

Buenas tardes, Estimados alumnos.

Gracias por su asistencia a la clase de Laboratorio del día de ayer. Sé que fue super intensa, pero como sólo nos vemos 1 vez a la semana, el contenido debe ser "cargadito". Pero no se preocupen, aquí estoy para ayudarles en todo lo que sea posible.

En unos minutos más subiré el video de la clase al sitio del curso. (verlo en: <https://ingenieriadesoftwareciencias.mx>) (También estará ya arriba el video de la clase de Teoría, de la profesora Hanna)

Creo que con la introducción inicial de los primeros 30 minutos quedó claro que, dado el avance tecnológico actual, ya no es posible programar "a solas" y es necesario aprender a trabajar en equipo con el apoyo de poderosas herramientas para trabajo colaborativo que eran impensables hace unas pocas décadas, dado el bajo poder de cómputo del que se disponía entonces.

Dicho lo anterior, y con base en el video de la clase y un segundo video del que adjunto su liga en este correo, les pido que por favor: instalen en sus equipos estas 5 herramientas:

- java 8 (puede ser el openjdk)
- maven (no olvidar teclear mvn para que Maven diga que si está instalado)
- docker (no olvidar agregar el usuario actual al grupo de docker (como le hice yo) y reiniciar el equipo)
- git (si es que no lo tienen aun)
- nodeJS

Los que tengan Linux Ubuntu, pueden seguir los pasos que yo seguí durante la clase de ayer. Los que tengan Mac, pueden hacer todo de manera gráfica y los que tengan windows, que Dios los proteja, porque no sé cómo se instalan las cosas en ese sistema operativo; sin embargo, aún tienen la alternativa de usar **vmware o virtualbox** y hacer una máquina virtual con ubuntu (versión 20.04) dentro de windows. Esto de las máquinas virtuales también aplica a quienes me comentaron que tienen sistemas operativos Linux tipo Fedora o CentOS o sistemas operativos raros.

**Lo segundo que les pido es "preparar maven" de la siguiente manera:**

Una vez que terminó la instalación del software de desarrollo, deberán copiar los archivos `settings.xml`, `settings-security.xml` y `archetype-catalog.xml` en el directorio `.m2` y `.m2/repository` acorde a los que les mostré durante la clase. (ver el video de la clase varias veces)

**Finalmente, les pido lo siguiente:**

Crear un directorio llamado "work" en su home y clonar los repositorios de proyecto-backend y proyecto-frontend ejecutando desde "work" el comando "git clone". A este momento, ya deberían de tener acceso a los repositorios de GitLab.

Lo anterior yo ya lo hice desde cero en una máquina nueva y la grabación está en esta liga:

<https://aws-multimedia-bucket.s3.us-east-2.amazonaws.com/videos-clases/intall.mp4>

(OJO: Durante la clase que dí ayer, a mi me salio un error casi al final de la instalación y quedó grabado en el video que subí de la clase (1:02:21).

El video install.mp4 instala todo lo que intenté hacer durante la clase, pero sin errores y de manera directa.

Le daré un crédito especial al alumno que me diga qué error cometí en clase y cómo lo arreglé en el video install.mp4)

----- Parte 2 -----

Como pudieron observar, el 90% del proceso de despliegue de un sitio web con funcionalidad básica, base de datos, dominio y SSL ya está creado, funcional y **ya no lo tendrán que hacer Ustedes**. Prácticamente podríamos decir que esta infraestructura es casi automática y sólo se tiene que hacer una vez y ya lo hice por ustedes.

**Durante el curso, prácticamente sólo se tendrán que concentrar en "código puro" asociado a la regla de negocio y la interfaz gráfica del aplicativo. Por fortuna, la publicación en Internet, el nombre de dominio, el certificado SSL, el balanceo de cargas, la tolerancia a fallos, el motor de base de datos y el servidor en la nube son aspectos que YA ESTÁN RESUELTOS, y que por ello, ya no se tendrán que preocupar por implementar ustedes.**

Ya asigné permisos de acceso a los distintos equipos para GitLab que me enviaron su nombre de equipo. Podrán bajar el código del frontend y del backend como se ve en el video cuya liga les puse unos párrafos atrás. de todas formas, les vuelvo a poner las ligas aquí:

<https://gitlab.ingenieriadesoftwareciencias.mx/root/proyecto-frontend.git>

<https://gitlab.ingenieriadesoftwareciencias.mx/root/proyecto-backend.git>

El código que les compartí en Gitlab indica cómo me conecté a la base de datos, cómo generé pantallas, cómo inyecté lógica de negocio y lo vamos a ver con más detalle la próxima clase, junto con Nexus y Jenkins. A ustedes les toca (en unas pocas semanas más, generar el mismo tipo de código, pero referente al problema que van a resolver)

Sólo deben de estar conscientes de que para generar código deberán seguir una mecánica que expliqué un par de veces ayer:

- Aceptar una orden de trabajo (esto, por el momento, es una acción informal. Luego veremos cómo se acepta una orden de trabajo de manera "oficial")
- Crear una rama
- Bajar la rama a un espacio local

- Trabajar la rama (haciendo varios commits durante ese proceso)
- Al concluir su trabajo, subir la rama
- Generar un Merge Request
- Atender las observaciones
- Ya que no existan más observaciones, realizar el "merge"
- Repetir desde el principio

Lo anterior es una variante simplificada del GitFlow “original”, que pueden checar en esta liga:

<https://datasift.github.io/gitflow/IntroducingGitFlow.html>

Bueno, alumnos, gracias nuevamente por su atención a la presente. Quedo atento a los comentarios que se sirvan tener a la presente.

Saludos cordiales,  
Gustavo