



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN  
IIC2233 - PROGRAMACIÓN AVANZADA

# Actividad 00

1º semestre 2018

8 de marzo

## Git: Un sistema de control de versiones

### *Cheatsheet:* Operaciones básicas

1. Clonar: `git clone <url del repositorio>`
2. Revisar estado: `git status`
3. Agregar un archivo al *staging area*: `git add file.txt`
4. Commitear al repo local: `git commit -m "Descripción del commit"`
5. Deshacer cambios de un archivo en el *staging area*: `git reset HEAD file_name`
6. Deshacer último commit: `git reset HEAD~1`
7. Actualizar repo remoto: `git push`
8. Actualizar repo local: `git pull`

### Parte 1: Hello World!

El objetivo de esta parte es tener un primer contacto con el sistema de control de versiones que se utiliza en el curso. Para ello, debes ejecutar los siguientes pasos:

1. Clona tu repositorio. Si no lo tienes, regístrate en el curso para obtenerlo.
2. Dentro de la carpeta `Actividades/AC00/` crea un archivo `hello.py` cuyo código imprima en pantalla `"Hello World!"`.
3. Revisa el estado del repositorio. Observa que tu archivo recién creado aparece bajo "Untracked files".
4. Agrega el archivo `hello.py` al *staging area*, luego revisa el estado del repositorio.
5. Haz *commit* de los cambios realizados. Recuerda escribir un mensaje **descriptivo**.
6. Ahora, haz *push* para escribir tus cambios en GitHub. Luego, ve el contenido del repositorio en un *browser* y verifica tus cambios.

## Parte 2: Recordando las clases

El objetivo de esta parte es recordar el contenido de programación orientada a objetos (OOP), en particular el concepto de clases visto en el curso de *Introducción a la programación*.

Usted deberá usar clases para modelar **círculos** y **rectángulos** en un plano cartesiano bidimensional. Para esto, será necesario guardar en qué coordenadas están situadas las figuras y cuáles son sus dimensiones. Ambas figuras deberán tener métodos para obtener su área y su perímetro. Además, para el caso del rectángulo, deberá existir un método que indique si corresponde a un cuadrado o no. Por último, ocupe el método `__str__` para que, al imprimir en pantalla una instancia de estas clases, se lea qué tipo de figura es la instanciada y cuáles son sus dimensiones.

### Notas

- Haga un *commit* cada vez que una idea lógica esté terminada.
- Recuerde que Python es un lenguaje con baterías incluidas: no dude en utilizar el módulo *built-in* llamado `math`, puesto que ofrece múltiples funciones que podrían serle de utilidad para desarrollar esta actividad.

### Requerimientos

- Clase `Circle`
  - Crear el inicializador (método `__init__`)
  - Crear el método `get_area`
  - Crear el método `get_perimeter`
  - Crear el método `__str__`
- Clase `Rectangle`
  - Crear el inicializador (método `__init__`)
  - Crear el método `get_area`
  - Crear el método `get_perimeter`
  - Crear el método `is_square`
  - Crear el método `__str__`
- Crear cuatro instancias diferentes (dos por cada clase)

### Entrega

- **Lugar:** En su repositorio de GitHub en la **carpeta** `Actividades/AC00/`
- **Hora:** 16:20