



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: **ALEJANDRO PIMENTEL**

Asignatura: **FUNDAMENTOS DE PROGRAMACIÓN**

Grupo: **BLOQUE 135**

No de Práctica(s): **PRACTICA 6 “Entorno C”**

Integrante(s): **ALITZEL ANAID GUTIÉRREZ RAMOS**

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: **9370**

Semestre: **1er SEMESTRE**

Fecha de entrega: **30-09-2019**

Observaciones: Muy bien, pero te recomiendo que uses acercamientos a las imagenes para que se puedan ver con claridad. Como las usas en este momento no se pueden distinguir facilmente.

CALIFICACIÓN: 10

INTRODUCCIÓN

EL lenguaje C es el resultado de un proceso de desarrollo que inició con un lenguaje denominado BCPL. Este influyó a otro llamado B (inventado por Ken Thompson). En los años 70; este lenguaje llevó a la aparición del C.

Con la popularidad de las microcomputadoras muchas compañías comenzaron a implementar su propio C por lo cual surgieron discrepancias entre sí.

OBJETIVO

Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

SECCIONES EQUIVALENTES

LENGUAJE C

C es un lenguaje de programación originalmente desarrollado por Dennis M. Ritchie entre 1969 y 1972 en los Laboratorios Bell, como evolución del anterior lenguaje B, a su vez basado en BCPL.

Al igual que B, es un lenguaje orientado a la implementación de Sistemas Operativos, concretamente Unix. C es apreciado por la eficiencia del código que produce y es el lenguaje de programación más popular para crear software de sistemas, aunque también se utiliza para crear aplicaciones.



Se trata de un lenguaje de tipos de datos estáticos, débilmente tipificado, de medio nivel pero con muchas características de bajo nivel. Dispone de las estructuras típicas de los lenguajes de alto nivel, pero, a su vez, dispone de construcciones del lenguaje que permiten un control a muy bajo nivel. Los compiladores suelen ofrecer extensiones al lenguaje que posibilitan mezclar código en ensamblador con código C o acceder directamente a memoria o dispositivos periféricos.

La primera estandarización del lenguaje C fue en ANSI, con el estándar X3.159-1989. El lenguaje que define este estándar fue conocido vulgarmente como ANSI C. Posteriormente, en 1990, fue ratificado como estándar ISO (ISO/IEC 9899:1990). La adopción de este estándar es muy amplia por lo que, si los programas creados lo siguen, el código es portable entre plataformas y/o arquitecturas.

CARACTERÍSTICAS

- Un núcleo del lenguaje simple, con funcionalidades añadidas importantes, como funciones matemáticas y de manejo de archivos, proporcionadas por bibliotecas.
- Es un lenguaje muy flexible que permite programar con múltiples estilos. Uno de los más empleados es el estructurado "no llevado al extremo" (permitiendo ciertas licencias de ruptura).
- Un sistema de tipos que impide operaciones sin sentido.
- Usa un lenguaje de preprocesador, el preprocesador de C, para tareas como definir macros e incluir múltiples archivos de código fuente.
- Acceso a memoria de bajo nivel mediante el uso de punteros.
- Interrupciones al procesador con uniones.
- Un conjunto reducido de palabras clave.
- Por defecto, el paso de parámetros a una función se realiza por valor. El paso por referencia se consigue pasando explícitamente a las funciones las direcciones de memoria de dichos parámetros.
- Tipos de datos agregados (struct) que permiten que datos relacionados (como un empleado, que tiene un id, un nombre y un salario) se combinen y se manipulen como un todo (en una única variable "empleado").
- Soporte para programación orientada a objetos, aunque la implementación original de C++ fue un preprocesador que traducía código fuente de C++ a C.

Aunque la lista de las características útiles de las que carece C es larga, este factor ha sido importante para su aceptación, porque escribir rápidamente nuevos compiladores para nuevas plataformas, mantiene lo que realmente hace el programa bajo el control directo del programador, y permite implementar la solución más natural para cada plataforma. Ésta es la causa de que a menudo C sea más eficiente que otros lenguajes. Típicamente, sólo la programación cuidadosa en lenguaje ensamblador produce un código más rápido, pues da control total sobre la máquina, aunque los avances en los compiladores de C y la complejidad creciente de los microprocesadores modernos han reducido gradualmente esta diferencia.

En algunos casos, una característica inexistente puede aproximarse. Por ejemplo, la implementación original de C++ consistía en un preprocesador que traducía código fuente C++ a C. La mayoría de las funciones orientadas a objetos incluyen un puntero especial, que normalmente recibe el nombre "this", que se refiere al objeto al que pertenece la función. Mediante el paso de este puntero como un argumento de función, esta funcionalidad puede desempeñarse en C.

<http://www.ictea.com/cs/index.php?rp=/knowledgebase/8834/iQue-es-el-lenguaje-de-programacion-C.html>

ACTIVIDADES DESARROLLADAS

1. Hice una investigación muy somera acerca de diferentes tipos de archivos:

El texto plano es el tipo de archivo más sencillo que hay, ya que en su contenido no hay otra cosa mas que el texto que lo conforma.

a) TXT

La extensión TXT representa "textfile" (archivo de texto), que sustituyó a su antiguo nombre "flatfile" (archivo sin formato). Este archivo informático estructura series de líneas de texto. El final del archivo se identifica habitualmente con un carácter especial definido como un marcador "end-of-file" (final de archivo), ubicado a continuación de la última línea de texto. Este tipo de contenedor incluye texto sin formato, pero tiene otras capacidades. Los archivos TXT sirven como almacenes de información a la vez que evitan las complicaciones propias de otros formatos de archivo. Los archivos afectados por la corrupción de datos son fácilmente recuperables y el usuario puede continuar su trabajo con la información restante. El inconveniente de utilizar archivos TXT proviene de su baja entropía, que provoca que los archivos TXT ocupen más espacio que otros archivos de texto.

b) MARKDOWN

Markdown es un lenguaje de marcado ligero creado por John Gruber que trata de conseguir la máxima legibilidad y facilidad de publicación tanto en su forma de entrada como de salida, inspirándose en muchas convenciones existentes para marcar mensajes de correo electrónico usando texto plano. Se distribuye bajo licencia BSD y se distribuye como plugin (o al menos está disponible) en diferentes sistemas de gestión de contenidos (CMS). Markdown convierte el texto marcado en documentos XHTML utilizando html2text creado por Aaron Swartz. Markdown fue implementado originariamente en Perl por Gruber, pero desde entonces ha sido traducido a multitud de lenguajes de programación, incluyendo PHP, Python, Ruby, Java y Common Lisp.

c) HTML

El Lenguaje de Marcado de Hipertexto (HTML) es el código que se utiliza para estructurar y desplegar una página web y sus contenidos. Es un lenguaje de marcado que define la estructura de tu contenido. HTML consiste en una serie de elementos que se usan para encerrar diferentes partes del contenido para que se vean o se comporten de una determinada manera. Las etiquetas de encierre pueden hacer de una palabra o una imagen un hipervínculo a otro sitio, se pueden cambiar palabras a cursiva, agrandar o achicar la letra, etc. Por ejemplo, sus contenidos podrían ser párrafos, una lista con viñetas, o imágenes y tablas de datos.

d) LATEX

Es un sistema de composición de textos, orientado a la creación de documentos escritos que presenten una alta calidad tipográfica. Por sus características y posibilidades, es usado de forma especialmente intensa en la generación de artículos y libros científicos que incluyen, entre otros elementos, expresiones matemáticas.

LaTeX está formado por un gran conjunto de macros de TeX, escrito por Leslie Lamport en 1984, con la intención de facilitar el uso del lenguaje de composición tipográfica, creado por Donald Knuth. Es muy utilizado para la composición de artículos académicos, tesis y libros técnicos, dado que la calidad tipográfica de los documentos realizados en LaTeX, se considera adecuada a las necesidades de una editorial científica de primera línea, muchas de las cuales ya lo emplean.

e) CSV

Un csv (comma-separated values) es un archivo de texto que almacena los datos en forma de columnas, separadas por coma y las filas se distinguen por saltos de línea.

El formato CSV no está estandarizado. La idea básica de separar los campos con una coma es muy clara, pero se vuelve complicada cuando el valor del campo también contiene comillas dobles o saltos de línea. Las implementaciones de CSV pueden no manejar esos datos, o usar comillas de otra clase para envolver el campo. Pero esto no resuelve el problema: algunos campos también necesitan embeber estas comillas, así que las implementaciones de CSV pueden incluir caracteres o secuencias de escape.

2. Editores de texto:

Un editor de texto es un programa que es capaz de editar texto plano. No confundir con los procesadores de texto

a) Atom.

Es un editor de código de fuente de código abierto para macOS, Linux, y Windows¹ con soporte para múltiples plug-in escritos en Node.js y control de versiones Git integrado, desarrollado por GitHub. Atom es una aplicación de escritorio construida utilizando tecnologías web.

b) BlueFish

Bluefish está dirigido a diseñadores web experimentados y programadores y se enfoca en la edición de páginas dinámicas e interactivas.

c) Brackets

Brackets es un editor de código fuente con un enfoque principal en el desarrollo web. Creado por Adobe Systems, es un software gratuito y de código abierto con licencia de la licencia MIT, y actualmente se mantiene en GitHub por Adobe y otros desarrolladores de código abierto.

d) Gedit

Es un editor de textos compatible con UTF-8 para GNU/Linux, macOS y Microsoft Windows. Diseñado como un editor de textos de propósito general, gedit enfatiza la simplicidad y facilidad de uso.

e) Geany

Geany es un editor de texto pequeño y ligero basado en Scintilla con características básicas de entorno de desarrollo integrado.

f) Emacs

Es un editor de texto con una gran cantidad de funciones, muy popular entre programadores y usuarios técnicos. GNU Emacs es parte del proyecto GNU y la versión más popular de Emacs con una gran actividad en su desarrollo. El manual de GNU Emacs lo describe como "un editor extensible, personalizable, auto-documentado y de tiempo real."

g) Nano

GNU nano es un editor de texto minimalista y amigable. Sin embargo, no solo nos permite editar texto, sino que además tiene otras características muy interesantes que lo hacen especialmente útil para modificar archivos de configuración en la terminal, crear lanzadores, y todo este tipo de acciones. Pero no solamente esto, puesto que al soportar coloreado de sintaxis, también puede ser utilizado para escribir código.

h) Pico

Pico (Pine composer) es un editor de texto para Unix y sistemas basados en Unix. Está integrado con el cliente de correo electrónico Pine. Fue diseñado por la Oficina de Computación y Comunicaciones de la Universidad de Washington.

i) Sublime Text

Sublime Text es un editor de texto y editor de código fuente está escrito en C++ y Python para los plugins. Desarrollado originalmente como una extensión de Vim, con el tiempo fue creando una identidad propia, por esto aún conserva un modo de edición tipo vi llamado Vintage mode.

j) Vim

Vim es una versión mejorada del editor de texto vi, presente en todos los sistemas UNIX. Su autor, Bram Moolenaar, presentó la primera versión en 1991. La principal característica tanto de Vim como de Vi consiste en que disponen de diferentes modos entre los que se alterna para realizar ciertas operaciones.

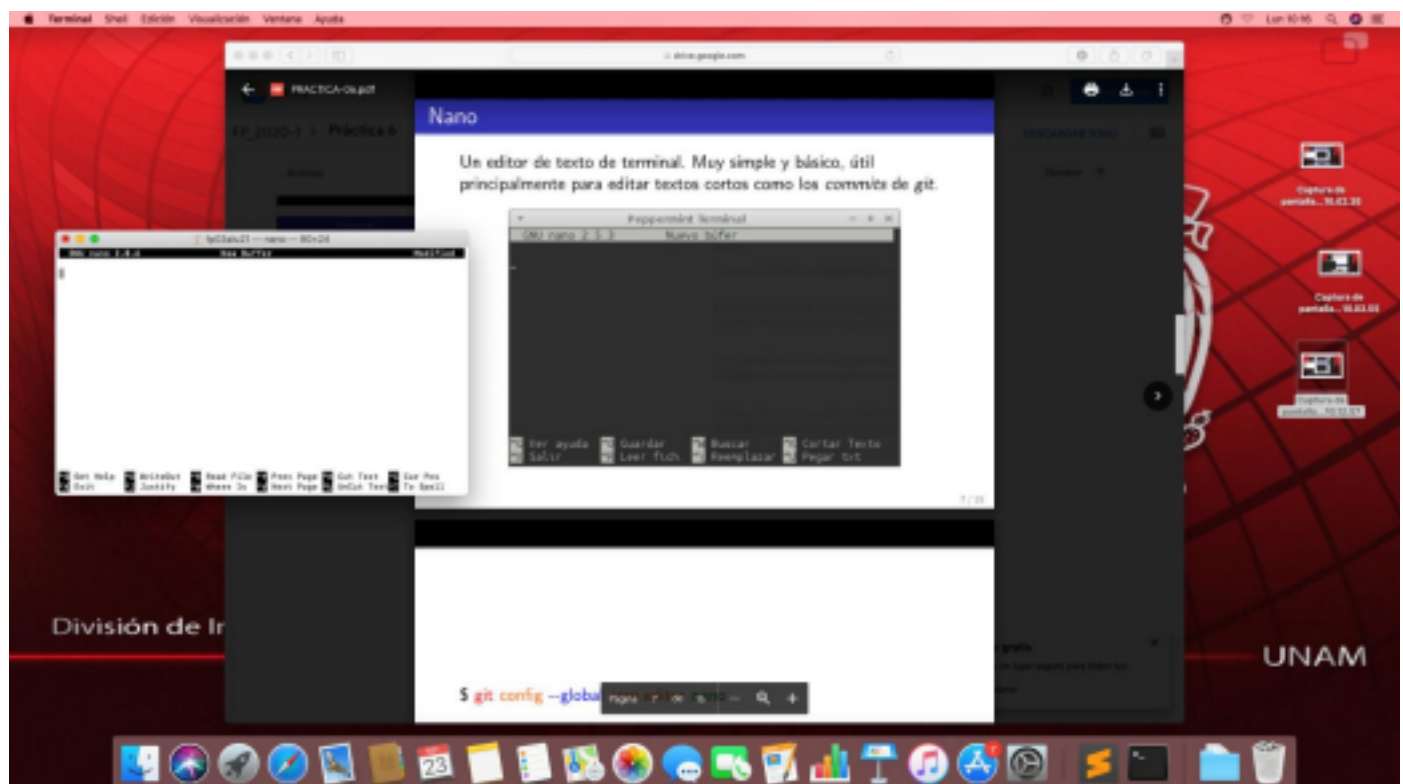
Editores

En general, los editores difieren en su modo de uso y en las características que ofrecen.

- Resaltado de palabras clave
- Autocompletado
- Lista de elementos definidos
- Autosangrado
- Identificación de pares de paréntesis
- Integración de compilador
- Integración de control de versiones
- Integración de terminal
- Búsquedas avanzadas
- etc.

3. Nano:

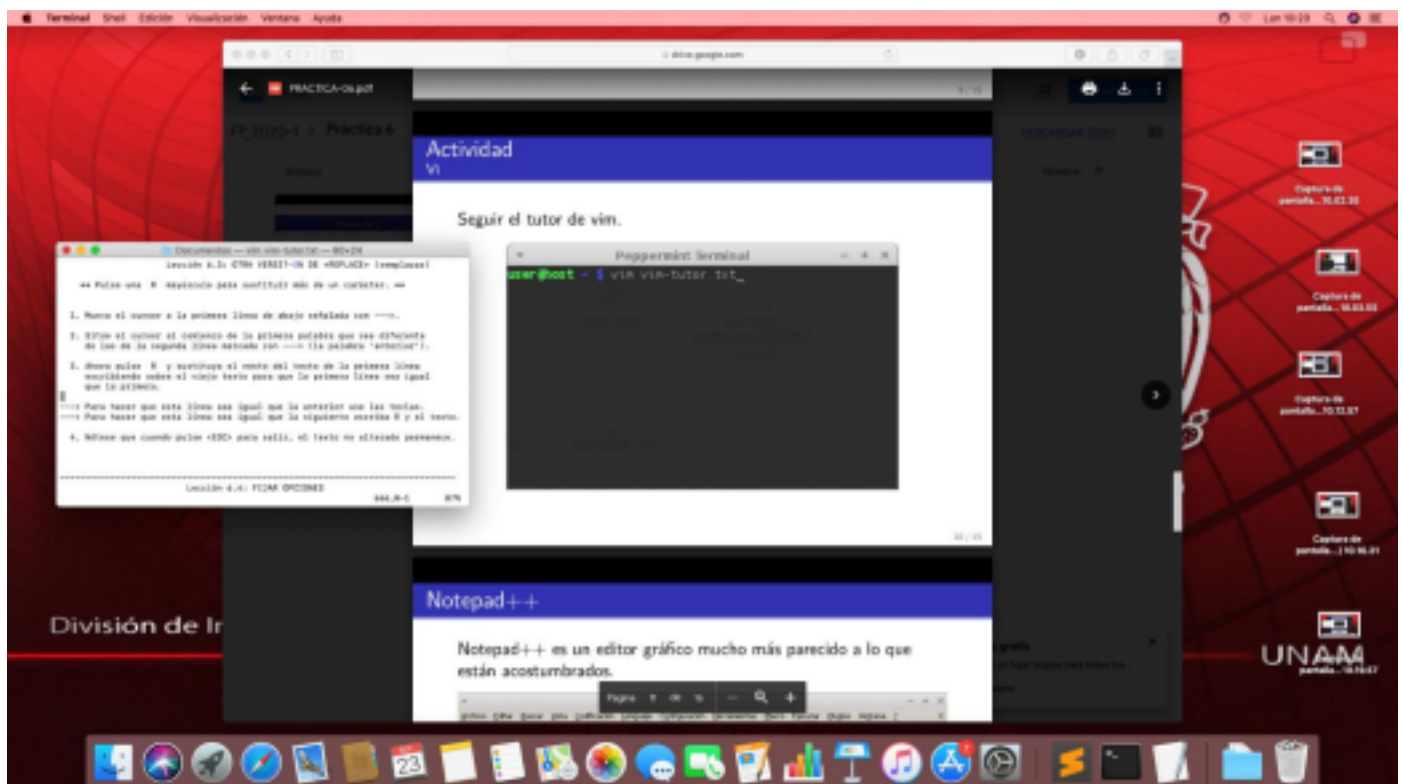
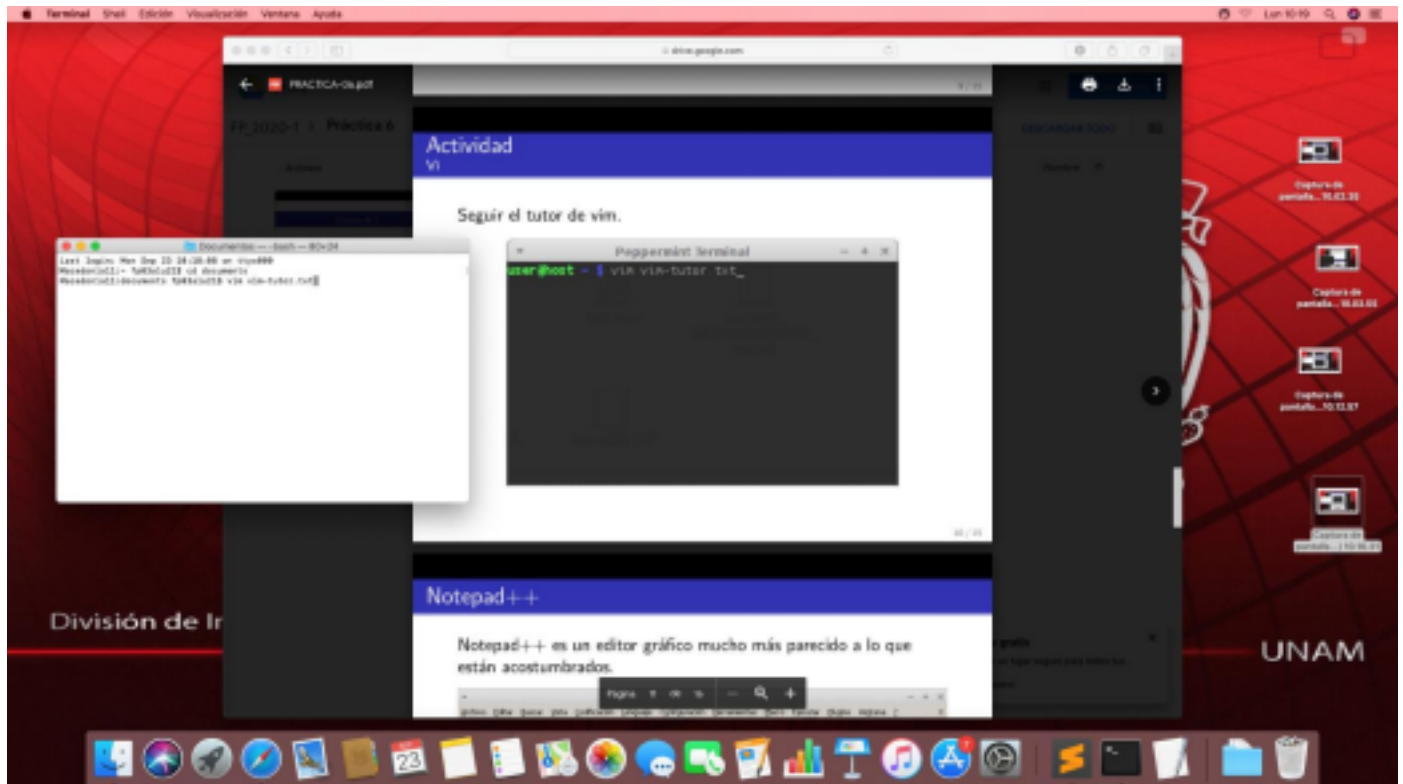
Un editor de texto de terminal. Muy simple y básico, útil principalmente para editar textos cortos como los *commits* de git.

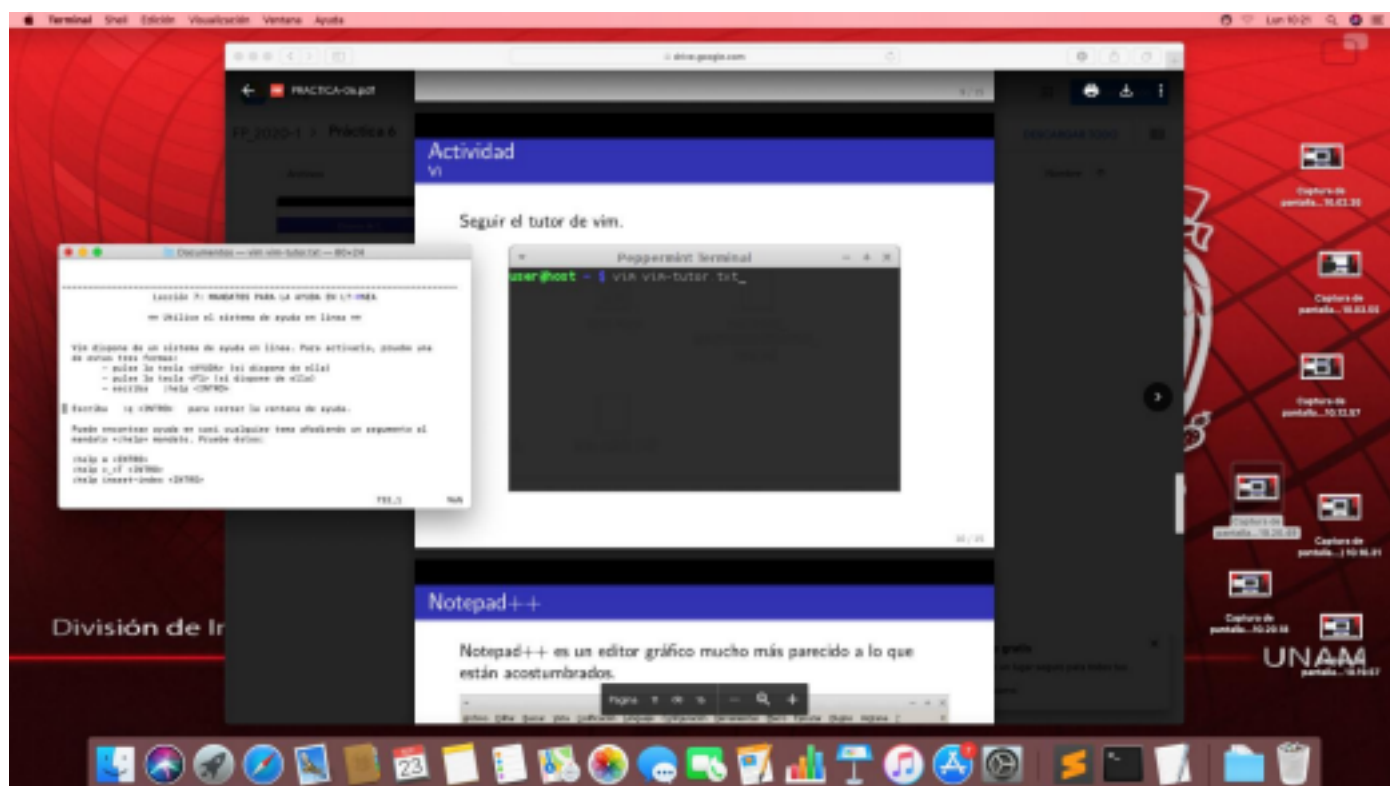
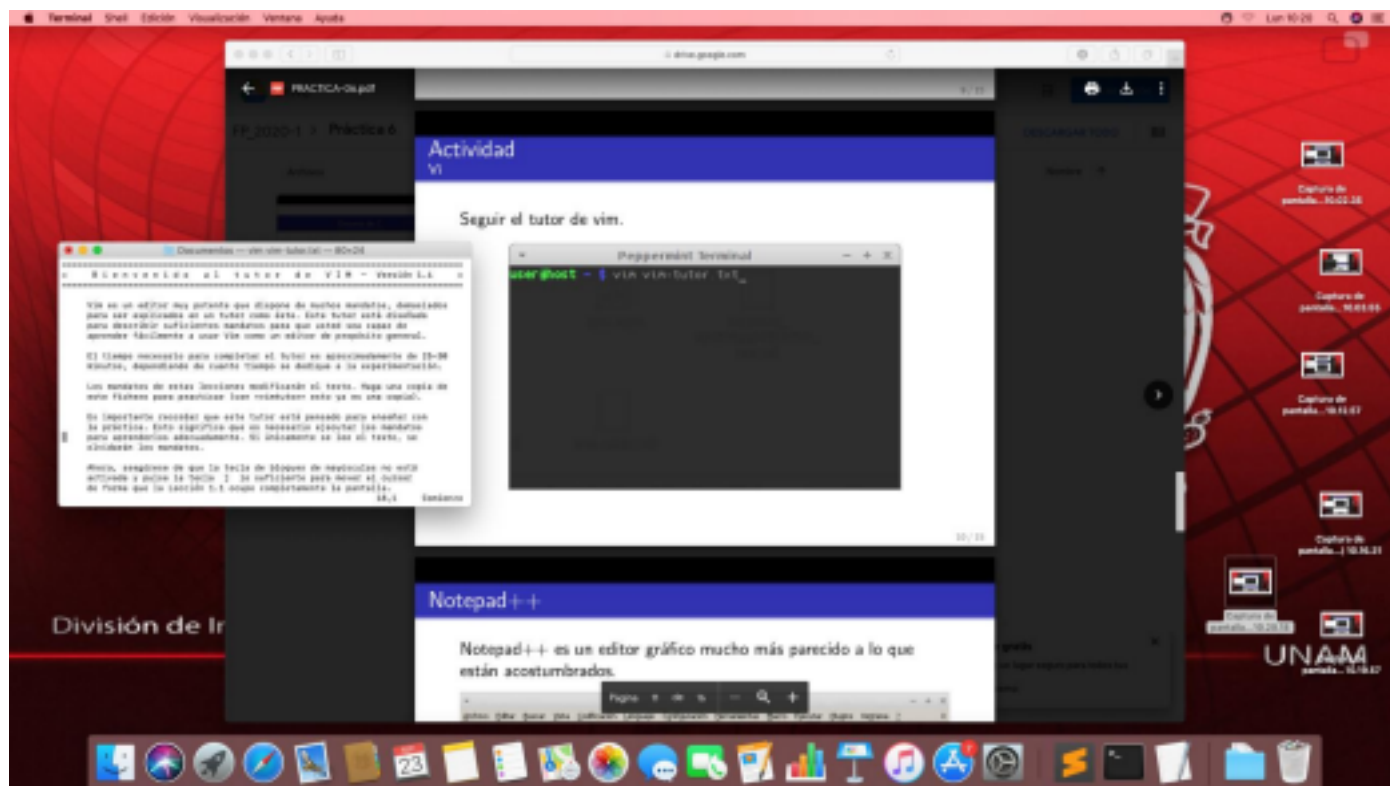


4. Vi:

Otro editor de terminal, aunque también tiene versiones gráficas. Este es uno de los editores más populares, con más características, y que los expertos aseguran que ofrecen mayor ganancia de velocidad para editar textos.

La razón es que este editor es bastante diferente a lo que están acostumbrados. Se requiere más tiempo para aprenderlo a usar y para dominarlo.

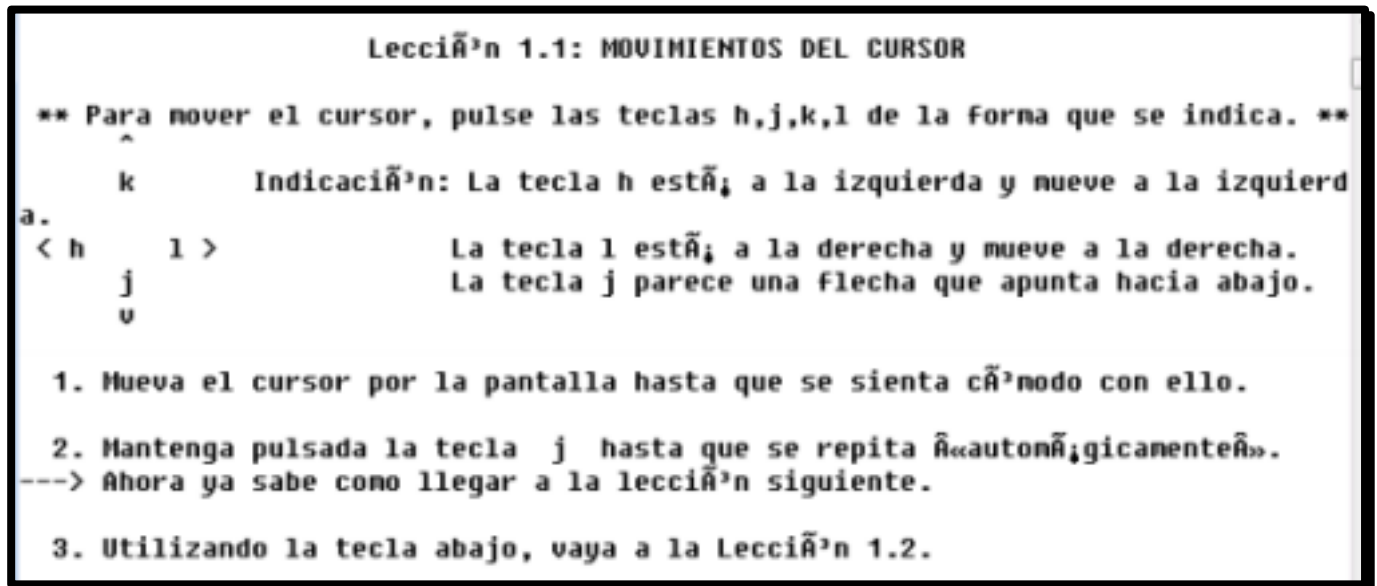




Seguir el tutor de vim:

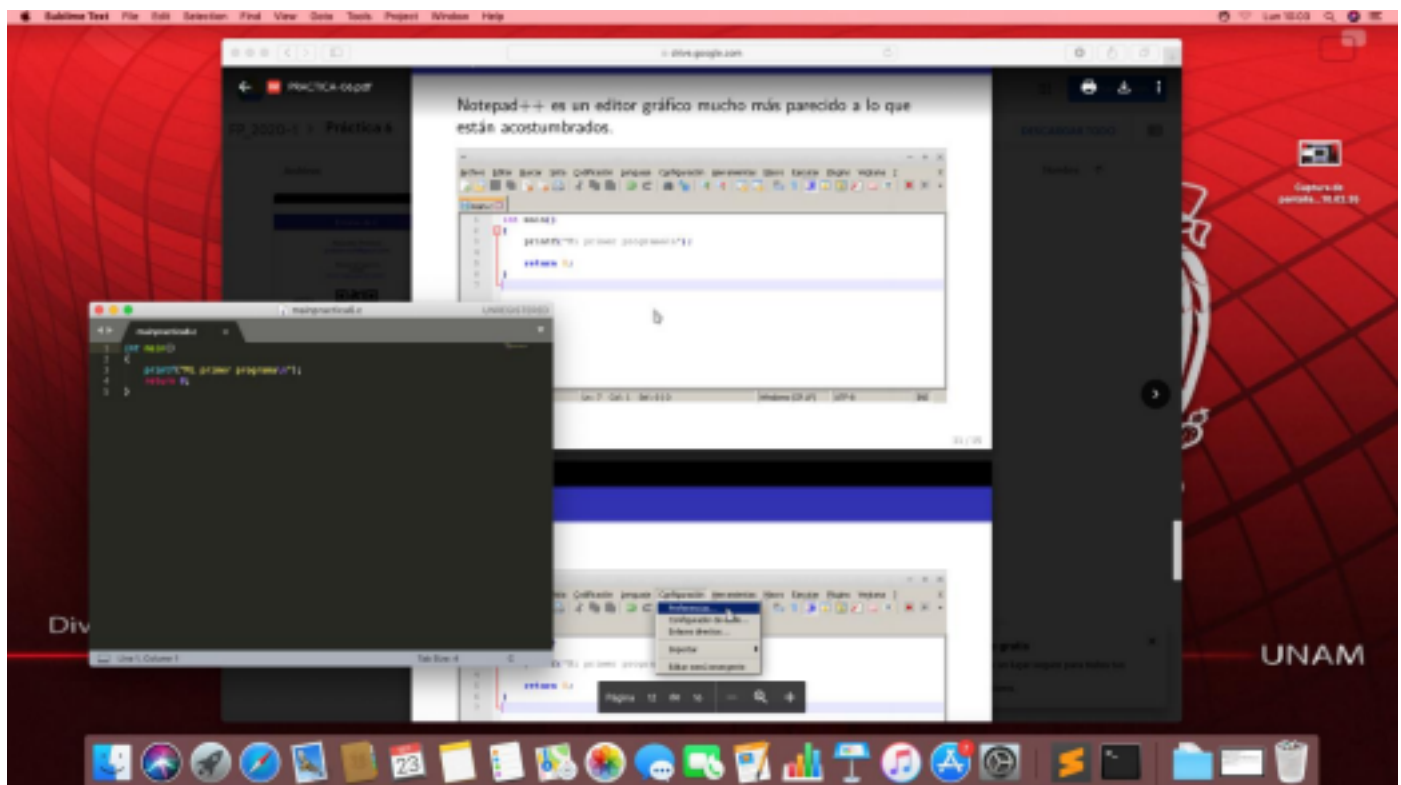
Me permitió hacer muchas aplicaciones para editar el texto, está diseñado para utilizar solamente el teclado y no el cursor.

Al principio me costó mucho trabajo mover solo con las letras J, K, L y H.

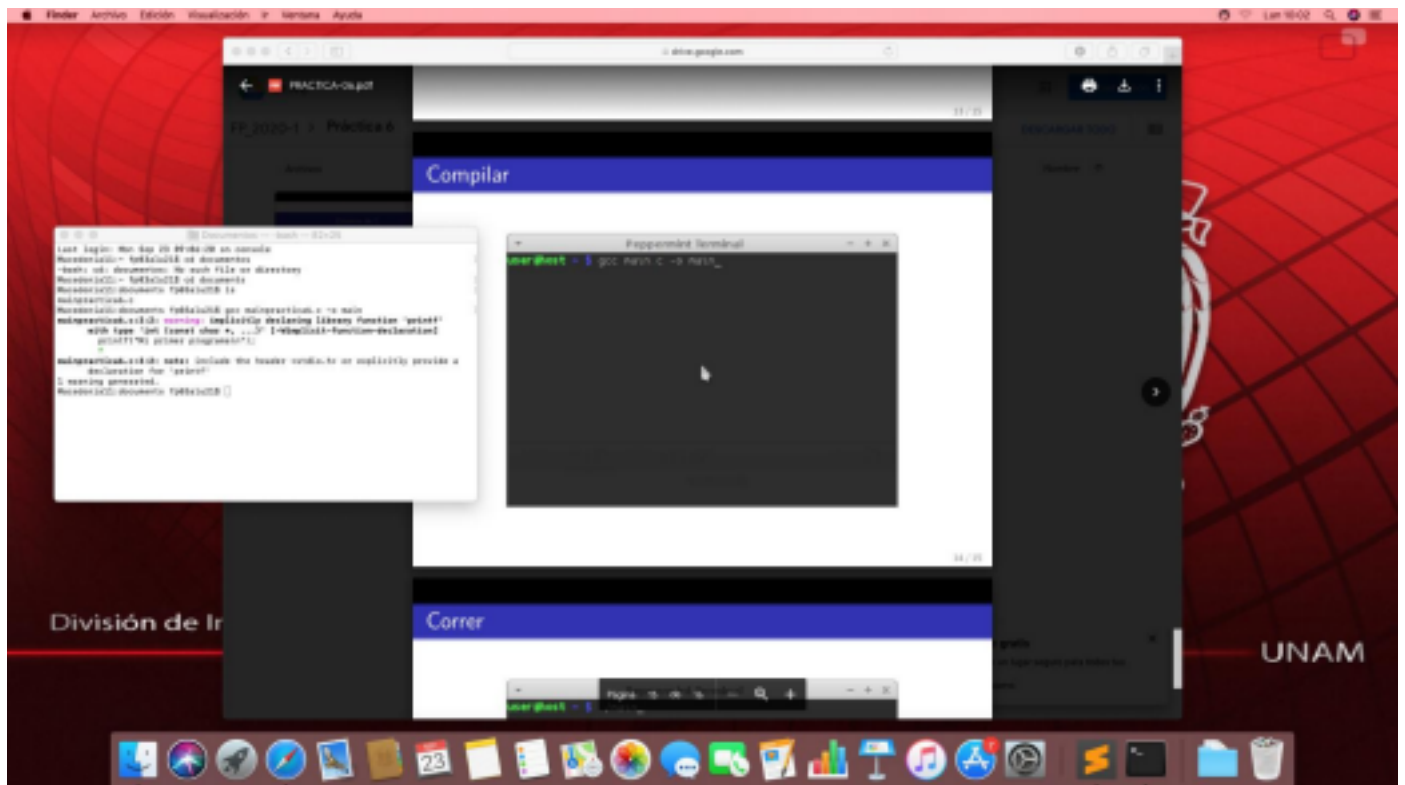


5. Notepad++:

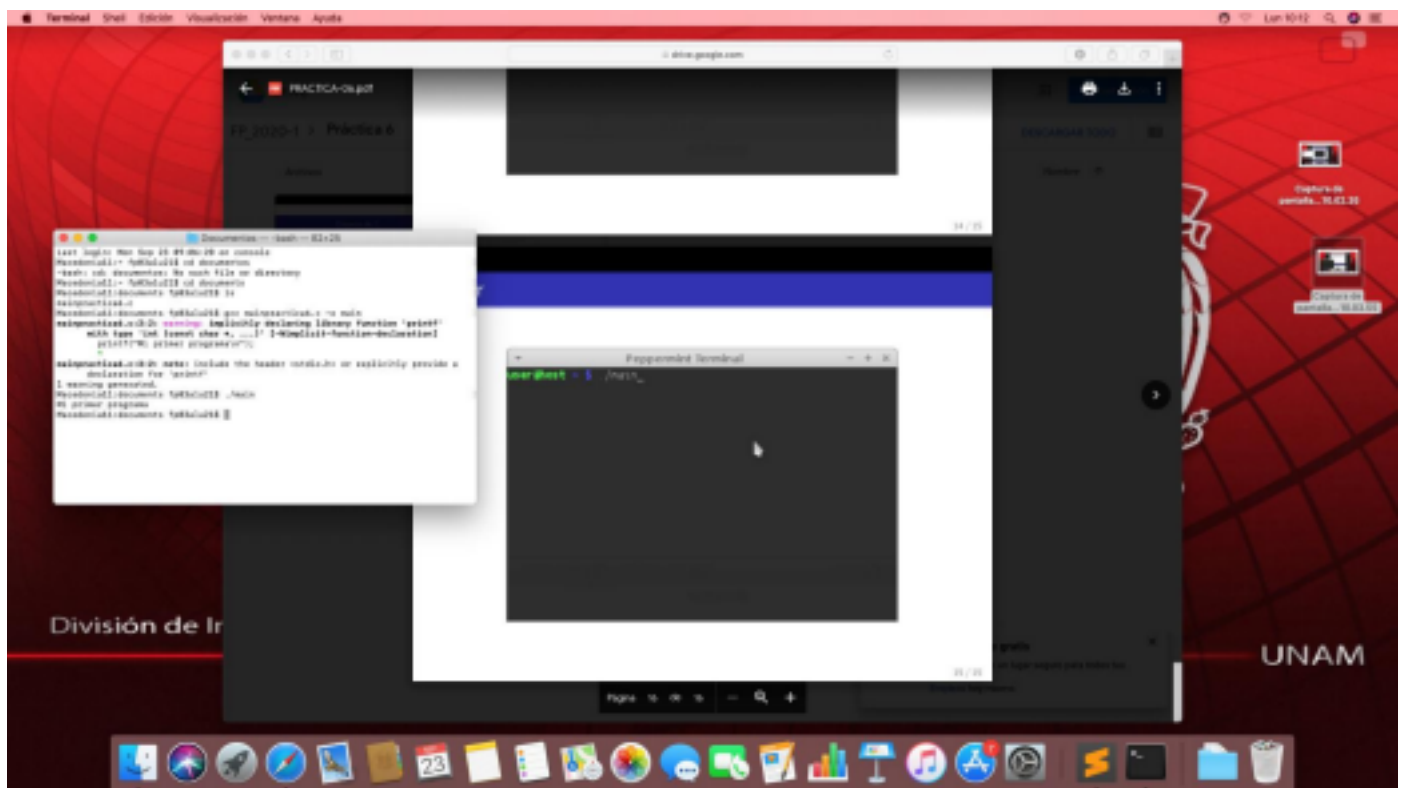
Notepad++ es un editor gráfico más parecido a lo que estamos acostumbrados.



Copilar:



Correr:



CONCLUSIÓN

En conclusión, C++ surge de fusionar dos ideas: la eficiencia del lenguaje C para poder acceder al hardware al ejecutar tareas que realmente demandaban recursos de memoria; y las ideas de abstracción que representan los nuevos conceptos de clases y objetos.