

# P01A: Entorno de pruebas. Preparación.



Para poder realizar las prácticas de la asignatura, tendremos que:

- Descargar y poner en marcha la máquina virtual de la asignatura
- Crear un repositorio Git remoto, en Bitbucket, y otro local, en nuestra máquina virtual
- Crear una cuenta educacional en JetBrains (si no la tenéis creada ya de otras asignaturas), para poder usar IntelliJ

## Máquina virtual de la asignatura

Para las prácticas en el laboratorio utilizaremos una Máquina Virtual en la que tendréis instalado el software que vamos a necesitar. En el laboratorio estará accesible desde el escritorio cuyo icono lleva el nombre "PPSS-2021-22".

El login y password de la máquina virtual es **ppss** en ambos casos.

Para trabajar con la máquina virtual desde vuestro ordenador necesitaréis tener instalado previamente **VirtualBox** (incluyendo VirtualBox **Extension Pack**) y crear una nueva máquina virtual a partir del fichero "*ppss-2021-22.vdi*".

Podéis descargar el fichero comprimido ("*ppss-2021-22.vdi.zip*") desde el siguiente enlace: [https://drive.google.com/file/d/150Q\\_LleefHoIWNeHlp941Am3DGcQm0s3/view?usp=sharing](https://drive.google.com/file/d/150Q_LleefHoIWNeHlp941Am3DGcQm0s3/view?usp=sharing)

Para CREAR la nueva máquina virtual desde VirtualBox, con la opción **Nueva**. Usaremos los siguientes valores:

- ❖ Nombre: Podéis poner cualquier nombre arbitrario
- ❖ Tipo: Linux
- ❖ Version: Ubuntu (64 bits)
- ❖ Usar un archivo de disco duro virtual existente. A continuación seleccionamos el fichero **ppss-2021-22.vdi** de nuestro disco duro.

Una vez creada la máquina virtual, y antes de iniciarla, es posible que necesites cambiar la configuración de la pantalla para que no se muestre en un tamaño demasiado pequeño (dependerá de tu ordenador). Desde **Configuración**→**Pantalla** puedes probar a ponerla al 200% si lo necesitas.

Finalmente seleccionamos la máquina que hemos creado y a continuación elegimos la opción **Iniciar**.

**IMPORTANTE**: Si al acceder a la máquina virtual se abre la aplicación que indica que hay actualizaciones pendientes, de momento NO actualicéis ninguna. Desde la ventana de dicha aplicación, podéis acceder a Configuración→Actualizaciones, y en "Comprobar actualizaciones automáticamente, seleccionáis "Nunca", para que no os aparezca el aviso cada vez.

**IMPORTANTE**: Antes de comenzar a hacer nada, recuerda que debes instalar **GuestAdditions** para no tener problemas al redimensionar la ventana de la máquina virtual. Cada vez que actualices la versión de VirtualBox, deberás volver a instalar **GuestAdditions** (desde el menú Devices)

Os recomendamos que activéis la compartición del portapapeles entre la máquina anfitrión y vuestra máquina virtual desde: **Devices**→**Shared Clipboard**→**Bidirectional**. De esta forma, podremos copiar/pegar el contenido del portapapeles entre ambas máquinas.

## Git: repositorio remoto, en Bitbucket

Es una muy buena práctica (indispensable para futuros trabajos profesionales en grupo) utilizar una **herramienta de gestión de versiones**, para así tener todo nuestro trabajo accesible desde cualquier lugar, y con el historial de versiones, por si nos interesa recuperar alguna versión anterior, crear nuevas versiones a partir de ellas, etc.

Aunque nosotros no vamos a trabajar en grupo, es interesante habituarnos a utilizar un repositorio remoto para, por ejemplo, tener siempre una copia de seguridad del trabajo realizado, continuar el

trabajo iniciado desde cualquier otro ordenador en el mismo punto que lo dejamos, ..., entre otras cosas.

**Bitbucket** es un servicio que nos permite trabajar con repositorios Git (Git es una herramienta de gestión de versiones). Lo primero que haremos será **crearnos una cuenta** (gratuita) en Bitbucket, proporcionando vuestro nombre y apellidos, un nombre de usuario y password de vuestra elección, y una dirección de correo electrónico (usad preferiblemente vuestro correo institucional).

Una vez creada la cuenta, tenéis que **crear UN ÚNICO repositorio** (usando el signo "+" del panel de la izquierda) que alojará TODO el trabajo de prácticas que realicéis durante el curso.

El repositorio tendrá como nombre: **ppss-2022-Gx-apellido1-apellido2**, en donde:

- ❖ **Gx** es el identificador del grupo de prácticas que tenéis asignado. Los valores posibles son: G1..G9, G40, según la siguiente tabla:

Grupo de prácticas	Identificador
Martes de 11 a 13h	G1
Lunes de 13 a 15h	G2
Martes de 9 a 11h	G3
Lunes de 09 a 11h	G4
Lunes de 17,30 a 19,30h	G5

Grupo de prácticas	Identificador
Miércoles de 19 a 21h	G6
Martes de 17 a 19h	G7
Miércoles de 9 a 11h	G8
Miércoles de 17 a 19h	G9
(I2ADE)Martes de 11 a 13h	G40

- ❖ **apellido1** es el primer apellido (todo en minúsculas y SIN acentos)
- ❖ **apellido2** es el segundo apellido (todo en minúsculas y SIN acentos). Si algún alumno no tiene segundo apellido, entonces omitiremos esta parte

The screenshot shows the Bitbucket 'Create repository' form. Annotations with red arrows point to specific fields:
 

- 'Untitled project' dropdown: 'Selecciona "Untitled Project"'
- 'Repository name' text box (containing 'ppss-2022-G...'): 'Indica el nombre del repositorio'
- 'Access level' dropdown (set to 'Private repository'): 'El repositorio será **PRIVADO**.'
- 'Include a README?' dropdown (set to 'No'): 'No incluyas un README.'
- 'Include .gitignore?' dropdown (set to 'Yes (recommended)'): 'Deja Yes en Include .gitignore'
- 'Advanced settings' link: 'Debes establecer, el Lenguaje usado a "Java" (desplegando las opciones avanzadas)'

Después de crear el repositorio, deberéis dar **permiso de lectura** al usuario **eli@gcloud.ua.es**, a través de las opciones "Repository Settings→GENERAL→Users and group access" de vuestro proyecto (desde el panel izquierdo).

## Git: repositorio (local, en nuestra máquina virtual)

**comando git clone**  
(para descargar una copia del repositorio)

Una vez creado el repositorio, vamos a "descargar y sincronizar" dicho repositorio remoto en un directorio de nuestra máquina virtual. A este proceso lo llamaremos CLONAR el repositorio remoto. Si trabajas en el laboratorio, este paso lo tendrás que hacer SIEMPRE.

Supongamos que nuestro directorio de trabajo va a ser **"\$HOME/practicas"**. Para **CLONAR** el repositorio en dicho directorio, necesitamos la URL del repositorio que acabamos de crear. Esta

información podemos consultarla desde la vista "<> Source", pulsando sobre el botón Clone (arriba a la derecha de esa pantalla), seleccionamos "https" y copiamos el comando en el portapapeles.

A continuación pegamos el comando en un terminal (desde el directorio \$HOME/practicass), y lo ejecutamos. Verás que se crea un directorio con el nombre de tu repositorio:

**\$HOME/practicass/ppss-2022-Gx-apellido1-apellido2**

Puedes comprobar que el nuevo directorio contiene a su vez un directorio oculto:

**\$HOME/practicass/ppss-2022-Gx-apellido1-apellido2/.git**

El **directorio .git** contiene nuestro repositorio local Git (que está conectado con el repositorio remoto en Bitbucket). A partir de ahora, cualquier fichero que añadamos en el directorio **\$HOME/practicass/ppss-2022-Gx-apellido1-apellido2/** podrá estar sujeto al control de versiones de Git.

También veremos el fichero oculto **.gitignore** (que se ha creado junto con el repositorio). Es un fichero de texto en el que se indican todos aquellos ficheros y/o directorios que serán "ignorados" por git. Es decir, que no serán "guardados" en el repositorio para gestionar los cambios realizados sobre ellos. Puedes ver y editar el contenido del fichero .gitignore.

Cuando construyamos nuestros proyectos java con Maven, se creará sistemáticamente el directorio **target**, que contendrá, entre otras cosas, los ficheros **.class** de nuestra aplicación. No tiene sentido guardar en Bitbucket información que podemos obtener en cualquier momento (volviendo a construir el proyecto), esto va a ocupar espacio en disco y consumir un tiempo innecesario (de subidas y descargas a/desde Bitbucket). Puedes comprobar que una de las líneas del fichero .gitignore es "target/", que significa que se ignorará el contenido de cualquier carpeta (y todo su contenido) con ese nombre.

A continuación vamos a **configurar Git**. Recuerda que este paso deberás hacerlo SIEMPRE cuando trabajes en los ordenadores del laboratorio

**comando git config**  
(ejecutar siempre en el laboratorio)

El comando **git config** nos permitirá guardar en el fichero .git/config algunos parámetros de configuración para trabajar con git. Concretamente nuestro nombre de usuario y e-mail. Si usamos nuestro ordenador sólo hay que hacerlo la primera vez.

Configuraremos nuestra **identidad** a través de nuestro nombre y dirección de correo electrónico con los siguientes comandos:

> **git config user.name <nombreUsuario>**

Siendo <nombreUsuario> el nombre que mostrará Git cuando hagamos un **commit**. Poned vuestro nombre y apellidos. P.ej. "Luis Lopez Perez" (si usáis espacios tenéis que poner dobles comillas)

> **git config user.email <emailUsuario>**

Siendo <emailUsuario> el email del usuario en Bitbucket (no es necesario poner comillas)

**comando git status**  
(para ver el estado de los ficheros)

El comando **git status** nos muestra el estado de los ficheros de nuestro directorio de trabajo (en **rojo** significa que el fichero todavía no está bajo el control de Git, o sea, que si lo borramos, lo perdemos).

Para que cualquier cambio realizado sobre un fichero de nuestro directorio de trabajo sea controlado por git, primero tenemos que "marcar" dicho fichero usando el comando **git add**. Y posteriormente guardaremos el fichero marcado en nuestro repositorio local con el comando **git commit**. Para subir (copiar) nuestro repositorio local a Bitbucket usaremos el comando **git push**.

**Comandos para "guardar"**  
**todo nuestro trabajo en local**  
**en Bitbucket**  
(ejecutar SIEMPRE)

```
> git add .
> git commit -m "Mensaje obligatorio"
> git push
```

**IMPORTANTE:** Siempre debes ejecutar los comandos git desde tu directorio de trabajo, es decir, desde el directorio que contiene el subdirectorio oculto .git

**IMPORTANTE:** Al ejecutar comandos git que accedan a Bitbucket se usará el valor de la propiedad user.email que has configurado, y te pedirá la contraseña. En lugar de usar la contraseña con la que creaste la cuenta en Bitbucket, tendrás que generar una desde Bitbucket, "*Profile and Settings*" (icono en el extremo inferior izquierda de la ventana), "*Personal Settings*", "*App Passwords*". Debes asignarle un nombre (el que tú quieras). Puedes marcar todos los permisos a nivel de proyecto. Una vez generada la contraseña, tendrás que recordarla y guardarla porque no la vas a poder volver a visualizar.

Si has hecho todo esto con **tu propio ordenador**, a partir de ahora, lo único que tendrás que hacer es utilizar los comandos add, commit y push para sincronizar (en tu repositorio remoto) los cambios realizados en local (suponiendo que nunca trabajas con el ordenador del laboratorio). Recuerda que SIEMPRE deberás hacerlo desde \$HOME/practicas/ppss-2022-Gx-apellido1-apellido2/ (aunque en cada práctica trabajemos en algún subdirectorío del mismo)

Si trabajas siempre utilizando los ordenadores del laboratorio, cuando llegues a casa y trabajes en tu ordenador, tendrás que (la primera vez):

**Secuencia de trabajo en el ordenador de casa**  
(sólo la primera vez)

1. clonar el repositorio de Bitbucket (**git clone**)
2. configurar nuestra identidad (**git config**)
3. trabajar en el directorio de trabajo (**ppss-2022-Gx-apellido...**)
4. subir los cambios a Bitbucket (**git add + git commit + git push**)

Para las siguientes veces: supongamos que has trabajado desde los ordenadores del laboratorio, y luego quieres seguir trabajando en casa (y en casa ya habías clonado previamente el repositorio). En ese caso tendrás que hacer primero un **git pull** desde el ordenador de tu casa para sincronizar (y descargarte) los cambios que hiciste en el laboratorio antes de continuar trabajando:

**Secuencia de trabajo en el ordenador de casa**  
(el resto de veces)

- > **git pull**
2. trabajar en el directorio de trabajo (**ppss-Gx-2022-apellido...**)
3. subir los cambios a Bitbucket (**git add + git commit + git push**)

## LICENCIA educativa IntelliJ

Para poder usar la versión Ultimate de IntelliJ necesitáis solicitar, cada uno de vosotros, una licencia educativa desde [www.jetbrains.com](http://www.jetbrains.com) (válida durante un año). Para ello tendréis que:

1. Crearos una cuenta. Para solicitar la licencia educativa necesariamente tendréis que proporcionar vuestro e-mail institucional. Recibiréis un correo con un enlace al que tendremos que acceder para confirmar la petición.
2. Una vez que entréis en vuestra cuenta, veréis una pantalla con el mensaje "No available Licences", y varios enlaces. Seleccionamos el enlace **Apply for a free student or teacher license for educational purposes**. Rellenamos la petición indicando que sois estudiantes (lógicamente). Y a continuación recibiréis de nuevo un correo para activar la licencia educativa.

Una vez obtenida la licencia, ésta será necesaria para poder ejecutar IntelliJ. Si usáis los ordenadores de los laboratorios tendréis que activarla SIEMPRE, puesto que cuando apagáis la máquina, NO se guarda ninguna información ni ningún cambio que hayáis podido hacer en la máquina virtual.

Para activar la licencia podemos hacerlo proporcionando nuestro e-mail y password de nuestra cuenta de JetBrains.