Universidad Autónoma de Baja California

Facultad de Ciencias Químicas e Ingeniería



**INTERACCION HUMANO-COMPUTADORA**

**Control de versionamiento Git y Unity**

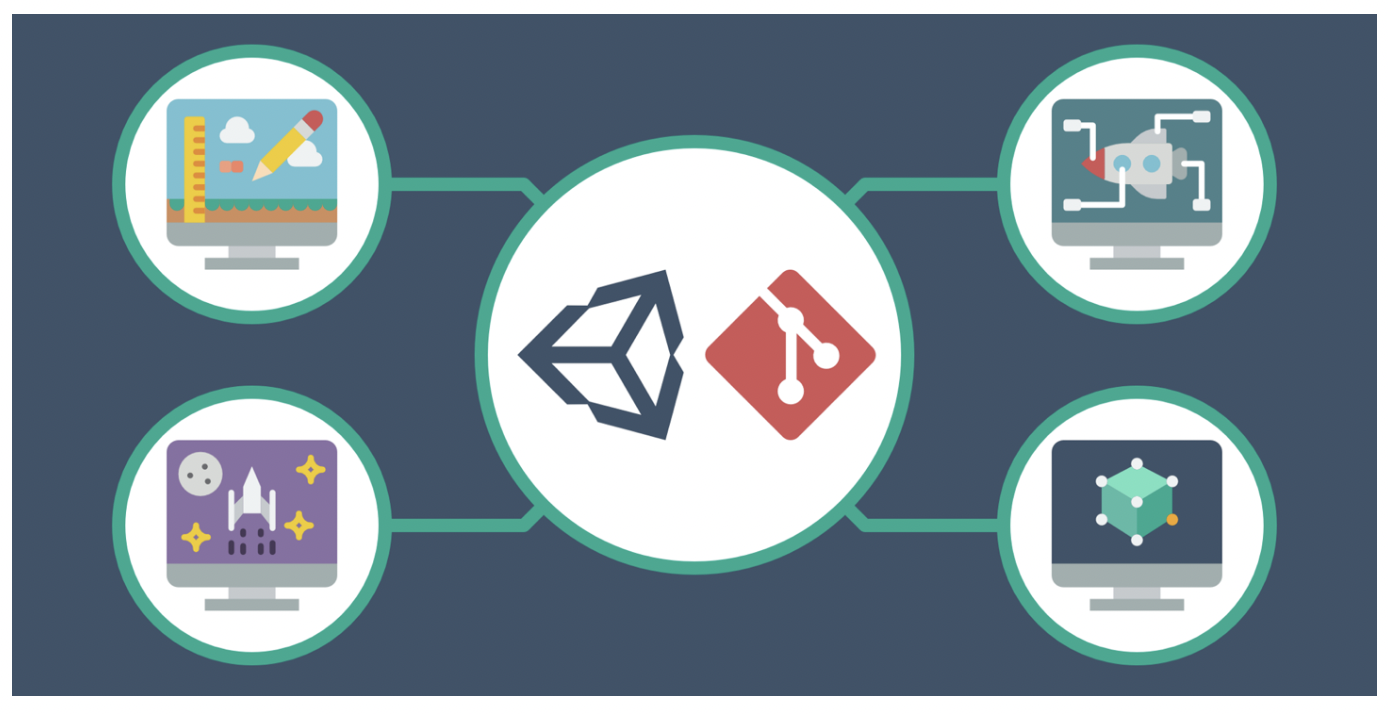
**Docente:** Trujillo Perez Humberto

**Alumno:** Gómez Cárdenas Emmanuel Alberto

**Matricula:** 01261509

Competencia

El alumno se familiarizará con el flujo de trabajo en la integración de control de versionamiento Git y Unity Game Engine.



Descripción

**1.- Crear un nuevo repositorio público en Github e inicializarlo con un Readme y archivo gitignore.**

**A screenshot of a computer

Description automatically generated with medium confidence**

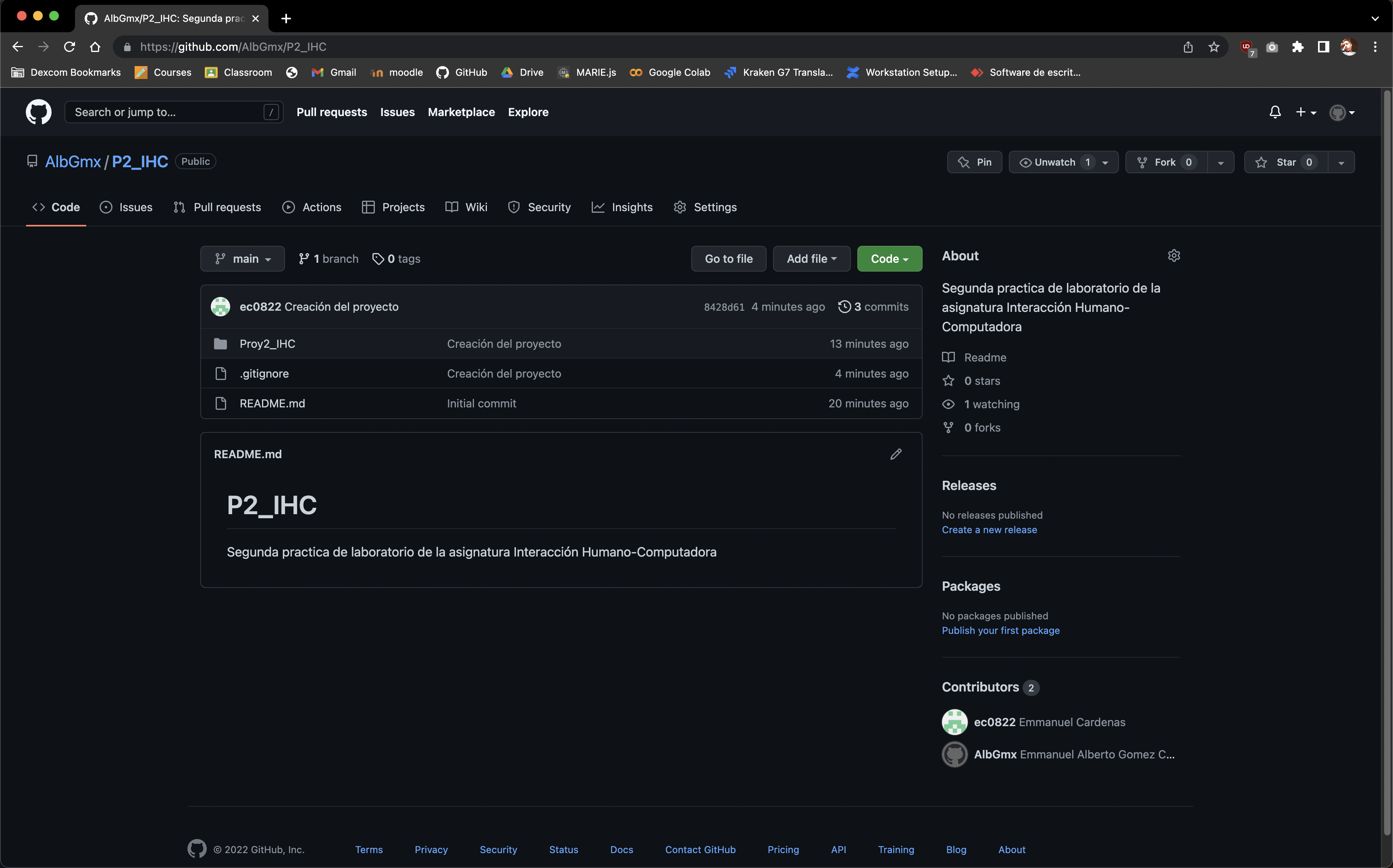
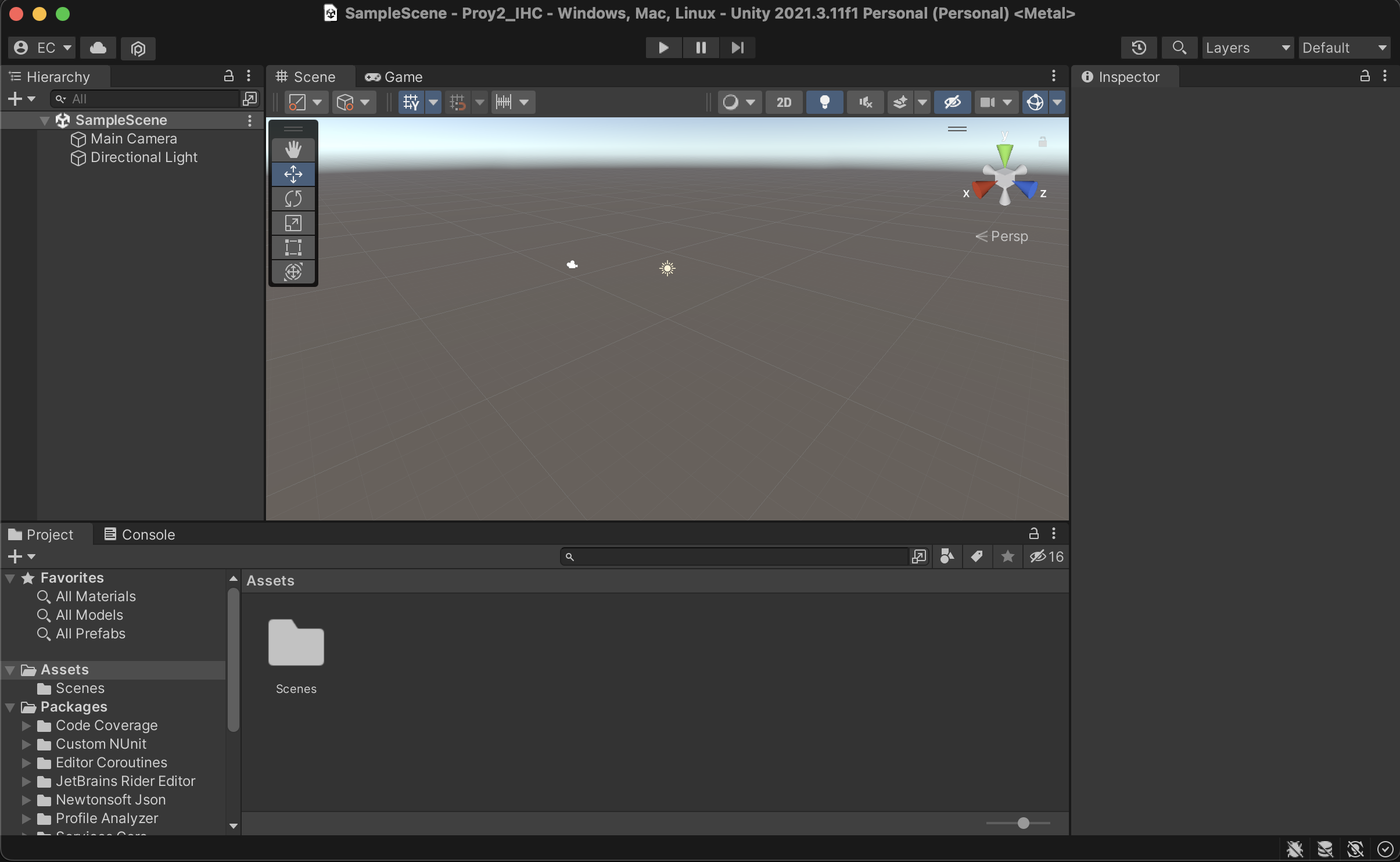
**2.- Instalar git y Github desktop (opcional) para manejo de repositorios locales.**

**3.- Clonar repositorio localmente.**

**Text

Description automatically generated**

**4.- Crear nuevo proyecto de Unity dentro del directorio del repositorio y realizar primer commit del proyecto vacío.**

****

**5.- Crear un nuevo branch llamado “Feature/practica2”, sobre este branch se trabajará el problema de programación descrito más adelante.**

**Graphical user interface

Description automatically generated**

**6. Realizar un merge para integrar el branch “Feature/practica2” sobre el branch principal “main”.**

**Text

Description automatically generated**

**7.- Responder las siguientes preguntas.**

* **¿Qué es Git?**
  + Es un sistema de control de versiones, distribuido, lo que significa que un clon local del proyecto es un repositorio de control de versiones completo.
* **¿Qué es un commit?**
  + El comando git commit captura una instantánea de los cambios preparados en ese momento del proyecto. Las instantáneas confirmadas pueden considerarse como versiones "seguras" de un proyecto
* **¿Qué es un branch?**
  + El comando git branch te permite crear, enumerar y eliminar ramas, así como cambiar su nombre. No te permite cambiar entre ramas o volver a unir un historial bifurcado. Por este motivo, git branch está estrechamente integrado con los comandos git checkout y git merge.
* **¿Qué es un merge?**
  + El comando git merge permite tomar las líneas independientes de desarrollo creadas por git branch e integrarlas en una sola rama.
* **Describa el funcionamiento de las acciones fetch, pull, push.** 
  + **Fetch:** Descarga los commits, archivos, etc, de un repositorio remoto a un repositorio local.
  + **Pull:** Extrae y descarga contenido desde un repositorio remoto y actualiza el repositorio local
  + **Push:** Es usado para cargar contenido a un repositorio remoto desde un repositorio local.

Problema

Dado un arreglo de números enteros **nums** y un entero **target**, implementar una función que regrese los **indices** de dos números que sumados den como resultado el **target**. Asumir que siempre habrá **al menos una solución**, y tampoco se puede repetir el mismo elemento más de una vez.

La respuesta se puede regresar en cualquier orden. public int[] SumaDos(int[] nums, int target) {} Ejemplo:  
Input: nums = [2,7,11,15], target = 18

Output = [1, 2]  
BONUS: Indicar la complejidad en espacio y tiempo.

using System;

internal class Program

{

private static void Main(string[] args)

{

int[] array = { 2,7,11,15 };

int target = 18;

int[] result = SumaDos(array, target);

Console.WriteLine("The indexes are: {0} and {1}", result[0], result[1]);

}

private static int[] SumaDos(int[] nums, int target)

{

int[] result = new int[2];

for (int i = 0; i < nums.Length; i++)

{

for (int j = i + 1; j < nums.Length; j++)

{

if (nums[i] + nums[j] == target)

{

result[0] = i;

result[1] = j;

return result;

}

}

}

return result;

}

}

//Complejidad en tiempo: O(n^2)

//En espacio: O(1)

A screenshot of a computer

Description automatically generated

Repositorio Github

<https://github.com/AlbGmx/P2_IHC>

Conclusion

En esta práctica conoci la herramienta de control de versionamiento Git asi como sus distintos comandos y aprendí a manejarlos.