Taller de Herramientas Computaionales

Brenda Paola García Rivas

17.01.19


CIENCIAS

Contents

Introducción

Este libro consiste en hacer un resumen del curso que se nos impartió en tres semanas de la materia de Taller de Herramientas Computacionales; esto con el fin de repasar nuestras clases, y así tener un resumen del curso completo.

Chapter 1

Inicio del curso

En la primer clase se inició con una introducción de cómo sería el curso de intensivo, se dijeron una serie de puntos que se seguirán durante estas tres semanas, partiendo de esto se dió inicio a la clase formal; en esta última se vieron una variedad de puntos como:

1. Sistemas operativos, algunos de los sistemas operativos son:
   (a) Linux, el más mencionado y aclamado por matemáticos, computólogos y en general la Facultad de Ciencias encabeza esta lista de sistemas operativos, este ultimo consiste en un software de codigo abierto, el cuál nos permite interactuar con el ordenador de forma que podemos editarlo. Mientras se introdujo este tema,también se hablo de algunos sistemas operativos que son distribuidos por Linux, algunos de éstos son:
      i. Ubuntu
      ii. Fedora
      iii. Debian
   (b) Windows, me parece que es uno de los sistemas operativos más utilizados, puede funcionar en muchos otros dispositivos electrónicos que hagan uso de microprocesadores. Es diferente de Linux, ya que no es un software libre que pueda editarse.
   (c) iO's. Es un sistema operativo manejado por la empresa Apple
2. El segundo punto que se comentó en clase fue el de los lenguajes de programación, para este se explicó que un lenguaje de programacion sirve para órdenes o instrucciones a una computadora para producir datos y existen muchos como:
   (a) Python
   (b) Java
   (c) C++, entre otros

3. El tercer punto importante del que se habló fue los registros de los archivos. Algunos de ellos son:

   (a) .exe, Es bueno saber que este archivo no se puede ejecutar o archivar en cualquier lado

   (b) .txt

   (c) .py

   También se vieron otros más comandos para una computadora, algunos de éstos son:

   i. top. Este comando carga información sobre la computadora, CPU. En cuanto se usa este comando aparecerá la actividad general que tiene la computadora en tiempo real. Cabe mencionar que para salir de este modo, únicamente se necesita presionar "a".

   ii. cd/. Se usa para buscar un directorio desde la raíz. Después de la diagonal se pueden usar otros comandos como:

      A. lib64. Qu se encarga de buscar las bibliotecas de 64 bits

      B. lib. Busca en las bibliotecas

      C. home. Se usa para usuarios

      D. media. Se usa para la USB o CD

      E. mnt. Busca los discos externos y discos duros

      F. . Se vuelve el directorio anterior

      G. .. Vuelve al directorio anteanterior

      ls "nombre del directorio". Muestra el contenido del directorio.

   iii. df -lh. Muestra que parte del disco duro se puede utilizar

   iv. set. Muestra todas las actividades del entorno

   v. set I less. Pagina, ya que "set" sólo los muetra como lista

   vi. less "archivo". Muestra el contenido de algún archivo paginado

   vii. file "archivo". Dspeja los archivos de otro archivo o de un directorio

Chapter 2

Github

2.1 ¿Cómo se usa GitHub

Además se aprendió cómo usar Github a partir de una terminal, para esto se usan los siguientes comandos:

1. Primero, se tiene que instar colocando los comandos en el siguiente órden:
   (a) git int
   (b) sudo apt-get upgrade
   (c) sudo apt-get install git (nota: en fedora en lugar de get es dnf)
2. Para cargar la cuenta de github a cualquier máquia se usan los siguenetes comandos en este orden(Para este paso ya se debe tener una cuenta en la página)
   (a) git config –global user.name "aquí se pone el nombre del usuario"
   (b) git config –global user.email "aquí se pone el email"
   (c) cd "Nombre del directorio donde se quiere clonar la información"
   (d) git clone "url del repositorio"
   (e) cd "nombre del repositorio"
   (f) git add * (este añade los últimos archivos modificados)
   (g) git commit (En este momento se tiene que hacer un comantario que aparecerá en el repositorio) Despues de escribir el comentario se debe hacer lo siguiente para procegir con la actualización de los documentos:
      i. Se escribe el comentario
      ii. Se pulsa la tecla: esc
      iii. Se colocan las letas "wq"
      iv. Se da enter

(h) git push (sube los cambios del documento)

(i) "User name"

(j) "password"

(k) git status (se ve el estado del árbol, este último paso no es necesario)

Nota: si únicamente se desea actualizar los datos, es decir, no clonar los archivos se omite el paso 4 y se sustituye por un: git status

Chapter 3

Python

3.1 ¿Cómo usar python?

En este día se inició la clase con un nuevo problema, el profesor nos pregunto acerca de la caída libre de una pelota, al principio se me hizo un poco extraño este inicio de la sesión pues no lograba captar bien que tenía que ver con la programación, pero después de un rato se vió claro. Nos explicó que si nosotros tenemos cualquier problema hay seis pasos importantes a seguir:

1. Definir el problema.
2. Que quede claro el problema
3. Encontrar una ecuación que modele la situación
4. Conocer los elementos que contiene el problema
5. Restringirlo
6. Definirlo

Partiendo de este procedimiento, se puede visualizar el problema con mayor facilidad.Es importante tener claro todos estos pasos ya que la computadora hace exactamente lo que nosotros pedimos, por lo que debemos ser claros, precisos y detallados al hacer cualquier tipo de programa. Por ejemplo,en el ejercicio de la pelota lo primero que hicimos fue entender que se nos decía, luego obtuvimos una fórmula y checamos que se conocieran todos los aspectos de ésta, despues delimitamos el mismo problema, evitamos los negativos, ya que no nos servían y unicamente dejamos la parte de interés.

Otra cosa que también se aprendió en clase fue para abrir python desde la terminal se usa el comando "idle", este último abre un shell en python. Se empezó a usar python despues de haber visto el problema de la pelota hecho en el pizarron. El código que se uso en un principio fue:

$$print\ 34 * 3 - 1/2 * 9.81 * 3 * *2$$

Sin embargo se descubrió que no daba el mismo resultado que salía al hacerlo en el pizarrón. Yo al principio llegué a creer que TODO EL GRUPO se había equivocado, así que hic nuevamente en los cálculos en mi cuaderno, para mi sorpresa, y según mis operaciones, la computadora estaba mal...¿eso podía ser verdad?.

Para resolver este dilema, separamos las operaciones hasta encontrar el error. Se descubrió que el error se encontraba en la división de 1/2. Pues bien, el profesor nos explicó que el resultado que python nos estaba imprimiendo era diferente porque sólo estaba contaba la divión entera; lo que hicimos para poder obtener el resultado correcto fue colocar un ".0" al denominador o al numerador de la división, esto para que python calculara la división flotante. Nos quedó así:

$$print\ 34 * 3 - 1.0/2 * 9.81 * 3 * *2$$

Fue justamente lo que le hizo falta, pues salió el mismo resultado que al hacerlo con calculadora. Después de esto, el profesor nos mostro una manera más sencilla de hacerlo. Primero colocamos los valores de cada variable y despues sólo ponemos la fórmula, quedando así:

```
v0 = 34
g = 9.81
t =5
y = v0*t - 1.0/2*g*t**2
print y
```

En la clase además se vió el tema de las cedenas en python (las cadenas son como notas para python), hay tres formas de iniciar una cadena, las formas son:

1. Usar tres comillas al inicio y al final sirve para las multicadenas. Ejemplo:

   """

   Hola, soy

   Brenda y estoy

   en la clase de T.H.C.

   """

2. usar el signo de gato:

   ```
   #Sólo puede usar un renglon
   #Si se necesitan más se usa otro de este mismo signo
   ```

3. Se usan " al inicio y al final del texto:

   $'Hola\ a\ todos'$

También se vieron nuevos comandos para usar números en python y facilitar nusestros cálculos. Algunos de ellos son:

1. %g. Este despliega un valor en su menor formato posible
2. %E, nos muestra el resultado en notacion científica
3. %f. Despliega el valor como flotante
4. %5.2f, despliega un número flotante con dos decimales y cinco espacios recorridos a la derecha (el cinco y el dos, pueden cambiarse al gusto)

Ya para las clases posteriores se retomó un ejercicio del día anterior, que consistía en un rectángulo de área x, con base=1 y altura x. El problema que se nos presentó era hacer que ese rectángulo tuviese sus lados de un tamaño muy similar, como convertirlo en cuadrado. Así mismo durante la clase anterior se nos explicó como obtener las fórmulas para usarlos en nuestro programa para que el día de hoy pudieramos usarlas. Se nos introdujo cómo y en qué momentos se debe hacer uso de "while" en python, y hacer uso de este último acompañado de un "return" el cuál ya se había visto en clases anteriores. Además de aprender eso, cómo contar el número de veces que se ejecuta una acción dentro de "while", para esto se utilizó "i". Para poder usarlo adecuadamente hay que tener en cuenta lo siguiente:

1. La numeración antes de usar el "while" debe ser 0, ya que aún no se ha contado nada. Entonces, antes de while, se colocará "i=0"
2. Ahora, hay que tener en cuenta que queremos contar cada ciclo después de que éste se ejecuta,entonces se tendrá que colocar debajo de las funciones de while el siguiente valor de i
3. Además debemos ver que cada vez que se ejecuta un ciclo, entonces se le va a sumar un ciclo a la función, por tanto se debe poner "i=i+1"

Además de esto, támbien se nos puso un reto en clase que debiamos hacer en equipo, esto se me hiso un método de enseñaza bastante divertido, porque nos dejaba pensar cómo resolver el problema.El problema consistía en identificar si un numero era par o impar dado que si es par, entonces x/2 es entero, y es impar si 3x*2 es entero; dado que x sea un entero. Ete ejercicio es llamado Ulam Se hace una funcion la cuálregrese un valor que depende de si es par o es impar, pero...¿cómo sabemos que es par? pues, si nosotros utilizamos la división entera y el resultado se multiplica por dos y da lo mismo que al principio, entonces es un número par, si al número le falta algo, entonces es impar. A partir de esta idea podemos ocuper un "if" que diga que si el resultado de x/2*2 es igual a 0 entonces se regresa x/2 que es par. Ahora, si no es par, entonces se usa un "else" para regresar 3x*2 (ya que es impar).

Despues de esto, debemos seguir calculando hasta que el último valor sea uno, es decir, debemos seguir calculando hasta que el valor sea igual a 1; para esto se necesita usar un "while". Mientras la x sea mayor o igual a 1, es igual a la función que se definió antes.

3.2 Programas hechos en clase

El primer programa consiste en tratar de convertir los grados C a grados F; debo decirlo que, cuando yo hice este problema lo que usé fue la fórmula para convertir de grados C a F qeu encontré en internet, utilicé un "imput" para que pudiera visualizarse una pregunta en python shell, y únicamente sólo coloqué la fórmula.

En su lugar, el profesor, nos mostró una manera más fácil y cool. Nos dijo: ¿Qué pasaría si nosotros queremos saber los grados de cinco en cinco, del -20 grados a los 30 grados?

```
 %Sirve para colocar
programas de python
C=-20  #Empieza en -20
iC=5 #Se coloca 5 porque se quiere hacer cada 5 grados
while C <= 40:
F = (9.0/5) * C +32
print C, F
C = C + iC #Esto es una asignación porque lo que está a la derecha se evalua y
lo que está a la izquierda se almacena
#C +=iC    Otra forma de escribirlo. Al valor almacendo de C, agrega el valor de iC
```

Después se inició con el problema de fibonacci, el cuál es un problema de recursividad; ayer se vió como resolverlo, sin embargo, esta vez el profesor nos explicó una forma más fácil de hacer los llamados para cada número. Este método implica guardar los resultados en una lista, en este se modificará que: el programa observe que esté un número en la lista para después guardarlo y posteriormente usarlo, pero ¿cómo se sabe que el décimo, onceavo, o el número que sea está en la lista?, pues fácil, teniendo en cuenta la longitud de la lista.

Además de esto, también se resolvieron algunas dudas de la tarea, se inició con el problema de la creación de un laberinto, para realizar éste, debemos tener en mente varias cosas:

1. Primero, se identificó en el pizarrón cómo se iba a hacer el laberinto, se intentó hacerlo lo más facil posible dado que sería únicamente un ejmplo. El laberinto tenía forma de 3x3, en dónde únicamente se podía pasar horizantalmente por la fila de enmedio.

2. Ahora bien, lo siguiente en cuestionarse es: ¿Puedo avanzar hacia enfrente?, si la respuesta es "sí", entonces, se debe mover una columna, y en la coordenada 'y' se le aumentaría un lugar.

3. Lo siguiente que hay que ver es que si se ya no se puede avanzar, entonces se regrese un mensaje que diga "Ya no es posible avanzar".

   El código queda así:

```
# -*- coding: utf-8 -*-
```

```
L=[[True, True, True],    #Se define el laberinto
[False, False, False],
[True, True, True]]
def resolver(L,e):
print (e)
n=len(L[0])  #Columna 1
x=e[0]
y=e[1]
if y==n-1: salida
return e[0]+1,e[1]+1  #Ya llegué
else: #Si no se ha llegado...
if L[x][y+1] == False: #Se puede mover una columna más, por eso se aumenta a y
e=[x,y+1]
return resolver(L,e)
else:
print 'Ya no se puede avanzar'
```

Lo siguiente que se hizo fue, complicar ligeramente el laberinto, esta vez sería de 3 filas y cuatro columnas; este nuevo laberinto tiene un bloqueo, si no se puede caminar hacia enfrente, entonces deberemos preguntar si hay otro camino disponible. El segundo camino que se eligió fu hacia abajo.

El programa con esta actualización quedó así:

```
# -*- coding: utf-8 -*-
L=[[True, True, True, True],
[False, False, False, True],  #Forma del laberinto
[True, True, False, True]]
def resolver(L,e):
print (e)
m=len(L)
n=len(L[0])  #Columna 1
x=e[0]
y=e[1]
if y==n-1 or x==m-1 : #casos de salida
return e[0]+1,e[1]+1  #Ya llegué
else: #Si no se ha llegado...
if L[x][y+1] == False: #Se puede mover una columna más, por eso se aumenta a y
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 se puede avanzar'

type(L)
e=[1,0]
```

```
r=resolver(L,e)
import numpy as np
print(np.matrix(L))
```

Además de esto, casi al final de la clase se alcanzó a ver un problema más; este último consistía en una cadena de ADN,lo que se intentaba hacer era un conteo de su contenido, se logró hacer de 4 formas distintas:

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

2. ```
   def contar_v2(adn,base):
   i=0
   for c in adn:
   if c == base:
   i += 1
   return i
   ```

3. ```
   def contar_v3(adn,base):
   i=0
   for j in range(len(adn)):
   if adn[j] == base:
   i += 1
   return i
   ```

4. ```
   def contar_v4(adn,base):
   i=0
   j=0
   while j < len(adn):
   if adn[j] == base:
   i += 1
   j +=1
   return i
   ```

Chapter 4

Latex

4.1 Usos de LaTeX

Usando Texstudio, primero se debe poner que tipo de documento se usará, pues en LaTex ya hay tipo planillas que le dan un formato al documento, después de colocar esto, se especificará que paquetes se usarán, ya que si no se colocan y después se quieren usar no se podrá. Existen paquetes como:

1. usepackagexcolor. Se usa para que se puedan cambiar las letras a otro color
2. usepackage[utf8]inputenc. Con este se validan los acentos
3. usepackageamsmath. Se usa para símbolos
4. usepackagegraphicx.
5. graphicspathaquí va la carpeta donde está la imágen/. Se usa para colocar imágenes

Después se coloca el título, autor y la fecha. Para iniciar el documento se usa la palabra "begin", depues se colocará la palabra "maketitle" que es lo que pondrá el título en su lugar junto con los demás datos proporcionados. Cabe añadir que si se desea colocar una imagen, entonces se tiene que poner el archivo donde se guardó en los usepackage, por ejemplo si la imahen está guardada en el archivo con nombre imagenes entonces se debe colocar "graphicspathImagenes/" y despues después de iniciar el documento se colocará "includegraphics[escala deseada]nombre de la imagen que se guardó en el archivo. Después de esto, para poder colocar una nueva página se usa el comando "newpage". He de añadir que se deben usar "äntes de cada comando, ya que si no se pone se leerá que hay un error.

Una vez teniendo en cuenta estas especificaciones, se puede iniciar la redacción del artículo.

También se aprendieron nuevos comandos para LaTeX. Casi al finalizar la clase se vieron algunos comandos para trabajar en Latex, debido al poco tiempo tuvimos que apresurarnos; se vió que:

1. Cómo se puede centrar texto en Latex. Esto se puede lograr fácilmente si colocamos un "begincenter

2. ¿Para qué sirve poner huge? Este sólo le da mejor aspecto a los tiítulos. Los hace más grandes.

3. ¿Cómo puedo hacer secciones? En esta clase también se nos mostró cómo hacer subsecciones; para esto, solo basta con poner diagonal sectionTítulo de la section, algo importante en este tema es que si despues de la palabra "section" colocas un "*" enumerará las secciones

Además se nos explicó cómo elaborar una tabla, acontinuación lo explicaré con mis palabras. Primero, si se desea, se puede abrir una sección llamada tablas, despues se incluye un "\begin{array}"lo siguiente que se hace es abrir otro par de corchetes para especificar cuantas columna habrá dividiendo la tabla, las separaciones se visualizarán con "—" y la letra "c" como si fuera el contenido de cada columna, por ejemplo, si se quieren colocar 3 columnas, entonces quedará: "—c—c—c—", si se quieren cuatro columnas sería "—c—c—c—c—". Lo que sigue de este paso es empezar a redactar lo que va a ir en cada columna, y para separar el contenido de una a otra columna se colocará un "&"

También se vió cómo hacer un libro en LaTeX, la verdad este tema me gustó bastante pues, me pareció dominarlo bastante bien :) Se inicia colocando el tipo de texto, anteriormente se ponía "article", sin embargo, ahora se colocará "book"; se usaran otras cosas como los usepakage (ya se han visto anteriormente). Además de esto, se al iniciar el documento se debe colocar la palabra "chapter" y delante de esta una diagonal, como es de costumbre, esto no sindicará que es un nuevo capítulo. Cada "chapter" viene seguido de su corchete para poder colocar el título. Si se desea, colocar un subcapítulo basta con usar "section" debajo del capítulo que se está viendo.

Además se aprendió como colocar una bibliografía y cabe añadir que se pudo ver cómo colocar un programa hecho en python; para este último caso se tuvo que usar "begin{verbatim}"