Resumen del Mítico Mes-Hombre

El libro del siguiente resumen se encuentra en inglés, está almacenado con el nombre de “[Mítico Mes-Hombre](../Libros%20de%20Diseño%20de%20Software/Mítico%20Mes-Hombre.pdf)” en la carpeta de libros de Diseño de Software.

El resumen será desarrollado en español y posteriormente en una actualización futura se añadirá su versión en inglés. Además, los títulos de los capítulos, así como los de las subdivisiones creadas estarán en inglés para mayor facilidad al buscar en el texto original. Finalmente, se identificará con referencias, por número de página de la lectura original, los contenidos resumidos con la intención de localizar dicha información.

Contenido

[Capítulo 1: The Tar Pit 1](#_Toc39609350)

[1.1 The Programming Systems Product 2](#_Toc39609351)

[1.1.1 Programming Product 2](#_Toc39609352)

[1.1.2 Programming System 3](#_Toc39609353)

[1.2 Joys of the Craft 3](#_Toc39609354)

[1.3 Woes of the Craft 4](#_Toc39609355)

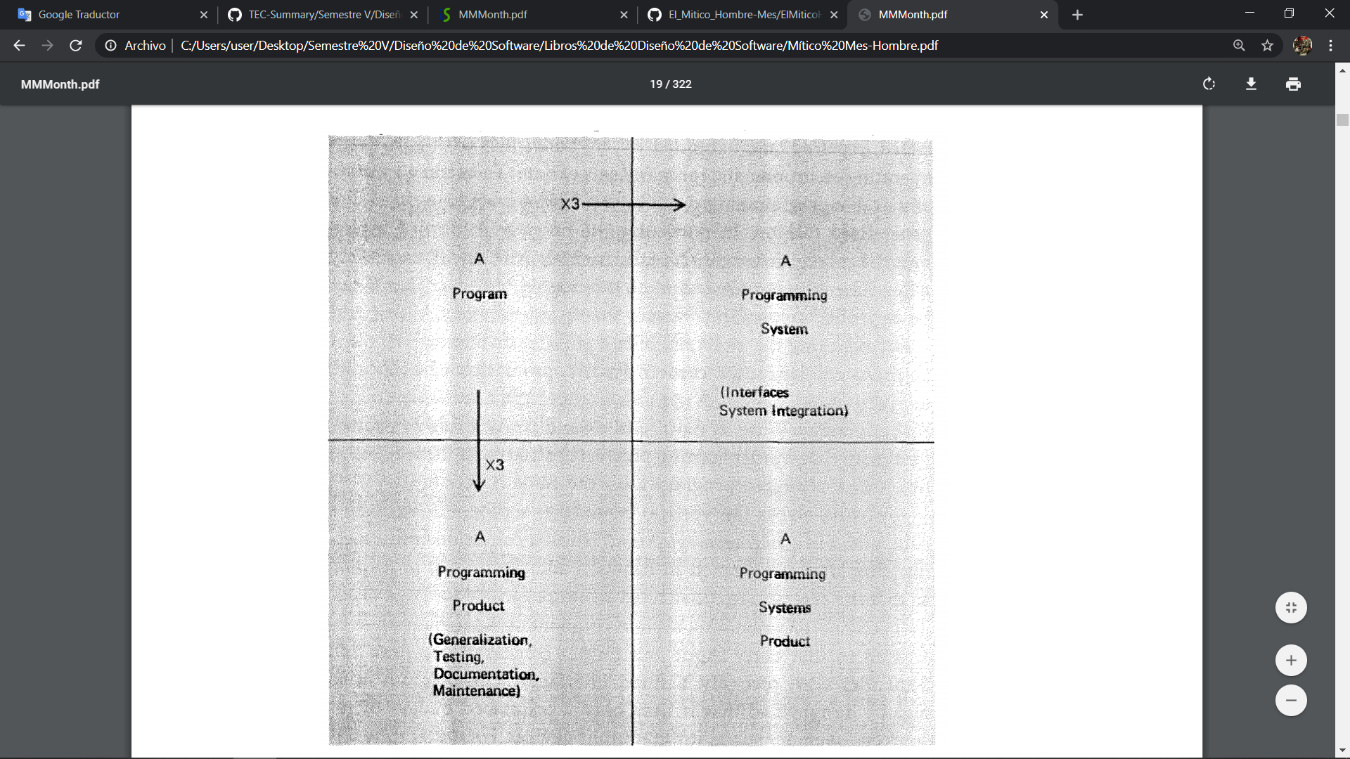
# Capítulo 1: The Tar Pit[[1]](#footnote-1)

La programación de sistemas grandes ha resultado en la creación de muchos sistemas funcionales, pero los cuales muy pocos cumplen con las metas, presupuestos y horarios. Sin importar las características de los equipos de trabajo todos se han visto involucrados en problemas al realizar esta tarea. La acumulación de tareas simultaneas y los factores que interactúan ralentizan el avance.

Para comprender porque sucede esta problemática debemos identificar:

* [El oficio de la programación de sistemas](#_1.1.1_The_Programming)
* Las satisfacciones
* Los problemas

## 1.1 The Programming Systems Product[[2]](#footnote-2)



Los programas tienen dos posibles caminos para convertirse en un software más útil, pero más costoso:

* Producto de programación
* Sistema de programación

### Programming Product[[3]](#footnote-3)

Este tipo de programa puede correrse, probarse, repararse y ser extendido por cualquiera. Es utilizado por diversos ambientes operativos, para muchos grupos de datos. Para convertirse en un producto generalizado por medio de la generalización del rango y forma de las entradas lo más que sea permitido por el algoritmo base. Posteriormente debe ser probado con un banco de casos de prueba. Finalmente, requiere documentarse para permitir su uso, arreglo o extensión. El costo aproximado crece al triple comparativamente a un programa depurado.

### Programming System[[4]](#footnote-4)

Es una colección de programas interactuando de forma coordinada en su funcionamiento y disciplinada en su formato, para que el ensamblaje constituya una instalación completa para tareas grandes. Las características que requiere un programa para ser componente de un sistema son:

* Las entradas y las salidas deben ajustarse de forma sintáctica y semántica con interfaces bien definidas.
* Solo debe utilizar una cantidad de recursos predefinida (memoria, dispositivos de entrada-salida, tiempo computacional).
* Funcionar correctamente con todos los componentes del sistema, probando cada combinación posible

El costo aproximado requerido es el triple para realizar un componente del sistema de programación comparativamente con un programa autónomo.

Los productos de los sistemas de programación costarían alrededor de unas nueve veces más.

## 1.2 Joys of the Craft[[5]](#footnote-5)

Los principales motivos para disfrutar del proceso de creación de un programa son:

* La pura alegría de hacer cosas.
* El placer de hacer cosas que son útiles para otras personas
* La fascinación de crear objetos complejos, como rompecabezas, de partes móviles entrelazadas y verlas realizar su trabajo en ciclos sutiles, generando las consecuencias de los principios incorporados desde el comienzo.
* La alegría de aprender siempre, el cual surge de la naturaleza no repetitiva de la tarea.
* Es un deleite trabajar en un medio tan manejable.

En conclusión, programar es divertido porque resulta gratificante el anhelo de creatividad inmerso en nosotros y deleita los sentimientos que tenemos en común.

## 1.3 Woes of the Craft[[6]](#footnote-6)

Una parte importante del proceso de creación es conocer los inconvenientes para poder enfrentarlos con mayor facilidad. Los motivos para esto son los siguientes:

* El deber de actuar de manera perfecta. El programador debe ajustarse de forma perfecta al requerimiento.
* Otras personas nos asignan objetivos, nos brindan recursos y nos facilitan la información.
* La dependencia de otros. Es decir, depender de los programas realizados por otras personas que tengan algún fallo o sean difíciles de comprender.
* La búsqueda minuciosa de algún problema (bug) resulta ser tedioso.
* La depuración puede complicar enormemente en la fase de pruebas al tener los bugs más difíciles de resolver y con probabilidad de incremento al intentar corregirlos.
* El producto que se ha trabajado parece ser obsoleto al acercarse a la completitud del mismo.

Los productos que se realizan poseen una implementación que debe medir su obsolescencia con otros ya existentes. El reto y la misión son encontrar una solución real a los problemas con el cronograma actual y los recursos disponibles.

1. Página 17 del PDF, 3 del libro [↑](#footnote-ref-1)
2. Página 19 del PDF, 5 del libro [↑](#footnote-ref-2)
3. Página 19 del PDF, 5 del libro [↑](#footnote-ref-3)
4. Página 20 del PDF, 6 del libro [↑](#footnote-ref-4)
5. Página 21 del PDF, 7 del libro [↑](#footnote-ref-5)
6. Página 22 del PDF, 8 del libro [↑](#footnote-ref-6)