

# Proyecto de Backtracking: Solucionador de Cubo Rubik 2x2

Julio Pérez

Abril - Julio 2024

## 1. Introducción

El propósito de este proyecto es proporcionar una comprensión mas detallada del algoritmo de búsqueda en anchura (Breadth-First Search, BFS) y del concepto de grafos implícitos. Para lograr este objetivo, se desarrollará un solucionador para el cubo de Rubik 2x2.

## 2. Implementación

La implementación se dividirá en varias etapas:

### 2.1. Representación del Estado del Cubo Rubik 2x2

Inicialmente, los estudiantes deben diseñar una estructura de datos adecuada para representar el estado de un cubo Rubik 2x2. Este estado debe reflejar los seis colores del cubo: blanco, amarillo, verde, rojo, naranja y azul.

### 2.2. Clase RubikCube

Se requiere la creación de una clase denominada `RubikCube` que encapsule el estado del cubo y permita la aplicación de movimientos válidos. Esta clase debe incluir métodos para realizar rotaciones y transiciones de estado, respetando las reglas del cubo Rubik.

### 2.3. Clase RandomRubikCubeGenerator

Se debe implementar una clase `RandomRubikCubeGenerator` cuya función será generar instancias de `RubikCube` con estados aleatorios válidos. Esta clase debe asegurar que los estados generados sean alcanzables desde una configuración resuelta mediante una serie de movimientos válidos.

### 2.4. Clase RubikCubeSolver

Finalmente, se debe desarrollar una clase `RubikCubeSolver` que, dada una instancia de `RubikCube` en cualquier configuración, calcule la secuencia óptima de movimientos para resolver el cubo utilizando el algoritmo BFS. Esta clase debe ser capaz de devolver la solución en la menor cantidad de pasos posible.

### 3. Entrega

El proyecto se realizará en equipos de 2 ó 3 integrantes y deberá ser alojado en un repositorio de GitHub. El repositorio debe contener, al menos, las siguientes clases en archivos individuales con el formato `<clase>.py`. Además, se debe incluir un archivo `README.md` que describa detalladamente la representación de los estados, transiciones y el funcionamiento del solucionador. Este archivo debe identificar a todos los miembros del equipo con sus nombres y números de matrícula.

#### 3.1. Pruebas

Es importante que el código sea probado utilizando `pytest` o `unittest` para garantizar su correcto funcionamiento.

#### 3.2. Fecha de Entrega

La fecha límite para la entrega del proyecto es el miércoles 3 de julio de 2024 a las 11:59 pm.