

# Taller de Herramientas Computacionales

Jorge S. Martínez Villafan

Enero 25, 2019



# Índice general

<b>1. Clase uno</b>	<b>5</b>
1.1. Bitacora de la clase número uno . . . . .	5
<b>2. Clase dos</b>	<b>7</b>
2.1. Bitacora de la clase número dos . . . . .	7
<b>3. Clase tres</b>	<b>9</b>
3.1. Bitacora de la clase número tres . . . . .	9
<b>4. Clase cuatro</b>	<b>11</b>
4.1. Bitacora de la clase número cuatro . . . . .	11
<b>5. Clase cinco</b>	<b>13</b>
5.1. Bitacora de la clase número cinco . . . . .	13
<b>6. Clase seis</b>	<b>15</b>
6.1. Bitacora de la clase número seis . . . . .	15
<b>7. Clase siete</b>	<b>17</b>
7.1. Bitacora de la clase número siete . . . . .	17
<b>8. Clase ocho</b>	<b>19</b>
8.1. Bitacora de la clase número ocho . . . . .	19
<b>9. Clase nueve</b>	<b>21</b>
9.1. Bitacora de la clase número nueve . . . . .	21
<b>10. Clase diez</b>	<b>23</b>
10.1. Bitacora de la clase número diez . . . . .	23
<b>11. Clase once</b>	<b>25</b>
11.1. Bitacora de la clase número once . . . . .	25
<b>12. Clase doce</b>	<b>27</b>
12.1. Bitacora de la clase número doce . . . . .	27

<b>13. Clase trece</b>	<b>29</b>
13.1. Bitacora de la clase número trece . . . . .	29
<b>14. Clase catorce</b>	<b>31</b>
14.1. Bitacora de la clase número catorce . . . . .	31
<b>15. Clase número seis</b>	<b>33</b>
<b>16. Clase número siete</b>	<b>35</b>
<b>17. Clase número ocho</b>	<b>37</b>
<b>18. Clase número nueve</b>	<b>39</b>
<b>19. Clase número diez</b>	<b>41</b>
<b>20. Clase número once</b>	<b>43</b>
<b>21. Clase número doce</b>	<b>45</b>
<b>22. Clase número trece</b>	<b>47</b>
<b>23. Clase número catorce</b>	<b>49</b>

## Introducción

El libro es para fortalecer el conocimiento de la materia taller de herramientas computacionales.

# Capítulo 1

## Clase uno

### 1.1. Bitacora de la clase número uno

**Lo que hice en esta primera clase del curso fue:**

1. El curso conta de lo siguiente:

- a) Hacer un bitacora diariamente de lo visto en clase, exponer sus dudas
- b) Quince asistencias totales durante las 3 semanas que dura el curso
- c) Exámenes al finalizar cada semana, es decir, los viernes
- d) Al final del curso haremos una investigación a cerca del tema que nos parezca conveniente o del agrado del alumno, en el cual se aplique lo visto durante las quince clases del curso. Se deberá hacer una presentación a cerca del tema elegido

Para el curso el profesor nos recomendó un libro que podemos encontrar en el siguiente enlace: [www.bibliotecas.unam.mx](http://www.bibliotecas.unam.mx)

La clase prosiguió y se nombraron algunas sistemas operativos entre ellos: Windows, Linux (el que se usará para el curso), iOS También se nombraron distintos lenguajes de programación: python, C, C++, java entre otros. El profesor nos dio información básica acerca de la computación en general como del lenguaje de programación Por ejemplo las distintas versiones de linux que son: Ubuntu, Fedora, Debian y Slack. También nos explicó que un bit es un uno o un cero, ya que así, en sistema binario es como se representa la información dentro de una computadora En Linux existen tres tipos de permisos: Usuario (rwx), Grupo (rw-) y Todos (r-) También nos dio algunos comandos para ejecutar en la terminal

- 1) touch /tmp/algo
- 2) ls -l /tmp/algo
- 3) chmod 751 /tmp/algo

4) /tmp/algo

Tanto en windows como en Linux existen las variantes de entorno conocidas como PATH Algunos comandos para utilizar el path son "set" que sirve para ver la variante de entorno y "pwd" para saber en qué dirección se encuentra. Finalizamos la clase hablando un poco acerca de python.

# Capítulo 2

## Clase dos

### 2.1. Bitacora de la clase número dos

En el segundo día del curso iniciamos probando los comando que vimos en el día anterior, pudimos saber la utilidad de varios comandos como:

1. `touch /tmp/algo` crea un archivo vacio
2. `ls -l /tmp/algo` arroja información sobre permisos
3. `chmod 751 /tmp/algo` cambia los permisos de usuarios, grupos y todos
4. `/top` muestra información sobre la computadora como CPU y su actividad
5. `cd/` muestra el directorio de raíz

La clase prosiguió con una explicación tanto del profesor como el de la ayudante Karla sobre qué era Git Hub: Github es un servidor que permite bajar, subir, compartir e intercambiar información de la nube, bastante utilizado por los programadores.

Todos los alumnos nos registramos en la página de github utilizando nuestros correos de ciencias y nuestro numero de clave: utilizando los siguientes comandos en la terminal de Linux

- a) `git config --global user.email .ejemplo@ciencias.unam.mx`
- b) `git config --global user.name "numero de cuenta"`

Si queremos bajar nuestra información a una computadora nueva, lo único que debemos hacer, una sola vez, es poner el comando "git clone", pegar el link de nuestro repositorio de github y así se descargarán nuestros archivos a la computadora deseada

Algunos otros comandos del github que vimos durante la clase del martes pasado fueron

- a)* `cd "nombre de la carpeta deseada"` para abrir una carpeta
- b)* `ls` para visualizar lo que hay en la carpeta anteriormente abierta
- c)* `git add *` para agregar nuevos archivos al repositorio de github
- d)* `git commit` para finalizar el agregado

Para instalar github en fedora, se deben escribir lo siguiente en la terminal:

- a)* `git init`
- b)* `sudo apt-get upgrade`
- c)* `sudo apt-get install git`

Para subir nuevos archivos a github Se debe abrir las carpetas donde se tiene los archivos deseados. Escribir en la terminal:

- a)* `git add *`
- b)* `git commit`
- c)* `git push`
- d)* usuario
- e)* contraseña



# Capítulo 3

## Clase tres

### 3.1. Bitacora de la clase número tres

**Al inicio de la tercer clase del curso se empezó a hablar de los servidores y sus clientes, por ejemplo Faceebook o el mismo github**

Se hablaron sobre las distintas terminaciones de algunos archivos, por ejemplo los archivos generados en Latex deben ir terminados con un .tex y los archivos generados en Python con un .py

La clase prosiguió haciendo, basicamente, un repaso de lo visto en la clase numero dos. Hicimos ejercicios similares a los de la clase pásada. El profesor nos enseñó a hacer carpetas utilizando la terminas, a su vez, estas carpetas servirían para guardar los archivos que hicieramos durante el curso. Los pasos para hacer dichas carpetas son las siguientes:

1. escribir en la termina `mkdir - p directorio/ directorio` esto crea un directorio para todas, es decir un directorio padre
2. nombrar las carpetas que queremos crear
3. abrir las carpetas creadas con `cd "nombre de la carpeta"` por si queremos crear otra carpeta dentro de la abierta

Estas carpetas fueron subidas al github, para saltarnos el paso de escribir el comentario, basta con poner `git commit -m comentario`. Para salir de la pantalla del comentario se presiona la tecla `.esc`, seguido por `shit` más dos puntos y finalizamos con `"wq"` si queremos guardar y salir o `"q!"` si queremos no guardar y salir.

Posteriormente vimos lo que era el `"vi"` que crea un documento y creamos un readme, con el comando `readme.md`

La clase finalizó con el profesor pidiendonos llevar un problema de fisica para la clase próxima.



## Capítulo 4

# Clase cuatro

### 4.1. Bitacora de la clase número cuatro

La cuarta clase del curso la iniciamos trabajando con un problema en que el profesor llevo, similar al que nos había encargado la clase pasada, en él nos pedían determinar la posición de un objeto si sabemos la formula para calcularla. El profesor desglosó cada asunto que se pudiera presentar en el problema, desde si podía ser negativa, como su función. La analizó desde un punto matemático como desde un objeto para darle explicaciones a la computadora de lo que debe hacer. El profesor hizo un gran énfasis en esto último ya que la computadora entiende exactamente lo que se le dice, es por ello que debemos ser claros y explicar detalladamente lo que queremos que haga

Las personas que no utilizaban las máquinas de la escuela tuvieron que instalar python. Fedora ya viene con python pero para poder abrirlo se deben escribir en la terminal el siguiente comando: `$] dnf install python tools`

Una vez con python todos, pudimos ver la versión de python escribiendo el comando `$] python --version`

Posteriormente abrimos python para meter la formula del problema del profesor, python se abre con el comando `!python` en la terminal. Cuando abrimos python prosedimos a meter los datos que teníamos, hubo un pequeño problema con el resultado debido a que python nos arrojaba divisiones enteras, para arreglar este problema solo se debe agregar un `.0` después del divisor o dividendo. Durante la resolución del ejercicio el profesor nos dio varios tips acerca de como escribir en python por ejemplo para agregar un comentario éste debe ser precedido por el signo gato (`#`), o para escribir en español correctamente, es decir, que python reconozca caracteres que en el inglés no existen, como la diéresis o las tildes, se debe agregar al inicio el siguiente código: `#!/usr/bin/env python3`. También nos enseñó que para correr un código en python debe ir precedido por el comando `python`. La clase prosiguió con el profesor dándonos instrucciones sobre como realizar una cadena en python

1. Comilla simple sirve para crear una cadena de texto de una línea
2. Comilla doble sirve para crea una cadena de texto de una línea
3. Comilla triple sirve para crea una cadena de texto multiple
4. `\n`: para salto de línea

NOTA: una cadena es un conjunto de caracteres en python que están delimitados por una comilla o comilla doble

La clase finalizó realizando la tarea en el salón, con el problema que nos encargó en la clase numero tres debíamos generar un código para resolverlo de una manera similar al problema que el profesor resolvió durante la clase. Una vez hecho se subió como es acostumbrado a github.

# Capítulo 5

## Clase cinco

### 5.1. Bitacora de la clase número cinco

En la clase numero cinco vimos varios comandos de python, que básicamente la explicación de ellos podría resumir gran parte de la clase, los comando fueron los siguientes:

1. `$| print` que sirve para mostrar lo que pedimos, el profesor suele explicarlo, como "piensa en un numero y dime cuál es"
2. `$| %` sirve para desplegar una variable a través de print. Se combina con varias letras (E,f,g), para desplegar un valor en distintos formatos
3. `$| %g` mostrar la variable en el formato numerico más corto posible
4. `$| %E` muestra el formato en notación científica
5. `$| % .f` muestra un valor con flotante con dos de su decimales (puede ser cualquier valor decimal)
6. `$| %10.f` muestra un numero flotante con dos decimales recorrido 10 espacios a la derecha
7. `$| %f` despliega el valor en formato flotante
8. `$| %s` si tengo una variable que contenga una cadena y quiero mostrar su contenido con print se utiliza este comando
9. `$| math.sqrt` para sacar raíces cuadradas
10. `$| def` para definir una función
11. `$| return` indica que regrese un valor

Continuamos la clase con un archivo que copiamos de la pantalla del profesor con ayuda del tigre, bastaba con solo dar clic medio con el raton para poder pegar lo que el profesor subrayó. A dicho archivo traducimos los comandos que venían en él.

La clase prosiguió y fue turno de Karla, quien nos enseñó los primeros pasos para iniciar en Latex Nos pidió abrir TeXstudio y nos empezó a decir los comandos y para qué sirve cada uno

1. `documentclass[letterpaper, 12pt, oneside]` Para iniciar y darle el formato de texto
2. `usepackageamsmath` analogo al `import` en python
3. `usepackagegraphicx` para las graficas
4. `usepackagecolor` para poder utilizar letras de colores
5. `usepackage[utf8]inputenc` para poder escribir caracteres que no existen en el inglés como la diéresis o las tildes
6. `usepackageenumitem` para poder enumerar
7. `%` para poder añadir un comentario, es analogo al `#` en python

Y eso fue lo último que hicimos en la primer semana del curso

# Capítulo 6

## Clase seis

### 6.1. Bitacora de la clase número seis

La cuarta clase del curso inició con un problema, trata de encontrar la raíz cuadrada de un, el problema radicaba en encontrar la raíz de equis, si tenemos dos figuras geometricas de área "X", la primera era un cuadrado con los lados raíz de equis y la segunda un rectangulo de altura equis y base 1. Todo el grupo dio diversas sugerencias para poder encontrar el valor de equis, finalmente el profe dio el metodo para encontrar el valor de dicha incognita. Haciendo un gran énfasis que en las computadoras es difícil encontrar el valor exacto, pero podemos encontrar una aproximación precisa que nos sirva (en realidad creo que esto sucede en las computadora y en los humanos ya que nunca sabemos el valor de un numero racional raíz de dos o pi, son algunos ejemplos en donde no se utilizan los valores exactos, solo aproximaciones).

La clase prosiguió con un problema en python, donde definimos una función llamada "vAbsoluto", es decir, el valor absoluto, este ejercicio sirvió como ejemplo para poder utilizar dos comandos muy utiles en python el

1. if
2. else
3. return

Estos comandos son codiciones:

El comando if, sirve para dar una sentencia, ejemplo: "Si sucede tal cosa.<sup>en</sup>tonces (print) pasa esto

El comando else, es para arrojar otro resultado, si el comando if resulta ser falso Tanto if como else tiene el mismo valor, es por ello que ambos deben ir a la misma altura del código y en el mismo bloque. Es importante resaltar que se puede utilizar un if, sin un else, pero es imposible colocar un else si un if, ya que else es una especie de negación del if

El comando return sirva para que nos devuelva el valor que hemos asignando

previamente También vimos el comando `import` que sirve para mostrar lo que deseemos en el `bash`

Después de haber explicado el uso de `if` y `else` el profesor nos puso un ejercicio, donde se tenía una diana infinita y al arrojar los dardos nos arrojaba un valor dependiendo del las condernas tanto en `x` como en `y` que hubiesemos sacado La clase finalizó con el profesor dejandonos una tarea del mismo problema de la diana pero con más condiciones, si en el centro de la diana hubiera un circulo de radio 10 y fuera de éste el valor que nos arrojara fuera 100, pero estando delimitado por las otras coordenadas.



# Capítulo 7

## Clase siete

### 7.1. Bitacora de la clase número siete

La clase numero siete del curso del Taller de Herramientas Computacionales fue el día martes. Esta clase inició con ejemplos de como utilizar estructuras de repetición, en este caso se utilizó el comando "while", este comando se encarga de formar un ciclo, primero evalua si la condición dada fue cierta el ciclo se repetirá hasta que le digamos que pare.

Los comando if y else que vimos una clase pasada son muy utiles a la hora de utilizar while, se le puede dar una condición con if para que el ciclo se repita si el if fue cierto (aunque realmente también se podría si fue falso), y poner un else para, por ejemplo detener el ciclo.

Es muy importante utilizar los bloques y espacios correctamente en el uso del while. Las condiciones en un bloque while deben ir en las líneas inferiores y estar con cuatro espacios de distancias del inicio de documento. Tanto if como else deben ir dentro del bloque while si éstas son codicones que queremos que el código corra con while. If y else deben estar a la misma altura, pero ambas deben estar en distinta altura de While si pertenecen a este comando.

El profesor hizo énfasis en que al momento de dar un asignación, (ejemplo:  $a=b+c$ ), los espacios no son importantes pero se recomienda ponerlos para poder facilitar la lectura tanto del programador como la del lector

La clase prosiguió y fue turno de Karla quién nos enseñó varias cosas que podemos hacer en TeXstudio.

Iniciamos tratando de poner una imagen, pero lamentablemente no se pudo, debido a un problema con la dirección donde se encontraban la imagen que queríamos, aunque algunos pudieron solucionararlo con poner una tilde y cambiar de lugar la ubicación de la imagen

Dejando de lado lo de la imagen, nos enfocamos en cómo colocar expresiones matemáticas en TeXstudio:

1.  $\$ \$$  sirven para indicar que escribiremos expresiones, matemáticas, existen

las "diagonalCorchete" que son analogas al signo de moneda

2. `$x_2$` para poner subindices, en este caso, es equis subindice dos
3. `$x^2$` sirven para poner exponentes, en este caso equis al cuadrado
4. `$`  
`frac22$` sirven para poner fracciones, en este caso, dos partidos dos
5. `beginmatrix` sirve para poner matrices, los terminos de éstas deben ir con un `&` entre ellos para poder diferenciarlos
6. `dots` puntos suspensivos
7. `vdots` puntos suspensivos verticales
8. `sqrt` para poder escribir raices (similar a python)
9. `$`  
`int_a\hat{x}^2 dx`, para poder escribir integrales, en este caso, la integral de a a b, de equis cuadrada

La clase finalizó

# Capítulo 8

## Clase ocho

### 8.1. Bitacora de la clase número ocho

En la clase del miercoles, es decir la clase numero siete del curso iniciamos conociendo varios comandos de la terminal que nos sirven para abrir python en segundo plano, esto nos permite seguir escribiendo en la terminal donde ejecutamos el idle para abrir python. Algunos comandos que vimos fueron los siguientes:

1. `ctrl c`: termina el programa
2. `kill -9` mata el programa que se esta ejecutando
3. `chmod +x "nombre de un archivo"` para darle los permisos, es decir, leer, escribir, ejecutar etc
4. `find . -name "*.es"` nos muestra todos los archivos .py que tengamos en la carpeta en que estemos ubicados
5. `where is python` nos muestra el lugar donde se encuentra python

Proseguimos la clase hablando de objetos y metodos con las siguientes definiciones:

Un objeto es un conjunto de datos y funciones relacionadas que posee características. Un objeto puede ser un árbol, como característica tiene hojas

Metodo es una acción que realiza el objeto, continuando con el ejemplo del árbol, este realiza la acción de alimentarse mediante el método de la fotosíntesis.

Clase: una clase es un conjunto para la creación de objetos de datos según un modelo predefinido.

Continuamos la clase trabajando con python, el profesor nos indicó que debemos iniciar los trabajos en python de la siguiente manera:

```
#!/usr/bin/python2.7
# -*- coding: utf-8 -*-
```

”

Jorge S Martínez Villafan, 418003839

Herramientas computacionales

Lo que nos explico el miercoles

”

Obviamente este es mi caso, los demás deberan llenarlo con sus datos. Continuamos la clase trabajando con `import math sqrt` para poder sacar la raíz cuadrada de un numero, en ese caso era la del número 16. Hicimos un ejercicio llamado Ulam, que se trata sobre si es par dividir en dos, si es impar multiplicarlo por 3 y añadirle uno.

Una vez terminamos de trabajar con Python, nuevamente fue turno de Karla quién nos habló un poco sobre TeXstudio, pero el tiempo no nos dejó avanzar mucho. La próxima clase la iniciaría ella.

# Capítulo 9

## Clase nueve

### 9.1. Bitacora de la clase número nueve

La clase número nueve, por primera vez fue iniciada por la ayudante Karla. Iniciamos aprendiendo a poner tablas en TeXstudio, hicimos una tabla en la cual iba seccionada con algunos caracteres como las barras del valor absoluto encerrado a una C, para crear una columna, o escribiendo hline para poner una línea horizontal Continuamos descubriendo nuevos paquetes del TeXstudio, por ejemplo:

1. documentclassbook, sirve para darle formato de libro al archivo que estamos haciendo
2. usepackage[spanish]babel, sirve para que nos lance en español las secciones que el libro crea
3. usepackagebiblatex para crear la bibliografía
4. usepackagehyperref nos permite poder hacer un enlace a un link deseado

Algunos otros comandos que conocimos de Latex en aquella clase fueron:

1. section\*Introducción con el uso de astedisco hace aparecer un número de conteo
2. url, sirve para poder crear un enlace a un link, similar al "hyperref"
3. chapter, sirve para poder crear nuevos capítulos dentro del libro
4. beginverbatim, este tiene un uso muy importante, nos permite leer un código de python (desconozco si funciona para otros lenguajes de programación), tal y como fue escrito, sin que TeXstudio nos marque error por caracteres propios del programa
5. input, sirve para poder abrir un documento dentro de TeXstudio

Una vez terminamos la parte de LaTeX, fue turno del profesor, quién nos preguntó cómo nos fue con la tarea de los diez problemas que dejó la clase pasada, a la mayoría le fue mal, así que el profesor empezó a resolver el primer problema, el del máximo común divisor por el método de euclides.

Primero planteó el problema en el pizarrón y con ayuda del grupo fue dándole forma a lo que sería el código.

El código tan solo utilizaba una definición, un while y un return.

En el archivo que lleva la S, nos enseñó que solo bastaba con importar la def utilizada en el código, y con ayuda de input podíamos hacer interactivo el programa

Nos pidió terminar los diez problemas que habían quedado de tarea. La clase finalizó

# Capítulo 10

## Clase diez

### 10.1. Bitacora de la clase número diez

Empezamos la clase número diez resolviendo dudas acerca de la tarea que fue dejada en días pasadas. Utilizamos nuevos comandos en el sheel:

1. += sirve para hacer una suma y arroje el resultado
2. != sirve como "diferente de"
3. \*= sirve para hacer una multiplicación y arroje el resultado
4. /= sirve para hacer una división y arroje el resultado
5. bool sirve para saber que contiene una cadena, da falso si es vacia, da true si tiene algo. También se puede hacer una asignación previa, bool de la asignación dará truesi si el valor de la asignación es distinto de 0, de lo contrario dará false
6. L.append para agregar elementos a un lista, es del tipo objeto metodo
7. L.append /(()) para meter una lista dentro de otra
8. L[X] Para saber el elemento numero del indice x
9. len(l) Para saber cuantos elementos tiene una lista
10. L[X].append Se puede agregar más valores a la lista dentro de otra lista
11. L.insert s para agregar algo antes del indice que queramos, el indice es la posición de un elemento de la lista
12. r=L.pop() se saca el ultimo elemento de la lista
13. r=L.pop(x) elimina el elemento del indice x
14. L.extend es para agregar varios elementos a la lista

Continuamos la clase y el profesor nos pidió hacer un ejercicio con while, bool, para poder vaciar una lista.

También quedó claro que un índice se empieza a correr desde 0,1,2... y una posición corre desde 1,2,... por lo tanto el número de índice siempre es el número de posición menos uno

El profesor nos pidió agregar a los diez problemas algo de las listas visto hoy. Empezamos a trabajar con el ejercicio de convertir los grados celsius en fahrenheit y viceversa. Hicimos el mismo ejercicio pero en lugar de while se usó for. Y después se hizo interactivo, para meter los valores, input y suplirlo por la lista. Posteriormente empezamos a trabajar con el método range(X) crea una lista con los índices de tamaño x. Se puede dar un intervalo con range x, es un intervalo abierto como en matemáticas.

Ejemplo: `for C in range(-20,45,5): gradosC.append(C)`: crea una lista desde menos 20 a 40 de 5 en 5.

De tarea quedó definir la función range f.

La segunda semana del curso finalizó.



# Capítulo 11

## Clase once

### 11.1. Bitacora de la clase número once

En la clase numero once del curso iniciamos preguntandole dudas al profesor de los 10 problemas que debía resolverse con listas. Posteriormente el profesor nos enseñó a sacar promedios utilizando las listas, teníamos que recorrer la lista con el metodo `for i in "nombre"`, después dimos una asignación de a la letra r para que en ella se asignara lo que deseabamos. Y volvimos a utilizar `for in` para hacer que el valor fuera recorriendo la lista y la asignación r cabiaría su a ella misma más el valor.

Para que el resultado de la división nos diera un decimal, de ser el caso que así fuera utilizamos el comando `Float`, dividido por `len` que es la longitud de la lista.

Hicimos una archivo llamado `gitignore` para poder evitar que se suabar archivos no deseados como:

1. `.aux`
2. `.pdf`
3. `.syntec.gz`
4. `.pyc`

Continuamos trabajando por python y nos enseñó el comando `for in` que debemos usar una i porque generalmente usamos esta letra para contar.

Nos explicó cuales son las sumas superiores e inferiores para las integrales y nos pidió calcular la área de  $2x+5x^2-6x^3+12$  de  $[a,b]$ .

`range(len)`, significa que len está anidado a range y primero se ejecuta len, después range

No pidió que de tarea checáramos para que sirven los demás L. y hacer un ejemplito con ellos

`Enumerate` da el índice y el valor de la lista en ese índice

Los corchetes sirven para acceder a un elemento de una lista o definir una lista

Una `tupla` es como una lista donde no se pueden cambiar los elementos

Para finalizar nos pidió hacer una lista con varias listas dentro con los valores de fahrenheit en centígrados

# Capítulo 12

## Clase doce

### 12.1. Bitacora de la clase número doce

La clase numero doce del curso se inicio hablando de las listas, de algunos comandos que ya habíamos visto pero también de otras funciones que no habíamos visto previamente. Trabajamos con un archivo de python en donde debíamos utilizar varias listas, con el comando `pprint` le dimos una distinta forma a la manera en cómo se mostraban las listas en el shell al correr el código. Posteriormente empezamos a utilizar nuevos comandos para las listas:

1. `L[X:]` Muestra los valores a partir del índice x
2. `L[X:Y]` Muestra los valores desde el índice X hasta el índice Y
3. `L[:Y]` Muestra los valores anteriores al índice Y
4. `L[1,-1]` Muestra todos los valores excepto el primero y el último
5. `L[-Y]` Las listas de cierto modo tienen una forma circular, con este comando el programa contaría hasta el índice -Y y mostraría su valor
6. `L1=L[:]` Creará una nueva lista independiente
7. `for in range` Recorre la lista por sus índices
8. `for in "lista"` Recorre la lista por sus valores
9. `[i][j]` pide el j-ésimo elemento dentro de la i-ésima lista

Continuamos haciendo un ejercicio en donde se debían crear varias listas dentro de una lista. En el ejercicio nos pedía que hicieramos una lista dentro de ella, el semestre, el nombre del alumno, sus materias, su promedio, las asistencias, tarea y exámenes. Aprendimos que una coma al final omite el salto de línea

Empezamos a trabajar con TeXstudio y Karla nos enseñó a hacer diapositivas tipo "powerpoint" dentro del programa, utilizamos los siguientes paquetes y comandos para hacer la diapositiva

1. `documentclassbeamer` para poder trabajar con diapositivas
2. `usethemenombre` sirven para poder utilizar temas predeterminados dentro del programa (la mayoría son azules y similares)
3. `beginframe` para poder crear un nuevo cuadro de diapositiva
4. `frametitle` para titular la diapositiva previamente creada
5. `trans` para cambiar el estilo en que se pasa de un cuadro de diapositiva a otro

## Capítulo 13

# Clase trece

### 13.1. Bitacora de la clase número trece

Iniciamos la clase leyendo los problemas de la pagina 58 del libro python facil, el profesor nos pidió que leyeramos los problemas y ver si lo podriamos resolverlos o no.

Hubo problemas con el ejercicio de las matrices porque no lo hemos visto.

También el profesor en general le pidió ayuda con el ejercicio del laberinto que también utilizaba matrices. Con ayuda de todo el grupo el profesor nos explicó como se debe resolver con 4 pasos. Desde situarse en la entrada, pedirle que avance, hacia el frente, si puede, sino dar ordenes que explore por otros caminos como arriba, abajo, o atras, con preguntas como: ¿Ya llegué a la salida? y ¿puedo avanzar?.

Se discutió acerca de si puede existir una lista infinita en una computadora, la mayoría estuvo de acuerdo de que no porque en la computadora los reales no existen, y los recursos de una computadora son finitos, pero se podría hacer un programa que arroje datos hasta que se le dé la indicación de parar con el comando `ctrl + c`. En el libro python fácil venía un ejercicio en el cual se realiza una "lista infinita", la copiamos y tratamos de correr, algunos tuvieron problemas pero otros pocos pudieron correrla realizando algunas modificaciones.

Los 8 problemas del libro python facil quedaron de tarea

El profesor empezó a resolver el problema de fibonacci de manera recursiva, no sin antes enseñarnos lo qué es una formula recursiva, es decir, una formula que está definida en terminos de ella misma.

También se resolvió la suma de n de manera recursiva

Le pregunté acerca del juego "Los postes de hanoi" el profesor explicó en qué consiste el juego: Se tienen 3 postes un con discos de distintos tamaños, ordenados desde el mayor a menor tamaño, utilizando uno de los postes vacios como auxiliar, se deben pasar los discos de un poste a otro, pero no deben quedar un disco mayor encima de un disco menor.

Dio una idea de cómo se puede resolver utilizando python, los postes serán listas, los discos serán números uno mayor que otro. Los discos de número 0 equivaldrán a que no hay discos en la torre de hanoi.

Nos pidieron hacer que una lista muestre el primer elemento primero y después el resto mediante una forma recursiva, muchos tuvimos problemas al generar el código, el profesor terminó dando la respuesta y al salón le quedó claro

Solo existen los variables en ámbitos de validez las variables locales y variables globales

Las variables locales cuando la variable pertenece a una función

Las variables globales todas las funciones responden a la variable

Continuamos la clase y el profesor nos mostró en el tigre una página que sirve para poder correr un código en python y otros lenguajes de programación y poder ver en tiempo real cómo el programa se va ejecutando

La página es la siguiente: url <http://www.pythontutor.com/visualize.html>

La clase finalizó

## Capítulo 14

# Clase catorce

### 14.1. Bitacora de la clase número catorce

Empezamos la clase numero catorce resolviendo dudas de la tarea de los ejercicios de la clase

Le pedimos ayuda al profesor para resolver el problema del laberinto y explicó nuevamente cómo debía de hacerse, y con los pasos que siempre se utilizaban en el curso, desde analizar, ejemplificar, codificar etc.

De la pizarra pasamos a la computadora y nos fue guiando como es que debe de resolverse:

El codigo fue el siguiente:

```
def resolver(L,e):
    print e
    n = len(L[0])
    m = len(L)
    x = e[0]
    y = e[1]
    if y==n-1 or x == m-1:
        return e[0]+1,e[1]+1
    else:
        if L[x][y+1] == False:
            e = [x, y+1]
            return resolver(L,e)
        elif L[x+1][y] == False:
            e=[x+1,y]
            return resolver(L,e)
        else:
            print ("ya no puede avanzar mas")
```

```

L= [[True, True, True, True],
     [False, False, False, True],
     [True, True, False, True]]
e=[1,0]
r=resolver(L,e)
import numpy as np
print(np.matrix(L))

```

Con algunas condiciones más dentro del laberinto el profesor nos pidió que tratáramos de generar el código para poder resolver laberintos más complejos

Empezamos a trabajar con una problema sobre las cadenas de ADN y generamos el siguiente código

```

def contar_v1(adn, base):
    adn=list(adn)
    i= 0
    for c in adn:
        if c == base:
            i += 1
    return i

```

```

def contar_v2(adn, base):
    adn=list(adn)
    i= 0
    for c in adn:
        if c == base:
            i += 1
    return i

```

```

def contar_v3(adn, base):
    i= 0
    for j in range(len(adn)):
        if adn[j] == base:
            i += 1
    return i

```

```

def contar_v4(adn, base):
    adn=list(adn)
    i = 0
    j = 0
    while j < len(adn):
        if adn[j] == base:
            i += 1
        j +=1
    return i

```



```
adn="ATGCGACCTAT"
base="C"
print contar_v1(adn, base)
print contar_v2(adn, base)

n= contar_v2(adn, base)
print n
print "%s aparece %d en %s" % (base, n, adn)
print "{base} aparece {n} veces en {adn}" .format(base=base, n=n, adn=adn)
print contar_v2(adn, base)
print contar_v3(adn, base)
print contar_v4(adn, base)
```

Este código nos sirvió para que nos explicara lo que hace el comando `sum` que sobre una lista bool solo nos devuelve los valores que son "True"

La penultima clase del curso finalizó.



# Bibliografía

Autor Jorge Salvador Martínez Villafan  
*Cualquier cosa* 2019