

**Instituto Tecnológico de Costa Rica**

**Ingeniería en Computación**

**IC6831-Aseguramiento de la Calidad**

**POC**

**Profesor: Saúl Calderón Ramírez**

**Estudiantes:**  
**María Fernanda Alvarado Vargas 2014084573**  
**Olman Castillo Picado 2015148651**  
**Yonattan Serrano Torres 2014005692**

**Primer Semestre 2018**

# Índice

|          |   |          |
|----------|---|----------|
| <b>1</b> | <b>Introducción</b>                         | <b>2</b> |
| <b>2</b> | <b>Indicaciones</b>                         | <b>2</b> |
| <b>3</b> | <b>Análisis del problema</b>                | <b>2</b> |
| <b>4</b> | <b>Trozos implementados</b>                 | <b>2</b> |
| 4.1      | Login . . . . .                             | 2        |
| 4.2      | Cargar imagen . . . . .                     | 3        |
| 4.3      | Formulario . . . . .                        | 3        |
| 4.4      | Cargar muestras . . . . .                   | 3        |
| <b>5</b> | <b>Diagrama de componentes</b>              | <b>4</b> |
| <b>6</b> | <b>Implementación y pruebas</b>             | <b>4</b> |
| 6.1      | Login . . . . .                             | 4        |
| 6.2      | Cargar imagen . . . . .                     | 4        |
| 6.3      | Agregar pacientes . . . . .                 | 4        |
| 6.4      | Muestras . . . . .                          | 4        |
| <b>7</b> | <b>Conclusiones y recomenda-<br/>ciones</b> | <b>4</b> |
| <b>8</b> | <b>Referencias</b>                          | <b>5</b> |

## 1 Introducción

Este documento se realiza con el fin de tener una idea de cómo realizar la documentación de un POC para implementarlo posteriormente en el proyecto semestral del curso de aseguramiento de calidad de software. El contexto de dicho proyecto se desarrolla en torno a una necesidad en el hospital nacional de niños para la automatización del cálculo de edad a partir de imágenes radiológicas.

Dicho lo anterior, el presente POC tiene como finalidad utilizar y aprender sobre las herramientas que nos indicó el profesor Saúl Calderón Ramírez, pues estas se necesitarán para llevar a cabo de manera exitosa el proyecto al que se denomina AES para esta tarea, en la cuál se implementan cuatro trozos de aplicación para probar que dichas herramientas nos funcionan en los ambientes y equipos en los que se trabajará el proyecto.

## 2 Indicaciones

- Metodología SCRUM.
- Uso de una herramienta de control de versiones, en este caso se utilizará github.
- Un diagrama de componentes.
- La implementación de cuatro trozos del sistema: Login, cargar imagen, cargar muestras, llenar formulario
- Documentación interna HTML.

## 3 Análisis del problema

Comenzar un nuevo proyecto conociendo las herramientas que se van a integrar es difícil, implementar uno sin tener previo conocimiento de las herramientas lo es aún más y puede poner en riesgo el proyecto. Es por eso que incluir un POC da una visión de lo que se puede lograr al integrar distintas herramientas en un mismo proyecto, da la idea al cliente e incluso a los mismos desarrolladores que determinada solución es posible de implementar al realizar una prueba con los elementos que se quieren incluir, dando más seguridad de que al menos el software a utilizar es compatible y que las personas del equipo lo pueden manejar.

## 4 Trozos implementados

Para implementar cada una de las funcionalidades que se describirán a continuación, fue necesario realizar una investigación sobre cada una de las herramientas disponibles para el desarrollo de este software.

### 4.1 Login

En la página principal se muestra la información general de la aplicación, sin embargo es necesario que el usuario: radiólogo, pediatra; inicie sesión para

que se le desplieguen las demás funcionalidades como cargar imagen, cargar muestras y crear un nuevo paciente. Para realizar login, el usuario ya debe estar registrado en el sistema; es decir, en la base de datos la cuál se desarrollo en el motor de base de datos SQLite. En general está funcionalidad no tiene mucho costo de implementación, tampoco se necesita aprender de las diferentes tecnologías utilizadas como lo es Keras, para poder implementarla.

## 4.2 Cargar imagen

El sistema requiere cargar una imagen radiográfica para analizarla y realizar la estimación; para poder utilizar esta funcionalidad es importante aclarar que el usuario debe hacer login. Para resolver este problema, se utilizó la función `cv2` de `openCV` la cuál permite leer, mostrar y guardar la imagen.

- **Leer imagen:** Para leer una imagen, se utiliza la función `cv2.imread ()`; para utilizar esta funcionalidad, se debe especificar el directorio donde se ubica la imagen; además se debe especificar la forma en que se va a leer la imagen: a color, blanco y negro o cualquier formato. En caso de que la ruta no sea la correcta, no se muestra ningún error pero a la hora de imprimir la imagen, va a mostrar `NONE`; es decir, no se encuentra la imagen.
- **Mostrar imagen:** Para mostrar la imagen, se utiliza `Django` ya que es

el framework que controla el front end de la aplicación. Esta se va a mostrar una vez que se haya leído con la función anteriormente descrita.

- **Guardar imagen:** Está funcionalidad se utilizará una vez realizada la estimación de la edad del paciente; ya que la imagen analizada se debe guardar en una base de datos; para ello se piensa utilizar la función `cv2.imwrite ()` de `OpenCV`. Esta función guarda la imagen en un formato `.PNG` y lo hace en el directorio especificado.

Es importante recalcar que para utilizar las funciones de `OpenCV` se debe importar la librería **numpy**.

## 4.3 Formulario

El software a desarrollar solicita que se almacenen algunos datos importantes acerca del paciente; para realizar esta acción, se maneja un formulario donde el usuario (pediatra, radiólogo) inserta la información del paciente; una vez que se confirma la imagen, se almacena en la base de datos del sistema, la cuál se realizó en `SQLite`.

## 4.4 Cargar muestras

En esta sección se carga un archivo `.csv` que contiene las muestras dadas por el profesor; estas muestras se cargan con la función de `python` para abrir archivos y se almacenan en la base de datos.

## 5 Diagrama de componentes

Para resolver el problema implementado es necesario utilizar eclipse como entorno de desarrollo, además se utilizan herramientas como Django para el front end de la aplicación a desarrollar, keras es otra de las herramientas que se implementará, la cuál es una API para redes neuronales de alto nivel.

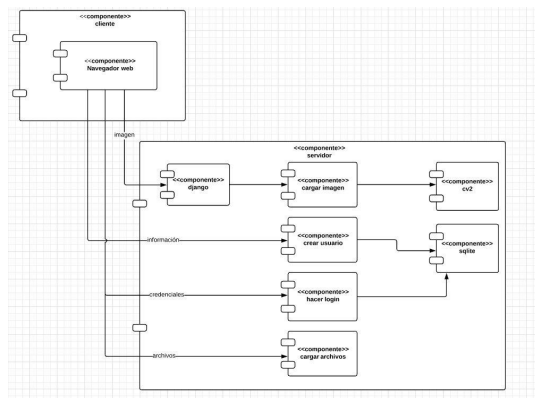


Figure 1: Diagrama de componentes

## 6 Implementación y pruebas

Para realizar las pruebas se va a utilizar la herramienta para pruebas unitarias DjangoTest; no se utiliza pythonTest debido a que da un error a la hora de realizar la prueba con la base de datos. A continuación, se muestra de forma detallada el código de las pruebas unitarias.

### 6.1 Login

```
class DjangoTest(TestCase):

    def test_1(self):
        print("Test1: Login")
        usuario = DaoBDUsuario()
        usuario.ingresar("olman", "12345")
```

Figure 2: Login

### 6.2 Cargar imagen

```
def test_3(self):
    print("Test3: Cargar imagen")
    imagen = Control()
    imagen.cargar_imagen("/website/main/media/upload")
```

Figure 3: Cargar imagen

### 6.3 Agregar pacientes

```
def test_2(self):
    print("Test2: Agregar paciente")
    dto_paciente = DTOPaciente(10,10," ", "olman", "castillo", "picado")
    paciente = DaoBDPaciente()
    paciente.guardar_informacion_paciente(dto_paciente)
```

Figure 4: Agregar pacientes

### 6.4 Muestras

```
def test_4(self):
    print("Test4: Cargar Muestras")
    muestra = DaoDBMuestra()
```

Figure 5: Muestras

## 7 Conclusiones y recomendaciones

Se expresan los aspectos más relevantes de las secciones implementadas

para esta tarea, como por ejemplo las tecnologías utilizadas para llevarla a cabo y la descripción general del funcionamiento de estas secciones, para poder brindar al lector el contexto de cada una de ellas.

Se brinda la opción al usuario para que haga login, esta implementación nos permitió experimentar con las herramientas que brinda Django para este tipo de funcionalidades específicamente en las páginas web. Así para una futura implementación, se tendrá en cuenta que esta herramienta nos facilitaría muchas tareas de desarrollo, minimizando el tiempo y esfuerzo.

Mediante la librería de CV2 de OpenCv, se permitió acceder a una imagen por píxeles y modificarlos, modificar propiedades de la imagen, configurar la región de interés, separar y combinar imágenes.

Para el almacenamiento de datos de la creación del paciente (como en el caso del login) se utilizó el motor de base de datos SQLite, el cual al estar integrado con el ambiente de desarrollo que se utilizó para esta tarea (Eclipse) facilita muchas labores de configuración.

Cabe destacar la importancia de recomendar al grupo, el trabajo organizado de las tecnologías controlando las versiones de cada una de estas, pues en algunos casos, la diferencia de estas pueden hacer que haya incompatibilidad a la hora de integrar el sistema con distintos componentes desarrollados con versiones o en máquinas diferentes.

## 8 Referencias

### References

- [1] OpenCv (2017) *Basic Operations on Images*. Available at: [https://docs.opencv.org/3.3.0/d3/df2/tutorial\\_py\\_basic\\_ops.html](https://docs.opencv.org/3.3.0/d3/df2/tutorial_py_basic_ops.html)

<https://github.com/olmancp/>