



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 1

Fecha: Viernes 13 de Abril de 2018

#### Parte 1 (40%): Preguntas teóricas

Responde, de forma precisa, las siguientes preguntas:

**NOTA:** Parte A y B en hojas separadas.

##### Parte A

1. (2 pts) ¿Cuál(es) es(son) la(s) diferencia(s) entre `var` y `const` en JavaScript? ¿Para qué podría haber sido incluido el segundo?.
2. (2 pts) ¿Porqué en el siguiente caso `a` y `b` no se comporta de la misma forma? ¿En que caso, `a` se comportaría igual que el caso `b`?

```
const a = { id: 'IIC2513'};  
a.name = 'Tecnologías y Aplicaciones Web'; // funciona!
```

```
const b = 5;  
b = 6; // falla!
```

3. (2 pts) Uno de los elementos que conforman el protocolo HTTP son los códigos de respuesta ¿Para qué se utilizan? De al menos un ejemplo de uso.
4. (2 pts) En el juego del gato (o TicTacToe) la condición para que un usuario gane en la fila `i` (suponiendo que los valores en las casillas pueden ser `'x'`, `'y'` y `null` - este último si está vacía) es:

```
board[i][0] === board[i][1] && board[i][1] === board[i][2] && board[i][2]
```

¿Para qué es el último término? (`&& board[i][2]`)

5. (2 pts) ¿Cuál es la característica de CSS que nos permite que, al colocar la siguiente definición:

```
body {  
  font-family: Arial, Helvetica, sans-serif;  
}
```

se aplique en todo el documento?

## Parte B

6. (2 pts) Comenta si estás a favor o en contra de la siguiente afirmación: Una función que tiene `async` en su definición sólo indica que es de tipo asíncrona, no tiene ninguna implicancia en el retorno ni en el cuerpo de esta.
7. (2 pts) Dé un ejemplo de cómo puede ocurrir que, en dos ramas distintas de su repositorio, puedan tener definidos dos esquemas de base de datos. Aplica tu respuesta al *template* de proyecto del curso.
8. (2 pts) Indica cual es el rol que tiene HTML, CSS y JavaScript en la interfaz de usuario
9. (2 pts) Explica cómo es el proceso desde que un *request* llega al servidor hasta que es respondido al cliente. Basa tu respuesta en el proyecto semestral.
10. (2 pts) Comenta si es correcto o incorrecto, e indique un ejemplo o contraejemplo: Todas las redirecciones se pueden hacer a nivel de DNS, para este tipo de labores no es necesario el protocolo HTTP.

## Parte 2 (60%): Preguntas prácticas

**NOTA:** Parte A y B en hojas separadas.

### Parte A (60%): Funciones Síncronas y Asíncronas

Te han encargado desarrollar un proyecto que muestre como resultado la información de los planetas del sistema solar. Para esto, han definido un objeto `planetInfo` para ti con las siguientes funciones:

- `getDistanceToSun(planetName, cb)`: Función asíncrona que recibe el nombre de un planeta y, una vez que obtiene la distancia al sol (en millones de kilómetros), llama a la función entregada como *callback* con el resultado como parámetro.

Ejemplo:

```
1 planetInfo.getDistanceFromSun('Earth', distance => console.log(distance));  
2 // Output: 150
```

- `getSatellitesNames(planetName, cb)`: Función asíncrona que recibe el nombre de un planeta y, una vez que obtiene los nombres de los satélites (en un arreglo), llama a la función entregada como *callback* con el resultado como parámetro.

Ejemplo:

```
1 planetInfo.getSatellitesNames('Earth', moons => console.log(moons));
2 // Output: ['Moon']
```

- `getSatelliteInfo(satelliteName)`: Función **síncrona** que recibe el nombre de un satélite y retorna su información.

Ejemplo:

```
1 const moonInfo = planetInfo.getSatelliteInfo('Moon');
2 console.log(moonInfo);
3 /* Output:
4  * {
5  *   name: 'Moon',
6  *   radius: 1.737,
7  *   gravity: 1.62
8  * }
9  */
```

Tienes una primera versión de tu programa, la cual es:

```
1 function getPlanetSummary(planet) {
2   console.log('Obteniendo la informacion...');
3
4   const planetSummary = { name: planet };
5
6   planetInfo.getDistanceFromSun(planet, (distance) => {
7     console.log('Distancia al sol: ${distance}');
8     planetSummary.distanceFromSun = distance;
9   });
10
11   planetInfo.getSatellitesNames(planet, (satellites) => {
12     console.log('Satellites: ${satellites}');
13     planetSummary.satellites = satellites;
14   });
15
16   console.log(planetSummary);
17
18 }
```

### Preguntas:

- (1 pt) Indica el output del programa ¿Es siempre el mismo? ¿Por qué?
- (3 pts) Decidiste que no quieres usar *callbacks*, quieres usar promesas. Ahora el objeto `planetInfo` tiene dos métodos nuevos que son:
  - `getDistanceFromSunPromise(planet)`
  - `getSatellitesNamesPromise(planet)`

Ambos análogos a los anteriores, siendo la única diferencia que entregan una promesa de resultado. Modifica el código de `getPlanetSummary` para que la función **retorne** una promesa. Al ser resuelta, por ejemplo para el valor `'Mars'`, entrega un objeto que tiene la siguiente forma;

```
{
  name: 'Mars',
  distanceFromSun: 228,
  satelllites: [
    {
      name: 'Phobos',
      radius: 11,
      gravity: 0.0057
    },
    {
      name: 'Deimos',
      radius: 6
      gravity: 0.003
    }
  ]
}
```

- (2 pts) Implementa el mismo código que en la pregunta anterior, pero ahora utilizando `async/await`.

## Parte B (40%): HTML + CSS

Deberás escribir el código HTML y CSS necesario para lograr un resultado similar al que se encuentra a continuación. Sólo escriba lo que va dentro del tag `body` para el HTML. Algunos detalles de estilo relevantes:

- El fondo del sitio tiene color `#2C3E50`
- El contenedor tiene un ancho mínimo de 600 pixeles y un ancho máximo de 900 pixeles; padding de 20px, centrado y de color blanco.
- El título “Más Vistas” está centrado y tiene color de fondo `#FAE5D3`
- Para las películas: El título está centrado, la descripción justificada, la cantidad de personas que ha visto la película alineado a la derecha, padding de 10px, ancho de 250px y borde marcado. Todas las películas están una al lado de la otra.
- El título “Testimonios” está centrado y tiene color de fondo `#FAE5D3`
- El testimonio está centrado, con letra cursiva y de tamaño de 24px.

# WebFlix

El sitio de películas para el curso de web

## Más Vistas

Las películas más vistas por los miembros del curso

<b>Kung Fu Panda</b> Película que trata de un oso que se convierte en maestro de kung fu Vista por 20 personas	<b>Buscando a Nemo</b> Un pez payaso intenta recuperar a su hijo que fue llevado por humanos Vista por 18 personas	<b>Cars</b> Un auto de carreras se pierde en un pueblo en donde aprende el significado de la amistad Vista por 17 personas
--	--	--

## Testimonios

*Las películas que ven los alumnos son excelentes!*

Gabriel Vidal - IIC2513 - 2018

Debe considerar lo siguiente:

- Sea ordenado, esto también será evaluado.
- El uso de clases, ids y selectores de CSS también es un punto a evaluar.

Puntajes:

- (1pt — 0.5pts HTML - 0.5 pts CSS) Header, contenedor y fondo
- (3pts — 1 pt HTML - 2 pts CSS) Sección más vistas
- (2pts — 1 pt HTML - 1 pt CSS) Sección testimonios