

## **Reporte Biculturalism**

Andrés Martínez Martínez

Joel Spolsky en su documento titulado Biculturalism, publicado hace poco más de 15 años, hace alusión al libro “The Art of UNIX Programming”, de Eric S. Raymond, y a su propia experiencia como desarrollador para exponer más allá de tecnicismos y conceptos informáticos las diferencias entre el desarrollo de aplicaciones con sistemas/ideología UNIX y Windows.

La principal diferencia según el documento entre ambos “estilos” de programación, es el público al que están dirigidos. UNIX enfocándose al desarrollo de aplicaciones para otros programadores, mientras que Windows lo hace para los no-programadores, es decir la mayor parte de la población.

Por otro lado, las aplicaciones para sistemas UNIX suelen ser desarrolladas primeramente para su utilización mediante la terminal y/o línea de comandos, al contrario que en Windows donde su implementación suele ser realizada teniendo en cuenta en todo momento el aspecto gráfico del programa. Esto suele ser así debido a la historia de las dos vertientes, por un lado UNIX fue realizado en una época en la que el público general no tenía acceso a las computadoras, lo que permitía a los desarrolladores el implementar solamente lo necesario para el funcionamiento del sistema en cuestión. Mientras que el objetivo de Windows fue vender la mayor cantidad posible de equipo informático, esto significaba que su utilización debía ser lo más sencilla posible.

No estoy muy familiarizado aún con aplicaciones UNIX, pero sí estoy de acuerdo con lo que plantea Spolsky, pues si alguien sabe lo que es un sistema UNIX y lo ha utilizado alguna vez en su vida es casi un hecho que por lo menos sabe lo que es una terminal. Por el contrario, el usuario promedio de Windows no tiene ni idea de que su máquina viene con una aplicación llamada cmd la cuál permite manejar la computadora mediante comandos y muy seguramente no podría visualizar una aplicación si no viene acompañada de una interfaz gráfica. Además comparto la visión de Spolsky al criticar la postura de “superioridad” que toma Raymond, ya que

ambas vertientes son aceptables y no podríamos tener una visión de la computación como la tenemos ahora si no fuera por la existencia de ellas.