PRIMER ACERCAMIENTO A SERVIDORES Y BASE PARA PROYECTO

Branko Aguirre Esquivel, Juan pablo Carvajal Rojas, Edward - La cabra.

*Universidad del Quindío.*

*Armenia, Colombia.*

*Abstract*—El presente informe tiene como objetivo exponer el trabajo realizado para la sección “Lab 2.”, para la comprensión correspondiente de los sistemas de servidores así como la interacción de archivos entre sí y un esquema base para la generación del proyecto propuesto.

**Palabras claves: Archivos, funciones, servidor, ciclo for, archivo de texto, variable, return y parámetros.**

# INTRODUCCIÓN

En este espacio se requiere configurar el sistema de ingreso para los usuarios, acceder con un usuario y contraseña, la recolección de puntos así como la diferenciación de usuarios y si están en juego o han cerrado sesión, por lo que el docente se asegura de proporcionar un código base en el cual están ciertas partes fundamentales para realizar un acercamiento a la comprensión del comportamiento entre el código, servidor y usuario, dándonos como experimentación la configuración de archivos distintos, realizar un llamado de todos en un nuevo archivo y configurar lo anteriormente acordado.

1. OBJETIVOS

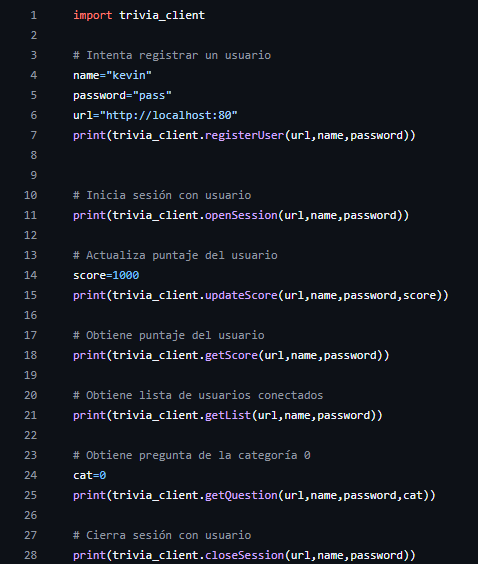
* Comprender las características de los archivos y servidores.
* Experimentar y analizar con el uso de información entre archivos.
* Crear una base experimental para el proyecto.
* Reconocer las optimizaciones posibles para un programa eficiente y carente de bugs.

# MÉTODOS E INSTRUMENTACIÓN.

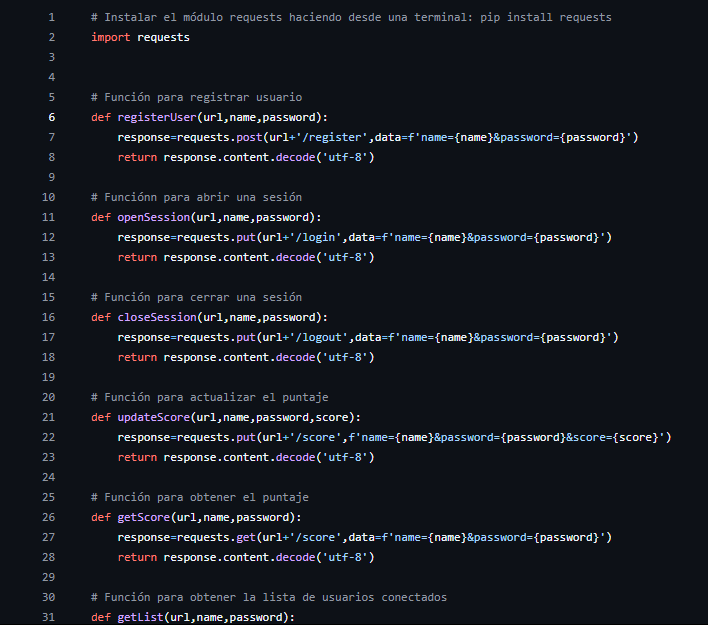
Se nos plantea la problemática de estructurar correctamente las funciones para satisfacer la parte técnica del manejo de usuarios para el acceso a la plataforma y la correcta repartición de puntos.

El docente se encarga de compartir ciertos archivos que contienen el código base donde se correrá el concepto final de esta práctica.

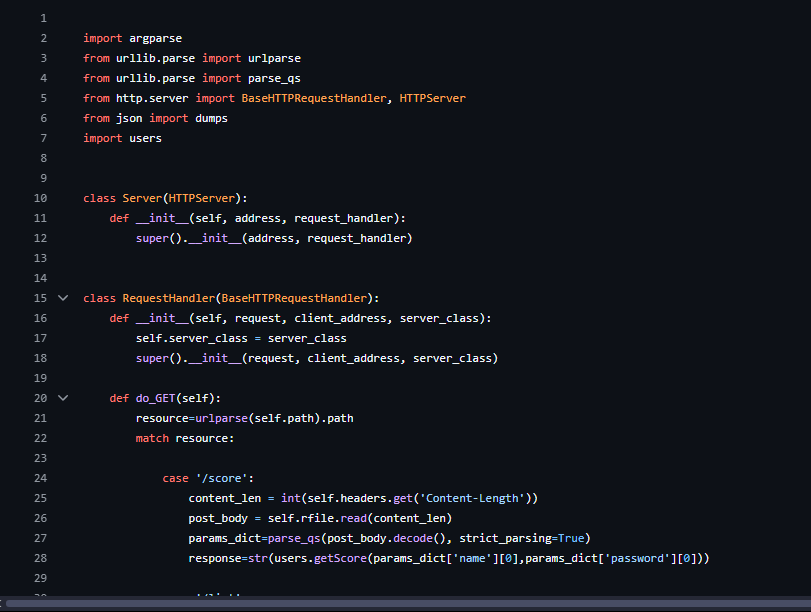
# Primer archivo:



Segundo archivo:

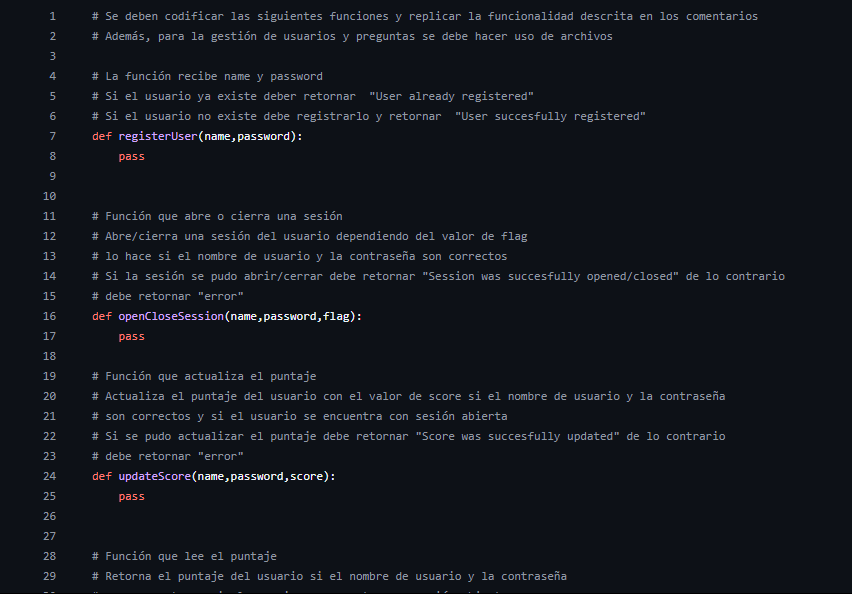


Tercer archivo:

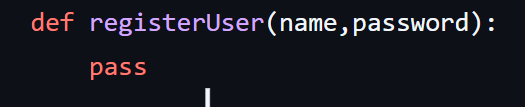


Finalmente, en este último archivo, se evidencian las funciones que van a dar a cabo en las codificaciones recientes:

Cuarto archivo:

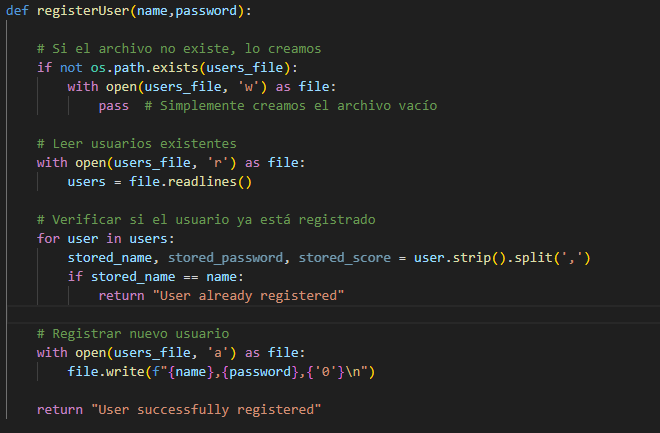


Yendo por partes, en la primer función:



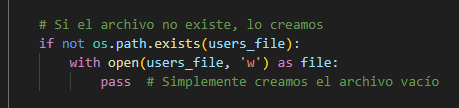
Primera función.

Se pide hacer uso del conocimiento adquirido para dar un registro exitoso a un usuario nuevo o, en dado caso que no esté registrado, registrarse correctamente y anexar al archivo correspondiente, de esta forma, se procede de la siguiente manera:



Función 1 correctamente codificada.

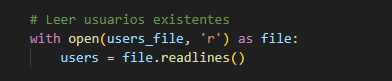
En este primer acercamiento, se recibe como parámetros “name” y “password” que será el nombre y la contraseña del usuario que se quiera o esté registrado, en dado caso de no existir el documento donde se guardará en texto los datos requeridos, se creará en las primeras líneas de esta función.



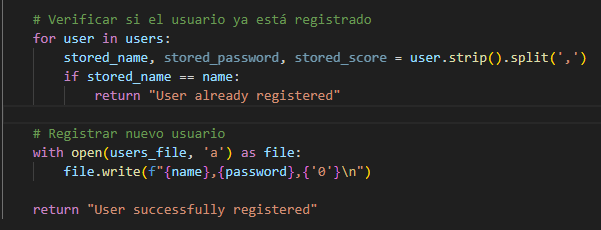
Creación del archivo en caso de no existir.

.

Seguido de esto, se dará un muestreo en cada línea del archivo donde contendrá los datos para verificar todos los usuarios que hayan con with () as ():

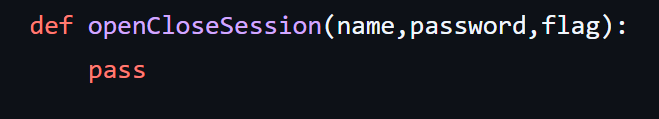


Y, finalmente, se verificará si el usuario es una persona que se registra por primera vez o un jugador con verificado, con un ciclo for, en rango de los usuarios anteriormente leídos por el muestreo se compara el nombre del parámetro “name” con los nombres dados en el archivo de texto, en caso de coincidir se imprimirá en pantalla que el usuario está satisfactoriamente registrado, de no haber ningún nombre relacionado a este contenido del parámetro, se añadirá en el documento con su nombre, contraseña y respectivo puntaje que, ya que está iniciando un registro, se pondrá en “0”.

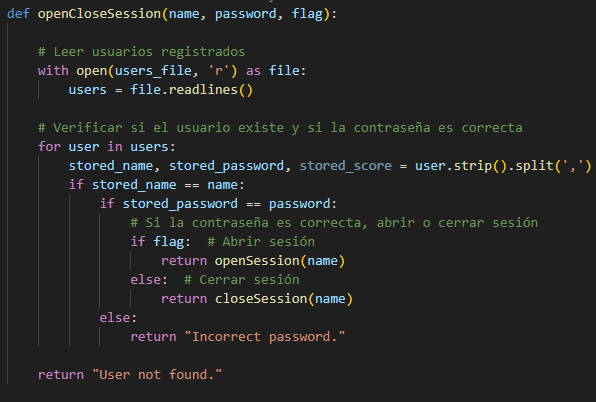


Registro y/o comprobante de ingreso.

En la segunda función:

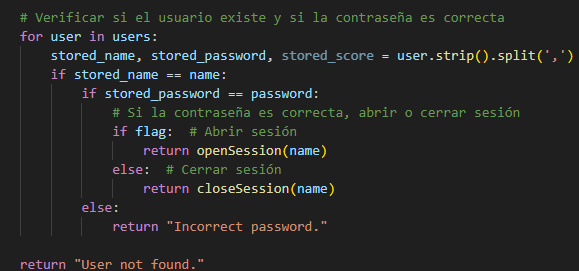


En este caso se nos pide configurar el cerrar o abrir la sesión dependiendo de la veracidad de los datos del usuario y su “flag” interpretado de esta forma:



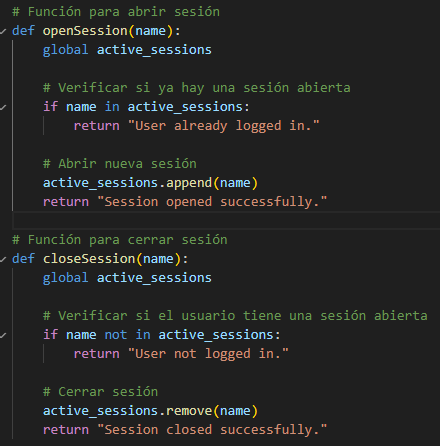
Segunda función correctamente codificada.

En este caso, reiterando como la función anterior, se realiza un muestreo de los usuarios registrados en el archivo de texto, para proceder a comparar con un ciclo for su nombre y contraseña, en caso de ser verdadera esta comparación retornara según su “flag” el valor que de la función Open o Close session que analizaremos próximamente, en dado caso de no corresponder la contraseña se imprimirá en pantalla que es incorrecta, y de ser el nombre no correspondiente se imprimirá usuario no encontrado.



Ciclo for para comparaciones mejor detallado.

Ahora se da inicio a la codificación de las funciones Open y Close Session para controlar el la entrada y salida de un usuario.



Funciones para abrir y cerrar sesión.

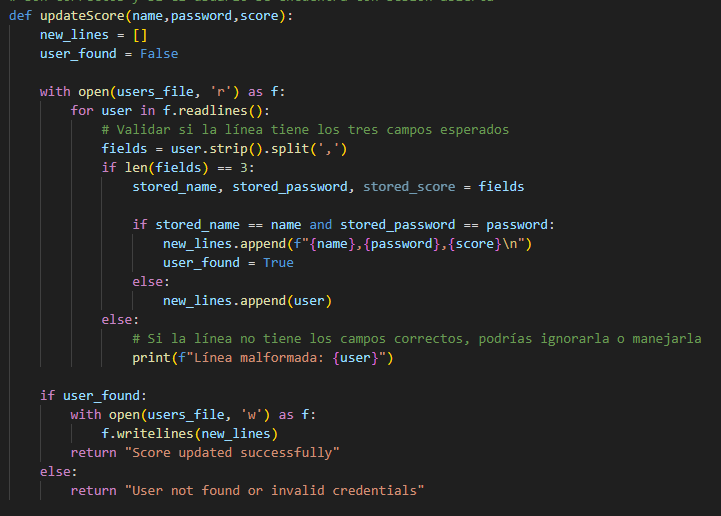
En este caso no hay una complejidad mayor, son 2 funciones simples para congeniar con la anterior y mejorar la legibilidad del código, en este caso se establece “active\_sessions” como variable global para que las funciones del código puedan acceder al valor de esta y modificarla, luego compara si el nombre tiene su sesión activa y, luego de hacer la comprobación, imprimirá en pantalla si está logueado o no, finalmente, dependiendo el escenario, si se quiere cerrar o abrir, se agrega el nombre o se remueve de la lista de activos y se imprime el mensaje avisando el cambio.

Siguiendo con las funciones, se nos presenta una función para actualizar la puntuación del jugador:



Actualizar puntaje.

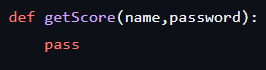
Se procede a realizar la codificación en base a lo pedido, obteniendo lo siguiente:



Función para la actualización del puntaje correcta.

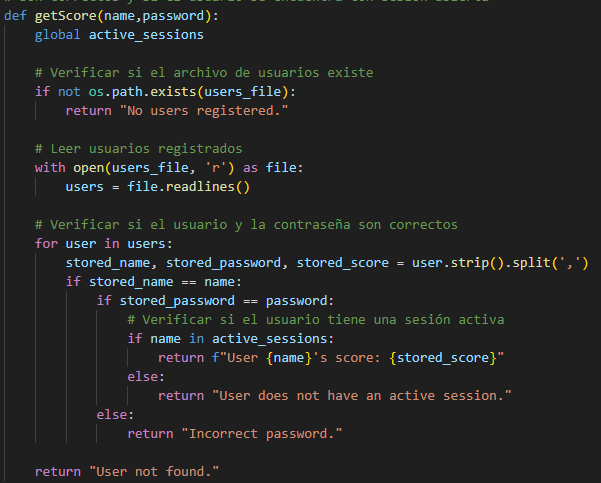
En este caso se obtienen como parámetros el nombre de usuario, su contraseña y su puntaje actual, se define “new\_lines” y “user\_found” de este modo, se procede a realizar las respectivas comparaciones, si en el momento del análisis los componentes del archivo no están separados correctamente como debería, se procede a realizar una advertencia del error cometido, si no pasa, entra a una comparación para verificar que el usuario y contraseña registrado este correcto, en caso de serlo se agrega los valores de cada concepto a la variable anteriormente declarada como “new\_lines” y se modifica la variable “user\_found” como verdadero, finalmente se realiza un análisis con if en la variable, “user\_found” ya que, al ser verdadera, significa que entró satisfactoriamente a la verificación de nombre y contraseña y de esta forma se sobreescribirá los datos sobre la ubicación en el archivo de texto, en caso de no ser verdadera, se imprimira que el usuario no se encontró o la contraseña es erronea.

Funcion obtener puntaje:

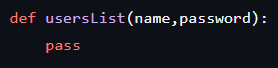


Leer el puntaje.

En esta función se nos solicita crear una forma de imprimir el puntaje del usuario utilizando como parámetros el nombre y la contraseña del mismo.

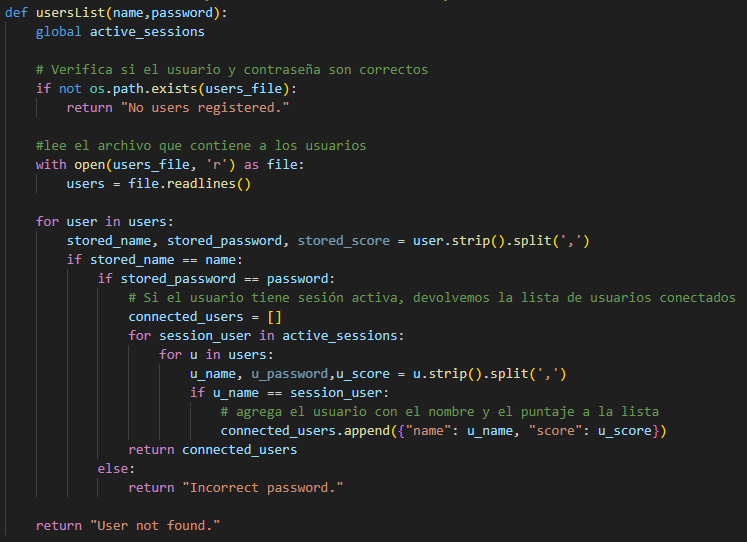
función correctamente estructurada para leer la puntuación.

Se inicia adquiriendo la variable global “active\_session” y se comprueba que efectivamente exista el documento con los usuarios, de no ser así se imprimirá que no hay usuarios registrados, seguido de esto, se realiza el muestreo de los usuarios en el archivo correspondiente y, similar a la función anterior, con un ciclo for en el rango de los usuarios correspondientes del archivo, se realiza la comparación del nombre y contraseña del usuario, en caso de ser verdadera la comparación la función validará que el usuario este con la sesion activa y, finalmente, retorna el nombre del usuario y su puntaje actualizado, en caso de no estar activa la sesión, tener mal la contraseña o no encontrar el nombre ingresado se le notificará al usuario para que verifique antes de volver a preguntar por su puntaje actualizado.  
  
Función para la lista de usuarios conectados:



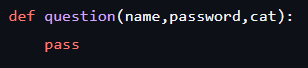
función para verificar los usuarios conectados.

En esta penúltima función se nos propone obtener la lista de usuarios actualmente conectados, de esta forma procedemos como se muestra a continuación:

Codificación correcta para la lista de usuarios conectados.

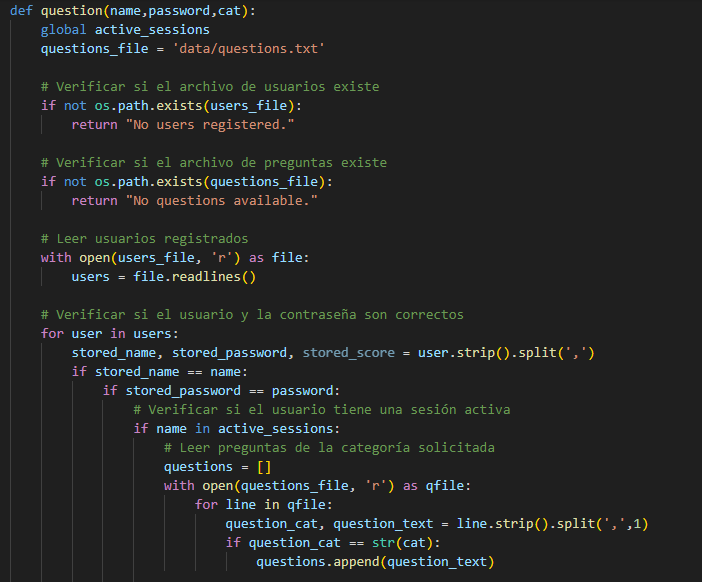
En los inicios se procede de manera idéntica a las funciones anteriores, comprobando la existencia del archivo de texto donde estará contenido los usuarios y se hará el muestreo de los datos de ellos, luego de esto, en un ciclo for, se realiza las debidas comparaciones para comprobar si el usuario tiene sus datos válidos y, en caso de serlo, entra a un for anidado donde evaluará si corresponde a la lista de sesiones activas y en caso de serlo lo agregara a la variable anteriormente definida como “connected\_users” con su nombre y su puntaje, excluyendo su contraseña y retornará la lista de usuarios conectados, también se tiene en consideración las posibilidades en funciones anteriores como el hecho de que algún dato sea invalidado, se imprimirá en pantalla la notificación del error.

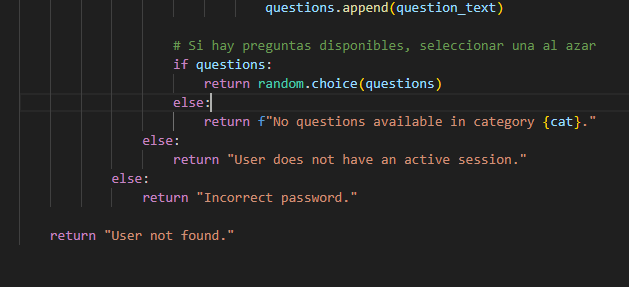
Para dar fin a esta práctica de laboratorio se da inicio con la función “Question”:



Función preguntas.

Finalmente, se nos pide realizar un código para que en dado caso que el usuario esté conectado y sus datos sean veraces, retorne una pregunta de una categoria “cat” y, en dado caso de que la veracidad no se cumpla, retorna error, de esta manera se da inicio a la codificación:

Parte 1

Parte 2

Esta función tiene como propósito verificar si tiene una sesión activa y, si es válido, retornar una pregunta al azar de una categoría específica. El proceso se basa en la existencia de dos archivos: el archivo que contiene los usuarios registrados y otro que almacena preguntas organizadas por categorías.

Se da inicio como funciones anteriores por lo que no se ahondará tan profundamente en estos conceptos, se verifica si los archivos existen, se hace el muestreo de usuarios registrados y se inicia con el reconocimiento del usuario en cuestión. Véase Parte 1.

En adición tenemos el archivo que contendrá las preguntas y el parámetro “cat” que filtra las preguntas para devolver solo aquellas que pertenecen a la categoría indicada.

con esto en cuenta y, seguido de comprobar la actividad del usuario en el momento, se hace un muestreo de las preguntas en el archivo correspondiente y se almacena en la variable questions, si existen preguntas se seleccionará una al azar con la función random.choice y se imprimirá la pregunta en pantalla, en caso de no haber preguntas, se avisará al usuario que no hay preguntas disponibles en la categoría seleccionada, y las condiciones básicas de las funciones anteriores, si el usuario tiene un dato erroneo o invalido se procederá a informar sobre este acontecimiento y no se imprimirá ninguna pregunta.

1. CONCLUSIONES
2. Se comprende correctamente las características de los archivos, y se interioriza de manera práctica el uso de estos para solución de problemas
3. Se realiza una base para el próximo proyecto.
4. Se formula y codifica de manera íntegra y óptima para cada problemática propuesta en el espacio de experimentación
5. Se especifica posibles errores y se explica qué solución se implementó para evitarse.