



**UNIVERSIDAD NACIONAL AUTÓNOMA DE  
MÉXICO**

**FACULTAD DE ESTUDIOS SUPERIORES ACATLÁN**

**LICENCIATURA EN MATEMÁTICAS APLICADAS Y  
COMPUTACIÓN**

**TRABAJO FINAL  
- ITERACIÓN QR -**

**PROYECTO FINAL DE LA ASIGNATURA:  
MÉTODOS NUMÉRICOS I**

**PRESENTA:**

Aldair Hernández Noguez

**PROFESOR(A):**

Nombre del profesor(a)

Santa Cruz Acatlán, Naucalpan, Estado de México  
Noviembre 2025

# Contents

<b>1</b>	<b>Introducción</b>	<b>2</b>
<b>2</b>	<b>REQUERIMIENTOS</b>	<b>3</b>
2.1	Requerimientos no funcionales . . . . .	3
2.2	3.3 Requerimientos del usuario . . . . .	3
<b>3</b>	<b>Propuesta de Solución</b>	<b>4</b>
3.1	Algoritmo en Pseudocódigo . . . . .	4
3.2	SOFTWARE . . . . .	5
<b>4</b>	<b>Ejemplo Práctico</b>	<b>6</b>
<b>5</b>	<b>BIBLIOGRAFIA</b>	<b>7</b>

# **1. Introducción**

Este proyecto presenta un sistema computacional que aplica el método numérico Iteración QR para analizar matrices de características faciales. Dicho método permite obtener vectores propios relevantes, los cuales representan patrones dominantes dentro de un conjunto de datos, fundamento utilizado en técnicas de reconocimiento facial. El objetivo del sistema es demostrar la aplicación práctica de un método numérico a un caso real mediante un software interactivo y fácil de usar.

## **2. REQUERIMIENTOS**

### **2.1 Requerimientos no funcionales**

Características que debe cumplir el software:

- 1 – Interfaz gráfica intuitiva, clara y accesible.
- 2 – El sistema debe ejecutar el algoritmo en menos de 2 segundos para matrices  $\leq 10 \times 10$ .
- 3 – Debe ser multiplataforma (Windows, Linux, Mac).
- 4 – El código debe ser legible y documentado.
- 5 – El sistema debe avisar adecuadamente errores de formato.
- 6 – Bajo consumo de recursos.

### **2.2 3.3 Requerimientos del usuario**

El usuario debe poder cargar matrices sin conocimientos de Python. El usuario debe usar el sistema para comprender cómo funciona la Iteración QR. El usuario necesita ver resultados claros y entendibles. El usuario requiere explicación del proceso numérico utilizado.

# **3. Propuesta de Solución**

La empresa “VisionTech Solutions” solicita una herramienta que permita analizar características faciales mediante métodos numéricos para comprender patrones dominantes dentro de imágenes. Para ello se requiere un sistema que procese matrices asociadas a imágenes y obtenga autovalores relevantes mediante iteración QR, con el fin de evaluar la estructura dominante en los datos y apoyar proyectos futuros de visión artificial. La solución propuesta es un software con interfaz gráfica, donde el usuario puede cargar o ingresar matrices derivadas de características faciales, procesarlas mediante Iteración QR y visualizar los autovalores resultantes. El lenguaje a utilizar sera Python debido a su facilidad de programacion y uso

## **3.1 Algoritmo en Pseudocódigo**

función QRIteración(A):

repetir hasta convergencia:

Q, R = descomposiciónQR(A)

A = R \* Q

retornar diagonal(A)

## 3.2 SOFTWARE

```
import numpy as np
import tkinter as tk
from tkinter import messagebox, filedialog
from tkinter import ttk

# --- Método QR ---
def qr_iteration(A, tol=1e-8, max_iter=1000):
    A = np.array(A, dtype=float)
    for _ in range(max_iter):
        Q, R = np.linalg.qr(A)
        A_new = R @ Q
        if np.linalg.norm(A_new - A) < tol:
            break
    A = A_new
    return np.diag(A), A

# --- Funciones GUI ---
def load_matrix():
    file_path = filedialog.askopenfilename(filetypes=[("Text Files", "*.txt"), ("CSV Fil
    if not file_path:
        return
    try:
        matriz = np.loadtxt(file_path, delimiter=",", dtype=float)
        entry.delete("1.0", tk.END)
        entry.insert(tk.END, "\n".join(", ".join(map(str, row)) for row in matriz))
    except:
        messagebox.showerror("Error", "No se pudo leer el archivo.")

def calculate():
    try:
        raw = entry.get("1.0", tk.END).strip().split("\n")
        matriz = [list(map(float, row.split(","))) for row in raw]
        eigenvalues, A_final = qr_iteration(matriz)

        result_box.config(state="normal")
        result_box.delete("1.0", tk.END)
        result_box.insert(tk.END, "Autovalores de características faciales:\n")
        result_box.insert(tk.END, str(eigenvalues) + "\n\n")
    except:
        messagebox.showerror("Error", "Ocurrió un error al calcular los autovalores.")
```

## 4. Ejemplo Práctico

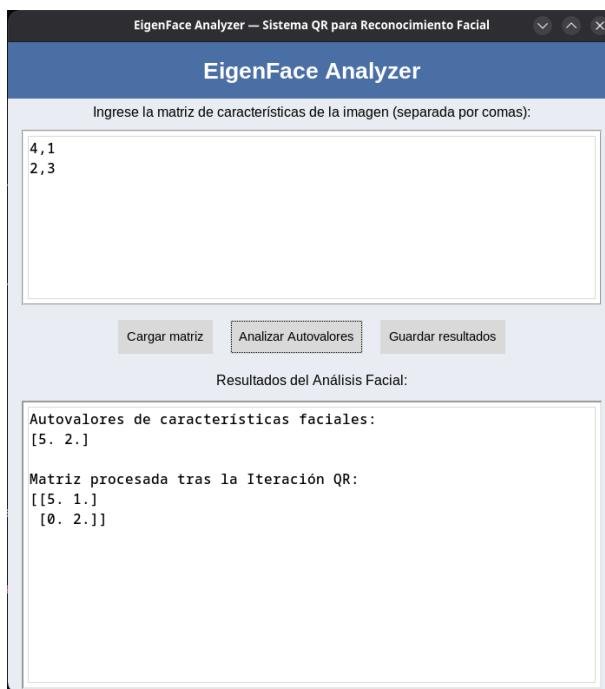


Figure 4.1: Ejemplo de uso de interfaz

```
GNU nano 8.3
Autovalores de características faciales:
[5. 2.]

Matriz procesada tras la Iteración QR:
[[5. 1.]
 [0. 2.]]
```

Figure 4.2: Ejemplo guardar resultados

## 5. BIBLIOGRAFIA

Burden, R. Faires, J. (2011). Análisis Numérico.

Trefethen, L. Bau, D. (1997). Numerical Linear Algebra.

Golub, G. Van Loan, C. (2013). Matrix Computations.

Python Software Foundation. NumPy Documentation. <https://numpy.org>