

Tarea 1 - Pruebas de Software

Integrantes: Iñaki Oyarzun M. (Enlace a repositorio) - Jorge Salas L.(Enlace a repositorio)
Profesor: Oscar Reyes H.

Septiembre 2022

1 Sobre el documento

El presente entregable tiene por finalidad, recopilar los desarrollos realizados de manera personal y conjunta para llevar a cabo las metodologías de Validación y Verificación a partir del desarrollo de un programa y especificación de requerimientos. Se organiza en un primer lugar una etapa individual para luego pasar a una conjunta con las pruebas cruzadas. Se adjunta al costado de los nombres de los integrantes el enlace al código correspondiente dentro de los repositorios respectivos, incluyendo archivo de programa, logs, excel de pruebas y readme del programa.

2 Desarrollo Individual

A partir de los siguientes requerimientos para un programa en línea de comandos:

- 1 Que permita almacenar un texto en una pila
- 2 Que dé al usuario la posibilidad de:
 - 2.1 Ver cuál es el texto más largo en la pila y el más corto.
 - 2.2 Imprimir cualquier texto de la pila en cualquier momento.
 - 2.3 Permitir comparar tamaños.

Se debe llevar a cabo un análisis para determinar una mejor manera en la que se podrían especificar los requerimientos, a la vez de determinar maneras en las que se podrían asegurar el cumplimiento del requerimiento. Se procede a indicar por cada integrante dichas resoluciones.

2.1 Iñaki Oyarzun M.:

2.1.1 Mejoras al requerimiento

Se identifica una necesidad de una mayor especificación por parte de los requerimientos, en diversos aspectos, con tal de buscar o explicar esta mejora buscada se presenta un punteo de acuerdo a los puntos y sub puntos de los requerimientos originales (se marcan en negrita los agregados) de esta forma se visualizan de mejor manera las dudas y las mejoras que se buscarían realizar:

- 1 Que permita almacenar texto en una pila
 - 1.1 **Especificar el tamaño de la pila:**
¿Cuántos textos debiese almacenar la pila?. Se estima un tamaño de 10 textos (identificables con espacios del 0 al 9)
 - 1.2 **La forma de almacenaje:**
¿Cómo debe almacenarse esta pila? (txt, ods, etc.). Se estima que una vez se termina el programa, guarda en formato txt sin codificar, y que una vez se inicia el programa recupera dicha pila (independiente si se cambia algo dentro/fuera del archivo o no, debe leer lo que esté si existe).
Mostrando el mensaje:
[Notificación] Pila recuperada
o [Notificación] Archivo de pila creado
o cada vez que se hace un cambio en la pila [Notificación] Pila almacenada en archivo

1.3 El tipo de texto a almacenar:

¿Los textos serán en un solo lenguaje?. Se estiman que los textos serán realizados con letras de la A-Z, números del 1-9 y espacios, sin contar acentos, tildes, comas, puntos, entre otros.

1.4 Los caracteres a permitir:

¿Se ingresarán o almacenarán textos en ascii o UTF-8?. Se asume que como es en txt, se realizara en UTF-8 y bajo esos caracteres.

2 Que dé al usuario la posibilidad de:

2.1 Ver cual es el texto mas largo de la pila y el mas corto:

Especificar el tipo del formato:¿Cantidad de caracteres o según el espacio que usan en memoria?. Se interpreta según cantidad de caracteres que conforman el texto. ¿Qué pasa si 2 o más textos son iguales en tamaños mayores o menores?. Se estima que mostrara uno de los textos que comparten ser los mas grandes o mas pequeños, en caso de estar vacía la pila mostrará vacío y para los espacios vacíos serán ignorados en la pila para esta contabilización.

2.2 Imprimir cualquier texto de la pila en cualquier momento:

Especificar a qué se refiere con "cualquier momento":Se asume que se refiere a que existirá una forma de acceder a esta función a partir de un menú de selección.

Como se imprimirá el texto: Se asume que será en el siguiente formato:

El texto número X es: [texto]

Con una cantidad de Y caracteres.

2.3 Permitir comparar tamaños:

¿Cómo será realizada esta función?: ¿Comparará tamaño en memoria o cantidad de caracteres?,¿Como se presentará este resultado?. Se asume que será comparando el texto más grande en cantidad de caracteres, presentándolo en la forma:

Texto X es mayor que el texto Y en Z cantidad de caracteres

2.4 Almacenado y creación de textos :¿Cómo se espera que se almacenen los textos?: Se estima que será a partir de la opción de añadir texto, y con este, pedir el ingreso del texto para almacenarlo en la ultima posición de la pila, y cuando se desee quitar un texto, será con una opción llamada quitar texto, la cual, vaciará la ultima ranura de la pila. Indicando que en caso de estar llena la pila, se consulte si desea reemplazar el ultimo espacio de la pila por el nuevo texto.

2.5 Forma de acceso a estas funcionalidades:

¿Cómo se hará el acceso a estas funciones?. Se asume que será por medio de una interfaz o menú principal, el cual permita acceder a estas funciones con palabras restringidas para recorrer dicha interfaz.

2.6 El tamaño de estos textos: Serán de un largo fijo o varían según algun parámetro?, en este caso se interpreta que se pide un texto de máximo 250 caracteres, cortando la palabra a 250 caracteres si sobrepasa el límite.

2.1.2 Asegurar el cumplimiento del requerimiento

Con tal de ser consistente en el cumplimiento del requerimiento, se cree que la mejor forma de realizar el cumplimiento es indicar casos borde en las pruebas los cuales puedan provocar eventuales fallas en el programa, de esta forma se puede saber si pueden cumplir completamente el requerimiento, para ello, se especifican las siguientes formas en las que se puede asegurar su cumplimiento (enumerando según punto indicado en el ítem anterior):

1.1 Para este caso, la forma en la que se puede asegurar el cumplimiento será a partir de 2 test, el primero consta de realizar un intento por agregar 11 textos, lo cual se esperaría que en vez de agregarlo con la ubicación 11, no debiese permitirlo y obligaría a sobre escribir alguno de los textos anteriormente ingresados (demostrando que se pueden almacenar los textos y a la vez mantener el tamaño) La segunda prueba consta de generar un archivo de guardado y a partir de este ingresar un 11vo texto, con el fin de ver si el tamaño de la pila crece, el resultado esperado es que la pila ignore dicho texto extra.

1.2 La forma de probar este requerimiento consta de 3 pruebas, la primera corresponde a generar un archivo de guardado, para luego cambiar su terminación y nombre, el resultado esperado es que el programa

vuelva a crear un archivo en blanco, ignorando el que fue cambiado de terminación o nombre. La segunda es realizar la incorporación de un texto de prueba para luego modificarlo dentro del archivo de guardado, debiendo poder leerlo igualmente (siempre y cuando cumpla con ser válido, sino se omite). Para la tercera prueba, corresponde a crear un archivo de guardado, almacenar un texto y luego de la notificación revisar si efectivamente fue almacenado.

- 1.3 Para este caso, se realizará una sola prueba de agregar un texto a la pila que incluya los caracteres no permitidos, en este caso el programa debiese mostrar el resultado del almacenaje de la palabra recortando aquellos caracteres no válidos.
- 1.4 Al igual que la prueba anterior, se buscará ingresar un texto para luego verificar por medio del txt creado en el formato UTF-8 si es que el texto sigue siendo el mismo.
- 2.1 Para esta prueba se puede proponer un caso borde de ingresar 2 textos con tamaños diferentes y verificar si efectivamente al realizar esta operación se indica según el formato que se espera. Por otra parte se puede hacer el ingreso de 2 textos de tamaños iguales y verificar si el programa puede mostrar los textos que son de igual tamaño para los mas grandes y los mas pequeños, también se puede probar lo ultimo mencionado con ejecutar esta función con la pila vacía y ver que se cumple lo indicado por el requerimiento, finalmente se puede probar con ingresar un solo texto, el cual debiese ser tomado como mayor y menor a la vez, ignorando los espacios vacíos de la pila.
- 2.2 En este caso, para la prueba a realizar, constará de almacenar un texto en un espacio de la pila, para luego hacer uso de esta función, debiendo mostrar el tamaño y el mismo texto. Por otra parte, para otro caso borde, sera consultar el tamaño en un espacio vacío de la pila, con tal de verificar que este puede indicar que no hay texto almacenado allí. Por otra parte, una prueba extra sería el corroborar que desde el menú principal se puede acceder a esta funcionalidad luego de hacer cualquier otra anteriormente.
- 2.3 Para verificar este requerimiento se estiman como casos borde el evaluar los casos en que se compare un espacio vacío con una palabra ingresada, que se comparen 2 espacios vacíos y el caso no borde que sería 2 palabras conocidas o palabras de igual tamaño.
- 2.4 Para este requerimiento, el caso borde que debiese ser evaluado sería corroborar que el programa en caso de querer agregar mas de 10 textos se cumpla esa sobre escritura, y en caso de querer borrar un texto de una pila vacía, se indique que ya esta vacía la pila. Y finalmente se tiene el caso no borde que sería querer agregar y quitar un texto de la pila.
- 2.5 La forma en que se podría evaluar o verificar este requerimiento sería con probar a lo largo de las pruebas anteriores si es que el menú carga correctamente durante las pruebas, controlando igualmente casos en los que se ingresen inputs del menú o acceso a funcionalidades erróneos.
- 2.6 La forma en que se podría validar este requerimiento creado constaría de evaluar el caso límite en el que se ingrese una cantidad de texto mayor a los 250 caracteres, donde se esperaría que la palabra fuese a ser recortada para aceptarla.

Luego de esta explicación y análisis se realizará un planteamiento de un plan de ejecución de pruebas, con tal de poder realizarlo de manera individual y luego unificarlo con el de mi compañero.

2.2 Jorge Salas:

2.2.1 Mejoras al requerimiento

En definitiva, el requerimiento entregado en el enunciado contiene asociado un gran número de vacíos y ambigüedades que dejan poco claro como proceder a armar un programa que satisfaga los requerimientos a cabalidad. De esa forma, es importante analizar cada uno de los requerimientos de forma específica para enfocar bien los esfuerzos a resolver el problema.

1. "Que permita almacenar texto en una pila"

Esta sección me recuerda a la discusión sostenida en la primera clase, en la cual se mostraba el sencillo método `lowerify` de una presunta clase `String`. Para este método o tarea a realizar se desprendían muchas preguntas que también aplican al ámbito de esta tarea.

- 1.1. Según los requerimientos que se mencionan posteriormente, da la sensación de que los 'textos' corresponderán a cada una de las palabras que estén separadas por un espacio. De esta forma, el tamaño de la pila será fijado para almacenar, a lo más, 20 textos. Por otro lado, se permitirá que el usuario pueda interactuar con una interfaz en la terminal para que el usuario tenga control total de lo que necesita que el programa realice. Necesidades tales como imprimir textos, ingresar textos, comparar textos o incluso eliminar textos (conforme a la lógica de una pila).
- 1.2. ¿El texto a almacenar estará escrito solamente en el alfabeto latino que empleamos? Es importante conocer si es que es necesario dar soporte a caracteres especiales presentes en otros idiomas o si es necesario dar soporte a los símbolos que permiten la comunicación en países asiáticos. Por ahora solamente se permitirá ingresar textos con caracteres a-z, A-Z, 0-9 y el espacio como delimitador. Respecto a esto ultimo, si se juntan múltiples espacios, se contarán como uno solo. Por otra parte, si los textos contienen caracteres de puntuación, estos desaparecerán uniendo los textos que habían alrededor del caracter eliminado.
2. "Que dé al usuario la posibilidad de:"
 1. "(...) ver cuál es el texto más largo en la pila y el más corto."

El tamaño de un texto, en términos computacionales, puede ser entendido de la forma más intuitiva, que es la cantidad de caracteres que posee, o también se puede entender como cuanto espacio este texto ocupa en memoria. Se escoge responder este requerimiento basado en la cantidad de caracteres que posee el texto. En caso de haber más de un texto que comparta longitud extrema (superior o inferior), se mostrará solamente uno de ellos.
 2. "(...) imprimir cualquier texto de la pila en cualquier momento."

Este requerimiento no implica mayor complicación en ser implementado ya que es un requerimiento muy directo. Sin embargo, esta funcionalidad se expandió a la posibilidad tanto de imprimir la totalidad de la pila como de un solo texto de la pila. Debería significar que el usuario siempre puede invocar esta funcionalidad tal vez por medio de una interfaz básica en la terminal, e indicar si quiere que se imprima toda la pila o que indique qué texto quiere que se imprima, pensando en que ingrese un número entre el 1 y el índice más alto en ese momento.
 3. "(...) permitir comparar tamaños."

Se interpretará este requerimiento solicitando las posiciones de los textos a comparar e indicando (en términos de caracteres, tal como en el requerimiento 2.1) en cuanto sobrepasa el texto más largo al mas corto. También asociado al tamaño de los textos ingresados a la pila, estos serán de un tamaño máximo de 100 caracteres (diferentes del espacio) para no contar con textos excesivamente largos, pensando en que la pila ya aceptará 20 textos como máximo.

2.2.2 Asegurar el cumplimiento del requerimiento

En base a los análisis realizados en el apartado anterior, se consideran las siguientes pruebas para cada uno de los requerimientos con el fin de probar todas las formas en las que el programa pudiera fallar.

- 1.1 Para verificar este requerimiento se efectuarán tres pruebas. Una de ellas es simplemente el ejecutar el programa para visualizar que la interfaz se despliegue de forma correcta y que esta misma le entregue la posibilidad de terminar con la ejecución del programa. La segunda prueba corresponderá a entregar un parrafo de más de 20 palabras de manera que el programa debería truncar dicho parrafo y solamente incluir los primeros 20 textos y deshacerse del resto. La tercera prueba, relacionada con la segunda apunta a agregar textos a una pila no vacía, de manera que se pueda llenar la pila con los elementos que faltan y nuevamente desechar los elementos que no caben.
- 1.2 Para verificar este requerimiento también se efectuarán dos pruebas. La primera es probar ingresar textos separados con más de un espacio y que el programa pueda gestionar esos grandes saltos como uno solo y que continúe su ejecución, ya que el caracter de espacio actuará siempre como delimitador. La segunda prueba comprende el entregarle textos al programa cuyo cuerpo contenga caracteres de puntuación, los cuales se deberían gestionar de manera tal que al momento de agregarlo a la cola, dichos símbolos de puntuación desaparezcan.
- 2.1 Este requerimiento puede ser verificado por tres pruebas, siendo la primera, la más común, que es cuando el texto más largo y más corto son distintos entre si. En este caso, si hay más de un texto

que comparta longitud, el programa debería mostrar solamente el primer texto con longitud máxima o mínima. La segunda prueba correspondería a cuando solamente se encuentran palabras de igual longitud, pero distintas entre si. Para esta situación, tal como la primera, solamente se mostrará el primer texto con dicha longitud y se indicará que todos los textos tienen la misma longitud en la pila en dicho instante. La tercera prueba permitirá distinguir el caso en el que la pila contiene solamente una palabra y el programa debería indicar que solamente hay una palabra en la pila y corresponde al texto más largo y al más corto de forma simultánea.

- 2.2 El segundo requerimiento de este apartado se puede separar en el caso de imprimir la pila completa y un único texto de la pila a elección. Esto, sin embargo nos lleva a hacer tres pruebas para verificar el cumplimiento del requerimiento. Es necesario probar el caso en el que no hayan textos, por lo que el programa debería indicar que la pila se encuentra vacía, otro caso en el que no tiene sentido la opción de impresión total o parcial que es cuando se encuentra un solo texto en la pila y el tercer caso en el que cuando hay más de un texto, el programa le da a elegir al usuario que tipo de impresión de la pila desea. Tanto para este requerimiento, como para el manejo de la interfaz y como para los índices para comparar tamaños del requerimiento siguiente, el programa debería alertar al usuario cuando este ingrese un valor que se encuentre fuera de rango o sea inválido.
- 2.3 Para el requerimiento de comparar tamaños de textos, las pruebas son semejantes al requerimiento anterior, pero este requerimiento igual necesita más pruebas. Si la pila no tiene elementos, una prueba de este programa debería indicar que no existen textos para comparar. Además, cuando la pila tiene un solo elemento, el programa debería indicar que al existir un solo texto en la pila, no existe otro texto con el cual compararlo, por lo que no tiene sentido la ejecución de ese procedimiento. Finalmente, dentro del caso en el que hay más de dos palabras, se pueden ejecutar tres pruebas más que corresponden a los casos que engloban las comparaciones binarias. Una es mas grande que la otra, el caso viceversa, o que ambos textos comparados tengan igual longitud en caracteres.

3 Desarrollo Grupal

Luego del planteamiento y desarrollo de la primera suite de pruebas realizada de manera individual, se realiza una discusión entre ambos integrantes del grupo con el fin de acordar una sola suite de pruebas, adaptada a las capacidades que poseen en común ambos programas creados. Siguiendo las respectivas formas que igualmente tienen para resolver dichos problemas. A continuación se describen las pruebas a realizar:

- **Limites de la pila:** Dependiendo de cómo lo maneje cada programa, de acuerdo a los límites establecidos por cada integrante, se harán pruebas asociadas a corroborar el manejo de sobrepasar el límite de la pila establecida.
- **Creación / Recuperación del archivo:** Dependiendo del manejo realizado por cada programa, se evaluará la creación de un documento de guardado y la recuperación del mismo.
- **Agregar e imprimir texto en la pila:** Según como haya sido implementado y realizado por cada programa, se probará la funcionalidad de agregar un texto en la pila, para luego imprimirlo por pantalla.
- **Agregar y eliminar texto de pila:** Se evaluará según cada programa, la creación y eliminación de un texto dentro de la pila.
- **Eliminar con la pila vacía:** Se evaluará el comportamiento de cada programa al realizar la función de eliminar un texto de la pila cuando esta ya se encuentra vacía.
- **Comparar textos iguales de la pila:** Se buscará comprobar el comportamiento de ambos programas en el momento de realizar una comparación de tamaños entre dos textos diferentes y de igual tamaño dentro de la pila.
- **Texto más corto y mas largo en pila con textos de igual tamaño:** Se buscará corroborar el correcto manejo del caso en que se ingresen textos de igual tamaño dentro de la pila y se realice la función de obtención del texto más largo y corto de la pila.

- **Comparación de textos en tamaño con el mismo texto:** Se ejecutará nuevamente la función de comparación de tamaño entre dos textos siendo esos dos textos la referencia a una misma ubicación de la pila
- **Comparación de textos con un texto vacío:** Utilizando la función de comparación de tamaño entre dos textos, se realizará la prueba de comparar un texto de la pila con un espacio vacío de la misma.
- **Comparar dos textos en pila:** Finalmente se ejecutará una prueba asociada a la comparación de dos textos diferentes dentro de la pila.

A partir de la definición de estos casos de prueba, se realizan las pruebas cruzadas en base a la adaptación de estas pruebas para cada uno de los programas. Para finalmente haber obtenido en ambos casos pruebas exitosas, siendo por ende en este caso la obtención de los mismos resultados durante la tercera fase de pruebas.