

Instituto Tecnológico de Costa Rica  
Ingeniería en Computación  
Aseguramiento de la Calidad del Software

*POC - SEGMENTADOR TEMPORAL*

Jorge Carranza

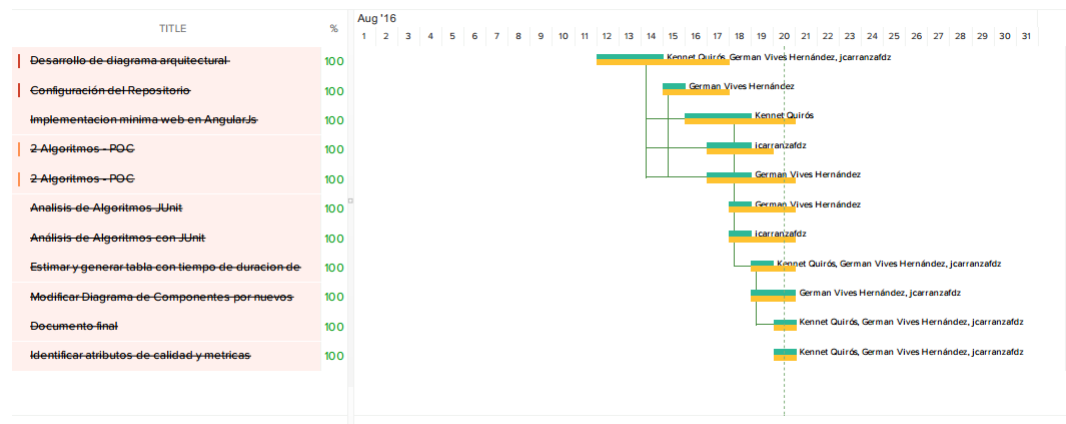
Kenneth Quirós

German Vives

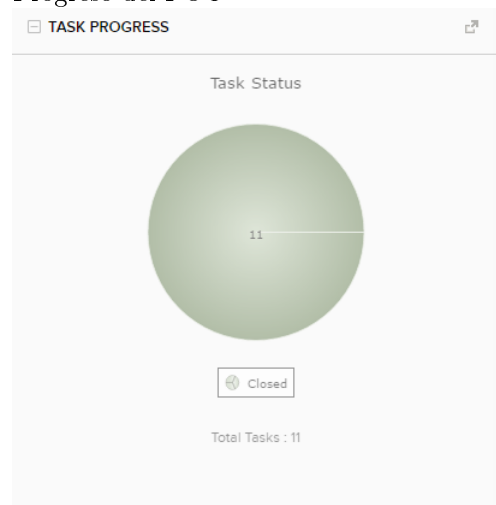
# Parte I

## Metodología SCRUM

Diagrama Gantt

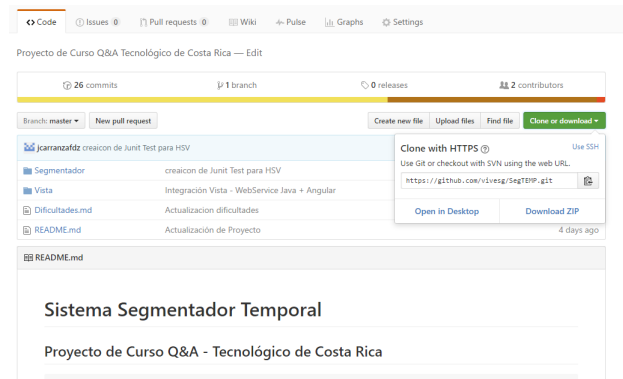


Progreso del POC



## Parte II

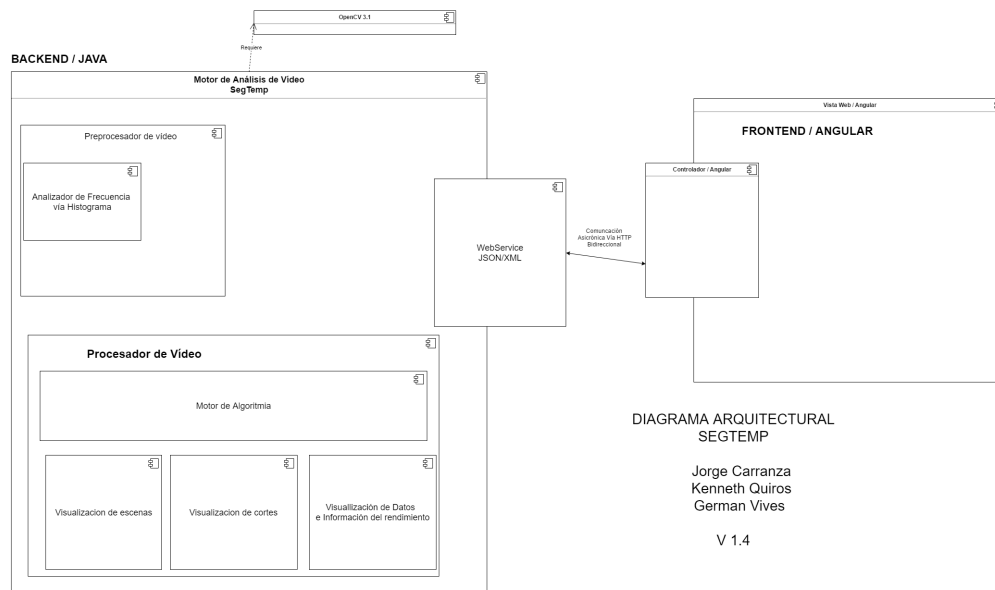
# Repositorio Oficial



La dirección del repositorio es <https://github.com/vivesg/SegTEMP.git>  
solicitar permisos para acceder a [gvives@outlook.com](mailto:gvives@outlook.com)

## Parte III

# Diagrama de Componente



- Backend: En el backend tenemos el servicio encargado de procesar el video

y ejecutar el algoritmo desegmentación temporal

- Preprocesador de Video: En este paquete están las clases relacionadas con el manejo de la entrada es decir el vídeo , este se procesa generando los frames posteriormente se generan los histogramas los cuales son la salida de este componente.
- Procesador de video: Este paquete tiene varios componentes importantes los cuales encierran la lógica más avanzada del sistema así
  - Visualización de escenas: Tenemos la lógica relacionada con escenas y algoritmos relacionados
  - Visualización de cortes: tenemos las clases que modelan el comportamiento de los cortes
  - Visualización de datos e información : Contiene el modelado de las clases con relación a estadísticas e información del video , rendimiento entre otros
  - Webservice : Componente que se encarga de generar el envío de datos y establecer un servidor web para recibir solicitudes de procesamiento de un video , envía un XML / JSON para ser consumido por la Vista Web

#### ■ Frontend

- Controlador AngularJS: Encargado de manejar la vista y consumir el webservice

## Parte IV

# Análisis básico de algoritmos:

1. Algoritmo conversor de BGR a HSV: Este algoritmo se encarga de recibir la dirección de la imagen, obtener sus datos en formato BGR y convertirlos a HSV

```
public Mat convertirBGR2HSV(String direccion){  
    File input = new File(direccion);
```

```

        BufferedImage image = new BufferedImage(0,0,0) ;
        try {
            image = ImageIO.read(input);
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Error abriendo la imagen");
            Mat nulo = new Mat();
            return nulo;
        }
        byte[] data = ((DataBufferByte) image.getRaster().getDataBuffer()).getData();
        Mat mat = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC3);
        mat.put(0, 0, data);
        Mat mat1 = new Mat(image.getHeight(), image.getWidth(), CvType.CV_8UC1);
        Imgproc.cvtColor(mat, mat1, Imgproc.COLOR_BGR2HSV);
        return mat1;
    }
}

```

2. Algoritmo generador de histograma: Este algoritmo se encarga de recibir la matriz de un frame normalizado y generar su histograma

```

public Mat calcular(Mat mat1){
    // añade matriz a la lista
    java.util.List<Mat> matList = new LinkedList<Mat>();
    matList.add(mat1);
    Mat histogram = new Mat();
    MatOfFloat ranges = new MatOfFloat(0,256);
    Imgproc.calcHist(
        matList,
        new MatOfInt(0),
        new Mat(),
        histogram,
        new MatOfInt(256),
        ranges, false);
}

```

3. Algoritmo generador de frames:

```

public void CargarVideo(String Ruta) {
    Mat frame = new Mat();
    VideoCapture camera = new VideoCapture(Ruta);
    camera.open(Ruta);
    System.out.println(Ruta);
    JFrame jframe = new JFrame("MyTitle");
    jframe.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JLabel vidpanel = new JLabel();
    jframe.setContentPane(vidpanel);
    jframe.setVisible(true);
}

```

```

        int contador = 0 ;
        while (true){
            if (camera.read(frame)) {
                ImageIcon image = new ImageIcon(Mat2BufferedImage(frame));
                vidpanel.setIcon(image);                                vidpanel.repaint();
                Image img = image.getImage();
                BufferedImage bi = new BufferedImage(img.getWidth(null),
                    img.getHeight(null),
                    BufferedImage.TYPE_3BYTE_BGR);
                Graphics2D g2 = bi.createGraphics();
                g2.drawImage(img, 0, 0, null);
                g2.dispose();
                try {
                    File theDir = new File("output");
                    theDir.mkdir();
                    ImageIO.write(bi, "jpg",
                        new File("output/img"+String.valueOf(contador) +
                            ".jpg"));
                } catch (IOException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }
                contador++;
            }
        }
    }
}

```

#### 4 Algoritmo Matriz a Imagen

```

public BufferedImage Mat2BufferedImage(Mat m){
    int type = BufferedImage.TYPE_BYTE_GRAY;
    if ( m.channels() > 1 )
        type = BufferedImage.TYPE_3BYTE_BGR;
    int bufferSize = m.channels()*m.cols()*m.rows();
    byte [] b = new byte[bufferSize];
    m.get(0,0,b);
    BufferedImage image = new BufferedImage(m.cols(),m.rows(), type);
    final byte[] targetPixels =
        ((DataBufferByte) image.getRaster().getDataBuffer()).getBytes();
    System.arraycopy(b, 0, targetPixels, 0, b.length);
    return image;
}

```

## Parte V

# Dificultades Enfrentadas

1. Dificultad 01 Bajo Plataforma Windows las direcciones de recursos ejemplo recurso/imagen.png en Java se resuelven de manera /c:/folder/recurso/imagen.png lo que cae en una condición de error Corrección Se debe eliminar el primer carácter de la hilera string siempre y cuando se trabaje en la plataforma Windows \*Estado \* Solucionada
2. Dificultad 02 Durante la instalación del ambiente se presento problemas con el Path de las librerías DLL y el JAR asociado a OpenCV 3.1 Por Motivos de documentación introductoria Corrección se procede a usar versión OpenCV 2.4.13 Estado Solucionada
3. Dificultad 03 Se presentaron problemas de conexión con la configuración del ambiente de desarrollo Eclipse junto con la librería de OpenCV 2.4.13 para el ejecucion de codigo Corrección Se añaden las librerías de forma nativa en el código con la línea "System.loadLibrary" Estado Solucionada
4. Dificultad 04 Durante el desarrollo de la prueba de concepto se presentaron problemas de compatibilidad de los SO utilizados por el equipo de desarrollo ya que un miembro poseía arquitectura x86 mientras los otros x64 Corrección Se añaden las librerías de ambos sistemas así como el código respectivo para su correcto funcionamiento y carga en ambas arquitecturas" Estado Solucionada
5. Dificultad 05 Debido a la poca experiencia del equipo de desarrollo con la metodología AGILE SCRUM se tuvo una curva de aprendizaje lenta con la utilización de la herramienta Zoho Projects Corrección El equipo de desarrollo estudia y se cerciora de que todos los miembros comprendan su funcionamiento, los sprint se manejan como Hitos junto con listas de tareas en la herramienta Zoho Projects Estado Solucionada

## Parte VI

# Estimaciones

Tabla con tiempos estimados por modulo o componente

| Tarea   | Estimación |
|---|------------|
| Preprocesador de vídeo                              | 2 Semanas  |
| Motor de algoritmia                                 | 2 Semanas  |
| Visualización de escenas                            | 1 Semana   |
| Visualización de cortes                             | 1 Semana   |
| Visualización de datos e información de rendimiento | 2 Semanas  |
| Vista GUI   | 2 Semanas  |