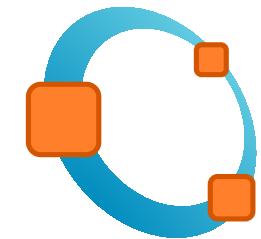




Capacitate en *Octave*



<https://introoctave.github.io/capacitoctave>



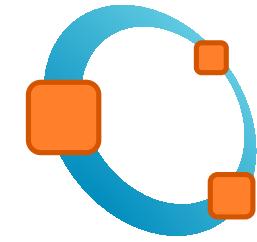
28 y 29 de noviembre 2019 - Facultad de Ciencias Aplicadas a la Industria

**3° JORNADAS DE ENSEÑANZA E INVESTIGACIÓN
DE LAS CIENCIAS EXPERIMENTALES**

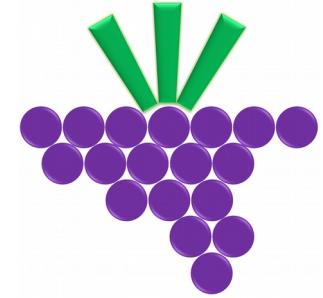
- 1. Introducción**
- 2. Octave como calculadora**
- 3. *Scripting* para cálculo avanzado**



Capacitate en *Octave*



Clase JEICE



**Daniel Millán, Nicolás Muzi,
Petronel Schoeman, Gabriel Rosa, Juan Cruz Luffi**

San Rafael, Argentina, 29 de Noviembre 2019



UNCUYO
UNIVERSIDAD
NACIONAL DE CUYO



FACULTAD DE
CIENCIAS APLICADAS
A LA INDUSTRIA



MoCCAi
MODELADO COMPUTACIONAL EN CIENCIAS APLICADAS E INGENIERÍA





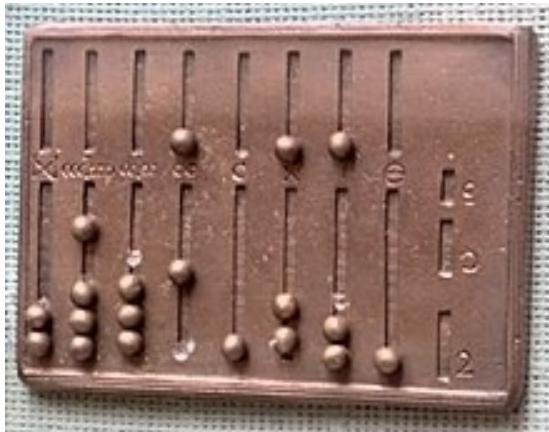
Introducción

En la antigua Mesopotamia, en la civilización Sumeria, tuvo su origen el **sistema sexagesimal**, es un sistema de numeración posicional que emplea como base aritmética el número **60**.

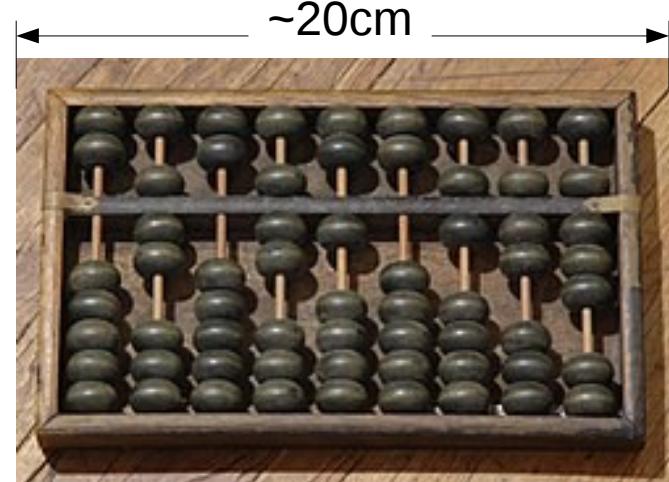
El sistema sexagesimal se usa para medir tiempos (horas, minutos y segundos) y ángulos (grados) principalmente.

Ábaco: permite realizar operaciones aritméticas sencillas, su origen se remonta a la antigua Mesopotamia, más de 2000 años antes de nuestra era.

[Wikipedia](#)



≤200 a.C.: Ábaco Romano.



≤200 a.C.: Ábaco Chino. El **suanpan** es un ábaco de origen chino.



Introducción

Mecanismo de Anticitera, la primer computadora mecánica de que se tenga conocimiento.

Sorprendente por su diseño y tamaño, un engranaje de relojería muy complejo y completo para su época.

[Wikipedia](#)



~100 a.C.: Reconstrucción del
Mecanismo de Anticitera, 2007.



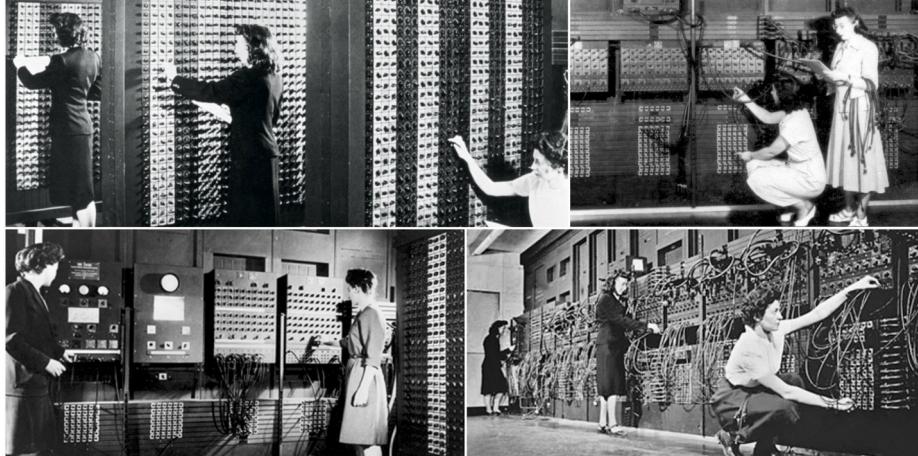
Introducción



1645: Blaise Pascal inventa la **pascalina**, una de las primeras calculadoras mecánicas. Funcionaba a base de ruedas de diez dientes, cada uno representaba un dígito del 0 al 9.

1949: La **EDVAC** fue la primer computadora de programas almacenados electrónicamente en forma binaria.

[Wikipedia](#).



1946: La **ENIAC** fue inicialmente diseñada para calcular tablas de tiro de artillería destinadas al Laboratorio de Investigación Balística de la Universidad de Pensilvania, para el ARMY USA.



John von Neumann



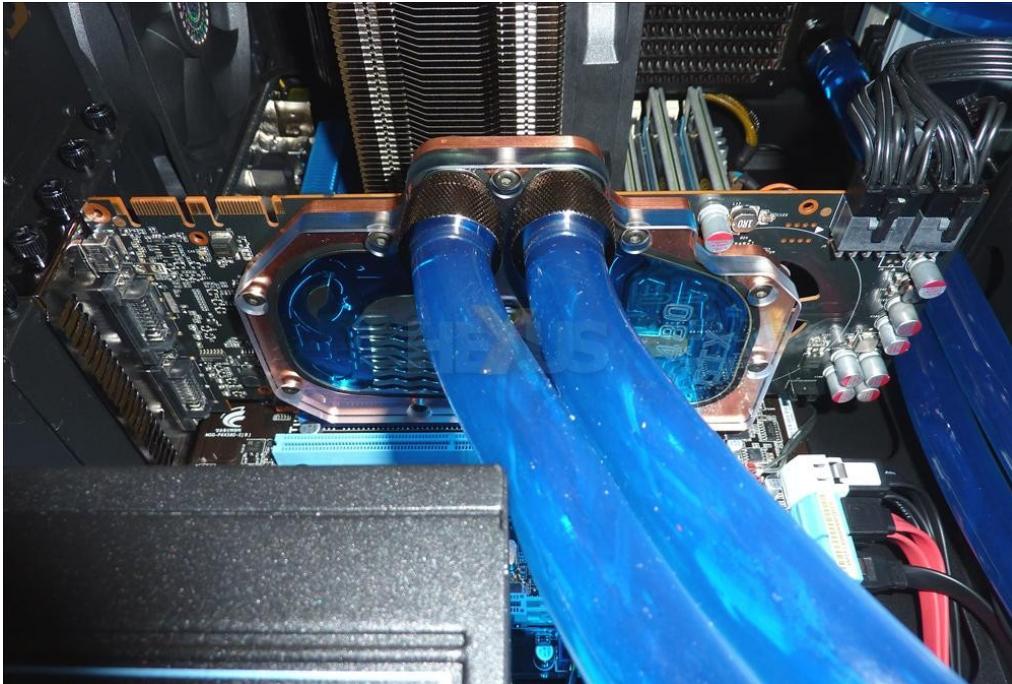
2010: Scan 3XS Cyclone PC

primer NVIDIA GeForce GTX 480 con refrigeración líquida

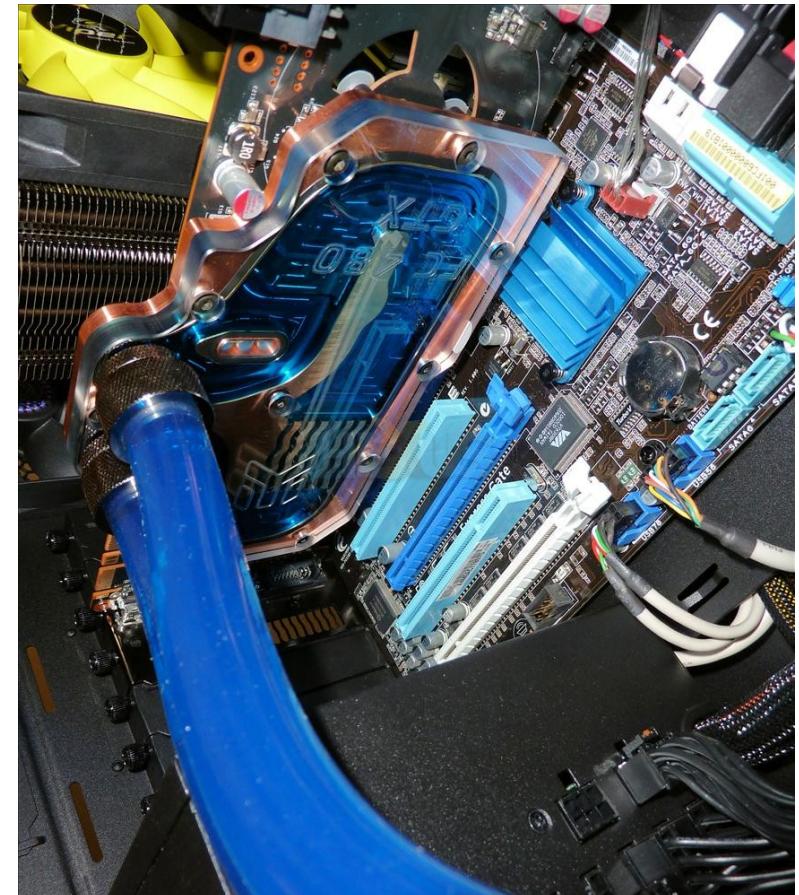


2010: Scan 3XS Cyclone PC

- primer tarjeta gráfica de NVIDIA con refrigeración líquida
- overclocked GeForce GTX 480, opera a 852MHz (701MHz)
- procesador i7 920, overclocked a 4GHz
- £1,646.84, incluyendo impuestos

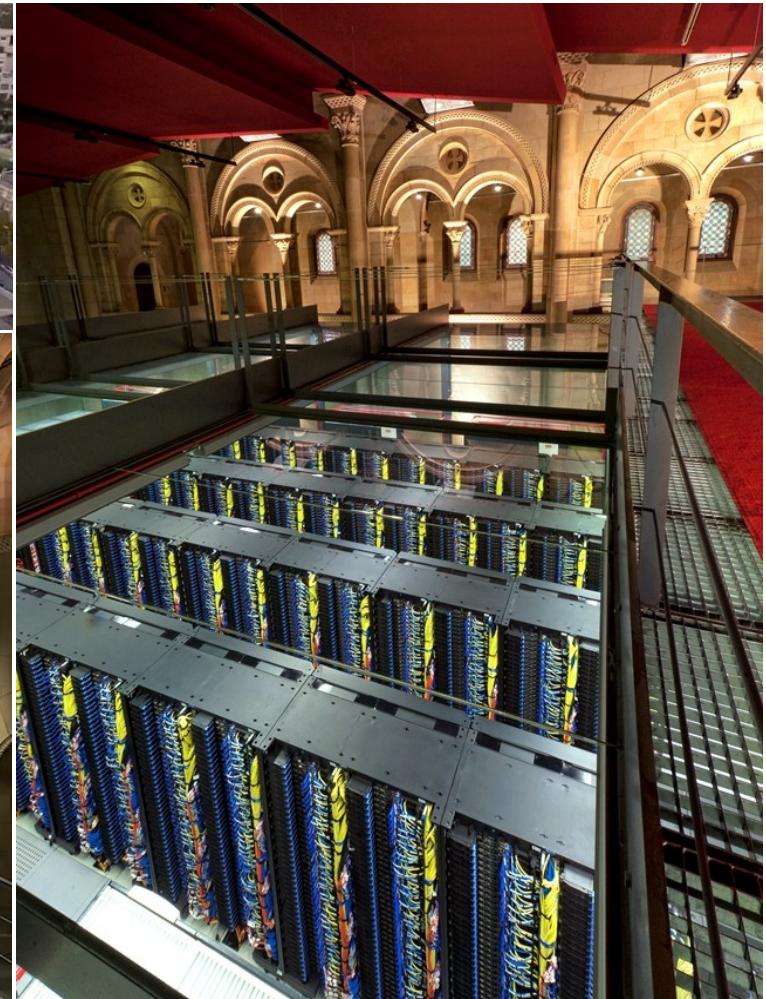


<http://hexus.net/tech/reviews/systems>



2019: MareNostrum Barcelona

- MareNostrum es el supercomputador más potente de España, el quinto más rápido de Europa y el 25º del mundo (nov - 2018).





2019: MareNostrum Barcelona

- **Composición atmosférica:** calidad del aire, aerosoles y como estos dispersan y absorben la radiación solar, ciudades inteligentes y la optimización del transporte y la salud humana.
- **Big Data:** herramientas visuales y algorítmicas para analizar y estudiar grandes volúmenes de datos.
- **Bioinformática:** integración, almacenamiento y transmisión de gran volumen de datos clínicos y datos de simulaciones, diseño de fármacos.
- **Biomecánica:** sistema cardiovascular y sistema respiratorio.
- **Predicción climática:** gestión de la agricultura y del agua, el pronóstico oceánico, estudio de los ciclones tropicales, estudio de dónde es más eficiente instalar un molino de viento.
- **Computación en la nube:** informática energética y optimización de los centros de datos.



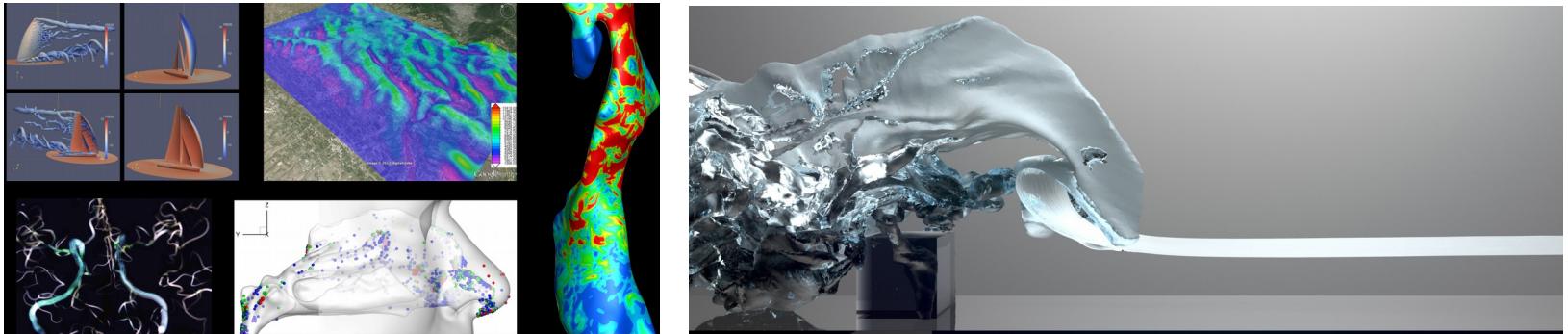
**Barcelona
Supercomputing
Center**
Centro Nacional de Supercomputación

<https://www.bsc.es/>

2019: MareNostrum Barcelona

10

- **Simulación de ingeniería:** reducción de las emisiones contaminantes, computación en mecánica de fluidos, mecánica no lineal de sólidos.



- **Geofísica:** terremotos, detección de la presencia de fluidos a grandes profundidades bajo la superficie de la Tierra, propiedades de la superficie de la Tierra.
- **Simulación social:** evolución cultural, eficiencia energética, seguridad pública de cara a tener ciudades inteligentes y resistentes.



2019: MareNostrum Barcelona

Alya Red

Proyecto Alya Red, su video promocional fue elegido mejor vídeo científico del 2012 por la National Science Foundation norteamericana y la revista Science. En él se explica cómo se crean los modelos con los que se simula el funcionamiento de un corazón, intentando imitar el comportamiento de los diferentes tejidos y de cómo las señales eléctricas viajan por su interior.

La tarea es tan compleja que para poder analizarlo con precisión se emplea el ordenador Mare Nostrum del Centro de Supercomputación de Barcelona.





1. Octave o GNU Octave

- **Octave o GNU Octave** es un programa y lenguaje de programación para realizar cálculos numéricos.
- Octave es parte del proyecto GNU (GNU no es Unix).
- Es considerado el equivalente libre de **MATLAB** (MATrix LABoratory).
- Ambos programas ofrecen un intérprete, permitiendo ejecutar órdenes en modo interactivo.
- Octave no es un sistema de álgebra computacional, como lo es Maxima o Mathematica, sino que está orientado al análisis numérico.
- El proyecto fue creado alrededor del año 1988 para ser utilizado en un curso de diseño de reactores químicos.
- El nombre es por *Octave Levenspiel*, profesor de uno de los autores y conocido por sus buenas aproximaciones, por medio de cálculos elementales, a problemas numéricos en ingeniería química.
- En el año **1994** apareció la versión 1.0 y en marzo de **2019** la 5.1.0. ¹²

1. Octave o GÑU Octave

The screenshot shows the Octave graphical user interface (GUI) with the title "Graphic User Interface (GUI) de Octave". The main window contains several panes:

- Explorador de archivos**: Shows the current directory as "IntroduccionOctave\2018_Electiva_FCAI\Folleto".
- Editor**: Displays the MATLAB script "sombrero_plot.m" which generates a sombrero plot. The code uses meshgrid and surf to create a surface plot of a sombrero function.
- Espacio de trabajo**: Shows variables like ans, h, r, tx, ty, tz, xx, and yy with their respective dimensions.
- Historial de comandos**: Shows command history including "help surf", "doc surf", and multiple entries for "sombrero_plot".
- Figura 1**: A 3D plot titled "Sombrero plot con Octave" showing a surface that rises to a peak at the origin and dips towards the edges.

2. El entorno de trabajo de Octave 4

Explorador de archivos

Barras de botones y herramientas

Directorio actual

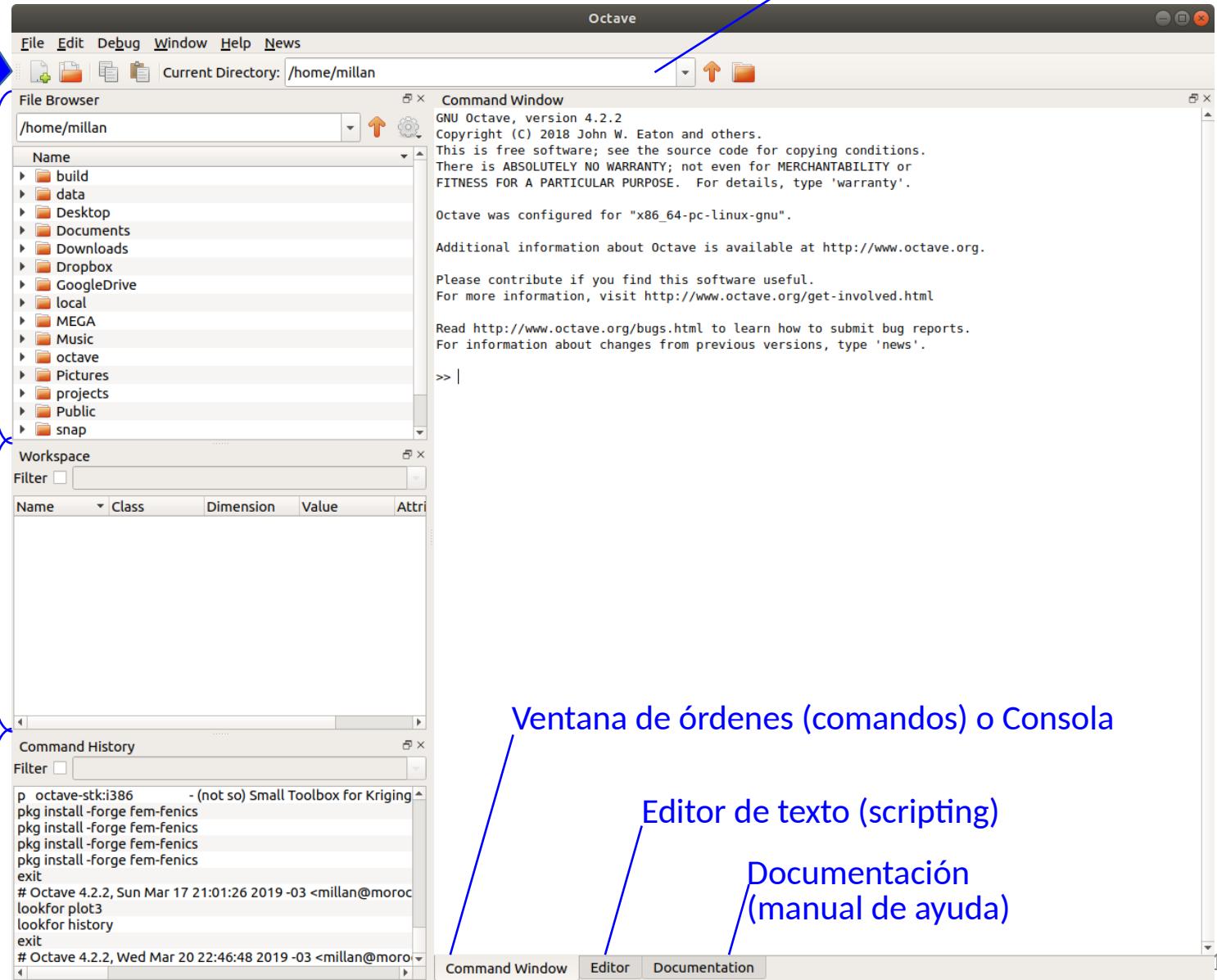
Espacio de trabajo

Historia de órdenes (comandos)

Ventana de órdenes (comandos) o Consola

Editor de texto (scripting)

Documentación (manual de ayuda)





2. El entorno de trabajo de Octave 4

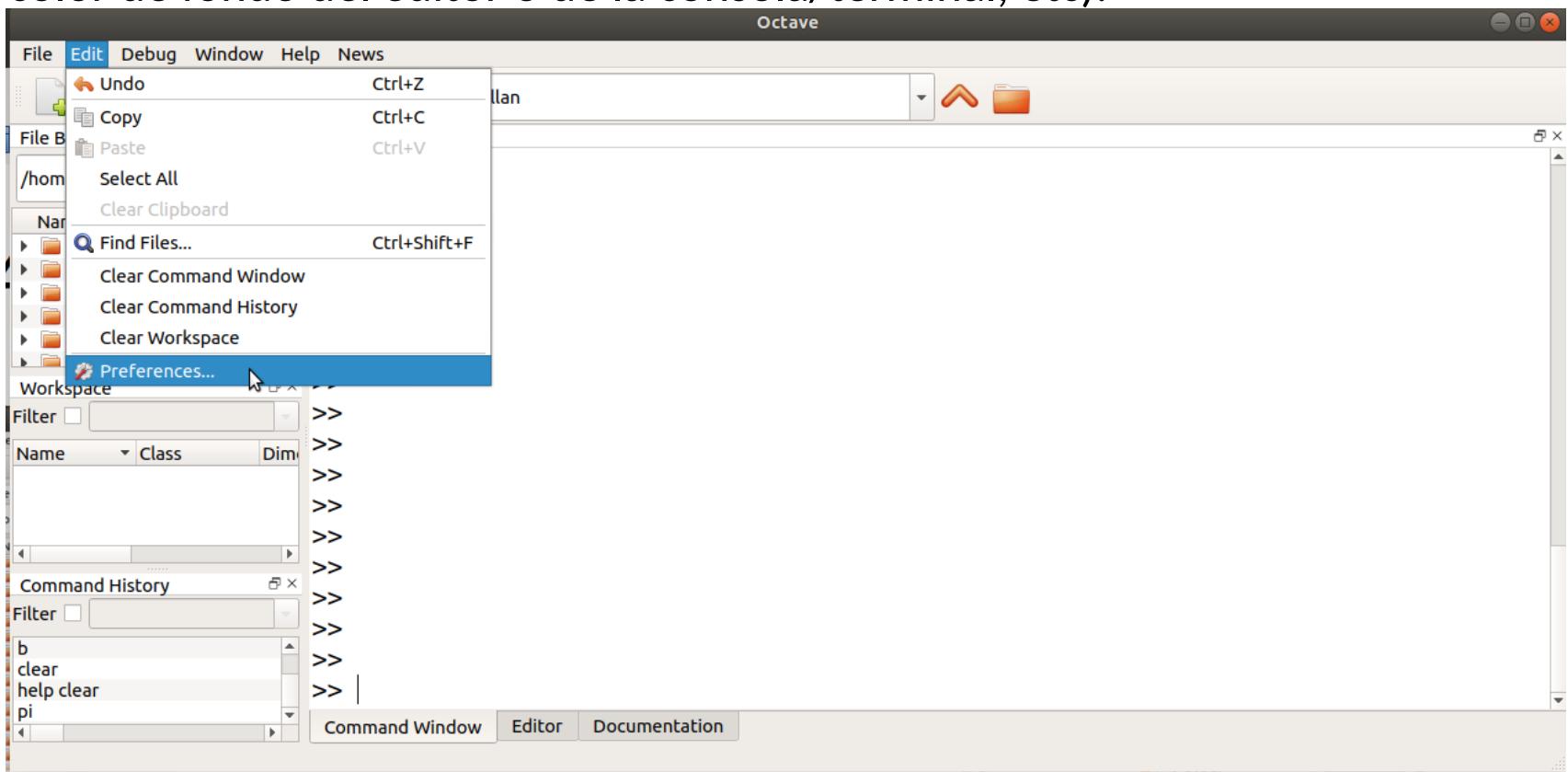
- El funcionamiento de la ventana de comandos es sencillo: se teclea una instrucción más la tecla **Enter**, luego Octave responde en la misma consola con el resultado de intentar realizar el comando tecleado. Si el comando es correcto nos devolverá el resultado, si no, nos devolverá un texto informando del error detectado por el intérprete de lenguaje de Octave.
- **Ejercicio:**

```
>> 4 + 3 * 12 + 60 / 12
>> a = 4 + 3 * 12;      b = 60 / 12;
>> a + b
>> a + b + c
```



4. Preferencias.

- En **barra del menú principal/Edit/Preferences** se dispone de un cuadro de diálogo desde el que se establecen casi todas las opciones que el usuario puede determinar por su cuenta (tipos, tamaño y color de letra, color de fondo del editor o de la consola/terminal, etc).





4. Preferencias.

Ejercicio: define que el idioma de Octave sea español.

Settings

General Editor Editor Styles Terminal File Browser Workspace Shortcuts Network

Interface

Language (requires restart) System setting ▾

Icon size Small Normal Large

Icon set for dock widgets Octave logo only Letter icons Graphic icons

Dock widget title bar Custom style Background inactive Active

3D 50 Text inactive Active

Show status bar

Confirm before exiting

Octave Startup

These preferences are applied after any .octaverc startup files.

Startup path Restore working directory of previous session



2. Funciones gráficas 2D elementales.

- Dibujo simplificado de funciones: **ezplot**, **ezcontour**.
- Las **ezALGO** poseen varias formas de uso, siendo la más útil por su simplicidad cuando se emplean cadenas de caracteres: **ezplot("x+2")** ó **ezplot('x+2')**
- Por defecto ezALGO grafica entre $-2\pi < x | y < 2\pi$.

Ejercicio: comprobar el funcionamiento de ezcontur

a) **ezcontour("cos(x/2)*sin(y/2)")**

b) **ezcontour("cos(x/2)+sin(y/2)")**



2. Funciones gráficas 2D elementales.

- Dado gráfico es posible definir el valor de retorno opcional de su “controlador de gráficos”.
H = ezALGO (...)

Ejercicio: comprobar las funciones anteriores en la curva plana que describe la “Espiral de Arquímedes”.

```
h=ezplot('t*cos(t)', 't*sin(t)', [0, 4*pi])
title('Espiral Arquimides')
xlabel('x(t)')
ylabel('y(t)')
legend('Espiral')
text(1,1,'Hola Mundo','fontsize',20,'color','red')
grid on
set(gca,'fontsize',20,'color',[0.2,0,0.2])
set(h,'linewidth',3,'color','green')
```



2. Operaciones con vectores y matrices

1. Vectores/matrices como arreglos de números.
2. Operaciones con vectores/matrices.
3. Tipos de matrices predefinidos.
4. Operador (:). Matriz vacía []. Borrado filas/columnas.
5. Operadores relacionales. Operadores lógicos.



1. Vectores/matrices

- Los vectores se pueden definir directamente introduciendo los elementos que lo componen, p.ej.
`>> u=[1,2,3] % vector fila`
- Observamos que **u** es un vector fila.
- En el caso de desear un vector columna utilizamos ;
`>> v=[1;2;3] % vector columna`

Ejercicio: ¿Es posible calcular la suma de **u** y **v**?

- Para transformar un vector fila en columna se debe transponer dicho vector mediante la orden **'** (comilla).
- Octave genera por defecto vectores fila:
`>> w(1)=1; w(2)=4; w(3)=9;`
`>> w`



1. Vectores/matrices

- Las matrices se crean introduciendo los elementos

```
>> A=[1 4 -3; 2 1 5; -2 5 3]
```

- Adicionalmente es posible crear matrices mediante vectores filas/columnas

```
>> f1=[1 4 -3]; f2=[2 1 5]; f3=[2 5 -3];
>> Af=[f1;f2;f3]
```

```
>> c1=[1 2 2]; c2=[4 1 5]; c3=[-3 5 -3];
>> Ac=[c1,c2,c3]
```



3. Archivos ***.m** que poseen un guión de órdenes (*scripts*).

- Los archivos con extensión **.m** son ficheros de texto sin formato (ficheros ASCII) que constituyen el centro de la programación en Octave.
- Estos archivos se crean y modifican con un editor de textos cualquiera.
- Octave 4 en adelante provee su propio editor de textos.
- Existen dos tipos de archivos ***.m**, los **ficheros de guiones de comandos** (llamados *scripts* en inglés) y las **funciones**.
- Los ***scripts*** contienen un conjunto de comandos (programa) que se ejecutan secuencialmente cuando se teclea el nombre del archivo en la ventana de órdenes o se incluye dicho nombre en otro fichero ***.m**.



Ejercicio

- **Encuentro de dos amigos**

sic. “El primer desafío serio que planteo en este libro es uno de los **rompecabezas** más bonitos de matemáticas recreativas y procede de Rusia. Su solución exige reflexión, concentración, creatividad, lógica, perspicacia y atención por los detalles.”

- **Dos matemáticos rusos se encuentran en un avión:**

- Iván: “Si no recuerdo mal, tienes 3 hijos, ¿qué edad tienen ahora?”
- Ígor: “Sí, tengo tres hijos. El producto de sus edades es 36 y la suma de sus edades es la fecha de hoy”.
- Iván: “Lo siento Ígor, pero eso no me dice la edad de tus hijos”.
- Ígor: “Oh, olvidé decírtelo. Mi hijo menor es pelirrojo”.
- Iván: “Ah, ahora está claro. Ya sé exactamente la edad de tus hijos”.

- **¿Cómo ha calculado Iván la edad de los hijos de Ígor?**



Programación en Octave

- 1. ***if/else***
 - 2. ***switch/case***
 - 3. ***for***
 - 4. ***while***
 - 5. ***function***
 - 6. Definición de funciones de usuario ***function()***.
 - 7. ***Help*** para las funciones de usuario.
 - 8. Funciones ***inline*** y ***anónimas*** “@”.
- Programación Estructurada



Programación Estructurada

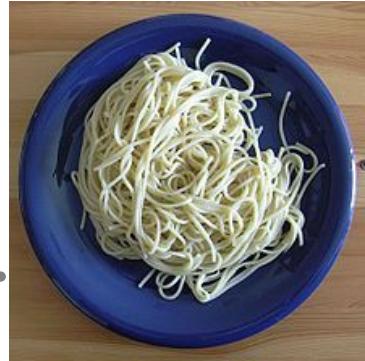
- Comúnmente es necesario realizar guiones que requieren utilizar ciertas órdenes estándares de **Programación Estructurada**.
- La **programación estructurada** es un paradigma de programación orientado a mejorar la claridad, calidad y tiempo de desarrollo de un programa de computadora, utilizando únicamente tres estructuras y subrutinas
- Estructuras: **secuencia, selección e iteración**.
 - Selección mediante sentencias condicionales o bifurcaciones:
 - **if/else**: de acuerdo a una condición
 - **case/switch**: de acuerdo al valor de una variable
 - Iteración o repetición mediante bucles:
 - **for**: un número determinado de veces
 - **while**: mientras se cumpla una condición
 - Ejecución independiente de una subrutina o subprograma:
 - **function**: realiza una tarea específica



Programación Estructurada

- Estructuras:

secuencia, selección e iteración.



- Programación secuencial comparada con el código **spaghetti**.
 - Los programas son más fáciles de entender, dado que es posible su lectura secuencial y no hay necesidad de hacer engorrosos **GOTO**.
 - La estructura de los programas es clara, puesto que las instrucciones están más ligadas o relacionadas entre sí.
 - Reducción del esfuerzo en las pruebas y depuración. El seguimiento de los fallos o errores del programa (**debugging**) es más simple.
 - Reducción de los costos de mantenimiento. Modificar o extender los programas resulta más fácil.
 - Los programas son más sencillos y más rápidos de confeccionar.



3. Bucle **for**

for-end

- En Octave *scripting* es posible realizar bucles **for**, lo que nos permite realizar ciertas operaciones un número determinado de veces.
- Este tipo de bucle es muy útil por ejemplo cuando queremos movernos a través de una lista de archivos e ir ejecutando algunas órdenes en cada archivo de la lista.

for *i=a:b* %*a, b son números enteros tal que a<b*
declaraciones

end

3. Bucle for

Ejercicio: Una estructura rígida se carga en el extremo libre como se muestra en la figura (ver figura). La tensión en el cable se puede ajustar hasta un valor máximo de 2.5kN.

La tensión del cable, que actúa en el punto **A**, produce el momento $\mathbf{M} = \mathbf{r} \times \mathbf{T}$ respecto al punto **O**. Se desea conocer como afecta la posición **B** sobre el plano **xy**, en la cual es fijado el cable si se mantiene constante la longitud del cable.

- 1) Grafique el sistema que debe ser analizado en función del ángulo $\alpha = [0, 2\pi]$ que parametriza la posición de **B** en el plano **xy**.
- 2) Graficar la norma de **M** en función del ángulo $\alpha = [0, 2\pi]$ que parametriza la posición de **B** en el plano **xy**.
- 3) Determine las posiciones de **B** en las que se logra el $\min(|\mathbf{M}|)$ y $\max(|\mathbf{M}|)$.

Ayuda: utilizar el script subido a la web del curso.

