

Programación Web Avanzada

PEC 1: Hello World! (en Vue.js)

Presentación

En esta primera PEC se abordarán conceptos básicos de JavaScript como Asincronía, DOM y los principios básicos de funcionamiento de Vue.js. Para ello usaremos el entorno de desarrollo creado previamente, por lo que es necesario realizar la PEC 0.

Conocimientos, Habilidades y Competencias

En esta PEC se desarrollan los siguientes conocimientos (K), habilidades (S) y Competencias (C) del Grado en Multimedia 2023:

- **K1.** Identificar y diferenciar los elementos de la teoría de la multimedia para analizar, conceptualizar y diseñar productos interactivos multimedia.
- **S3.** Utilizar de manera adecuada los lenguajes de programación, las herramientas de desarrollo y las tecnologías disponibles para el análisis, el diseño y la implementación de aplicaciones multimedia.
- **S7.** Analizar y sintetizar información para evaluar soluciones tecnológicas y elaborar propuestas de proyectos multimedia teniendo en cuenta los recursos, las alternativas disponibles y las condiciones de mercado.


Objetivos

Los Objetivos concretos de esta PEC son:

- Conocer conceptos básicos de asincronía en JavaScript
- Poner en práctica conceptos básicos de JavaScript
- Familiarizarse con el funcionamiento de Vue.js

Recursos de lectura y Guía de Aprendizaje

A continuación, podéis encontrar los recursos de lectura que consideramos importantes para revisar y estudiar con el fin de completar esta PEC.

 *El recurso principal que seguiremos en el curso es el material de **Azaustre Rodríguez, C. (2024) "Conceptos y técnicas avanzadas de programación web"**, disponible en el aula. Sin embargo, dada la alta volatilidad de las tecnologías web, es difícil recomendar un único recurso de aprendizaje fijo para todos los semestres del curso. Por esta razón, en cada semestre seleccionamos las fuentes más adecuadas que consideramos útiles para presentar los conceptos de la asignatura.*

Además, como guía de aprendizaje, indicamos las secciones (o capítulos específicos) que debéis leer atentamente para la correcta realización de esta **PEC**. Para cada una de las partes recomendadas, también incluimos un comentario para motivar nuestra recomendación. Por supuesto, si deseáis profundizar más, podéis revisar las secciones no recomendadas en esta lista de cada recurso.

El concepto de Framework

 **Book: Azaustre Rodríguez, C. Conceptos y técnicas avanzadas de la programación web. FUOC, 2024**

- 1. Estructura Básica y Funcionalidades de un *Framework Web*.

En este primer capítulo, profundizaremos en el concepto de *framework* web y el uso de `<template>`.

Repaso de Javascript

Podemos refrescar o ampliar nuestros conocimientos en JavaScript con el siguiente recurso:

 **Libro: Ethan Brown. [Learning JavaScript, 3rd Edition](#). O'Reilly, 2016**

- Capítulo 2. JavaScript Development.
Partes de este capítulo no te resultarán nuevas, ya que lo hemos practicado en la PEC 0. Su lectura te ayudará a afianzar los conceptos relacionados con el desarrollo de aplicaciones Web.
- Capítulo 3. Literals, Variables, Constants, And Data Types
Presta atención sobre todo a los diferentes tipos de datos que ofrece JavaScript. Aunque te puedan parecer muy numerosos, conocer sus diferencias te permitirá programar en JavaScript de forma más eficiente y adecuada a la solución Web que tengas entre manos.
- Capítulo 4. Control Flow
Como todo lenguaje de programación, JavaScript también incorpora estructuras de control para gestionar el flujo de ejecución. Este capítulo hace un repaso de lo que podrás encontrar

en JavaScript. Los últimos apartados del capítulo incluyen algunos patrones que te resultarán de utilidad.

- **Capítulo 5. Expressions and Operators**
Al igual que con las estructuras de control, las expresiones y los operadores son una parte fundamental de los lenguajes de programación. En este capítulo se hace un repaso a las expresiones y operadores de JavaScript con todas sus peculiaridades.
- **Capítulo 6. Functions**
En este capítulo se recorren todas las posibles modos de generación de funciones en JavaScript, tanto en su forma básica como las conocidas *arrow function*. A su vez, se describe y explica uno de los conceptos básicos de JavaScript: la palabra **this**, por lo que su lectura es de especial importancia.
- **Capítulo 7. Scope**
El **scope** determina dónde y cuándo las variables, constantes y funciones se considera que están definidas. Este es otro de los conceptos básicos a tener en cuenta en el desarrollo en JavaScript; sobre todo tras la introducción de ES6.
- **Capítulo 8. Arrays and Arrays processing**
Una de las principales estructuras de datos en programación son los *arrays*. Como veremos en la asignatura, se utilizan con mucha frecuencia. En este capítulo se revisan las diferentes formas de crear *arrays* y su manipulación en JavaScript.
- **Capítulo 9. Objects And Object-Oriented Programming**
Otra de las estructuras básicas son los objetos, pero a su vez esta estructura define un paradigma específico de programación: La programación orientada a objetos. En este capítulo se hace un resumen de los principios básicos de este paradigma y se muestra la forma de manipular objetos en JavaScript.
- **Capítulo 18. JavaScript in the Browser**
Una vez comprendido todos los elementos básicos de JavaScript, es necesario ubicarlo dentro de su ámbito natural: el navegador. En este capítulo se cubren los elementos básicos que forman parte de la navegación web y su relación con JavaScript.

Empezando con Vue

Vue docs

- The application instance: <https://vuejs.org/guide/essentials/application.html>

Esta página describe los principios básicos sobre los que se asienta una aplicación en Vue, cómo se genera una instancia, cómo se usa un componente, y cómo se renderiza contenido en una página.

- Template syntax: <https://vuejs.org/guide/essentials/template-syntax.html>


Este apartado de la documentación describe los elementos básicos de un template en Vue, cómo se renderiza contenido, cómo se llaman a funciones, qué son las directivas y cómo se usan.

- Component basics: <https://vuejs.org/guide/essentials/component-basics.html>

Los componentes son el elemento básico sobre el que se construye cualquier aplicación en Vue. En este apartado se describen los elementos básicos de un componente, cómo se invocan, y cómo se le pasan parámetros.

Vite docs

- <https://vitejs.dev/guide/>
Documentación básica de Vite.

 *Vuejs está en su tercera versión, por lo que muchos desarrolladores Web tienen que estudiar las características más recientes para poder desplegar aplicaciones Web en esta nueva entrega del framework. Este es otro ejemplo de la volatilidad de las tecnologías Web y la capacidad de todo desarrollador Web de estar preparado al cambio.*

En nuestro caso, desarrollaremos las PECs en Vuejs 3, por lo que os recomendaremos un conjunto de recursos de aprendizaje más actualizados, siempre que sea posible. En cualquier caso, como algunos conceptos son parecidos (o iguales) a Vuejs 2, puede que también nos basemos en la documentación de dicha versión.

Formato de entrega

A través del campus deberéis entregar **un único fichero PDF** con las respuestas a las preguntas realizadas. El documento deberá contar como mínimo con el nombre del alumno en el encabezado.

Puntuación

- La puntuación de cada ejercicio está detallada en su enunciado, **sobre un total de 10 puntos**.
- Se valorará la concreción y la precisión de las respuestas por encima de la extensión. No es necesario grandes explicaciones en ninguna pregunta. Con una descripción somera destacando los puntos claves de cada respuesta es suficiente.
- Para las preguntas teóricas, es sumamente importante incluir las **referencias bibliográficas** de todos los conceptos consultados, de manera clara y no genérica. Por ejemplo, no sería válido incluir un enlace directo a la página principal de una web, o mencionar un libro sin indicar el capítulo. Puede consultarse la Biblioteca de la UOC para aprender cómo citar correctamente.

Consultas

En caso de que tengáis que consultar algo mediante el foro o por correo electrónico, copiad vuestro código a una plataforma online y enviad el link para evitar problemas con el correo de la UOC (ya que elimina los ficheros .js y .vue para evitar inyectar código malicioso así como la indentación del código). Os recomendamos utilizar cualquiera de estas dos opciones:

- [Codepen](#) (para sencillos snippets de código)
- [CodeSandBox](#) (para ejercicios más complejos)

Nota: En caso de publicar algún código en el foro, deberán ser consultas genéricas y no directamente soluciones a los ejercicios. Hacer accesible soluciones de ejercicios a otros compañeros, aunque pueda no tener una intención directa, se considerará copia y se penalizará académicamente a nivel de asignatura.

Propiedad intelectual y plagio

La Normativa académica de la UOC dispone que el proceso de evaluación se cimenta en el trabajo personal del estudiante y presupone la autenticidad de la autoría y la originalidad de los ejercicios realizados.

En esta PEC no está permitido el uso de herramientas de Inteligencia Artificial. En el plan de estudios y en [la web sobre integridad académica y plagio de la UOC](#) encontrarás más información sobre las faltas y las consecuencias del plagio. Recuerda que si utilizas material externo siempre tienes que hacer referencia a las fuentes, también puedes leer <https://biblioteca.uoc.edu/es/pagina/Como-citar/>

El estudiante será calificado con un suspenso (D/0) si se detecta falta de originalidad en la autoría de alguna prueba de evaluación continua (PEC) o final (PEF), sea porque haya utilizado material o dispositivos no autorizados, sea porque ha copiado textualmente de internet, o ha copiado apuntes, de PEC, de materiales, manuales o artículos (sin la cita correspondiente) o de otro estudiante, o por cualquier otra conducta irregular.

Cuestiones



Las respuestas que requieran consultar material externo debe incluir referencias a las fuentes consultadas.

Ejercicio 1. (1 punto)

Escribe un informe sobre JavaScript en el desarrollo Web y el uso de frameworks. El informe deberá responder a las siguientes preguntas:

- ¿Cómo pasó JavaScript de ser un simple lenguaje de scripting a convertirse en el núcleo de las aplicaciones web modernas?
- ¿Qué problemas intentaron resolver los primeros frameworks como jQuery y cómo evolucionaron hacia soluciones como React, Vue o Angular?
- ¿Qué es una SPA (*Single Page Application*) y cómo cambió el desarrollo web?

El informe no puede tener más de 600 palabras.

Ejercicio 2. (0.5 puntos)

¿Qué es el DOM virtual? ¿Qué relación tiene con Vue 3, React y otros frameworks modernos?

Ejercicio 3. (0.5 puntos)

Enuncia y describe brevemente los tipos de ámbitos (*scope*) que puede tener una variable en ES6 (JavaScript moderno). Pon ejemplos de cada uno e ilustra su comportamiento

Ejercicio 4. (0.5 puntos)

Explica qué es la palabra clave "this" en JavaScript e ilústralo con algunos ejemplos. Debe incluir al menos 3 ejemplos.

Ejercicio 5. (0.5 puntos)

¿Qué significa el concepto de "hoisting" en JavaScript? ¿Cómo afecta a las variables declaradas con var, let y const? ¿Y a las funciones? Proporciona un código de ejemplo donde se aplique correctamente y otro donde daría error, y analiza su comportamiento.

Ejercicio 6. (0.5 puntos)

¿Cuál es el resultado de los siguientes códigos JavaScript y por qué?

```
for (var j = 1; j < 4; j++) {
  setTimeout(() => console.log(j), 100);
}

for (let k = 1; k < 4; k++) {
  setTimeout(() => console.log(k), 100);
}
```

Ejercicio 7. (0.5 puntos)

¿Cuál es el resultado de los siguientes códigos JavaScript y por qué? ¿Cómo se llama el concepto en cuestión?

```
console.log(2 * '3' + '3' - '2');
```

Ejercicio 8. (0.5 puntos)

Utilizando el siguiente código, responde a la pregunta:

```
let a = [1, 2];
let b = new Array(1, 2);

console.log(a == b);
console.log(a === b);
```

Explica por qué `a == b` se evalúa como falso y `a === b` se evalúa como falso.

Ejercicio 9. (0.5 puntos)

¿Cuál es la salida del siguiente código JavaScript y por qué?

```
const sym1 = Symbol('foo');
const sym2 = Symbol('foo');

console.log(sym1 === sym2);

const bool1 = Boolean(0);
const bool2 = Boolean('false');

console.log(bool1 === bool2);
```

Ejercicio 10. (0.5 puntos)

Ejecuta el siguiente código en una consola JavaScript y observa el resultado que se obtiene. ¿Por qué ocurre este comportamiento? ¿Cómo solucionarías este problema?

```
let obj1 = { name: "Alice" };
let obj2 = obj1;

obj2.name = "Bob";

console.log(obj1.name);
console.log(obj2.name);
```

Ejercicio 11. (0.5 puntos)

Respecto al siguiente código JavaScript:

```
function divide(a, b) {
    return a / b;
}

const result = divide('6', '3');
console.log(result);
```

Determina el tipo de la variable “result” después de que se ejecute este código y comenta el resultado.

Ejercicio 12. (0.5 puntos)

Dado el siguiente código en JavaScript:


```
const array = [1, 2, 3, 4];
const newArray = array.map((num) => num * 2);
console.log(newArray);
```

1. ¿Cuál será la salida del código?
2. ¿Cómo se diferencia `map()` de `forEach()`?
3. Proporciona un ejemplo donde `forEach()` sea más útil que `map()`.

Ejercicio 13 (0.5 puntos)

Atendiendo al contenido del fichero `HelloWorld.vue`, que se crea automáticamente cuando generamos el proyecto en Vue con Vite.

- ¿Cuál es el valor de `{{ msg }}`? ¿Por qué?
- ¿De qué tipo es ese parámetro? ¿Por qué?
- ¿Qué significa "scoped" en el apartado de estilos?

Ejercicio 14 (0.5 puntos)

Después de instalar un proyecto Vue con Vite, vemos que se genera una estructura de carpetas con varios archivos clave. Explica brevemente la función de los archivos `main.js` (o `main.ts`), `App.vue` y `vite.config.js`.

Ejercicio 15 (0.5 puntos)

¿Es posible usar preprocesadores CSS como SCSS o Less dentro de los estilos de un archivo `.vue`? ¿Cómo lo harías, suponiendo que ya están instaladas las dependencias?

¿Qué significa *setup* en la declaración `<script>`? Explica brevemente el concepto que hay detrás en Vue 3 y la diferencia entre usarlo y no usarlo.

Ejercicio 16 (0.5 puntos)

Modifica el archivo `HelloWorld.vue` añadiendo en la plantilla una cabecera `<h2>` con tu nombre y apellidos. Adjunta una captura de pantalla y describe el procedimiento.

Ejercicio 17 (0.5 puntos)

¿Cuál sería el procedimiento necesario para añadir un nuevo componente al archivo `App.vue`? Enumera y describe los pasos, y adjunta captura de pantalla si es necesario para acompañar la explicación.

Ejercicio 18 (0.5 puntos)

- ¿Qué comando es necesario ejecutar en la consola para ejecutar el entorno de desarrollo?
- ¿Qué comando es necesario ejecutar en la consola para generar una versión de la aplicación lista para desplegar en producción?

Ejercicio 19 (0.5 puntos)

Ejecuta la aplicación Vue con el comando correspondiente del terminal. Adjunta una captura de pantalla de la aplicación de Vue en ejecución.