Planificación Proyecto De Análisis De Datos

Integrantes:

- Ardanny Romero
- Ariel Macias
- Brandon Huera
- Juan Lucero

Objetivo:

Determinar la guía de trabajo para la elaboración del proyecto, misma que además debe ser documentada paso a paso por parte de todos los integrantes para así finalmente construir el informe general (los informes independientes y el general irán en formato IEEE).

Análisis de temáticas

1. Tráfico vehicular en las 5 principales ciudades del Ecuador:

- Facilidad: Moderada.
- Fuentes: No existen muchos datasets públicos y centralizados. La mejor opción sería usar APIs como la de Waze for Cities o Google Maps Directions API para obtener datos de tiempos de viaje y congestión en rutas específicas de Quito, Guayaquil, Cuenca, Ambato y Manta. Extraer datos del INEC aquí es poco probable. La recolección de un gran volumen de datos requeriría scripts que consulten estas APIs de forma periódica.

2. Eventos deportivos a nivel mundial:

- Facilidad: Alta.
- Fuentes: ¡Este es uno de los más sencillos! Existen numerosas APIs gratuitas y de pago (freemium) como TheSportsDB, API-Football, ESPN API. Puedes obtener millones de registros sobre resultados, estadísticas de jugadores, calendarios y más, en formatos como JSON. El web scraping a sitios de resultados también es muy factible.

3. Pulso político en 5 ciudades de Ecuador:

- Facilidad: Difícil.
- Fuentes: Este tema se basa casi exclusivamente en el análisis de sentimientos. La fuente principal sería la API de X (antes Twitter) o el scraping de portales de noticias ecuatorianos (El Universo, Primicias, El Comercio). El reto es filtrar por ciudad y obtener un volumen masivo de datos

relevantes. Es complejo pero ideal para cumplir el requisito de análisis de sentimientos.

4. Actividades y hobbies:

- Facilidad: Alta (si se acota bien).
- Fuentes: Es un tema muy flexible. Puedes enfocarlo en un área con excelentes fuentes de datos, como:
    - Videojuegos: Usar la API de Steam o IGDB.
    - Cine/Series: Usar la API de The Movie Database (TMDb).
    - Libros: Usar la API de Goodreads o Google Books.

5. Conciertos y eventos públicos:

- Facilidad: Moderada.
- Fuentes: Se puede hacer web scraping a sitios como TicketShow o Meetup. A nivel internacional, APIs como las de Ticketmaster o Songkick proveen buena información. La clave es la consistencia en la recolección.

7. Restaurantes y sitios de esparcimiento:

- Facilidad: Moderada a Alta.
- Fuentes: Ideal para el análisis de sentimientos. Se puede usar la API de Google Places (tiene costos) o hacer scraping de sitios como TripAdvisor o guías de restaurantes locales. Se pueden obtener miles de reseñas, calificaciones y detalles de locales.

8. Eventos o noticias mundiales:

- Facilidad: Alta.
- Fuentes: Al igual que los deportes, hay muchas APIs de noticias como NewsAPI.org, The Guardian API o GNews. Permiten buscar por tema, idioma o fecha, y devuelven miles de artículos en formato JSON, perfectos para análisis de tendencias y sentimientos a gran escala.

Recomendación: Para cumplir fácilmente el requisito del millón de registros y la diversidad de fuentes, te sugiero enfocarte en estas 5 temáticas:

1. Eventos deportivos a nivel mundial
2. Actividades y hobbies (ej. enfocado en videojuegos o cine)
3. Eventos o noticias mundiales
4. Restaurantes y sitios de esparcimiento
5. El tema libre que elijas.

Para el tema libre

1. Tendencias del Mercado Laboral en Tecnología

- **Interés analítico:** Analizar la demanda de lenguajes de programación, frameworks, roles (Front-end, Back-end, DevOps, Data Scientist), y la evolución del trabajo remoto vs. presencial en Ecuador y LATAM. Los hallazgos serían de gran utilidad para tu carrera.
- **Fuentes de datos:**
    - Web Scraping: Extraer ofertas de empleo de portales como LinkedIn, Get on Board, Indeed o Zonajobs.
    - APIs: Algunas plataformas de freelance como Upwork tienen APIs.
    - Datasets públicos: El Stack Overflow Developer Survey publica anualmente sus datos crudos, que contienen decenas de miles de respuestas de desarrolladores a nivel mundial.

2. Análisis del Ecosistema de Código Abierto en GitHub

- **Interés analítico:** Descubrir qué lenguajes de programación son tendencia, cuáles son los proyectos más activos, analizar patrones de colaboración, o incluso realizar análisis de sentimientos sobre los comentarios en issues y pull requests.
- Fuentes de datos:
    - **API de GitHub:** Es una de las APIs más completas que existen. Permite obtener información sobre repositorios, usuarios, commits, issues, etc.
    - **GH Archive:** Es un proyecto que registra toda la actividad pública de GitHub y la hace accesible. ¡Aquí tienes terabytes de datos para asegurar el millón de registros!

3. Economía y Reseñas en el Mercado de Videojuegos

- **Interés analítico:** Correlacionar precios, géneros, y calificaciones de la crítica con las reseñas de los usuarios (análisis de sentimientos). Analizar tendencias de mercado por plataforma (PC, PlayStation, Xbox, Nintendo) y el impacto de las ofertas.
- **Fuentes de datos:**
    - API de Steam: Proporciona datos masivos sobre juegos, precios, número de jugadores y reseñas de usuarios.
    - Web Scraping: Sitios como Metacritic para obtener reseñas de críticos y usuarios.
    - Datasets: Kaggle tiene numerosos datasets relacionados con ventas y calificaciones de videojuegos.

Módulo de trabajo

Dado que deben usar dos gestores SQL y dos NoSQL, y que el repositorio final debe ser SQL Server, te sugiero la siguiente combinación que es potente, flexible y muy popular en la industria:

Bases de Datos SQL

1. **Microsoft SQL Server:** (Obligatorio) Este será su repositorio final de datos. Es un sistema muy robusto y completo. Pueden usar la

   Developer Edition, que es gratuita y tiene todas las funcionalidades para instalarla en sus máquinas locales o en un servidor.

2. **SQLite:** Es la elección perfecta como segunda base de datos relacional. Al ser un motor de base de datos ligero y basado en archivos, es ideal para que cada miembro del equipo guarde los resultados iniciales de su *web scraping* en un formato estructurado y local antes de moverlos al sistema central. El propio diagrama de ejemplo del proyecto lo sugiere.

Bases de Datos NoSQL

1. **MongoDB:** Es la opción más lógica y potente para este proyecto. Como base de datos NoSQL orientada a documentos, es perfecta para recibir datos semi-estructurados o en formato JSON directamente de las APIs y el *scraping*. El diagrama de ejemplo también la utiliza extensivamente.
2. **Redis:** Como segunda opción NoSQL, Redis es excelente para mostrar versatilidad. Es una base de datos en memoria de tipo clave-valor, increíblemente rápida. Podrían usarla para tareas específicas como gestionar una cola de URLs para el *scraping*, almacenar en caché datos consultados frecuentemente o para el análisis de sentimientos en tiempo real.

Plataformas de Bases de Datos (SQL y NoSQL)

Dado que deben usar dos gestores SQL y dos NoSQL , y que el repositorio final debe ser SQL Server, te sugiero la siguiente combinación que es potente, flexible y muy popular en la industria:

Bases de Datos SQL

1. **Microsoft SQL Server:** (Obligatorio) Este será su repositorio final de datos. Es un sistema muy robusto y completo. Pueden usar la

   Developer Edition, que es gratuita y tiene todas las funcionalidades para instalarla en sus máquinas locales o en un servidor.

2. **SQLite:** Es la elección perfecta como segunda base de datos relacional. Al ser un motor de base de datos ligero y basado en archivos, es ideal para que cada miembro del equipo guarde los resultados iniciales de su *web scraping* en un formato estructurado y local antes de moverlos al sistema central. El propio diagrama de ejemplo del proyecto lo sugiere.

Bases de Datos NoSQL

1. **MongoDB:** Es la opción más lógica y potente para este proyecto. Como base de datos NoSQL orientada a documentos, es perfecta para recibir datos semi-estructurados o en formato JSON directamente de las APIs y el *scraping*. El diagrama de ejemplo también la utiliza extensivamente.
2. **Redis:** Como segunda opción NoSQL, Redis es excelente para mostrar versatilidad. Es una base de datos en memoria de tipo clave-valor, increíblemente rápida. Podrían usarla para tareas específicas como gestionar una cola de URLs para el *scraping*, almacenar en caché datos consultados frecuentemente o para el análisis de sentimientos en tiempo real.

Estrategia de Trabajo: Local vs. Nube

Para un equipo de 4 personas, la mejor estrategia no es "local o nube", sino una

combinación de ambas (híbrida), lo cual además es un requisito del proyecto.

La recomendación es adoptar un flujo de trabajo centralizado en la nube para facilitar la colaboración.

1. **Desarrollo y Extracción (Local):** Cada uno de los 4 miembros del equipo trabaja en su propia máquina. Escriben sus *scripts* de Python para extraer datos y los guardan en una base de datos local y temporal, como un archivo SQLite o una instancia local de MongoDB. Esto les da agilidad para programar y depurar sin depender de una conexión a internet ni interferir con el trabajo de los demás.
2. **Integración y Limpieza (Nube):** Una vez que los *scripts* funcionan, se ejecutan para poblar las bases de datos centrales en la nube.
    - **MongoDB Atlas:** Usen el nivel gratuito (*free tier*) de MongoDB Atlas como su base de datos NoSQL compartida. Cada miembro envía sus datos extraídos aquí.
    - **Azure SQL Database o SQL Server en una VM:** Para la parte SQL, pueden usar el servicio de base de datos de Azure (la contraparte en la nube de SQL Server) o instalar SQL Server en una máquina virtual de cualquier proveedor de nube (Azure, AWS, Google Cloud). Aquí es donde integrarán los datos provenientes de todas las fuentes.
3. **Repositorio Final (Nube):** El Microsoft SQL Server final, que contendrá el millón de registros limpios y transformados, debe estar en la nube. De esta forma, todos pueden conectarlo con Power BI para crear los

   dashboards y realizar la defensa del proyecto sin problemas.

Este modelo híbrido permite que cada uno aporte su cuota de ~250,000 registros de manera independiente a un repositorio central, cumpliendo todos los requisitos técnicos y facilitando enormemente la colaboración.