Proyecto – *App Airbnb*

Jose Daniel Carrera Bolaños, David Melo Valbuena, Juan Andrés Ruiz, Martín García Chagüezá, Santiago Murgueitio

Universidad Autónoma de Occidente

Cali, Colombia.

Selección del dataset objetivo

Para este análisis, se necesitó un conjunto de datos que incluya información detallada sobre las propiedades listadas en Airbnb en Nueva York. Para ello, se utilizó la plataforma kaggle.com, la cual proporciona herramientas y recursos para la ciencia de datos.

El dataset en cuestión se titula "New York Airbnb Listings" y proporciona información actualizada sobre las actividades de las listas de Airbnb en Nueva York hasta el 5 de enero de 2024. Este conjunto de datos contiene varias características útiles como latitud, longitud, tipo de habitación, precio, calificaciones, número de dormitorios, camas y baños, entre otras.

Este dataset fue modificado a partir del original que puede encontrarse en Inside Airbnb. Las modificaciones incluyen la adición de columnas como calificaciones, dormitorios, camas y baños, además de la eliminación de valores nulos, garantizando así una mayor integridad y utilidad de los datos para el análisis.

El dataset tiene un tamaño de 4.44 MB y se actualiza anualmente, proporcionando una visión periódica y actualizada de las tendencias y características del mercado de alquileres en Nueva York. La licencia del dataset es CC BY-SA 4.0, lo que permite su uso y distribución con atribución y bajo la misma licencia.

Este conjunto de datos es especialmente útil para realizar análisis exploratorios de datos, visualización de datos y clustering, proporcionando una base sólida para entender el mercado de alquileres a corto y largo plazo en Nueva York.

Definición de microservicios:

Microservicio de Usuarios: Este microservicio se encarga de la gestion de la autenticacion,registro y manejo del perfil de los usuarios. Maneja datos como credenciales de acceso, información del perfil e historial de actividad de los usuarios. Interactúa con el microservicio de Reservas para validar las credenciales de los usuarios al crear o consultar reservas.

Microservicio de Reservas: Este microservicio tiene es el encargado de la creación, actualización y consulta de reservas. Maneja detalles de las reservas, como fechas, usuario que reserva, propiedad reservada y estado de la reserva. Requiere información del microservicio Usuario para validar a los usuarios y hacer la reserva para que a su vez no salga error. Además, interactúa con el microservicio de Airbnbs para verificar si está disponible la propiedad. Dicho esto, se puede ver que los 3 microservicios tienen relación entre sí.

Microservicio de Airbnbs: Este microservicio se encarga de la gestion , creacion , actualización y consulta de las propiedades de Airbnb. Maneja

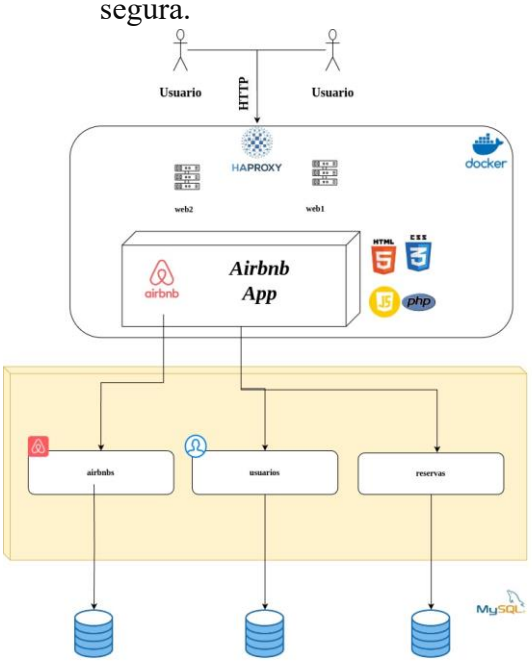
información detallada como ubicacion,tipo de cuarto, número de habitaciones, etc. Depende del Microservicio de Usuarios para validar a los propietarios y al usuario la reserva valida.

Descripción de componentes:

1. Bases de Datos (DBs):
    - Función en la aplicación web: Almacenar y gestionar los datos persistentes del sistema, incluyendo información de usuarios, detalles de reservas y propiedades de Airbnb.
    - Descripción general: MySQL es un sistema de gestión de bases de datos relacional utilizado para almacenar datos estructurados y gestionar la información de manera eficiente.
2. Microservicios:
    - Función en la aplicación web: Gestionar diferentes aspectos del sistema como la autenticación de usuarios, la creación y actualización de reservas, y la gestión de propiedades de Airbnb.
    - Descripción general: Los microservicios son componentes independientes que realizan funciones específicas dentro del sistema, facilitando la escalabilidad y el mantenimiento.
3. App Web:
    - Función en la aplicación web: Proveer una interfaz interactiva y amigable para que los usuarios puedan buscar propiedades, realizar reservas y gestionar sus perfiles.
    - Descripción general: La aplicación web utiliza tecnologías como HTML5, CSS3, JavaScript y PHP para crear una interfaz de usuario dinámica y receptiva.
4. Balanceo de Carga (Balanceo):
    - Función en la aplicación web: Distribuir el tráfico entrante entre múltiples instancias de servidores web para asegurar la disponibilidad y escalabilidad del sistema.
    - Descripción general: HAProxy es una solución de balanceo de carga que distribuye el tráfico de red de manera uniforme entre varios servidores para mejorar la resiliencia y el rendimiento del sistema.
5. Analítica de Datos:
    - Función en la aplicación web: Analizar y visualizar datos para proporcionar insights y facilitar la toma de decisiones estratégicas.
    - Descripción general: Power BI es una herramienta de análisis de datos que permite crear visualizaciones interactivas y reportes a partir de datos procesados.
6. Procesamiento de Datos:
    - Función en la aplicación web: Procesar grandes volúmenes de datos para análisis avanzado, incluyendo operaciones de limpieza, transformación y agregación.
    - Descripción general: PySpark es una herramienta de procesamiento de datos que facilita la manipulación y el análisis de grandes conjuntos de datos.
7. Docker:
    - Función en la aplicación web: Contenerización de aplicaciones para asegurar consistencia en diferentes entornos de desarrollo y producción.
    - Descripción general: Docker permite empaquetar aplicaciones y sus dependencias en contenedores para facilitar el despliegue y la escalabilidad.
8. Docker Swarm:
    - Función en la aplicación web: Orquestación de contenedores

Docker para gestionar el despliegue, escalabilidad y operaciones de los contenedores en un clúster.
- Descripción general: Docker Swarm coordina y gestiona el despliegue de contenedores Docker en diferentes nodos del sistema.

9. Microsoft Azure:
    - Función en la aplicación web: Proveer infraestructura en la nube, incluyendo servicios de almacenamiento, bases de datos y recursos computacionales.
    - Descripción: Microsoft Azure es una plataforma de computación en la nube que ofrece servicios para hospedar aplicaciones y gestionar recursos de forma escalable y segura.



Definición de la arquitectura completa del sistema

La arquitectura completa del sistema inicia con un balanceador de carga utilizando HAProxy para garantizar la disponibilidad de la aplicación y facilitar el manejo del número de solicitudes realizadas. Para ello, se hace la réplica de la aplicación, creando copias idénticas que permiten distribuir la carga de manera equilibrada entre ellas. En caso de que uno de los servidores que aloja la aplicación experimente algún problema o falle, el balanceador de carga redirigirá automáticamente las solicitudes al servidor de respaldo, asegurando la continuidad del servicio sin interrupciones.

La aplicación web utiliza tecnologías como HTML5, CSS3, JavaScript y PHP para crear una interfaz interactiva y amigable, permitiendo a los usuarios buscar propiedades, realizar reservas y gestionar sus perfiles. Los usuarios interactúan con la aplicación web, la cual envía solicitudes a los microservicios a través del balanceador de carga.

El backend de la aplicación está basado en microservicios, cada uno corriendo en puertos diferentes pero gestionados a través de una API Gateway para exponer múltiples servicios a través de un único puerto. Esto facilita la gestión centralizada de las solicitudes de la aplicación.

Los microservicios incluyen:

10. Usuarios: Gestiona la autenticación, registro y manejo del perfil de usuario.
11. Reservas: Gestiona la creación, actualización y consulta de reservas.
12. Airbnbs: Gestiona la creación, actualización y consulta de propiedades de Airbnb.

Estos microservicios interactúan con las bases de datos MySQL para realizar operaciones CRUD (Crear, Leer, Actualizar, Borrar) y devolver datos procesados a la aplicación web.

Para el almacenamiento de datos, se utiliza MySQL, un sistema de gestión de bases de datos relacional que almacena información crítica del sistema como los detalles de los usuarios, las reservas y las propiedades de Airbnb. Este sistema permite gestionar eficientemente la información estructurada y realizar consultas rápidas.

Para el procesamiento de datos, se emplea PySpark, que permite realizar operaciones de limpieza, transformación y agregación de grandes

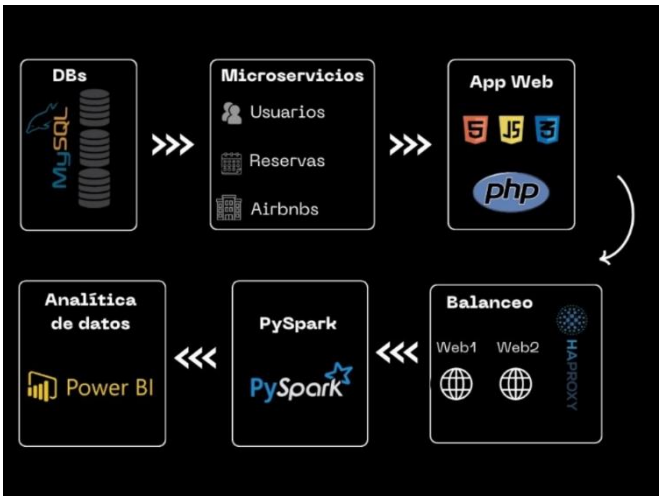
volúmenes de datos. Los datos brutos de los microservicios y las bases de datos son procesados y luego enviados a Power BI para su análisis y visualización. Power BI se utiliza para analizar y visualizar estos datos, proporcionando insights y facilitando la toma de decisiones estratégicas a través de reportes y dashboards interactivos.

Además, la aplicación está contenerizada utilizando Docker, lo que asegura la consistencia en diferentes entornos de desarrollo y producción. Docker facilita el empaquetado de microservicios y otros componentes en contenedores, permitiendo un despliegue y escalabilidad eficientes. Para la orquestación de estos contenedores, se utiliza Docker Swarm, que coordina y gestiona el despliegue de contenedores Docker en diferentes nodos del sistema.

Finalmente, la infraestructura del sistema está alojada en Microsoft Azure, que provee servicios de almacenamiento, bases de datos y recursos computacionales en la nube, asegurando alta disponibilidad y escalabilidad del sistema.

Esta arquitectura propuesta permite una alta escalabilidad y flexibilidad, ya que cada microservicio puede ser desarrollado, implementado y escalado independientemente de los demás. Además, la API REST facilita la integración con otras aplicaciones y sistemas externos, y el análisis distribuido permite procesar grandes volúmenes de datos de manera simultánea y paralela.

"Figura arquitectura completa"



13. Bases de datos (DBs)

Tecnología: MySQL: Almacenar y gestionar los datos persistentes del sistema.

Datos que maneja:

- Usuarios: Información personal (nombre, correo electrónico, contraseña, detalles del perfil).
- Reservas: Detalles de las reservas y realizadas (fechas de inicio y fin, usuario que reserva, propiedad reservada, estado de la reserva).
- Airbnbs: Información sobre las propiedades disponibles para alquiler (descripción, ubicación, precio, disponibilidad).

La base de datos contiene información detallada sobre listados de alojamientos de Airbnb en la ciudad de Nueva York. Cada fila representa un listado diferente y las columnas incluyen detalles como:

- ID del listado y nombre descriptivo
- ID del anfitrión y su nombre
- Ubicación: vecindario, borough, coordenadas de latitud y longitud
- Tipo de propiedad (entire home/apt o private room)
- Precio por noche
- Noches mínimas requeridas
- Cantidad de reseñas recibidas
- Calificación promedio

- Cantidad de dormitorios, camas y baños

Las columnas en esta base de datos se relacionan de la siguiente manera:

- id y name identifican de manera única cada listado de propiedad.
- host_id y host_name permiten rastrear cuáles listados pertenecen a un mismo host.
- neighbourhood, neighbourhood_group, latitude y longitude permiten ubicar geográficamente cada propiedad dentro de la ciudad de Nueva York y su área metropolitana.
- room_type indica si el listado es para un hogar/apartamento entero (Entire home/apt) o una habitación privada dentro de un hogar (Private room).
- price y minimum_nights son dos aspectos clave para evaluar la oferta y tarifas de cada propiedad.
- number_of_reviews y rating brindan una medida de la reputación y satisfacción de los huéspedes con cada listado específico.
- bedrooms, beds y baths proporcionan detalles sobre la capacidad y tipo de alojamiento dentro de cada propiedad listada.

Relación con los demás componentes

- Punto Central de Almacenamiento: Las bases de datos son el núcleo donde se almacena toda la información crítica del sistema, como los detalles de los usuarios, las reservas y las propiedades de Airbnb.
- Relación: Todos los componentes interactúan con las bases de datos directa o indirectamente para obtener, almacenar o modificar datos esenciales para el funcionamiento del sistema.

14. Microservicios

Componentes: Usuarios, Reservas, Airbnbs

Tecnología: se implementan usando Frameworks de node.js

Función:

- Usuarios: Gestionar la autenticación, registro, y manejo del perfil de usuario. El inicio de sesión varia dependiendo de los roles de la siguiente manera.

Host:

- Crear Airbnbs
- Actualizar un Airbnb
- Eliminar un Airbnb
- Ver las reservas

Cliente:

- Reservar Airbnbs
- Ver todos los Airbnbs disponibles.

- Reservas: Gestionar la creación, actualización, y consulta de reservas.
- Airbnbs: Gestionar la creación, actualización, y consulta de propiedades de Airbnb.

Datos que manejan:

- Usuarios: Credenciales de acceso, información del perfil, historial de actividad.
- Reservas: Detalles de la reserva y el estado de la reserva.
- Airbnbs: Detalles de la propiedad.

Coordinación entre Microservicios

Microservicio de Usuarios

El microservicio de Usuarios gestiona la autenticación, registro y manejo del perfil de los usuarios. Necesita interactuar con el microservicio de Reservas para validar las credenciales de los usuarios al crear o consultar reservas.

Microservicio de Reservas

El microservicio de Reservas se encarga de la creación, actualización y consulta de reservas. Requiere información del microservicio de Usuarios para validar a los usuarios y acceder a

sus perfiles, y del microservicio de Airbnbs para actualizar su estado cuando se realicen o cancelen reservas.

Microservicio de Airbnbs

El microservicio de Airbnbs gestiona la creación, actualización y consulta de propiedades de Airbnb. Depende del microservicio de Usuarios para validar a los propietarios y acceder a la información del perfil de los usuarios, y del microservicio de Reservas para las reservas realizadas o canceladas.

Relación con los demás componentes

- Con DBs: Los microservicios se comunican directamente con las bases de datos para realizar operaciones CRUD (Crear, Leer, Actualizar, Borrar).
- Con App Web: Los microservicios proporcionan APIs que la aplicación web consume para realizar operaciones y mostrar datos a los usuarios.
- Con Analítica y Procesamiento de Datos: Los datos operacionales generados por los microservicios son procesados y analizados para obtener insights y generar reportes.

15. App Web

Tecnologías: HTML5, CSS3, PHP: Provee una interfaz interactiva y amigable para que los usuarios puedan interactuar con el sistema.

- Mostrar información relevante (propiedades disponibles, detalles de reservas).
- Facilitar acciones como búsqueda de propiedades, realización de reservas, y gestión del perfil de usuario.

Datos que maneja:

- Entrada: Datos ingresados por los usuarios como por ejemplo en el login (búsquedas, información de reserva, detalles del perfil).
- Salida: muestra “la información que reciben los microservicios de las bases de datos y después las envía dependiendo de las operaciones CRUD que demande el usuario

Relación con los demás componentes

- Cuenta con un balanceo de carga por HAproxy

16. Balanceo de Carga (Balanceo)
Tecnología: HAProxy : Distribuir el tráfico entrante entre múltiples instancias de servidores web para asegurar la disponibilidad del sistema. Mejorar la resiliencia del sistema manejando fallos de servidor de manera efectiva.

Docker Swarm: Docker Swarm facilita la gestión de contenedores en un clúster, permitiendo la implementación, escalado y administración de servicios distribuidos de manera automática.

- Escalabilidad: Docker Swarm permite escalar los servicios hacia arriba o hacia abajo según la demanda. Se pueden agregar o eliminar instancias de microservicios fácilmente para manejar variaciones en la carga.

Datos que maneja:

- Entrada: Solicitudes HTTP de los usuarios.
- Salida: Redirige las solicitudes a las instancias de servidores web disponibles (Web1, Web2).

Relación con los demás componentes:

Con App Web: Asegura que las solicitudes de los usuarios sean distribuidas de manera uniforme entre los servidores web, optimizando el rendimiento y la disponibilidad del sistema.

17. Analitica de datos

Tecnología: Power BI : Analizar y visualizar datos para proporcionar insights y facilitar la toma de decisiones. Además genera reportes y dashboards interactivos.

Datos que maneja:

- Entrada: Datos procesados provenientes de PySpark (análisis de tendencias, comportamientos de usuario, estadísticas de uso).

- Salida: Visualizaciones, reportes, gráficos y dashboards basados en los datos analizados.

Relación con los demás componentes:

- Con PySpark: Recibe datos procesados de un clúster de datos distribuidos con apache spark utilizando PySpark. para generar visualizaciones y reportes.
- Con Microservicios: Utiliza los datos operacionales generados por los microservicios (a través de Operaciones de RDDs con PySpark) para el análisis y la generación de insights.

18. Procesamiento de datos

Tecnología: PySpark : Procesar grandes volúmenes de datos para análisis avanzado. Realiza operaciones de limpieza, transformación y agregación de datos. También crea Resilient Distributed Dataset y RDDs.

Google BigQuery: Almacenar y analizar grandes conjuntos de datos con consultas SQL, permitiendo análisis en tiempo casi real y escalabilidad masiva.

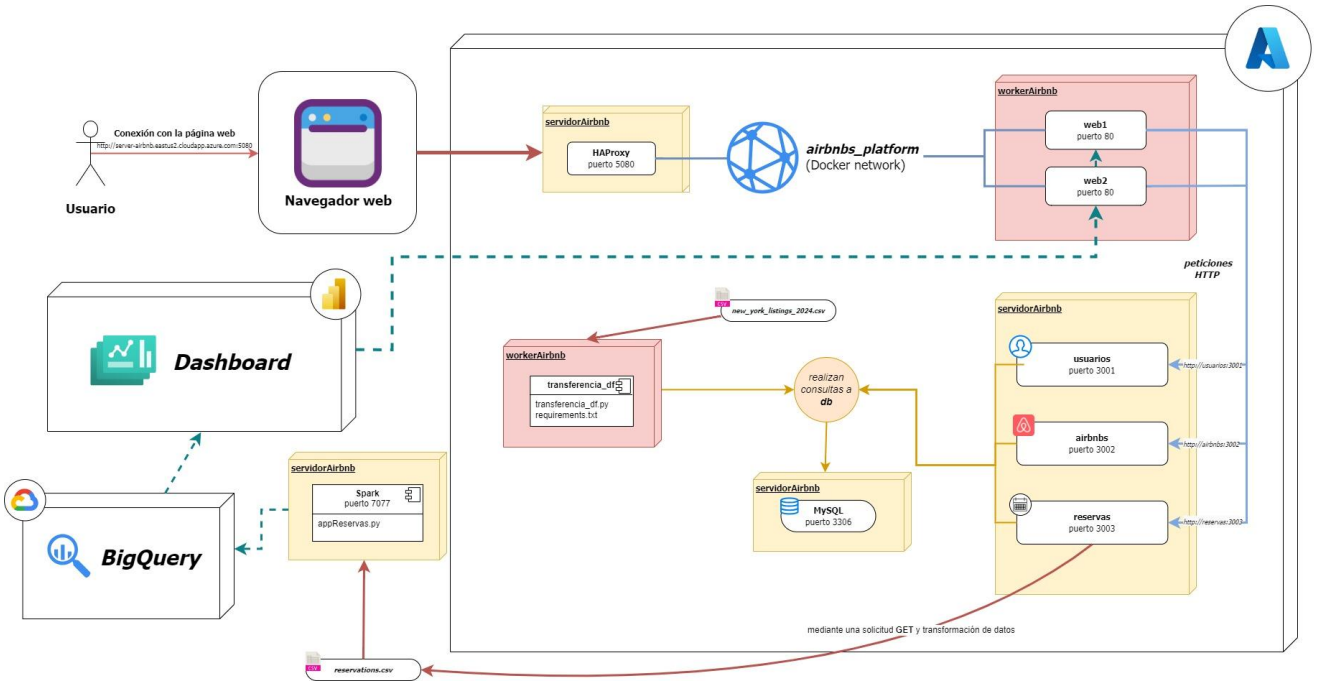
Datos que maneja:

- Entrada: Datos brutos de las diferentes operaciones de microservicios (sobre todo el de reservas). en el sistema).
- Salida: Datos csvs derivados de operaciones de mapReduce con pyspark y mandados a google bigQuery que posteriormente lo manda a Power BIRelación con los demás componentes:
- Con DBs y Microservicios: Datos brutos de las diferentes operaciones de microservicios
- Con Analítica de Datos: Envía datos csv para procesarlos en el powerBI para su visualización y análisis.

Flujo Completo del sistema

1. Usuarios interactúan ya sea como host o cliente con la App Web para realizar diversas acciones (búsqueda de propiedades, reservas, gestión de perfil).
2. Se envían las solicitudes de los clientes a través de la APPWeb y el Haproxy, se asegura de balancear las peticiones que hacen los usuarios en la página.
3. Los Microservicios procesan las solicitudes, interactúan con las Bases de Datos para realizar las operaciones respectivas de cada uno de ellos y devuelven los datos procesados a la App Web
4. Los datos generados por las interacciones del usuario son enviados a un cluster de datos distribuidos para su procesamiento.
5. PySpark realiza operaciones de mapReduce a estos datos y los envía la nube (google BigQuery), que posteriormente hace envío de estos datos al Power BI.
6. Power BI genera visualizaciones y reportes que pueden ser utilizados por los administradores para la toma de decisiones estratégicas.

Diagrama de despliegue



Pruebas de escalabilidad y desempeño

Mediante la herramienta JMeter realizaremos la ejecución de estas pruebas, en donde buscaremos estimar el rendimiento del componente web y los microservicios. Para ello, establecimos los siguientes parámetros con el fin de desarrollar estas pruebas:

Prueba de carga normal: Simulación de carga usando 100 usuarios, en 3 ciclos durante 30 segundo. Mediante esta prueba evaluamos el rendimiento de la aplicación en el contexto de un tráfico estable.

Prueba de carga alta: Aumentamos la carga de la aplicación con el fin de replicar escenarios de mayor presión y demanda de los servicios. Para ello, usamos 500 usuarios que, en 5 loops, realizaron diversas acciones.

Prueba de estrés: Se sometió la aplicación a escenarios extremos en donde evaluamos la capacidad máxima de ella con respecto al manejo del tráfico. Se utilizan 1000 usuarios en 10 loops para realizar dicho testeo.

A analizar

1. Tiempo de respuesta promedio de las acciones realizadas por los usuarios (THROUGHPUT)

2. Porcentaje de error en las acciones realizadas.

Tablas de resultados – Pruebas de carga normal

Componente	Réplicas	Path (GET - HTTP)	Error %	Tiempo de respuesta promedio (THROUGHPUT)
HAProxy	1	N/A	0.00	10.1
Micro Usuarios	1	/usuarios	0.00	3.6

Tablas de resultados – Pruebas de carga alta

Componente	Réplicas	Path (GET - HTTP)	Error %	Tiempo de respuesta promedio (THROUGHPUT)
HAProxy	1	N/A	0.00	4.8
Micro Usuarios	1	/usuarios	0.00	18.0

Tablas de resultados – Pruebas de carga estrés

Componente	Réplicas	Path (GET - HTTP)	Error %	Tiempo de respuesta promedio (THROUGHPUT)
HAProxy	1	N/A	0.00	332.8
Micro Usuarios	1	/usuarios	0.00	85.7

Enlaces de interés

[1] Repositorio de GitHub

[2] Vídeos – Demostración (OneDrive)

Referencias

[1] (2022) kaggle Website.[Online].Disponible en: https://www.kaggle.com/datasets/robikscube/flight-delay-dataset-20182022 ?select=Airlines.csv

[2] Docker Inc. (2024). Docker overview. Docker Documentation. Retrieved from https://docs.docker.com/get-started/overview.

[3] Mozilla Developer Network (MDN). (2024). *JavaScript overview*. MDN Web Docs. Retrieved from https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/Introduction.