

# PrebeShell



## Objetivo

El presente proyecto pretende que demuestren lo aprendido durante el curso que recientemente se les impartió, así como también su capacidad de investigación, creatividad e inventiva propia. Para el desarrollo de este proyecto necesitarán conocimientos sólidos del sistema operativo GNU/Linux, del manejo de archivos y sobre conocimientos de scripting en ShellScript.

## Especificaciones

El proyecto consiste en desarrollar una terminal de trabajo “prebeshell”. Su programa deberá proporcionar al usuario una interfaz eficaz para manejar archivos, teclear distintos comandos, consultar información del sistema, reproducir música y jugar distintos juegos programados por ustedes. Esta terminal debe contener:

- Un sistema de acceso (logueo) para los usuarios: al ejecutar su programa debe mostrar una pantalla que permita a los usuarios actuales del sistema operativo anfitrión acceder o loguearse a su terminal. Se debe pedir el nombre de usuario y contraseña, y sólo con la combinación correcta se permitirá el acceso. Dichos usuarios y contraseñas deben de existir en el sistema operativo anfitrión, no es válido que algún usuario real del sistema no se pueda loguear a su terminal.
- La interacción con el usuario será por medio de una línea de prompt (intérprete de comandos). Esta línea de prompt puede mostrar la información que quieran y lucir como ustedes gusten, pero debe de mostrar al menos el nombre del usuario activo y la carpeta en la que están trabajando actualmente. Esta terminal deberá interpretar los comandos que ustedes vayan a programar (especificados en los siguientes puntos) y también los del sistema operativo anfitrión, teniendo como prioridad, obviamente, los suyos. La única manera de salir de esta terminal es escribiendo el comando “salir”. No podrá salirse de la terminal presionando ctrl+c ó ctrl+z.
- Se requiere que se programen los siguientes comandos:
  - Comando “árbol”: simula la salida del comando tree, es decir, muestra de manera gráfica la jerarquía de un directorio que ustedes le especificaron al comando. Si al comando no se le especifica directorio, el comando se ejecuta sobre el directorio actual.
  - Se necesita implementar un comando “ayuda” que dé al usuario información acerca de los comandos que ustedes mismos hayan desarrollado y con respecto al uso de su Terminal.
  - Dos comandos distintos que den fecha y hora, Respectivamente.

- Un comando que busque un archivo en específico en un directorio en particular. Reciba dos argumentos: el archivo y el directorio a buscar.
- Un comando que muestre los créditos de los programadores.
- Van a programar un juego que se pueda jugar en la terminal, de forma textual. Se recomienda que sea gato (de dos jugadores), pero pueden programar los que ustedes gusten, mientras sea del mismo nivel de complejidad que el juego antes mencionado. La interfaz de estos juegos lucirá como ustedes gusten. No se permite el juego del ahorcado.
- Van a hacer un reproductor mp3 llamado “prebeplayer”. La interfaz gráfica de este reproductor es de diseño libre, aunque debe incluir al menos una manera para navegar sobre la biblioteca musical, un área que muestre el status actual de la reproducción y una sección que indique al usuario cómo utilizar el programa.
- Una interfaz para hacer tareas programadas con crontab el cual le pida al usuario los parámetros de minutos, horas, días, etc. y el comando, acción o script que quiera ejecutar.

## Especificaciones extra

- Su script debe correr sobre el shell bash.
- Para hacer el prebeplayer, ustedes requerirán del programa mpg123, y harán la programación de esta parte basados en comandos para mpg123. Sin embargo, al principio de la ejecución de la terminal, esta debe revisar si el programa está instalado y realizar la instalación si es que el usuario no lo tiene. Si está instalado, simplemente no se muestra nada al inicio de la ejecución de la terminal. Extra
- En todo momento se deberá proporcionar una línea editable para trabajar. Es decir, el usuario debe de poder usar las flechas izquierda y derecha del teclado para poder editar lo que escribió. Extra
- La funcionalidad es lo más importante, pero el diseño importa mucho y se contará para la calificación final. Por eso, es necesario que su interfaz contenga letras de colores.
- Pueden ejecutar su script principal como super-usuario, pero se otorgará un punto extra a quien pueda hacer toda la funcionalidad del script sin necesidad de ser super-usuario.
- Es MUY recomendado que tengan cada uno de sus comandos programados en un archivo .sh distinto. Sobre todo por el prebeplayer y los juegos. Debe estar documentado.
- Para aquellos programas que así lo requieran deberán manejar archivos de configuración, por ejemplo, para el reproductor musical se necesita que en este archivo de configuración se indique la carpeta donde el usuario tenga su música.
- Todo su sistema deberá ser capaz de trabajar con archivos que lleven espacios en sus nombres, las canciones por ejemplo, un formato similar a este: “/home/usuario/Musica/The Beatles/Abbey Road/01 Come Together.mp3”. Extra

**Requisitos a evaluar:**

- Este proyecto se realizará en equipo, como fueron asignados.
- No pueden utilizar comandos que estén por obvias razones no permitidos, por ejemplo, no usarán tree para su implementación de “arbol” ni otro shell para la implementación de su intérprete de comandos. Los comandos y juegos deben estar programados en shellscrip, no se permite lanzar un archivo compilado de c con un juego, o cosas por el estilo.
- La fecha límite de entrega del proyecto es 22-Febrero-2019, se entregará en GitHub y se revisará que los commits se hayan realizado por línea de comandos. El Último commit solo puede ser hasta las 11:00 del 22 de Febrero del 2019
- La revisión personal a cada equipo será sobre un GNU/LINUX que está basado en Debian.
- Como una ayuda especial ¡Piensen sencillo! Todos los comandos que les solicitamos son bastante fáciles, si sienten que alguno de ellos está difícil simplemente cambien su enfoque. Google es su mejor amigo, aprendan a comunicarse con el.