# Universidad de Costa Rica

# Ciencias de Computación e Informática

# CI-1201 Programación II

# Profesor Adolfo Di Mare

Datos de prueba para la [Matriz Chirrisquitica](http://www.di-mare.com/adolfo/p/Matrix.htm)

# Gabriel Castañaza B31520 ([gcmelmejor@gmail.com](mailto:gcmelmejor@gmail.com))

# David Ceciliano B31676 ([dacs200595@gmail.com](mailto:dacs200595@gmail.com) )

## Matrices

Dentro de las estructuras de datos más usadas, ya sea por su utilidad, versatilidad o para fines académicos, es la matriz. Un arreglo numérico, cuyas dimensiones son variables, su utilidad tanto matemática, para la ejecución de algoritmos, como contenedor, la hacen una de las más usadas en la solución de problemas. EN el caso de C++, esta no se ha implementado directamente en la Biblioteca Estándar (STL). Normalmente, se sugiere la solución de usar un vector de vectores, pero está solución no es la más eficiente, en especial para el desarrollo de algoritmos. De ahí, surge la necesidad de crear una matriz simple, eficiente, y con altas capacidades algorítmicas, en su variedad de métodos y operadores. La Matriz Chirrisquitica, implementada por Adolfo Di Mare, y utilizada en la tarea programada, pretende ser esta matriz de alta utilidad.

## Matriz Chirrisquitica ([documentación doxygen](http://soygcm.github.io/Programacion2/Tarea4/Matrix/html/classMx_1_1Matrix.html))

La matriz chirrisquitica incluye en el Rep tres datos simples, la cantidad de filas, la cantidad de columnas, y un vector de valores, asociados a la matriz. Incluye sobrecarga de operadores, con las operaciones aritméticas, siguiendo los procedimientos correspondientes, así como la sobrecarga del operator(unsigned,unsigned), el cual le permite acceder al usuario al valor referido, de manera inmediata, sin tener que interferir con el Rep. Además de varias operaciones más, como el cambio de tamaño y la transposición (.reSize(m,n) y .transpose()), existen funciones booleanas, que permiten saber características de la matriz, como si es cuadrada, o es diagonal.

Para efectos de la Tarea Programada número 4, se tomó esta implementación, y se diseñaron casos de pruebas para los principales métodos de la clase, aplicando el método de pruebas unitarias, con ayuda del módulo BUnit, diseñado también por Adolfo Di Mare.