



UNIVERSIDAD AUTÓNOMA DE AGUASCALIENTES

Centro de Ciencias Básicas

Materia: Desarrollo y Diseño Web

Profesor(a): M.I.T.C. Ma. Guadalupe Ibarra Nava

Lic. En Informática y Tecnologías Computacionales

**Juan Carlos Galindo Gaeta
Christian Enrique González Márquez
Bryan Issur Macías
Leonardo Alejandro Yáñez Martínez**

Manual técnico

1. Descripción General:

- Introducción a la API:

- Descripción breve del propósito de la API: el propósito del api es recopilar información de personas y algún gusto por un videojuego con el fin de analizar esta información.

- Su funcionalidad principal y su objetivo: la función principal es hacer consultas altas bajas cambios acerca de información como lo es el nombre de la persona El juego favorito el género favorito con el objetivo de poder analizar esta información.

2. Documentación de Endpoints:

- Lista completa de Endpoints:

- Tabla o lista de todos los endpoints disponibles.

METODO HTTP	Para que sirve
GET /	Listar Videojuegos
GET / id	Obtener videojuego por id
POST /	Crear un videojuego
PUT / id	Actualizar videojuego
DELETE / id	Eliminar videojuego

3. Autenticación y Autorización:

- Procesos de Autenticación:

- debido a que el lápiz funciona de manera local esta solamente se autentifica en base a la base de datos de mongodb.

4. Ejemplos y Casos de Uso:

- Ejemplos Prácticos:

- Escenarios de uso comunes con ejemplos de código:

El alta, baja, cambio y consulta son los mas básicos que con el FRONTEND que hicimos debe de ser sencillo y lo muestro a continuación:

Crear Videojuego
Nombre de Usuario:

Juegos Favoritos:

Género Favorito:

Personaje Favorito:

Eliminar Videojuego
ID del Videojuego a Eliminar:

Actualizar Videojuego
ID del Videojuego a Actualizar:

Nuevo Nombre de Usuario:

Nuevos Juegos Favoritos:

Nuevo Género Favorito:

Nuevo Personaje Favorito:

Buscar Videojuego por ID
ID del Videojuego a Buscar:

5. Formato de Datos y Respuestas:

- Formatos de Datos:

- el formato que se utiliza es JSON para hacer peticiones desde POSTMAN.

6. Gestión de Errores y Códigos de Estado HTTP:

- Códigos de Estado HTTP:

200 - OK:

El código de estado "200 OK" indica que la solicitud HTTP fue exitosa.

Significa que el servidor ha procesado la solicitud correctamente y ha devuelto el contenido solicitado.

500 - Internal Server Error:

El código de estado "500 Internal Server Error" indica que ha ocurrido un error interno en el servidor.

Este error es genérico y señala que algo salió mal en el servidor al intentar procesar la solicitud, pero no proporciona detalles específicos sobre el problema.

404 - Not Found:

El código de estado "404 Not Found" indica que el recurso solicitado no pudo ser encontrado en el servidor.

Se utiliza cuando el servidor no puede encontrar el recurso solicitado por el cliente y no está disponible en la ubicación especificada.

201 - Created:

El código de estado "201 Created" indica que la solicitud HTTP ha sido completada y ha resultado en la creación exitosa de un nuevo recurso.

Se utiliza normalmente en respuestas a operaciones de creación, como cuando se agrega un nuevo elemento a una base de datos a través de una solicitud POST.