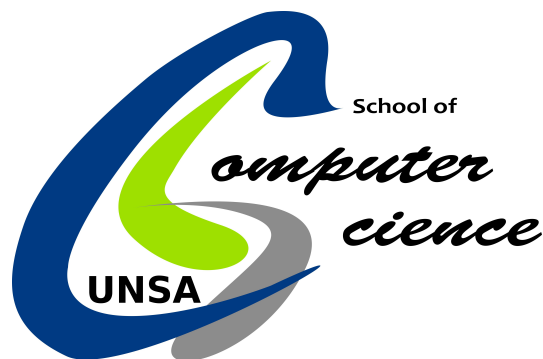


UNIVERSIDAD NACIONAL DE SAN AGUSTÍN  
ESCUELA PROFESIONAL DE CIENCIA DE LA COMPUTACIÓN



TEMA:

**QR ALGORITHM IN C++**

Curso:

**MATEMÁTICA APLICADA A LA COMPUTACIÓN**

**Presentado por:**

Christofer Chávez Carazas

---

Arequipa - Perú  
2017

# 1. Código

```
#include <iostream>
#include <tuple>
#include "OperacionesMatriz.h"

using namespace std;

Num MulList(Lista A, Lista B){
    Num res = 0;
    for(int i = 0; i < A.size(); i++){
        res += A[i] * B[i];
    }
    return res;
}

vector<Lista> getColumns(Matriz & A){
    vector<Lista> res;
    for(int i = 0; i < A.size(); i++){
        Lista temp;
        for(int j = 0; j < A.size(); j++){
            temp.push_back(A[j][i]);
        }
        res.push_back(temp);
    }
    return res;
}

Matriz generarMatriz(vector<Lista> E, int tam){
    Matriz Q = zeros(tam);
    for(int i = 0; i < tam; i++){
        for(int j = 0; j < tam; j++){
            Q[j][i] = E[i][j];
        }
    }
    return Q;
}

tuple<Matriz, Matriz> QRdescomposition(Matriz & A){
    vector<Lista> Es;
    Matriz Q = zeros(A.size());
    Matriz R = zeros(A.size());
    vector<Lista> As = getColumns(A);
    for(int i = 0; i < A.size(); i++){
        Lista u = As[i];
        for(int j = 0; j < i; j++){
            Num temp = MulList(As[i], Es[j]);
            R[j][i] = temp;
            Lista lTemp = Es[j] * temp;
            u = u - lTemp;
        }
        Es.push_back(u * (1.0/norm(u)));
        R[i][i] = MulList(As[i], Es[i]);
    }
    Q = generarMatriz(Es, A.size());
    return make_tuple(Q, R);
}

int main(int argc, char * argv[]) {
    if(argc != 2){
        cout<<"Faltan argumentos <Numero de iteraciones"<<endl;
        return 0;
    }
    string temp(argv[1]);
    int n = stoi(temp);
    Matriz A = {{3,8,1},{2,3,8},{0,2,1}};
    cout<<"Matriz Original:"<<endl;
    mostrarMatriz(A);
    for(int i = 0; i < n; i++){
        auto Q_R = QRdescomposition(A);
        Matriz Q = get<0>(Q_R);
        Matriz R = get<1>(Q_R);
        A = R * Q;
        cout<<"Iteracion " <<i+1<<":"<<endl;
        mostrarMatriz(A);
    }
}
```

# 2. Ejemplo

El ejemplo mostrado se realiza con la siguiente matriz:

$$A = \begin{bmatrix} 3 & 8 & 1 \\ 2 & 3 & 8 \\ 0 & 2 & 1 \end{bmatrix}$$

Número de iteraciones: 5

### 3. Resultados

Se muestra el resultado de cada iteración.

```
xnpio@xnpio-Satellite-U40t-A:~/Documentos/Xnpio/MAC/qr/c++$ ./run 5
Matriz Original:
3 8 1
2 3 8
0 2 1

Iteration 1:
7.61538 0.35209 7.20291
1.54613 -4.15004 -0.796352
0 3.64125 3.53465

Iteration 2:
7.51975 6.07328 3.84716
1.09657 -1.41366 6.5688
0 0.786773 0.893916

Iteration 3:
8.35751 -2.83923 6.01282
0.347392 -3.99213 -4.25532
0 0.911045 2.63462

Iteration 4:
8.23281 4.59424 4.90891
0.165148 -2.72528 5.7629
0 0.351276 1.49247

Iteration 5:
8.32384 -3.71735 5.52598
0.0569319 -3.48887 -5.0421
0 0.269989 2.16504
```