

# Prueba de Caja Blanca

*“Detección de Blancos Biológicos en Rosales”*

Versión 1.2

**Integrantes:**

Nicolas Cedillo  
Alisson Clavijo  
Lizzette Zapata

**Fecha 2024-02-04**

## Ingreso del Usuario (LOGIN)

### CÓDIGO FUENTE

```
@router.post("/api/historia/image/{historia_id}")
async def set_image(historia_id: str, file: UploadFile = File(...), user: User = Depends(current_user)):
    print(historia_id)
    try:
        # Asegurarse de que la historia exista
        collection: Collection = Database.get_connection().historias
        obj_id = ObjectId(historia_id)

        historia = collection.find_one({"_id": obj_id})

        if not historia:
            raise HTTPException(status_code=404, detail="Historia no encontrada")
            # raise HTTPException(status_code=HTTP_404_NOT_FOUND, detail="Historia no encontrada")

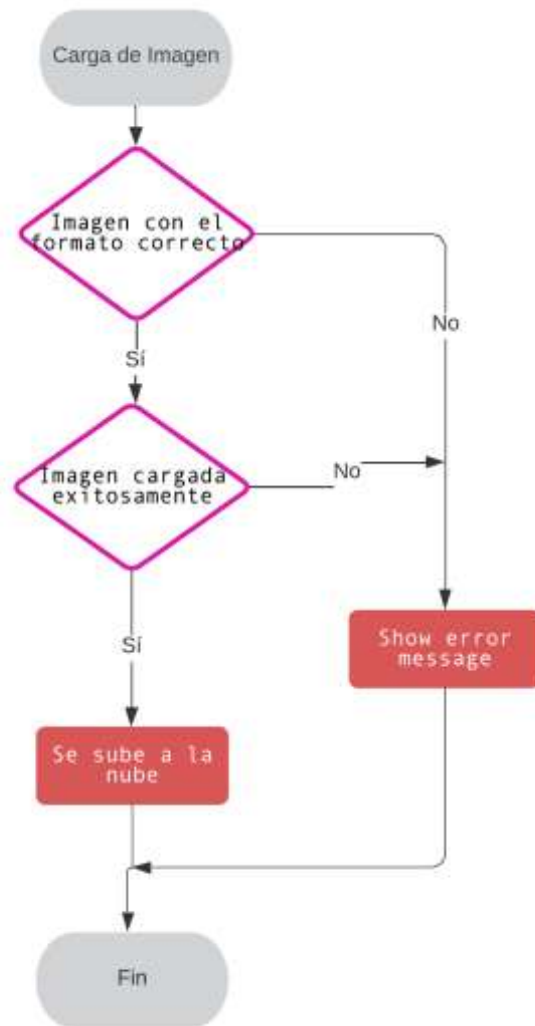
        print(historia)
        # Leer el contenido del archivo
        file_content = await file.read()
        # Subir el archivo a Cloudinary
        upload_result = cloudinary.uploader.upload(file_content)

        image_url = upload_result['secure_url']

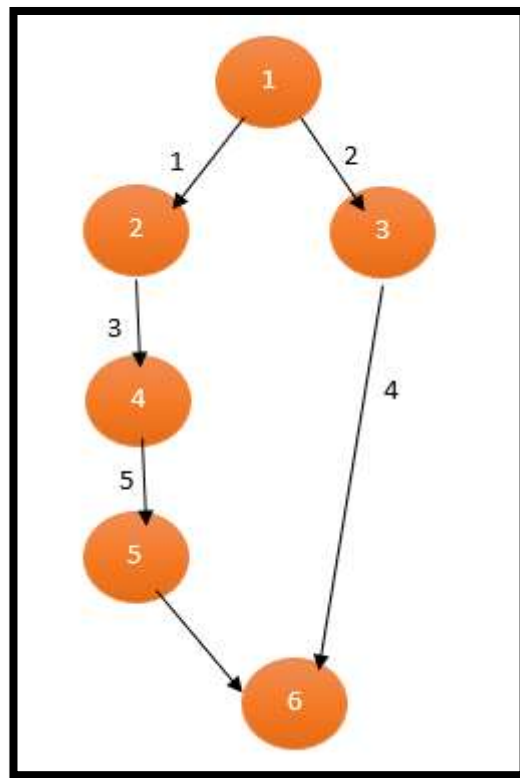
        # Actualizar la Historia con la URL de la imagen
        collection.update_one({"_id": obj_id}, {"$set": {"imageUrl": image_url}})

        # Actualizar la respuesta con la historia actualizada
        # historia["imageUrl"] = image_url
        # return historia
        return image_url
    except Exception as e:
        print("Error", str(e))
        raise HTTPException(status_code=500, detail=f"Error al subir la imagen a Cloudinary: {str(e)}")
```

## Diagrama de Flujo



## GRAFO



## RUTAS

**R1:** 1,3,5

**R2:** 2,4

## COMPLEJIDAD CICLOMÁTICA

Se puede calcular de las siguientes formas:

- $V(G) = A - N + 2 = 2$
- $V(G) = 5 - 6 + 2 = 2$

DONDE:

**A:** Número de aristas

**N:** Número de nodos