

# Ciencias de la Computación

Curso propedéutico



# ¡BIENVENIDOS!

$$\mathbb{C}1\varepsilon(\eta)\mathbb{C}1\alpha\int_x$$

Generación 2017

**¡Bienvenidos a Ciencias de la Computación!**

¿Qué son las ciencias de la  
computación?

# Áreas de Ciencias de la Computación

- Bases de Datos
- Animación
- Teoría de la computación
- Arquitectura de computadoras
- Diseño de lenguajes de programación
- Análisis de algoritmos
- Computación distribuida
- Sistemas operativos
- Ingeniería de software
- Desarrollo de aplicaciones
- Inteligencia artificial
- Programación orientado a objetos
- Programación funcional
- Redes de computadora
- Criptografía
- Desarrollo Web
- Lógica matemática y computacional
- Graficación por computadora
- Riesgo tecnológico
- Visión por computadora
- Proceso de imágenes digitales
- Análisis y desarrollo de videojuegos
- Robótica
- Investigador
- ...

# Ciencias de la Computación

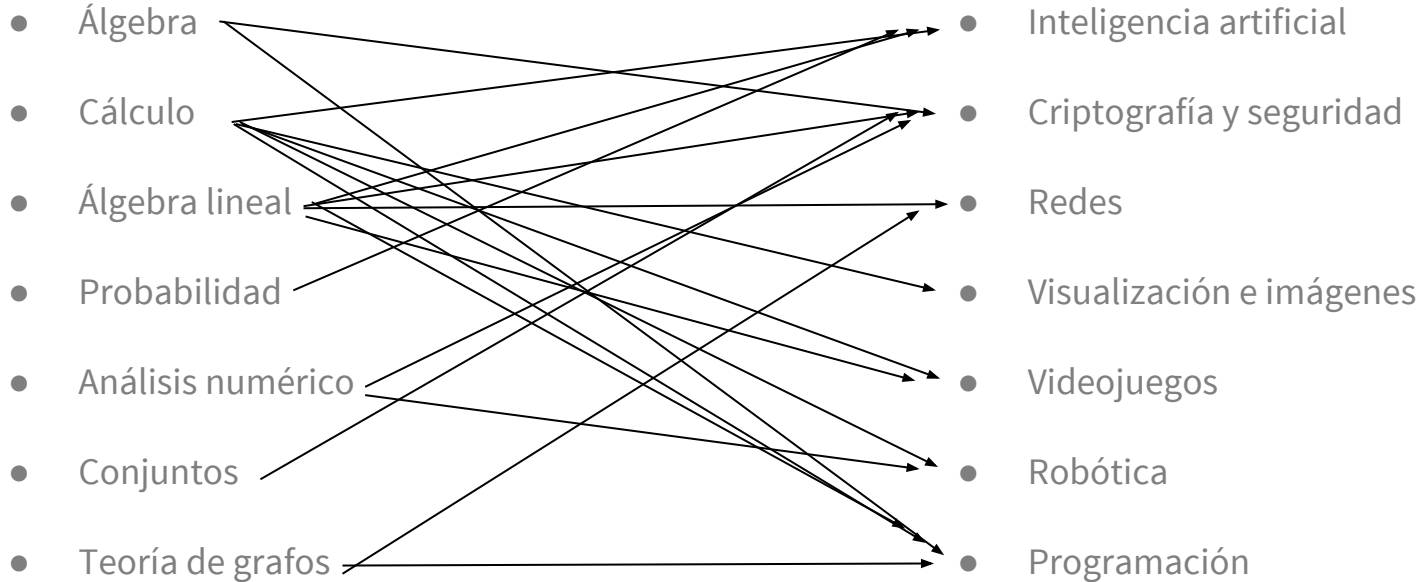
¿Cuáles son las bases?

## Matemáticas

Programación

---

# ¿Matemáticas?

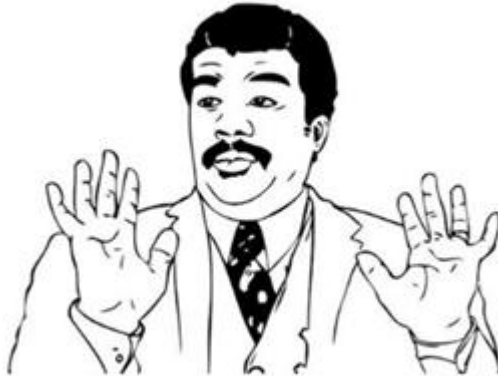


# ¿Matemáticas?

“Yo era el mejor de mi clase”

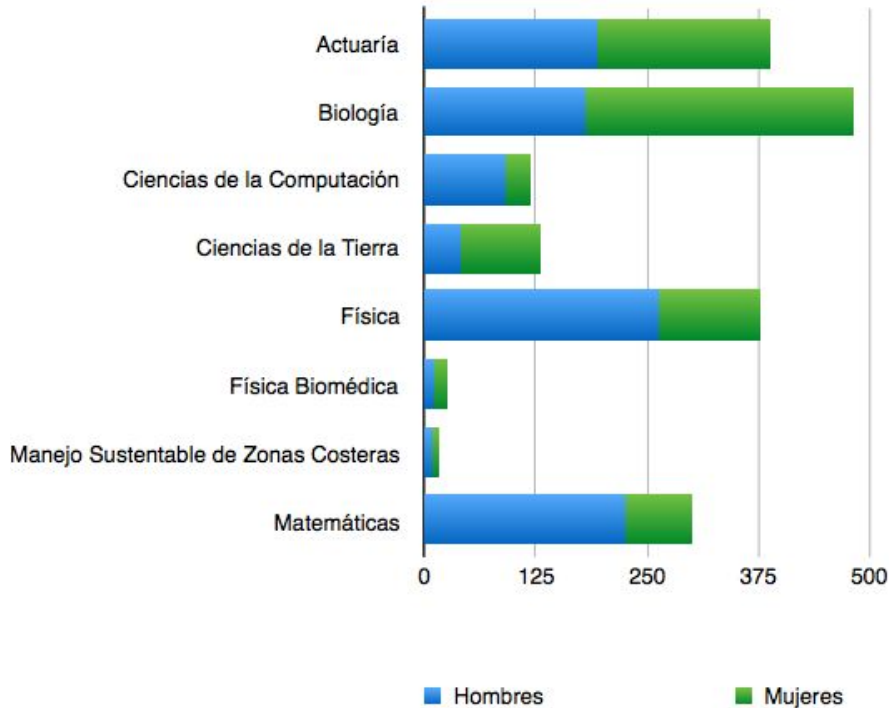
“Yo sacaba puro 10”

“Pfffff.... Las matemáticas me hacen lo que el viento a Juárez”

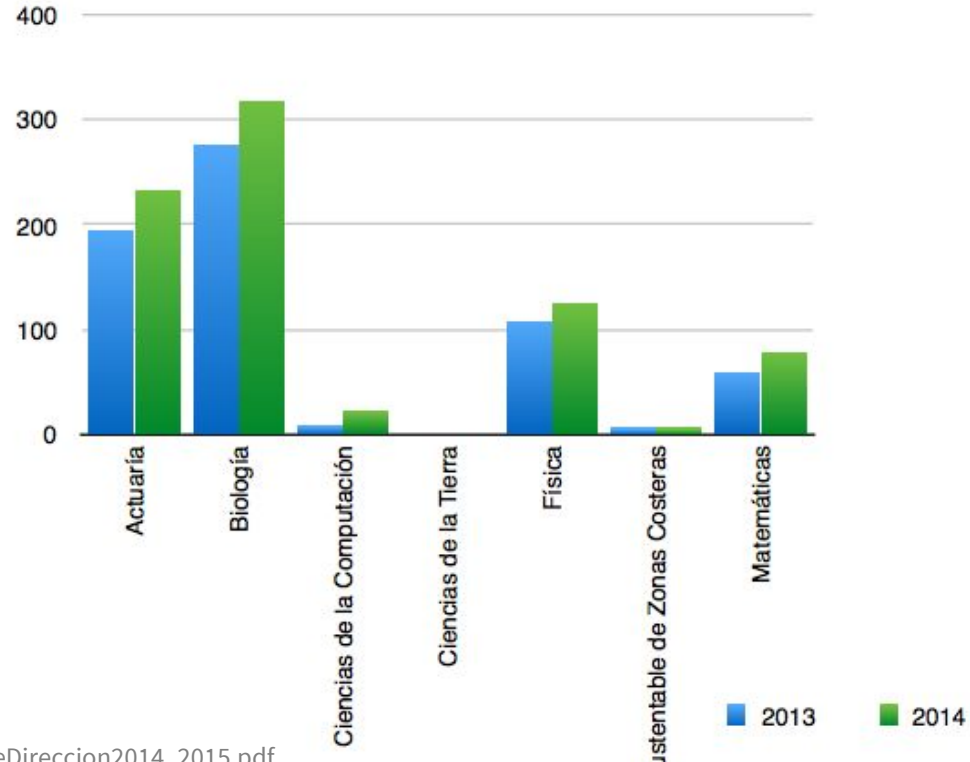


# ¡Bienvenidos a la Facultad de Ciencias!

Distribución del Primer Ingreso por carrera y género



Titulación Total por Carrera







*OH VOSOTROS LOS  
QUE ENTRÁIS,  
ABANDONAD TODA  
ESPERANZA:*

*(EXAMEN DE CÁLCULO)*

# Dificultad de la carrera

*Tabla 10. Estudiantes titulados por licenciatura en la Facultad de Ciencias, 2010-2013.*

	2010	2011	2012	2013	Tasa de crecimiento
Actuaría	89	102	156	192	115.7
Biología	227	251	242	276	21.6
Ciencias de la Computación	13	18	21	9	-30.8
Física	100	118	141	108	8
Manejo Sustentable de Zonas Costeras	0	1	5	5	100
Matemáticas	69	68	81	60	-13
<b>Total</b>	<b>498</b>	<b>558</b>	<b>646</b>	<b>650</b>	<b>30.5</b>

Fuente: DGEI con datos de la Agenda Estadística 2011, 2012, 2013 y 2014. Los años se refieren al egreso. DGAE. UNAM.

# ¿Qué debo de hacer?

1. Guardar la calma
2. Mantener la mente abierta y obedecer indicaciones
3. Aprender a demostrar
4. Aprender a programar
5. Si no funciona, volver al paso 0.

# Obtuve un cero. ¿Ahora qué?



Recuerden el siguiente tip:

“Más vale NP/5 que 6 y no saber programar/demostrar”

# Servicios de la Facultad de Ciencias

¿Qué, cómo y dónde?

Aulas de clases

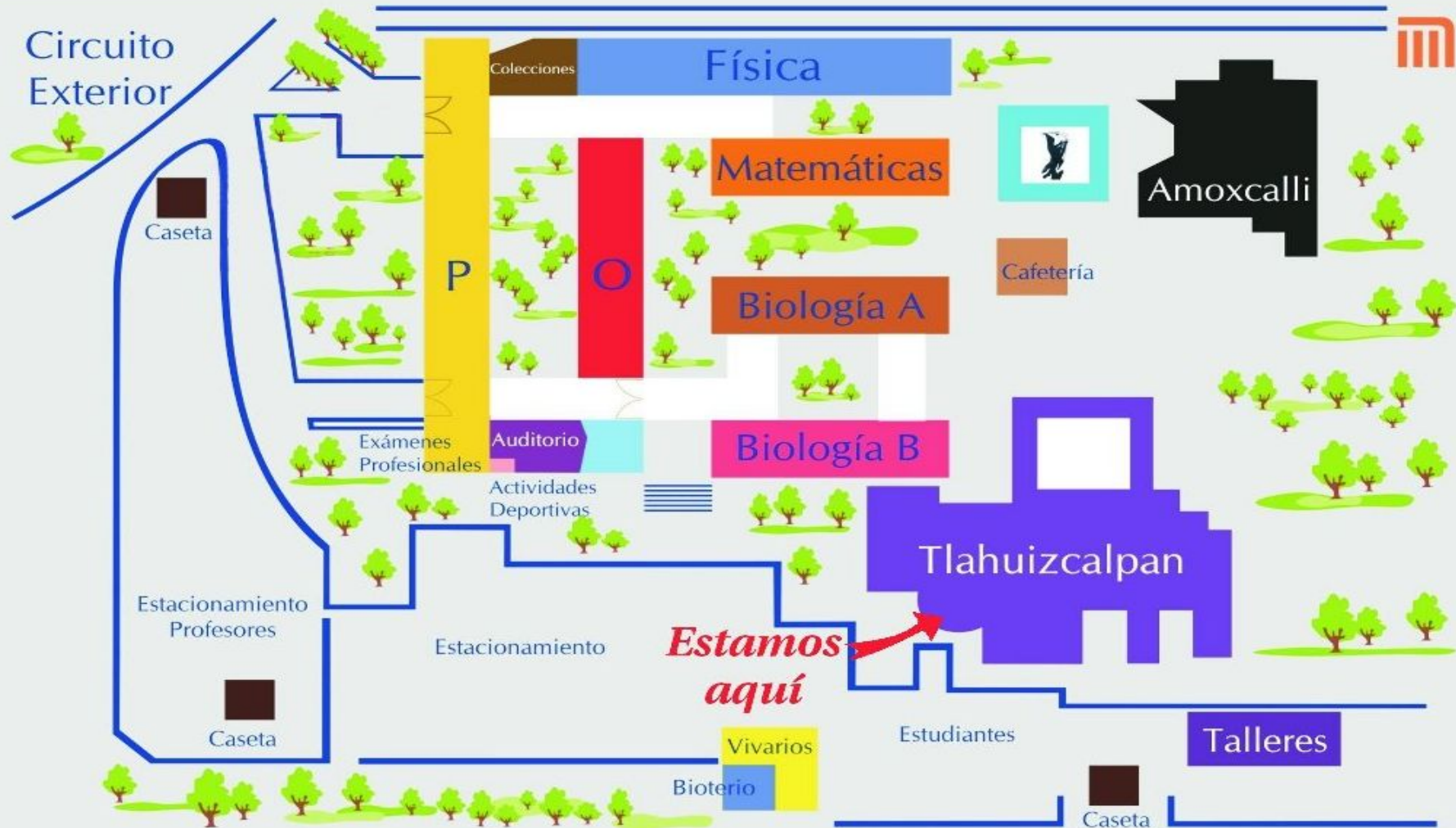
Biblioteca

Laboratorios

Técnicos académicos

ESPORA

---



# Herramientas

¿Qué necesitaré como futurox  
computólogox?

¿Computadora?

¿Programas?

¿Internet?

---

# ¿Qué necesito realmente?

- Una computadora es recomendable
  - Linux!!!! (O sistema Unix)
- Algunos programas de propósito específico
  - Editor de texto
- Algunos conocimientos previos de computación
  - Inglés



# El propedéutico

- ¿Por qué necesitamos un propedéutico?
  - Creadores  $\neq$  Usuarios
- ¿Qué vamos a ver en el propedéutico?
  - Unix
  - Linux
  - Sistema de archivos
  - Edición de texto
  - Lenguajes de marcado
  - LaTeX

Material: <http://aztlan.fciencias.unam.mx/~canek/propedeutico/>

# Unix

Un sistema operativo

¿Qué es Unix?

¿Por qué tenemos la necesidad de  
cambiar a Unix?

---

# UNIX®

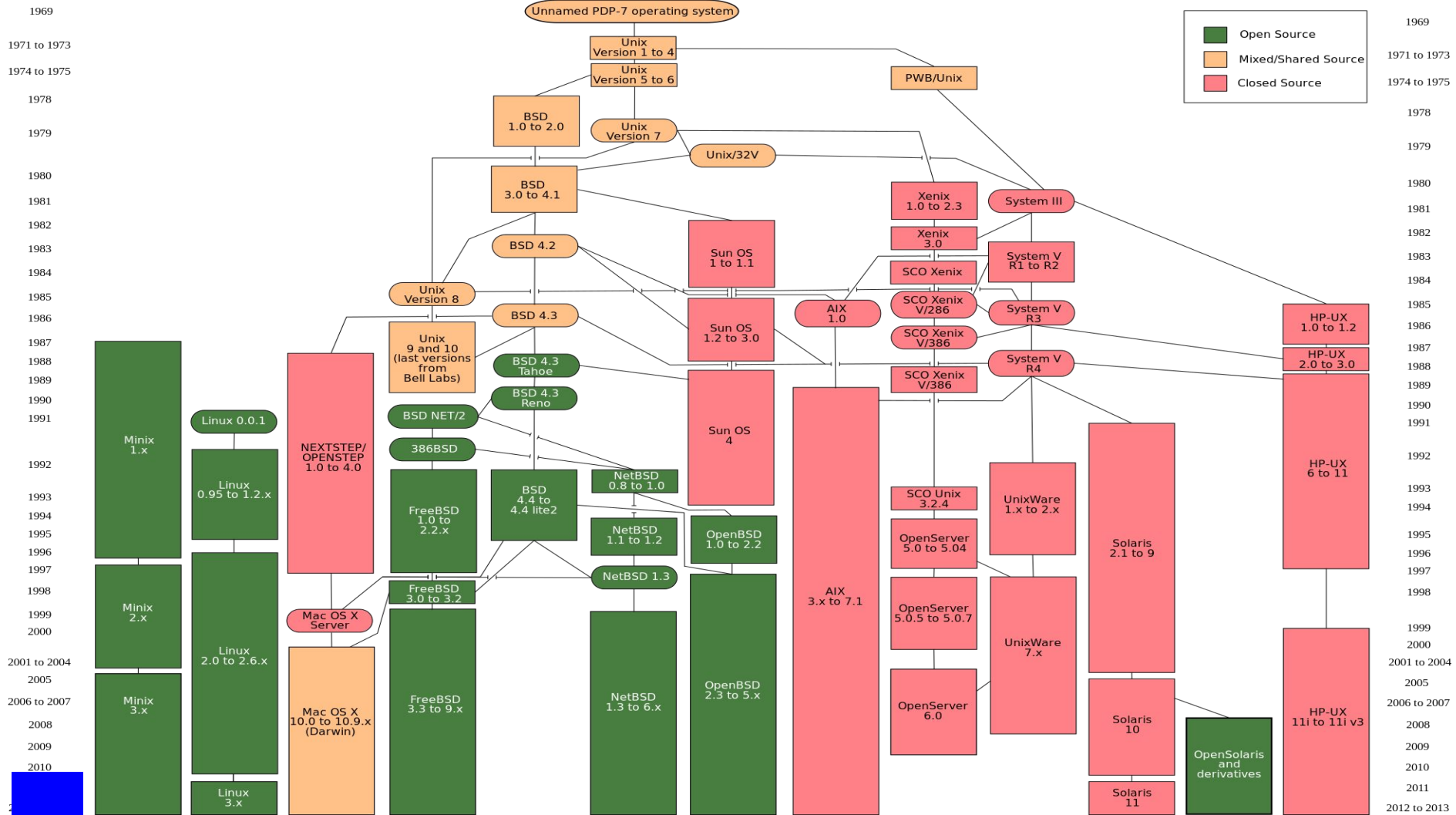
00011110 00011110 00011110 00011110 00011110 00011110 00011110 00011110

VS



# Tipos de Unix





# ¿Software?



# Linux

GNU/Linux

Historia

Distribuciones famosas

---



TODAS



# Terminal

¿Cómo nos comunicamos con la computadora?

¿Qué es?

¿Para qué nos sirve?

Importancia

¿Cómo nos comunicamos?

---

# Inicio de Sesión

GNU/Linux

¿Cómo empiezo?

¿Qué necesito?

---

# Emacs

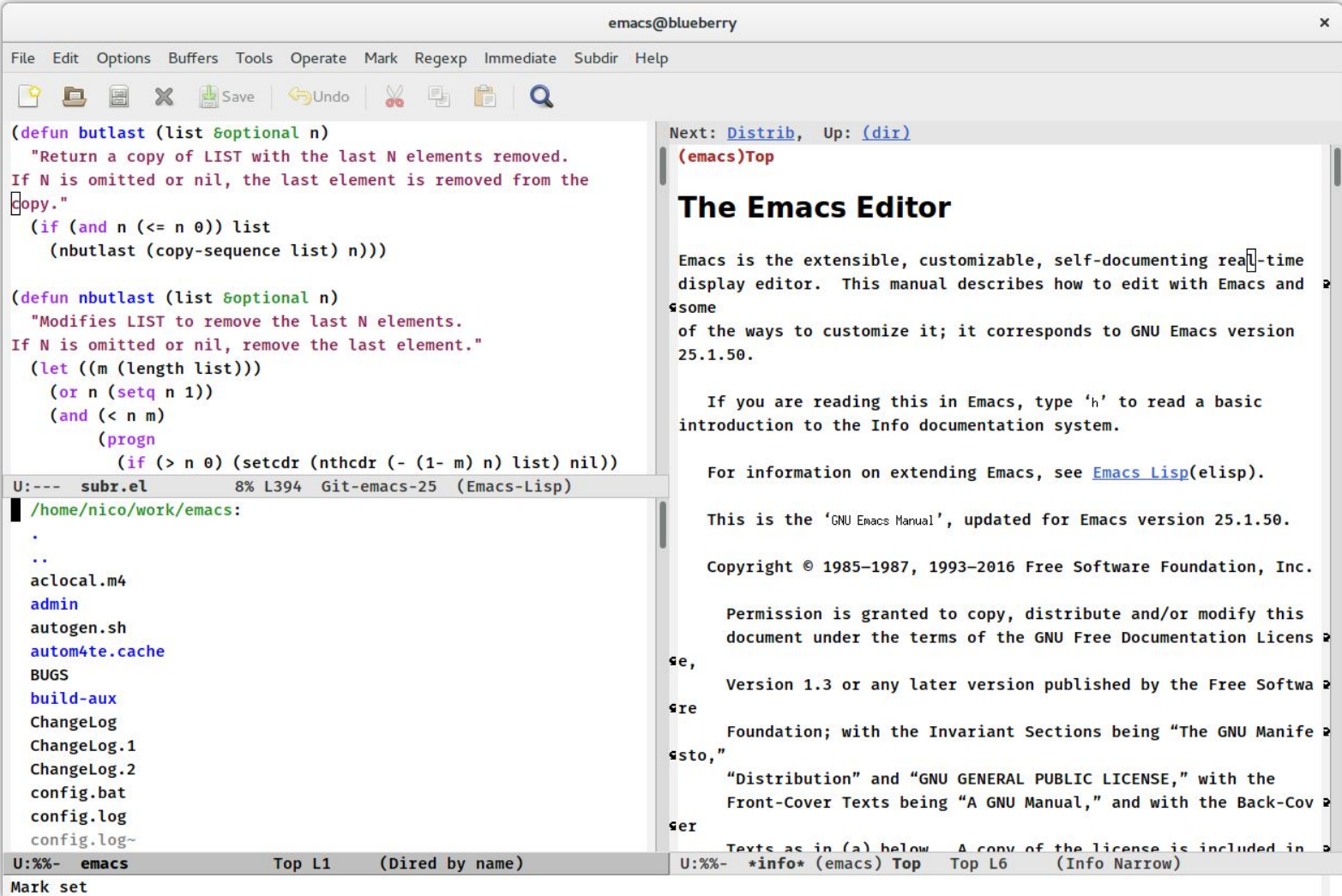
Un poderoso editor

¿Qué es emacs?

¿Por qué usar emacs?

Otros editores de pantalla

---



```
Terminal
File Edit View Search Terminal Help

48 end
49
50
51 def open
52   @imap = Net::IMAP.new(@imap_server, @imap_port, true, nil, false)
53   log @imap.login(@username, @password)
54   list_mailboxes # prefetch mailbox list
55 end
56
57 # expects a block, closes on finish
58 def with_open
59   @imap = Net::IMAP.new(@imap_server, @imap_port, true, nil, false)
60   log @imap.login(@username, @password)
61   yield self
62   close
63 end
64
65 def close
66   log "Closing connection"
67   Timeout::timeout(5) do
68     @imap.close rescue Net::IMAP::BadResponseError
69     @imap.disconnect rescue IOError
70   end
71 rescue Timeout::Error
72 end
73
74 def select_mailbox(mailbox, force=false)
75   if mailbox_aliases[mailbox]
76     mailbox = mailbox_aliases[mailbox]
77   end
78   log "Selecting mailbox #{mailbox.inspect}"
79   reconnect_if_necessary(30) do
80     log @imap.select(mailbox)
81   end
82   log "Done"
83
84   @mailbox = mailbox
85   @label = Label[name: @mailbox] || Label.create(name: @mailbox)
86
87   log "Getting mailbox status"
88   get_mailbox_status
89   log "Getting highest message id"
90   get_highest_message_id
91   return "OK"
92 end
93
94 def reload_mailbox
95   return unless STDIN.tty?
96   select_mailbox(@mailbox, true)
end

 ActionController::Helpers
-----
= Includes:
AbstractController::Helpers (from gem actionpack-3.0.5)
(from gem actionpack-3.0.5)
-----
The Rails framework provides a large number of helpers for working with
assets, dates, forms, numbers and model
objects, to name a few. These helpers are available to all templates by
default.

In addition to using the standard template helpers provided in the Rails
framework, creating custom helpers to extract complicated logic or reusable
functionality is strongly encouraged. By default, the controller will include
a helper whose name matches that of the controller, e.g.,
MyController will automatically include MyHelper.

Additional helpers can be specified using the helper class method in
ActionController::Base or any controller which inherits from it.

==== Examples
The to_s method from the Time class can be wrapped in a helper method
to display a custom message if the Time object is blank:

module FormattedTimeHelper
  def format_time(time, format=:long, blank_message="&nbsp;")
    time.blank? ? blank_message : time.to_s(format)
  end
end

FormattedTimeHelper can now be included in a controller, using the
helper class method:

class EventsController < ActionController::Base
  helper FormattedTimeHelper
  def index
    @events = Event.find(:all)
  end
end

Then, in any view rendered by EventController, the
format_time method can be called:

<% @events.each do |event| -%>
  <p>
    <%= format_time(event.time, :short, "N/A") %> | <%= event.name %>
  </p>
end

projects/vmail/lib/vmail/imap_client.rb 48,5 10% <cache/ActionController::Helpers.rvim | Press ,? for help 1.1 Top
~/ri_vim/cache/ActionController::Helpers.rvim" 58L, 2108C
0 0:vmail 1:ri_vim- 2:hellenic 3:diary 4:bash 5:bash* 6:bash 7:specky
"kaja" 16:31 07-Jul-1
```

## OPEN FILES

cuckoo.js

styles.less

app.php

t\_js

t\_Date.js

timer.php

## FOLDERS

▼ timelysis

▶ application

▶ css

▶ img

▶ js

▼ system

▶ core

▶ database

▶ fonts

▶ helpers

▶ language

▶ libraries

.htaccess

index.html

▶ templates

.htaccess

.project

\_index.html

index.php

license.txt

names.rtf

cuckoo.js

x styles.less

x app.php

x t\_js

x t\_Date.js

x timer.php

x

```
132     };
133
134     /**
135     * Custom events
136     */
137     Cuckoo.Event = {
138         bind : function(ev, callback) {
139             var calls = this._callbacks || (this._callbacks = {});
140             var list = this._callbacks[ev] || (this._callbacks[ev] = []);
141             list.push(callback);
142             return this;
143         },
144
145         unbind: function(ev, callback) {
146             var calls;
147             if (!ev) {
148                 this._callbacks = {};
149             }
150             else if (calls === this._callbacks) {
151                 if (!callback) {
152                     calls[ev] = [];
153                 }
154                 else {
155                     var list = calls[ev];
156                     if (!list) return this;
157                     for (var i = 0, l = list.length; i < l; i++) {
158                         if (callback === list[i]) {
159                             list[i] = null;
160                             break;
161                         }
162                     }
163                 }
164             }
165             return this;
166         },
167
168         fire: function(ev) {
169             var calls = this._callbacks[ev];
170             var args = Array.prototype.slice.call(arguments, 1);
171             var i, len;
172
173             for (i = 0, len = calls.length; i < len; i++) {
174                 calls[i].apply(this, args);
175             }
176             return this;
177         }
178     };
179
180     /**
181     * Model
182     */
183     Cuckoo.Model = Class.extend({
184         storageKey : null,
```

```
<%@ WebService Language="C#" Class="MathService.MathService" %>

using System;
using System.Web.Services;

namespace MathService
{
    [WebService (Namespace = "http://tenpuri.org/NumberService")]
    public class MathService : WebService
    {
        [WebMethod]
        public int AddNumbers (int number1, int number2)
        {
            return number1 + number2;
        }

        [WebMethod]
        public int SubtractNumbers (int number1, int number2)
        {
            return number1 - number2;
        }
    }
}
```

[ Read 23 lines ]

F1 Get Help

^S WriteOut

^O Read File

F7 Prev Page

^X Cut Text

^T To Spell

^Q Exit

^J Justify

^F Where Is

F8 Next Page

^U UnCut Text

^\_ First Line

# GUI

Graphical User Interface

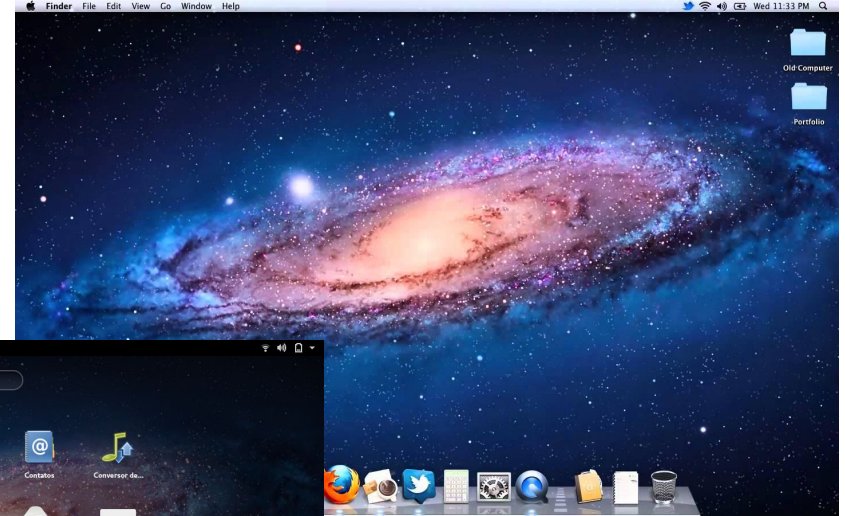
Definición

GUI vs CLI

---



# Windows Vs Linux Vs Mac



# GNOME



# Sistema de archivos

Usando la computadora

Rutas

Posiciones del árbol y permisos

Archivos estándar

Comandos básicos

---

# Comandos (Pte 1)

- Rutas absolutas y relativas
- pwd
- ls [-la]
- cd [ruta]
- cat [file]
- echo [text]
- jobs
- fg [#]
- C+z, C+c
- touch [file]
- mv [origen] [destino]
- rm [file]
- mkdir [dir]
- rmdir [dir]
- less
- yes
- > (redirección de texto)

# Comandos (Pte 2)

- |                   |                                   |
|-------------------|-----------------------------------|
| i. chmod          | i. cp                             |
| ii. <             | ii. file                          |
| iii. > vs >>      | iii. cmp                          |
| iv. sort          | iv. wc (lineas, palabras y bytes) |
| v.   entubamiento | v. &&,   , ;                      |
| vi. ln            |                                   |

¿Qué pasa si quieren crear un archivo que se llame #hola ?hola?

# Comandos (Pte 3)

- i. Comodines: ?,\*[list]tar (create, verbose, file)
- ii. command &
- iii. kill
- iv. ps (process status)
- v. grep
- vi. paquetería: apt-get [update|upgrade|install]
- vii. C+r
- viii. \$USER

# Actividad (15 min)

Ejecuta los siguientes comandos de UNIX en una terminal. Al finalizar entrega un reporte donde describas lo que crees que hace cada uno.

\$ echo hello world

\$ hostname

\$ uname -a

\$ uptime

\$ id

\$ w

\$ echo \$SHELL

\$ cal 2016

\$ history

\$ date

\$ arch

\$ dmesg | more

\$ who am i

\$ last

\$ top

\$ clear

\$ yes please

\$ time sleep 5

**Un comando útil**

man



# Usando Emacs

Un poderoso editor de textos

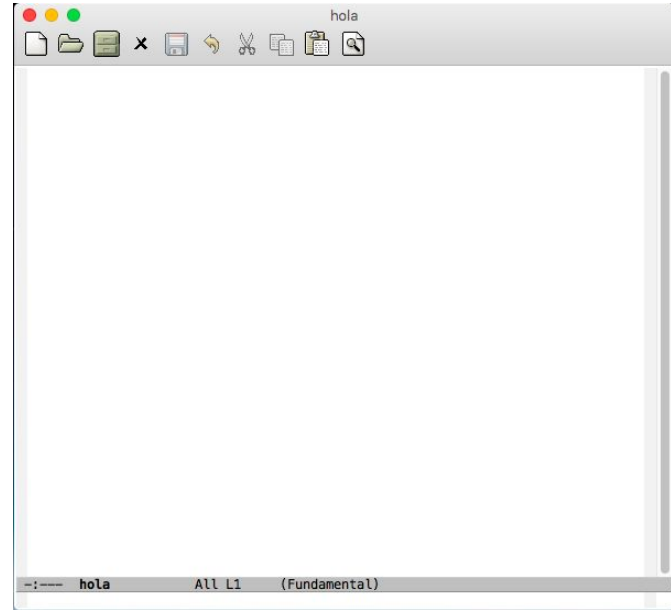
Sistema de archivos

Edición de texto

---

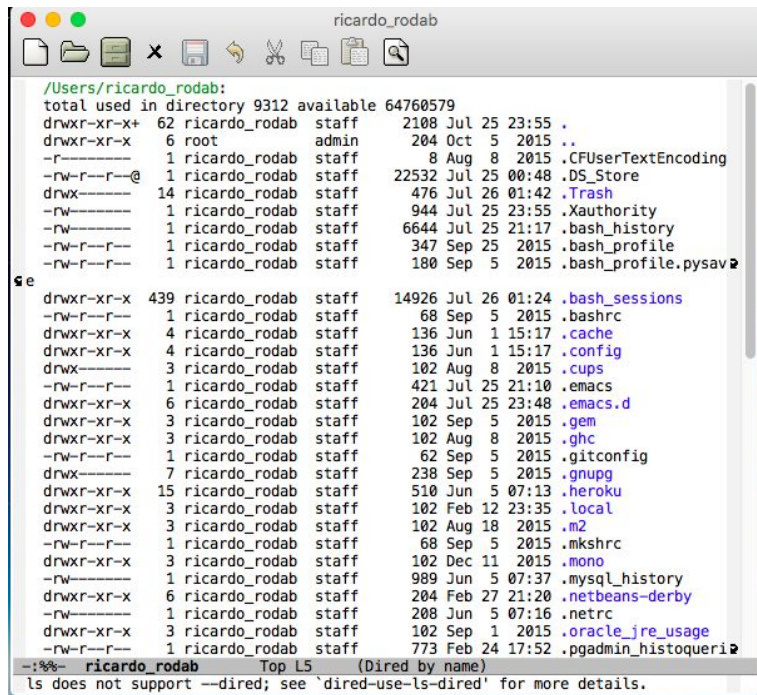
# Buffers

- Un buffer o una pantalla son aquellas en las que trabajamos.
  - Crear un buffer: C+x b
  - Cerrar (matar) un buffer: C+x k
  - Cambiar de buffer: C+x -> ó C+x <-
  - Dividir pantalla: C+x 2 ó C+x 3
  - Cerrar división: C+x 0
  - Cambiar de división: C+x o



# Navegación de archivos

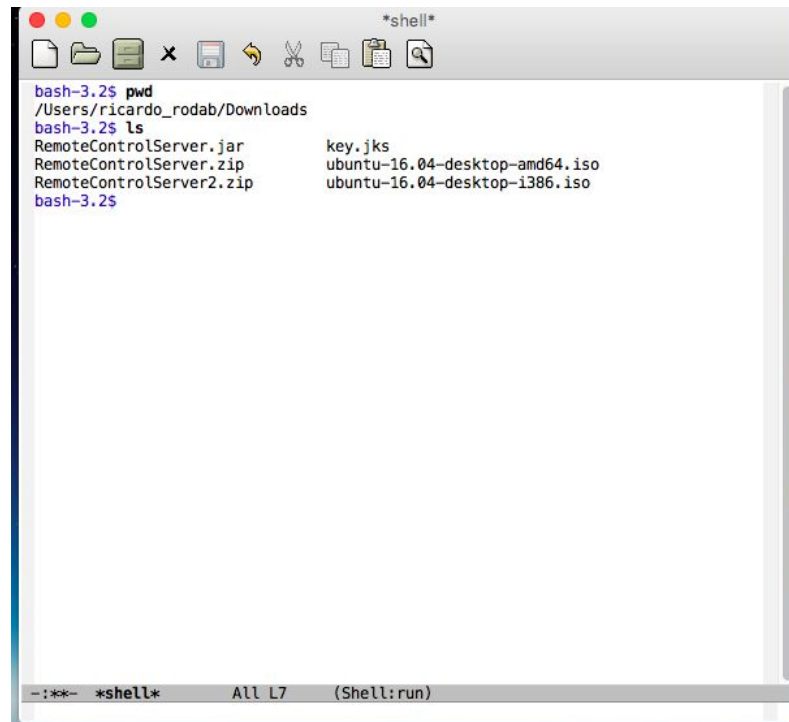
- Navegar por el sistema de archivos del computador
  - Con m marcas un archivo
    - R: rename
    - C: copy
  - Con d marcas un archivo para eliminar
    - Con x lo eliminas



```
ricardo_rodab
/Users/ricardo_rodab:
total used in directory 9312 available 64760579
drwxr-xr-x+ 62 ricardo_rodab staff 2108 Jul 25 23:55 .
drwxr-xr-x  6 root admin 204 Oct 5 2015 ..
-r-----  1 ricardo_rodab staff  8 Aug 8 2015 .CFUserTextEncoding
-rw-r--r--@ 1 ricardo_rodab staff 22532 Jul 25 00:48 .DS_Store
drwx----- 14 ricardo_rodab staff  476 Jul 26 01:42 .Trash
-rw-----  1 ricardo_rodab staff  944 Jul 25 23:55 .Xauthority
-rw-----  1 ricardo_rodab staff 6644 Jul 25 21:17 .bash_history
-rw-r--r--  1 ricardo_rodab staff  347 Sep 25 2015 .bash_profile
-rw-r--r--  1 ricardo_rodab staff  180 Sep 5 2015 .bash_profile.pysav
%~
drwxr-xr-x 439 ricardo_rodab staff 14926 Jul 26 01:24 .bash_sessions
-rw-r--r--  1 ricardo_rodab staff  68 Sep 5 2015 .bashrc
drwxr-xr-x  4 ricardo_rodab staff  136 Jun 1 15:17 .cache
drwxr-xr-x  4 ricardo_rodab staff  136 Jun 1 15:17 .config
-rwx-----  3 ricardo_rodab staff  102 Aug 8 2015 .cups
-rw-r--r--  1 ricardo_rodab staff  421 Jul 25 21:10 .emacs
drwxr-xr-x  6 ricardo_rodab staff  204 Jul 25 23:48 .emacs.d
drwxr-xr-x  3 ricardo_rodab staff  102 Sep 5 2015 .gem
drwxr-xr-x  3 ricardo_rodab staff  102 Aug 8 2015 .ghc
-rw-r--r--  1 ricardo_rodab staff  62 Sep 5 2015 .gitconfig
drwx-----  7 ricardo_rodab staff  238 Sep 5 2015 .gnupg
drwxr-xr-x 15 ricardo_rodab staff  510 Jun 5 07:13 .heroku
drwxr-xr-x  3 ricardo_rodab staff  102 Feb 12 23:35 .local
drwxr-xr-x  3 ricardo_rodab staff  102 Aug 18 2015 .m2
-rw-r--r--  1 ricardo_rodab staff  68 Sep 5 2015 .mkshrc
drwxr-xr-x  3 ricardo_rodab staff  102 Dec 11 2015 .mono
-rw-----  1 ricardo_rodab staff  989 Jun 5 07:37 .mysql_history
drwxr-xr-x  6 ricardo_rodab staff  204 Feb 27 21:20 .netbeans-derby
-rw-----  1 ricardo_rodab staff  208 Jun 5 07:16 .netrc
drwxr-xr-x  3 ricardo_rodab staff  102 Sep 1 2015 .oracle_jre_usage
-rw-r--r--  1 ricardo_rodab staff  773 Feb 24 17:52 .pgadmin_histoqueria
~%~ ricardo_rodab Top L5 (Dired by name)
ls does not support --dired; see 'dired-use-ls-dired' for more details.
```

# Shell en Emacs

- Emacs incluye un prompt:
  - M+! ejecuta un solo comando
  - M+x shell abre una ventana con el shell



The screenshot shows an Emacs window titled '\*shell\*' with a standard macOS title bar (red, yellow, green buttons). The window contains a terminal-like interface with the following text:

```
bash-3.2$ pwd
/Users/ricardo_rodab/Downloads
bash-3.2$ ls
RemoteControlServer.jar      key.jks
RemoteControlServer.zip      ubuntu-16.04-desktop-amd64.iso
RemoteControlServer2.zip     ubuntu-16.04-desktop-i386.iso
bash-3.2$
```

At the bottom of the window, the status bar displays: `--:*-- *shell* All L7 (Shell:run)`.

# Comandos importantes

- a. M-# [command]
- b. M-x
- c. query-replace
- d. string-replace
- e. Selección de texto
- f. Copiar texto
- g. Cortar texto
- h. Pegar texto
- i. Buscar
- j. Escribir y guardar

# Shell scripts

Archivos ejecutables

Crear un archivo shell

Asignar parámetros

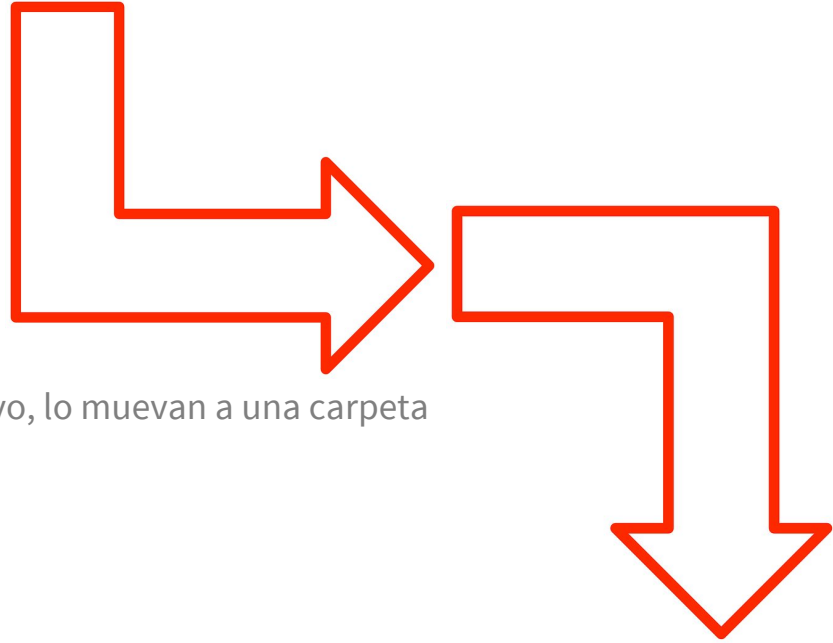
Editar permisos

---

# ¿Qué es un script?

- Archivo de órdenes:
  - Evitar repetir
  - Tedioso de volver a escribir
  - Versatilidad

- Ejercicio:
  - Crear un script donde creen un archivo, lo muevan a una carpeta nueva y escriban algo dentro de él.



# Convención

- TODOS los scripts deben empezar con la siguiente línea:
  - `#!/bin/bash`
- Los comentarios empiezan con #



# Parámetros

- Para generar archivos de ejecución genéricos les debemos pasar parámetros.
  - El parámetro y número es reconocido por el símbolo de \$ y el número de parámetro

Ejemplo:

echo \$1

# Actividad (20 minutos)

1. Ingresar a emacs
2. Abrir un buffer con nombre "script"
3. Escribir un script que haga lo siguiente:
  - a. Imprimir en pantalla "Inicia el script 1"
  - b. Mostrar en pantalla lo que hay en la carpeta actual
  - c. Crear una carpeta llamada "magia"
  - d. Crear un archivo llamado "alibaba"
  - e. Dentro de alibaba escribir el siguiente texto "Hocus pocus"
  - f. Mostrar en pantalla lo que tiene el archivo alibaba
  - g. Imprimir "Fin del script 1"
4. Guardar el script con nombre "s1.sh"
5. Cambiar los permisos para hacer s1.sh ejecutable
7. Escribir un script que haga lo siguiente:
  - Imprimir en pantalla "Inicia el script 2"
  - Correr el script "s1.sh"
  - Crear un archivo con el nombre que le pase como parámetro
  - Mostrar en pantalla lo que hay en la carpeta actual
  - Mover el archivo creado renombrándolo con el nombre que le pase como segundo parámetro
  - Imprimir "Fin del script 2"
8. Levantar la mano para que probemos su script y anotar su nombre

# Manejador de versiones

Git

¿Cómo no perder archivos?

---

# Lenguajes de marcado

XML

HTML

CSS

---

# **XML**

eXtensible Markup Language

¿Qué es?

¿Para qué nos sirve?

---

# Esquema DTD

---

## DTD para archivo en XML

---

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY iacute "&#262;">
<!ENTITY oacute "&#268;">
...
<!ELEMENT libro ( titulo , autor+ , parte+ ) >
<!ELEMENT titulo (#PCDATA) >
<!ELEMENT autor (#PCDATA) >
<!ELEMENT parte ( capitulo+ >)
<!ATTLIST parte numero CDATA #REQUIRED >
<!ELEMENT capitulo ( resumen , parrafo+ ) >
<!ATTLIST capitulo numero CDATA #REQUIRED >
<!ELEMENT resumen ( #PCDATA ) >
<!ELEMENT parrafo ( #PCDATA ) >
```

---

# Ejemplo

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- Este es el DTD de Edit_Mensaje -->

<!ELEMENT Mensaje (Remitente, Destinatario, Texto)*>
<!ELEMENT Remitente (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Destinatario (Nombre, Mail)>
<!ELEMENT Nombre (#PCDATA)>
<!ELEMENT Mail (#PCDATA)>
<!ELEMENT Texto (Asunto, Parrafo)>
<!ELEMENT Asunto (#PCDATA)>
<!ELEMENT Parrafo (#PCDATA)>
```

```
<?xml version="1.0" encoding="UTF-8" ?>
<!DOCTYPE Edit_Mensaje SYSTEM "Edit_Mensaje.dtd">

<Edit_Mensaje>
  <Mensaje>
    <Remitente>
      <Nombre>Nombre del remitente</Nombre>
      <Mail>Correo del remitente </Mail>
    </Remitente>
    <Destinatario>
      <Nombre>Nombre del destinatario</Nombre>
      <Mail>Correo del destinatario</Mail>
    </Destinatario>
    <Texto>
      <Asunto>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Asunto>
      <Parrafo>
        Este es mi documento con una estructura muy sencilla
        no contiene atributos ni entidades...
      </Parrafo>
    </Texto>
  </Mensaje>
</Edit_Mensaje>
```

# HTML

Etiquetas

Origen

¿Cómo funciona?

---



# LaTeX

Lenguaje de programación

¿Qué es LaTeX?

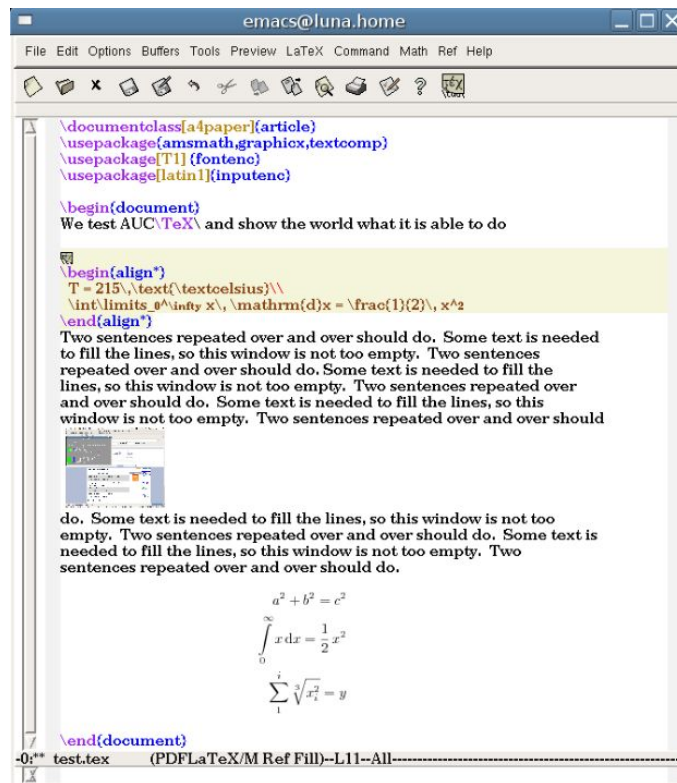
¿Para que me sirve LaTeX?

Compilado

---

# ¿Qué es LaTeX?

- Lenguaje de programación
- Poderoso procesador de texto
  - Procesador de textos ≠ Editor de textos



Cuando escribes tus ensayos en lenguajes de programación

# UNIX SHELL



# LATEX



# HTML



# ¿Para qué me sirve?

## 3 Mathematical Equations

Simple equations, like  $x^y$  or  $x_n = \sqrt{a+b}$  can be typeset right in the text line by enclosing them in a pair of single dollar sign symbols. Don't forget that if you want a real dollar sign in your text, like \$2000, you have to use the `\$` command.

A more complicated equation should be typeset in *displayed math* mode, like this:

$$z \left( 1 + \sqrt{\omega_{i+1} + \zeta - \frac{x+1}{\Theta+1}y+1} \right) = 1$$

The “equation” environment displays your equations, and automatically numbers them consecutively within your document, like this:

$$\left[ \mathbf{X} + \mathbf{a} \geq \hat{\mathbf{a}} \sum_i^N \lim_{x \rightarrow k} \delta C \right] \quad (1)$$

# Código ejemplo:

```
\documentclass{article}
\textheight = 20cm
\textwidth = 18cm
\topmargin = -2cm
\parindent = 0mm
\usepackage{amsmath,amssymb,amsfonts,latexsym,cancel}
%usepackage{graphicx} Esto es para importar el paquete para cosas gráficas
% Empieza el documento:
\begin{document}
% Sección:
\section{Preliminares}
    {\bf Distancia entre dos puntos.}
    Recordemos que la distancia euclidiana de un punto  $A=(a,b)$  a otro punto  $B=(p,q)$  es
    
$$d(A,B)=||A-B||=\sqrt{(a-p)^2+(b-q)^2}$$

\end{document}
```

# Resultado ejemplo:

## 1 Preliminares

Distancia entre dos puntos. Recordemos que la distancia euclidiana de un punto  $A = (a, b)$  a otro punto  $B$  es

$$d(A, B) = \|A - B\| = \sqrt{(a - p)^2 + (b - q)^2}$$

# Compilado de .tex

- Compilar es crear (dado un código fuente) un archivo ejecutable o resultado.
  - Java, C, C++, LaTeX ...
- Comandos para compilar
  - latex, pdflatex, pdftex



# Encabezado

- `\documentclass{param}` Donde param= article | report | book
- `\textheight=19cm` Establece el largo del texto de tamaño 19 centímetros
- `\textwidth=14cm` Establece el ancho. El default es de 14 cm
- `\topmargin=-2cm` Establece un margen superior. Si es -2 sube 2 cm.
- `\usepackage{amsmath,amssymb,amsfonts,latexsym,cancel}`
- `\usepackage[latin1]{inputenc}` Esta instrucción se usa para incluir un paquete que nos permite usar los acentos
- ...etc.



# Codificación

- Problema muy grande con LaTeX.

Solución:

1. `\usepackage[latin1]{inputenc}`
2. `\usepackage[utf8]{inputenc}`

Comando	Símbolo	Comando	Símbolo
<code>\'a</code>	á	<code>?`</code>	¿
<code>\'e</code>	é	<code>!`</code>	¡
<code>\'{\i}</code>	í	<code>`` "</code>	""
<code>\'o</code>	ó	<code>` '</code>	''
<code>\'u</code>	ú	<code>\~n</code>	ñ

**Tabla 2.1** Acentos en modo texto y otros símbolos

# Caracteres especiales

Caracter	
\	carácter inicial de comando
{ }	abre y cierra bloque de código
\$	abre y cierra el modo matemático
&	tabulador (en tablas y matrices)
#	señala parámetro en las macros
_, ^	para subíndices y exponentes
~	para evitar cortes de renglón
%	para comentarios

# El comando verb

- El comando `\verb` permite imprimir los caracteres tal y como aparecen en pantalla. Por ejemplo, si se digita `\verb@{ $ x^2+1$ }@` se imprimirá `{ $ x^2+1 $ }`

## Modificadores de texto

Comando	Produce
<code>{\rm Roman }</code>	Roman
<code>{\em Enf\'atico }</code>	<i>Enfático</i>
<code>{\bf Negrita }</code>	<b>Negrita</b>
<code>{\it It\'alica }</code>	<i>Itálica</i>

Comando <code>{\sl Slanted }</code>	<i>Slanted</i>
<code>{\sf Sans Serif }</code>	Sans Serif
<code>{\sc Small Caps }</code>	SMALL CAPS
<code>{\tt Typewriter }</code>	Typewriter
<code>\underline{Subrayado }</code>	<u>Subrayado</u>

# Ejercicio 1: ¿Qué salida produce esto?

```
\documentclass{article}
\usepackage{fix-cm} % En algunos casos es necesario
\usepackage{anyfontsize}
\begin{document}
Este es {\fontsize{80}{0}\selectfont LaTeX}
\end{document}
```

# Modo matemático

LaTeX

Potencias

Índice

Raíces

Expresiones

Etc.

---

# ¿Fórmula o texto?

**TODA fórmula matemática en LaTeX se escribe entre los caracteres **\$\$****

**Ejemplo:**

**$\$x^2\$$  va a escribir  $x^2$**

# Potencias, subíndices y superíndices

Expresión	Código	Expresión	Código
$x^p$	<code>x<sup>p</sup></code>	$x^{n+1}$	<code>x<sup>{n+1}</sup></code>
$(2^2)^n$	<code>(2<sup>2</sup>)<sup>n</sup></code>	$2^{2^n}$	<code>2<sup>{2<sup>n</sup>}</sup></code>
$\sin^2(x)$	<code>\sin<sup>2</sup>(x)</code>	$x^{\sin(x)+\cos(x)}$	<code>x<sup>{\sin (x)+ \cos (x)}</sup></code>
$a_n$	<code>a<sub>n</sub></code>	$a_{n+1}$	<code>a<sub>{n+1}</sub></code>
$u_{N+1}$	<code>u<sub>{N+1}</sub></code>	$u_{_{N+1}}$	<code>u<sub>{_{N+1}}</sub></code>
$a_i^j$	<code>a<sub>i</sub><sup>j</sup></code>	$\int_a^b f(x) dx$	<code>\int<sub>a</sub><sup>b</sup> f(x) \, dx</code>
$\sum_{n=1}^N u_n$	<code>\sum<sub>{n=1}</sub><sup>{N}</sup> u<sub>n</sub></code>	$u_{ij}$	<code>u<sub>{ij}</sub></code>

# Raíces

Expresión	Código
$\sqrt{x+1}$	<code>\sqrt{x+1}</code>
$\sqrt[n]{x+\sqrt{x}}$	<code>\displaystyle{ \sqrt[n]{x+\sqrt{x}} }</code>
$\sqrt[n]{x+\sqrt{x}}$	<code>\sqrt[n]{x+\sqrt{x}}</code>

---



# Fracciones

Expresión	Código
$\frac{x+1}{x-1}$	<code>{x+1 \over x-1}</code>
$\frac{x+1}{x-1}$	<code>\displaystyle \frac{x+1}{x-1}</code>
$\frac{\frac{x+1}{3}}{x-1}$	<code>{{x+1 \over 3} \over x-1}</code>
$\left(1 + \frac{1}{x}\right)^{\frac{n+1}{n}}$	<code>\displaystyle{\left( 1+ {1 \over x} \right)^{n+1 \over n}}</code>
$\left(1 + \frac{1}{x}\right)^{\frac{n+1}{n}}$	<code>\displaystyle \left( 1+ \frac{1}{x} \right)^{\frac{n+1}{n}}</code>
$\left(1 + \frac{1}{x}\right)^{\frac{n+1}{n}}$	<code>\displaystyle{\left( 1+ {1 \over x} \right)}^{\displaystyle{n+1 \over n}}</code>
$\frac{x+1}{x-1}$	<code>{x+1 \atop x-1}</code>

## Ejercicio 2 (Pte 1)

Escribir la siguiente plantilla en un archivo .tex

```
\documentclass{article}
\textheight=19cm
\textwidth=14cm
\topmargin=-2cm
\usepackage{amsmath,amssymb,amsfonts,latexsym,cancel}
\usepackage[spanish]{babel}
\usepackage[utf8]{inputenc}

\begin{document}
% A partir de aqui empezarán a escribir.
\end{document}
```

## Ejercicio 2 (Pte 2)

- Con lo visto en clase, escribir las siguientes fórmulas con su respectivo nombre:
  1. La chicharronera
  2. La distancia entre dos puntos en el plano euclidiano
  3. La fórmula general de la hipérbola
  4. La integral definida de 2 a 3 de “dos equis dx”
  5. La identidad de Euler
  6. El producto desde  $i=0$  hasta  $i = 1$  de  $(2i+2)$
  7. La suma desde  $i=1$  hasta  $i=3$  de  $(2i-1)$
  8.  $5\cos(\pi/2) + (1/\cos 60^\circ)$
  9. La integral de menos infinito a infinito de  $1/(x+1)^3$
  10. La vaca vestida de uniforme. ;)