

Apartado 1: Video “primera toma de contacto Git”

Git es un software de control de versiones libre creado por Linus Torvalds. Sirve para facilitar el desarrollo y mantenimiento de aplicaciones tanto de forma individual como sobre todo al trabajar en equipo. Realiza un guardado del registro de todos los cambios que sufran los archivos.

Apartado 2: Conociendo más sobre Git

Contesta a las siguientes preguntas:

1. ¿Qué es un control de versiones?

Un control de versiones es un sistema que registra los cambios realizados en un archivo o conjunto de archivos a lo largo del tiempo, de modo que puedas recuperar versiones específicas más adelante.

2. ¿Para qué se usa en el ámbito del desarrollo web?

Se utiliza para regresar a versiones anteriores de los archivos, regresar a una versión anterior del proyecto completo, comparar cambios a lo largo del tiempo, ver quien modificó por última vez algo que pueda estar causando problemas, ver quién introdujo un problema y cuándo, y muchos más. También significa generalmente que, si arruinas o pierdes archivos, será posible recuperarlos fácilmente. Adicionalmente, obtendrás todos estos beneficios a un costo muy bajo.

3. Explica en qué consisten los 3 tipos de “Sistemas de Control de Versiones”

- **Sistemas de Control de Versiones Locales**: guarda conjuntos de parches en un formato especial en disco y es capaz de recrear cómo era un archivo en cualquier momento a partir de dichos parches.
- **Sistemas de Control de Versiones Centralizados**: pensados para colaborar con desarrolladores en otros sistemas. Tienen un único servidor que contiene todos los archivos versionados y varios clientes que descargan los archivos desde ese lugar central.
- **Sistemas de Control de Versiones Distribuidos**: realizan una replicación completa del repositorio, permitiendo que los clientes descarguen la última copia instantánea de los archivos.

4. ¿Por qué se dice que “los dioses de la velocidad han bendecido Git con poderes sobrenaturales”?

Porque la mayoría de las operaciones de Git sólo necesitan archivos y recursos locales para funcionar, no se necesita información de ningún otro ordenador de nuestra red (de manera general). Es por esto que la mayoría de las operaciones parecen inmediatas.

5. ¿Por qué se caracterizan los 3 estados en que pueden encontrarse tus archivos en Git (confirmado, modificado y preparado)?

- Confirmado: los datos están almacenados de manera segura en tu base de datos local.
- Modificado: el archivo ha sido modificado, pero todavía no se ha confirmado en la base de datos.
- Preparado: el archivo modificado ha sido marcado en su versión actual para que vaya en la próxima confirmación.

6. ¿Qué tres secciones habrá en un proyecto Git? ¿Qué fin tiene cada una?

- El directorio de Git: es donde se almacenan los metadatos y la base de datos de objetos para tu proyecto. Es la parte más importante de Git, y es lo que se copia cuando clonas un repositorio desde otro ordenador.
- El directorio de trabajo: es una copia de una versión del proyecto. Estos archivos se sacan de la base de datos comprimida en el directorio de Git, y se colocan en disco para que los puedas usar o modificar.
- El área de preparación: es un archivo, generalmente contenido en el directorio de Git, que almacena información acerca de lo que va a ir en la próxima confirmación. A veces se le denomina índice (“index”), pero se está convirtiendo en estándar el referirse a ella como el área de preparación.

7. Explica cómo es el flujo de trabajo normal cuando se trabaja con Git.

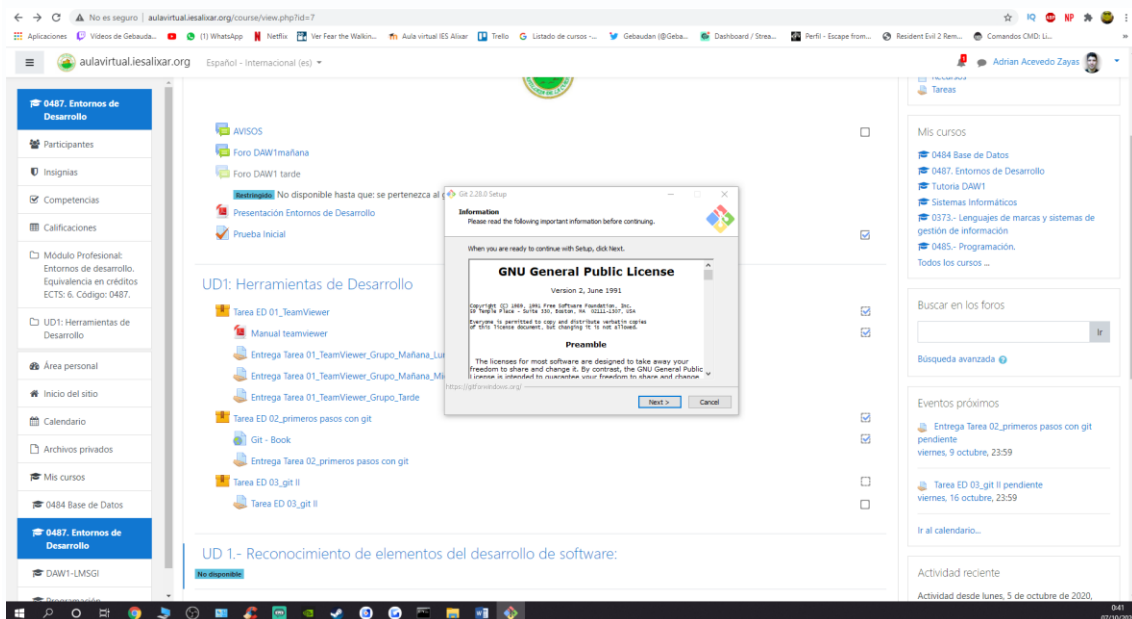
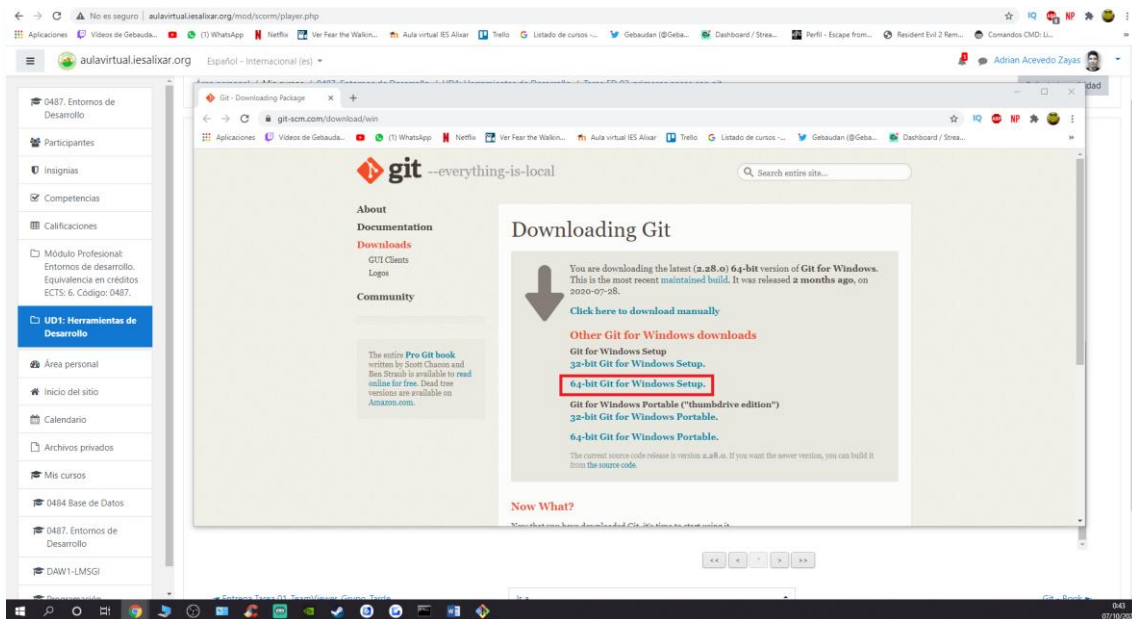
En primer lugar, modificas una serie de archivos en tu directorio de trabajo. A continuación, preparas los archivos, añadiéndolos a tu área de preparación. Por último, confirmas los cambios, lo que toma los archivos tal y como están en el área de preparación y almacena esa copia instantánea de manera permanente en tu directorio de Git,

Apartado 3: Instalando Git

1. ¿Qué ventaja tiene usar Git desde la línea de comandos frente a usarlo desde una interfaz gráfica integrada en algún Entorno de Desarrollo?

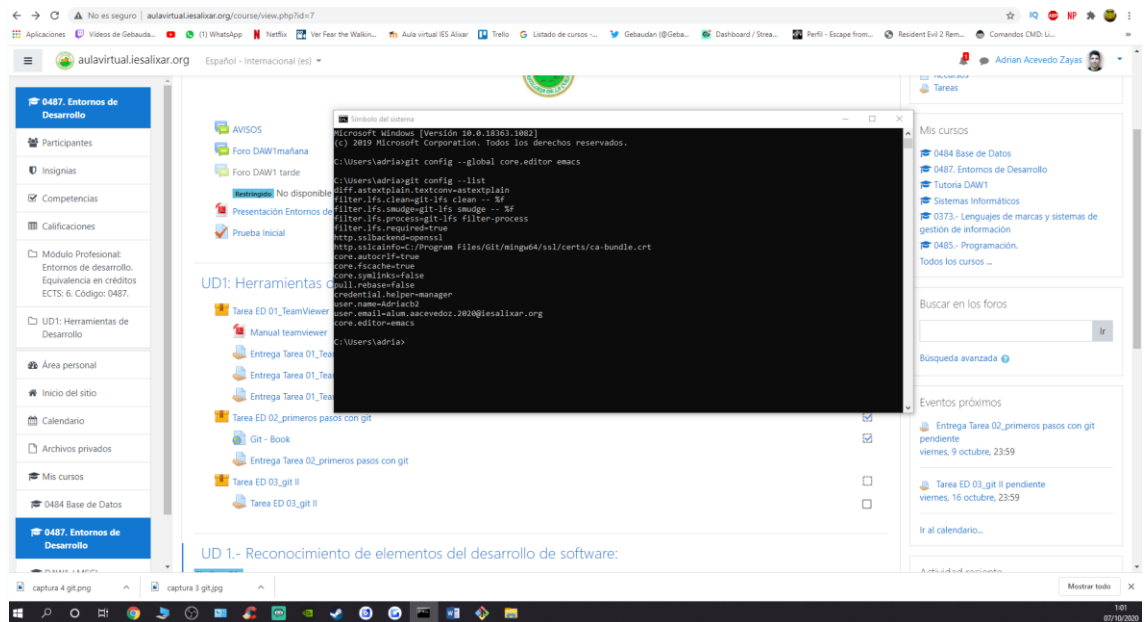
La línea de comandos nos permite ejecutar todos los comandos de Git ya que la mayoría de interfaces gráficas de usuario solo implementan una parte de las características de Git por motivos de simplicidad.

2. Instala Git en tu sistema operativo. Haz una captura de pantalla del proceso



The screenshot shows the website interface for 'aulavirtual.iesalixar.org'. The left sidebar contains navigation links such as 'Participantes', 'Insignias', 'Competencias', 'Calificaciones', 'Módulo Profesional: Entornos de desarrollo.', 'UD1: Herramientas de desarrollo', 'Área personal', 'Inicio del sitio', 'Calendario', 'Archivos privados', 'Mis cursos', and '0484 Base de Datos'. The main content area displays the title 'UD1: Herramientas de desarrollo' and a list of resources including 'Manual de configuración de entorno de desarrollo', 'Entrega Tarea ED_01_git I', 'Entrega Tarea ED_02_primeros pasos con git', 'Entrega Tarea ED_03_git II', and 'Entrega Tarea ED_04_git III'. A terminal window is open, showing the output of the command 'git config --list', which lists various Git settings like user.name, user.email, core.editor, and repositoryformatversion.

5. Comprueba tu configuración



6. Desde la línea de comandos, usa el comando para obtener la ayuda de Git.

