



Escuela de Ingenieria en Computacion  
Taller de programacion

# Proyecto 1

Kendal Chacon Chaves  
Carné:2024122026  
Profesor:Alexandre Brenes  
ALajuela  
12/4/2024

## 0.1. Resumen Ejecutivo

El enfoque para resolver el problema de representar un número entero en forma de fracción con restricciones específicas se basa en dos funciones principales: `CrearFracciones` y `AparicionesDeN`. El usuario primero proporciona el número de casos de prueba (T) y el número máximo de repeticiones de cualquier dígito en la fracción (R).

La función `CrearFracciones` genera fracciones a partir de números enteros, asegurándose de que el numerador y el denominador no excedan los 5 dígitos cada uno. Esto implica un manejo cuidadoso de los rangos y las combinaciones posibles para garantizar que se cumplan las restricciones establecidas.

Por otro lado, la función `AparicionesDeN` se encarga de verificar si una fracción generada cumple con la restricción de que ningún dígito se repita más de R veces en la fracción. Esto implica un análisis exhaustivo de cada fracción generada para contar las apariciones de cada dígito y compararlas con el valor R dado.

Durante el desarrollo del código, se realizaron múltiples ajustes y mejoras tanto en las funciones principales como en otros componentes del programa. Esto incluyó optimizaciones en los bucles y rangos utilizados, así como la gestión de errores y casos especiales que podrían surgir durante la ejecución del programa.

A pesar de los desafíos encontrados durante el desarrollo, los resultados finales del código fueron positivos, logrando cumplir con las restricciones establecidas y proporcionando una solución eficiente para el problema planteado.

## 0.2. Introduccion

Este documento tiene la finalidad de dar una justificacion de como fue generada la solucion, problemas que surgieron, estadisticas y demas elementos que fueron relevantes para el proyecto 1 de taller de programacion. El proyecto propuesto por los docentes encargados del curso de Taller de Programacion tenia el siguiente desarrollo; Se debe crear un codigo en Python3 que tuviera como finalidad encontrar un numero de 5 digitos que pudiera ser dividido por otros 5 digitos y diera como resultado un numero C que iba a ser un input que nosotros quisieramos que estuviese en el rango de (1,20). Como hasta aqui el problema se veia muy sencillo se le introdujeron las siguientes condicionantes:

- Se va a introducir un numero T el cual va a indicar la cantidad de casos que vamos a poner a prueba (los cuales pueden ir desde 1 hasta 100)
- Se va a introducir un numero R el cual va a limitar la cantidad de veces que cualquier digito puede aparecer entre los posibles 10 digitos de nuestra fraccion

### 0.3. solucion

Se nos presenta un problema, el cual nos pide representar un numero entero en forma de fraccion que contenga maximo 5 digitos en numerador y 5 digitos en el denominador, ademas de eso vamos a recibir un numero T que es equivalente a la cantidad de casos que vamos a validar y un R que nos dice el numero maximo de repeticiones de cualquier digito en la fraccion. Decidi en primera instancia guardar las fracciones generadas en una lista llamada fracciones y otra llamada fracciones vistas con el fin de verificar si una funcion ya fue generada para que no se convierta en un duplicado. Los denominadores los genere con un ciclo For con un rango desde 1000 hasta 100000. Luego para sacar el numerador este seria el equivalente a multiplicar el denominador por C. En el mismo codigo agregue una validacion extra que me preguntaba si el denominador era menor a 100000, si esto era correcto la fraccion seria agregada satisfactoriamente. Seguidamente tenemos la funcion para verificar cuantas veces aparece N en la fraccion tanto en numerador como denominador, la funcion mas viable que encontré fue mediante la utilizacion de diccionario en py. seguidamente utilicé division euclidea en 10 mientras el numerador o denominador fueron mayores a 10. La siguiente funcion unicamente organiza los inputs de T, C y R para cumplir con la normativa propuesta. Este bloque de código procesa una lista de casos de prueba, donde cada caso de prueba es un par de valores (c, r). Para cada caso de prueba, se generan fracciones utilizando la función crearfracciones(c, r). Si se generan fracciones, se itera sobre cada fracción en orden ascendente. Para cada fracción, se calcula el número de apariciones de cada dígito en el numerador y el denominador utilizando la función apariciones(num, den). Si todas las apariciones son menores o iguales a R se imprime la fraccion. Este codigo llevo muchisimo tiempo y muchisimas modificaciones, Esto debido a que durante el proceso se utilizaron muchas tecnicas y ideas diferentes que fueron descartadas mediante se fueron implementado ideas nuevas y el profesor fue dando pistas a lo largo de las clases. Por ejemplo un cambio que se hizo fue a la funcion principal de generar las fracciones debido a que al principio utilizabamos un ciclo for para generar el denominador y otro ciclo for para generar el numerador, dicha idea fue desechada debido a que se encontro una mejor que era encontrar el numerador mediante multiplicar C por el denominador. Otro cambio importante fue producto de una incorrecta interpretacion de las reglas del presente proyecto, Mi interpretacion fue que el numero R unicamente restringia las apariciones del digito C en la fraccion y resulta que el numero R era aplicado a cualquier digito presente en la fraccion.

## 0.4. resultados de las pruebas



## 0.5. conclusiones

En general me encuentro muy satisfecho con el resultado final de mi código, considero que el tiempo de ejecución es bastante bueno, las líneas de código son las justas y necesarias para su correcto funcionamiento. Hablando más sobre el proyecto me parece una propuesta bastante interesante principalmente por que me puso a pensar en maneras diferentes de como se ve una fracción, sumado a esto las condicionales propuestas que generaron un reto extra.

## 0.6. Aprendizajes

En primer lugar aprendí que debo indagar más acerca de este lenguaje de programación debido a que contiene muchos elementos que no conozco y que podrían ser demasiado útiles en un futuro, uno de esos elementarios podrían ser los diccionarios, la función `set()`. otra cosa que aprendí es que debo buscar siempre enfoques diferentes de realizar una tarea y no cerrarme tanto en uno solo.

## 0.7. Bibliografía

- “Listas en Python,” El Libro De Python. <https://ellibrodepython.com/listas-en-python>
- “Función map,” CódigoFacilito. <https://codigofacilito.com/articulos/funcion-map>
- “Sorting techniques,” Python Documentation. <https://docs.python.org/3/howto/sorting.html>
- Mingorafa, “Divisibilidad – Las matemáticas al alcance de todos,” Las Matemáticas Al Alcance De Todos. <https://www3.gobiernodecanarias.org/medusa/ecoblog/dharnav/categoria/divisibilidad/>