# Especificación de Requisitos de Software (SRS) Sistema de Web Scraping para X (antes Twitter) RascaCazuelas S.A.

### Administración de Proyectos I

**Profesora:** Alma Delia De Jesús Islao

**Ciclo escolar:** 2025-I

### Elaborado por:

Leonardo Valdes Palafox - 18-011-0249

Andrea Marlene Ortega Almendares - 17-003-1026

**Fecha de entrega:** 06 de mayo de 2025

**Versión:** 1.2 **Apegado a estándar:** IEEE 830 -1998

## Historial de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Fecha | Versión | Descripción | Autor |
| 01/03/2025 | 1.0 | Versión inicial del documento | Leonardo Valdes,  Andrea Ortega |
| 06/05/2025 | 1.2 | Inclusión de índice de requisitos, criterios de aceptación, sección de riesgos, estrategia V&V, apéndices numerados y requisitos adicionales  de usabilidad y escalabilidad | Leonardo Valdes, Andrea Ortega |
| C C | | | |

**Índice de Requisitos**

|  |  |
| --- | --- |
| ID | Nombre |
| RF-01 | Configuración de Parámetros de Extracción |
| RF-02 | Extracción de Tweets por Perfil |
| RF-03 | Extracción de Métricas de Tweets |
| RF-04 | Extracción de Metadatos de Tweets |
| RF-05 | Validación y Filtrado de Datos |
| RF-06 | Almacenamiento de Datos en CSV |
| RF-07 | Generación de Resumen Estadístico |
| RF-08 | Generación de Registros de Actividad |
| RF-09 | Gestión de Errores y Recuperación |
| RNF-01 | Consumo de Recursos |
| RNF-02 | Tiempo de Extracción |
| RNF-03 | Acceso a Datos |
| RNF-04 | Almacenamiento de Datos |
| RNF-05 | Tolerancia a Fallos |
| RNF-06 | Consistencia de Datos |
| RNF-07 | Disponibilidad del Sistema |
| RNF-08 | Documentación del Código |
| RNF-09 | Modularidad |
| RNF-10 | Compatibilidad con Sistemas Operativos |
| RNF-11 | Compatibilidad con Navegadores |
| RNF-12 | Usabilidad CLI |
| RNF-13 | Escalabilidad |

#### Tabla de Contenidos

1. Introducción
   1. 1Propósito

1.2 Alcance

1.3 Definiciones, Acrónimos y Abreviaturas

1.4 Referencias

1.5 Visión General del Documento

1. Descripción General

2.1 Perspectiva del Producto

2.2 Funciones del Producto

2.3 Características de los Usuarios

2.4 Restricciones

2.5 Suposiciones y Dependencias

2.6 Requisitos Futuros

1. Requisitos Específicos

3.1 Interfaces Externas

3.1.1 Interfaces de Usuario

3.1.2 Interfaces de Hardware

3.1.3 Interfaces de Software

3.1.4 Interfaces de Comunicación

4. Requisitos Funcionales

5. Requisitos No Funcionales

5.1 Requisitos de Rendimiento

5.2 Requisitos de Seguridad

5.3 Requisitos de Fiabilidad

5.4 Requisitos de Disponibilidad

5.5 Requisitos de Mantenibilidad

5.6 Requisitos de Portabilidad

5.7 Requisitos de Usabilidad

5.8 Requisitos de Escalabilidad

6. Atributos del Sistema

7. Información de Respaldo

7.1 Modelo de Datos

7.2 Diagrama de Flujo de Proceso

7.3 Escenarios de Uso Típico

7.4 Ejemplos de Salidas del Sistema

7.5 Matriz de Trazabilidad

7.6 Estrategia de Verificación y Validación

7.7 Riesgos y Priorización

7.8 Criterios de Aceptación

7.9 Gestión de Versiones del Código

8. Criterios de Aceptación

9. Glosario

10. Apéndices: A, B, C

11. Referencias Generales

# Introducción

## 1.1 Propósito

Este documento especifica los requisitos funcionales y no funcionales para el desarrollo de una herramienta de web scraping destinada a extraer información pública de la red social X (anteriormente Twitter). La especificación sigue el estándar IEEE 830-1998 y está dirigida al equipo de desarrollo, stakeholders y usuarios finales de RascaCazuelas S.A.

Este SRS servirá como contrato entre el equipo de desarrollo y los stakeholders, estableciendo las bases para la validación y verificación del sistema final.

## 1.2 Alcance

El sistema de web scraping, denominado "XScraper", permitirá automatizar la extracción de datos públicos de X (antes Twitter), procesar esta información y almacenarla en archivos CSV estructurados. Estos datos posteriormente podrán ser importados a la base de datos corporativa para análisis.

#### El sistema incluye:

 Extracción de contenido de tweets públicos

 Extracción de estadísticas asociadas (likes, retweets, comentarios)

 Recopilación de metadatos (fecha de publicación, URL, presencia de contenido multimedia)

 Procesamiento y validación de datos extraídos

 Almacenamiento estructurado en formato CSV  Registro de actividades (logs)

#### El sistema NO incluye:

 Extracción de datos de otras redes sociales

 Acceso a información privada o protegida por autenticación

 Análisis avanzado de los datos recopilados

 Interfaz gráfica de usuario (GUI)

 Programación interna de tareas (cron)

## 1.3 Definiciones, Acrónimos y Abreviaturas

 **API**: Application Programming Interface (Interfaz de Programación de Aplicaciones)

 **BD**: Base de Datos

 **CLI**: Command Line Interface (Interfaz de Línea de Comandos)  **CSV**: Comma-Separated Values (Valores Separados por Comas)

 **GUI**: Graphical User Interface (Interfaz Gráfica de Usuario)

 **HTTPS**: Hypertext Transfer Protocol Secure (Protocolo Seguro de Transferencia de Hipertexto)

 **IEEE**: Institute of Electrical and Electronics Engineers

 **RF**: Requisito Funcional

 **RNF**: Requisito No Funcional

 **SRS**: Software Requirements Specification (Especificación de Requisitos de Software)

 **Web Scraping**: Técnica de extracción automatizada de información de sitios web

 **XPath**: XML Path Language, lenguaje de consulta para seleccionar nodos en documentos XML/HTML

## 1.4 Referencias versión 1

 Documento de Visión y Alcance, RascaCazuelas S.A., Versión 1.0, 17/02/2025

 IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications

 Código fuente de twitter\_scraperV2.py, versión 2.0, 01/03/2025

 Términos de Servicio de X (Twitter): <https://twitter.com/en/tos>

 Laínez Fuentes, J. R. (2024). Gestión de proyectos de software: Metodologías, herramientas y casos prácticos (2ª ed.). RA-MA Editorial.

 Documentación oficial de Selenium WebDriver: <https://www.selenium.dev/documentation/>

## 1.5 Visión General del Documento

Este documento está organizado en cuatro secciones principales:

1. **Introducción**: Proporciona una visión general del propósito, alcance y estructura del documento.
2. **Descripción General**: Ofrece una perspectiva de alto nivel del sistema, incluyendo su contexto y funcionalidades principales.
3. **Requisitos Específicos**: Detalla todos los requisitos del sistema con suficiente detalle para permitir su diseño e implementación.
4. **Información de Respaldo**: Incluye material complementario como modelos de datos, diagramas de flujo, escenarios de uso, matriz de trazabilidad y glosario.

Cada requisito específico incluye un identificador único, descripción detallada, entradas, proceso, salidas, prioridad y dependencias cuando aplica.

# Descripción General

## 2.1Perspectiva del Producto

XScraper será un componente nuevo dentro de la infraestructura de análisis de datos de RascaCazuelas

S.A. El sistema se ejecutará de forma independiente como una herramienta de línea de comandos desarrollada en Python, utilizando Selenium WebDriver para automatizar la navegación web.

Los datos extraídos se almacenarán inicialmente en archivos CSV con formato predefinido, que luego podrán ser importados a la base de datos corporativa para su análisis por parte del equipo de analistas de datos.

El sistema no requiere integración directa con otros sistemas, aunque sí puede ser ejecutado mediante tareas programadas del sistema operativo (cron en Linux/macOS o Task Scheduler en Windows).

## 2.2Funciones del Producto

Las principales funciones del sistema son:

**Extracción automatizada de datos**: Capacidad para navegar por X (antes Twitter) y extraer información pública de tweets según parámetros definidos.

**Recopilación estructurada de datos**: Extracción de:

 Contenido textual de tweets

 Estadísticas (likes, retweets, comentarios, guardados)

 Metadatos (fecha de publicación, URL, presencia de multimedia)

 Cuenta de origen

**Validación y filtrado de datos**: Eliminación de contenido promocional, duplicados y procesamiento de fechas.

**Almacenamiento en formato CSV**: Generación de archivos CSV estructurados con los datos extraídos.

**Resumen estadístico**: Generación de resúmenes con conteos por cuenta analizada.

**Generación de registros de actividad**: Creación de logs detallados para seguimiento y depuración.

## 2.3 Características de los Usuarios

|  |  |  |
| --- | --- | --- |
| Tipo de Usuario | Nivel de  Experiencia | Responsabilidades |
| Administrador del  Sistema | Técnico | Configuración inicial del sistema, ejecución de scripts, gestión de  parámetros y monitoreo de operaciones |
| Analista de Datos | Medio | Definición de perfiles a extraer, revisión de datos extraídos y análisis  posterior |
| Usuario Final | Básico | Consulta de datos procesados a través de la plataforma de análisis  corporativa |
| C C | | |

**2.4 Restricciones**

 El sistema debe operar sin autenticación en X (antes Twitter), accediendo únicamente a contenido público.

 El sistema debe respetar los términos de servicio de X (antes Twitter) y las leyes de protección de datos.

 El desarrollo debe completarse en un plazo máximo de 4 meses.

 El sistema debe operar en equipos domésticos con recursos limitados (mínimo 4GB RAM, procesador dual-core).

 El sistema debe ser compatible con Chrome/Chromium versión 120 o superior.  La solución debe utilizar Python 3.9 o superior como lenguaje de programación.

 La solución debe utilizar Selenium WebDriver como principal herramienta de automatización.

## 2.5 Suposiciones y Dependencias

 Se asume que los usuarios tendrán conexión a Internet estable con velocidad mínima de 5Mbps.

 Se asume que la estructura DOM de la plataforma X (antes Twitter) permanecerá relativamente estable durante el desarrollo.

 El sistema depende de las políticas de acceso a datos públicos de X (antes Twitter), que pueden cambiar sin previo aviso.

 El sistema depende de la disponibilidad del navegador Chrome/Chromium y su correspondiente ChromeDriver.

 Se asume que los equipos donde se ejecutará el sistema tendrán Python 3.9+ instalado con los permisos necesarios para instalar dependencias.

## 2.6 Requisitos Futuros

 Soporte para extracción de datos de otras redes sociales.

 Implementación de una interfaz gráfica de usuario (GUI).

 Integración directa con sistemas de análisis de datos y visualización.

 Programador de tareas interno para automatización completa.

 Capacidad de análisis de sentimiento básico del contenido extraído.

# Requisitos Específicos

## 3.1 Interfaces Externas

### 3.1.1 Interfases de Usuario

El sistema implementará una interfaz de línea de comandos (CLI) con las siguientes características:

 Ejecución mediante comando Python con argumentos de configuración (nombres de perfil a extraer)

python twitter\_scraperV2.py --cuentas "elonmusk,BillGates" --tweets

 Ejemplo de ejecución:

100 --output ./data/

 Mensajes de progreso y estado en la consola durante la ejecución

 Notificaciones de éxito/error al finalizar cada extracción

 Mensajes detallados sobre el número de tweets extraídos por perfil

### 3.1.2 Interfaces de Hardware

El sistema no requiere hardware especializado, pero debe ejecutarse en equipos con:

 Procesador: Mínimo dual-core, 2.0 GHz o superior  Memoria RAM: Mínimo 4GB

 Almacenamiento: 1GB disponible para la aplicación y datos extraídos

 Conexión a Internet: 5Mbps mínimo, estable

### 3.1.3 Interfaces de Software

El sistema dependerá de:

 Sistema Operativo: Windows 10/11, macOS 12+, o distribuciones Linux modernas

 Python 3.9 o superior

 Selenium WebDriver 4.0 o superior

 Navegador Chrome/Chromium versión 120 o superior con su correspondiente ChromeDriver

 Bibliotecas estándar de Python: csv, re, time, datetime, random, os

### 3.1.4 Interfaces de Comunicación

 Conexión HTTPS para acceder a X (antes Twitter)  Protocolo TCP/IP para comunicación con Internet

 Sin requerimientos de comunicación con otros sistemas internos

# Requisitos Funcionales

### RF-01: Configuración de Parámetros de Extracción

**Descripción**: El sistema debe permitir configurar los perfiles de X (antes Twitter) desde los cuales se extraerán datos.

**Entradas**: Lista de nombres de usuario (perfiles) a extraer. **Proceso**: El sistema registra los perfiles objetivo para la extracción.

**Salidas**: Confirmación de perfiles configurados para extracción.

**Prioridad**: Alta

**Dependencias**: Ninguna

### RF-02: Extracción de Tweets por Perfil

**Descripción**: El sistema debe extraer tweets públicos de los perfiles configurados, incluyendo su contenido textual.

**Entradas**: Perfil configurado para extracción.

**Proceso**: El sistema navega al perfil especificado en X (antes Twitter) y extrae el contenido textual de los tweets disponibles.

**Salidas**: Contenido textual extraído de los tweets del perfil.

**Prioridad**: Alta

**Dependencias**: RF-01

### RF-03: Extracción de Métricas de Tweets

**Descripción**: El sistema debe extraer métricas asociadas a cada tweet (likes, retweets, comentarios, guardados).

**Entradas**: Tweet identificado para extracción.

**Proceso**: El sistema localiza y extrae los contadores de interacción para cada tweet.

**Salidas**: Valores numéricos de likes, retweets, comentarios y guardados por tweet.

**Prioridad**: Alta

**Dependencias**: RF-02

### RF-04: Extracción de Metadatos de Tweets

**Descripción**: El sistema debe extraer metadatos como fecha de publicación, URL y presencia de multimedia.

**Entradas**: Tweet identificado para extracción.

**Proceso**: El sistema identifica y extrae metadatos asociados a cada tweet.

**Salidas**: Fecha de publicación (formato ISO 8601), URL completa y flags de presencia multimedia.

**Prioridad**: Alta

**Dependencias**: RF-02

### RF-05: Validación y Filtrado de Datos

**Descripción**: El sistema debe validar y filtrar los datos extraídos para garantizar su calidad.

**Entradas**: Datos extraídos sin procesar.

**Proceso**: El sistema aplica reglas de validación (eliminación de tweets promocionados o duplicados, filtrado por antigüedad máxima de 2 años).

**Salidas**: Datos validados y estructurados.

**Prioridad**: Alta

**Dependencias**: RF-02, RF-03, RF-04

### RF-06: Almacenamiento de Datos en CSV

**Descripción**: El sistema debe almacenar los datos procesados en archivos CSV estructurados.

**Entradas**: Datos validados y estructurados.

**Proceso**: El sistema genera archivos CSV con nombres que incluyen el perfil y timestamp.

**Salidas**: Archivos CSV con datos extraídos y validados.

**Prioridad**: Alta

**Dependencias**: RF-05

### RF-07: Generación de Resumen Estadístico

**Descripción**: El sistema debe generar un resumen estadístico de los datos extraídos.

**Entradas**: Datos extraídos y procesados por perfil.

**Proceso**: El sistema contabiliza métricas por perfil y genera un archivo de resumen.

**Salidas**: Archivo CSV de resumen con conteos por perfil.

**Prioridad**: Media

**Dependencias**: RF-06

### RF-08: Generación de Registros de Actividad

**Descripción**: El sistema debe generar registros detallados de todas las operaciones realizadas.

**Entradas**: Eventos del sistema durante la ejecución.

**Proceso**: El sistema registra cada actividad con timestamp y detalles relevantes.

**Salidas**: Mensajes de consola y potencialmente archivos de log.

**Prioridad**: Media

**Dependencias**: RF-01, RF-02, RF-03, RF-04, RF-05, RF-06, RF-07

### RF-09: Gestión de Errores y Recuperación

**Descripción**: El sistema debe detectar y gestionar errores durante la extracción y procesamiento de datos.

**Entradas**: Errores detectados durante la operación.

**Proceso**: El sistema captura excepciones específicas, registra el error y ejecuta procedimientos de recuperación para continuar con la siguiente extracción.

#### Tipos de errores manejados:

:

NoSuchElementException

Cuando un elemento esperado no se encuentra en la página

:

TimeoutException

Cuando una operación tarda demasiado en completarse

:

StaleElementReferenceException

Cuando un elemento del DOM ya no es válido

:

WebDriverException

Errores generales del WebDriver

#### Estrategia de recuperación:

 Implementación de bloques try/except para capturar errores específicos

 Omisión del tweet problemático y continuación con el siguiente

 Reintentos automáticos con tiempos de espera incrementales

 Registro detallado del error para análisis posterior

**Salidas**: Notificaciones de error y registros de recuperación.

**Prioridad**: Alta

**Dependencias**: RF-02, RF-03, RF-04, RF-05, RF-06

# 5. Requisitos No Funcionales

## 5.1 Requisitos de Rendimiento

#### RNF-01: Consumo de Recursos

**Descripción**: El sistema debe realizar extracciones con un uso de recursos limitado.

**Métrica**: El proceso de extracción no debe consumir más del 30% de CPU y 2GB de RAM en equipos que cumplan los requisitos mínimos.

**Prioridad**: Alta

#### RNF-02: Tiempo de Extracción

**Descripción**: El sistema debe extraer datos con una velocidad razonable.

**Métrica**: El tiempo promedio de extracción no debe exceder los 5 minutos por cada 50 tweets.

**Prioridad**: Media

## 5.2 Requisitos de Seguridad

#### RNF-03: Acceso a Datos

**Descripción**: El sistema solo debe acceder a información pública de X (antes Twitter). **Métrica**: No debe utilizarse autenticación ni acceder a contenido privado o protegido. **Prioridad**: Alta

#### RNF-04: Almacenamiento de Datos

**Descripción**: Los datos extraídos deben almacenarse de forma segura.

**Métrica**: Los archivos CSV se guardarán en directorios con permisos apropiados según el sistema operativo.

**Prioridad**: Media

## 5.3 Requisitos de Fiabilidad

#### RNF-05: Tolerancia a Fallos

**Descripción**: El sistema debe ser capaz de recuperarse automáticamente de errores comunes.

**Métrica**: El 95% de los errores durante la extracción de un perfil no deben interrumpir el procesamiento de otros perfiles configurados.

**Prioridad**: Alta

#### RNF-06: Consistencia de Datos

**Descripción**: Los datos extraídos deben mantener consistencia y estructura.

**Métrica**: 100% de los archivos CSV generados deben mantener el formato definido y ser importables sin errores.

**Prioridad**: Alta

## 5.4 Requisitos de Disponibilidad

#### RNF-07: Disponibilidad del Sistema

**Descripción**: El sistema debe estar disponible para ejecutarse cuando sea necesario.

**Métrica**: El sistema debe poder iniciar la extracción en el 99% de los intentos de ejecución, siempre que se cumplan los requisitos de hardware/software.

**Prioridad**: Media

## 5.5 Requisitos de Mantenibilidad

#### RNF-08: Documentación del Código

**Descripción**: El código debe estar bien documentado para facilitar su mantenimiento.

**Métrica**: Cobertura de documentación del 90% (docstrings, comentarios) y cumplimiento de estándares PEP 8 para Python.

**Prioridad**: Media

#### RNF-09: Modularidad

**Descripción**: El código debe estructurarse de forma modular para facilitar mantenimiento y extensiones.

**Métrica**: Funciones independientes con responsabilidad única y máximo 100 líneas por función.

**Prioridad**: Media

### 5.6 Requisitos de Portabilidad

#### RNF-10: Compatibilidad con Sistemas Operativos

**Descripción**: El sistema debe ser compatible con diferentes sistemas operativos.

**Métrica**: Funcionamiento verificado en Windows 10/11, macOS 12+ y Ubuntu 20.04 o superior.

**Prioridad**: Media

#### RNF-11: Compatibilidad con Navegadores

**Descripción**: El sistema debe ser compatible con diferentes versiones del navegador Chrome/Chromium.

**Métrica**: Funcionamiento correcto con Chrome/Chromium versiones 120 a 125.

**Prioridad**: Media

## 5.7 Requisitos de Usabilidad

#### RNF-12: Usabilidad CLI

**Descripción**: El sistema debe proporcionar una experiencia de usuario adecuada para usuarios técnicos desde la línea de comandos.

**Métrica**: Los mensajes de error deben ser claros y sugerir soluciones; los comandos deben tener ayuda integrada y argumentos con nombres intuitivos.

**Prioridad**: Media

**Dependencias**: RF-01, RF-08

## 5.8 Requisitos de Escalabilidad

#### RNF-13: Escalabilidad

**Descripción**: El sistema debe ser capaz de manejar un volumen creciente de datos sin modificaciones significativas.

**Métrica**: El sistema debe poder procesar hasta 1000 tweets por cuenta sin cambios en el código ni degradación del rendimiento superior al 15%.

**Prioridad**: Baja

**Dependencias**: RNF-01, RNF-02

# 6. Atributos del Sistema

### Seguridad

El sistema garantizará el acceso únicamente a información pública, sin almacenar credenciales de acceso ni realizar operaciones de escritura en X (antes Twitter).

### Fiabilidad

El sistema implementará mecanismos robustos de manejo de excepciones para recuperarse de errores comunes y continuar la operación incluso cuando ciertos elementos no se pueden extraer.

### Mantenibilidad

El código fuente seguirá convenciones estándar de Python (PEP 8) y estará organizado en funciones modulares con comentarios explicativos y docstrings.

### Portabilidad

El sistema será desarrollado utilizando tecnologías multiplataforma (Python, Selenium) para asegurar su funcionamiento en diferentes sistemas operativos.

# 7. Información de Respaldo

## 7.1 Modelo de Datos

El sistema utilizará el siguiente modelo de datos simplificado para almacenar la información extraída:

#### Archivos CSV individuales por perfil:

:

texto

Texto completo del tweet (cadena)

:

fecha

Fecha de publicación (formato ISO 8601)

:

url

URL completa del tweet

:

me\_gusta

Número de "me gusta" (entero)

:

retweets

Número de retweets (entero)

:

comentarios

Número de comentarios (entero)

:

guardados

Número de marcadores/guardados (entero)

:

tiene\_media

Indicador de presencia de multimedia (booleano)

:

cuenta

Nombre del perfil extraído (cadena)

:

timestamp\_extraccion

Marca de tiempo de la extracción (datetime)

#### Archivo CSV de resumen:

:

cuenta

Nombre del perfil extraído

:

total\_tweets

Número total de tweets extraídos

:

promedio\_me\_gusta

Promedio de "me gusta" por tweet

:

promedio\_retweets

Promedio de retweets por tweet

:

promedio\_comentarios

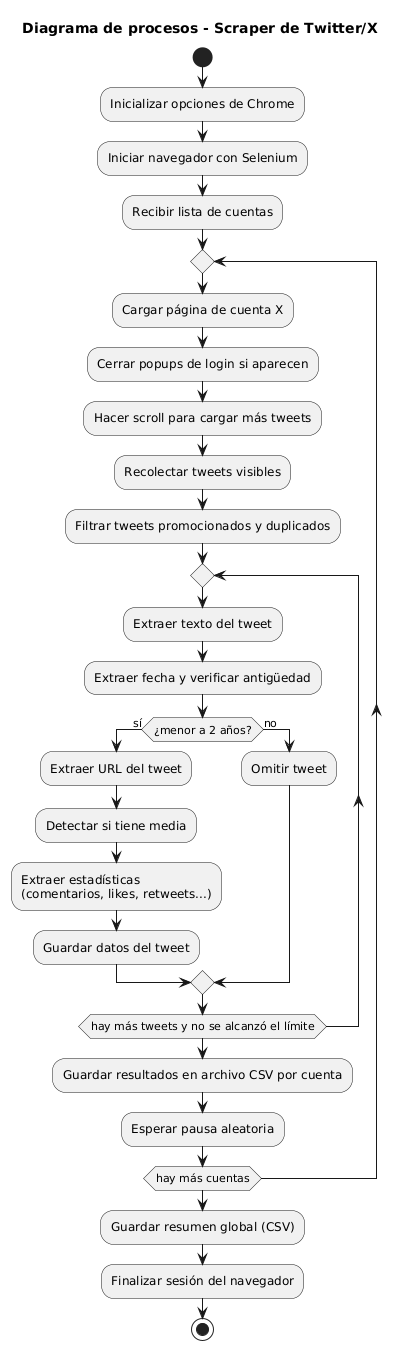
Promedio de comentarios por tweet

:

fecha\_extraccion

Fecha de la extracción

## 7.2 Diagrama de Flujo de Proceso



## 7.3 Escenarios de Uso Típico

**Escenario 1: Extracción básica de una cuenta**

**Usuario:** Administrador del sistema

**Entrada:**

python twitter\_scraperV2.py --cuentas "<https://x.com/elonmusk>" --tweets 50 --output ./data/

**Proceso:**

1. Inicializa navegador en modo headless.
2. Accede al perfil elonmusk.
3. Realiza scroll dinámico para cargar 50 tweets.
4. Extrae texto, métricas, metadatos.
5. Filtra tweets duplicados o antiguos.
6. Almacena en un archivo CSV.

**Salida:**

./data/elonmusk\_YYYYMMDD\_HHMMSS.csv

./data/resumen\_YYYYMMDD\_HHMMSS.csv

Log en consola y archivo con nivel DEBUG.

**Escenario 2: Extracción de múltiples cuentas**

**Usuario:** Analista de Datos

**Entrada:**

python twitter\_scraperV2.py --cuentas "<https://x.com/elonmusk,https://x.com/BillGates>" --tweets 100 --output ./data/

**Proceso:**

1. El sistema itera por cada cuenta.
2. Repite el proceso del escenario 1.
3. Crea archivos CSV por cuenta y uno de resumen.

**Salida:**

elonmusk\_\*.csv, BillGates\_\*.csv

resumen\_\*.csv

Consola con progreso y logs de ejecución.

## 7.4 Ejemplos de Salidas del Sistema

**Ejemplo de archivo CSV por perfil:** elonmusk\_20250506\_120530.csv

texto,fecha,url,me\_gusta,retweets,comentarios,compartidos,tiene\_media,cuenta,timestamp\_extraccion  
"AI is transforming the world.",2025-05-05T16:32:15Z,https://x.com/elonmusk/status/123456,2341,198,321,76,True,elonmusk,2025-05-06T12:05:30Z

**Ejemplo de archivo de resumen:** resumen\_20250506\_121045.csv

cuenta,total\_tweets,promedio\_me\_gusta,promedio\_retweets,promedio\_comentarios,fecha\_extraccion  
elonmusk,50,8123.2,245.4,95.3,2025-05-06

## 7.5 Matriz de Trazabilidad

|  |  |  |
| --- | --- | --- |
| **Requisito** | **Módulo/Función en Código** | **Descripción** |
| RF-01 | parse\_arguments() | Procesamiento de argumentos CLI |
| RF-02 | scrape\_account() | Extracción de tweets por perfil |
| RF-03 | extract\_tweet\_stats() | Extracción de métricas |
| RF-04 | extract\_tweet\_metadata() | Extracción de metadatos |
| RF-05 | validate\_tweet() | Validación y filtrado |
| RF-06 | save\_to\_csv() | Almacenamiento en CSV |
| RF-07 | generate\_summary() | Generación de resumen estadístico |
| RF-08 | setup\_logging() | Configuración de registros |
| RF-09 | handle\_exceptions() | Manejo de excepciones |
| RNF-01 | setup\_webdriver() | Optimización de recursos |

## 7.6 Estrategia de Verificación y Validación (V&V)

Verificación funcional: Ejecutar scripts para cada requisito funcional con entradas conocidas y comparar resultados esperados.

Validación de formato: Inspección manual y automática de los archivos CSV generados para confirmar estructura y contenido.

Pruebas de tolerancia a fallos: Ejecutar con perfiles no válidos, sin conexión o con cambios en el DOM.

Pruebas de rendimiento: Medir uso de CPU y memoria y tiempos de ejecución para distintos volúmenes de datos.

Revisión de logs: Confirmar trazabilidad y manejo de errores registrados.

## 7.7 Riesgos y Priorización

|  |  |  |  |
| --- | --- | --- | --- |
| **Riesgo** | **Impacto** | **Probabilidad** | **Mitigación** |
| Cambios en la estructura DOM de X | Alto | Alta | Modularizar scraping con selectores robustos |
| Detección del scraper por X y bloqueo de acceso | Medio | Media | Usar user-agent aleatorio y pausas entre interacciones |
| Fallo en ChromeDriver o incompatibilidad con versiones | Bajo | Media | Control de versiones y pruebas continuas |

## 7.8 Criterios de Aceptación

|  |  |
| --- | --- |
| **Requisito** | **Criterio de Aceptación** |
| RF-02 | El sistema debe guardar al menos 10 tweets por perfil con estructura definida en CSV. |
| RF-06 | El archivo CSV debe incluir: texto, fecha, url, me\_gusta, retweets, comentarios, cuenta. |
| RF-09 | Ante error en un tweet, continuar con el siguiente y registrar el incidente. |
| RNF-01 | El scraper debe usar máximo 30% de CPU y 2GB de RAM en condiciones estándar. |

## 7.9 Gestión de Versiones del Código

|  |  |  |
| --- | --- | --- |
| **Versión** | **Fecha** | **Cambios clave** |
| 1.0 | 01/03/2025 | Versión inicial con scraping básico y guardado en CSV. |
| 2.0 | 05/05/2025 | Añadido soporte multicuenta, logs, manejo de errores, y generación de resumen estadístico. |

# 8. Criterios de Aceptación

Los criterios de aceptación permiten validar que cada requisito ha sido implementado correctamente y cumple con su propósito dentro del sistema. A continuación, se presentan los criterios correspondientes a los requisitos clave del sistema XSraper.

|  |  |
| --- | --- |
| **Requisito** | **Criterio de Aceptación** |
| RF-01 | El sistema debe aceptar correctamente una lista de perfiles como entrada y mostrarlos por consola. |
| RF-02 | Al ejecutar con un perfil válido, el sistema debe extraer al menos 10 tweets con contenido textual. |
| RF-03 | Cada tweet extraído debe incluir valores numéricos para likes, retweets, comentarios y guardados. |
| RF-04 | Cada tweet debe registrar correctamente la fecha ISO 8601, URL y si incluye multimedia. |
| RF-05 | El sistema debe filtrar correctamente tweets duplicados, promocionados o con antigüedad mayor a 2 años. |
| RF-06 | Se debe generar un archivo CSV con encabezados y columnas definidas. El nombre del archivo debe incluir el nombre del perfil y un timestamp. |
| RF-07 | Se debe generar un archivo de resumen CSV con totales y promedios de métricas por cuenta. |
| RF-08 | El sistema debe imprimir en consola mensajes de progreso, éxito o error durante toda la ejecución. |
| RF-09 | Ante cualquier excepción, el sistema debe continuar con la siguiente cuenta o tweet y registrar el error en consola o log. |
| RNF-01 | En un equipo con 4 GB de RAM, el uso del sistema no debe superar el 30% de CPU y 2 GB de RAM. |
| RNF-02 | El tiempo promedio para extraer 50 tweets no debe superar los 5 minutos. |
| RNF-06 | Todos los archivos generados deben poder ser abiertos con software de análisis sin errores de formato. |

# 9. Glosario

**Tweet:** Publicación individual en X (antes Twitter), usualmente hasta 280 caracteres.

**CSV:** Formato plano de almacenamiento de datos en texto con valores separados por comas.

**XPath:** Lenguaje de selección para navegar estructuras XML o HTML.

**DOM:** Document Object Model, estructura jerárquica de una página HTML.

**CLI:** Command Line Interface. Interfaz basada en comandos de texto.

**Scraping:** Técnica de programación para extraer datos de páginas web de forma automatizada.

# 10. Apéndices

## Apéndice A: Excepciones Manejadas

Durante la ejecución del sistema XSraper, pueden ocurrir errores derivados de la interacción automatizada con el navegador o la plataforma X (antes Twitter). A continuación se describen las principales excepciones contempladas en el sistema, así como las estrategias de manejo implementadas.

|  |  |  |  |
| --- | --- | --- | --- |
| **Excepción** | **Descripción** | **Causa común** | **Estrategia de manejo en XSraper** |
| NoSuchElementException | Indica que un elemento esperado no fue encontrado en el DOM. | El tweet o métrica aún no ha cargado o el selector está mal definido. | El sistema registra el error y omite el tweet correspondiente. |
| TimeoutException | El navegador excede el tiempo máximo esperando que algo ocurra. | Scroll lento, red lenta, o cambios en el DOM de X. | Se aplica reintento con espera incremental. |
| StaleElementReferenceException | El elemento ya no está disponible porque el DOM cambió. | Se ejecutó un scroll o refresco que reemplazó el nodo HTML. | El sistema ignora el tweet en curso y continúa con el siguiente. |
| WebDriverException | Error general de Selenium WebDriver. | Pérdida de sesión, fallo de ChromeDriver, incompatibilidad o cierre abrupto. | Se intenta reiniciar el navegador o registrar el fallo crítico. |

### Mecanismos generales de recuperación

* Uso de bloques try/except para encapsular interacciones con el navegador.
* Reintentos limitados con sleep() incremental.
* Registro detallado en consola o log para análisis posterior.
* Continuación del scraping con el siguiente tweet o cuenta en caso de error local.

## Apéndice B: Estructura de Archivos del Sistema

/xscraper/  
├── twitter\_scraperV2.py  
├── /data/  
│ ├── /raw/  
│ └── /summary/  
├── /logs/  
├── requirements.txt

## Apéndice C: Tiempos Estimados de Extracción

Los tiempos estimados para la ejecución del sistema XSraper se han calculado considerando su comportamiento real bajo condiciones operativas estándar:

* Conexión a internet estable (≥5 Mbps).
* Equipo con procesador dual-core (2.0 GHz) y 4 GB de RAM.
* Navegación automatizada con Selenium WebDriver en modo headless.
* Introducción de pausas aleatorias para evitar detección por la plataforma X (antes Twitter).

### Cálculo desglosado:

#### 1. Tiempo por scroll

* X carga entre 6 y 8 tweets por scroll.
* Cada scroll requiere entre **1 y 2 segundos** (incluyendo pausa aleatoria).
* Scrolls necesarios por volumen de tweets:

|  |  |
| --- | --- |
| **Tweets a extraer** | **Scrolls estimados** |
| 50 | 7–9 |
| 100 | 12–15 |
| 500 | 60–70 |

#### 2. Tiempo por tweet procesado

* Extracción de texto, métricas, metadatos: ~0.5 segundos/tweet.
* Validación y filtrado: incluido en el mismo tiempo.

#### 3. Total estimado

|  |  |  |  |
| --- | --- | --- | --- |
| **Cantidad de Tweets** | **Tiempo por scrolls** | **Tiempo por extracción** | **Tiempo total estimado** |
| 50 | 15–18 s | 25 s | ~2 minutos |
| 100 | 25–30 s | 50 s | ~4 minutos |
| 500 | 120–140 s | 250 s | ~8 minutos |

### Tiempos Estimados de Ejecución

|  |  |
| --- | --- |
| **Cantidad de Tweets** | **Tiempo Estimado** |
| 50 | 2 minutos |
| 100 | 4 minutos |
| 500 | 8 minutos |

**Notas:**

* Estos valores pueden variar ligeramente según la conexión, carga del servidor de X, y uso de recursos locales.
* El código incluye pausas aleatorias (sleep) para evitar la detección de scraping automatizado, lo cual introduce variabilidad.

# 11. Referencias en formato APA (7.ª edición)

1. Institute of Electrical and Electronics Engineers. (1998). *IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)*. IEEE. <https://standards.ieee.org/standard/830-1998.html>
2. Selenium. (s. f.). *The Selenium WebDriver documentation*. Selenium.dev. <https://www.selenium.dev/documentation/>
3. Twitter, Inc. (s. f.). *Terms of service*. X (antes Twitter). <https://twitter.com/en/tos>
4. Van Rossum, G., & Python Software Foundation. (2001–2025). *The Python Language Reference (version 3.9+)*. Python.org. <https://docs.python.org/3/>
5. Python Software Foundation. (s. f.). *PEP 8 – Style Guide for Python Code*. Python.org. <https://peps.python.org/pep-0008/>
6. Laínez Fuentes, J. R. (2024). *Gestión de proyectos de software: Metodologías, herramientas y casos prácticos* (2.ª ed.). RA-MA Editorial.
7. RascaCazuelas S.A. (2025). *Documento de visión y alcance del sistema XSraper* (Versión 1.0, 17 de febrero de 2025).
8. Valdés Palafox, L., & Ortega Almendares, A. M. (2025). *twitter\_scraperV2.py* RascaCazuelas S.A.
9. Google. (s. f.). *ChromeDriver – WebDriver for Chrome*. Chrome for Developers. <https://sites.google.com/chromium.org/driver/>