitio de películas para el curso de web</p>
      </header>

      <div id="most-viewed">
        <h2>Más Vistas</h2>
<p>Las películas más vistas por los miembros del curso</p>
<div>
  <div class="movie">
            <h3>Kung Fu Panda</h3>
            <p>Película que trata de un oso que se convierte en maestro de kung fu</p>
            <p class="quantity">Vista por 20 personas</p>
  </div>

          <div class="movie">
            <h3>Buscando a Nemo</h3>
            <p>Un pez payaso intenta recuperar a su hijo que fue llevado por humanos</p>
            <p class="quantity">Vista por 18 personas</p>
  </div>

  <div class="movie">
            <h3>Cars</h3>
```

```
            <p>Un auto de carreras se pierde en un pueblo en donde
              aprende el significado de la amistad</p>
            <p class="quantity">Vista por 17 personas</p>
  </div>
</div>
      </div>

      <div id="testimony">
        <h2>Testimonios</h2>
<p class="phrase">Las películas que ven los alumnos son excelentes!</p>
<p class="author">Gabriel Vidal - IIC2513 - 2018</p>
      </div>

    </div>
  </body>
</html>
```

```
// app.css

body {
  background-color: #2C3E50;
  font-family: 'Open Sans', sans-serif;
}

#container {
  margin: 0 auto;
  background-color: white;
  max-width: 900px;
  min-width: 600px;
  padding: 20px;
}

#most-viewed{
  text-align: center;
}

#most-viewed h2{
  background-color: #FAE5D3;
}

#most-viewed .movie {
  border: 1px solid black;
  display: inline-block;
  padding: 10px;
  width: 250px;
}

.movie p {
  text-align: justify;
}

.movie .quantity {
  text-align: right;
}

#testimony {
  text-align: center;
}

#testimony h2{
  background-color: #F4ECF7;
}

#testimony .phrase {
  font-size: 24px;
  font-style: italic;
}

#testimony .author {
  text-align: right;
}
```

Para ayudarte un poco, aquí hay algunas cosas que podrían ser útiles:

- El evento que se lanza, al tener el `DOM` cargado, es `DOMContentLoaded`
- En el caso de jQuery, se puede hacer con `$(function() { ... } )`
- `document.getElementById(id)`
- `document.getElementsByTagName(name)`
- `element.style`
- `element.setAttribute(name, value)`
- `element.getAttribute(name)`
- `element.addEventListener(name, function)`
- `$(selector).css(name)`
- `$(selector).css(name, value)`
- `$(selector).addClass(name)`. Una o múltiples separadas por espacio
- `$(selector).removeClass(name)`. Si no hay name, se eliminan todas.
- `$(selector).click(function)`.
- `$(selector).change(function)`.

Puntajes:

- HTML: 1.5 pts
- CSS: 2 pts
- JS: 2.5 pts