Documento de Arquitectura de Software (SAD)

SRICA\_030\_000

Proyecto:

“Uso de Sistema de Reconocimiento de Iris basado en Deep Learning para la identificación humana en el control de acceso al área de Tesorería del Gobierno Regional de Tacna – Tacna 2020”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **CONTROL DE VERSIONES** | | | | |
| **Versión** | **Identificador** | **Hecho por** | **Fecha** | **Motivo** |
| 1.0 | SRICA\_030\_000 | Luis Eduardo Mamani Bedregal |  | Se crea el documento de arquitectura de software (SAD). |

|  |  |
| --- | --- |
| **NOMBRE DEL PROYECTO** | **SIGLAS DEL PROYECTO** |
| Uso de Sistema de Reconocimiento de Iris basado en Deep Learning para la identificación humana en el control de acceso al área de Tesorería del Gobierno Regional de Tacna – Tacna 2020. | SRICA |

**Índice**

[**1.** **Introducción** 4](#_Toc93229840)

[**1.1.** **Propósito** 4](#_Toc93229841)

[**1.2.** **Alcance** 4](#_Toc93229842)

[**1.3.** **Referencias** 4](#_Toc93229843)

[**2.** **Requisitos Atendidos** 5](#_Toc93229844)

[**3.** **Estándar de Desarrollo** 5](#_Toc93229845)

[**4.** **Detalles de Implementación** 5](#_Toc93229846)

[**4.1.** **Plataforma y Lenguajes** 5](#_Toc93229847)

[**4.2.** **Patrón de Diseño** 6](#_Toc93229848)

[**5.** **Arquitectura del Sistema** 7](#_Toc93229849)

[**5.1.** **Diagrama de Paquetes** 7](#_Toc93229850)

[**5.2.** **Diagrama de Clases** 7](#_Toc93229851)

[**5.3.** **Diagrama de Secuencia** 7](#_Toc93229852)

[**5.4.** **Diagrama de Componentes** 7](#_Toc93229853)

[**5.5.** **Diagrama de Despliegue** 7](#_Toc93229854)

[**6.** **Modelo de Datos** 7](#_Toc93229855)

[**6.1.** **Diagrama Entidad – Relación** 8](#_Toc93229856)

[**6.2.** **Modelo Lógico** 8](#_Toc93229857)

[**6.3.** **Modelo Físico** 8](#_Toc93229858)

[**6.4.** **Diccionario de Datos** 8](#_Toc93229859)

# **Introducción**

El presente documento de arquitectura de software (SAD) del proyecto “Uso de Sistema de Reconocimiento de Iris basado en Deep Learning para la identificación humana en el control de acceso al área de Tesorería del Gobierno Regional de Tacna – Tacna 2020”, tiene por finalidad proveer una visión general de la arquitectura del sistema, usando diferentes vistas para apreciar los diferentes aspectos que posee, mediante el uso de Lenguaje Unificado de Modelado (UML) 2.0. Así mismo, se establecen los modelos de datos y estándares de desarrollo (codificación) que se usarán.

## **Propósito**

Los objetivos del presente documento son:

* Plasmar, mediante diagramas y modelos UML 2.0, la arquitectura del sistema.
* Esbozar los aspectos funcionales de la aplicación.
* Definir mecanismos de despliegue y distribución del sistema.
* Esbozar el modelo de datos de la arquitectura de datos a desarrollar.
* Definir el estándar de desarrollo (codificación) del sistema.

## **Alcance**

En el presente documento se detalla la arquitectura propuesta por el equipo de desarrollo. Además, se plasman modelos de dominio y datos, diagramas de diseño necesarios para comprender el comportamiento de los componentes y funcionalidad del sistema, y el estándar de desarrollo (codificación) a usar.

## **Referencias**

* SRICA\_029\_000 – Especificación de Requerimientos de Software (SRS)
* SRICA\_031\_000 – Estándar de Codificación
* SRICA\_024\_000 – Diagrama de Paquetes
* SRICA\_025\_000 – Diagrama de Clases
* SRICA\_026\_000 – Diagrama de Secuencia
* SRICA\_027\_000 – Diagrama de Componentes
* SRICA\_028\_000 – Diagrama de Despliegue
* SRICA\_032\_000 – Diagrama Entidad – Relación
* SRICA\_033\_000 – Modelo Lógico
* SRICA\_034\_000 – Modelo Físico
* SRICA\_035\_000 – Diccionario de Datos

# **Requisitos Atendidos**

Los requerimientos funcionales y no funcionales, y los casos de uso del sistema, son declarados en el documento: SRICA\_029\_000 – Especificación de Requerimientos de Software (SRS) (adjunto al presente documento).

# **Estándar de Desarrollo**

El documento que especifica el estándar de desarrollo para la codificación del sistema, se declara en: SRICA\_031\_000 – Estándar de Codificación (adjunto al presente documento).

# **Detalles de Implementación**

## **Plataforma y Lenguajes**

* JavaScript - React & Redux: Tecnología usada para el desarrollo del frontend del sistema (vistas web).
* C# .NET Core 3.1 – API RESTFUL: Tecnología usada para el desarrollo del backend del sistema (lógica y operaciones del sistema).
* Entity Framework Core con soporte MySQL para .NET Core 3.1: Tecnología ORM para el uso de datos relacionales con el servicio de base de datos MySQL.
* MySQL +8.0: Servicio de base de datos relacional.
* Raspberry Pi v4 Model B: Componente electrónico base para el equipo biométrico.
* Raspbian OS LTS: Sistema operativo utilizado por Raspberry Pi v4.
* CentOS 7 LTS: Sistema operativo del servidor de despliegue para el servicio web, servicio API, base de datos y servicios.
* Docker for Linux LTS: Sistema de contenedores para el despliegue del servicio web, servicio API y demás servicios.
* Python +3.6: Lenguaje utilizado para el desarrollo de los modelos de inteligencia artificial y deep learning.
* Node.js LTS: Tecnología usado por React para la creación y ejecución de módulos y paquetes node.
* .NET Core SDK 3.1 LTS: Plataforma para poder crear y ejecutar aplicaciones en .NET Core.
* Nginx LTS: Tecnología usado para el despliegue del frontend.
* YOLOv4 LTS: Tecnología usada para la detección de objetos.
* Detectron2 LTS: Tecnología usado para la segmentación de objetos.
* Tensorflow 1.14.0: Plataforma para machine learning.
* Keras 2.3.1: API Deep Learning que se ejecuta sobre Tensorflow.
* FastAPI LTs: Herramienta para desarrollar APIs con Python.
* Uvicorn LTS: Herramienta que despliega un servidor para APIs con Python.
* OpenCV compatible con Python +3.6: Conjunto de herramientas para Visión por Computadora en tiempo real.

## **Patrón de Diseño**

El patrón de diseño utilizado para el desarrollo del frontend del sistema es Compound Components & Provider Pattern.

El patrón de diseño utilizado para el desarrollo del backend del sistema y microservicios es DDD (Domain Driven Design).

# **Arquitectura del Sistema**

A continuación, se declaran los diagramas UML para representar arquitectónicamente al sistema:

## **Diagrama de Paquetes**

El diagrama de paquete del sistema está declarado en el documento: SRICA\_024\_000 – Diagrama de Paquetes (adjunto al presente documento).

## **Diagrama de Clases**

El diagrama de clases del sistema está declarado en el documento: SRICA\_025\_000 – Diagrama de Clases (adjunto al presente documento).

## **Diagrama de Secuencia**

El diagrama de secuencia del sistema está declarado en el documento: SRICA\_026\_000 – Diagrama de Secuencia (adjunto al presente documento).

## **Diagrama de Componentes**

El diagrama de componentes del sistema está declarado en el documento: SRICA\_027\_000 – Diagrama de Componentes (adjunto al presente documento).

## **Diagrama de Despliegue**

El diagrama de despliegue del sistema está declarado en el documento: SRICA\_028\_000 – Diagrama de Despliegue (adjunto al presente documento).

# **Modelo de Datos**

A continuación, se declara el modelo de datos para representar las entidades del sistema:

## **Diagrama Entidad – Relación**

El diagrama entidad – relación del sistema está declarado en el documento: SRICA\_032\_000 – Diagrama Entidad – Relación (adjunto al presente documento).

## **Modelo Lógico**

El modelo lógico del sistema está declarado en el documento: SRICA\_033\_000 – Modelo Lógico (adjunto al presente documento).

## **Modelo Físico**

El modelo físico del sistema está declarado en el documento: SRICA\_034\_000 – Modelo Físico (adjunto al presente documento).

## **Diccionario de Datos**

El diccionario de datos del sistema está declarado en el documento: SRICA\_035\_000 – Diccionario de Datos (adjunto al presente documento).