Futoshiki Versión 2

Valeria Chinchilla Mejias

Escuela de Computación, Instituto Tecnológico de Costa Rica

IC1803: Taller de programación

William Mata Rodríguez

17 agosto, 2020

# **Tabla de Contenido**

# **Enunciado del Proyecto**

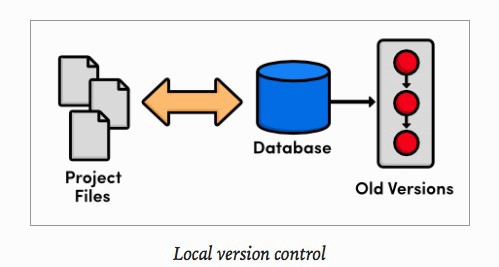
Este proyecto esta destinado a practicar una de las principales actividades que consumen más tiempo en la ingeniería de software: el mantenimiento de software. Además de esto se agregan nuevas funcionalidades al juego Futoshiki que se desarrollo anteriormente. El objetivo principal de este proyecto es mejorar el funcionamiento de este juego y de esta manera ofrecer un mejor producto.

# **Marco Teórico**

El mantenimiento de software puede ser definido como la modificación de un producto que ya fue entregado ya sea para corregir errores, mejorar el rendimiento o entre otros. Esta es una de las actividades que se utilizan día a día en la ingeniería en software. Muchas personas creen que el mantenimiento de software es el simple hecho de corregir errores. No obstante, este es comúnmente utilizado para realizar mejoras a la funcionalidad de un producto. Así de esta manera se puede llevar un control sobre los problemas que puedan llegar a surgir durante la utilización de este producto.

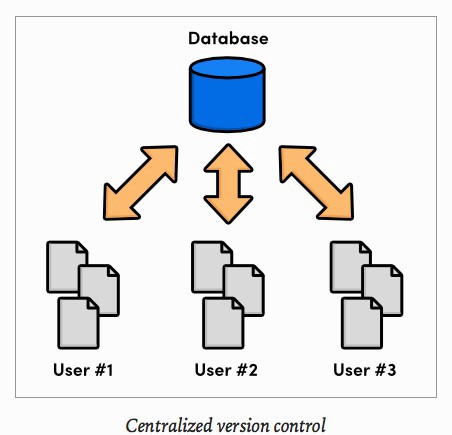
Se dice que la vida de un desarrollador de software se divide en un antes y en un después de conocer los sistemas de control de versiones. El control de versiones es un sistema que permite registrar todos los cambios que se han hecho a un archivo a lo largo del tiempo, de manera que se pueda acceder a cualquiera de las versiones específicas. A lo largo del tiempo estos tipos de sistemas han evolucionado y en la actualidad existen tres tipos de estos sistemas: Locales, Centralizados y Distribuidos. (Medium, 2017)

El sistema de control de versiones locales almacena todas las versiones de un programa en una base de datos. De esta manera en cualquier momento se tiene una sola copia del proyecto, para así eliminar la posibilidad de confundir o deshacerse de algunas versiones. En este control de todas las versiones se llevaban a cabo en el computador de cada desarrollador por lo que no había manera de compartir el código con otras personas. En la siguiente imagen se representa un sistema de control de versiones locales. (Medium, 2017)



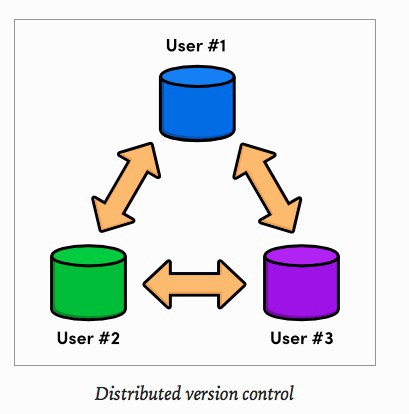
*Figura 1.* Recuperado de: <https://medium.com/@jointdeveloper/sistemas-de-control-de-versiones-qu%C3%A9-son-y-por-qu%C3%A9-amarlos-24b6957e716e>

Por otro lado, se tiene sistemas de control de versiones centralizados. Para evitar el problema que se tenia de no poder compartir programas con otros desarrolladores se evolucionaron los sistemas creando así los centralizados, estos en vez de almacenar los cambios y versiones en el disco duro de los desarrolladores, se comenzaron a almacenar en un servidor. A continuación, una figura con este sistema de control de versiones:



*Figura 2.* Recuperado de: <https://medium.com/@jointdeveloper/sistemas-de-control-de-versiones-qu%C3%A9-son-y-por-qu%C3%A9-amarlos-24b6957e716e>

Por último, se tiene el sistema de control de versiones distribuidos. En este tipo se decidió darle una copia del archivo a cada desarrollador una copia local de todo el proyecto. De esta manera se construyó una red distribuida de repositorios, en la que cada desarrollador podía trabajar de manera aislada. (Medium, 2017) A continuación, una imagen de este sistema:



*Figura 3.* Recuperado de: <https://medium.com/@jointdeveloper/sistemas-de-control-de-versiones-qu%C3%A9-son-y-por-qu%C3%A9-amarlos-24b6957e716e>

Los sistemas de control de versiones representan una gran importancia a la hora de diseñar proyectos que son difíciles de controlar y gestionar. Es importante contar con uno de ellos debido a que estos sistemas les permiten a varias personas trabajar en un solo proyecto, facilitando de esta forma el trabajo en equipo para los desarrolladores. El principal beneficio de ellos es que se puede tener trazabilidad de todos los cambios y mejoras realizados. Esto es de suma importancia debido a que, si se realiza un cambio y este genera problemas en el proyecto, se va a tener un respaldo o una versión anterior que no haya sido alterada.

El control de versiones usado para el presente proyecto es GIT.

La solución para resolver las partidas del juego Futoshiki se obtuvo con ayuda del algoritmo de Backtracking o también conocido en español como vuelta atrás. El Backtracking puede ser definido como un algoritmo general que sirve para obtener soluciones para problemas computacionales. Esta solución la obtiene construyéndola gradualmente con posibles candidatos y abandona todos aquellos que no cumplan con las restricciones dadas. Este algoritmo descarta candidatos retrocediendo de manera que retrocede hasta que encuentra una solución adecuada. Al hablar de retroceder se refiera a volver a un punto anterior para probar alternativas que si logren adaptarse a las restricciones. (Iacono, 2017)

Un ejemplo para tener mas claro el funcionamiento del Backtracking es un laberinto. Cuando se llega a un lugar con varios caminos se prueba con una dirección. Si este camino era el correctos entonces el problema fue resuelto. Sin embargo, si este no era el camino correcto se debe retroceder al cruce de camino y elegir otra opción. Este proceso se repite las veces que sea necesario. En caso de que se agoten las opciones eso significaría que no hay solución o salida. El Backtracking es un algoritmo utilizado por miles de usuarios para encontrar soluciones principalmente a Sudokus y otros juegos con la misma lógica que este. (Microsiervos, 2017)

# **Conclusiones**

A continuación, se listan algunos de los problemas encontrados durante la realización del proyecto:

* Uno de los principales obstáculos en la realización del proyecto fue aprender a usar el software de control de versiones ya que durante clases no se estudio nada relacionado con ello.
* Otro problema fue exportar datos de Python y lograr convertir esto a un pdf para así lograr que se imprimieran
* Por otro lado, el Backtracking fue uno de los principales problemas ya que el algoritmo que se utilizó solo funcionaba con la validación de restricciones de que los números no se podían repetir en las filas ni en las columnas
* La falta de motivación fue otro de los problemas enfrentados durante la realización del presente trabajo. Sin embargo, después se logro continuar.
* El realizar una opción de multinivel represento una dificultad, esto se resolvió con prosperidad y paciencia.

# **Estadísticas de tiempo**

|  |  |
| --- | --- |
| **Actividad Realizada** | **Horas** |
| Análisis del problema | 1h 10 min |
| Diseño de algoritmo | 2h 00 min |
| Investigación de | 2h 00 min |
| Programación | 3h 00 min |
| Elaboración del manual de usuario | 30 min |
| Elaboración de documentación del proyecto | 1h 47min |
| **TOTAL** | **8h 17 min** |

# **Lista de revisión de proyecto y análisis de resultados**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Concepto** | **Puntos** | **Puntos**  **Obtenidos** | **Avance**  **100 / % / 0** | **Análisis de resultados** |
| Rehacer Jugada | 10 |  | 100 |  |
| Solucionar Juego | 35 |  |  |  |
| Juegos multinivel (8pts por cada nivel) | 24 |  |  |  |
| Opción de configuración | 1 |  | 100 |  |
| Posibles jugadas para una  casilla | 10 |  | 100 |  |
| Impresión de información | 5 |  |  |  |
| Uso de software de control de versiones | 10 |  |  |  |
| Ayuda: el programa debe  desplegar el manual de  usuario. | 5 |  | 100 |  |
| **TOTAL** | **100** |  |  |  |

# **Bibliografía**