

Prompt de Ingeniería para el Desarrollo del Agente QMC

Contexto del Proyecto: Desarrollar un sistema de Scraping Agéntico para automatizar la recolección y filtrado de datos desde la Qlik Management Console (QMC) de una empresa financiera. El objetivo es monitorear tareas específicas mediante un flujo de trabajo cíclico y robusto.

Stack Tecnológico Requerido:

- Orquestación: LangGraph (para manejar estados, reintentos y lógica de decisión).
- LLM: Grok (vía LangChain/xAI) para razonamiento y estructuración de datos.
- Automatización de Navegador: Playwright (Python) para manejar login, SPA y clicks dinámicos.

Especificaciones de la Tarea de Scraping:

1. Acceso y Autenticación: * URL inicial: <https://apqs.grupoefe.pe/qmc/tasks>

Credenciales: Usuario svrlimprdap011\consultorqs1 y Password proporcionado. El login se realiza mediante formulario o Enter.

2. Interacción con la Interfaz (Qlik Sense QMC):

- La página es una SPA que carga mediante require.js. Se debe esperar a que el spinner desaparezca y la tabla .qmc-grid sea visible.
- Filtro 1: En la columna "Last execution", abrir el selector de filtro y seleccionar la opción "Today".
- Filtro 2: En la columna "Tags", abrir el filtro, escribir "FE_HITOS" en el buscador y aplicar.

3. Extracción y Procesamiento:

- Recolectar todas las filas resultantes de la tabla.
- El LLM debe convertir el texto crudo de la tabla en un JSON estructurado con campos: Name, Status, Last execution, Next execution, Tags.

Estructura del Grafo (LangGraph):

Nodo Navegador: Realiza el login y aplica los filtros. Devuelve el contenido de la tabla o un error si no carga. (o podemos separarlo en dos nodos uno para el login y otro para el filtro)

Nodo Analista: Toma la data cruda, la procesa con Grok para validarla y estructurarla.

Lógica de Control: Si el nodo Navegador falla por tiempo de carga, debe reintentar hasta 3 veces antes de finalizar con error.

Instrucciones para el LLM receptor: "Genera el código en un cuaderno de Jupyter notebook .ipynb de manera modular para poder ir probando celda por celda utilizando langchain, langgraph y playwright. Incluye la definición del State de LangGraph y las funciones de las herramientas (Tools) decoradas adecuadamente. Asegúrate de manejar los selectores de CSS de forma genérica o explicativa para que puedan ser ajustados tras una inspección de elementos."

Html de la página:

<https://apqs.grupoefe.pe/qmc/tasks>

```
<!doctype html>

<html class="qmc" lang="en-US">

<head>

    <meta charset="utf-8" />

    <meta http-equiv="X-UA-Compatible" content="IE=edge" />

    <meta name="google" content="notranslate" />

    <meta name="HandheldFriendly" content="True" />

    <meta name="MobileOptimized" content="320">

        <meta name="viewport" content="width=device-width, initial-scale=1.0, maximum-scale=1.0, minimum-scale=1.0, user-scalable=no, minimal-ui">

    <meta name="apple-mobile-web-app-capable" content="yes">

    <meta name="apple-mobile-web-app-status-bar-style" content="black">

    <meta http-equiv="cleartype" content="on">

    <title>Qlik Management Console</title>

    <base href="/" />

    <link rel="stylesheet" href="../resources/qmc.css?1705491826149">

    <link rel="shortcut icon" href="../resources/favicon_light.ico" media="(prefers-color-scheme:no-preference)">

        <link rel="shortcut icon" href="../resources/favicon_dark.ico" media="(prefers-color-scheme:dark)" id="dark-scheme-icon">
```

```
    <link rel="shortcut icon" href="../resources/favicon_light.ico"
media="(prefers-color-scheme:light)" id="light-scheme-icon">

    <link rel="apple-touch-icon" href="../resources/qmc/img/android-chrome-
60x60.png">

        <link rel="apple-touch-icon" sizes="76x76"
href="../resources/qmc/img/apple-touch-icon-76x76.png">

        <link rel="apple-touch-icon" sizes="120x120"
href="../resources/qmc/img/android-chrome-120x120.png">

        <link rel="apple-touch-icon" sizes="152x152"
href="../resources/qmc/img/apple-touch-icon-152x152.png">

    <noscript>

        <style>      .qmc-loading .spinner-overlay { display: none; }</style>

    </noscript>

</head>

<body>

    <div id="qmc">

        <div class="qv-preload-icons"></div>

        <div class="qv-preload-font"></div>

        <div class="qmc-loading">

            <noscript>

                <qmc-error-popup class="qmc-popup-overlay">

                    <qmc-popup popup-name="qmcErrorPopup">

                        <div class="qmcErrorPopup">

                            <div class="popupHeader">

                                <div
class="topHeader"></div>

                                <div
class="bottomHeader">

                                    <header>Initialization error</header>

                                    </div>


```

```
        </div>

        </div>

        <div class="popupContent">

            <div class="popupMain">

                <main class="qmc-error-
popup-main">

                    <i class="icon-qmc-
error qmc-flex-none"></i>

                    <div class="qmc-
flex-fill">The Qlik Management Console will not work if JavaScript is
disabled.</div>

                </main>

            </div>

            </div>

            <div class="popupFooter">

            </div>

        </qmc-popup>

    </qmc-error-popup>

</noscript>

<qmc-spinner ng-non-bindable>

    <div class="spinner-overlay"><div class="qv-
loader"></div></div>

    </qmc-spinner>

    <script
src="../resources/qmc/assets/external/requirejs/require.js?1705491826149" data-
main="../resources/qmc/require-main" ></script>

</div>

</div>

</body>

</html>
```