**Product Backlog**

**Épica 1: Web Scraping de portales competidores**

El sistema debe realizar web scraping de portales de competidores utilizando RPA, para obtener información relevante sobre productos, precios y descuentos.

**Historias de usuario derivadas:**

1. **Historia de Usuario: Web Scraping para "Cuidado con el Perro"**
   * El sistema debe realizar web scraping específicamente del portal "Cuidado con el Perro", para obtener información sobre los productos disponibles, precios y promociones actuales.
2. **Historia de Usuario: Web Scraping para "Shasa"**
   * El sistema debe web scraping del portal "Shasa", para obtener información sobre los productos, precios y descuentos ofrecidos.
3. **Historia de Usuario: Manejo de errores en el scraping**
   * El sistema debe manejar errores comunes durante el proceso de scraping para asegurar que el scraping no falle completamente y pueda recuperarse.
4. **Historia de Usuario: Programación del scraping en intervalos definidos**
   * El scraping se debe ejecutar en intervalos de tiempo definidos (diarios, semanales), para que los datos se mantengan actualizados sin intervención manual.
5. **Historia de Usuario: Notificaciones en caso de fallos en el scraping**
   * Que se pueda recibir notificaciones por correo o a través del sistema en caso de que falle el proceso de scraping, para poder intervenir y solucionar problemas de manera oportuna.

**Épica 2: Extracción de datos clave**

El sistema debe extraer los datos clave como productos, precios y descuentos de los portales de competidores, para poder analizarlos más tarde.

**Historias de usuario derivadas:**

1. **Historia de Usuario: Extracción de productos**
   * El sistema debe extraer específicamente la información de productos (nombre, descripción, categoría), para poder analizar la variedad de productos ofrecidos por los competidores.
2. **Historia de Usuario: Extracción de precios**
   * El sistema debe extraer los precios de los productos de los competidores, para compararlos y realizar análisis de mercado.
3. **Historia de Usuario: Extracción de descuentos y promociones**
   * El sistema debe identificar y extraer los descuentos o promociones vigentes en los sitios competidores.
4. **Historia de Usuario: Filtrado y limpieza de datos extraídos**
   * Los datos extraídos deben pasar por un proceso de filtrado y limpieza automática (eliminar duplicados, corregir errores), para asegurar que los datos sean precisos y útiles para el análisis.
5. **Historia de Usuario: Normalización de datos para diferentes portales**
   * El sistema normalice los datos extraídos de diferentes portales (formato de precios, categorías de productos), para que todos los datos tengan un formato uniforme y puedan ser comparados fácilmente.

**Épica 3: Almacenamiento de datos en base de datos**

La información obtenida del web scraping debe ser almacenada automáticamente en una base de datos, para asegurar que los datos están listos para su análisis.

**Historias de usuario derivadas:**

1. **Historia de Usuario: Crear esquema de base de datos para productos**
   * Se debe crear un esquema específico para almacenar la información de productos extraídos, para que los datos estén organizados y listos para análisis.
2. **Historia de Usuario: Crear esquema de base de datos para precios**
   * Se debe crear un esquema en la base de datos para almacenar los precios extraídos de los portales, de manera que se puedan analizar fácilmente los cambios en los precios.
3. **Historia de Usuario: Crear esquema de base de datos para descuentos y promociones**
   * Se debe crear un esquema para almacenar la información sobre descuentos y promociones, para poder hacer un seguimiento del comportamiento de las ofertas en el tiempo.
4. **Historia de Usuario: Almacenamiento automatizado de datos extraídos**
   * Los datos extraídos del scraping se deben almacenar automáticamente en la base de datos, para que no haya necesidad de procesos manuales que introduzcan errores.
5. **Historia de Usuario: Gestión de versiones de datos**
   * El sistema debe gestionar las versiones de los datos almacenados (mantener registros de precios antiguos), para poder comparar las variaciones a lo largo del tiempo.
6. **Historia de Usuario: Validación de datos antes de almacenar**
   * El sistema debe validar los datos antes de almacenarlos (revisar que no haya valores faltantes, formatos incorrectos, etc.), para asegurar que los datos almacenados sean correctos y completos.

**Épica 7: Seguridad y control de acceso a dashboards**

El sistema debe implementar mecanismos de seguridad que aseguren que solo usuarios autorizados puedan acceder a los dashboards y sus datos, protegiendo la información sensible del sistema.

**Historias de usuario derivadas:**

1. **Historia de Usuario: Control de acceso para dashboards**
   * Como administrador, quiero que el sistema implemente control de acceso a los dashboards, de modo que solo usuarios autorizados puedan ver o interactuar con los datos.
2. **Historia de Usuario: Gestión de roles y permisos**
   * El sistema debe permitir la configuración de roles y permisos específicos para que los usuarios solo accedan a las áreas que les correspondan según su perfil.
3. **Historia de Usuario: Integración con sistemas de autenticación**
   * El sistema debe integrarse con mecanismos de autenticación como OAuth, LDAP o Active Directory, para gestionar de manera eficiente los accesos y garantizar la seguridad de la información.
4. **Historia de Usuario: Auditoría de accesos**
   * El sistema debe registrar todas las acciones de acceso a los dashboards, permitiendo generar informes de auditoría que ayuden a identificar accesos indebidos o fallos de seguridad.
5. **Historia de Usuario: Notificación de accesos no autorizados**
   * En caso de intentos de acceso no autorizados, el sistema debe notificar al administrador a través de correo electrónico o alertas en el sistema, para intervenir oportunamente.

**Épica 8: Interfaz de usuario amigable y personalizable para dashboards**

El sistema debe proporcionar una interfaz web intuitiva y fácil de usar para los usuarios finales, permitiendo la personalización de los dashboards para que se adapten a las preferencias y necesidades de visualización de cada usuario.

**Historias de usuario derivadas:**

1. **Historia de Usuario: Interfaz amigable para dashboards**
   * Como usuario final, quiero que la interfaz del portal web sea amigable para que pueda navegar y acceder a los datos fácilmente sin necesidad de conocimientos técnicos avanzados.
2. **Historia de Usuario: Personalización de dashboards**
   * El sistema debe permitir la personalización de los dashboards (cambio de layout, colores, tipos de gráficos, filtros) para que cada usuario pueda visualizar la información de la manera más clara posible.
3. **Historia de Usuario: Guardado de preferencias de personalización**
   * El sistema debe guardar las preferencias de personalización de cada usuario para que, al volver a iniciar sesión, los dashboards mantengan la misma configuración seleccionada previamente.
4. **Historia de Usuario: Opciones predefinidas de personalización**
   * El sistema debe ofrecer opciones de personalización predefinidas para usuarios que no deseen configurar los dashboards desde cero, agilizando el proceso de adaptación.
5. **Historia de Usuario: Soporte para dispositivos móviles**
   * El sistema debe adaptar la interfaz de usuario para ser amigable y funcional en dispositivos móviles (responsive design), para que los usuarios puedan acceder a los dashboards desde cualquier dispositivo.

**Épica 9: Modularidad del sistema**

El sistema debe ser modular, permitiendo la adición de nuevos portales de competidores y funcionalidades sin afectar el rendimiento o la arquitectura existente.

**Historias de usuario derivadas:**

1. **Historia de Usuario: Sistema modular para nuevos portales**
   * Como desarrollador, quiero que el sistema sea modular para poder añadir nuevos portales de competidores fácilmente, sin afectar el rendimiento o la estructura existente del sistema.
2. **Historia de Usuario: Aislamiento de módulos**
   * El sistema debe aislar cada módulo de forma que se puedan agregar, actualizar o eliminar sin causar interferencias o fallas en el sistema principal.
3. **Historia de Usuario: Desempeño estable con módulos adicionales**
   * El sistema debe asegurar que, al añadir nuevos módulos, el rendimiento y la estabilidad general no se vean comprometidos, garantizando una operación fluida.
4. **Historia de Usuario: Integración de APIs modulares**
   * El sistema debe permitir que nuevos módulos se integren a través de APIs definidas, facilitando la expansión de funcionalidades sin modificar el código base.
5. **Historia de Usuario: Pruebas automatizadas para nuevos módulos**
   * El sistema debe incluir pruebas automatizadas para verificar que cada nuevo módulo funciona correctamente antes de ser implementado en producción, evitando errores que afecten a todo el sistema.