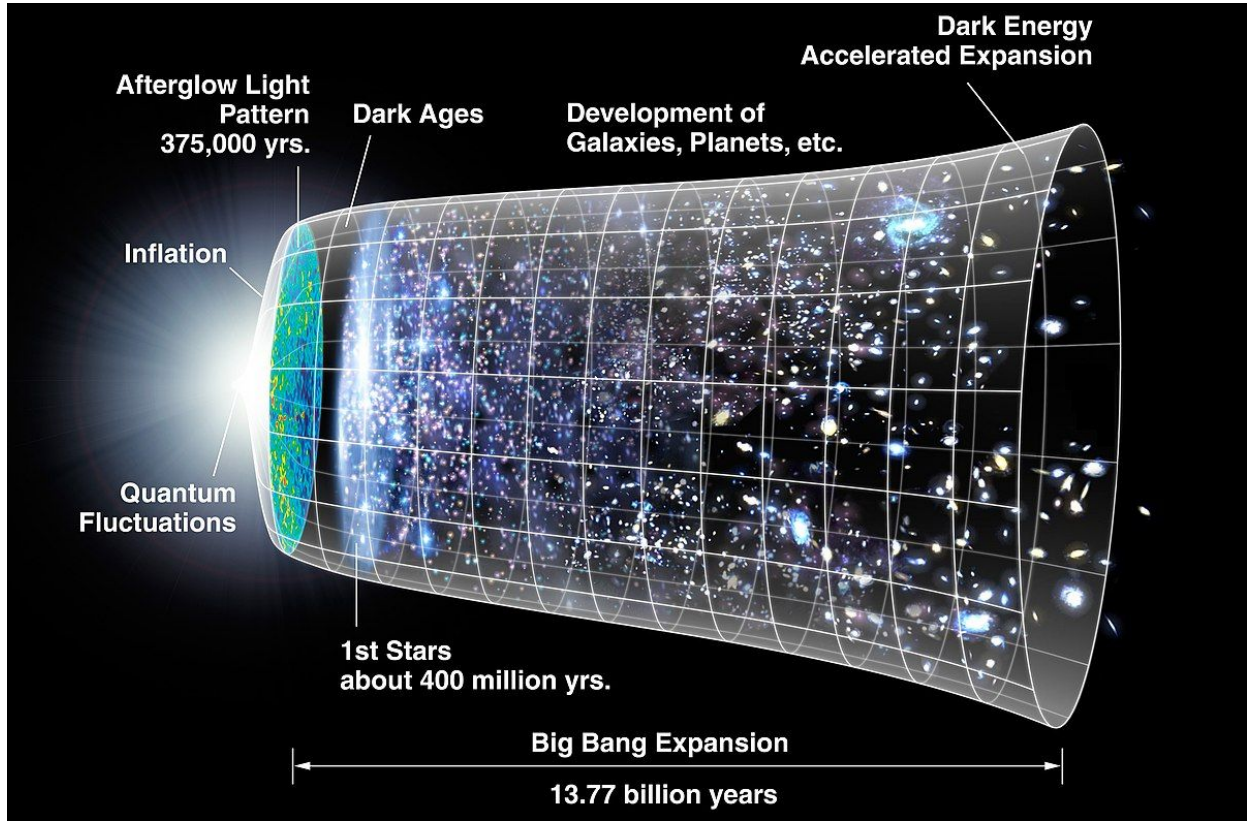
# Solución numérica de la ecuación de Boltzmann para materia oscura

UNIVERSIDAD DE ANTIOQUIA

Andrés Felipe Riaño Quintanilla
Santiago Julio Dávila

# Problema original
## Producción de materia oscura (DM) en el universo temprano

UNIVERSIDAD DE ANTIOQUIA




$$\mathcal{L}\{f\} = \mathcal{C}\{f\}$$

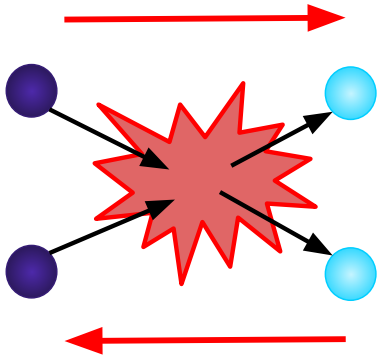
$$\frac{\mathrm{d}n}{\mathrm{d}t} + 3Hn = -\langle\sigma v\rangle(n^2 - n_{eq}^2)$$

$$\frac{\mathrm{d}Y}{\mathrm{d}x} = -\frac{\langle\sigma v\rangle s}{Hx}(Y^2 - Y_{eq}^2)$$

# Física de la producción de DM

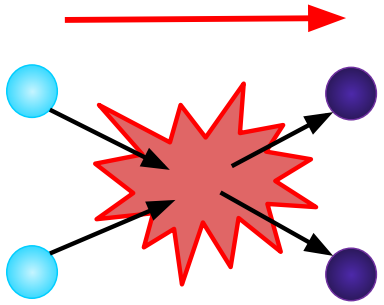
UNIVERSIDAD DE ANTIOQUIA

WIMPs



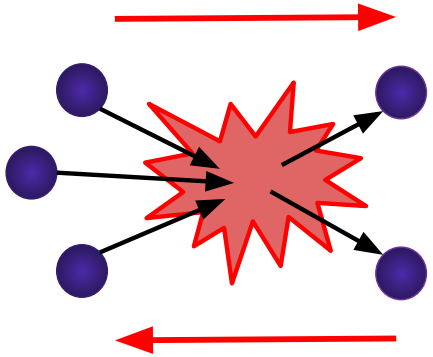
$$\frac{\mathrm{d}Y}{\mathrm{d}x} = -\frac{\langle\sigma v\rangle s}{Hx}(Y^2 - Y_{eq}^2)$$

FIMPs



$$\frac{\mathrm{d}Y}{\mathrm{d}x} = \frac{\langle\sigma v\rangle s}{Hx}Y_{eq}^2$$

SIMPs



$$\frac{\mathrm{d}Y}{\mathrm{d}x} = -\frac{\langle\sigma v^2\rangle s^2}{Hx}(Y^3 - Y^2 Y_{eq})$$

$$s = \frac{2\pi^2}{45}h_{eff}(T)T^3$$

$$\rho = \frac{\pi^2}{30}g_{eff}(T)T^4$$

$$H(T) = \sqrt{\frac{\rho}{3M_P^2}}$$

# Estructura del proyecto

UNIVERSIDAD DE ANTIOQUIA

```cpp
#include <iostream>
#include <vector>
#include <fstream>
#include <string>
#include <cmath>
#include <functional>
#include <iomanip>
```

```cpp
class Cosmology{
    private:
        std::vector<long double> geffvec,heffvec,Tvec,dlngeffdlnTvec,dlnheffdlnTvec;
    public:
        long double T;
        Cosmology(long double T);
        void readDegreesOfFreedom(const std::string& path);
        void calculate(const std::string& path);
        long double geff,heff;
        long double entropyDensity,energyDensity,hubbleRate;
};

class DarkMatterModel{
    public:
        DarkMatterModel(std::string name,int DMini,int DMfin,int SMfin);
        std::string name;
        int DMini;
        int DMfin;
        int SMfin;
        void printProcess();
};
```

# Estructura del proyecto

UNIVERSIDAD DE ANTIOQUIA

```
class BoltzmannEquation{
    public:
        BoltzmannEquation(long double x,long double MS,int g,long double sigmav,DarkMatterModel model,const std::string& path);
        long double x;
        long double MS;
        int g;
        long double sigmav;
        std::string modelname;
        const std::string path;
        Cosmology C;
        int DMini, DMfin, SMfin;
        long double Yeq();
        long double dYdx(long double Y);
        void setX(long double new_x);
};
```

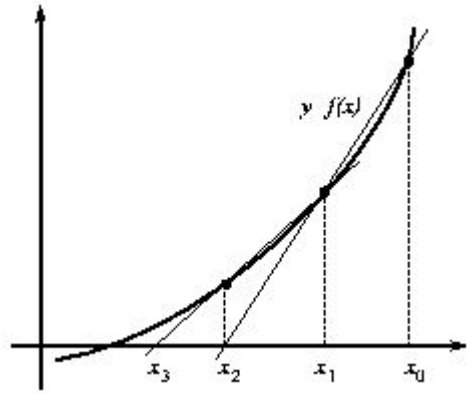
# Estructura del proyecto

UNIVERSIDAD DE ANTIOQUIA

```
class Secante {
public:
    // Constructor
    Secante(double tol, int maxIter);

    // Método para encontrar la raíz
    double encontrarRaiz(const std::function<double(double)>& f, double x0, double x1);

private:
    double tolerancia;
    int maxIteraciones;
};
```




$$x_2 = x_0 - \frac{x_1 - x_0}{f(x_1) - f(x_0)} f(x_0)$$

# Estructura del proyecto

UNIVERSIDAD DE ANTIOQUIA

```
class BEqSolver{
    public:
        BEqSolver(long double xi0,long double xif,int nxi,long double Y0,BoltzmannEquation BEQ);
        long double x0;
        int nxi;
        long double Y0;
        BoltzmannEquation BEQ;
        long double Omegah2,delta;
        std::vector<long double> xis,xs,Ys;
        void solve();
};

class FieldWriter {
    public:
        FieldWriter();
        std::ofstream solution, params;
        void write(BEqSolver sol);
};
```

Backward Differentiation Formula (BDF-2)

$$\frac{\mathrm{d}Y}{\mathrm{d}x} = F(x, Y)$$

$$\left(\frac{\mathrm{d}Y}{\mathrm{d}x}\right)_{n+1} \simeq \alpha_1 Y_{n+1} + \alpha_0 Y_n + \alpha_{-1} Y_{n-1}$$

$$\implies \alpha_1 Y_{n+1} + \alpha_0 Y_n + \alpha_{-1} Y_{n-1} = F(x_{n+1}, Y_{n+1})$$

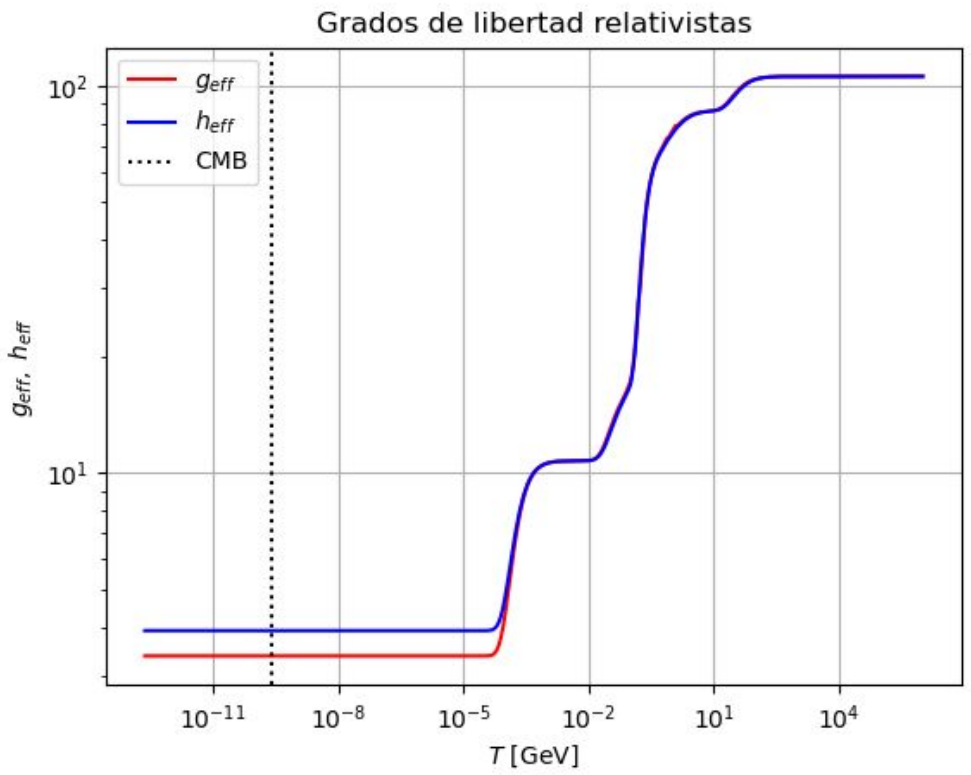
$$\alpha_1 = \frac{1}{\Delta x_n}\left(1 + \frac{\Delta x_n}{\Delta x_{n-1} + \Delta x_n}\right)$$

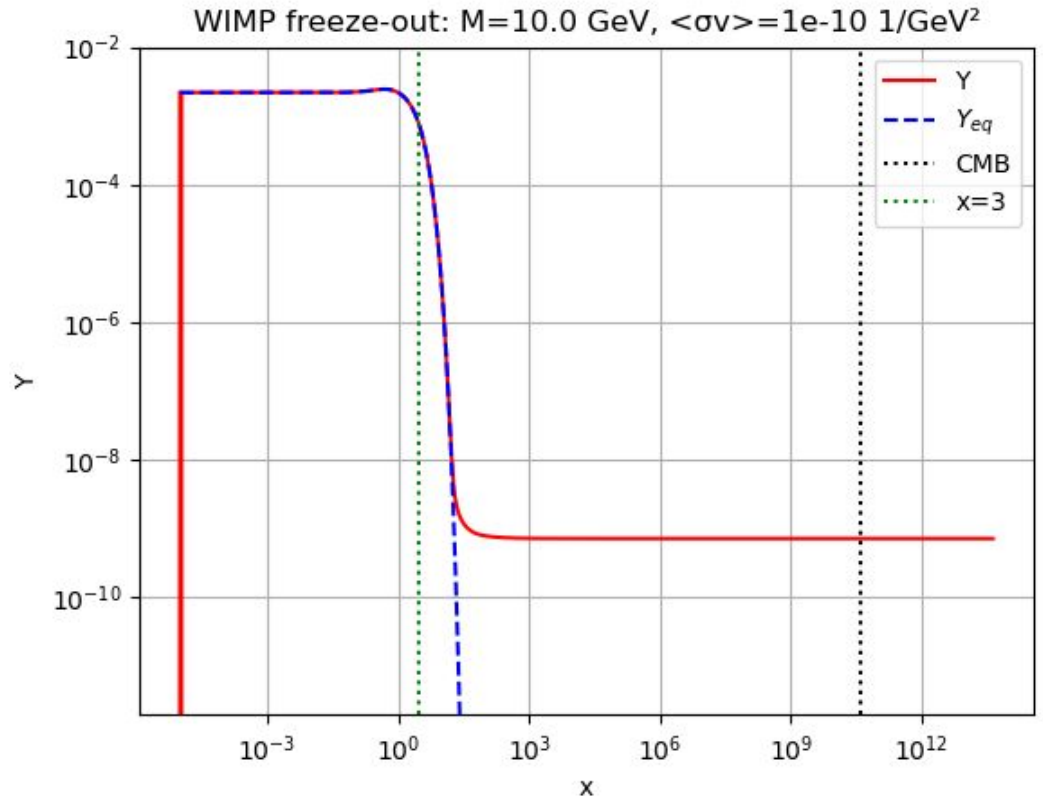
$$\alpha_{-1} = \frac{1}{\Delta x_{n-1}}\left(\frac{\Delta x_n}{\Delta x_{n-1} + \Delta x_n}\right)$$

$$\alpha_0 = -\alpha_1 - \alpha_{-1}$$

# Resultados

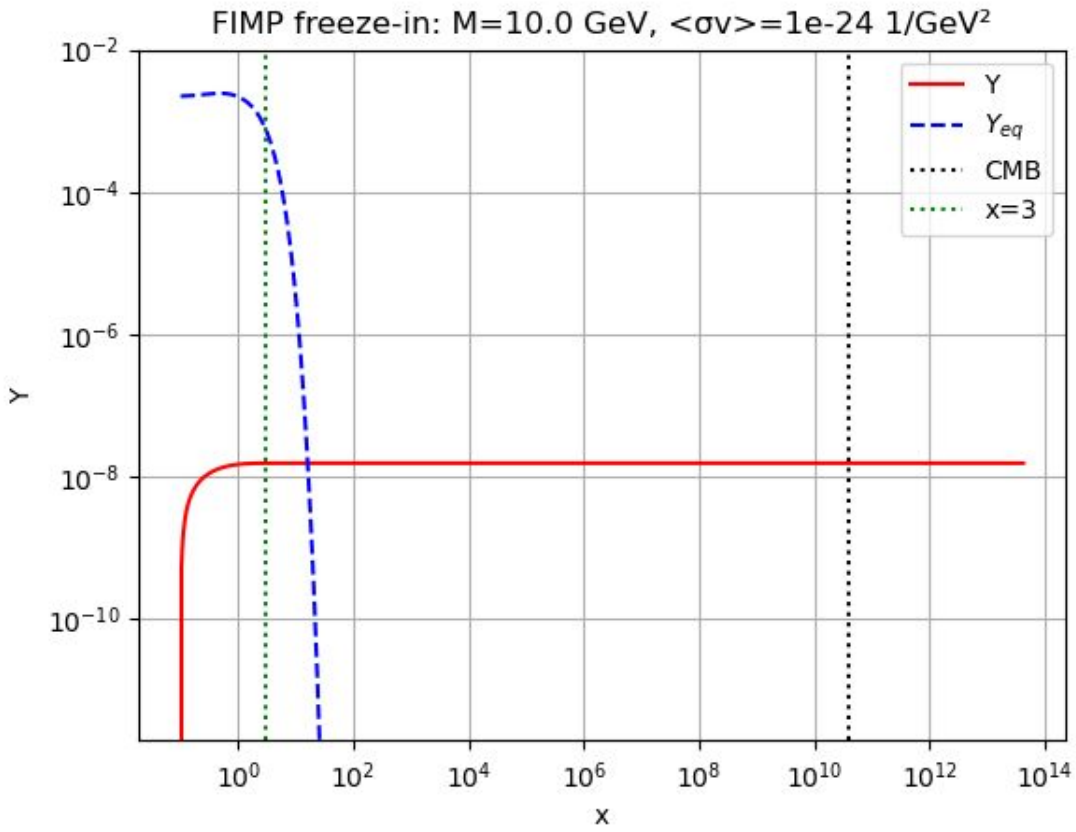
UNIVERSIDAD DE ANTIOQUIA

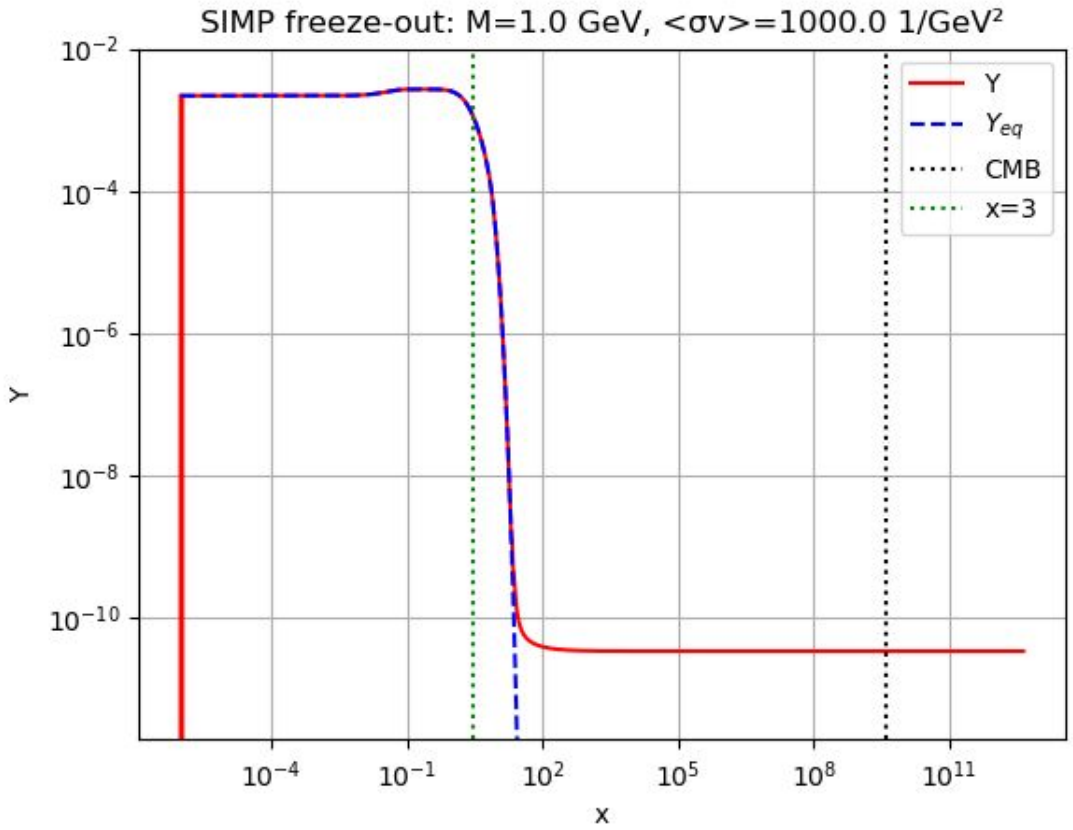


# Resultados

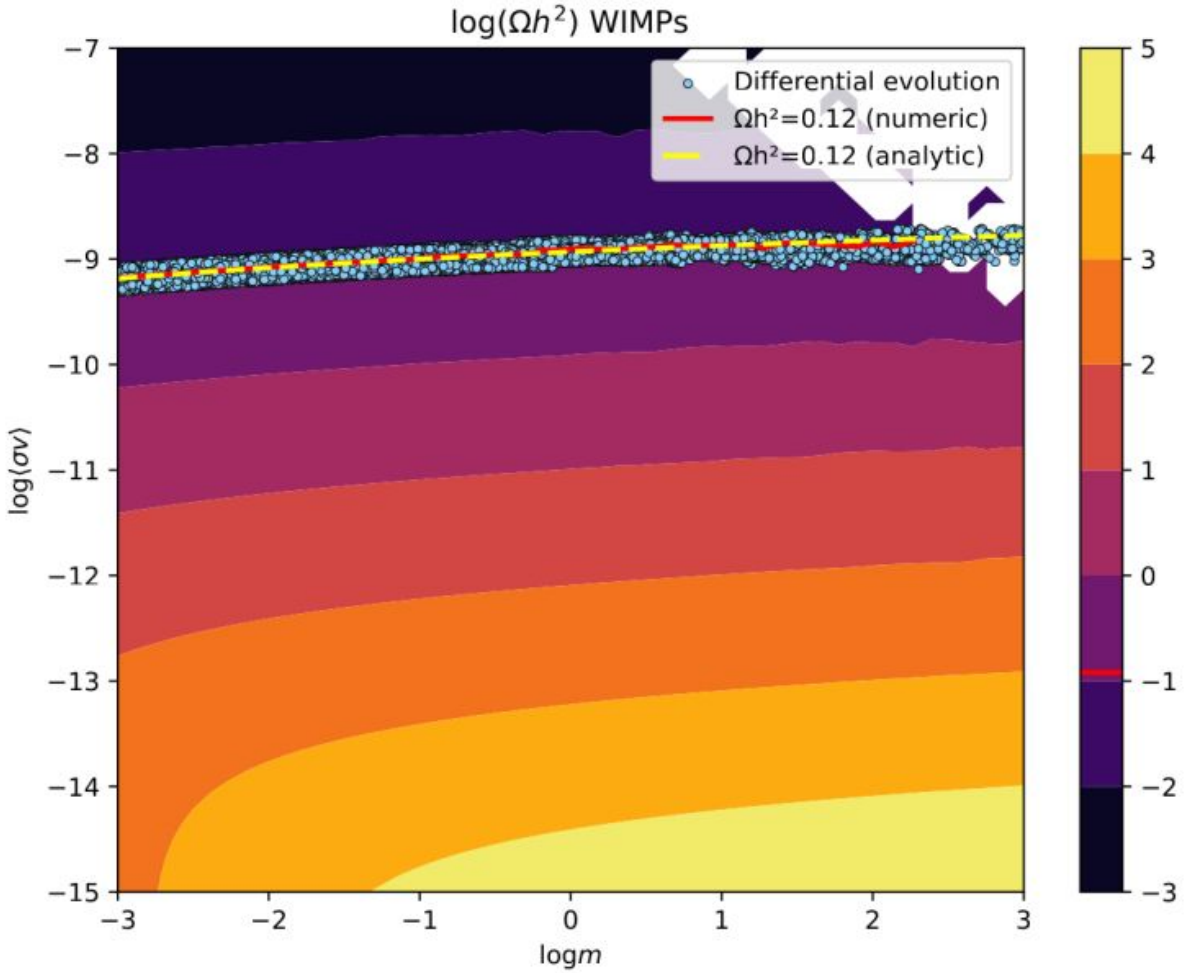
UNIVERSIDAD DE ANTIOQUIA



# Perspectivas

UNIVERSIDAD DE ANTIOQUIA



# UNIVERSIDAD DE ANTIOQUIA

@UdeA @universidaddeantioquia @UdeA

@universidaddeantioquia @UdeA