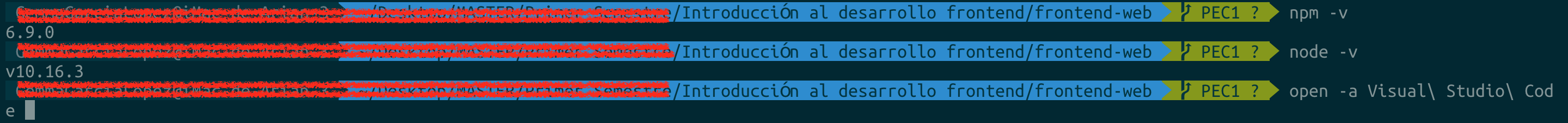
PEC1

ENTORNO DE DESARROLLO

  
PREGUNTAS TEÓRICAS

**1. La aparición de HTML5/CSS3/JS ha supuesto el nacimiento del desarrollo front-end moderno (0.75 puntos)**

a) ¿Cuál es la ventaja del uso de etiquetas semánticas? Nombra y explica al menos 3 de ellas.

**Beneficios en el SEO**Elementos como los formularios o menús principales pueden resultar fáciles de localizar a simple vista cuando el contenido se renderiza en el navegador. Sin embargo, esta tarea no es tan obvia para los motores de búsqueda.  
Las actualizaciones en el algoritmo de Google no son públicas, pero según los expertos, el algoritmo del buscador valora positivamente (indexa mejor los resultados) un marcado correcto.  
De esta manera, los motores de rastreo pueden reconocer mejor la importancia de los distintos elementos web cuando disponen de un etiquetado que se ajusta al contenido.  
  
**Accesibilidad**  
Enlazando con el punto anterior, una buena sintaxis hace que los contenidos sean comprensibles y coherentes para personas que utilicen lectores de pantalla, como por ejemplo: JAWS (Job Access with Speech), NVDA (Non-Visual Desktop Access), Voiceover (OSx y IOS) o Talkback (Android).  
  
**Reducción del código**HTML5 ofrece mayor soporte al contenido, como es el ejemplo del video y el audio, mientras que ninguno de estos formatos recibía soporte en versiones anteriores. En este sentido, HTML5 también ofrece nuevos controles de formulario.   
  
**Mejora la comunicación de datos**  
Según W3C, una Web Semántica: "Permite que los datos se compartan y reutilicen en aplicaciones, empresas y comunidades".

b) Cita al menos 3 APIs HTML5 y explica brevemente su funcionalidad.

**Drag and Drop**Permite arrastrar y soltar elementos dentro y entre las ventanas del navegador.

**Geolocation**Proporciona aplicaciones web con acceso a la ubicación geográfica, así como datos sobre el dispositivo del usuario.  
  
**Offline apps**  
Permite la implementación de aplicaciones web disponibles en modo offline (sin conexión).

c) Cita qué opción ofrece CSS3 para conseguir que se apliquen diferentes estilos CSS sobre el mismo elemento en su visualización en diferentes dispositivos (i.e. diferentes tamaños de pantalla).

@rules (reglas-at) son declaraciones CSS que comienzan con el símbolo arroba, '@'. Una de las declaraciones más comunes es @media (Media Query) y agrupar un conjunto de reglas que se aplicarán de manera condicional si el dispositivo cumple los criterios especificados.

d) Cita al menos 4 de las características principales de TypeScript (importante superset de JavaScript que trataremos en la siguiente PEC/PAC).

**Variables tipadas (Static Types)**  
Sin duda la principal característica de TypeScript es que se trata de un lenguaje tipado. Lo que significa que nos permite definir el tipo de variables que vamos a utilizar. Por ejemplo:

|  |
| --- |
| let someNum:number = 123456; |

Resulta interesante saber que el tipado disponible en TypeScript es optativo. El desarrollador puede utilizarlo cuando considere que es realmente beneficioso.

**Compatibilidad en Navegadores**TypeScript permite el uso de funcionalidades modernas de JavaScript que luego serán compiladas a JavaScript compatible con la configuración deseada.

**Programación Orientada a Objetos**TypeScript ofrece todas coberturas que requiere un lenguaje orientada a objetos, como las clases, las interfaces, herencia, módulos, etc.

**Decorators**El patrón de diseño Decorator sirve para otorgar de funcionalidad a un objeto de forma dinámica.   
  
**Sintaxis sencilla para inyectar dependencias**  
La inyección de dependencias resulta realmente sencilla de implementar en comparación con ES6.  
  
**Errores en compilación, no ejecución**TypeScript permite la localización de errores en la fase de compilado, no durante la ejecución, como en el caso de JavaScript.  
  
**Datos públicos o privados de una clase**En TypeScript se pueden definir las variables accesibles desde fuera del scope (ámbito).

**2. El lenguaje CSS es muy rígido y poco práctico y ordenado a la hora de programar. Para evitar estos problemas se han creado los preprocesadores CSS que ofrecen evidentes ventajas. (0.5 puntos)**

a) Cita al menos 2 de estos preprocesadores.

Sass, LESS, Stylus o PostCSS son los preprocesadores más destacados actualmente.

b) Cita al menos 4 ventajas que ofrecen estos preprocesadores.

Posibilidad de utilizar expresiones, mixins y funciones.  
Evita la duplicidad de código.  
Permite el anidado.  
Favorece la organización del código, lo que se traduce en menor tiempo de desarrollo.  
Métodos disponibles de serie, como por ejemplo: darken, lighten o fade.

c) Explica brevemente en qué consisten los sourcemaps.

Como parte de los procesos de optimización web podemos aplicar una concentración y compresión (minification) de nuestros archivos CSS y JS. A menudo surge la necesidad de depurar el código, pero tras la compresión, la tarea se se hace realmente compleja. En este punto es en el que entran en juego los Source Maps. Un Source Map proporciona una forma de mapear el código de un archivo comprimido para devolverlo a su estado original.

d) Explica qué es un transpilador.

Un transpilador es una herramienta capaz de leer el código fuente escrito en un determinado lenguaje de programación, y producir el código equivalente en otro lenguaje. Sirva como ejemplo el caso de Babel, uno de los transpiladores más populares del sector. Babel compila JavaScript moderno a versiones anteriores interpretables por los navegadores.

**3. El flujo de trabajo profesional en front-end hace indispensable el uso de herramientas avanzadas como son los controles de versiones o build systems (i.e. automatizadores como Gulp/Grunt). (0.75 puntos)**

a) Cita al menos dos sistemas de control de versiones.

Actualmente existen múltiples sistemas de control de versiones. Sin duda, el más popular es Git (a través de herramientas como el Git y GitLab). Al márgen de Git, existen otros sistemas como CVS, SVN (Apache Subversion), Mercurial o Monotone.

b) Cita y explica al menos 3 comandos de Git (revisa el apartado siguiente de esta parte práctica).

**git stash drop**Descarta el conjunto de cambios “escondido” (stashed) más recientemente **git reset [commit]**Deshace los commits posteriores al [commit], preservando todos los cambios en el local

**git log --follow [archivo]**Devuelve una lista del historial de cambios en un archivo, incluyendo los casos de renombrado.

c) Cita y explica brevemente las características más definitorias/destacables de dos build systems.

**Webpack**Webpack es una herramienta que se utiliza para agrupar muchos paquetes web y diferentes dependencias del proyecto, incluyendo archivos CSS e imágenes. Como resultado devuelve archivos manejables. Una de las principales características que lo hace popular es la mejora en la productividad y reusabilidad, así como la reducción del tiempo de carga. Los tiempos de carga se optimizan mediante el fraccionamiento el árbol de dependencias en porciones más pequeñas, mientras las carga bajo demanda, por lo que se consume menos espacio en la memoria.  
  
**Grunt**Grunt es una gran herramienta que se caracteriza por ejecutar tareas para automatizar los procesos de desarrollo más repetitivos. Algunos de los procesos son: compactado/minification, pre-procesado de CSS, desarrollo, compilación, ejecución de Tests Unitarios, corrección de estilos (linting), etc.Grunt también puede gestionar el tamaño y la calidad de tus imágenes después del desarrollo.  
  
**Gulp**Como en el caso de Grunt, Gulp también sirve para la automatizar tareas repetitivas. En esencia ayuda en procesos como la creación de plantillas HTML, compactado de CSS y JS, optimización de imágenes y a creación de servidores locales de desarrollo.

PRÁCTICAS GUIADAS

**Práctica 1 – Contador de tiempo inverso (3 puntos)**

(2 puntos) En esta práctica se va a construir un contador de tiempo inverso tal y como se muestra a lo largo del video curso.

[https://learning.oreilly.com/learning-paths/learning-path-javascript/9781838644123/9781789805048-video1\_1](https://learning.oreilly.com/learning-paths/learning-path- javascript/9781838644123/9781789805048-video1_1)

(1 punto) Una vez construido el contador de tiempo inverso, modifica el código para hacer un contador de tiempo que permita aplicar la técnica de pomodoro ([https://es.wikipedia.org/wiki/T%C3%A9cnica\_Pomodoro](https://es.wikipedia.org/wiki/Técnica_Pomodoro)) tal y como se muestra en la siguiente Web (<https://tomato-timer.com/>).

**Práctica 2 - Lista de tareas almacenadas en LocalStorage (2 puntos)**

En esta práctica se va a utilizar el almacenamiento local de los navegadores (es similar a utilizar un fichero de texto pero almacenado en el navegador). Para ello se debe seguir el siguiente vídeo-curso: <https://learning.oreilly.com/videos/a-javascript-local/9781789801460/9781789801460-video1>

**Práctica 3 – Introducción a la API de Twitter (3 puntos)**

(2 puntos) En esta práctica haremos uso de código JavaScript de lado del servidor usando Node.js y una aplicación front-end (primera aplicación en la que JavaScript se encuentra tanto en back-end como en front-end). Se hará uso de la API de Twitter para enviar tweets a una cuenta de Twitter desde un panel Web. El objetivo de esta práctica es realizar peticiones AJAX, para ello se debe seguir el siguiente vídeo-tutorial:

<https://learning.oreilly.com/videos/a-twitter-api/9781789611724/9781789611724-video1>

(1 punto) Una vez realizado el vídeo-tutorial anterior, debes realizar todas las modificaciones necesarias para que allá donde en el código aparece un objeto XMLHttpRequest se use la API Fetch. Ten en cuenta que Fetch utiliza promesas, mientras que XMLHttpRequest usa callbacks.