

Contents

-Creación de repositorio en GitHub para tener registro de los cambios.....	1
-Se añadió:	2
-void editarArchivoV0 ()	2
-void editarArchivoV2 (replace, replacement, filepath)/3.....	2
-void editarArchivoV3 ()	2
bool obtenerDatosArchivo (SLL<string, Candidato> &list)	2
void empacarCandidatosEnCiudades (SLL<string, Candidato> &lista).....	3
-Actualización de la clase SLL (en IndexedList.h).....	3
inline std::vector<std::string> tokenize(const std::string &s, const std::string &delimiter)	3
void tokenizeInPlace(std::string &src, const std::string &delim).....	3
int edad1970(std::string src).....	3
std::string edad(std::string src)	3
int edad1970(std::string src).....	3
std::string timeNow()	3
void buildTimeStruct(std::tm &tmStruct, const int years, const int months, const int days)	4
-Se eliminó:.....	5
-void editarArchivoV0 ()	5
-bool obtenerDatosArchivo (LinkedList<Candidato> &list).....	5
-void imprimirCandidatos (LinkedList<Candidato> &list)	5
int edad1970(std::string src).....	5
Commits relevantes:	5
Antes de borrar comentarios/funciones basura:	5
Despues de:.....	5

-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 pseudocódigo 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 (SSL).
//Esta función empaca los contenidos del archivo siguiendo reglas fijas sobre el formato esperado en el archivo.

///
//El parámetro es SSL 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 (SSL<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 (SLL<string, Candidato> &lista)

```

-Actualización de la clase SLL (en IndexedList.h)

Para que funcione el método bool **obtenerDatosArchivo (SLL<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 SLL<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>