



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 3

Fecha: Miércoles 14 de Junio de 2018

Parte 1 (20%): Preguntas teóricas

Responde, de forma precisa, las siguientes preguntas:

1. (1 pt) ¿Por qué al cambiar el estado de un componente **React** se debe hacer con la función `this.setState()` en vez de modificar el estado directamente?
2. (1 pt) ¿Cuál es el flujo de TDD? ¿Cuáles son sus ventajas y desventajas?
3. (1 pt) ¿Qué práctica de desarrollo de *software* que involucra a encargados, desarrolladores, al sistema de control de versiones y a la misma aplicación es apoyado por la realización de *tests*?
4. (1 pt) ¿Cuál es la diferencia entre un **dumb** y un **smart component**?
5. (1 pt) ¿Cómo se soluciona, en una **API** cualquiera, que al hacer algunos cambios en ella no dejen de funcionar los clientes?
6. (1 pt) Indique qué es un *layout* líquido, por qué no basta para adaptar un sitio web en todos los dispositivos y qué herramienta de **CSS** ayuda en este proceso.

Parte 2 (80%): Preguntas prácticas

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

Parte A (10%): CSS

(6 pts) Completa el siguiente código para lograr lo mostrado en las siguientes imágenes:

Large screens:



Small screens:



HTML

```
1 <div class="topnav">
2   <a href="#">Home</a>
3   <a href="#">Link 1</a>
4   <a href="#">Link 2</a>
5   <a href="#">Link 3</a>
6 </div>
```

CSS

```
1 .topnav {
2   background-color: #333;
3 }
4
5 .topnav a {
6   /* Completar */
7   color: #f2f2f2;
8   text-decoration: none;
9 }
10
11 .topnav a:hover {
12   background-color: #ddd;
13   color: black;
14 }
15
16 @media screen and (max-width: 600px) {
17   /* Completar */
18 }
```

Parte B (30%): API

La universidad te ha pedido realizar una aplicación web para responder la encuesta docente. (Ahora sí podrás vengarte de ese profesor de web que te hizo sufrir con sus famosos HTML, CSS y JavaScript - React).

Tu contraparte te ha pedido que entregues un documento con el detalle de las rutas para una API que permita lo siguiente:

- Ver todas las encuestas docentes de un curso
- Ver/añadir/editar/borrar una encuesta docente de un curso

Para esto debes suponer lo siguiente:

- Por simplicidad, el curso tiene un identificador (número) y un nombre (texto)
- Las evaluaciones docentes tienen un identificador (número) y un texto (son anónimas)
- No te preocupes por la autorización a un cierto recurso.
- No es necesario que la API sea RESTful.

Preguntas:

1. (1 pt) Indica todo lo necesario para realizar este ejercicio (modelos, relaciones, etc).
2. (2 pts) Haz un listado de las rutas que necesitarías para cumplir con las acciones indicadas más arriba, mencionando claramente: método HTTP, la ruta del recurso y una pequeña descripción de para qué es la ruta.
3. (3 pts) Implementa la ruta para poder ver todas las evaluaciones docentes de un curso cualquiera. Debe considerar los casos donde podría haber error, por ejemplo, indicar un id inexistente.

Un ejemplo de salida:

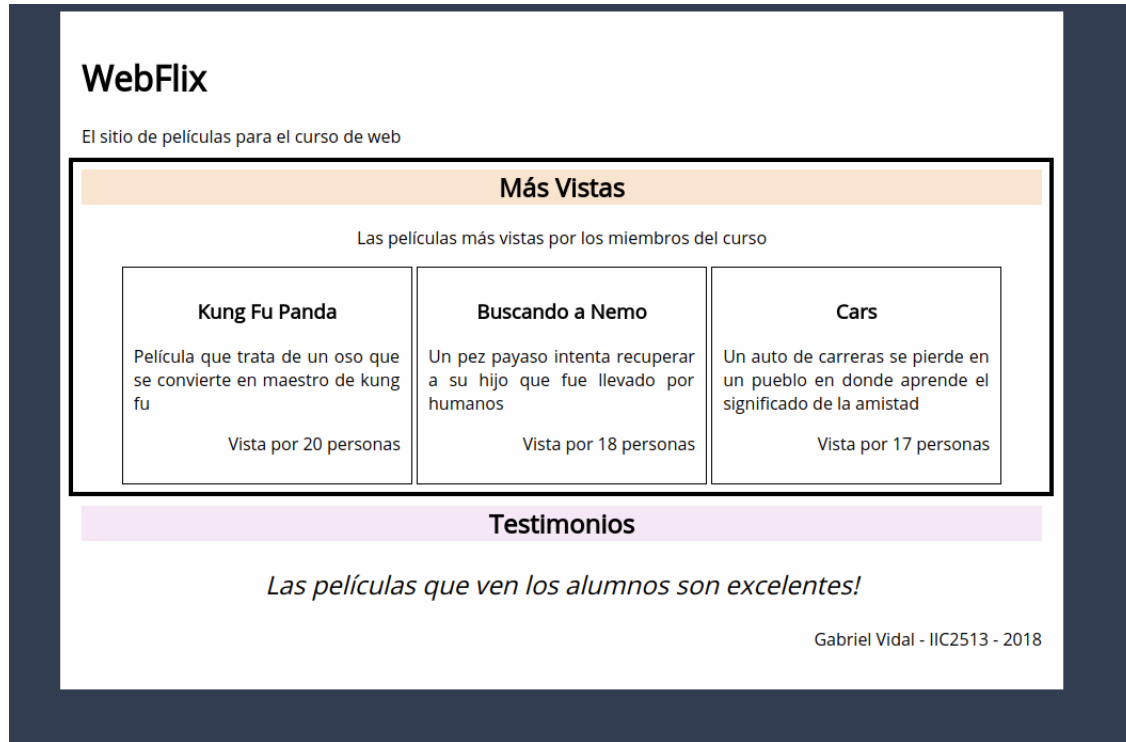
```
{
  id: 3456,
  name: 'Tecnologías y Aplicaciones Web',
  surveys: [
    {
      id: 843,
      text: 'Siempre nos hace programar en HTML y CSS',
    }
  ]
}
```

En caso de error (ejemplo):

```
{
  status: 'ERROR',
  error: 'id no encontrado',
}
```

A continuación tienes un ejemplo para el router:

```
1 router.post('orders.new', '/', async (ctx) => {
2   [...]
3 });
```



Parte C (60%): React

¿Te acuerdas de este sitio web?

Deberás transformar la sección encerrada en un rectángulo en componente(s) **React**. Para esto, la sección debe comportarse de la siguiente forma:

- Debes mostrar siempre el título.
- Mientras se esté cargando información debes informar de esto al usuario (puede ser a través de texto)
- Debes agregar un botón en cada una de las películas más vistas que diga 'Más información'. Una vez apretado alguno de los botones, debes esconder las tres películas más vistas y mostrar el texto de la película. Debe haber un botón, en esa sección, que oculte el texto y vuelva a mostrar las tres películas.

Para implementar lo anterior debes considerar lo siguiente:

- En alguno de tus componentes puedes recibir una propiedad llamada `mostWatched` que es un arreglo que contiene tres identificadores (números).
- No debes preocuparte por el estilo de/los componente(s).
- Puedes obtener el resumen de una película en la dirección

`http://estecursoesbkn.cl/movies/:id/summary`

Que retorna un objeto de la siguiente forma:

```
{
  id: 87654,
  name: 'Cars',
  description: 'Un auto [...]',
  people: 17,
}
```

- Puedes obtener el texto de una película en la dirección

`http://estecursoesbkn.cl/movies/:id/text`

Que retorna un objeto de la siguiente forma:

```
{
  id: 87654,
  name: 'Cars',
  fullText: 'Esta es una película animada [...]',
}
```

Aquí tienes un ejemplo de componente React:

```
1 import React from 'react';
2 import PropTypes from 'prop-types';
3 import faker from 'faker';
4
5 export default function TodoList(props) {
6   return (
7     <ul>
8       { props.items.map(item => <li key={`_${faker.random.uuid()}`} >{item}</li> ) }
9     </ul>
10   );
11 }
12
13 TodoList.propTypes = {
14   items: PropTypes.arrayOf(PropTypes.string).isRequired,
15 };
```

Puntajes:

- División de componente(s): 1 pt
- Uso de patrón para componente(s): 1 pt
- Código de componente(s): 4 pts
- **Bonus:** (1 pt) Agregar estilo a el/los componente(s)