



Cómo convertirse en desarrollador en un proyecto de Software Libre (...y no morir en el intento)

6° Encuentro Linux
Iquique - 2005

Federico Mena Quintero <federico@novell.com>

Germán Poo Caamaño <gpoo@ubiobio.cl>



El verdadero nombre

Cómo convertirse en hacker (sin morir en el intento)

Federico Mena Quintero <federico@novell.com>
Germán Poo Caamaño <gpoo@ubiobio.cl>



Aclaración

El término hacker se refiere a un programador no a un vándalo en Internet o cibervandálico



¿Por qué estás aquí?

(respuestas que no se valen)

- “Para aprender”
- “Para ver qué dicen”



Cuestionamientos típicos

- ¿Cómo contribuir en un proyecto de Software Libre/Código Abierto?
- ¿Qué herramientas usar?
- ¿Qué libros leer?
- ¿Qué sitios visitar?
- ¿Dónde y cómo comenzar?



Dos respuestas

1. El camino largo
2. El camino corto



El camino largo



Misión de la carrera de Ingeniería Civil en Informática (UNAP)

- Profesional con una consistente base científica y tecnológica, altamente capacitado en las materias de ciencias de la computación, en lo que refiere a **análisis, diseño y modelamiento de sistemas de software, lo que permitirá analizar y sistematizar la información mediante el uso computacional.** de este modo le será posible desenvolverse exitosamente y liderar proyectos tecnológicos para la solución de problemas y necesidades informáticas de las empresas productivas y de servicios, tanto en el sector público como privado



Objetivos comunes en Informática

- "Formar profesionales propositivos, comprometidos con su superación y el avance de su disciplina, vinculados con las necesidades sociales de su entorno, **que comprendan y utilicen de manera analítica, crítica y creativa los diferentes modelos, enfoques, metodologías y herramientas de la computación, en la solución de problemas relacionados con el manejo de la información.**"



¿Cuál es tu objetivo?

¿Qué esperas obtener de la carrera?



Perfil del egresado (1)

- El egresado de una licenciatura de informática será capaz de:
 - Ofrecer soluciones a problemas relacionados con la computadora y el manejo automático de la información, tanto dentro como fuera de la empresa.
 - **Diseñar y elaborar programas y paquetes computacionales de servicio para usuarios especializados y no especializados.**



Perfil del egresado (2)

- El egresado de una licenciatura de informática será capaz de:
 - **Analizar, diseñar, actualizar y administrar sistemas de información**, desde los más sencillos hasta los que involucran bases de datos.
 - Capacitar a usuarios en la instalación y soporte a equipo de cómputo.
 - Contribuir a su formación a través de estudios de posgrado, de educación continua o **de manera autodidáctica**.



¿En qué quieres trabajar?

¿Tienes los conocimientos necesarios?



Asignaturas típicas

- Algoritmos I (primer semestre): **Conceptos básicos**, herramientas para desarrollar algoritmos, técnicas de diseño, proyectos sencillos.
- Algoritmos II (segundo semestre): Elementos de un lenguaje estructurado, pilas, recursión, listas ligadas, colas grafos, **árboles**, conjuntos.
- Algoritmos III (tercer semestre): Organizaciones básicas de archivos, métodos de ordenamiento, método de búsqueda.



Entrevista de trabajo (1)

- Saber aplicar los conocimientos:
 - **Escribe una función que invierta una cadena:**

“ABCDEFGH” -> “HGFEDCBA”



Entrevista de trabajo (2)

- Describe un árbol binario
 - **Escribe una función que inserte un número en un árbol binario de números.**



Entrevista de trabajo (3)

- Nivel de inglés.
 - By default, Glib assumes that file names on disk are in UTF-8 encoding. This is a valid assumption for file systems which were created relatively recently: most applications use UTF-8 encoding for their strings, and that is also what they use for the file names they create. However, older file systems may still contain file names created in "older" encodings, such as ISO-8859-1. In this case, for compatibility reasons, you may want to instruct Glib to use that particular encoding for file names rather than UTF-8.



¿Cómo piensas obtener los conocimientos necesarios?

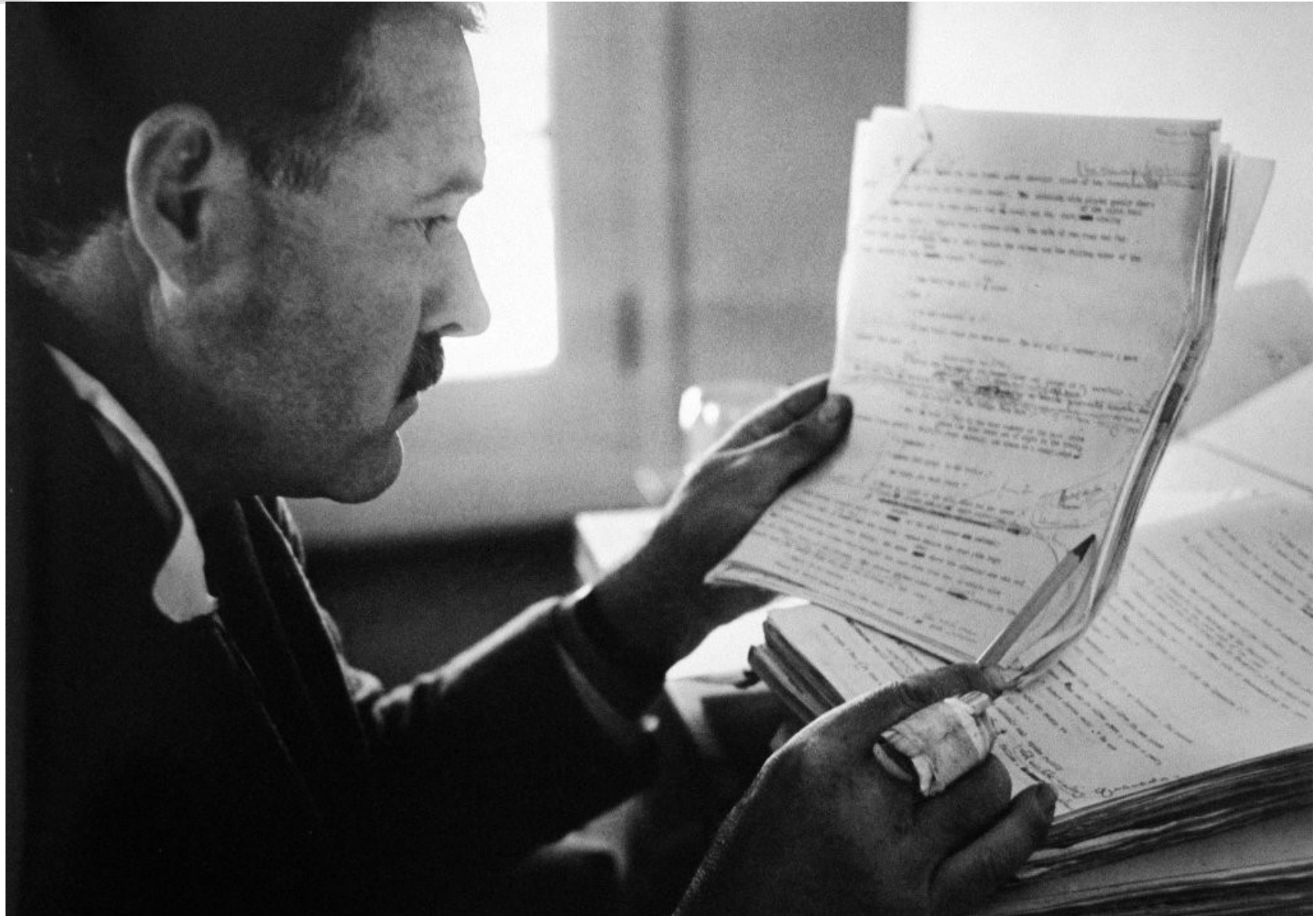


Para ser un buen programador

- Iniciativa propia
 - Interés por aprender
 - Buen nivel de inglés
- **Práctica,**
- **... Práctica,**
- **... Mucha práctica**



Ernest Hemingway





¿Cuánta práctica?

- Ernest Hemingway, foto de 1940
- Premio Nobel de Literatura 1954

- ***Las grandes obras no salen al primer intento.***



Software Libre

- Libertad de utilizarlo.
- Libertad de copiarlo.
- Libertad de modificarlo.
- Libertad de distribuir tus modificaciones.

- *Siempre y cuando no le quites ninguna de estas libertades.*



Utilizarlo

- Usalo; no tienes que pedirle permiso a nadie
- Puedes usarlo en tu casa, escuela o trabajo
- Puedes usarlo para fines que el autor no se esperaba



Copiarlo

- Dale una copia a tu mejor amigo.
- Pónselo a todas tus computadoras.
- Vende CDs con el software.



Modificarlo

- Examínalo
 - Abre el capó del auto
- Aprende cómo está hecho
 - Desarma un pedazo del motor
- Hazle cambios
 - ¿Qué pasa si enchufo esto aquí?
- Aprende de lo que hiciste
 - Y vuélvete un mecánico o programador experto



Distribuirlo

- ¿Le hiciste cambios interesantes?
- Véndelos / distribúyelos / cópialos
- **Hazlos parte del bien común.**



Libertad de aprender (1)

- El software propietario es una caja negra





Libertad de aprender (2)

- El software libre es transparente

```
using System;  
  
public class Principal {  
    public static void Main ()  
    {  
        Console.WriteLine ("Hola, Mundo");  
    }  
}
```



Es como comprar un coche

- El propietario tiene el capó soldado
- El libre lo puedes destripar a tu gusto.



Ponte a leer código ajeno

- Aprende de programas reales
- Modifícalo
- Límpialo
- Aprende técnicas nuevas



Aplica la teoría

¿De qué te sirve si no sabes
cuándo utilizarla?



Contribuye a un proyecto ya existente

- Arregla lo que está roto
 - **Son sólo programas.**
- Aprende de programadores experimentados.



El camino corto



El camino corto: Pasos

1. Descargar el código fuente del programa que se desea modificar
2. Desempaquetar (tar/gzip)
3. Construir y ejecutar la aplicación en forma exitosa
(leer los archivos de instrucciones correspondientes, tales como README, INSTALL)
4. Abrir el código fuente en un editor
5. Encontrar las partes del código que se necesita modificar para hacer que el programa haga lo que uno quiera
6. Aplicar los cambios y probar hasta estar seguros que el programa funciona correctamente.
7. Ejecutar el comando "*diff -u*" y enviar un correo de la salida a la lista de correo correspondiente.



Repetir los pasos anteriores
todas las veces que sea necesario



Consideraciones generales

- Si no hay programas que se deseen modificar:
 - Tal vez no quieras ser hacker después de todo
 - Tal vez no hayas utilizado suficientemente el software en cuestión
 - Tal vez este año sólo seas usuario del software y no hay cosas que quieras modificar
- Es imposible saltarse los pasos 1 al 4.
- Si no tienes el código fuente en el computador, ¿podrías llegar a ser hacker?
 - Al menos nadie lo ha logrado así hasta el momento (Nat Friedman).



Paso 3: construir y ejecutar la aplicación en forma exitosa

- El paso 3 puede requerir varias iteraciones antes de lograrlo con éxito.
 - Puede resultar ridículo y humillante
 - Algunas veces puede ser el paso que toma más tiempo y ser el más difícil.



Paso 5: Encontrar las partes del código que se necesita modificar para hacer que el programa haga lo que uno quiera

- Al llegar al paso 5:
 - Podría llegar a ser tan fácil como bucear en los archivos del código fuente y buscar una cadena relevante
 - En la interfaz
 - En alguna alguna salida común del programa
- Requisitos
 - Invertir tiempo averiguando la disposición y “formato” del código
 - Puede valer añadir algunos *printf* para mostrar el área correcta en la cual trabajar.
 - Puede ser útil disponer de un depurador para aprender el comportamiento interno del programa.



Mientras más experiencia se alcanza
el paso 5 se va tornando más fácil.



Si el software en el cual estás trabajando fue escrito por un programador con mucha experiencia el paso 5 será mucho más fácil,

Los programadores con experiencia realizan esfuerzos extras para escribir código que sea fácil de entender.



Paso 6: Aplicar los cambios y probar hasta estar seguros que el programa funciona correctamente

Es lo que comúnmente se conoce como
"hacking"

Pero no siempre es la parte
que toma más tiempo.



Muchas personas no llegan a este punto.

Si lo has hecho,
con toda seguridad puedes considerarte
un "**hacker**".



Pero, es muy probable que:

No escribirás buen código
sino hasta que hayas programado
una gran cantidad.



La clave para ser un **hacker** es llegar al punto donde estás "**hackeando**".

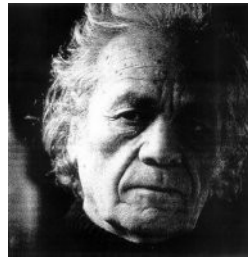


Nunca es tarde para comenzar



Nunca es tarde para comenzar

- Violeta Parra comenzó su carrera artística a los 35 años



- Nicanor Parra escribió su primer libro en 1937
 - Su segundo libro, “Poemas y antipoemas” en 1954 (a los 40 años)
 - El 2004, a los 90 años, celebró sus 50 años de antipoeta



Referencias

- Artículos de Joel Spolsky
 - Joel on Software, <http://www.joelonsoftware.com/>
 - “The Guerrilla Guide to Interviewing”, <http://www.joelonsoftware.com/articles/fog0000000073.html>
- Nicanor Parra <http://www.antiweb.cl/>
- Blog de Nat Friedman, <http://www.nat.org/>



Gracias por asistir