Desarrollo Web en Entorno Cliente

**U1. Navegador y Lenguajes Web**

|  |
| --- |
| **Práctica 1 Objetivo:** |
| * • Conocer las características del editor Atom y las opciones de personalización que proporciona. |
| * • Conocer y hacer uso del servidor Browser-Sync, como herramientas que nos ayudarán al desarrollo. |
| * • Conocer y empezar a usar las herramientas de desarrolladores Git y Github para crear un primer repositorio.     **1. Editor Atom**  Atom es un editor Opensource que cuenta con numerosas capacidades de personalización. Visita su web y describe 3 características que consideres que lo hacen interesante.  Lee la siguiente entrada en la que se comentan opciones de personalización de Atom y aplica las opciones siguientes:   1. a. Indica qué nombres de fichero debe ignorar 3. b. Configura la carpeta de los proyectos      1. c. Instala el paquete color-picker y comprueba que funciona 3. d. Instala otro paquete que consideres interesante del artículo y comprueba que funciona   **2. Browser-Sync**  Browser-Sync es un servidor virtual que como puedes comprobar en su web proporciona numerosas funcionalidades. El primer paso para instalarlo es instalar NodeJS (Puedes comprobar la versión instalada con node - - version en la línea de comandos). Sigue las indicaciones que sugiere la web de Browser-Sync y realiza un guía sencilla con los pantallazos de los pasos que has seguido (HowTo de instalación). Comprueba con una web sencilla, que Browser-Sync es capaz de sincronizar el navegador y el editor. Ejecutando la instrucción siguiente en el directorio de trabajo: Browser-sync start --server --directory --files "\*\*/\*"    La implementación de la sentencia en un .bat para ejecutar la vista de la carpeta en el navegador.  **4. Crear un repositorio en Github**  • Crea un usuario de Github  • Crea un repositorio y una vez creado, copia la url y ejecuta en el prompt del terminal y las instrucciones necesarias para copiar el repositorio en local, actualizar cambios en local en el servidor remoto y actualizar cambios desde el servidor remoto:  **Git init**  (Lo utilizas en la ruta local en la que quieras que esté el repositorio para inicializarlo)  **Git clone <url>**  **Git status**  (Comprueba los cambios realizados)  **Git add <recurso>**  Marca algo localmente para enviar  **Git commit –m “Mensaje a guardar”**  Guarda los cambios en el histórico del proyecto  **Git push**  Actualiza en el repositorio remoto los cambios locales  **Git pull**  Actualiza localmente los cambios del servidor  Copia la url y realiza un pantallazo de la carpeta local con los ficheros del repositorio. |
|  |
|  |
|  |
|  |
|  |
|  |