Word: Ctrl + av pag: pasar de pagina

Exámenes

Uf2404 – Principios POO: 12.3.2020: 105 poo (mecanismo entrada salida) . Ver colleccions., interfaces

Uf2405 y uf2406: programación Web y ciclo de vida del proyecto: 29.4.2020

2-3-2020 LUNES

**1ª parte – GIT** (09:15 – 11:20)

Github 🡺 tutorial bueno <https://www.youtube.com/watch?v=HiXLkL42tMU&t=2096s>

* Repositorio local/remoto de código
* Es un sistema control de versiones (VCS): Capaz de volver a versiones anteriores. Hay un registro de código: qué, cuándo y por quién.
* Todos los desarrolladores tienen una copia del código. Ayuda a coordinar al estar trabajando varios desarrolladores en un mismo proyecto.
* creado por el desarrollador de Linux. Es OpenSource.
* Uso por consola (por medio de comandos de Linux):
  1. Iniciar 🡺 git init. Git tiene tres estados:
     + Working directory: donde trabajamos en primera instancia.
     + Staging área: para controlarlo por versiones. 🡺git add <file>
     + Repository (local o remoto): pasar al respositorio 🡺git commit
  2. añadir proyectos o carpetas
  3. ponerlo en el repositorio
  4. crear cuenta Github

Git status🡺 ver estado actual, para ver donde están.

Git commit🡺 pasar del staging area al repostorio

Git push🡺 para subirlo a un servidor remoto (a uno q no es local)

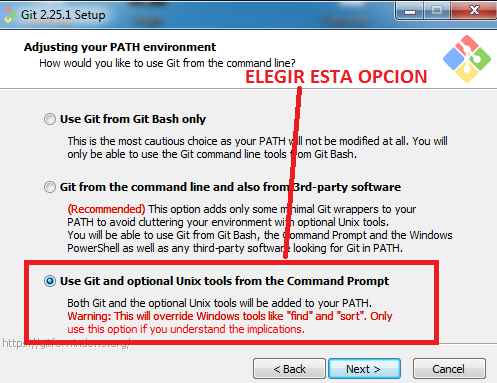
Git pull 🡺 volver a versiones anteriores.

Git clone🡺 para bajarte todo (hace una copia del servidor a mi ordenador)

Maven: repositorio de dependencias.

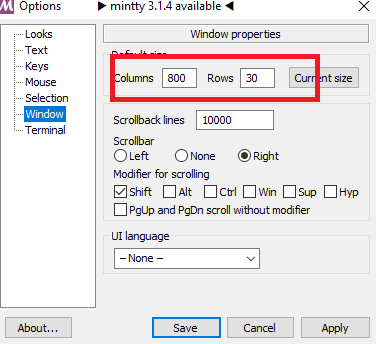
Instalacion: <https://git-scm.com/download/win>

” Git-2.25.1-64-bit.exe”: todo por defecto, excepto esto: (nos permitirá con la consola abierta, podemos usar los comandos de Linux: ls (listar ficheros), pwd (dd estamos),…)



Crear distintas ramas (queremos que el código vaya separado porque p.e por agregar funcionalidades en uno): Git Branch, cambiar de rama git chechout

Despues se iniciará, y para verlo en tamaño grande, damos a estas opciones:

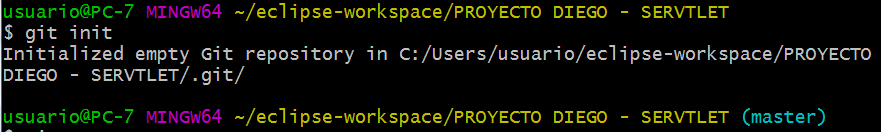


Lo reiniciamos.

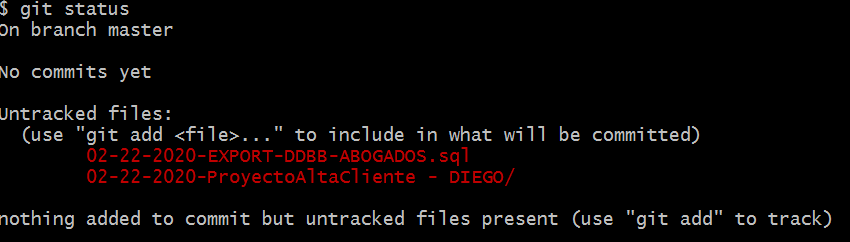
Pasos:

1. Agregar un proyecto:

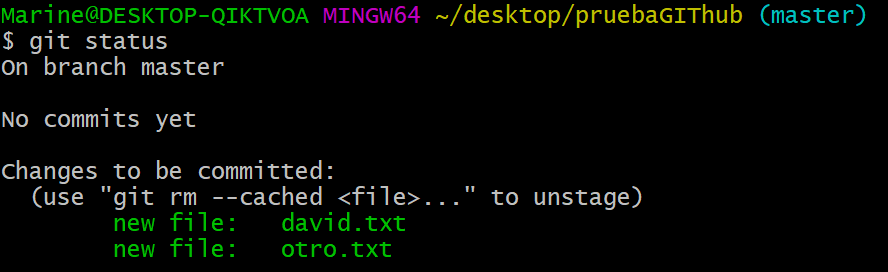
* botón dcho: añadir al bash la carpeta
* después poner en la consola: git init (para chequear que funcione, entramos en la carpeta del proyecto y vemos los archivos ocultos, y veremos que aparece una carpeta oculta .git. Podemos usar ls -A



1. get status: para ver los archivos q tenemos



1. Añadir contenido: git add “nombre del fichero” (dp poner git Status para ver como ha cambiado) y se verán añadidos en verde:



git add WebContent/

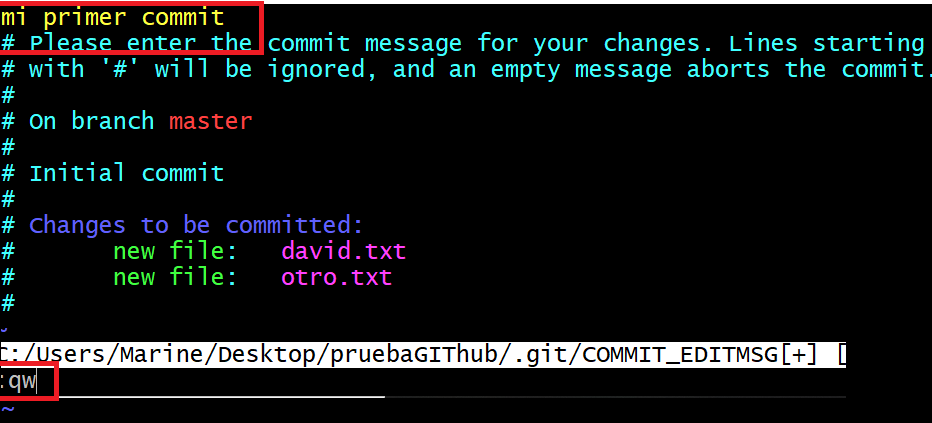
1. Configurar Git hub: es necesario decir quienes somos para q cuando subamos un proyecto estén separados:

git config --global user.email "thunderdome@gmail.com"

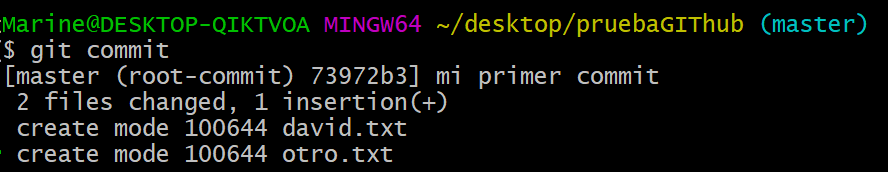
git config --global user.name "david"

1. git commit: crea un punto de control de nuestro código. Es grabar.

Se escribe en la parte superior: (si no deja, pulsar tecla”i” o insert) ponemos el nombre que queramos y luego pulsamos tecla escape para q vaya a la línea de abajo, y escribimos “:wq”.



Y saldrá esto, q indicará q todo está correcto:

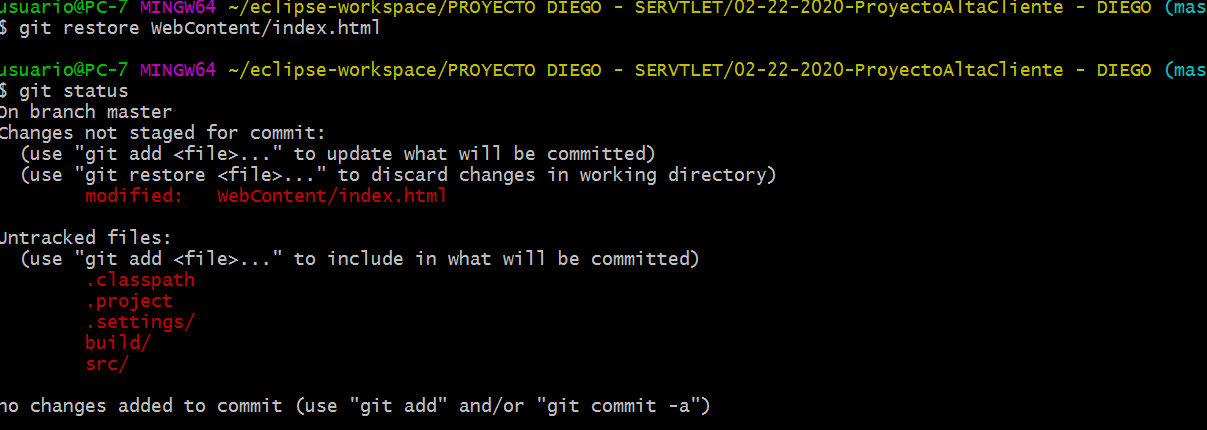


Con Git log podemos acceder a la información del proyecto

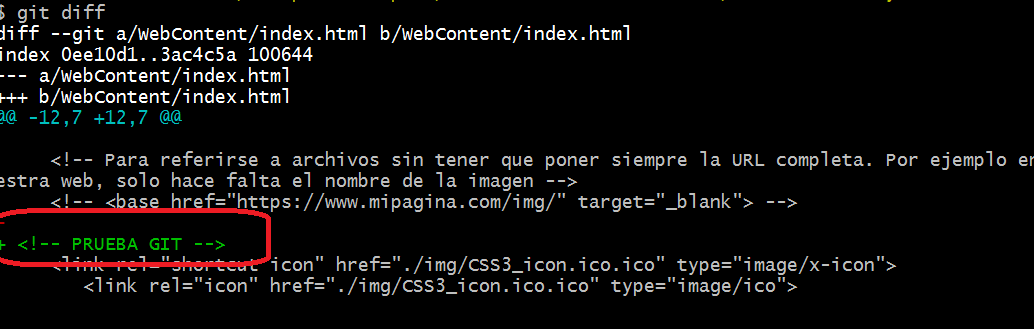
Metodo rápido de grabación:git commit – m “nombre del mensaje” y asi no hace falta q entremos en el editor de código.

1. Si modificamos un fichero y hacemos un git status, nos indicará el fichero q se ha modificado y se pueden revertir los cambios con Git checkout -- “nombre fichero”

PARA REVERTIR LOS CAMBIOS: git restore WebContent/index.html



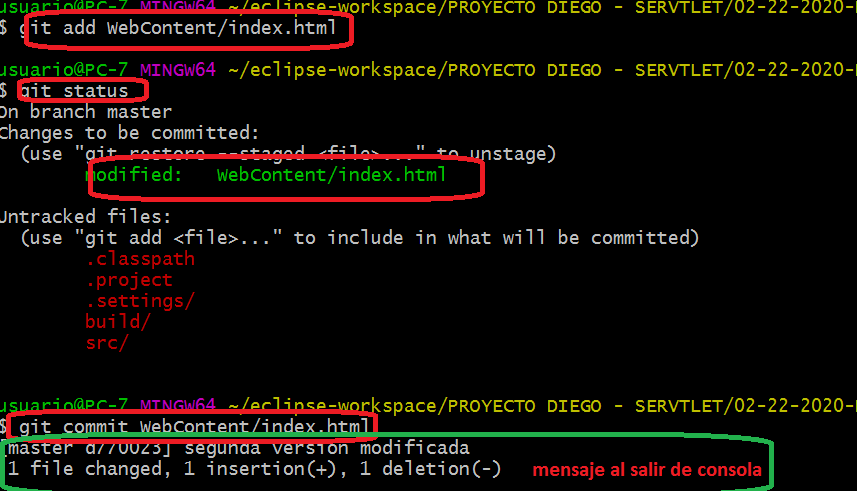
1. Git diff “nombre fichero”: para q muestre las diferencias del nuevo con el antiguo. Se marcaran en color verde.



1. Aceptar los cambios:

git add WebContent/index.html (no muestra ningún mensaje)

Para ver que se ha ejecutado correctamente, vemos el status y nos saldrá en verde.



Git log: nos mostará las versiones grabadas del fichero:

1. Git IGNORE 🡺Evitar cambios: git commit -m

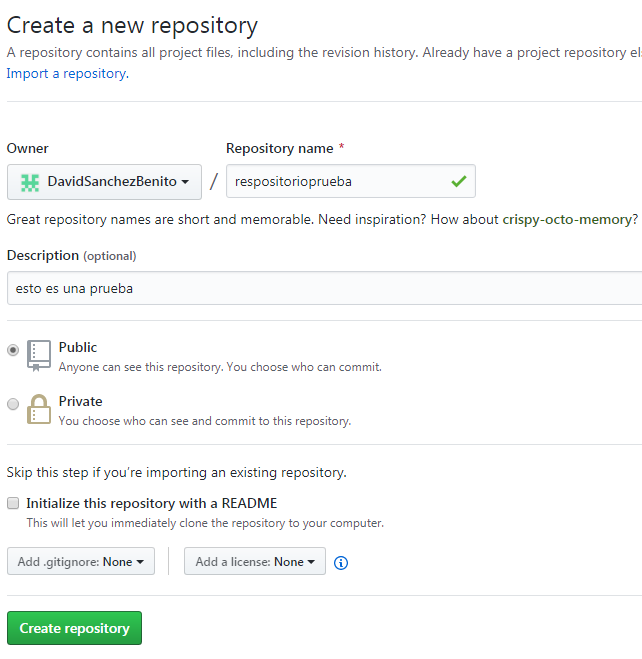
Podemos crear una nuevo fichero en el proyecto: .gitignore (sin extensión) y escribimos los nombres de carpetas o ficheros a ignorar.

1. Git Branch: nos servirá para hacer una versión alternativa del proyecto con el q estmos trabajando.
   1. Git Branch “nombre”: crea otra nueva (la existente es master)
   2. Git checkout “nombre creado”: nos pasamos a la versión elegida.

<https://swcarpentry.github.io/git-novice-es/06-ignore/>

<https://github.com/>

1. dar de alta la cuenta: Gi111111
2. Crear repositorio:vamos a mi usuario, y elegimos MIS REPOSITORIOS, y luego new



Nos indicará un link donde se cargarán los ficheros:



Como ya lo habíamos hecho desde el bash, tendremos que incluir las dos últimas líneas incluidas desde el bash:

1. Subir

git remote add origin <https://github.com/DavidSanchezBenito/respositorioprueba.git>

git push -u origin master

Añadir la carpeta al bash,

Git init

Git add . (el . significa desde la ruta actual hacia adelante(

Git status (nos aparecerá todo lo añadido)

git config --global user.email "thunderdome@gmail.com"

git config --global user.name "david"

git commit -m "facturacion" (asi no nos abre el editor)

git log (para chequear q me ha creado la versión prueba)

Pegar los links del repositorio que me facilita via web:

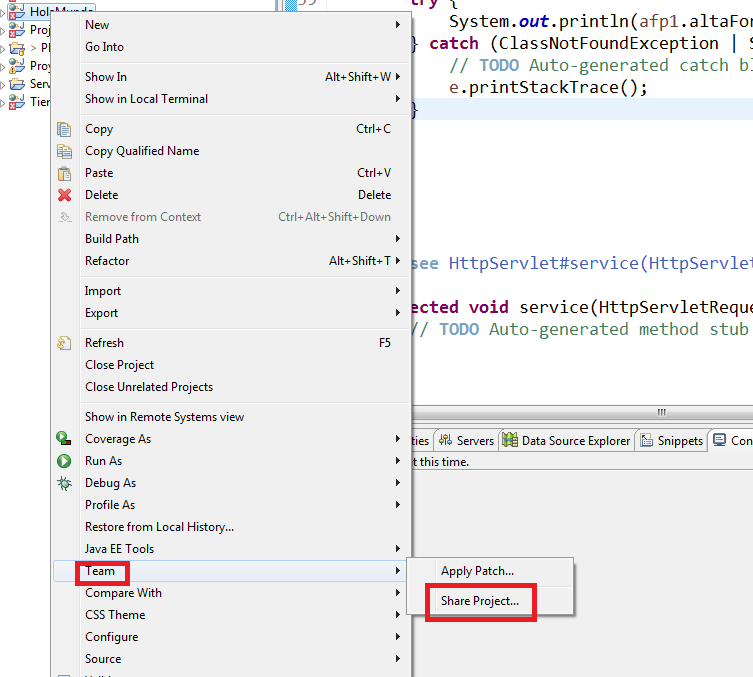
**2ª parte: HTML con Eclipse** (12:15 - 14:20)

Seguimos con el proyecto de equipo.

3-3-2020 MARTES

**1ª parte – Gihug en Eclipse: Egit** (09:15 – 11:20)

Egit: botón dcho en proyecto, si nos aparece TEAM es que está instalado:

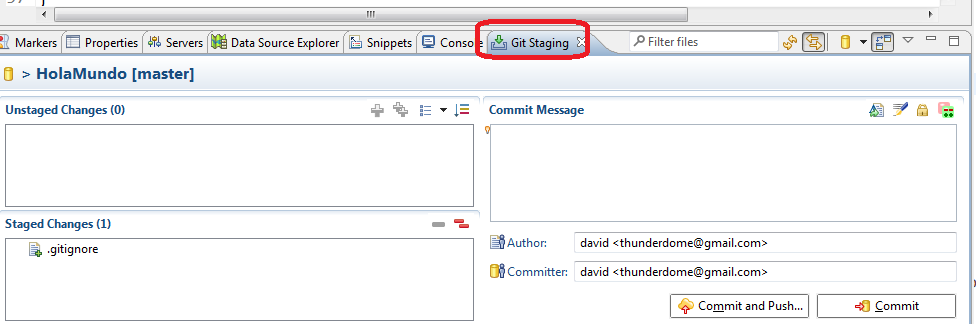


Finalizar y nos quedará asi (si vamos dentro de la carpeta del proyecto, tendremos un .git)



-Luego le damos add index.

-Mostramos la vista de git Staging y aparecerá esto:

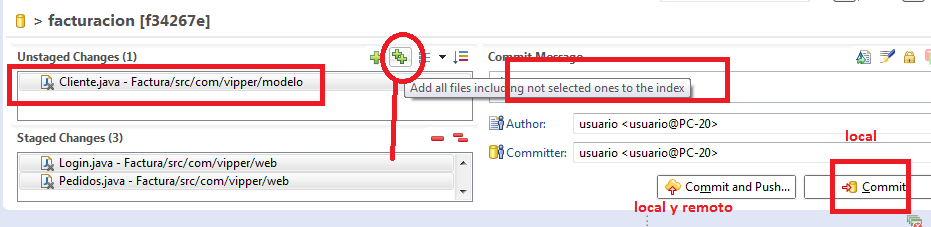


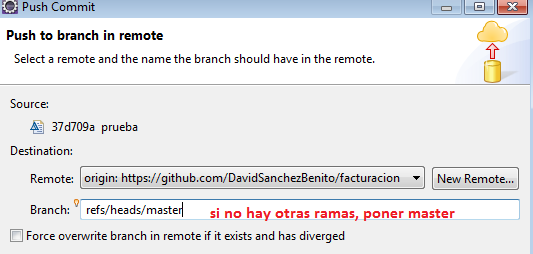
Vamos a commit message (team🡺 commit)

Remote – push

**Bajar otra versión grabada:**

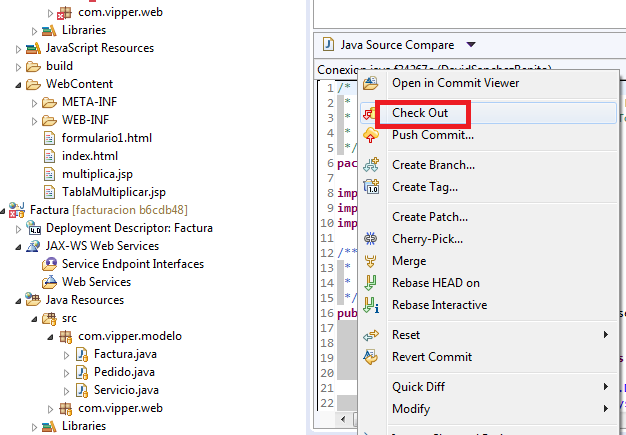
Si queremos volver justo a la última grabada, lo que hay que hacer es subir la actual (aunque sabemos que esta mal) con un commit:



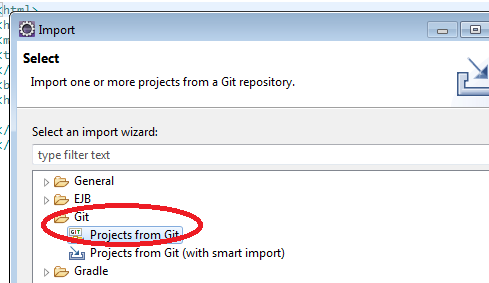


Luego pinchamos en la pestaña history:

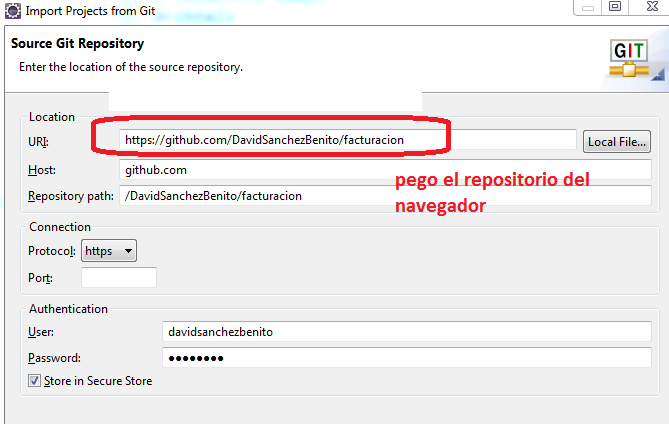
luego seleccionar la anterior, que nos borrará la ultima, con solo pulsar check out:



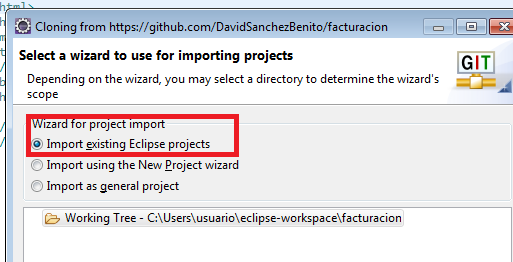
**IMPORTAR PRIMERA VEZ DESDE GIHUG (sin tener ninguno instalado):**

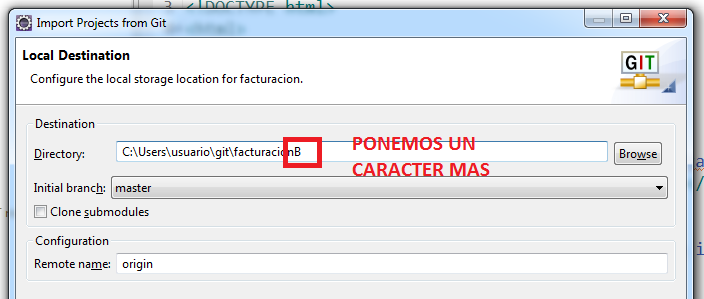


Elegimos cloneuri y en la siguiente pegamos el repositorio



Siguiente: seleccionamos el Branch, como hay solo uno elegimos el q aparece.





En la siguiente next

Con esto ya nos aparecería en el listado de proyectos, con el texto master:



**2ª parte: HTML con Eclipse** (12:15 - 14:20)

Proyecto equipo: problemas capa conexión(sigue funcionando). Pruebas de backup con github.

**body** {

background-image: *url('images/candado.jpg')*;}

**img** {

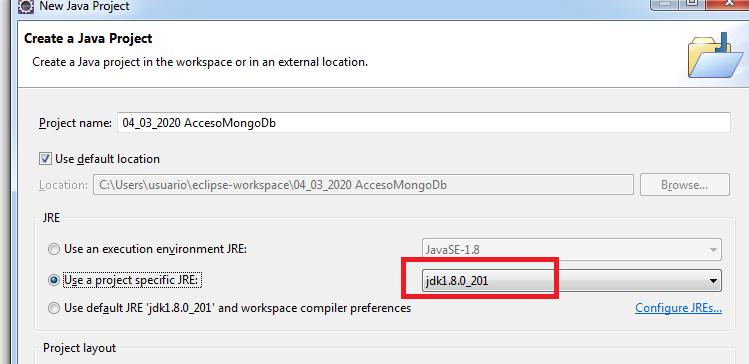
opacity: 50%;

}

4-3-2020 MIERCOLES

**1ª parte – Egit** (09:15 – 11:20)

* Probamos Egit en eclipse
* MongoDB: acceso 🡺 getDB(nombreBD); dbColeccition
* **Maven** es una herramienta de software para la gestión y construcción de proyectos [Java](https://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n_Java) creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a [Apache Ant](https://es.wikipedia.org/wiki/Apache_Ant) (y en menor medida a [PEAR](https://es.wikipedia.org/wiki/PEAR) de [PHP](https://es.wikipedia.org/wiki/PHP) y CPAN de [Perl](https://es.wikipedia.org/wiki/Perl)), pero tiene un modelo de configuración de construcción más simple, basado en un formato [XML](https://es.wikipedia.org/wiki/XML). Estuvo integrado inicialmente dentro del [proyecto Jakarta](https://es.wikipedia.org/wiki/Proyecto_Jakarta) pero ahora ya es un proyecto de nivel superior de la [Apache Software Foundation](http://www.apache.org/).
* Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.



Una vez creado, elegimos el proyecto: botón derecho: configure, convert to Maven Project.

(si hacemos primero el Maven, no nos crea la estructura web)

Despues creamos etiqueta dependencias y pegamos esto

<https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver/2.12.3>

<!-- https://mvnrepository.com/artifact/org.mongodb/mongo-java-driver -->

<dependency>

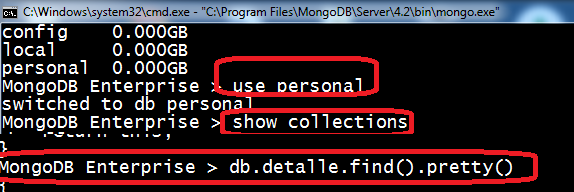
<groupId>org.mongodb</groupId>

<artifactId>mongo-java-driver</artifactId>

<version>2.12.3</version>

</dependency>

CMD: ejecutamos el mongo.exe (local) 🡺 (arrastrarlo a la consola)y ponemos show dbs



**2ª parte: HTML con Eclipse** (12:15 - 14:20)

Continuamos con la conexión de mongo desde eclipse en proyecto: 04\_03\_2020 AccesoMongoDb

5-3-2020 JUEVES

No hubo clase (asuntos personales de Gabriela)

**1ª parte – BBDD** (09:15 – 11:20)

**2ª parte: HTML con Eclipse** (12:15 - 14:20)

6-3-2020 viernes

**1ª parte – BBDD** (09:15 – 11:20)

MongoCliente: método getDB(le damos el nombre de la DB).

Obtenemos el cliente con getcollection, dp el método find. Esto nos deja un DBCurso (tiene un hasnext: da true o false, el next lee).

Al hacer un next, nos deja El basicDBObject(nos da los campos getINT, get XXX) y lo mismo con BAsicDBObjectbuilder

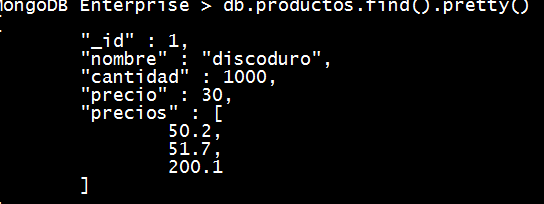
Hacemos el método de filtro de >35 años del proyecto 04\_03\_2020 AccesoMongoDb

Duplicamos después el proyecto: 06\_03\_2020 InsercionMongo: vamos a crear un BBDD nueva insertando un producto

Ctrl+may+7: comentar



* Añadir documento embebido.
* insertarDocumentoConArray:



Borra registros de mongodb: db.productos.remove({})

**2ª parte: HTML con Eclipse** (12:15 - 14:20)

9-3-2020 LUNES

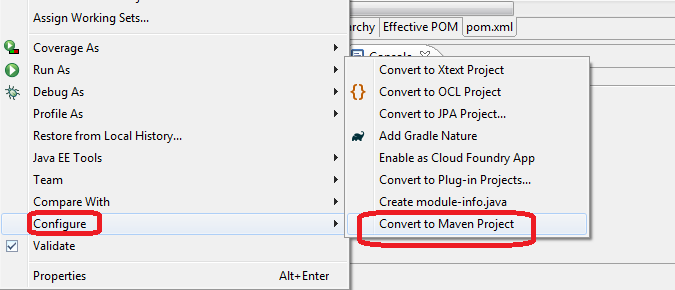
**1ª parte – HIBERNATE** (09:38 – 11:20) – RETRASO por archivos borrados en carpeta de Dropbox de Gabriela

hIBERNate: es un framework de la capa de persistencia . Vemos e T5A – Hibernate.pdf

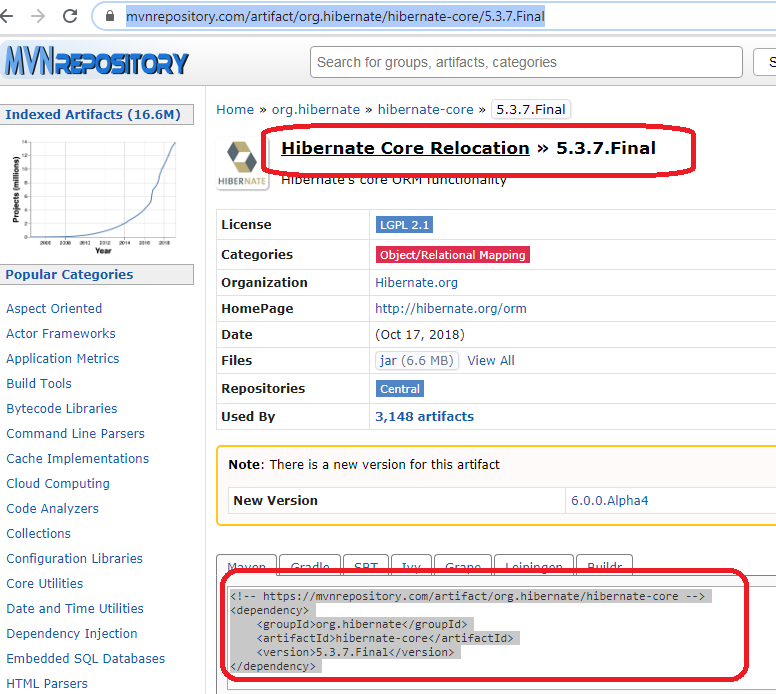
desde la 41: Teoria Hibernate y JPA.pdf

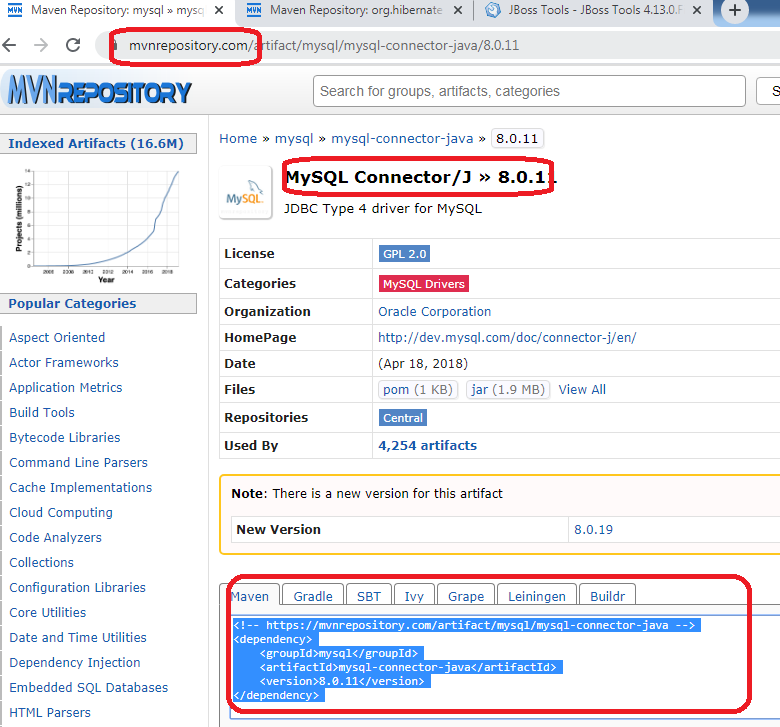
Ejercicio:

Creamos un proyecto java. Luego lo seleccionamos del explorador, botón dcho y transformamos a MAVEn:

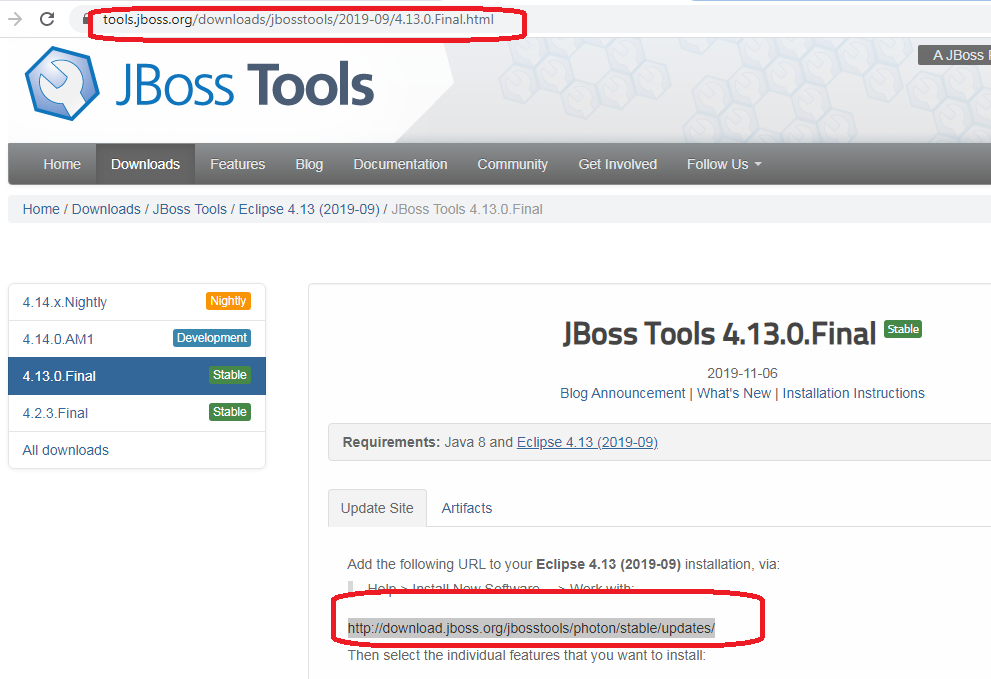


Entramos en la web y copiamos esto, creando la etiqueta <dependencias>, después de la etiqueta </input>:

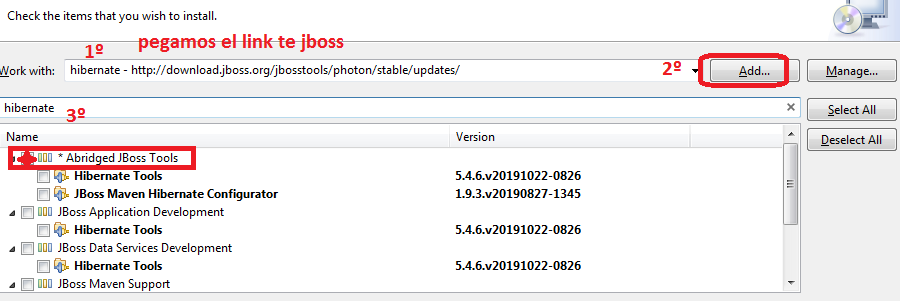




Copiamos el siguiente link:

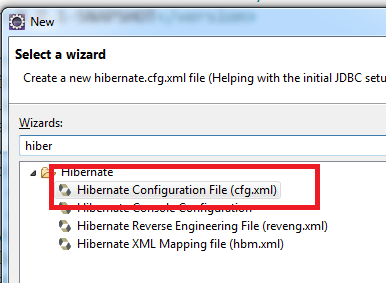


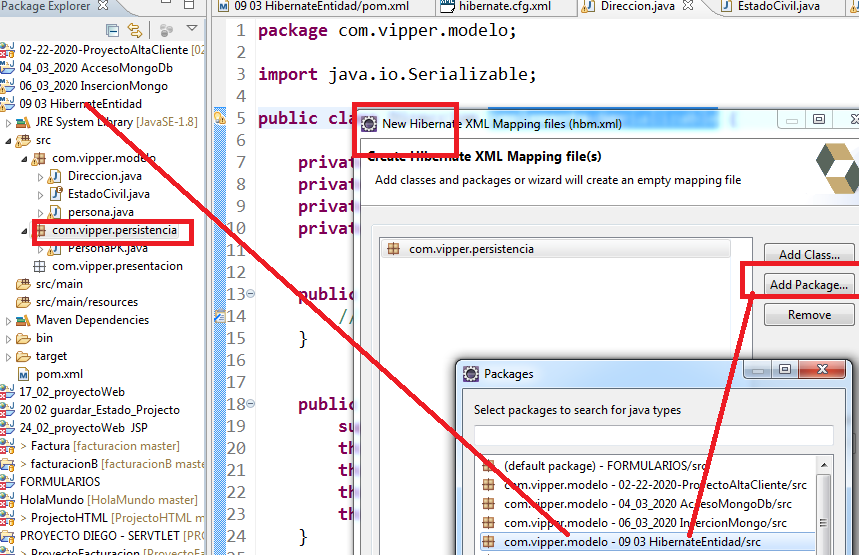
Después: help, install new software, y pegamos el link anterior:



Tardará un poco, y nos pedirá reiniciar eclipse. (si vemos que tarda, cancelamos, reiniciamos eclipse y repetimos proceso).

Para comprobar q está instalado: file, new proyect, other:

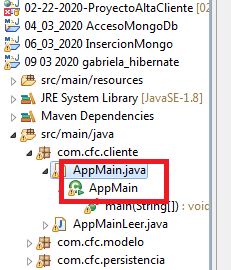




**2ª parte: HTML con Eclipse** (12:15 - 14:20)

Continuamos con lo anterior

Ejectura el proyecto de Gabrilea q si funciona: 09 03 2020 gabriela\_hibernate



10-3-2020 MARTES

**1ª parte – Repaso examen JAVA** (09:15 – 11:38)

Repaso pdf java

**2ª parte: HTML con Eclipse** (12:15 - 14:20)

Repaso pdf java: equals y hashcode (ÉSTE es como un cifrado)

Clase abstracta: //si el metodo es abstracto la clase es abstacta y no se puede

//instanciar. Se puede ver como una clase inacabada; //este le creo con un parametra de entrada q sea el ivaç

//metodo abstracto: metodo q no tiene implementacion

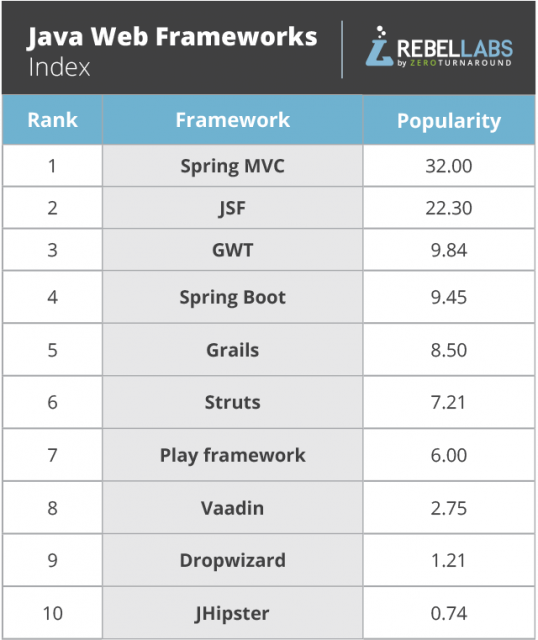
10-3-2020 MIERCOLES

**1ª parte – BBDD** (09:15 – 11:20)

**2ª parte: HTML con Eclipse** (12:15 - 14:20)

Github: mirar q es

Framework: es como un plato preparado comparado con preparar la comida.



* Hibernate : pero hay que ver primero bases de datos.

<https://www.youtube.com/watch?v=kKPec73bGnc&t=147s>

Hibernate ORM (or simply Hibernate) is an object-relational mapping tool for the Java programming language. It provides a framework for mapping an object-oriented domain model to a relational database. Hibernate handles object-relational impedance mismatch problems by replacing direct, persistent database accesses with high-level object handling functions. It´s is free software that is distributed under the GNU Lesser General Public License 2.1.

Hibernate's primary feature is mapping from Java classes to database tables, and mapping from Java data types to SQL data types. Hibernate also provides data query and retrieval facilities. It generates SQL calls and relieves the developer from the manual handling and object conversion of the result set.

* Spring: utiliza los objetos q están en un contenedor.

<https://www.youtube.com/watch?v=75PyjEfX3ws>

### [JavaServer Faces (JSF)](https://en.wikipedia.org/wiki/JavaServer_Faces)

Java Swing => ya no se utiliza, porque es para hacer ventanas de escritorio y ya todo se crea en web

Notas David JAVA

Números múltiplos: if (n1%n2==0

Apples= ya no se utilizan

**Front End**: se estudia Html y hojas de estilo

insert INTO Products