

Instituto Tecnológico de Costa Rica

Estructura de Datos – Grupo 3

Ingeniería en Computación

Tarea Corta Programada No. 2

Profesor:

Juan Carlos Gomez Pereira

Estudiantes:

Heyler Johel Mora Calderón

Daniel Hoffman González (Zeta)

Introducción

El presente documento pretende explicar los aspectos más importantes de la segunda tarea corta. La primer sección llamada **ListB** contiene una explicación breve de los algoritmos implementados en la clase. La sección **GitHub** presenta en forma detallada los conceptos relacionados con GitHub. También incluye toda la explicación necesaria y los paso a paso para hacer uso de GitHub desde Visual Studio.

El desarrollo de la tarea se puede decir que es la implementación de dos clases, **ListaSimple** y **ListaSimpleB**. **ListaSimple** como su nombre lo dice es una lista simple que en cada uno de sus nodos almacena un elemento. Por otro lado, **ListaSimpleB** se implementa por medio de nodos que tienen la capacidad de almacenar, cada uno de ellos, una cantidad N de elementos.

Este trabajo se desarrolló con el lenguaje C++ en Visual Studio y con una plataforma de administración de código llamada GitHub. Esta herramienta permite tener un buen control del código, de sus versiones y de otros componentes del proyecto y así tener la posibilidad de volver atrás en caso de error. También nos permite mantener el código de una forma ordenada, por usuarios.

Objetivos de dicho trabajo son:

- Reforzar el uso de punteros y manejo de memoria dinámica a través de bloques de memoria.
- Definir e implementar clases como un mecanismo de abstracción.
- Investigar y utilizar una plataforma de control de código, basada en la nube, por medio de la cual se pueda coordinar el trabajo de los miembros de un equipo de desarrollo.

ListB

La clase ListB contiene una estructura que posee como atributos, un booleano que nos indica si el nodo está lleno, un puntero al siguiente nodo y un arreglo de N elementos. Además posee dos constructores, uno de ellos crea un nodo y lo inicializa sus atributos, el otro genera un nuevo nodo, y el nodo actual se convierte en el nodo siguiente. Esta estructura de datos pretende crear un nodo nuevo cada vez que el arreglo del último nodo se llene, por ende cada uno de los métodos de la clase ListB debe cumplir esta regla.

A continuación se detalla el objetivo de cada uno de los métodos de la clase ListB, **List(string nombre)**; Crea una lista vacía cuyo nombre fue establecido con el parámetro respectivo. **len()**; Retorna el tamaño, cantidad de elementos, de la lista. **push_front(T x)**; Inserta x al frente de la lista. **push_back(T x)**; Inserta x al final de la lista. **insertar(T x, int pos)**; Inserta x en la posición pos de la lista. **remove(int pos, T& x)**; Remueve el elemento en posición pos de la lista y lo deja en x. Retorna true si la operación se realizó. **pop(T& x)**; Remueve el elemento al inicio de la lista y lo deja en x. Retorna true si la operación se realizó. **pop_back(T& x)**; Remueve el elemento al final de la lista y lo deja en x. Retorna true si se realizó. **get(int pos, T& x)**; Deja en x el elemento en posición pos de la lista. Retorna true si la operación se realizó. **get_front(T& x)**; Deja en x el elemento al inicio de la lista. Retorna true si la operación se realizó. **get_back(T& x)**; Deja en x el elemento al final de la lista. Retorna true si la operación se realizó. **print()**; Imprime la lista en el formato: Nombre_lista = [array, array, ...]. **~List()**; Destructor de la clase.

GitHub

Git es uno de los sistemas de control de versiones más populares y del que más estamos oyendo hablar últimamente. Gracias a su potencia y versatilidad muchos grandes proyectos de software libre están migrando sus repositorios a Git. Cada vez más es más importante saber usar Git, tanto a nivel personal como laboral.

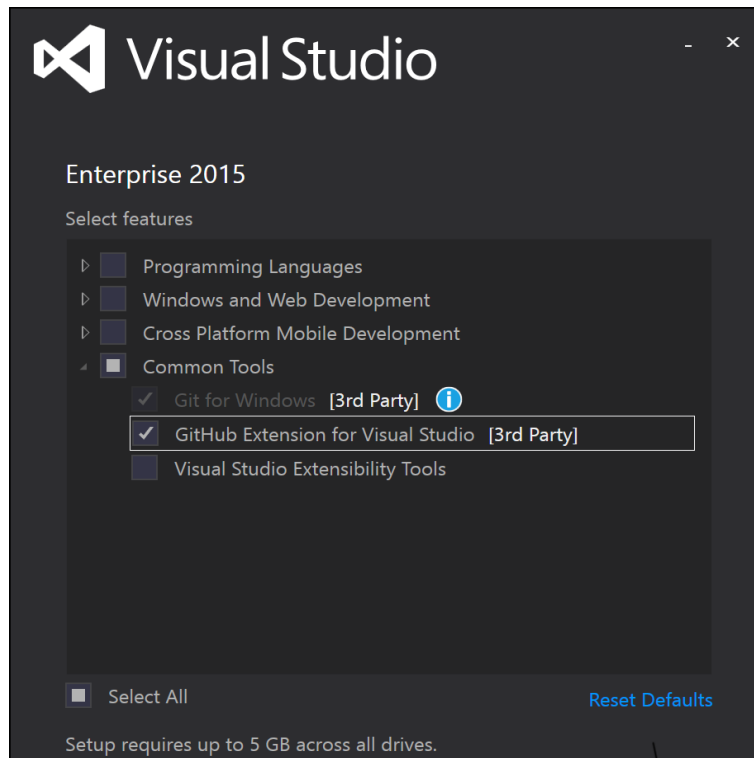
Conceptos

Un **repositorio** es un espacio centralizado donde se almacena, organiza, mantiene y difunde información digital, habitualmente archivos informáticos. Cualquier archivo que quieras mandar a la zona de index lo haces con el comando "**add**". A la acción de guardar los archivos en el repositorio se llama, en la jerga de Git, hacer un "**commit**".

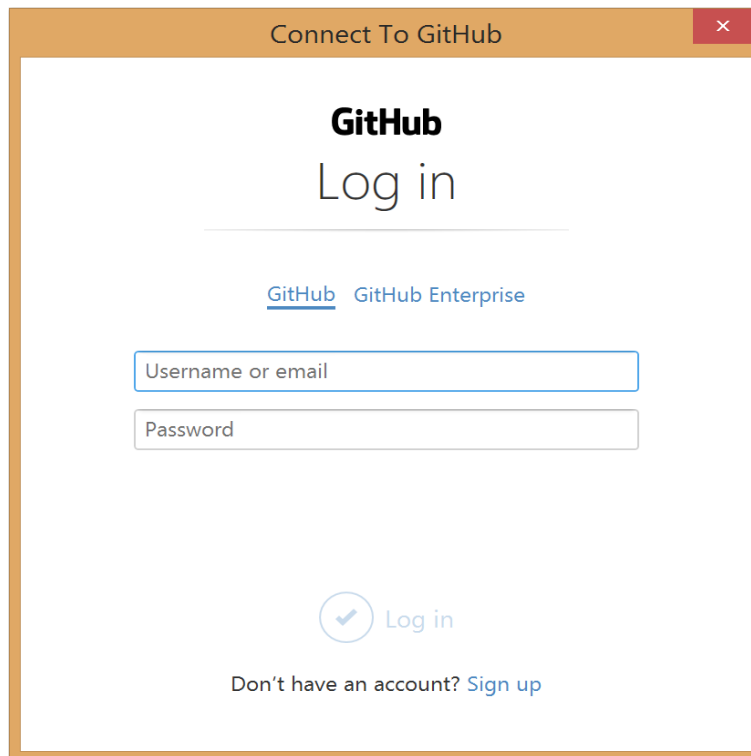
Una **rama** nueva no estará disponible para los demás a menos que subas (**push**) la rama a tu repositorio remoto. Se recomienda crear **etiquetas** para cada nueva versión publicada de un software.

GitHub desde Visual Studio.

Cuando instale Visual Studio, seleccione la opción para personalizar la instalación y asegúrese de marcar la casilla de verificación Extensión de GitHub para Visual Studio.



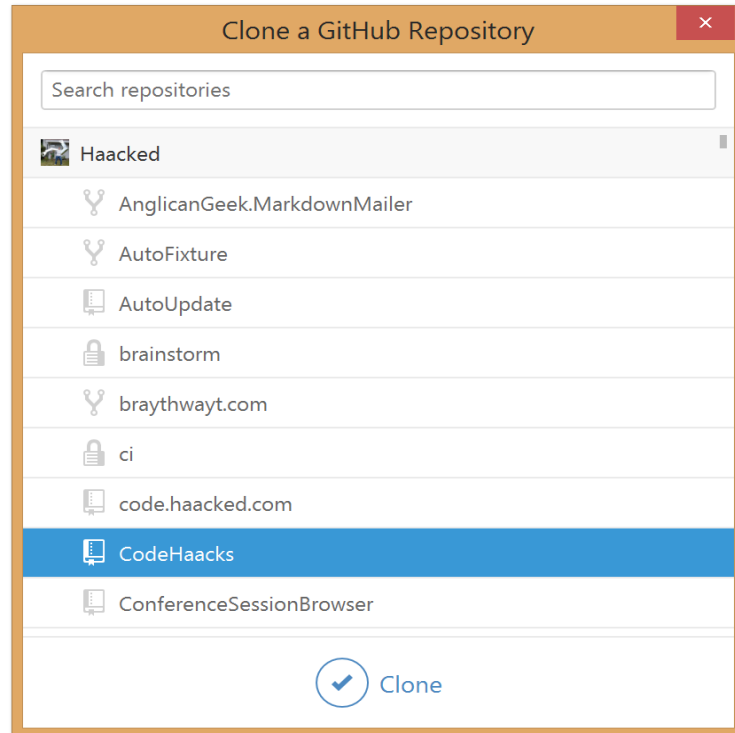
Una vez que instale la extensión, puede iniciar sesión en su cuenta de GitHub.com o GitHub Enterprise en Visual Studio con soporte completo para la autenticación de dos factores.



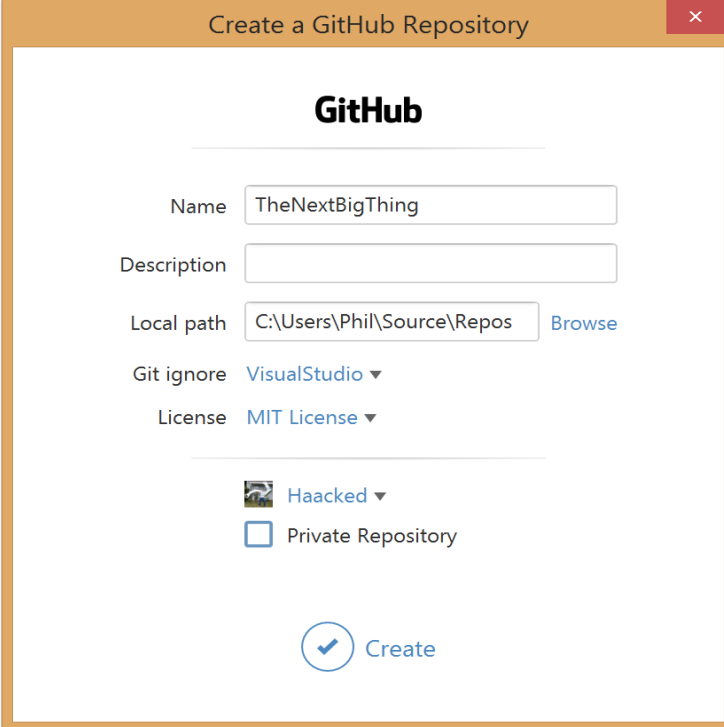
The image shows a 'Connect To GitHub' dialog box with a close button (X) in the top right corner. Inside the dialog, the GitHub logo is displayed above the text 'Log in'. Below this, there are two links: 'GitHub' (underlined) and 'GitHub Enterprise'. There are two input fields: 'Username or email' and 'Password'. At the bottom, there is a 'Log in' button with a checkmark icon and a link 'Don't have an account? Sign up'.

Una vez conectado, es rápido clonar los repositorios de su cuenta o de cualquier organización a la que pertenezca.

Haga clic en el botón de clonar para abrir un cuadro de diálogo que muestra todos los repositorios a los que tiene acceso. Utilice el cuadro de texto del filtro para encontrar rápidamente el repositorio que le interesa.



¿Necesita un nuevo repositorio en GitHub? ¡No hay problema! El diálogo de creación facilita la creación de un repositorio en GitHub y su clonación en Visual Studio, todo en un solo paso.



Create a GitHub Repository

GitHub


Name

Description


Local path [Browse](#)

Git ignore [VisualStudio](#) ▼

License [MIT License](#) ▼

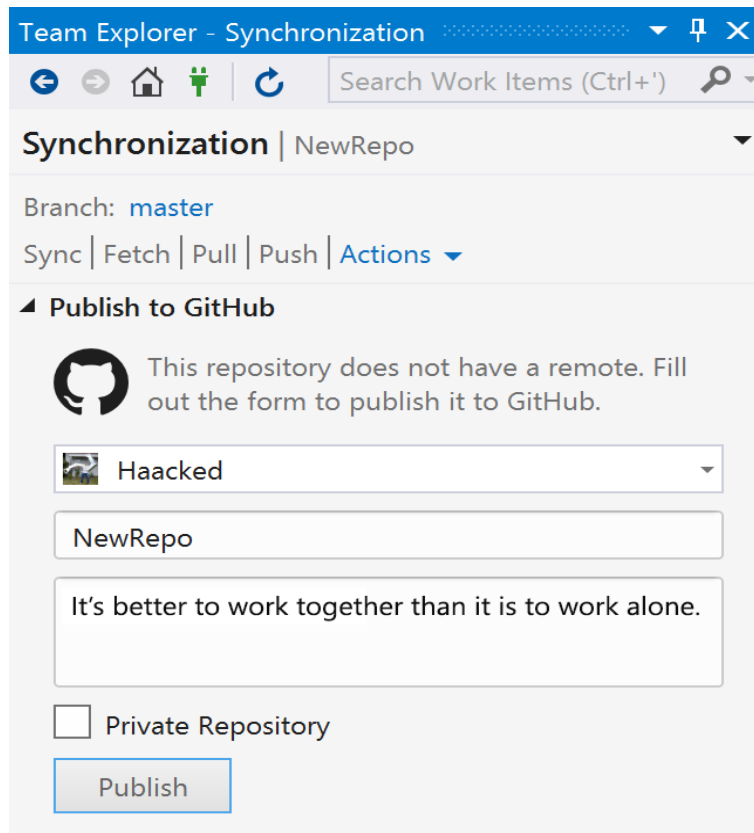
 [Haacked](#) ▼

☐ Private Repository

 Create

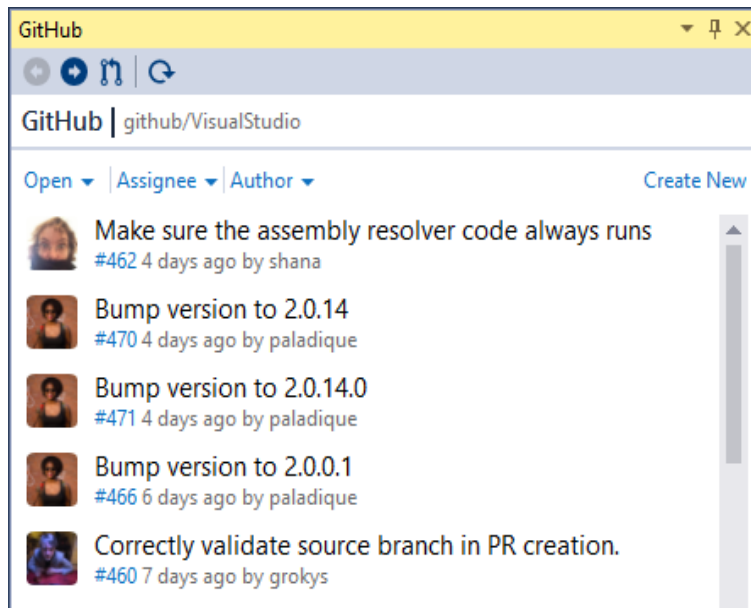
¿Ya tienes un proyecto en tu máquina? ¿Bueno, qué estás esperando? Publícalo en GitHub y comienza a colaborar con otros.

En la ventana de Team Explorer, haga clic en la pestaña Sincronizar para acceder al cuadro de diálogo Publicar en GitHub.



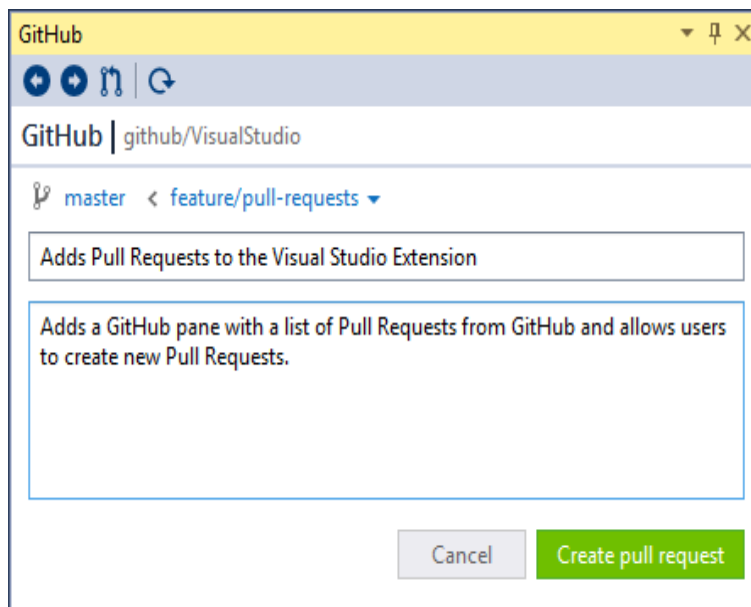
Vea todas las solicitudes de extracción para su proyecto en el panel de GitHub, y clasifíquelas y filtre por estado Abierto / Cerrado, Asignado y Autor.

Abra el panel GitHub escribiendo GitHub en Inicio rápido de Visual Studio (Ctrl + Q).



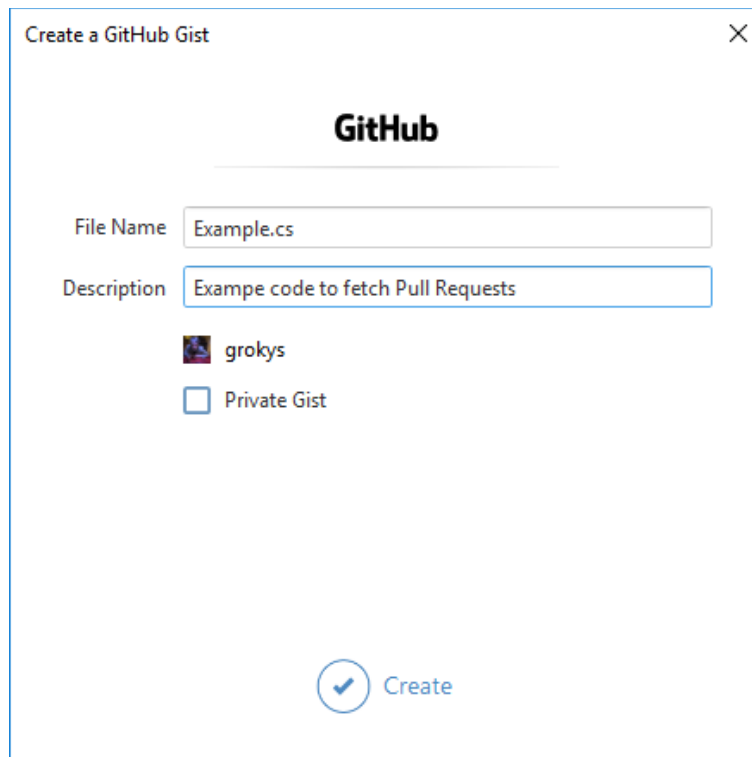
Convierta una rama en una solicitud de extracción directamente desde Visual Studio.

En el panel de GitHub, haga clic en el enlace Crear nuevo para crear una nueva solicitud de extracción en GitHub.



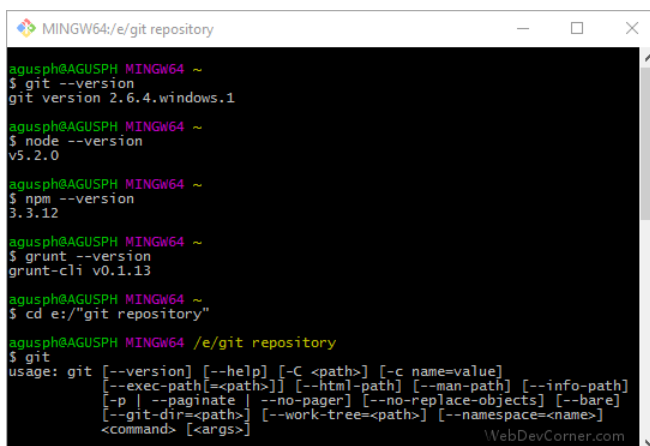
Crea listas y cárgalas a GitHub directamente desde Visual Studio.

Simplemente seleccione el código para cargar y seleccione Crear lista en el menú contextual de GitHub.



Git Bash

Es la implementación de Git mediante el uso del terminal del S.O. , de esta manera se puede hacer cambios al repositorio en la plataforma que esté (en este caso GitHub) sin necesidad de abrir el Visual Studio



```

MINGW64:/e/git repository
agusp@AGUSPH MINGW64 ~
$ git --version
git version 2.6.4.windows.1

agusp@AGUSPH MINGW64 ~
$ node --version
v5.2.0

agusp@AGUSPH MINGW64 ~
$ npm --version
3.3.12

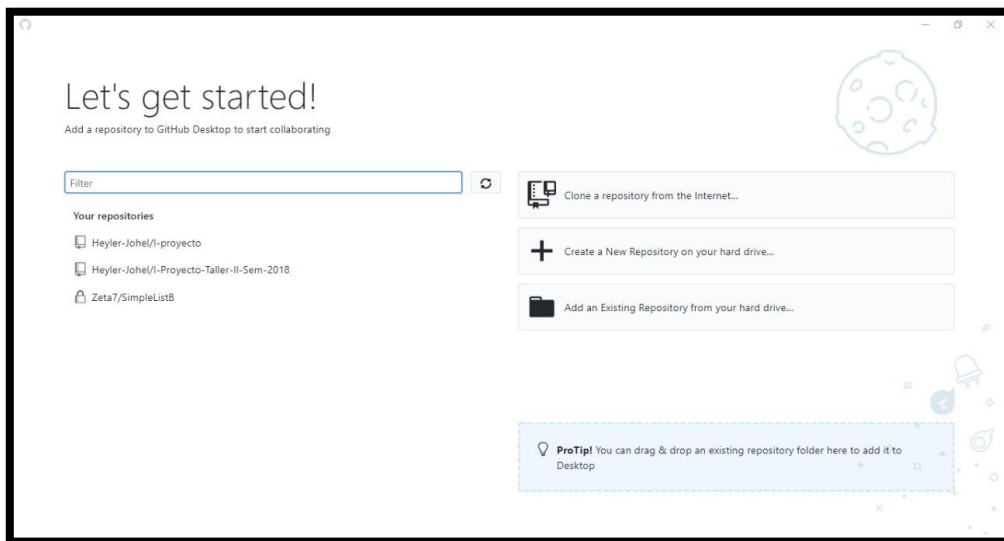
agusp@AGUSPH MINGW64 ~
$ grunt --version
grunt-cli v0.1.13

agusp@AGUSPH MINGW64 ~
$ cd e:/git repository

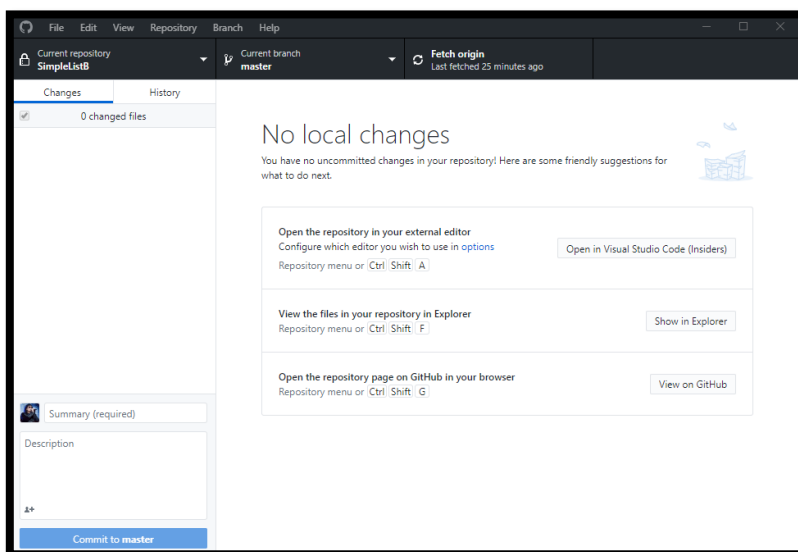
agusp@AGUSPH MINGW64 /e/git repository
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
       [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
       [-p | --paginate] [--no-pager] [--no-replace-objects] [--bare]
       [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
       [<command>] [<args>]
  
```

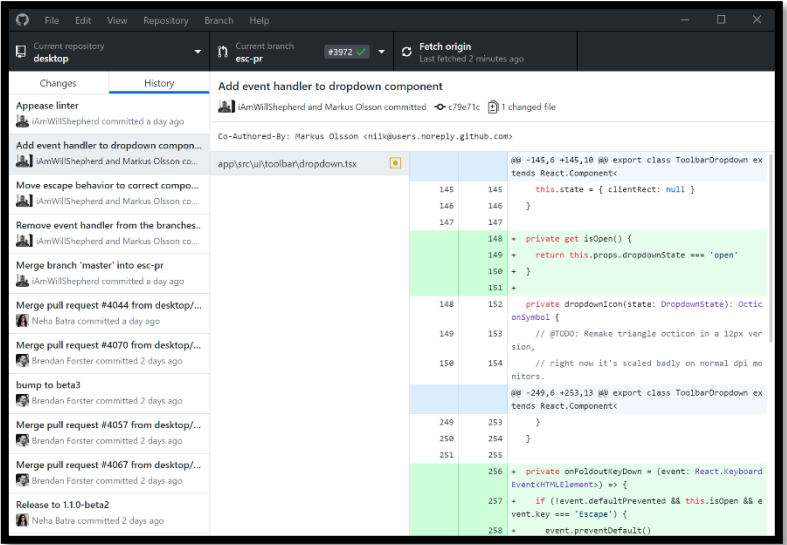
GitHub Desktop

Es el cliente de Git diseñado por GitHub, este es mediante el uso de una GUI para el usuario y así brindarle un manejo más simple de todas las herramientas de Git



Este cliente de Git maneja una interfaz muy amigable con el usuario, es muy intuitivo y tanto para la creación de branch como para realizar, inspeccionar y depurar los commits es muy simple de usar





Resultados Obtenidos

Se obtuvo un 100% de completitud en los objetivos impuestos para este proyecto, esto es más simple verlo mediante la siguiente tabla juntamente con el estado en que se encuentran las funciones que describe.

Método	Estado
SimpleListB(string n) [Constructor]	100%
~SimpleListB() [Destructor]	100%
length() [largo de la lista]	100%
push_front(T x) [Inserción al frente]	100%
push_back(T x) [Inserción al final]	100%
insert(T x, int pos) [Inserción en posición determinada]	100%
remove(int pos, T& x) [Eliminación en posición determinada]	100%
pop(T& x) [Eliminación al frente]	100%
pop_back(T& x) [Eliminación al final]	100%
get(int pos, T& x) [Obtención de posición determinada]	100%
get_front(T& x) [Obtención del frente]	100%
get_back(T& x) [Obtención del final]	100%
print() [Impresión de los elementos de la lista juntamente con formato para facilitar su lectura]	100%

Conclusiones

Podemos concluir los objetivos iniciales del proyecto han sido cumplidos en su totalidad, pues hemos logrado manipular los punteros y manejar la memoria dinámica a través de bloques de memoria, para la implementación de listas simples y listas con N cantidad de nodos, pues es necesario conocer el uso de punteros para cada uno de los métodos de estas clases.

Hemos logrado implementar la abstracción de datos solicitada por el profesor, ya que hemos implementado las clases para obtener los datos necesarios y realizar los métodos necesarios para generar la estructura de datos pedida.

Por último, podemos decir que la herramienta de GitHub ha sido de mucha ayuda durante el desarrollo de esta tarea, pues ha servido para versionar el código, y esto es una ventaja en caso de que el programa en algún momento falle. Además simplifica el trabajo en equipo ya que varias personas pueden trabajar sobre el mismo

Bibliografía

GitHub, M. (2019). *Getting Started with GitHub Desktop*. Obtenido de <https://help.github.com/en/desktop/getting-started-with-github-desktop>

git-scm. (s.f.). *Git Basics*. Obtenido de <https://git-scm.com/video/what-is-version-control>

git-scm. (s.f.). *Git Documentation*. Obtenido de <https://git-scm.com/docs>

Programación, E. A. (16 de Abril de 2018). *Youtube*. Obtenido de <https://www.youtube.com/watch?v=a0lD9rrQOWE&feature=youtu.be>