Especificación de requerimientos del sistema – API

# Editor:

Reinaldo Espinel Torres

Plataforma de Orquestación para Automatizaciones RPA

**21/05/2024**

# Detalles del documento

|  |  |
| --- | --- |
| Seguridad | Publico |
| Version | Version 1.1 |
| Editor | Reinaldo Espinel Torres |

# Resumen

Este documento incluye los requerimientos del Sistema.

# Tabla de contenido

Contenido

[Editor: 1](#_Toc167471142)

[Detalles del documento 2](#_Toc167471143)

[Resumen 3](#_Toc167471144)

[Tabla de contenido 4](#_Toc167471145)

[1.1. Introducción y Contexto 5](#_Toc167471146)

[1.2. Introducción 5](#_Toc167471147)

[1.3. Proyecto 6](#_Toc167471148)

[2. Requerimientos del sistema 6](#_Toc167471149)

[2.1. Definición de términos 6](#_Toc167471150)

[2.1.1. Requerimiento del sistema 6](#_Toc167471151)

[2.1.1.1. Requerimientos funcionales y no funcionales 6](#_Toc167471152)

[2.1.1.2. clasificación de requerimientos en base a la funcionalidad 7](#_Toc167471157)

[2.2. Plantilla 8](#_Toc167471158)

[2.2.1. Plantilla para Requerimientos del sistema 8](#_Toc167471159)

[2.3. Lista de Requerimientos 8](#_Toc167471160)

[3. Características de usuario 12](#_Toc167471161)

[3.1. Usuarios del sistema 12](#_Toc167471162)

[3.1.1. Administrador del sistema 12](#_Toc167471163)

[3.2. Usuarios Finales 12](#_Toc167471164)

[3.2.1. Desarrollador RPA 12](#_Toc167471165)

[3.2.2. Usuario Operativo 12](#_Toc167471166)

[4. Abreviaturas 13](#_Toc167471167)

[5. References 14](#_Toc167471168)

# Introducción y Contexto

# Introducción

Este documento detalla los requerimientos funcionales y no funcionales para la API de gestión de automatizaciones. La API permite la gestión de ejecuciones de automatizaciones, simulación de claves seguras, y la autorización de ejecuciones en máquinas específicas.

# Requerimientos del sistema

**2.1. Gestión de Ejecuciones de Automatizaciones**

* **Objetivo**: Registrar el inicio y el fin de cada ejecución de automatización.
* **EndPoints**:
  + **POST /api/executions/start**: Registrar el inicio de una ejecución.
    - **Request Body**: JSON con detalles de la ejecución.
  + **POST /api/executions/end**: Registrar el fin de una ejecución.
    - **Request Body**: JSON con el ID de la ejecución y detalles de finalización.
* **Modelos**: Definidos en models.py en el directorio control\_automations.

**2.2. Simulación de Claves Seguras de Microsoft**

* **Objetivo**: Proveer una funcionalidad para simular la generación y validación de claves seguras.
* **EndPoints**:
  + **POST /api/keys/generate**: Generar una nueva clave segura.
    - **Request Body**: JSON con detalles necesarios para la generación de la clave.
  + **POST /api/keys/validate**: Validar una clave segura existente.
    - **Request Body**: JSON con la clave a validar.
* **Modelos**: No especificados, implementación probablemente en Functions.py.

**2.3. Autorización de Ejecuciones en Máquinas Específicas**

* **Objetivo**: Permitir o denegar la ejecución de automatizaciones en máquinas específicas.
* **EndPoints**:
  + **POST /api/authorization/grant**: Permitir la ejecución en una máquina específica.
    - **Request Body**: JSON con detalles de la máquina y la automatización.
  + **POST /api/authorization/deny**: Denegar la ejecución en una máquina específica.
    - **Request Body**: JSON con detalles de la máquina y la automatización.
* **Modelos**: No especificados, implementación probablemente en Functions.py.

**3. Requerimientos No Funcionales**

**3.1. Seguridad**

* **Autenticación y Autorización**: Uso de tokens JWT para la autenticación y autorización de usuarios.
* **Encriptación**: Encriptación de datos sensibles en tránsito y en reposo.

**3.2. Rendimiento**

* **Escalabilidad**: La API debe ser capaz de manejar múltiples solicitudes concurrentes sin degradación significativa del rendimiento.
* **Tiempo de Respuesta**: El tiempo de respuesta para las operaciones críticas debe ser menor a 500ms.

**3.3. Mantenimiento**

* **Documentación**: La API debe estar bien documentada, con ejemplos de uso para cada endpoint.
* **Pruebas**: Implementación de pruebas unitarias y de integración para asegurar la calidad del código.

**4. Dependencias**

* **Django**: Framework principal para el desarrollo del proyecto.
* **Django Rest Framework**: Para la creación de endpoints y manejo de serialización de datos.
* **PyJWT**: Para la generación y validación de tokens JWT.
* **Otras dependencias**: Listadas en requirements.txt (si está disponible).

# Referencias

[1] [Microsoft PowerPoint - 2-requirements.pptx (ugr.es)](https://elvex.ugr.es/idbis/db/docs/design/2-requirements.pdf)

[2] Python Software Foundation. Python.org.

[3] Selenium. Selenium.dev.