Registro de cambios - 5/20/2018  
Manuel Ferreira Benavides

Contents

[-Creación de repositorio en GitHub para tener registro de los cambios. 1](#_Toc514619973)

[-Se añadió: 2](#_Toc514619974)

[-void editarArchivoV0 () 2](#_Toc514619975)

[-void editarArchivoV2 (replace, replacement, filepath)/3 2](#_Toc514619976)

[-void editarArchivoV3 () 2](#_Toc514619977)

[bool obtenerDatosArchivo (SSLL<string, Candidato> &list) 2](#_Toc514619978)

[void empacarCandidatosEnCiudades (SSLL<string, Candidato> &lista) 3](#_Toc514619979)

[-Actualización de la clase SSLL (en IndexedList.h) 3](#_Toc514619980)

[inline std::vector<std::string> tokenize(const std::string &s, const std::string &delimiter) 3](#_Toc514619981)

[void tokenizeInPlace(std::string &src, const std::string &delim) 3](#_Toc514619982)

[int edad1970(std::string src) 3](#_Toc514619983)

[std::string edad(std::string src) 3](#_Toc514619984)

[int edad1970(std::string src) 3](#_Toc514619985)

[std::string timeNow() 3](#_Toc514619986)

[void buildTimeStruct(std::tm &tmStruct, const int years, const int months, const int days) 4](#_Toc514619987)

[-Se eliminó: 5](#_Toc514619988)

[-void editarArchivoV0 () 5](#_Toc514619989)

[-bool obtenerDatosArchivo (LinkedList<Candidato> &list) 5](#_Toc514619990)

[-void imprimirCandidatos (LinkedList<Candidato> &list) 5](#_Toc514619991)

[int edad1970(std::string src) 5](#_Toc514619992)

[Commits relevantes: 5](#_Toc514619993)

[Antes de borrar comentarios/funciones basura: 5](#_Toc514619994)

[Despues de: 5](#_Toc514619995)

### -Creación de repositorio en GitHub para tener registro de los cambios.

# -Se añadió:

///Intenta demostrar sobre escritura en un solo archivo. Casi imposible por carencia de un indicador para el offset de seekp ()

## -void editarArchivoV0 ()

/// Para hacerlo tiene que examinar cada cadena y luego reescribirla.

/// Crea 2 archivos. Uno temporal en el que toda ocurrencia de toReplace cambia a replacement en el archivo dado.

## -void editarArchivoV2 (replace, replacement, filepath)/3

/// SIN IMPLEMENTAR. Describe el seudocódigo necesario para la implementación definitiva que se usara para la modificación de registros.

## -void editarArchivoV3 ()

///Versión 2 de obtenerDatosArchivo. Su parámetro pasó a tipo Single Sorted Linked List (SSLL).

///Esta función empaca los contenidos del archivo siguiendo reglas fijas sobre el formato esperado en el archivo.

///El parámetro es SSLL dado que este tipo de lista es capaz de sortear sobre la clave de tipo string al momento de insertar.

///Tras ejecutar la función (dependiendo de la clave que la función) elija, lista estará sorteada por clave.

///Lo anterior es necesario para poder usar empacarCandidatosEnCiudades/1

## bool obtenerDatosArchivo (SSLL<string, Candidato> &list)

///Llenar lista de candidatos

/// O bien usando lista indexada por código de ciudad

/// O bien sorteando la lista al terminar de insertar por código de ciudad

///El código de inserción podría modificarse para deducir la lista de ciudades en simulación o simplemente tomarla las ciudades de archivo aparte.

///La lista sorteada presentara todos los candidatos a alcaldía. Los de una misma ciudad aparecerán de forma consecutiva.

///En esta lista entonces se evidenciara grupos de candidatos a una misma alcaldía.

///Entonces Con un solo condicional (siCiudadProcesadaActualmente!=CiudadNuevaEntrada) que aplicara sobre cada entrada de la lista

///Si además seguimos un estándar de orden alfabético para las ciudades activas (con candidatos), se sabe que cada vez que el condicional falle se pasara a la siguiente ciudad en orden alfabético

///Con lo anterior habrán unificado los condicionales

///if(nombreCiudad1) insertar candidato en ciudad1

///if(nombreCiudad2) insertar candidato en ciudad2

///...

///...

///if(noombreCiudad199) insertar candidato en ciudad199

///if(noombreCiudad200) insertar candidato en ciudad200

///En este contexto la ciudad tiene

///Arreglo CANDIDATOS de tamaño fijo=# partidos

///Arreglo RESULTADOS de tamaño votoNulo=0+votoBlanco=1+#partidos con candidato por ciudad

## void empacarCandidatosEnCiudades (SSLL<string, Candidato> &lista)

## -Actualización de la clase SSLL (en IndexedList.h)

Para que funcione el método bool **obtenerDatosArchivo (SSLL<string, Candidato> &lista)**

## inline std::vector<std::string> tokenize(const std::string &s, const std::string &delimiter)

## void tokenizeInPlace(std::string &src, const std::string &delim)

//Method using diff in seconds between timestamps and then deducing amount of yyyy mm dd passed from remainder.

## int edad1970(std::string src)

//Natural human method

## std::string edad(std::string src)

//Method using diff in seconds between timestamps and then deducing amount of yyyy mm dd passed from remainder.

## int edad1970(std::string src)

//Parses time using ISO-8601

## std::string timeNow()

## void buildTimeStruct(std::tm &tmStruct, const int years, const int months, const int days)

# -Se eliminó:

## -void editarArchivoV0 ()

Versiones que usaban LinkedList en vez de SSLL<string, Candidato>

## -bool obtenerDatosArchivo (LinkedList<Candidato> &list)

## -void imprimirCandidatos (LinkedList<Candidato> &list)

## int edad1970(std::string src)

# Commits relevantes:

Antes de borrar comentarios/funciones basura: https://github.com/jmferreirab/Proyecto/commit/0cbadc05288ae005e83064a5c1ca05d583c123b9

### Despues de:

https://github.com/jmferreirab/Proyecto/commit/2475ff28d2cde237aa7700057b945f3895da4f93