

Proyecto 3

Advice Maquine

Taller de Programación

Estudiantes:

Carlos Andrés Mata Calderón

Jose Ignacio Calderón Díaz

Carne:

2019033834

2019031750

Profesor:

Jeff Smith

I Semestre, 2020

Contentidos

[Tabla de Imágenes 3](#_Toc48320290)

[Introducción 4](#_Toc48320291)

[Descripción del problema 5](#_Toc48320292)

[Análisis de resultados 6](#_Toc48320293)

[Funcionamiento y distribución del código: 6](#_Toc48320294)

[Bases de datos y mensajes: 6](#_Toc48320295)

[Base de ventas: 6](#_Toc48320296)

[Funciones administración: 7](#_Toc48320297)

[Contraseñas: 7](#_Toc48320298)

[Setea el idioma: 8](#_Toc48320299)

[Abrir Ventanas: 8](#_Toc48320300)

[Tienda: 8](#_Toc48320301)

[Animación: 9](#_Toc48320302)

[Ventanas: 11](#_Toc48320303)

[Explicación de la clase: 11](#_Toc48320304)

[Uso de GitHub 12](#_Toc48320305)

[Ventanas del proyecto: 13](#_Toc48320306)

[Reglas de Grupo y Roles 15](#_Toc48320307)

[Dificultades encontradas: problemas en el desarrollo y que se hizo para corregirlos 15](#_Toc48320308)

[Bitácora de actividades 16](#_Toc48320309)

[Estadística de tiempos: 18](#_Toc48320310)

[Conclusión 18](#_Toc48320311)

[Literatura o Fuentes Consultadas 19](#_Toc48320312)

# Tabla de Imágenes

[Ilustración 1: Diagrama de Clases de Message 9](#_Toc48320107)

[Ilustración 2: Diagrama de GitHub 10](https://d.docs.live.net/eede2584e5404a82/Documents/Manual_de_Usuario_3.docx#_Toc48320108)

[Ilustración 3: Ventana Idioma 11](#_Toc48320109)

[Ilustración 4: Ventana Machine 1 11](#_Toc48320110)

[Ilustración 5: Ventana Machine 2 11](#_Toc48320111)

[Ilustración 6: Message Ejemplo 11](#_Toc48320112)

[Ilustración 7: Contraseña 12](#_Toc48320113)

[Ilustración 8: Ventana Admin 1 12](#_Toc48320114)

[Ilustración 9: Ventana Admin 2 12](#_Toc48320115)

[Tabla 1 : Roles 13](#_Toc48320158)

[Tabla 2: Bitácora 14](#_Toc48320159)

[Tabla 3: Estadística de Tiempo 16](#_Toc48320160)

# Introducción

Una “vending machine” o máquina dispensadora, son máquinas automáticas que proporciona un producto a cambio de dinero. La primera máquina expendedora que se tiene registró es de una máquina expendedora de agua bendita esta fue creada por Héroe de Alejandría en el siglo I en Egipto romano. Sin embargo, las primeras máquinas expendedoras modernas fueron creadas a principios de la década de 1880 en Inglaterra. Con este nuevo tipo de máquinas se observó que estas tienen un gran éxito en áreas muy transitadas, recolectando así grandes ganancias.

En el Instituto Tecnológico de Costa Rica se pueden encontrar máquinas expendedoras de comida, cajeros automáticos, expendedoras de agua, etc. En este proyecto se propondrá hacer una máquina de consejos o “Advice Machine”. Con el fin que los estudiantes del ITCR, cuando quiera un consejo, dicho o chiste, y estos no quieran acudir a una persona. Ellos tengan la opción conseguirlo atraves de una máquina expendedora.

Cabe agregar, para elaborar dicha máquina se tiene que contar con un conocimiento básico de circuitos, el cual no poseemos por ahora. Como estudiantes que recién empiezan a estudiar ingeniería computadores, crearemos una simulación en Python del funcionamiento de dicha máquina. Y a su vez poner en práctica los conocimientos que hemos adquirido en el Curso de Taller de Programación. Sin más que agregar, se procederá a explicar el proyecto.

# Descripción del problema

Como anteriormente fue mencionada, se simulará atraves de Python el funcionamiento de dicha máquina.

Nuestra máquina contará con dos posibles idiomas, una en inglés y otra en español. También contará con mecanismo de devolver el vuelto y contará con sección de administrador. En la sección de administrador, se encuentran otras funciones como cambiar la contraseña, acceder los datos de la máquina, incluso también para poder apagar máquina y restablecer los datos.

Además de contar, con un apartado para simular dinero de la vida real, más adelante explicará al detalle este último.

Las librerías que se utilizarán para programar dicho proyecto serán:

* Tkinter: Será la base del proyecto.
* Pickle: Se usará para guardar datos cifrados.
* Random: Se utilizará para escoger de forma aleatoria los mensajes y el dinero del comprador.
* Time: Para controlar el tiempo de la animación.
* Pygame: Será útil para mostrar el mensaje en pantalla, y simular como si fuera un papel.
* Thread: Se usará en las animaciones en Tkinter.

Cabe agregar, que se utilizará el paradigma de programación de POO, para implementar los mensajes en el código.

# Análisis de resultados

## Funcionamiento y distribución del código:

El proyecto se encuentra en un solo archivo Python llamado “Machine”. A continuación, se dará una explicación de todas las funciones que almacena el código.

### Bases de datos y mensajes:

Función load\_base\_message  
  
Se encarga de cargar la base de datos de los mensajes en la simulación dependiendo del idioma que el usuario elija.  
  
Entrada: idioma seleccionado por el usuario  
Salida: la base de datos cargada  
Restricciones: ---  
  
Función data\_text\_funcion  
  
cambia el modo de impresión a pantalla.  
  
Entrada: ----  
Salida: load\_base\_message()  
Restricciones: ---  
  
Función data\_text\_funcion\_aux\_1  
  
Setea los datos en pantalla resumidos.  
  
Entrada: ---   
Salida: texto\_en\_imprimir\_en\_pantalla  
Restricciones: ---

Función data\_text\_funcion\_aux\_2  
Setea los datos en pantalla extendidos.  
  
Entrada: ---   
Salida: texto\_en\_imprimir\_en\_pantalla  
Restricciones: ---

### Base de ventas:

Función ventas\_read  
  
Se encarga de leer el .txt con las transacciones y así aumentar el número de transacción.   
  
Entrada: --  
Salida: Numero de transacción  
Restricciones: --

Función ventas\_load  
  
Se encarga de escribir cada transacción nueva en la base de datos.   
  
Entrada: número de transacción, el total de monedas pagadas, vuelto, fecha y hora  
Salida: Una nueva línea en la base de datos de ventas   
Restricciones: --

### Funciones administración:

Función login\_validation  
  
Se encarga de validar la contraseña guardada en el sistema con la digitada por el usuario.  
  
Entrada: texto de entrada, ventanas a destruir, canvas a trabajar  
Salida: acceso a la parte administrativa  
Restricciones: --

Función reset  
  
Se encarga se encarga de "resetear" los valores de la base de datos, dejándolas en ceros.  
  
Entrada: --  
Salida: base de datos desde cero  
Restricciones: --

Función turn\_off  
  
Ventana donde se le pregunta al administrado/ra si realmente desea apagar la maquina  
si la persona decide apagarla, el programa se finaliza  
  
Entrada: --  
Salida: Fin de la simulación   
Restricciones: --

### Contraseñas:

Función password\_validationRealiza la encriptación de la contraseña para así compararla con la guardad en el sistema  
y además se utiliza para cambiar la contraseña por una nueva.  
  
Entrada: cuadro de texto  
Salida: String con la contraseña codificada  
Restricciones: Lo símbolos y las mayúsculas no logran identificarse correctamente

Función save\_pass  
Se llama cuando el usuario quiere cambiar la contraseña, esta necesita de la validación de la contraseña anterior por seguridad.  
  
Entrada: cuadros de texto y canvas a trabajar  
Salida: Escritura de la nueva contraseña codificada  
Restricciones: --

### Setea el idioma:

Función set\_idi  
  
Se encarga de poner el idioma de la máquina.  
  
Entrada: idioma seleccionado por el usuario  
Salida: variables globales con el texto  
Restricciones: --

### Abrir Ventanas:

#### Funciones openAdmin, openMachine, openOff, openLogin

#### Estas cuatro funciones se encargan de abrir las ventanas correspondientes:

Entrada: ventana que se quiere cerrar

Salida: una nueva ventana abierta

Restricciones: --

#### Función destroy

Esta función se encarga de cerrar la ventana seleccionada.   
  
Entradas: ventana que se quiere cerrar  
Salidas: ventana cerrada  
Restricciones: --

### Tienda:

#### Función set\_shop

Escoge un mensaje aleatorio de alguna lista del tipo de mensaje e imprime en pantalla lo que debe.   
  
Entradas: entero (tipo de mensaje)  
Salidas: impresión cuánto dinero debe  
Restricciones: --

#### Función coins\_rest

Resta el dinero que se pagó.  
  
Entradas: entero (tipo de moneda), tk (ventana), tk (canvas)  
Salidas: paying()  
Restricciones: tipo(moneda) sea mayor a la cantidad de dinero del usuario o igual.

#### Función vuelta\_return\_aux

Suma el vuelto total que queda almacenado, y borra el vuelto temporal.  
  
Entradas: tk(canvas)  
Salidas: vuelto\_volar()  
Restricciones: --

#### Función vuelto\_volar

Quita la imagen si y solo si hay vuelto almacenado.  
  
Entradas:tk(canvas)  
Salidas: quita\_imagen  
Restricciones: vuelto\_almacenado

Función vuelto\_move

Pone en pantalla el vuelto.  
  
Entradas: tk (canvas)  
Salidas: pone la imagen de monedas de sobra  
Restricciones: funciona 1 que haya vuelto las demas no

### Animación:

Función paying  
  
Imprime en mensaje en pantalla si no hay deuda del pago.   
  
Entradas: tk (window), tk (canvas)  
Salidas: paying\_aux()  
Restricciones: di no paga la deuda no se ejecuta

#### Función paying\_aux

Hilo de animación del papel saliendo y sonido de impresora.  
  
Entradas: tk (window)  
Salidas: paying\_aux\_2()  
Restricciones: --

#### Función paying\_aux\_2

Animación de papel saliendo, crea un canvas cuadrado que aumenta de tamaño cada 0.5s.  
  
Entradas: Tk (window)  
Salidas: paying\_aux\_3()  
Restricciones: si aún está en la animación hace y presiona el mensaje para verlo en pantalla

#### Función paying\_aux\_3

Escoge un mensaje aleatorio de alguna lista del tipo de mensaje e imprime en pantalla   
lo que debe.  
  
Entradas: string()  
Salidas: impresión cuánto dinero debe  
Restricciones: cuando cierra el programa no tire error debido a   
 que está en bucle

#### Función paying\_aux\_4

Escoge un mensaje aleatorio de alguna lista del tipo de mensaje e imprime en pantalla lo que debe.  
  
Entradas: string()  
Salidas: impresión cuánto dinero debe  
Restricciones: cuando cierra el programa no tire error debido a   
 que está en bucle

#### Función text\_pygame

Imprime el texto en la superficie.  
  
Entradas: texto, fuente, color, superficie,   
 posicion X, posicion Y  
Salidas: impresion el texto en la superficie  
Restricciones: --

### Ventanas:

#### Funciones idioma\_screen, machine, activar\_admin, new\_password, admin

Estas son las funciones de las ventanas de Tkinter, donde se desarrolla la simulación

## Explicación de la clase:

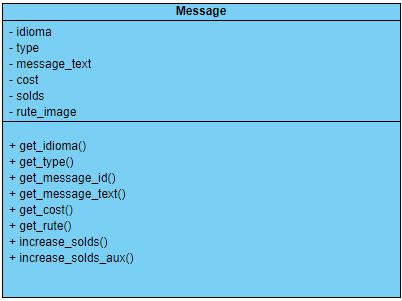


Ilustración : Diagrama de Clases de Message

Objeto Message:

Atributos:  
 -idioma: el idioma del mensaje  
 -type: tipo del mensaje  
 -message\_id: el id del mensaje  
 -message\_text: todo el texto del mensaje  
 -cost: costo del mensaje  
 -solds: cantidad de vendidos  
 -rute\_image: ruta de la imagen  
Métodos:  
 -get\_idioma:  
 -get\_type:  
 -get\_mesaage\_id:  
 -get\_message\_text:  
 -get\_cost:  
 -get\_rute  
 -increase\_solds:  
 Entrada: boolean, numero entero  
 Salida: llama a increase\_solds\_aux con la ruta en español y ruta en ingles  
 Restricciones: --

-icrease\_solds\_aux:  
 Entrada: entero, ruta\_txt  
 Salida: busca en todo el archivo el id y tipo de mensaje  
 renueva el valor correspondiente  
 vuelve a escribir todos los datos, pero con el costo del mensaje actualizado  
 Restricciones: --

Uso de GitHub:

Para el desarrollo del proyecto fue clave utilizar GitHub como plataforma de desarrollo colaborativo, ya que este nos permite modificar de una manera segura y en cualquier momento sobre el mismo código, en esta ocasión se logró mejorar el orden. En las siguientes imágenes se ve el diagrama que crea Pycharm del historial de branches y commits.

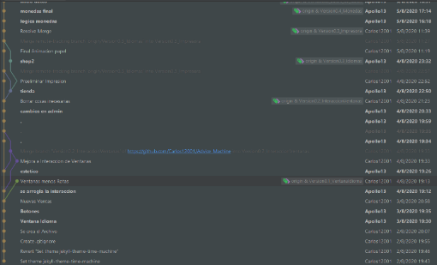
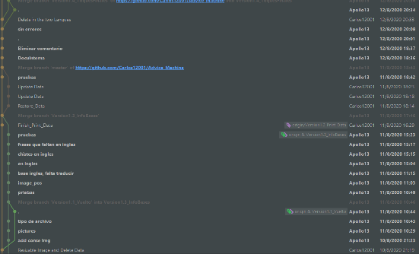
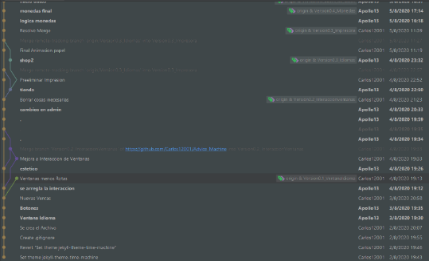
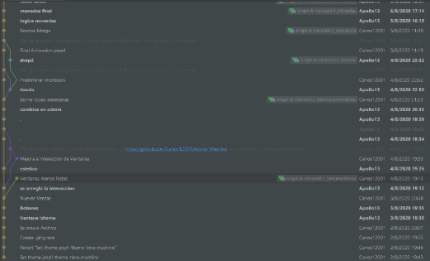
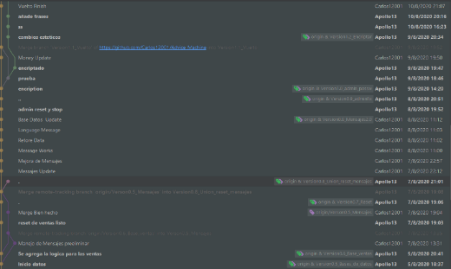
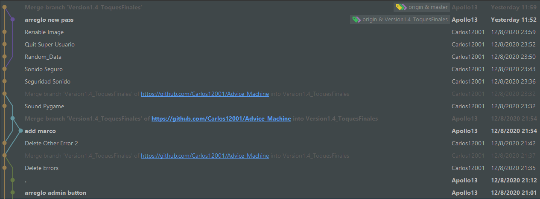


Ilustración : Diagrama de GitHub



## Ventanas del proyecto:

En este apartado se mostrará los Resultados del proyecto.

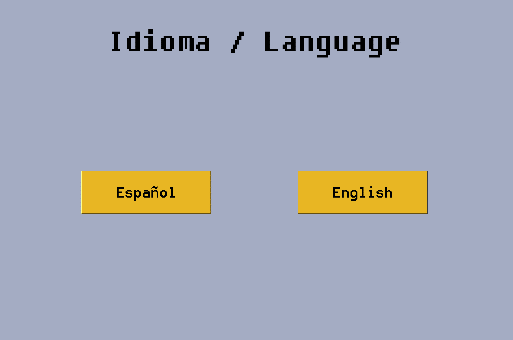


Ilustración : Ventana Idioma

La primera función que se llama la función de idioma\_screen.



Ilustración : Ventana Machine 1



Ilustración : Ventana Machine 2

Aquí se ejecuta la función machine, en la Ilustración 4 se ve como inicia la máquina y en la Ilustración 5 se muestra cómo funciona la máquina.



Ilustración : Message Ejemplo

La Ilustración 6 muestra un mensaje de ejemplo de los 50 que hay en total para cada idioma.



Ilustración : Contraseña

Cuando se presione el botón de admin, se ejecutará una mini ventana (Ilustración X) que se encuentra en la función login (). Esta se encargará de verificar que el usuario intente entrar al reporte de ventas sólo sea un administrador.

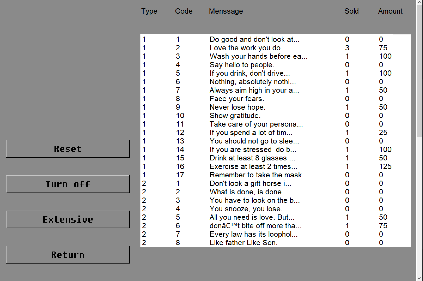


Ilustración : Ventana Admin 1

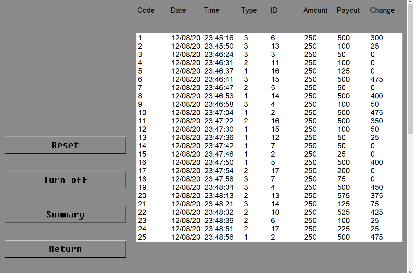


Ilustración : Ventana Admin 2

Esta es la ventana de administrador, se ejecuta con la función admin(). En esta ventana de administrador aquí si pude ver todos los reportes de ventas de la máquina en resumen (Ilustración X) y después de presionar el botón de extenso muestra todas las transacciones que realizado la máquina (Ilustración X). En esta misma ventana, además de ver las ventas de la máquina, también se puede borrar todos los datos de las ventas y se puede apagar la máquina.

# Reglas de Grupo y Roles

Al desarrollar este proyecto cada integrante fue parte clave para su éxito. No se definieron reglas como tal, aun así, se puede rescatar lo siguiente:

Cuando ocurriera algún problema se comunica al otro compañero.

Respeto al horario y horas destinadas a otros cursos.

Utilizar GitHub para cualquier cambio, indicando un nuevo *Branch* como VersionX\_” NOMBRE”.

En un grupo de trabajo es importante definir los roles de cada integrante para si maximizar la productividad y ser eficientes en cuestión del tiempo. Así que en la siguiente tabla se indica los roles de cada programador:

Tabla : Roles

|  |  |
| --- | --- |
| Nombre | Rol |
| Carlos Calderón Mata | * Animación * Base de Datos de Fortuna * Funcionamiento del Mensaje * Vuelto del Usuario * Imprimir Base de Datos Ventas en Pantalla * Documentador |
| Ignacio Calderón Díaz | * Dinero del Usuario * Contraseña * Base de Datos Ventas * Reset * Añadir más mensajes de la Base de Datos de Fortuna * Video explicativo * Documentador |

\*El trabajo se realizó de manera equitativa y colaborativa

# Dificultades encontradas: problemas en el desarrollo y que se hizo para corregirlos

Al desarrollar la contraseña para la ventana de administrador, la contraseña se guardó con serialización. Al abrir el archivo, se puede abrir con el bloc de notas, al hacer eso deja a la vista la contraseña. Para solucionar este problema se procedió a encriptar la contraseña del administrador, para evitar posibles fraudes.

Por otra parte, se tuvo un problema con la creación de la ventana del mensaje después de ser animado, ya que, al parecer tkinter cuenta con una incompatibilidad al destruir más de 3 ventanas e importar una imagen a la tercera, tkinter se cae y deja de funcionar. Para solucionar esto se soluciona utilizando pygame en la tercera ventana.

Además, a la hora de hacer hilos, python nos generó problemas al pasar argumentos de a las funciones del hilo. Después de consultar con un tutor de la carrera, se logra corregir exitosamente.

Finalmente, a la hora de mostrar los datos en la ventana de administrador, los datos salían desordenados, ya que en él .txt los espacios variaban dependiendo de la longitud del mensaje. Para solucionar esto se coloca un nuevo formato utilizando \t que significa una sangría. Así que al separar todas las palabras a una misma distancia.

# Bitácora de actividades

Tabla : Bitácora

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Actividad | Tipo | Responsable | Duración | Descripción |
| Creación ventana Idioma | Programación | Ignacio Calderón y Carlos Mata | 40 min | Se crea la primera ventana y los botones que colocan el idioma. |
| Se agrega un poco de diseño y se comienza la tienda | Programación | Ignacio Calderón | 3 horas | Se comienza la lógica de la tienda. |
| Se programa los primeros Mensajes | Programación | Carlos Mata | 6 horas | Se realiza la clase con todos los atributos y métodos. Además, crear el txt y el método de almacenamiento de este mismo. |
| Monedas e inicio de bases de datos venta | Programación | Ignacio Calderón | 4 horas | Se finaliza la simulación de las monedas y se inicia la estructura de las bases de datos. |
| Canvas evento clic | Investigación | Carlos Mata | 2 horas | Se investiga como conseguir el evento mouse en un canvas. |
| Se programa la animación | Programación | Carlos Mata | 4 horas | Se realiza la animación de salida del mensaje. |
| Ventana administrador y contraseña | Programación | Ignacio Calderón | 4 horas | Se crea las funciones de reset(), encriptación, stop, así como la validación de la contraseña. |
| Monedas Mejora | Programación | Carlos Mata | 2 horas | Se mejoró el uso de las monedas en la máquina. Debido a que tenía problemas, como dinero negativo, etc. |
| Vuelto | Programación | Carlos Mata | 4 horas | Se implementó un vuelto si llega introducir más dinero del que debía. |
| Búsqueda y compresión de frases e imágenes | Investigación | Ignacio Calderón | 5 horas | Se da una búsqueda de frases que encajen en español e inglés. Además de la compresión de imágenes. |
| Mejora e impresión de Base de Datos | Programación | Carlos Mata | 8 horas | Se imprimió todas las bases datos. Sin embargo, se tuvo que cambiar algunas funciones para que esta funcionará de forma correcta. |
| Documentación Interna | Documentación | Ignacio Calderón y Carlos Mata | 2 horas | Se realiza la documentación interna de las funciones y clases. |
| Mejoras al proyecto final y corrección de diversos errores pequeños | Programación | Carlos Mata | 5 horas | Se revisó de forma rigurosa el funcionamiento de muchas funciones, debido a que había problemas al momento de comprar un mensaje mientras se accedía la ventana de admin. |
| Documentación Externas | Documentación | Ignacio Calderón y Carlos Mata | 10 horas | Se realiza el documento de Word con la explicación del código. |
| Evento quit() | Investigación | Ignacio Calderón y Carlos Mata | 30 min | Se investiga como accionar una función al cerrar una ventana con la equis |
| Video | Filmación | Ignacio Calderón | 5 horas | Se filma el video grabando la pantalla de la computadora y se procede a editarlo y subirlo a YouTube. |

# Estadística de tiempos:

Tabla : Estadística de Tiempo

|  |  |
| --- | --- |
| Actividad | Horas |
| Programación | 41.3 |
| Investigación | 7.3 |
| Grabar video | 5 |
| Documentación | 12 |

# Conclusión

Ignacio Calderón

En resumen, el desarrollo de la “advice machine” me enseño lo importante que puede llegar a ser la programación en el día a día. Ver cómo puede facilitar a los humanos las tareas cotidianas para muchos. Si se hiciera este proyecto en la vida real se puede agregar general un Excel con información extra, datos accesibles por internet, entre otras cuantas de opciones más.

Por otra parte, pienso que estos consejos pueden salvar vidas, si se hace una selección más profesional. Al elegir frases que los expertos digan que afectan de manera positiva a los humanos, muchas de ellas pueden llegar a hacer un cambio a la sociedad.

Carlos Mata

Con este proyecto, se demuestra como nuestros conocimientos de programación mejoraron mucho a comparación cuando se inició el curso de Taller de Programación. Debido a que, en proyectos pasados, había algo de desorden en el código, en cambio en este tercero notamos como la distribución del tiempo y el uso de buenas de programación, dieron como resultado con código y un proyecto más ordenado.

Otro punto, importante en destacar que en este proyecto como ya se tenía un vasto conocimiento en el uso de la librería de Tkinter, se pudo invertir el tiempo que se tardaría en investigar en mejorar el programa.

Para finalizar, creo lo más importante que se puedo resaltar es el trabajo en equipo con mi compañero Ignacio, ya que, en el desarrollo de este proyecto tuvimos una excelente comunicación y división del trabajo. Y me parece lo más importante que se aprendió en el desarrollo de este proyecto.

# Literatura o Fuentes Consultadas

Frases:

A. 2012,06. Ejemplo de Ejemplo de dichos o refranes. Revista Ejemplode.com. Obtenido de <https://www.ejemplode.com/12-clases_de_espanol/2194-ejemplo_de_ejemplo_de_dichos_o_refranes.html>

Antena, 3. (2019). 99 chistes cortos que te harán llorar de risa al instante. Recuperado de <https://www.antena3.com/liopardo/memes/chistes-cortos-que-haran-llorar-risa-instante_201905105cd5aa820cf26b6ffc6a0941.html>

Programación:

TkInter para python: Barra para desplazamiento vertical/scrollbar (básico). cctmexico (Director). (2017, may).[Video/DVD]