Universidad de Chile – Facultad de Ciencias Físicas y Matemáticas
CC3501: Modelación y Computación Gráfica para Ingenieros. Otoño 2020

Guía Git/GitLab

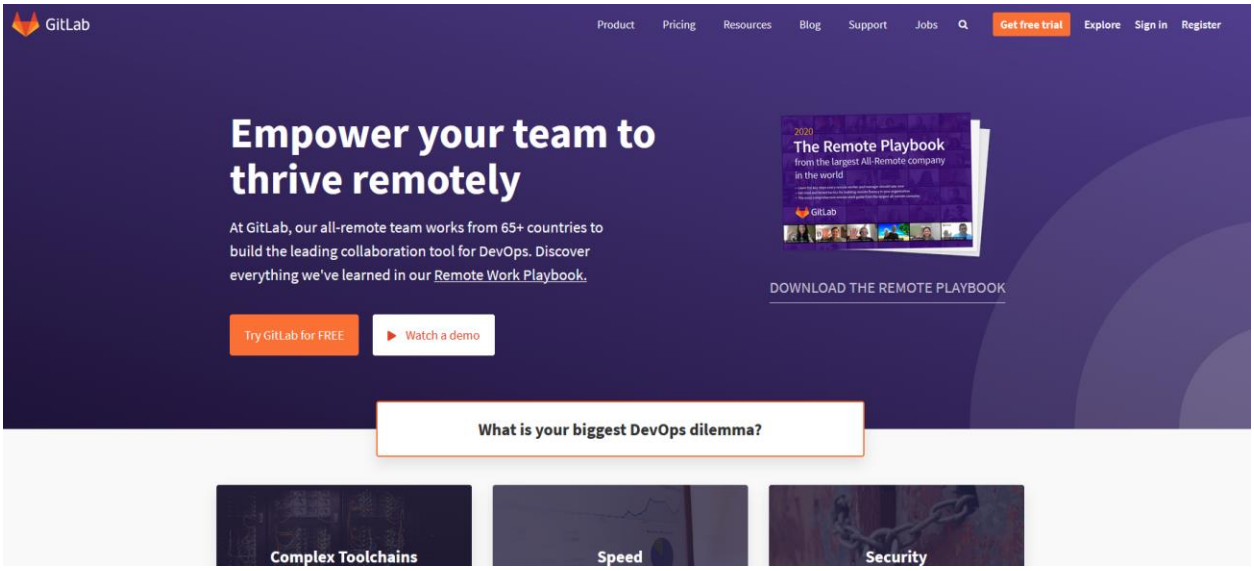
Introducción

En CC3501 utilizaremos el sistema de control de versiones Git de la plataforma GitLab. La entrega de su tarea se realiza al trabajar por esta vía. Usted debe crear una cuenta en este sistema (www.gitlab.com) y enviar su nombre de usuario al equipo docente (vía tareas u-cursos). El equipo docente le asignará un repositorio privado sobre el cual podrá trabajar su tarea.

Configuración del espacio de trabajo

1. Crear cuenta GitLab

Ingrese a www.gitlab.com y cree una cuenta. La opción gratuita de GitLab es suficiente para los fines del curso. Debe enviar un archivo de texto (extensión .txt) vía la sección tareas de u-cursos con su nombre de usuario. El plazo límite para enviar su repositorio es indicado en la misma sección.



2. Asignación del Repositorio

Una vez enviado su usuario. El equipo docente le asignará un repositorio privado donde usted podrá trabajar. Cuando se realice esta asignación, recibirá un correo desde GitLab automáticamente.

Su repositorio tendrá la forma:

https://gitlab.com/cc3501/20201/students/student4

Para trabajar en él, podrá clonarlo vía SSH con:

```
git clone git@gitlab.com:cc3501/20201/students/student4.git
```

- "student4" será reemplazado por un identificador único para usted.

- Si no está familiarizado con git, en las siguientes secciones se explica lo que es necesario aprender :).

3. Espacio de trabajo para tareas

En este repositorio usted deberá crear una carpeta por cada tarea realizada. Cada carpeta de tarea debe utilizar un nombre correcto según la opción que escoja. A modo de ejemplo, puede al finalizar el semestre puede tener las carpetas: tarea1a, tarea2c, tarea3b y tarea4.

Cada carpeta de tarea debe ser absolutamente independiente. Incluya todo el código y archivos que necesite dentro de ella, incluyendo material que haya sido entregado en cátedras o clases auxiliares y que esté utilizando.

Instalación de Git

Si ya ha instalado git antes en su sistema operativo, puede utilizar dicha instalación.

En caso de que aún no lo haya instalado:

- En Windows, descargue el software desde: https://gitforwindows.org/
- En Debian/Linux ejecute en una terminal: `sudo apt-get install git`

Más instrucciones de instalación en: https://www.atlassian.com/git/tutorials/install-git

Commandos Git Básicos

Para trabajar en sus tareas, usted solo necesita los comandos: clone, add, commit y push.

1. Para comenzar a trabajar, clone su nuevo repositorio en algún directorio de su computador.
   - En Windows, con clic derecho en la carpeta, seleccione “Git Bash here” para abrir una terminal git.
   - En Linux, una vez instalado git, simplemente vaya al directorio mediante la terminal.

   Luego ejecute:

   ```
   git clone git@gitlab.com:cc3501/20201/students/student4.git
   ```

   Se le pedirá su contraseña de GitLab. En el directorio se creará una carpeta con los archivos del repositorio. Inicialmente es solo un archivo de texto llamado “README.md”.

2. Trabaje modificando sus archivos. Luego, abra la terminal de git, y añada estos cambios con:

   ```
   git add --all
   ```

   Nota: --all agrega todos sus cambios. Puede agregar los cambios archivo a archivo independientemente.
3. Luego, guarde estos cambios en su repositorio Git local ejecutando commit

   ```
   git commit -m “Nota explicando sus cambios”
   ```

4. Para enviar este paquete de cambios al servidor remoto de GitLab, ejecute

   ```
   git push origin master
   ```

Se le pedirá su contraseña cada vez que ejecute este comando.

5. Si por casualidad pierde su repositorio, o si desea trabajar desde otro computador o directorio, puede simplemente clonar su repositorio nuevamente con el paso 1.
6. Siempre puede verificar el contenido del repositorio remoto en su sesión online de GitLab. Esto es lo mismo que puede ver el equipo docente.

Comandos Git "Avanzados"

Los siguientes comandos pueden serle de ayuda, pero no son necesarios para la tarea.

1. Si trabaja desde dos computadores o carpetas, producirá desfase entre los cambios. Puede actualizar/sincronizar los archivos en su carpeta con pull.

```
git pull origin master
```

   Si modifica ambas copias locales, sin sincronización, producirá diferencias que deberán ser resueltas caso a caso. Por simplicidad evite esta situación hasta que domine bien el manejo de este escenario con git.

2. Puede conocer el historial de cambios con el comando log. Se especificaran los commits con sus mensajes y número de commit asociado.

```
git log
```

3. Si quiere volver a algún cambio anterior, puede utilizar el comando git reset. Existen las opciones hard y soft. Con hard perderá los cambios posteriores al punto de sincronización, mientras que con soft los cambios posteriores quedarán como cambios locales para su revisión.

```
git reset –soft 0ad5a7a6
```

4. Si solo desea obtener una vista de su repositorio en algún cambio anterior, sin modificar nada, puede utilizar el comando checkout

```
git checkout -b old1 0ad5a7a6
```

   Este comando crea una rama (branch) de nombre `old1`, que puede seguir trabajando de manera paralela. Para volver al último estado de su repositorio, ejecute

```
git checkout master
```

   "master" es el nombre de su rama principal. Su intuición es correcta, puede trabajar en muchas ramas. Cada rama tendrá sus propios cambios. El comando checkout lo llevará a la rama en la que desee trabajar. Puede mezclar ramas con el comando `merge`, pero dejaremos eso para otra ocasión.

   Para enviar otra rama al servidor remoto, utilice

```
git push -u origin branch_name
```

Modo de trabajo

En resumen, cada vez que trabaje en su tarea:

1. Agregue sus cambios con: `git add --all`
2. Agréguelos al paquete de cambios actuales con: `git commit -m "Descripcion"`
3. Envíelos a su repositorio remoto con: `git push origin master`
4. Siempre puede verificar el contenido del repositorio remoto en su sesión online de GitLab.

De esta forma, tendrá su trabajo siempre respaldado.