

Universidad de La Laguna. Escuela Técnica Superior de Ingeniería Informática  
Tercero del Grado de Informática  
**DESARROLLO DE SISTEMAS INFORMÁTICOS: 3ª PARTE**  
16/05/2017 7 páginas

Nombre: \_\_\_\_\_  
Alu: \_\_\_\_\_ GitHub Id: \_\_\_\_\_ GitHub Team: \_\_\_\_\_

### Preguntas de Strategy Pattern

1. Describa que se entiende por *Code Smell*
2. ¿En que consisten cada uno de estos smells?
  1. Duplicated code
  2. Contrived complexity
  3. Large class
  4. Feature envy
  5. Inappropriate intimacy
  6. Refused bequest (herencia negada)
  7. Lazy class / Freeloader
  8. Excessive use of literals
  9. Large Cyclomatic complexity
  10. Downcasting
    1. In a Java-like language, when learning about polymorphism you commonly see something like this

```
class Base { ... }
```

```
class Derived extends Base { ... }
```

Is (Derived\*)base\_object; an upcast or downcast?
  11. Too many parameters
  12. Long method
  13. Excessively long identifiers
  14. Excessively short identifiers
  15. Excessive return of data
  16. Switch Smell
3. Enuncie el OCP: Open/Closed Principle
4. ¿Que se entiende por *Design Pattern*?
5. Explique como se resuelve el Switch Smell usando el *Strategy Pattern*

## Preguntas de Como crear y publicar un paquete npm

10. ¿Cuales son los pasos para escribir y publicar un paquete **npm**?
11. ¿Cómo instalo una versión anterior de un package npm?
12. ¿Cómo encuentro la versión de un paquete NodeJs instalado?
13. ¿Que se entiende por **scope** de un paquete npm?

1. ¿Cual es la notación para referenciar un paquete **somepackage** que se aloja en el ámbito **somescopes**?
2. ¿Como se hace un **require** de un paquete **somepackage** que se aloja en el ámbito **somescopes**?
3. Cualquier usuario npm puede publicar sus módulos en cualquier ámbito: ¿Verdadero o falso?
4. ¿Cómo puedes hacer para tener varios ámbitos npm?
5. ¿Cual es el tipo de acceso por defecto de mi paquete con ámbito: público o privado?
6. ¿Cómo indico a la hora de publicar mi paquete con ámbito que quiero que sea de acceso público?
7. ¿Que hace este comando?

```
npm login --registry=http://reg.example.com --scope=@myco
```

8. ¿Que hace este comando?

```
npm init --scope=username
```

9. ¿Que hace este comando?

```
npm config set scope username
```

14. ¿Que hace este comando?

```
npm version patch -m "Upgrade to %s for reasons"
```

1. Cuando el comando anterior se ejecuta en un git repo, ¿Crea un commit? ¿Crea un tag?
2. If **preversion**, **version**, or **postversion** are in the scripts property of the package.json, they will be executed as part of running npm **version**. Take the following example:

```
"scripts": {  
  "preversion": "npm test",  
  "version": "npm run build && git add -A dist",  
  "postversion": "git push && git push --tags && rm -rf build/temp"  
}
```

Describe the set of actions that will happen when running **npm version**

## JsDoc

1. ¿Cual es la sintaxis de los comentarios de documentación en JSDoc? ¿Como empiezan?
2. ¿Donde debo ubicar un comentario de documentación JSDoc para documentar un cierto código?
3. Special *JSDoc tags* can be used to give more information. For example,
  1. If the function is a constructor for a class, you can indicate this by adding a \_\_\_\_\_ tag (fill the gap)
  2. ¿Cómo se documenta un parámetro de una función?
  3. ¿Que hace esta documentación JSDoc?

```
/**
 * See {@tutorial gettingstarted}
 */
function myFunction() {}
```

4. ¿Que hace esta documentación JSDoc?

```
/**
 * See {@link MyClass} and [MyClass's foo property]{@link MyClass#foo}.
 * Also, check out {@link http://www.google.com|Google} and
 * {@link https://github.com GitHub}.
 */
function myFunction() {}
```

## Preguntas de Repaso de Pruebas con Mocha, Chai y Should

1. ¿Como creamos el directorio con el esqueleto inicial para las pruebas con mocha?
2. En este ejemplo se usa Chai assert. Rellene lo que falta en estas pruebas del código del conversor de temperatura:

```
var assert = chai._____;

suite('temperature', function() {
  test('[1,{a: 2}] == [1,{a: 2}]', function() {
    assert._____( [1, {a:2}], [1, {a:2}] );
  });
  test('5X = error', function() {
    original.value = "5X";
    calculate();
    assert._____(converted.innerHTML, /ERROR/);
  });
});
```

3. Este es un fichero `test/index.html` apto para ejecutar las pruebas con Mocha y Chai en la práctica de la Temperatura en un navegador. Rellene las partes que faltan
- Sugerencias: El id es el usado por mocha para producir su salida de las pruebas, es necesario cargar `chai` y `mocha` y establecer el estilo de pruebas (`tdd`, `bdd`, etc.) y por último ejecutar `mocha`

```
[~/srcPLgrado/temperature(karma)]$ cat tests/index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Mocha</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="mocha.css" />
  </head>
  <body>
    <div id="____"></div>    <!-- para la salida de las pruebas -->
    <input id="original" placeholder="32F" size="50">
    <span class="output" id="converted"></span>

    <script src="_____"></script>
    <script src="_____"></script>
    <script src="../temperature.js"></script>
    <script>mocha.____('___')</script>
    <script src="tests.js"></script>

    <script>
      mocha.____();
    </script>
  </body>
</html>
```

4. Rellene las partes que faltan del fichero con las pruebas TDD en Mocha y Chai para la práctica de la temperatura:

```
[~/srcPLgrado/temperature(karma)]$ cat tests/tests.js
var assert = _____.assert;

_____('temperature', function() {

  _____('32F = 0C', function() {
    original.value = "32F";
    calculate();
    assert._____(converted.innerHTML, "0.0 Celsius");
```

```
    });
  });
```

5. ¿Como puedo ejecutar las pruebas escritas usando Mocha y Chai usando el comando `npm test`?. (no se asume el uso de Karma en esta pregunta) Explique como hacerlo.
6. El siguiente ejemplo corresponde al ejemplo de pruebas que vimos para la renderización de una tabla correspondiente al capítulo *The Secret Life Of Objects* que usa `mocha` y `should`. Rellena las partes que faltan:

```
-----("drawTable", function() {
  __("must draw the checkerboard correctly", function() {
    /* There are 5 columns and 5 rows and a white space between columns*/
    drawTable(checkerboard()).should._____(/^(#[ ]{2}(\s|$)){5}){5}$/);
  })
});
```

## Preguntas de Repaso de Karma

1. Explique como funciona Karma
2. ¿Con que comando puedo crear el fichero de configuración de Karma?
3. ¿Que debemos poner en la entrada `frameworks` de karma para el ejemplo de la temperatura?

```
frameworks: ['____'],
```

4. La librería/plugin `karma-mocha` provee el adapter de Karma para Mocha. ¿Como le pasamos opciones para configurar Mocha desde Karma? Rellene las partes que faltan:

```
client: {
  args: ['--grep', 'pattern'], // solo pruebas que casan con pattern
  mocha: {
    __: '___'
  }
},
```

5. Explique que debe ponerse (y que no) en la sección `files` del fichero de configuración de Karma ¿Donde son cargados dichos ficheros?:

```
files: [ ... ],
```

6. Los preprocesadores en Karma nos permiten procesar los ficheros en `files` antes de que sean cargados en el navegador.

```
preprocessors = {
  '**/*.coffee': 'coffee',
  '**/*.html': 'html2js'
};
```

¿Que hace el preprocesador ``html2js``? ¿Que hace el preprocesador ``coffee``?

7. Complete la función `setup` de las pruebas de la práctica de la temperatura con Mocha, Chai y Karma:

```
setup(function(){
  if (typeof _____ !== 'undefined') {
    document.body.innerHTML = _____['tests/test.html'];
    original = document._____('original');
    converted = document._____('converted');
  }
});
```

¿Como se llama la variable en la que se guardan los HTML de los ficheros cargados en los navegadores?

8. ¿Que es PhantomJS? ¿Como funciona?

## Preguntas de Repaso de Mongo y Mongoose

1. ¿Que diferencia hay entre una tabla de una base de datos relacional y una collection de MongoDB?
2. ¿Como se llama la primary key de un documento MongoDB?
3. Escriba el código Mongoose para conectarse a la MongoDB database `chuchu`
4. Escriba el código Mongoose para crear un esquema para modelar una colección de `usuarios`
5. Escriba el código Mongoose para crear un modelo a partir del esquema anterior
6. Escriba el código Mongoose para crear un documento usuario y guardarlo en la base de datos
7. Escriba el código Mongoose para buscar por un documento usuario con nombre `'Pepe'`

## Preguntas de Repaso de localStorage

1. ¿Que diferencias y que similitudes hay entre los cookies y localStorage?
2. ¿Que diferencias hay entre localStorage y sessionStorage?
3. ¿En que consiste las reglas de segregación conocidas como *same origin policy*? ¿Como se relacionan con localStorage?
4. ¿Como guardo en localStorage el contenido de una variable **chuchu**? ¿Como leo el contenido almacenado en localStorage con clave **chuchu**?
5. ¿Como borro una clave almacenada en localStorage?
6. 1. ¿Que es LocalStorage? ¿Que hace la siguiente línea?

```
if (window.localStorage) localStorage.original = temp;
```

2. ¿Cuando se ejecutará esta callback? ¿Que hace?

```
window.onload = function() {  
    // If the browser supports localStorage and we have some stored data  
    if (window.localStorage && localStorage.original) {  
        document.getElementById("original").value = localStorage.original;  
    }  
};
```