Manual Técnico

SRICA\_043\_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\_043\_000 | Luis Eduardo Mamani Bedregal |  | Se elabora el manual técnico del proyecto. |

|  |  |
| --- | --- |
| **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](#_Toc93261329)

[**2.** **Requerimientos Técnicos** 4](#_Toc93261330)

[**3.** **Herramientas Utilizadas para el Desarrollo** 4](#_Toc93261331)

[**4.** **Base de Datos** 6](#_Toc93261332)

[**4.1.** **Instalación** 6](#_Toc93261333)

[**4.2.** **Despliegue de Objetos SQL** 7](#_Toc93261334)

[**5.** **Docker** 8](#_Toc93261335)

[**5.1.** **Instalación** 8](#_Toc93261336)

[**6.** **Despliegue de Servicios** 9](#_Toc93261337)

[**6.1.** **Docker-Compose** 9](#_Toc93261338)

[**6.1.1.** **Configuración de Variables** 9](#_Toc93261339)

[**6.1.2.** **Configuración de Rutas** 9](#_Toc93261340)

[**6.1.3.** **Ejecución** 10](#_Toc93261341)

[**6.1.4.** **Servicio Activos** 11](#_Toc93261342)

[**7.** **Casos de Uso** 13](#_Toc93261343)

[**8.** **Modelo Entidad-Relación** 13](#_Toc93261344)

[**9.** **Diccionario de Datos** 13](#_Toc93261345)

[**10.** **Manual de Usuario** 13](#_Toc93261346)

# **Introducción**

El presente documento técnico 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 proporcionar aspectos técnicos para el correcto funcionamiento del sistema en los servidores donde éste se despliegue y/o use.

Para evitar la dependencia a una plataforma de despliegue, como es Microsoft Azure, se está utilizando un servidor físico de ejemplo (Linux con Centos7) para demostrar un procedimiento más genérico, que pueda ser probado por cualquier persona en sus equipos locales.

# **Requerimientos Técnicos**

* Requerimientos Mínimos de Hardware
* *RAM:* +2 GB
* *Procesador:* +2 CPUs cores a 3.2 GHz
* *Disco Duro:* +64 GB
* Requerimientos Mínimos de Software
* Privilegios de administrador
* *Sistema Operativo:* Cualquier sistema operativo que soporte Docker y MySQL

# **Herramientas Utilizadas para el Desarrollo**

* 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.

# **Base de Datos**

La base de datos utilizada para almacenar la información del sistema es MySQL. A continuación, se presentan los pasos de instalación y despliegue de objetos SQL para el motor MySQL +8.0 en el servidor Centos 7.

## **Instalación**

* Ejecutar el siguiente comando para habilitar MySQL en Centos 7: **rpm -Uvh https://repo.mysql.com/mysql80-community-release-el7-3.noarch.rpm**



* Deshabilitar los repositorios de MySQL para sólo instalar la versión 8: **sed -i 's/enabled=1/enabled=0/' /etc/yum.repos.d/mysql-community.repo**



* Ejecutar el siguiente comando para instalar MySQL: **yum --enablerepo=mysql80-community install mysql-community-server**

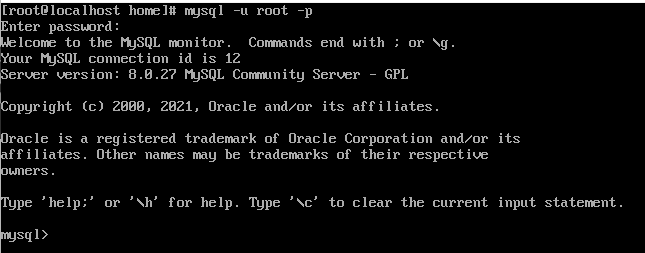


* Iniciar el servicio MySQL: **service mysqld start**

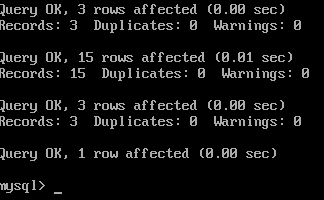


## **Despliegue de Objetos SQL**

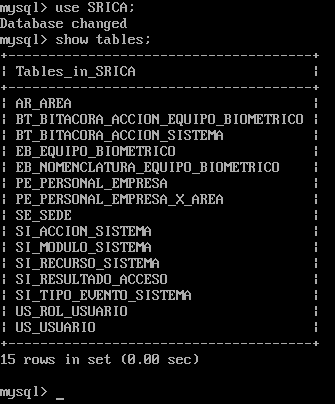
* Copiar el archivo “**1.2.12.3 SRICA\_034\_000 - Modelo Físico (SCRIPT).sql**” en el servidor Centos 7 (renombrarlo a un nombre más simple).
* Dentro del servidor, ejecutar el siguiente comando para ingresar al motor MySQL: **mysql -u “usuario” -p**



* Escribir el comando: **source /ruta\_archivo\_sql/script.sql**



* Verificar la base de datos creada mediante el siguiente comando:   
  **use SRICA;  
  show tables;**

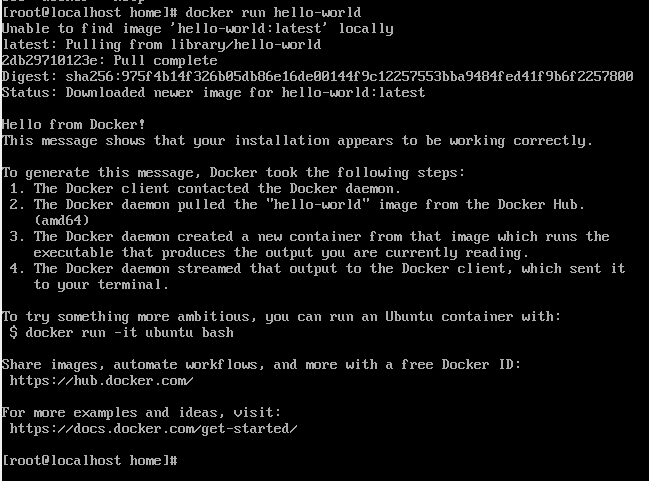


# **Docker**

Para el despliegue de los servicios en el servidor Centos 7, se utiliza Docker como herramienta de contenedores para tener un mayor control y facilitar el despliegue de aplicaciones. A continuación, se detallan los pasos para su instalación en el servidor.

## **Instalación**

* Los pasos de instalación lo pueden encontrar en: <https://docs.docker.com/engine/install/centos/>
* Verificar la funcionalidad de Docker mediante el siguiente comando: **docker run hello-world**



# **Despliegue de Servicios**

Para el despliegue de servicios, se usará la herramienta **docker-compose** para unificar servicios y realizar un despliegue más fácil.

## **Docker-Compose**

Docker-compose es una herramienta para definir y ejecutar varias aplicaciones Docker en contenedores. Esta herramienta utiliza un archivo de configuración de extensión YAML para la configuración de los servicios de la aplicación.

### **Configuración de Variables**

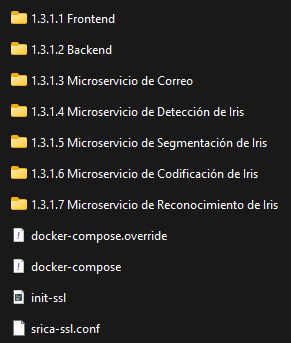
Para la configuración de base de datos y demás servicios externos que usa la aplicación, se debe modificar (según necesidad) el archivo **docker-compose.override.yml**. En ese archivo se podrá encontrar todas las configuraciones de los servicios del sistema.

### **Configuración de Rutas**

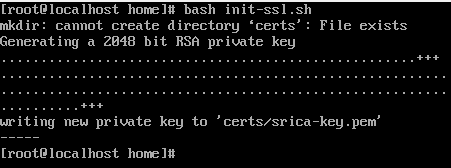
Para configurar la ruta del archivo *Dockerfile* de cada proyecto, se debe modificar (según necesidad) el archivo **docker-compose.yml**. En ese archivo se podrá encontrar todas las rutas y referencias a los proyectos y sus archivos *Dockerfile*.

### **Ejecución**

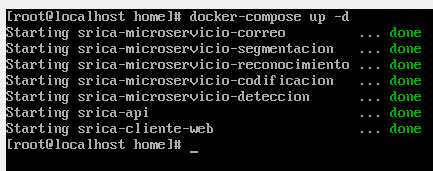
Una vez configurada las variables y/o rutas de proyectos, situarse en el directorio raíz, y ejecutar los siguientes comandos:



* Ejecutar el comando: **bash init-ssl**

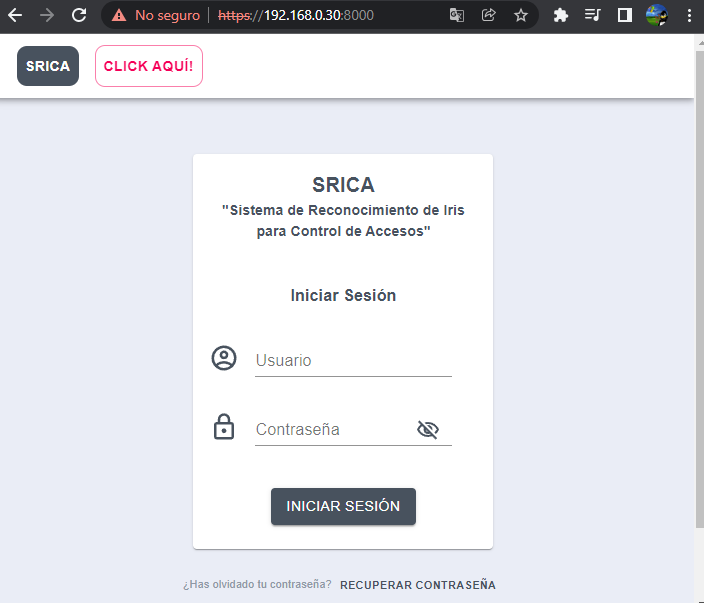


* Ejecutar el comando: **docker-compose up -d**

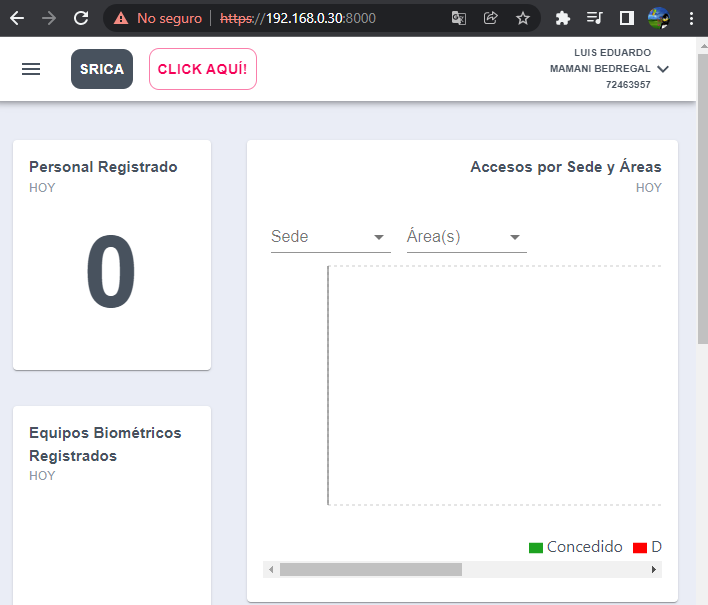


### **Servicio Activos**

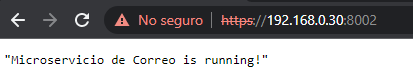
* **Servicio Web**



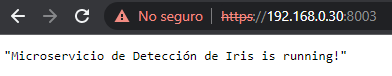
* **Servicio API**



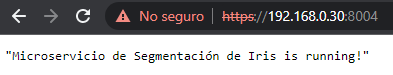
* **Microservicio de Correo**



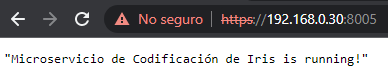
* **Microservicio de Detección de Iris**



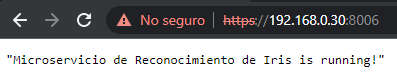
* **Microservicio de Segmentación de Iris**



* **Microservicio de Codificación de Iris**



* **Microservicio de Reconocimiento de Iris**



# **Casos de Uso**

El diagrama de clases del sistema está declarado en el documento: 1.2.3 SRICA\_023\_000 - Diagrama de Casos de Uso.

# **Modelo Entidad-Relación**

El diagrama de clases del sistema está declarado en el documento: 1.2.12.3 SRICA\_034\_000 - Modelo Físico.

# **Diccionario de Datos**

El diagrama de clases del sistema está declarado en el documento: 1.2.12.4 SRICA\_035\_000 - Diccionario de Datos.

# **Manual de Usuario**

El diagrama de clases del sistema está declarado en el documento: 1.4.2 SRICA\_044\_000 - Manual de Usuario.