

**GITHUB DESKTOP PARA EL MANEJO DE LOS RECURSOS, DOCUMENTACIÓN Y
MANEJO DEL PROYECTO DE GAMEMAKER.**

PINGÜI

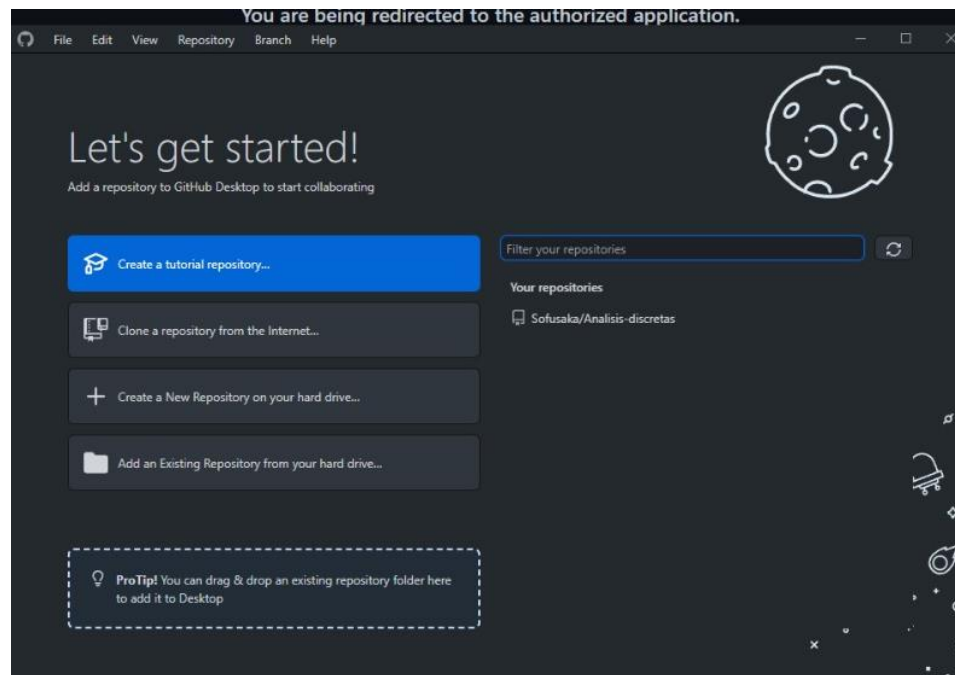
Documento creado por Santiago Cárdenas

Tabla de Contenido

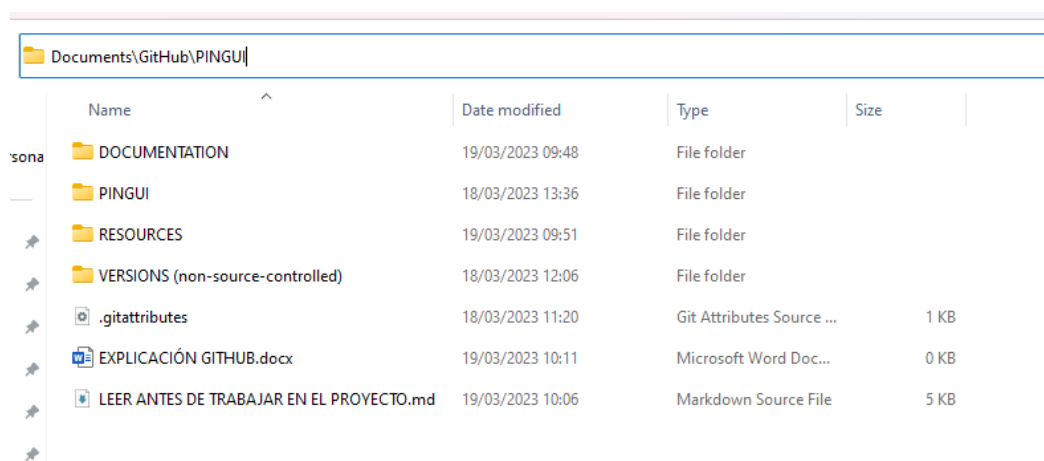
| | |
|---|----|
| 1. CLONACIÓN DE PINGÜI | 3 |
| 2. BRANCHES..... | 4 |
| 2.1. ¿CÓMO FUNCIONAN? | 4 |
| 2.2. CÓMO CREAR BRANCHES, TRABAJAR Y PUBLICAR | 5 |
| 2.3. CÓMO HACER UN MERGE DE UN PULL REQUEST..... | 9 |
| 3. DISTRIBUCIÓN DEL REPOSITORIO: PINGÜI | 11 |
| 3.1. CARPETA DOCUMENTATION..... | 11 |
| 3.2. CARPETA PINGUI..... | 11 |
| 3.3. CARPETA RESOURCES | 11 |
| 3.4. CARPETA VERSIONS (non-source-controlled) | 12 |
| 3.5. ARCHIVO .gitattributes..... | 12 |
| 3.6. ARCHIVO “REGISTRO CAMBIOS.md” | 12 |

1. CLONACIÓN DE PINGÜI

1. El primer paso que se hará es tener GitHub Desktop instalado en el PC. Cuando este se tenga en la máquina a trabajar, se dará click en la opción “Clone a repository from the Internet.” Y se pondrá el siguiente link: <https://github.com/davopaco/PINGUI.git>



2. Después de que se tenga el repositorio, se descargará localmente la “main branch” (ASEGÚRENSE EN DÓNDE LO DESCARGAN; sin embargo, si no se dieron cuenta, por default se encuentra en: “Documentos\GitHub\PINGUI”).



3. La anterior imagen describe un clon del repositorio de PINGUI el cuál va a ser dónde van a trabajar con todo lo que tengan que hacer.

4. Estando en esa carpeta NO MODIFIQUEN/ELIMINEN/AGREGUEN NADA AÚN, porque se debe establecer una BRANCH o su espacio de trabajo.

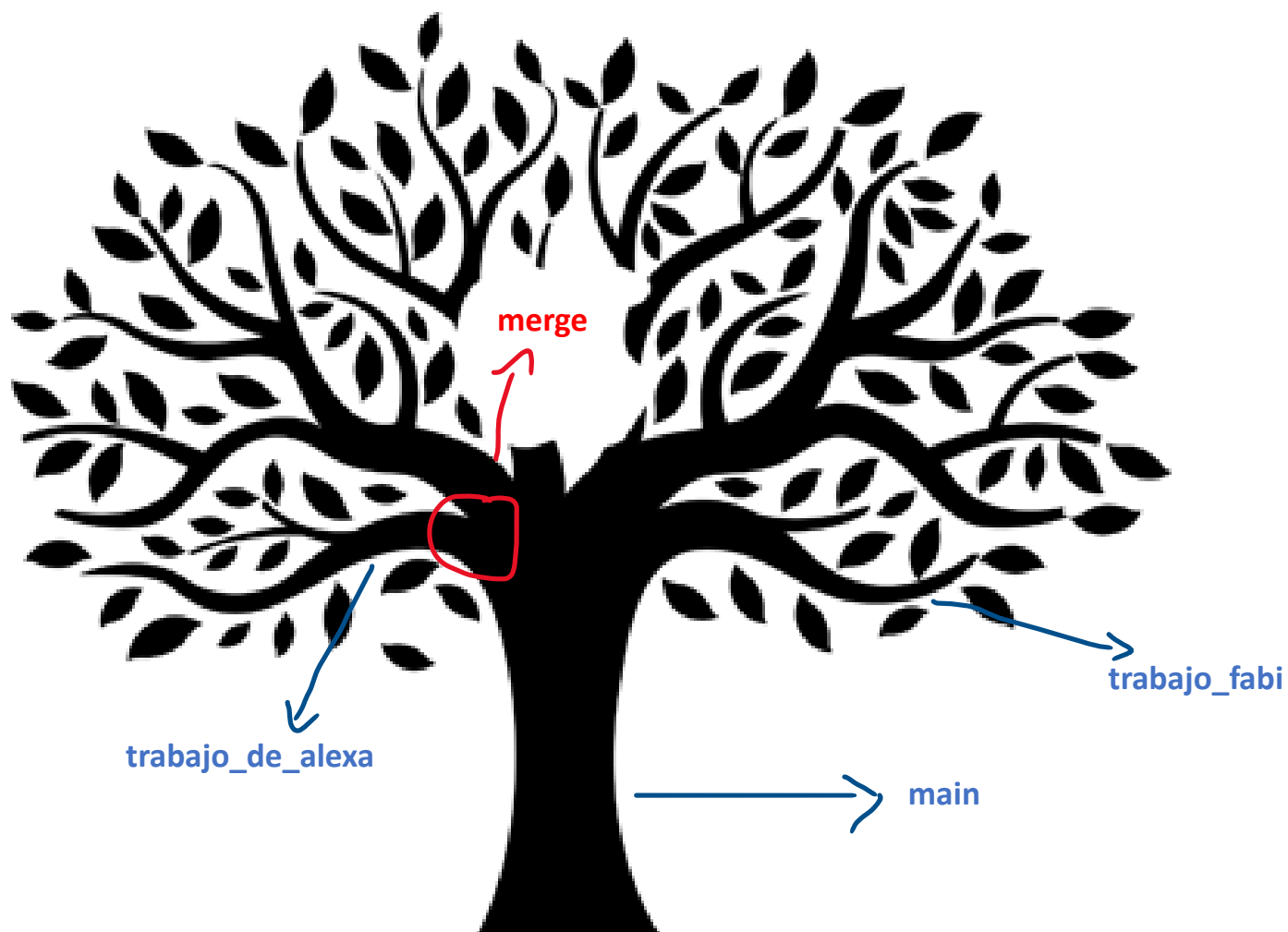
2. BRANCHES

2.1. ¿CÓMO FUNCIONAN?

Las famosas “**branches**” son conocidas como las ramas de trabajo en GitHub y son “**versiones alternativas**” que se pueden hacer de otras branches, aunque comúnmente se hacen de la rama por **default** o, como me gusta llamarlo, el **tronco** (En nuestro caso la “**main**” branch).

Este sistema funciona como si fuese un árbol (de la vida real, no estructura de datos) y que se pueden unir/fusionar, mediante la función **merge**.

Pongamos un **EJEMPLO** para poder comprender lo básico hasta esta parte.



Como se puede observar, “**trabajo_de_alexa**” es una branch que creó Alexa, que está basada en el “**main**” y, cuando acabó de trabajar, llamó a la función **merge** para que TODO aquello que estaba trabajando o modificando, pasase al “**main**”.

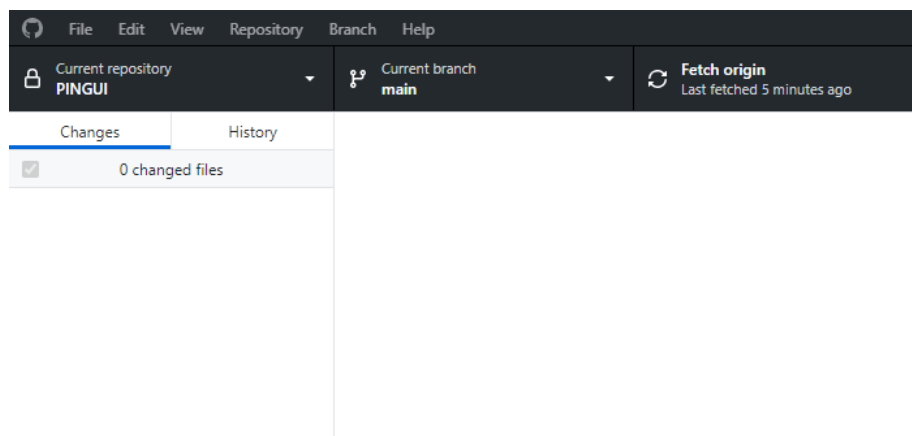
Lo mismo sucedería con la Branch “**trabajo_fabi**” que fue una rama que creó Fabiana trabajando desde la versión original “**main**” y cuando acabó el trabajo se hizo **merge** con la rama “**main**”.

IMPORTANTE: Esta función **merge** solo se va a hacer cuando realicen un **PULL REQUEST** que funciona como pedir una autorización para que se haga la función **merge**. ¿Por qué se hace esto? Porque supongamos que Fabiana y Alexa ambas están trabajando desde la versión del **main**, entonces eso quiere decir que aquello que haga Alexa, no lo va a ver Fabiana y lo mismo en el sentido contrario. Entonces cuando Alexa acabe, ella hace un **PULL REQUEST** para que se pueda comprobar si no causa ningún conflicto el unir su rama “**trabajo_de_alexa**” con la rama “**main**” y lo mismo hace Fabiana con la suya.

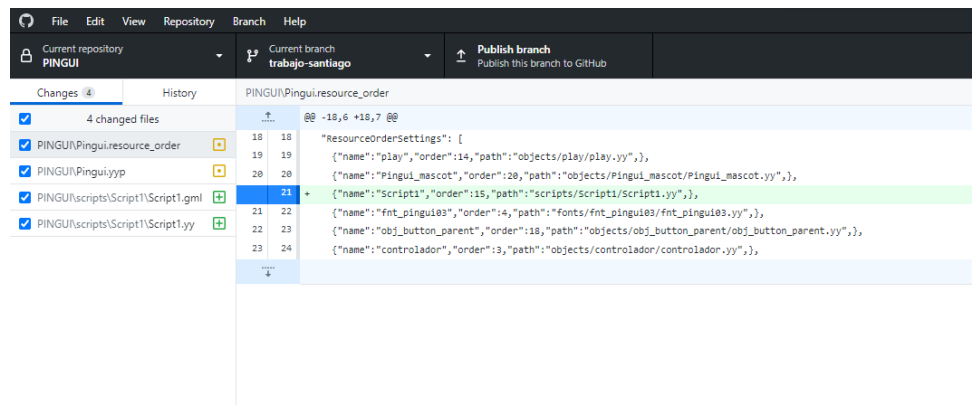
Ya al final, se hace el cambio a la rama “**main**” y, si ya se hizo la función **merge**, entonces todos pueden ver los cambios en el proyecto principal.

2.2. CÓMO CREAR BRANCHES, TRABAJAR Y PUBLICAR

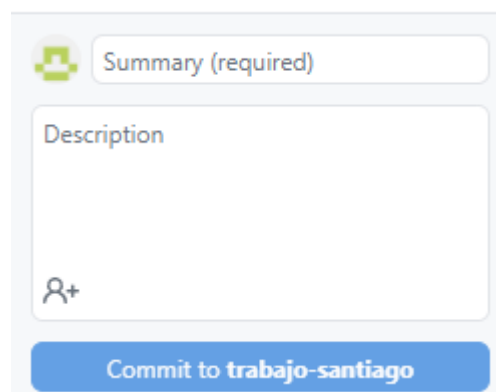
1. Primero, antes de crear una rama, es NECESARIO darle click a “**Fetch Origin**”, puesto que se necesita traer la versión más actualizada de la rama que da la base a la que van a crear en el repositorio (en este caso “**main**”). Posteriormente, en GitHub Desktop se va al apartado “**current branch**” y se da click en “**New Branch**”.



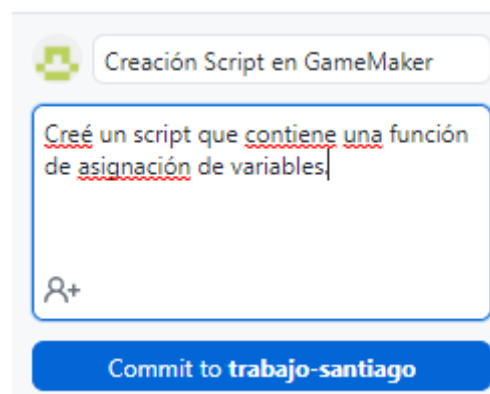
5. Después de eso le doy a CTRL + S o guardo el proyecto. Al guardar el proyecto, GitHub Desktop me detecta los cambios que hice y me los enumera en el cuadrito “Changes”



6. Ahora ya que he acabado mi trabajo, es hora de pasar a la función **COMMIT** que funciona de la misma manera en cómo lo hace en base de datos. Esta función me permite confirmar de que he hecho **x, y, z** modificaciones (en mi caso agregar un script a GameMaker). Acá le puedo agregar un título y una descripción (opcional), antes de hacerle **commit**. Esta función se encuentra en la **PARTE INFERIOR IZQUIERDA** de la interfaz.



7. Ahora le agrego mi título, mi descripción y le doy click a “Commit to trabajo-santiago”.



8. Después de hacerle **commit**, es hora de publicar nuestra branch a GitHub, entonces le damos click a “**Publish branch**”. Esto se hace para que TODOS puedan ver mi branch y lo que se ha hecho ahí.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Publish your branch

The current branch (trabajo-santiago) hasn't been published to the remote yet. By publishing it to GitHub you can share it, open a pull request, and collaborate with others.

Always available in the toolbar or `Ctrl | P`

Publish branch

Open the repository in your external editor

Select your editor in [Options](#)

Open in Visual Studio Code

Repository menu or `Ctrl | Shift | A`

View the files of your repository in Explorer

Repository menu or `Ctrl | Shift | F`

Show in Explorer

Open the repository page on GitHub in your browser

Repository menu or `Ctrl | Shift | G`

View on GitHub

9. Entonces después de publicada nuestra branch, ahora hacemos un **Pull Request** dándole click a **“Create Pull Request”**.

No local changes

There are no uncommitted changes in this repository. Here are some friendly suggestions for what to do next.

Create a Pull Request from your current branch

The current branch (trabajo-santiago) is already published to GitHub. Create a pull request to propose and collaborate on your changes.

Branch menu or `Ctrl | R`

Create Pull Request

Open the repository in your external editor

Select your editor in [Options](#)

Open in Visual Studio Code

Repository menu or `Ctrl | Shift | A`

View the files of your repository in Explorer

Repository menu or `Ctrl | Shift | F`

Show in Explorer

Open the repository page on GitHub in your browser

Repository menu or `Ctrl | Shift | G`

View on GitHub

Esto nos va a abrir el navegador web seleccionado por defecto y nos lo abrirá en una sección llamada **“Open a pull request”** en la que se puede escribir el título de nuestro pull, la descripción (se pueden agregar imágenes si se desea/requiere) y ver una *preview* de cómo queda estructurado.

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also [compare across forks](#).

base: main ← compare: trabajo-santiago ✓ Able to merge. These branches can be automatically merged.

Creación Script en GameMaker

Write Preview H B I `< >`

Creé un script que contiene una función de asignación de variables.

Attach files by dragging & dropping, selecting or pasting them.

Create pull request

Remember, contributions to this repository should follow our [GitHub Community Guidelines](#).

Reviewers
No reviews

Assignees
No one—assign yourself

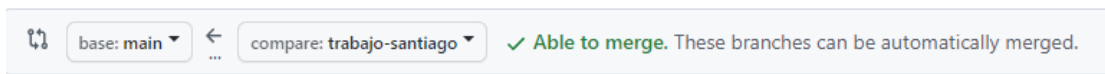
Labels
None yet

Projects
None yet

Milestone
No milestone

Development
Use [closing keywords](#) in the description to automatically close issues

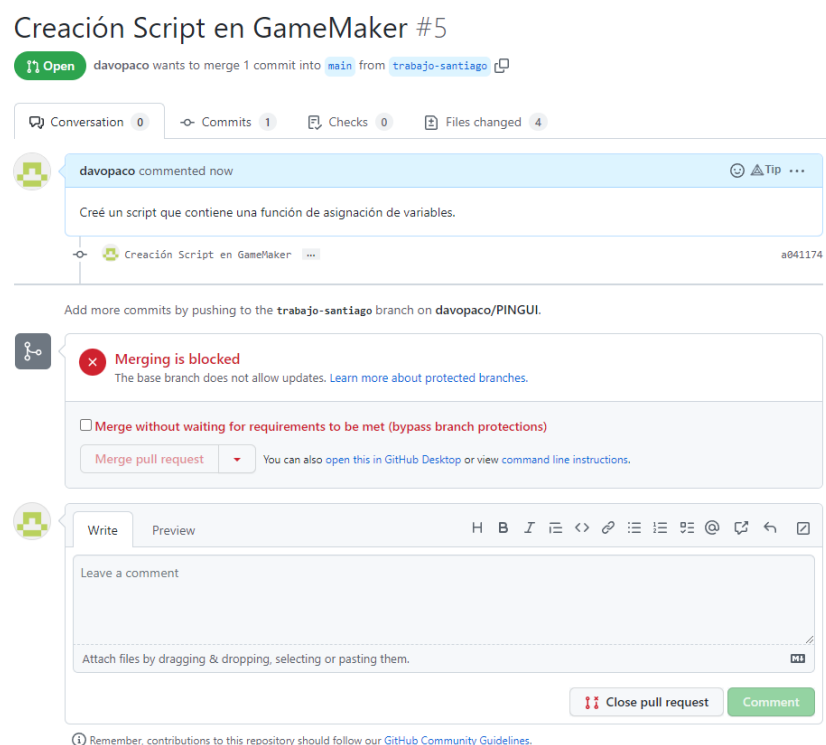
Aquí no solo se puede hacer eso, si no que también muestra a qué rama quiero hacerle **merge** y en la parte derecha muestra si tiene conflictos o no al hacer esto.



En mi caso, estoy comparando mi rama “**trabajo-santiago**” con la rama “**main**” y me da un aviso que me dice que sí puedo unir los cambios en mi rama a la rama principal. (No hay conflicto alguno).

Finalmente, creo mi pull request, dándole click a “**Create Pull Request**”.

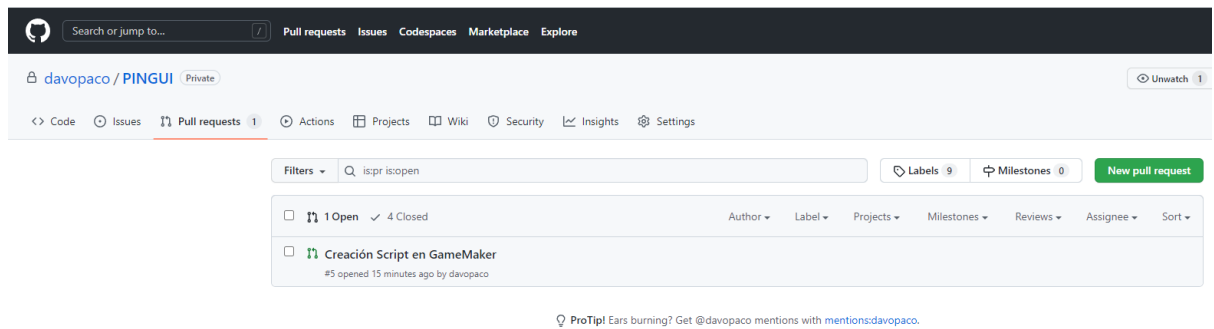
10. Ya que se ha creado el **Pull Request**, ahora se abre una especie de foro, en la que los contribuidores pueden comentar. Igualmente, si necesito hacerle **merge** de una, lo puedo hacer ya que no hay conflictos y con eso queda listo en el “**main**” para que todos lo vean.



ACLARACIÓN: Aquí aparece que Merging is blocked, porque es una regla que se le puso a las ramas y no permite que se haga, eso; sin embargo, eso ya queda cambiado cuando revisen este documento.

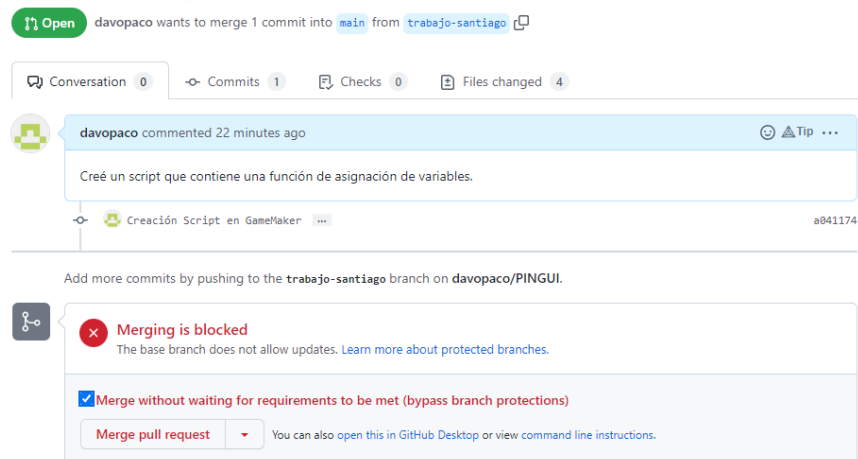
2.3. CÓMO HACER UN MERGE DE UN PULL REQUEST

En caso de que no seamos nosotros el que hicimos el **Pull Request**, si no lo hizo otra persona, nos dirigimos al apartado de “**Pull requests**” en **GitHub** (navegador) y ahí encontraremos todos los pull requests que se quieren hacer en el repositorio, en este caso el de “**Creación Script en GameMaker**”:

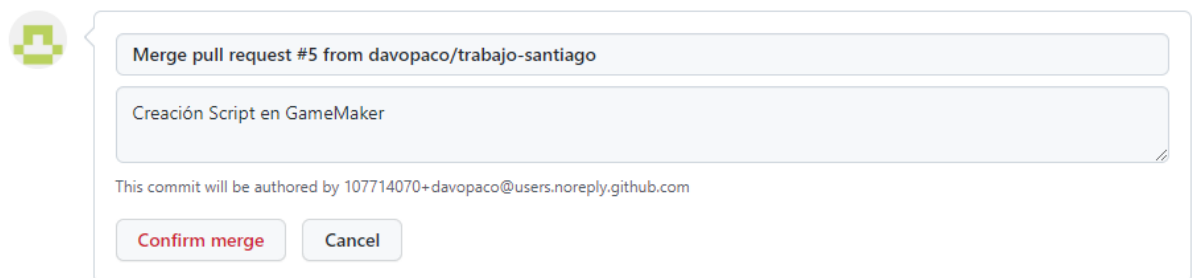


Le damos click a ese y lo revisamos.

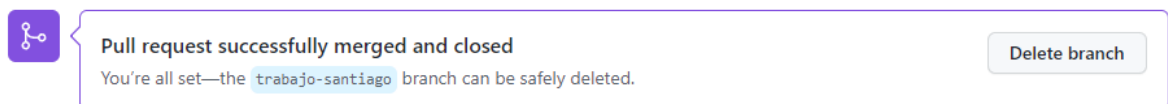
Creación Script en GameMaker #5



Aparece, nuevamente, esta pantalla que se comentó en el punto 10 de la sección 1.1. y le damos click a “Merge pull request”.



Confirmamos el **merge** dando click a “Confirm merge” y voilá ya quedó unido el proyecto.







Finalmente, el **merge** se realizó y si queremos podemos eliminar la rama con “Delete branch”.

3. DISTRIBUCIÓN DEL REPOSITORIO: PINGÜI

De manera inicial se encuentran cuatro carpetas: “**DOCUMENTATION**”, “**PINGUI**”, “**RESOURCES**”, “**VERSIONS (non-source-controlled)**”; y tres archivos: “.gitattributes”, “**EXPLICACIÓN GITHUB.pdf** (ESTE DOCUMENTO)” y “**REGISTRO CAMBIOS.md**”. A continuación, los explico detalladamente.

3.1. CARPETA DOCUMENTATION

Esta carpeta contiene TODOS AQUELLOS archivos que son la documentación de nuestro proyecto (PINGÜI). **Ejemplo:** Especificación de Requerimientos y Formulación del Proyecto.

| | | | | |
|---|--|------------------|---------------------|--------|
|  | Esepcificación de Requeriminetos v1.0.do... | 19/03/2023 11:07 | Microsoft Word D... | 635 KB |
|  | Especificación de Requerimientos vFinal.d... | 19/03/2023 11:07 | Microsoft Word D... | 344 KB |
|  | Formulación de Proyecto v1.0.pdf | 19/03/2023 11:07 | Documento PDF | 864 KB |
|  | LINKS DE DOCUMENTACIÓN (LATEX).txt | 19/03/2023 11:07 | Text Document | 1 KB |


3.2. CARPETA PINGUI

Esta carpeta contiene el **proyecto de GameMaker** y es ahí donde se trabaja en conjunto para realizar todo con respecto a la IDE.

| | | | | |
|---|-----------------------|------------------|----------------------|------|
|  | datafiles | 18/02/2023 20:10 | File folder | |
|  | fonts | 18/03/2023 11:23 | File folder | |
|  | objects | 18/03/2023 11:24 | File folder | |
|  | options | 18/03/2023 11:24 | File folder | |
|  | rooms | 18/03/2023 11:24 | File folder | |
|  | scripts | 19/03/2023 11:24 | File folder | |
|  | sprites | 18/03/2023 11:24 | File folder | |
|  | Pingui.resource_order | 19/03/2023 11:26 | RESOURCE_ORDER Fi... | 5 KB |
|  | Pingui.yyp | 19/03/2023 11:26 | GameMaker Project | 7 KB |





3.3. CARPETA RESOURCES

Esta carpeta contiene todo lo que a **recursos** se refiere. Recursos tales como imágenes; documentos extras: storyboard, cronograma; archivos html; música; pixil de PixelArt; entre otros.

| | | |
|--|------------------|-------------|
|  DOCEXTRA | 18/03/2023 12:06 | File folder |
|  FUENTES | 19/03/2023 11:07 | File folder |
|  IMÁGENES | 19/03/2023 11:07 | File folder |
|  LOCAL PACKAGES (.ymps) | 19/03/2023 11:07 | File folder |
|  PHOTOSHOP | 19/03/2023 11:07 | File folder |
|  PIXIL | 19/03/2023 11:07 | File folder |

3.4. CARPETA VERSIONS (non-source-controlled)

Esta carpeta contiene versiones de **PINGÜI** que trabajamos antes de empezar a manejar el repositorio. Por lo tanto, **NO MODIFIQUEN NADA ACÁ**, puesto que estas versiones están ahí solo como para saber qué teníamos antes.

| | | | |
|--|------------------|----------------|-------|
|  Diseño_pingui | 28/02/2023 09:08 | File folder | |
|  Main Menu | 18/03/2023 11:03 | File folder | |
|  Pinguiv1.9 | 18/03/2023 08:56 | File folder | |
|  Diseno_pingui2.rar | 28/02/2023 09:03 | WinRAR archive | 59 KB |

3.5. ARCHIVO .gitattributes

Este archivo es propio de **GitHub** y por lo tanto no se toca. Si necesitan explicación, son los atributos que el repositorio tiene, como por ejemplo poder bloquear que no se registren cambios de cierto tipo de archivos.

3.6. ARCHIVO “REGISTRO CAMBIOS.md”

Este archivo es muy importante de leer y modificar. Acá vamos a registrar las versiones/cambios de PINGÜI, los cambios en las carpetas del repositorio y agregado de archivos. Dentro del mismo pueden encontrar cómo se registran los cambios.

VERSION CAMBIOS

VERSION 2.0

ROOMS

- mainmenu: El Menú Principal con su respectivo fondo y los botones tipo objeto que ya pueden ser usados para linkear otros rooms.
- GameOver: El menu cuando el juego de "Incremento" se pierde.
- JueguitRecolectorMonedas: Juego de "Incremento".
- Room3: Pingüi con sus animaciones.

nuevo

-

OBJECTS

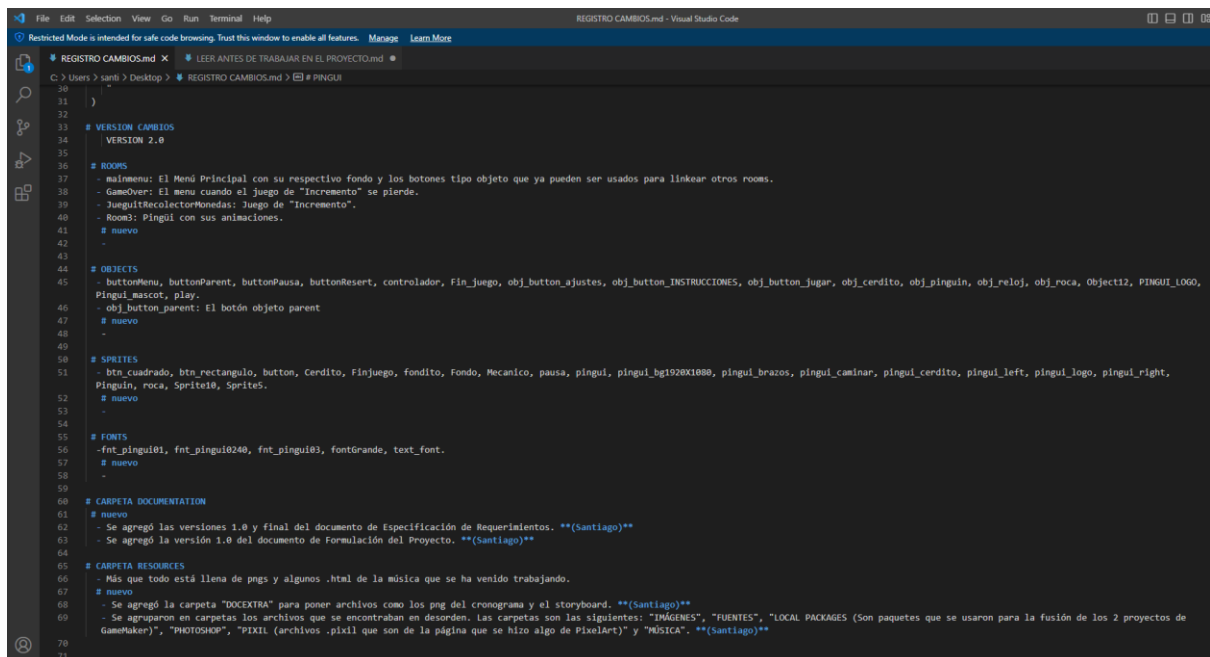
- buttonMenu, buttonParent, buttonPausa, buttonReset, controlador, Fin_juego, obj_button_ajustes, obj_button_INSTRUCCIONES, obj_button_jugar, obj_credito, obj_pinguin, obj_reloj, obj_roca, Object12, PINGUI_LOGO, Pingui_mascot, play.
- obj_button_parent: El botón objeto parent

nuevo

-

SPRITES

La anterior imagen es la vista en **GitHub**, pero para modificar este archivo, pueden usar su **editor de texto de preferencia**. En mi caso yo uso **Visual Studio Code**.



```
File Edit Selection View Go Run Terminal Help
Restricted Mode is intended for safe code browsing. Trust this window to enable all features. Manage Learn More

REGISTRO CAMBIOS.md x LEER ANTES DE TRABAJAR EN EL PROYECTO.md
C:\Users> santi > Desktop > REGISTRO CAMBIOS.md > # PINGUI

30
31
32
33 # VERSION CAMBIOS
34 VERSION 2.0
35
36 # ROOMS
37 - mainmenu: El Menú Principal con su respectivo fondo y los botones tipo objeto que ya pueden ser usados para linkear otros rooms.
38 - GameOver: El menu cuando el juego de "Incremento" se pierde.
39 - JueguitRecolectorMonedas: Juego de "Incremento".
40 - Room3: Pingüi con sus animaciones.
41 # nuevo
42 -
43
44 # OBJECTS
45 - buttonMenu, buttonParent, buttonPausa, buttonReset, controlador, Fin_juego, obj_button_ajustes, obj_button_INSTRUCCIONES, obj_button_jugar, obj_credito, obj_pinguin, obj_reloj, obj_roca, Object12, PINGUI_LOGO,
46   Pingui_mascot, play.
47 - obj_button_parent: El botón objeto parent
48 # nuevo
49 -
50
51 # SPRITES
52 - btn_cuadrado, btn_rectangulo, button, Credito, Finjuego, fondito, Fondo, Mecanico, pausa, pingui, pingui_bg1920X1080, pingui_brazos, pingui_caminar, pingui_credito, pingui_left, pingui_logo, pingui_right,
53   Pinguin, roca, Sprite10, Sprite5.
54 # nuevo
55 -
56
57 # FONTS
58 -fnt_pinguil01, fnt_pinguil0240, fnt_pinguil03, fontGrande, text_font.
59 # nuevo
60 -
61
62 # CARPETA DOCUMENTATION
63 # nuevo
64 - Se agregó las versiones 1.0 y final del documento de Especificación de Requerimientos. **(Santiago)**
65 - Se agregó la versión 1.0 del documento de Formulación del Proyecto. **(Santiago)**
66
67 # CARPETA RESOURCES
68 - Más que todo está llena de pngs y algunos .html de la música que se ha venido trabajando.
69 # nuevo
70 - Se agregó la carpeta "DOCEXTRA" para poner archivos como los png del cronograma y el storyboard. **(Santiago)**
71 - Se agruparon en carpetas los archivos que se encontraban en desorden. Las carpetas son las siguientes: "IMÁGENES", "FUENTES", "LOCAL PACKAGES (Son paquetes que se usaron para la fusión de los 2 proyectos de
  GameMaker)", "PHOTOSHOP", "PIXIL (archivos .pixel que son de la página que se hizo algo de PixelArt)" y "MÚSICA". **(Santiago)**
```