



Introducción

Un profesional de la Actuaría, debe estar constantemente actualizado y en conocimiento suficiente de las herramientas de tecnología de la información (TI) disponibles en su campo de trabajo. En nuestro estudio abordaremos primeramente un repaso básico de conceptos esenciales que nos abrirán el panorama y fungirán como la columna vertebral de todo el curso.

Programación Estructurada

Iniciemos con el paradigma de programación más elemental. Para ello nos valdremos del siguiente:

Teorema (de Böhm y Jacopini): *Un programa propio puede ser escrito utilizando únicamente 3 estructuras de control: secuenciales, de selección y de repetición.*

Un programa se dice propio si cumple las siguientes condiciones:

- i) Se tiene solo un punto de entrada y un punto de salida
- ii) Todas las sentencias del algoritmo son alcanzables (existe al menos un camino que conecta el principio y el fin)
- iii) No posee ciclos infinitos

Revisemos a detalle cada una de las estructuras de control básicas:

Estructura secuencial

Indica que cada instrucción del programa será ejecutada una detrás de la otra en el orden en que aparecen en el código.

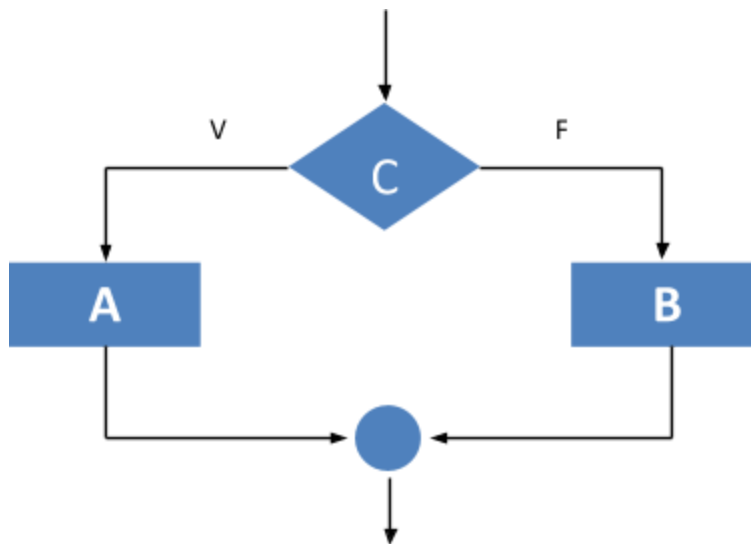


A y *B* pueden representar desde una instrucción simple hasta un programa completo.



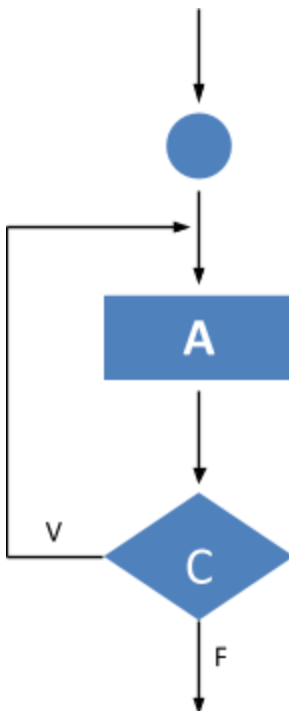
Estructura de selección

Permite elegir entre dos alternativas como resultado de la evaluación de una condición (verdadero-falso). Es la clásica instrucción IF incluida en todos los lenguajes de programación.



Estructura repetitiva

Ejecuta una instrucción repetidamente mientras se cumpla una condición dada.





Con esto en mente, realicemos un sencillo algoritmo para ejemplificar los conceptos con ayuda del lenguaje de programación Python. Python es un poderoso lenguaje de programación multipropósito que se ha abierto paso en el mundo hasta convertirse en el más utilizado de acuerdo al IEEE. Fue creado en 1991 por Guido Van Rossum. El lenguaje es multiplataforma y está disponible para sistemas MacOS, Linux y Windows, pertenece a la comunidad open source y no tiene costo.

Para poder desarrollar código en python, requerimos dos cosas: el intérprete y un editor de código.

Para principiantes, se recomienda la instalación de la distribución **Anaconda** que contiene una gran variedad de paquetes preinstalados para hacer todo tipo de tareas además del intérprete y editor de código en Notebooks.

Puede descargarse (versión 2.7 por favor) aquí: <https://www.continuum.io/downloads>

En cuanto a un editor de código tenemos Pycharm <https://www.jetbrains.com/pycharm/download/> una excelente opción que incrementa nuestra productividad al escribir código.

Una herramienta adicional que hará nuestro trabajo más fácil además de ser una tendencia a nivel mundial será el uso de **Git**, el cual es muy popular en el mundo del desarrollo de software. Como actuarios, git no será utilizado a profundidad, sin embargo, será una herramienta imprescindible para compartir código, prácticas, tener todos las mismas versiones y evitar compartir archivos por correo electrónico u otros medios de manera innecesaria. Git es un software de control de versionado escrito por Linus Torvalds (el creador del kernel de linux), para poder instalarlo, podemos seguir las instrucciones del sitio <https://git-scm.com/download/>. Se recomienda encarecidamente el **no usar Windows** ya que actualmente, prácticamente cualquier herramienta dentro del mismo puede ser sustituida en otro sistema operativo. Una manera más cómoda de gestión de repositorios git es GitHub(<https://github.com/>), puede ser vista como una “red social de código” hay que crear un perfil ahí para poder acceder a una cantidad inmensa de recursos públicos de código y otras cuestiones científicas. El repositorio que tendrá todo el contenido de nuestro curso está en la dirección https://github.com/JGFuentesC/tsc_2017.

Ahora mostraremos el proceso para clonar el repositorio, conforme avancemos en el curso, dicho repositorio será actualizado periódicamente y será cuestión únicamente de decirle a nuestro repositorio local que se sincronice con el repositorio remoto en la nube. Entramos al repositorio y vamos a la sección *clone or download*:



Repository de código para la asignatura de Temas Selectos de Computación Actuaría FES Acatlán UNAM

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/JGFuentesC/tsc_2017`
Download ZIP

tsc_2017

Repositorio de código para la asignatura de Temas Selectos de Computación Actuaría FES Acatlán UNAM

Copiamos la dirección que se nos presenta, abrimos una terminal en nuestro sistema operativo y cambiamos al directorio donde queramos ubicar nuestro repositorio:

Documentos/python/2018-1/tsc/

```
jose@jose-Galaxy-TabPro-S ~/Documentos/python/2018-1/tsc
Archivo Editar Ver Buscar Terminal Ayuda
jose@jose-Galaxy-TabPro-S ~ $ cd Documentos/python/2018-1/tsc/
jose@jose-Galaxy-TabPro-S ~/Documentos/python/2018-1/tsc $
```



Una vez ahí, escribimos en la terminal:

```
git clone https://github.com/JGFuentesC/tsc_2017.git
```

y listo, tendremos clonado el repositorio de la nube en nuestro equipo local:

```
jose@jose-Galaxy-TabPro-S ~/Documentos/python/2018-1/tsc $ git clone https://git
hub.com/JGFuentesC/tsc_2017.git
Cloning into 'tsc_2017'...
remote: Counting objects: 7, done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 7 (delta 0), reused 4 (delta 0), pack-reused 0
Unpacking objects: 100% (7/7), done.
Checking connectivity... done.
jose@jose-Galaxy-TabPro-S ~/Documentos/python/2018-1/tsc $
```

Ahora, procedemos a abrir pycharm para poder editar y ejecutar los archivos del repositorio.

Para este primer ejemplo, tenemos un pequeño programa que calcula el factorial de un entero.



```
tsc_2017 - [-/Documentos/python/2018-1/tsc/tsc_2017] - .../ej_01_factorial.py - PyCharm Community Edition 2017.1.2
File Edit View Navigate Code Refactor Run Tools VCS Window Help
tsc_2017 ej_01_factorial.py ej_01_factorial
Project tsc_2017 -/Documentos/python/2018-1/tsc/tsc_2017 README.md ej_01_factorial.py
ej_01_factorial.py
1 def fact():
2     f = 1
3     f = 1
4     n = 5
5     if n < 0:
6         print "Error"
7     elif n == 0 or n == 1:
8         print "1"
9     elif n >= 2:
10        while i <= n:
11            f *= i
12            i += 1
13            print "id" % f
14
15
16 if __name__ == "__main__":
17     fact()
18
Run ej_01_factorial
/usr/bin/python2.7 /home/jose/Documentos/python/2018-1/tsc/tsc_2017/ej_01_factorial.py
128
Process finished with exit code 0
Platform and Plugin Updates: PyCharm Community Edition is ready to update. (today 01:02 PM) 18:1 n/a UTF-8 Git: master
```

Se muestra el resultado de la ejecución del programa, el programa ha sido escrito de acuerdo al teorema de estructura, se deja al lector la investigación de la aplicación en código de estos conceptos.