

Programación Avanzada

II C2233

Nebil Kawas – Belén Saldías – Cristian Ruz – Jaime Castro



Agenda

- Programa del curso
- Herramientas del curso
- Recomendaciones para pasar el curso
- Repaso de *Introducción a la programación*

Programa



Este curso enseña técnicas para diseñar, implementar, ejecutar y evaluar software que resuelva problemas a partir de especificaciones detalladas.

Objetivos

1. **Descomponer problemas complejos**, para diseñar y estructurar sus soluciones.
2. Crear **diseños orientados a objetos y comunicar** estos diseños a través de documentación externa y comentarios en el código.
3. Aplicar conceptos de orientación a objetos y **estructuras de datos** fundamentales, para diseñar y escribir programas complejos en el lenguaje de programación Python, pudiendo **extender este conocimiento** a distintos lenguajes.
4. **Usar herramientas de programación estándares**; técnicas de programación; y un entorno de desarrollo de software para editar, ejecutar y depurar programas.
5. Generar software **desde cero**, con código de alto nivel y calidad, de fácil re-utilización, actualización y mantenimiento. Incluyendo también interfaces gráficas totalmente funcionales.

Contenidos del curso

- Programación orientada a objetos
- Estructuras de datos
- Excepciones
- *Testing*
- Programación funcional
- Metaclases
- Simulación
- *Threading*
- Interfaces Gráficas
- I/O (*strings*, *bytes*, serialización)
- *Networking*
- Contenidos adicionales

Programación avanzada vs.

Exploratorio de computación



Introducción a la programación



Comenzar su formación como desarrolladores de software



Metodología



FLIPPED

CLASSROOM

Flipped Classroom



Actividades

Antes de clases

Equipo docente sube material de la clase para que ustedes lo **estudien** y resuelvan dudas en las *issues*.

Antes de la actividad

Repaso de la materia. Resuelven más dudas con los profesores.

Durante la actividad

Resuelven **dudas** de materia y actividad con ayudantes y profesores. Las actividades serán en parejas o individuales.

Actividades

- Tendrán puesto asignados y se publicarán en el [syllabus](#).
- Los ayudantes pasarán lista: si están marcados como ausentes tendrán un 1,0 (uno) en esa actividad.
- Es responsabilidad de ustedes verificar que los ayudantes los hayan puesto presentes.
- No se recuperan actividades.

Tareas

Tienen por objetivo que resuelvan un problema complejo.

1. Leer el enunciado.
2. Hacer el entregable que les pidan.
(4 a 5 días)
3. Reciben *feedback* general sobre el entregable.
(2 a 3 días después)
4. Tienen una semana para **seguir** desarrollando la tarea.

Controles y examen

- Los controles duran entre 10 y 20 minutos.
- Son similares a las preguntas más fáciles del examen.
- Enfocados a la lectura de código.
- El examen es el día **6 de julio a las 15:30** y durará ~2,5 horas.

Evaluaciones

- 14 actividades (**A**)
- 6 o 7 tareas (**T**)
- 8 controles (**C**)
- 1 examen final (**E**)

$$NP = 0,20 \times \mathbf{A} + 0,45 \times \mathbf{T} + 0,15 \times \mathbf{C} + 0,20 \times \mathbf{E}$$

Evaluaciones

La nota de tareas para cada alumno será calculada como

$$T = \frac{\sum_{i=1}^{6+s} p_i \times t_i}{\sum_{i=1}^{6+s} p_i}$$

$s :=$ inscribe la tarea 7

Nota tarea t_i	Ponderador p_i
t_1	70
t_2	85
t_3, t_4, t_5, t_6	100
t_7 (opcional, con inscripción)	100

Evaluaciones

- Adicionalmente para aprobar el curso el alumno debe cumplir con:

$$E \geq 3,500$$

$$A \geq 3,950$$

$$T \geq 3,950$$

- Si el alumno cumple con las condiciones anteriores
NF = NP. En caso contrario, **NF = min(3,9 ; NP)**

Evaluaciones

- La inasistencia a alguna de las evaluaciones (actividad, control y examen) se evalúa con nota 1.0.
- Se eliminan:
 - las dos peores notas de actividades.
 - el peor control.
- **NO se borrará ninguna otra evaluación.**
- Solo será aproximada la nota final NF.
El resto de las notas serán usadas con dos decimales.

Correcciones y recorreciones

- Las notas de actividades y tareas se publican **a más tardar 15 días** hábiles después de haber realizado la evaluación.
- Tendrán **una semana** para corregir después de que se publiquen las notas.
- Los controles se van a buscar al DCC. Para corregirlos, deben escribir su solicitud en un papel y se lo entregan a Yessenia (secretaria del DCC).
- Las actividades y tareas se recorrigien a través del [formulario de corrección](#).

Solicitud de corrección

- La nota **puede bajar**.
- No se aceptarán correcciones del tipo:
 - “Me merezco más puntaje”.
 - “El ayudante dice que mi programa no corre, pero a mí sí me funciona”.
- La solicitud de corrección debe indicar cuáles fueron los puntos mal corregidos y por qué están mal corregidos.
- Si no están de acuerdo con la respuesta de los ayudantes después de la corrección, deberán esperar hasta el día de la corrección final para resolver el problema.

La recorreción final es
el **10 de julio**.
NO HAY OTRA FECHA.

En la recorreción final no se corregirán evaluaciones que no fueron mandadas a corregir en el período que correspondía.

Normas en evaluaciones

No respetar las indicaciones de cada evaluación tiene como sanción inmediata un 1,0 en dicha evaluación.

Por ejemplo: cambios arbitrarios en las parejas de trabajo, no respetar los medios de entrega de evaluaciones, formatos, etc.

Integridad académica

*“Cualquier situación de copia en alguna evaluación tendrá como **sanción un 1,1 final en el curso**. Esto sin perjuicio de sanciones posteriores que estén de acuerdo a la Política de Integridad Académica de la Escuela de Ingeniería y de la Universidad, que sean aplicables para el caso.”*

Integridad académica

- Deben **indicar la fuente** de cualquier código que encuentren en internet y que usen en sus tareas y/o actividades.
- Deben indicar si están usando código del material del curso o de las ayudantías.
- Si no lo hacen, se considerará plagio.

Fechas

<https://iic2233.github.io/calendario/>

Cuerpo docente



Nebil (S1)



Belén (S2)



Cristian (S3)



Jaime (S4)

Ayudantes jefes



Florence



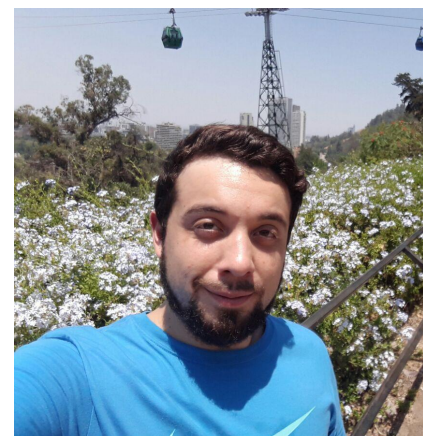
Fernando



Hernán



Joaquín



Benjamín

Ayudantes

Tareas

- **Hugo Navarrete**
- **Ignacio Acevedo**
- **Camilo López**
- **Jessica Hormazábal**
- Ricardo Schilling
- Rodrigo Hanuch
- Pablo Flores
- Leonardo Olivares
- Yoav Navón
- Mónica Ballesteros
- Claudio Scheihing
- Sebastián Behrmann
- Ignacio Hermosilla
- Javier Bahamondes
- Martín Anselmo
- Sebastián Orrego Mora
- Ignacio Contreras

Docencia

- **Sebastián Guerra**
- **Enzo Tamburini**
- **Gabriel Lyon**
- Pablo Olea
- Benjamín Earle
- Nicolás Villalobos
- Franco Bruña
- Diego Quezada
- Tomás Rivera
- Javier Dreves
- José Pablo Domínguez
- Tien Villalobos
- Alfredo De Goyeneche

Optimización de salas

Objetivo: 55 por sección

Ingeniería											Vacantes		
NRC	Sigla	Permite Retiro	¿Se dicta en inglés?	Sec.	¿Requiere Aprob. Especial?	Categoría	Nombre	Profesor	Campus	Créd	Total	Disponibles	Reservadas
10760	 IIC2233	SI	NO	1	NO		Programación Avanzada	Kawas Nebil	San Joaquín	10	70	2	
10754	 IIC2233	SI	NO	2	NO		Programación Avanzada	Saldias Belen	San Joaquín	10	70	10	
18284	 IIC2233	SI	NO	3	NO		Programación Avanzada	Ruz Cristian	San Joaquín	10	70	1	
20252	 IIC2233	SI	NO	4	NO		Programación Avanzada	Castro Jaime	San Joaquín	10	60	38	

Consultas

- Administrativas:

ayudantesiic2233@gmail.com

- Contenidos del curso, enunciados y pautas

<https://github.com/IIC2233/Syllabus/issues>

NO MANDEN MAILS A LAS DIRECCIONES PERSONALES

Herramientas del curso



Python

<https://www.python.org/>

<https://zen-of-python.info/>



Guido van Rossum, creador de Python, en la convención OSCON 2006. Fuente: [Wikipedia](#).

PEP8

Guía de estilo

PEP8

- *Python Enhancement Proposal 8* es la guía de estilo de Python
- Se usa para hacer más legible y consistente el código
- <https://www.python.org/dev/peps/pep-0008/>

PEP8

- *Imports* al comienzo del módulo.
- Nombres de variables descriptivos.
- Espacios entre líneas
 - 2 líneas después de los *imports*
 - 2 líneas alrededor de las clases y funciones
 - 1 línea entre métodos de clase
 - 1 espacio después de “,” y a cada lado de los operadores
- Líneas de máximo 80 caracteres (incluyendo espacios).
- **No** usar *tabs*. Sólo usar espacios.

CamelCase y snake_case

```
CONST_PI = 3.1415
```

```
class ClaseDeEjemplo:  
    def __init__(self, parametro):  
        self.variable_de_ejemplo = parametro  
  
    def metodo_de_ejemplo(self):  
        return 1 + 1 == 2
```

**Siempre recuerda
que el código se
lee más veces de
lo que se escribe
y que es otro el
que lo va a leer.**

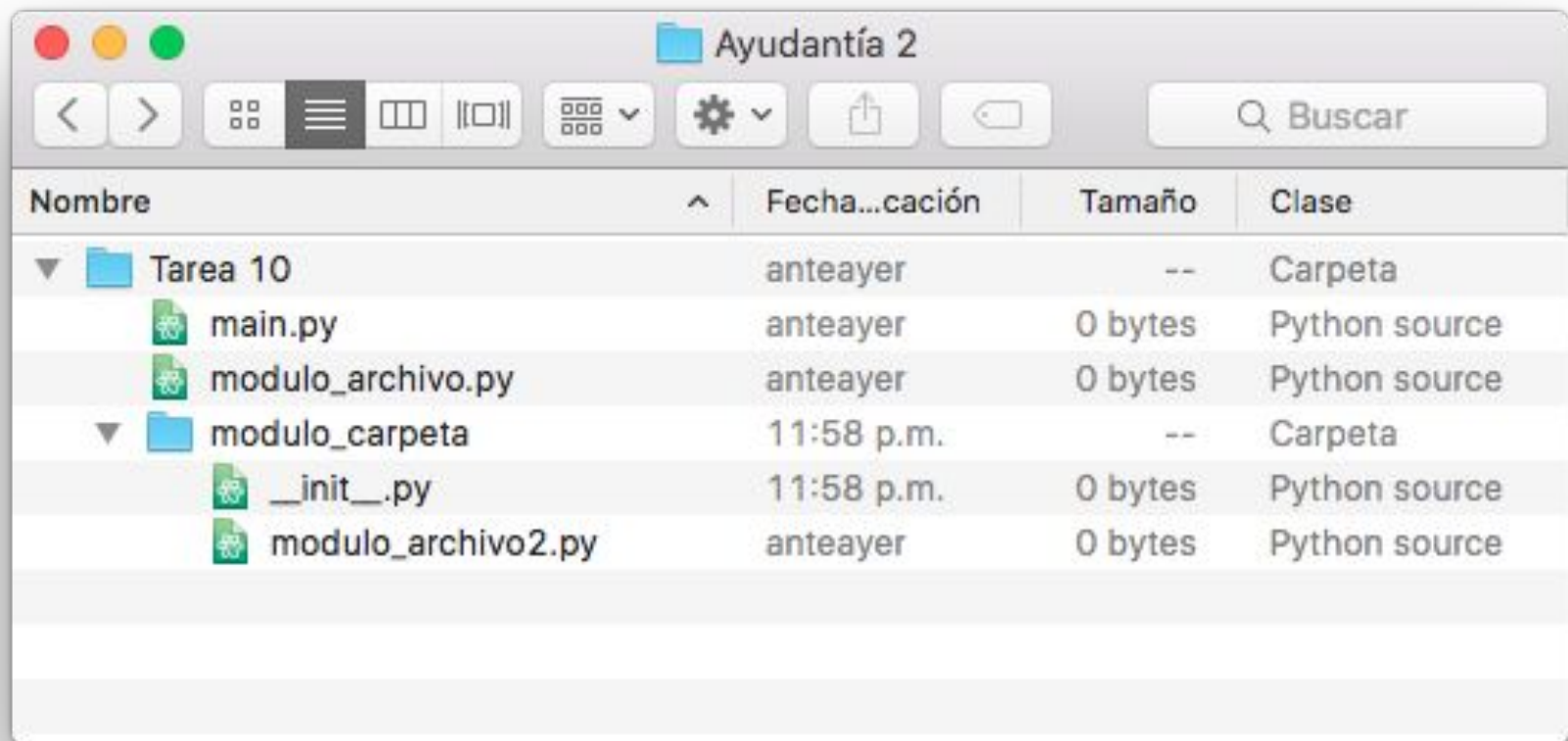
Modularización

—

Modularización: ¿por qué?

- Cuando un programa crece, se hace inviable mantenerlo en un solo archivo:
 - El mantenimiento es difícil
 - El trabajo en equipo es difícil
 - Es desordenado
- Un módulo es un archivo de Python normal, y puede tener:
 - Variables
 - Métodos
 - Clases

Modularización



Cómo usar módulos

Importándolo entero

```
import modulo_archivo
```

```
if __name__ == '__main__':
```

```
    variable_tipica = modulo_archivo.VALOR_FIJO
```

```
    objeto_tipico = modulo_archivo.Clase()
```

```
    modulo_archivo.funcion()
```

Cómo usar módulos

Importándolo entero con un alias

```
import modulo_archivo as ma
```

```
if __name__ == '__main__':  
    variable_tipica = ma.VALOR_FIJO  
  
    objeto_tipico = ma.Clase()  
  
    ma.funcion()
```

Cómo usar módulos

Importando lo necesario

```
from modulo_archivo import VALOR_FIJO, Clase, funcion
```

```
if __name__ == '__main__':  
    variable_tipica = VALOR_FIJO  
  
    objeto_tipico = Clase()  
  
    funcion()
```

Cómo usar módulos

- Cuando se importa un módulo se ejecuta todo el código en él
- Para evitar que se ejecute código de un módulo al ser importado se utiliza el siguiente `if`:

Código del módulo

```
if __name__ == '__main__':
```

```
    # Mucho código escrito
```

Cómo **NO** usar módulos

Importando todo sin referencia al módulo

```
from modulo_archivo import *
```

```
if __name__ == '__main__':  
    variable_tipica = VALOR_FIJO  
    objeto_tipico = Clase()  
    funcion()
```



Cómo **NO** usar módulos

- Evita crear módulos que se llamen igual a los que vienen incluidos en Python.
- ¿Cómo Python busca los módulos?
 1. Módulo de la librería estándar
 2. Módulo en la misma carpeta
 3. Módulo en el directorio de instalación

Git

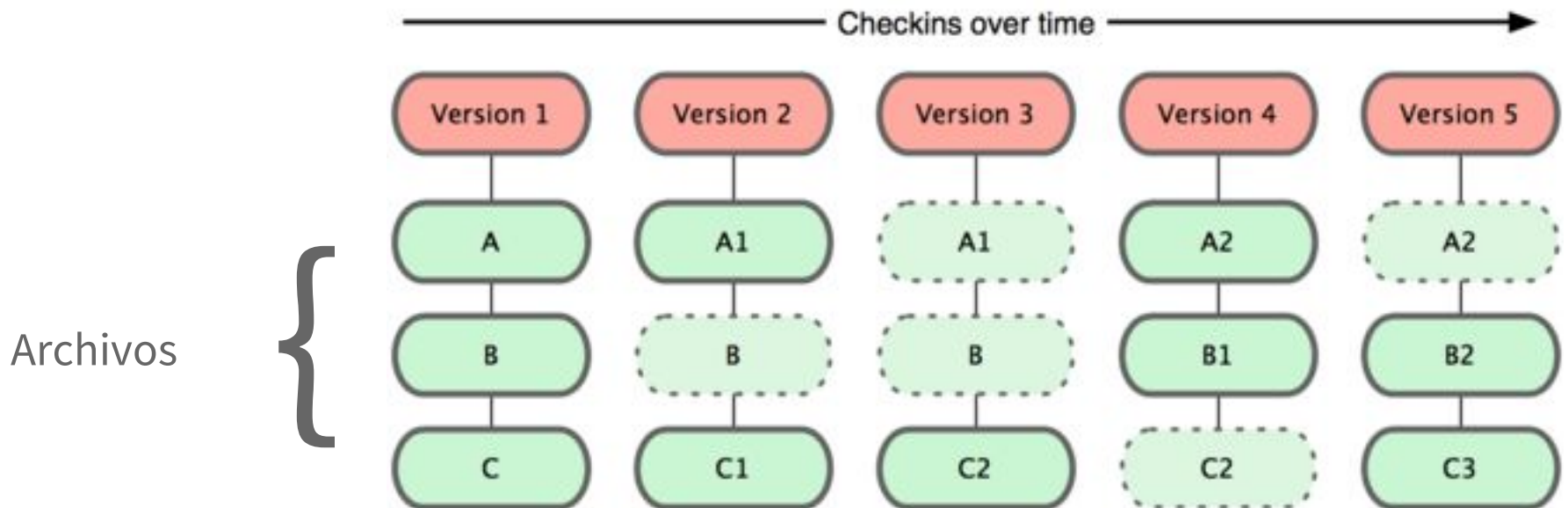


¿Qué es Git?

Git es un sistema distribuido de control de versión, gratuito y *open source*, diseñado para manejar de pequeños a enormes proyectos de forma rápida y eficiente¹.



¹ <https://git-scm.com/>



Fuente: git-scm.com

Ventajas

- Trabajo en equipo fluido (sin problemas como en Dropbox)
- Versiones disponibles en cualquier momento
- Control de cambios
- Programar versiones en paralelo y luego juntarlas
- Múltiples *backups* de sus programas

¿Qué es GitHub?

Es una plataforma para alojar proyectos, usando el sistema de control de versiones Git.



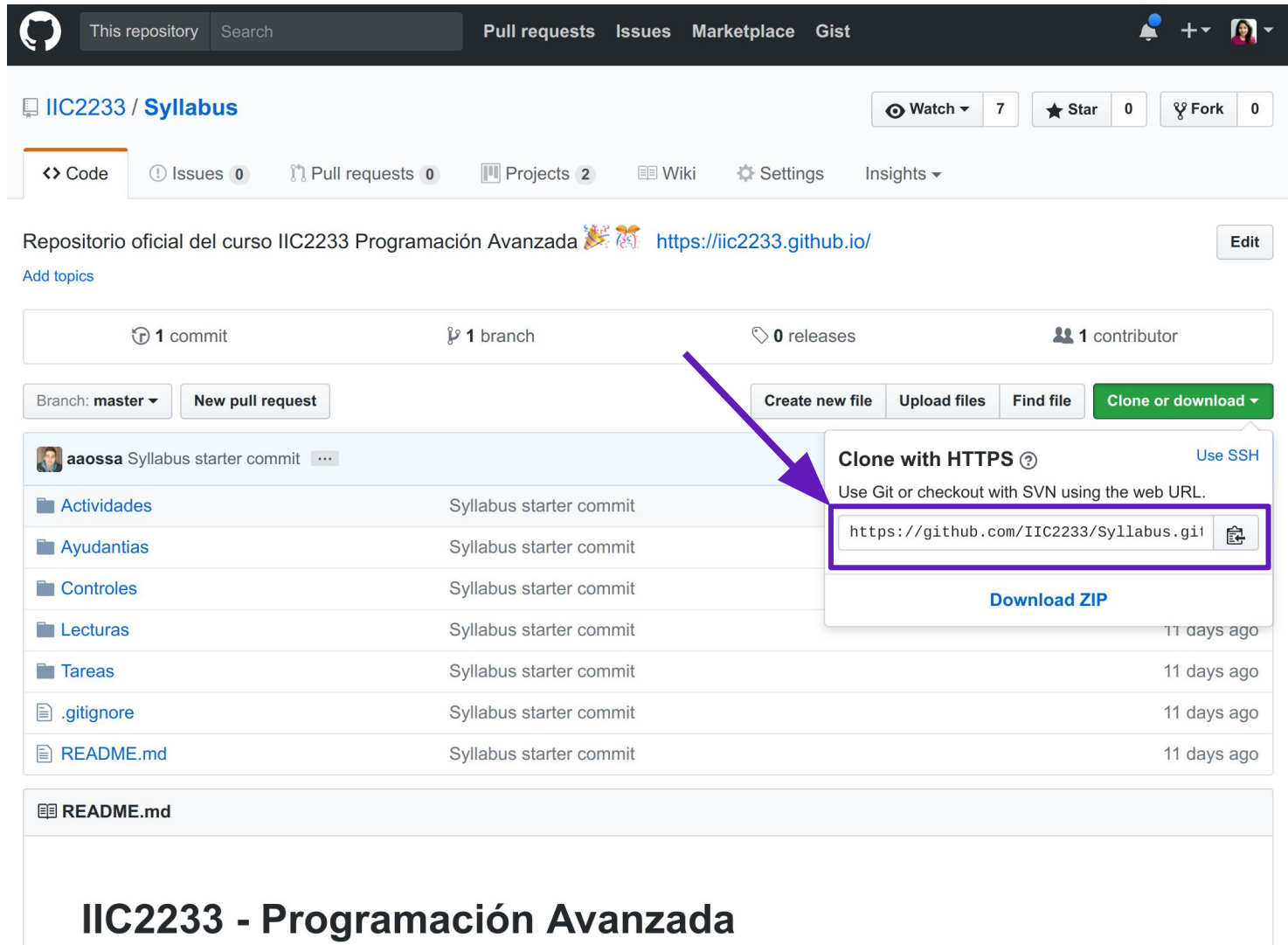
Se usa en la vida real.

Es **obligatorio** conocerlo.

Setup

—

Obtener dirección



The screenshot shows the GitHub interface for the repository 'IIC2233 / Syllabus'. The repository has 7 watches, 0 stars, and 0 forks. The 'Code' tab is selected, showing the repository's metadata and a list of files. A purple arrow points to the 'Clone with HTTPS' button, which is highlighted with a purple box. The URL 'https://github.com/IIC2233/Syllabus.git' is displayed in the box. Below the box, the 'Download ZIP' button is visible.

Repository: IIC2233 / Syllabus

Watch 7 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 2 Wiki Settings Insights

Repositorio oficial del curso IIC2233 Programación Avanzada <https://iic2233.github.io/> Edit

Add topics

1 commit 1 branch 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file Clone or download

Clone with HTTPS Use Git or checkout with SVN using the web URL.

`https://github.com/IIC2233/Syllabus.git`

Download ZIP

File	Commit	Time
Actividades	Syllabus starter commit	11 days ago
Ayudantias	Syllabus starter commit	11 days ago
Controles	Syllabus starter commit	11 days ago
Lecturas	Syllabus starter commit	11 days ago
Tareas	Syllabus starter commit	11 days ago
.gitignore	Syllabus starter commit	11 days ago
README.md	Syllabus starter commit	11 days ago

README.md

IIC2233 - Programación Avanzada

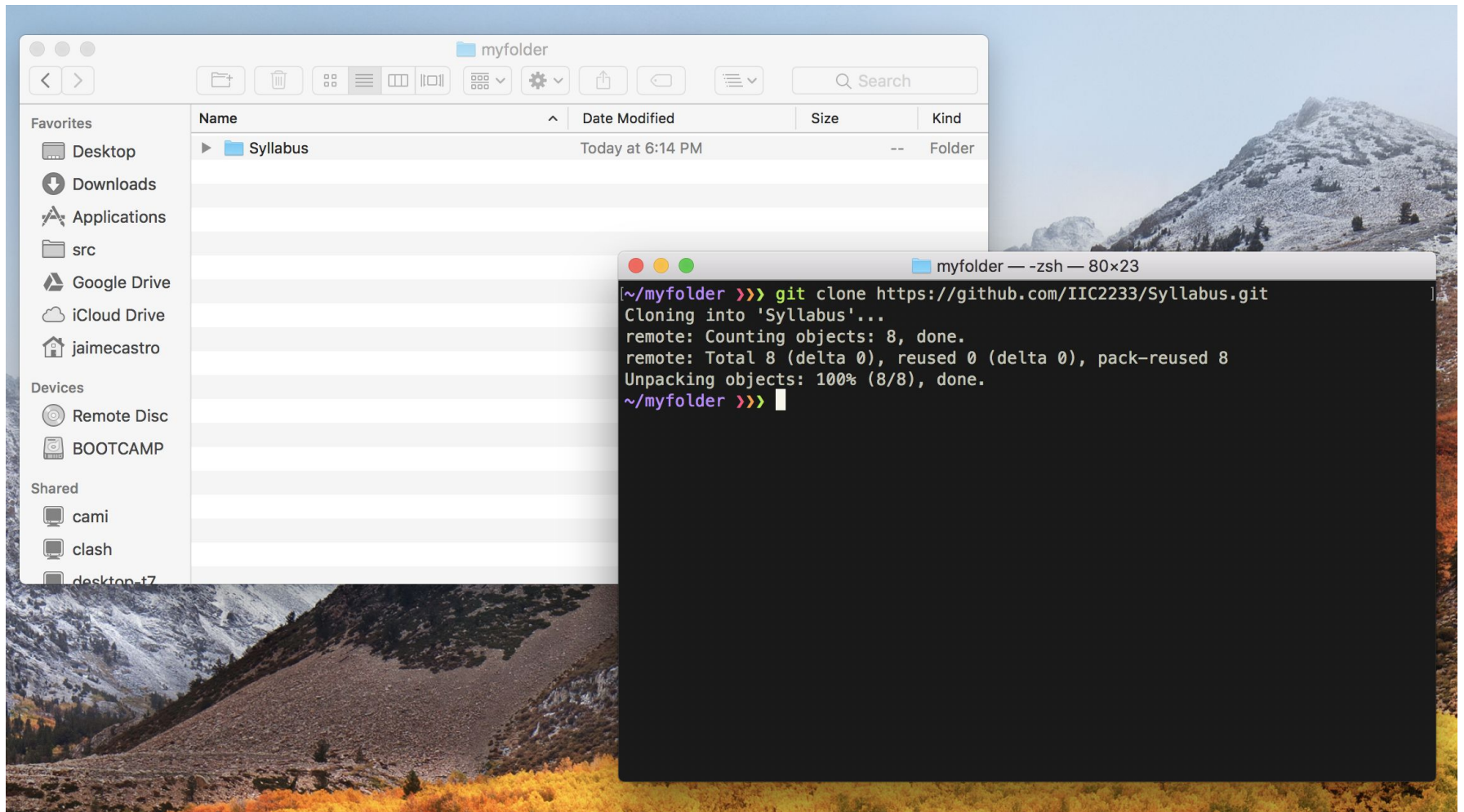
Clonar el repositorio

Escribe en la consola

```
git clone https://github.com/IIC2233/Syllabus.git
```

Recuerda estar en la carpeta en la que quieren mantener el repo.

Clonar el repositorio

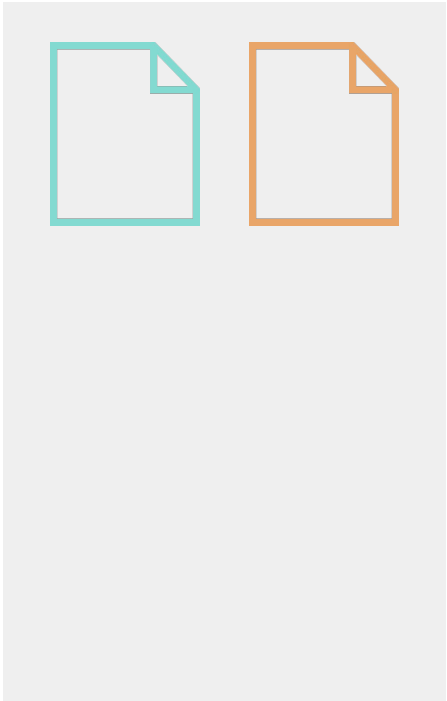


¿Cómo funciona Git y GitHub?

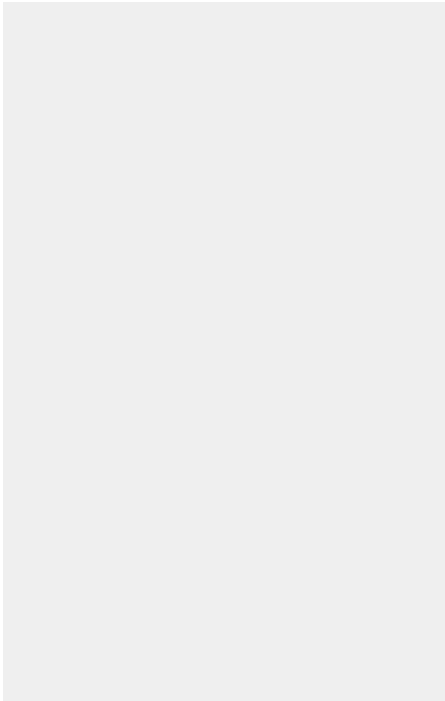
Conceptos

1. ***Working directory***: lugar local en donde están los archivos que luego serán parte del repositorio.
2. ***Staging area***: lugar en dónde están los archivos que se van a ser incluidos en un *commit*.
3. **Repositorio local (o repo)**: lugar local que contiene todos los archivos que han sido “commiteados”.
4. **Repositorio remoto**: lugar remoto (en un servidor) que contiene todos los archivos que han sido “commiteados”. Para esto utilizamos el servicio de GitHub.

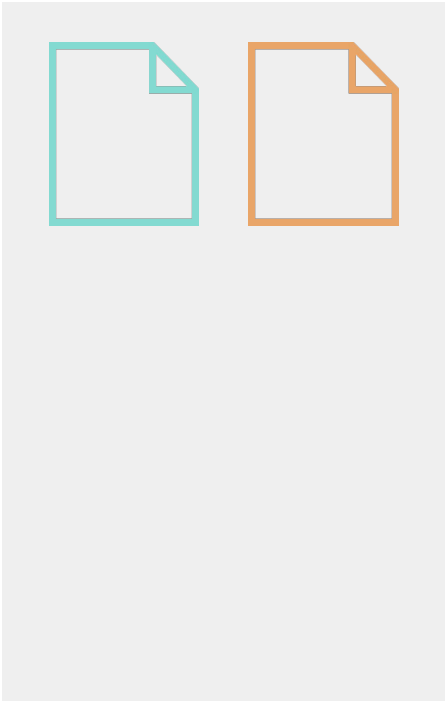
Working directory



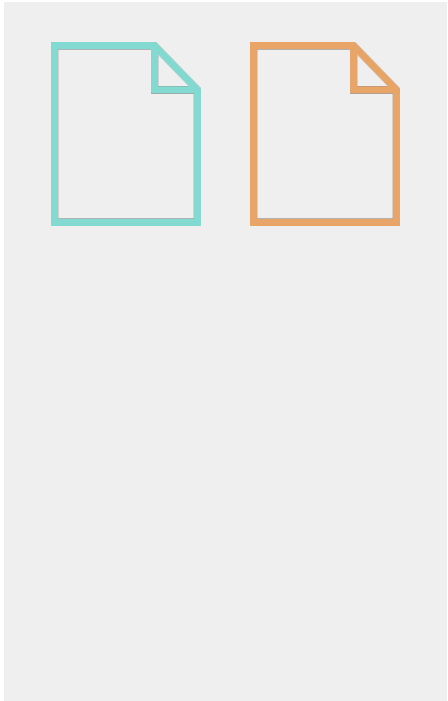
Staging area



Repositorio local



Repositorio remoto

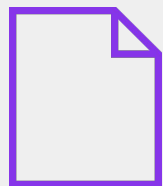
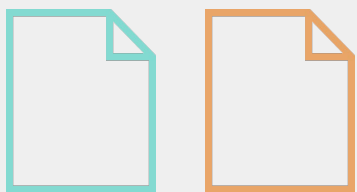


Working directory

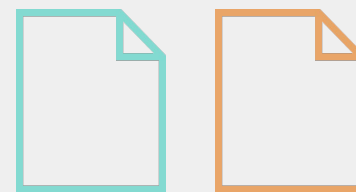
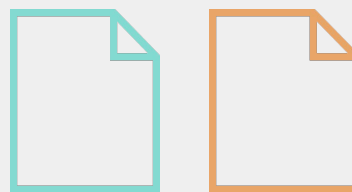
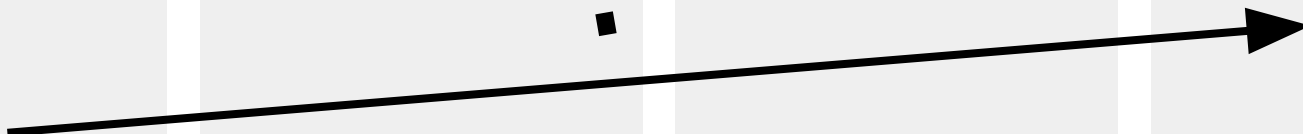
Staging area

Repositorio local

Repositorio remoto

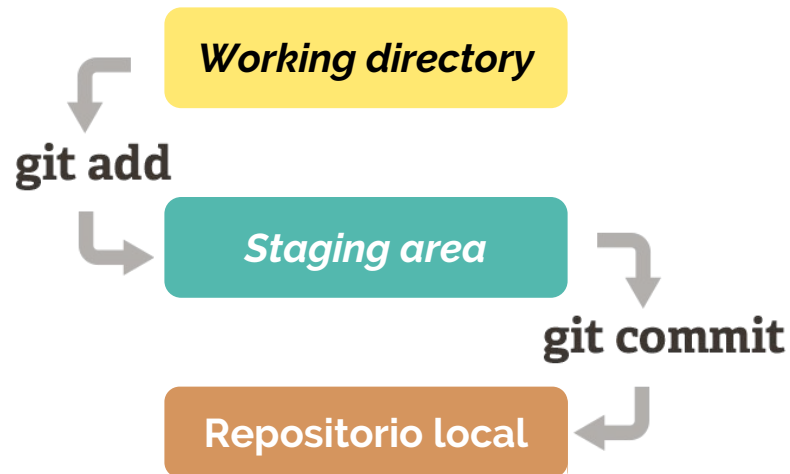


?



Crear una versión

```
git add file_name  
git add *.txt
```



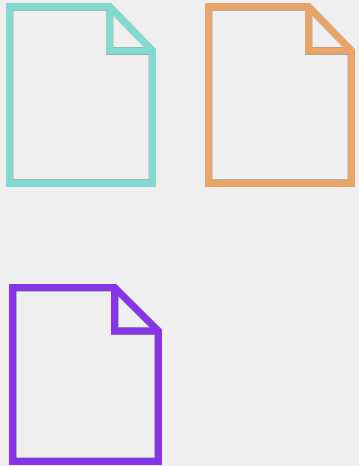
```
git commit -m "Mensaje descriptivo"
```


Los mensajes son MUY importantes. Son una ayuda a ustedes en el futuro.

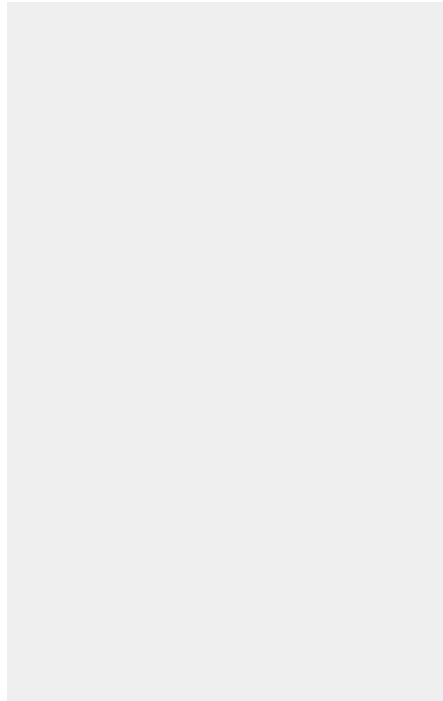
Revisen esta [guía de estilo](#)

Subir un archivo

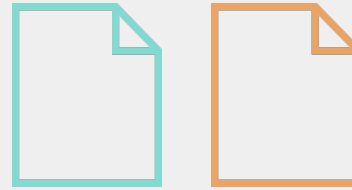
Working directory



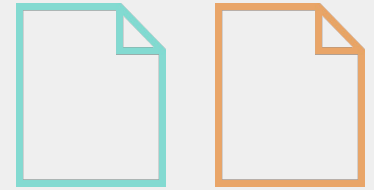
Staging area



Repositorio local



Repositorio remoto



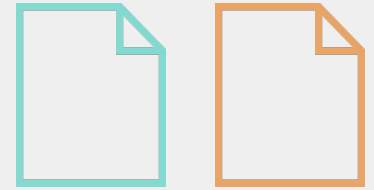
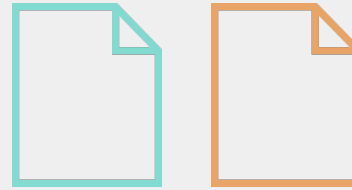
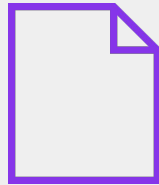
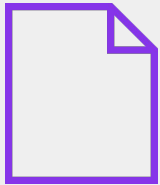
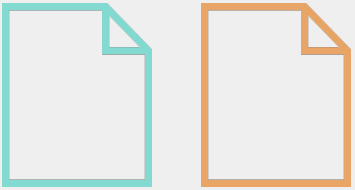
Subir un archivo

Working directory

Staging area

Repositorio local

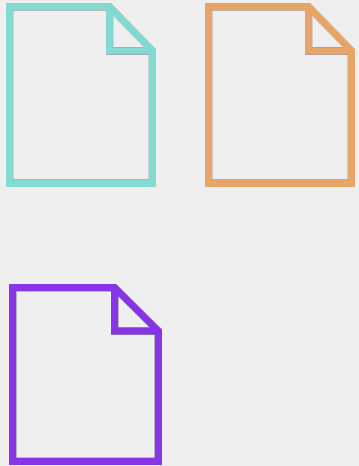
Repositorio remoto



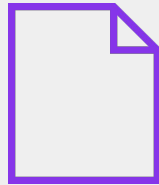
git add

Subir un archivo

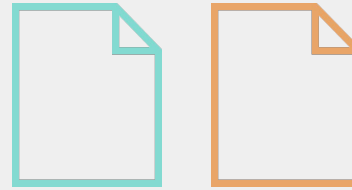
Working directory



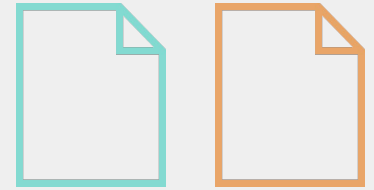
Staging area



Repositorio local



Repositorio remoto



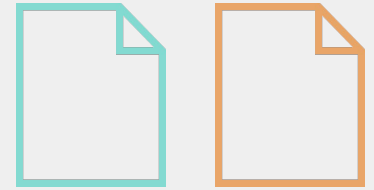
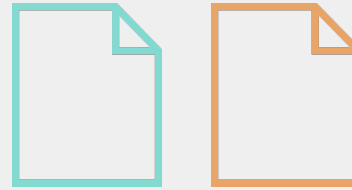
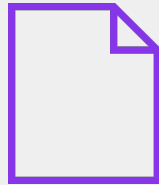
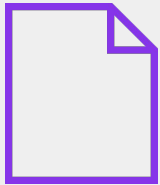
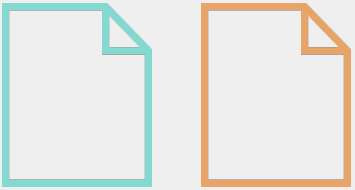
Subir un archivo

Working directory

Staging area

Repositorio local

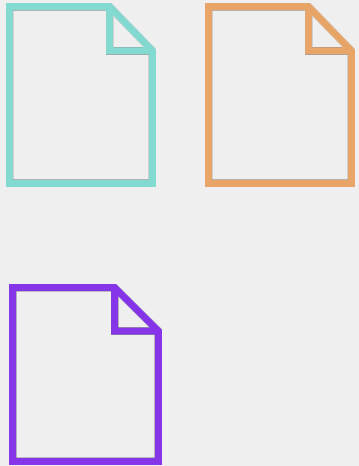
Repositorio remoto



git commit

Subir un archivo

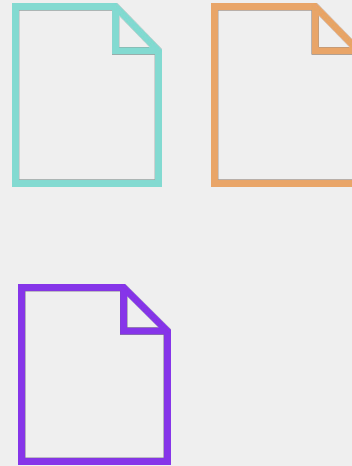
Working directory



Staging area



Repositorio local



Repositorio remoto



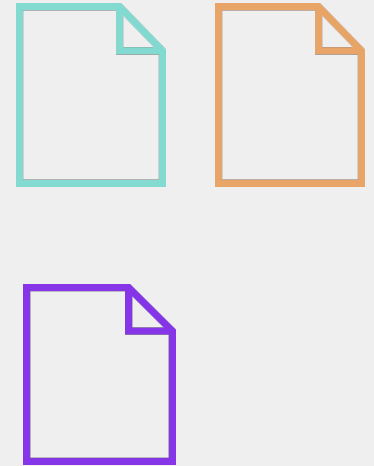
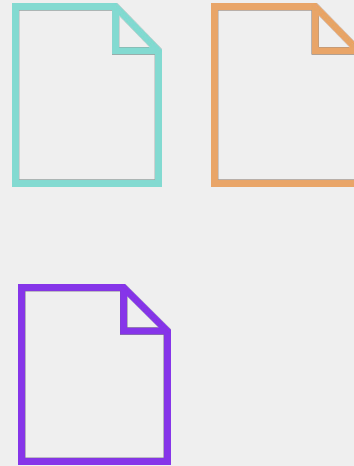
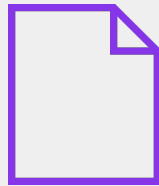
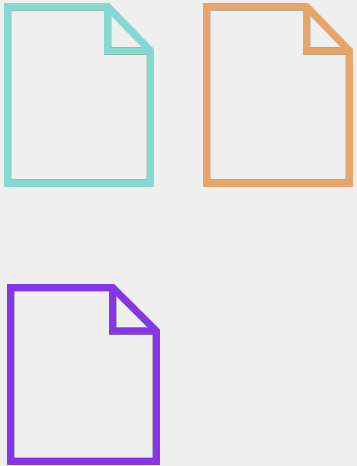
Subir un archivo

Working directory

Staging area

Repositorio local

Repositorio remoto



git push

Cambiar el stage

Son las 16:18. Las instrucciones dicen que no debo subir el archivo `VeryHeavyFile.txt` que pesa 100 MB.

Hice `git add *.txt` y solo me queda un minuto para poder subir la actividad.



I'm having a panic attack.

```
git reset HEAD file_name
```

Ya hice commit



```
git reset HEAD~1
```

Sitios útiles

- www.git-scm.com
- [Una guía de estilo de *commits*](#)
- [Una metaguía de Git](#)

Jupyter Notebook



*“The Jupyter Notebook is a web application that allows you to create and share documents that contain **live code**, equations, visualizations and **explanatory text**. ”* (<http://jupyter.org/>)

Instrucciones para instalar: <http://jupyter.org/install.html>

Google

+



stackoverflow

¿Cómo buscar soluciones?

python [versión] [librería] [duda]



¡En inglés!

¿Cómo imprimir una cola con Python? **X**

python 3.6 collections print queue



¿Cómo buscar soluciones?

python [versión] [error]



¡En inglés!

NameError: name “MiVariable” is not defined



NameError: name * is not defined





python3.5 NameError: name * is not defined



[Todos](#) [Videos](#) [Maps](#) [Imágenes](#) [Noticias](#) [Más](#) [Preferencias](#) [Herramientas](#)

Genera de 95,800 resultados (0.50 segundos)

[In Python3.5:NameError: name 'image_to_string' is not defined](#)

[https://stackoverflow.com/.../in-python3-5nameerror-name-image...](https://stackoverflow.com/.../in-python3-5nameerror-name-image-...) ▼ Traducir esta página

11 jun. 2017 - Please post your source code so we can look over the code and get more details. Also your error is caused by a variable declaration without a ...

[oop - Python3 NameError: name 'method' is not defined - Stack Overflow](#)

[https://stackoverflow.com/.../python3-nameerror-name-method-is...](https://stackoverflow.com/.../python3-nameerror-name-method-is-...) ▼ Traducir esta página

18 mar. 2016 - consider you have the function **defined** in the global scope: def recursive(x): if (x>5): print (x) recursive(x - 1). you would simply call this with ...

[input\(\) error - NameError: name '...' is not defined - Stack Overflow](#)

<https://stackoverflow.com/.../input-error-nameerror-name-is-not-...> ▼ Traducir esta página

14 ene. 2014 - input_variable = input ("Enter your name: ") print ("your name is" + input_variable) ...
input ("Enter your name: ") File "<string>", line 1, in <module> **NameError: name 'dude' is not defined**
... I did what Kevin said and it is version 2.7.5! ... If you are using **Python 3.x**, raw_input has been renamed to input .

[python NameError: name 'file' is not defined in python 3.5 - Stack ...](#)

<https://stackoverflow.com/.../python-nameerror-name-file-is-not-...> ▼ Traducir esta página

26 nov. 2015 - Traceback (most recent call last): File "c:\python3.5\lib\runpy.py", line python 3.x from
is Q: python **NameError: name 'file' is not defined** But ...

[python 3.x - NameError: name 'value' is not defined - Stack Overflow](#)

<https://stackoverflow.com/.../nameerror-name-value-is-not-defined-...> ▼ Traducir esta página

3 abr. 2014 - **NameError: name 'value' is not defined** ... A variable defined in a function isn't visible outside the function. ... answered Apr 5 '14 at 2:39.

[NameError: global name 'unicode' is not defined - in Python 3 - Stack ...](#)

<https://stackoverflow.com/.../nameerror-global-name-unicode-is-...> ▼ Traducir esta página

9 nov. 2013 - **Python 3** renamed the unicode type to str , the old str type has been replaced by bytes . if
isinstance(unicode or str, str): text = unicode or str ...

Otras recomendaciones



- Empezar las tareas cuando entreguen el enunciado
- Buscar más en Google
- Estudiar el material de clases
- Ir a las ayudantías
- Estudiar el ramo en serio desde el principio
- Ser estratégico con las tareas
- Dedicarle tiempo a otros ramos
- Dormir

Referencias

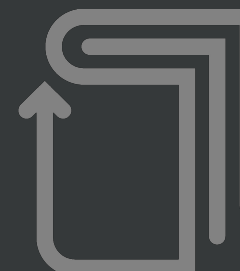
- Pichara K., Pieringer C., Advanced Computer Programming in Python, <https://advancedpythonprogramming.github.io/>
- www.git-scm.com



Información para Estudiantes



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA



Estimación de Carga Académica (ECA) 2018-1

¿Qué es ECA?

ECA (Estimación de la Carga Académica) tiene por objetivo que los estudiantes entreguen información respecto de los tiempos dedicados en las diferentes actividades académicas consideradas en sus cursos.

¿Para Qué?

Permitirá saber cuánto **tiempo** dedicas a los cursos, y así se podrá generar información que permita retroalimentar a los cursos respecto a su nivel de exigencia.

Invitación

Para tener éxito en lo anterior, te solicitamos que una vez a la semana accedas al sitio web o aplicación celular para entregarnos los tiempos que dedicaste a los diferentes cursos. Es fundamental que registres cada semana durante todo el semestre. Si por algún motivo no pudiste ingresar en una semana, puedes seguir participando de esta encuesta durante el semestre.

Metodología

1

Los días domingos en la noche te llegará el link a tu mail UC para acceder al sitio web o aplicación celular.

2

En la encuesta podrás indicar los tiempos que dedicaste esa semana al curso o los cursos que te están preguntando.

3

La encuesta de cada semana estará abierta desde el domingo hasta el día miércoles.

Estimación de Carga Académica (ECA) 2018-1

¿Cómo lo haremos?

1. Recibes el domingo en la noche un mail con la dirección

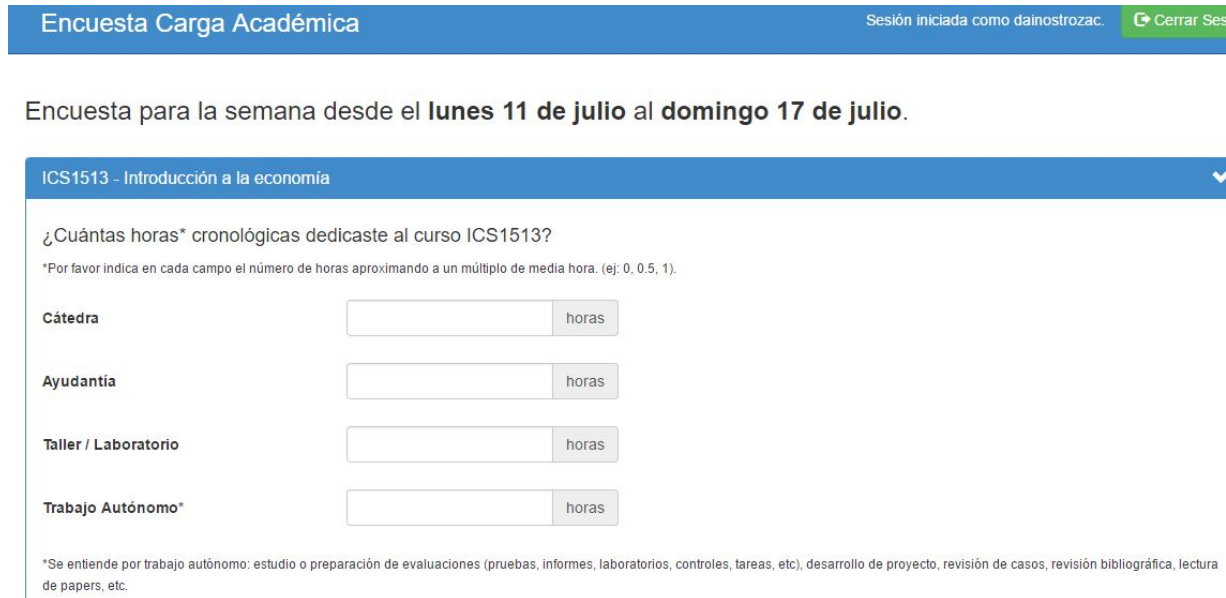
<http://eca.ing.uc.cl>

2. Para ingresar debes entrar con tus datos de autenticación UC.



The screenshot shows the authentication page of the Pontificia Universidad Católica de Chile. At the top, there is a blue header with the text 'Ayuda' and a question mark icon. Below the header, the university's logo and name are displayed on the left, and the word 'Autenticación' is on the right. The main area contains a login form with a text input field for the username, a password input field labeled 'Clave UC', a 'Limpiar' button, and an 'Iniciar Sesión' button. At the bottom, there are links for 'Cambio de Clave UC' and 'Recuperar Clave UC'.

3. Una vez que ingreses, podrás visualizar tus cursos, la encuesta y responder cada semana.



The screenshot shows the 'Encuesta Carga Académica' interface. At the top, there is a blue header with the text 'Encuesta Carga Académica' and a green button labeled 'Cerrar Sesión'. Below the header, there is a message: 'Encuesta para la semana desde el **lunes 11 de julio** al **domingo 17 de julio**.' The main area is titled 'ICS1513 - Introducción a la economía' and contains a question: '¿Cuántas horas* cronológicas dedicaste al curso ICS1513?'. Below the question, there is a note: '*Por favor indica en cada campo el número de horas aproximando a un múltiplo de media hora. (ej: 0, 0.5, 1)'. The form has four rows, each with a label and a text input field followed by a 'horas' button: 'Cátedra', 'Ayudantía', 'Taller / Laboratorio', and 'Trabajo Autónomo*'. At the bottom, there is a note: '*Se entiende por trabajo autónomo: estudio o preparación de evaluaciones (pruebas, informes, laboratorios, controles, tareas, etc), desarrollo de proyecto, revisión de casos, revisión bibliográfica, lectura de papers, etc.'



ingenieríauc

Sistema de Créditos Transferibles (SCT)
Medición carga académica
de los estudiantes



ESCUELA DE INGENIERÍA
FACULTAD DE INGENIERÍA

ingenieríauc

Actividad

1. En el *Syllabus*, vayan a la carpeta “Actividades” y descarguen el enunciado de la actividad 0 (AC 00).
<https://github.com/IIC2233/Syllabus>
2. Trabajen en la actividad hasta las 16:20