

- > **Motores de videojuegos II**
- > Juan Diego Alegre Brun
- > Curso 2022/2023

## Programación Gráfica

### MEMORIA

<b>1 ¿Qué es una API?</b>	<b>2</b>
1.1 ¿Cómo utilizar una API?	2
1.2 Ejemplo de API (Jose Maria Maestre)	3
1.3 Ejemplo de API (Hector Ochando)	3
<b>2 ¿Que es un SDK?</b>	<b>4</b>
2.2 ¿Cómo utilizar un SDK?	5
2.3 Ejemplo de SDK (Hector Ochando)	6
2.4 Ejemplo de SDK (Jose Maria Maestre)	6
<b>3. Bibliografía</b>	<b>7</b>

## 1 ¿Qué es una API?

Una API (Application Programming Interface) es un conjunto de protocolos, herramientas y definiciones de comunicación que permiten a diferentes aplicaciones o sistemas interactuar entre sí de manera programática.

En otras palabras, una API es una interfaz de programación que permite a los desarrolladores acceder a ciertas funciones o datos de una aplicación o sistema, sin necesidad de conocer todos los detalles internos de su funcionamiento. Las APIs suelen ser utilizadas para integrar servicios o aplicaciones de terceros en una plataforma existente, para automatizar tareas o para crear nuevas herramientas.

Las APIs pueden ser de diferentes tipos, como REST, SOAP, GraphQL, entre otros, y pueden utilizar diferentes protocolos de comunicación, como HTTP, TCP/IP, UDP, entre otros.

### 1.1 ¿Cómo utilizar una API?

Para utilizar una API, generalmente se necesitan seguir los siguientes pasos:

1. Obtener una clave de acceso (API key): muchas APIs requieren que los desarrolladores obtengan una clave de acceso para poder utilizar sus servicios. Esta clave es única para cada desarrollador y se utiliza para autenticar las solicitudes a la API.
2. Leer la documentación de la API: es importante leer la documentación de la API para conocer los detalles de su funcionamiento, incluyendo los endpoints disponibles, los parámetros que acepta cada solicitud, los formatos de respuesta y cualquier otra información relevante.
3. Hacer una solicitud a la API: una vez que se tiene una clave de acceso y se ha leído la documentación, se puede empezar a utilizar la API haciendo solicitudes a sus endpoints. Esto generalmente se hace a través de llamadas HTTP utilizando un lenguaje de programación, como Python, JavaScript, Ruby, etc.
4. Procesar la respuesta: una vez que se ha hecho la solicitud a la API, se recibirá una respuesta en el formato especificado en la documentación, generalmente en formato JSON. Esta respuesta generalmente incluirá los datos solicitados por el desarrollador, que pueden ser procesados y utilizados en la aplicación o sistema en el que se está trabajando.

## 1.2 Ejemplo de API (Jose Maria Maestre)

Para este ejemplo, utilizaré la API REST [movieDB](#).

Buscaremos una película con el título “Jack Reacher” con esta petición https

```
https://api.themoviedb.org/3/search/movie?api_key={api_key}&query=Jack+Reacher
```

En la parte {api\_key} deberemos sustituirlo por la “key” proporcionada por la organización al registrarnos.

Esta petición nos devolverá toda la información relevante sobre la película “Jack Reacher” en formato JSON

```
{
  "poster_path": "/IfB9hy4JH1eH6HEfIgIGORXi5h.jpg",
  "adult": false,
  "overview": "Jack Reacher must uncover the truth behind a major government conspiracy in order to clear his name.",
  "release_date": "2016-10-19",
  "genre_ids": [
    53,
    28,
    80,
    18,
    9648
  ],
  "id": 343611,
  "original_title": "Jack Reacher: Never Go Back",
  "original_language": "en",
  "title": "Jack Reacher: Never Go Back",
  "backdrop_path": "/4ynQYtSEuU5hyipcGkfD6ncwtwz.jpg",
  "popularity": 26.818468,
  "vote_count": 201,
  "video": false,
  "vote_average": 4.19
}
```

También podemos buscar películas sobre un género, en la que aparezcan determinados actores, dirigidas por algún director en concreto, etc.

### 1.3 Ejemplo de API (Hector Ochando)

La API (Application Programming Interface) de Unity es un conjunto de funciones y procedimientos que los desarrolladores pueden utilizar para interactuar con el motor de juego Unity y crear sus propias aplicaciones y juegos.

La API de Unity está diseñada para ser utilizada con el lenguaje de programación C#, aunque también se puede utilizar con otros lenguajes de programación, como JavaScript o Boo.

La API de Unity ofrece una amplia variedad de funciones y procedimientos que permiten a los desarrolladores realizar tareas comunes en los juegos y aplicaciones. Estas funciones y procedimientos cubren áreas como la creación de objetos en la escena, la gestión del tiempo, la entrada del usuario, la física, la animación, el sonido, la red y la inteligencia artificial.

Además, Unity también proporciona una serie de componentes prefabricados que se pueden utilizar en las aplicaciones. Estos componentes prefabricados son objetos preconstruidos que incluyen scripts y propiedades configurables que permiten a los desarrolladores crear rápidamente elementos comunes en los juegos, como personajes, cámaras, efectos visuales y otros elementos.

La API de Unity también incluye una serie de herramientas y recursos adicionales que permiten a los desarrolladores crear juegos y aplicaciones de alta calidad. Por ejemplo, Unity incluye un sistema de física avanzado que permite simular objetos y colisiones de forma realista, y un sistema de animación que permite crear animaciones complejas y realistas.

En resumen, la API de Unity es un conjunto de funciones y procedimientos que permiten a los desarrolladores interactuar con el motor de juego Unity y crear sus propias aplicaciones y juegos. La API cubre una amplia variedad de áreas y ofrece una serie de herramientas y recursos adicionales que permiten a los desarrolladores crear juegos y aplicaciones de alta calidad.

## 2 ¿Que es un SDK?

Un SDK (Software Development Kit) es un conjunto de herramientas de software y recursos que los desarrolladores utilizan para crear aplicaciones para una plataforma específica. Los SDKs son un recurso muy valioso para los desarrolladores, ya que permiten crear aplicaciones con mayor rapidez y eficiencia al proporcionar herramientas y recursos para simplificar y acelerar el proceso de desarrollo.

Un SDK típico puede incluir una variedad de recursos y herramientas, como bibliotecas de programación, herramientas de desarrollo, documentación, ejemplos de código, emuladores, simuladores y otros recursos.

La biblioteca de programación proporcionada por el SDK contiene una serie de funciones y procedimientos que los desarrolladores pueden utilizar para interactuar con el sistema o la plataforma para la que se está desarrollando la aplicación.

Las herramientas de desarrollo incluidas en un SDK pueden ayudar a los desarrolladores a crear, depurar y probar aplicaciones de manera más eficiente. Estas herramientas incluyen editores de código, depuradores, compiladores, emuladores, simuladores y otros recursos que facilitan el trabajo de los desarrolladores.

La documentación proporcionada por el SDK es muy importante, ya que ayuda a los desarrolladores a comprender cómo utilizar las herramientas, bibliotecas y recursos proporcionados por el SDK. Esta documentación incluye manuales de usuario, guías de programación, tutoriales y otros recursos que explican cómo utilizar las herramientas y recursos del SDK.

Los ejemplos de código proporcionados por el SDK son muy valiosos para los desarrolladores, ya que les permiten ver cómo se pueden utilizar las funciones y procedimientos proporcionados por la biblioteca de programación. Estos ejemplos de código suelen ser programas simples que demuestran cómo utilizar diferentes funciones y procedimientos del SDK.

En general, un SDK es una herramienta esencial para los desarrolladores, ya que les permite crear aplicaciones de manera más rápida y eficiente al proporcionar herramientas y recursos útiles. Al utilizar un SDK, los desarrolladores pueden ahorrar tiempo y esfuerzo al tener acceso a bibliotecas de programación, herramientas de desarrollo, documentación, ejemplos de código y otros recursos que les permiten centrarse en la creación de aplicaciones en lugar de tener que escribir todo el código desde cero.

## 2.2 ¿Cómo utilizar un SDK?

El proceso para utilizar un SDK puede variar dependiendo de la plataforma o tecnología para la que se está desarrollando una aplicación, pero aquí te presento los pasos generales que se pueden seguir:

**Descargar e instalar el SDK:** lo primero que debes hacer es descargar e instalar el SDK correspondiente para la plataforma o tecnología en la que deseas desarrollar. En la página web del proveedor del SDK se suelen encontrar instrucciones detalladas sobre cómo instalarlo y configurarlo.

**Configurar el entorno de desarrollo:** una vez instalado el SDK, debes configurar el entorno de desarrollo que vayas a utilizar para crear tus aplicaciones. El SDK suele incluir herramientas de desarrollo y compiladores que se pueden utilizar en diferentes entornos, como Visual Studio o Eclipse.

**Estudiar la documentación:** es importante estudiar la documentación del SDK para comprender cómo utilizar las herramientas y recursos que ofrece. La documentación suele incluir manuales de usuario, guías de programación, tutoriales y otros recursos que explican cómo utilizar el SDK.

**Utilizar la biblioteca de programación:** la biblioteca de programación del SDK es una colección de funciones y procedimientos que puedes utilizar para interactuar con la plataforma o tecnología para la que estás desarrollando la aplicación. Es importante estudiar la documentación del SDK para comprender cómo utilizar estas funciones y procedimientos.

**Crear código de ejemplo:** para aprender cómo utilizar el SDK y la biblioteca de programación, es recomendable crear pequeños programas de ejemplo que utilicen las funciones y procedimientos del SDK. Los ejemplos de código proporcionados por el SDK también pueden servir de ayuda.

**Compilar y probar la aplicación:** una vez que hayas creado tu aplicación utilizando el SDK, debes compilarla y probarla en la plataforma o tecnología correspondiente. Si el SDK incluye herramientas de depuración, utilízalas para identificar y solucionar cualquier error que pueda surgir.

En resumen, utilizar un SDK implica descargar e instalar el SDK, configurar el entorno de desarrollo, estudiar la documentación, utilizar la biblioteca de programación, crear código de ejemplo, compilar y probar la aplicación. Al seguir estos pasos, los desarrolladores pueden utilizar el SDK para crear aplicaciones de manera más rápida y eficiente.

### 2.3 Ejemplo de SDK (Hector Ochando)

Unreal es un SDK (Software Development Kit) que incluye un motor de juego, la API de Unreal, herramientas de desarrollo, bibliotecas de programación y otros componentes para la creación de aplicaciones y juegos.

El motor de juego Unreal Engine es la parte central del SDK de Unreal. Proporciona un conjunto de herramientas y características para el desarrollo de juegos y aplicaciones, como gráficos de alta calidad, física avanzada, animaciones, inteligencia artificial, sonido, redes y más. Los desarrolladores pueden utilizar el motor de juego Unreal Engine para crear proyectos de juego y aplicaciones para múltiples plataformas, como PC, consolas y dispositivos móviles.

La API de Unreal es un conjunto de funciones y procedimientos que permiten a los desarrolladores interactuar con el motor de juego Unreal Engine. La API cubre una amplia variedad de áreas y ofrece una serie de herramientas y recursos adicionales que permiten a los desarrolladores crear juegos y aplicaciones de alta calidad.

Además, el SDK de Unreal también incluye herramientas de desarrollo como el editor de Unreal, que es una herramienta visual para la creación y edición de contenido de juego y aplicaciones, y herramientas de compilación y depuración para la creación y prueba de proyectos.

En resumen, Unreal es un SDK completo que incluye el motor de juego Unreal Engine, la API de Unreal, herramientas de desarrollo, bibliotecas de programación y otros componentes para la creación de aplicaciones y juegos de alta calidad. Los desarrolladores pueden utilizar el SDK de Unreal para crear proyectos de juego y aplicaciones para múltiples plataformas y aprovechar las herramientas y recursos que ofrece el SDK.

### 2.4 Ejemplo de SDK (Jose Maria Maestre)

[Twilio](#) es un SDK que te permite crear una aplicación de mensajería y de llamadas utilizando su API interna.

Twilio ofrece una gran variedad de lenguajes de programación para trabajar, como JavaScript, Python, Ruby, PHP, entre otros, lo que hace que sea fácil de usar independientemente del lenguaje de programación que prefiera el desarrollador.

Además, el SDK de Twilio ofrece una documentación muy completa, que incluye ejemplos de código para cada uno de los lenguajes de programación compatibles. Esto permite a los desarrolladores empezar a trabajar con la API de Twilio de manera rápida y sin tener que pasar mucho tiempo investigando.

Por ejemplo, en JavaScript, para enviar un mensaje de texto con Twilio se podría utilizar el siguiente código:



```
const accountSid = 'tu_account_sid';
const authToken = 'tu_auth_token';
const client = require('twilio')(accountSid, authToken);

client.messages
  .create({
    body: 'Hola, este es un mensaje de prueba',
    from: '+1415XXXXXX', // tu número de Twilio
    to: '+52155XXXXXXX' // número al que quieres enviar el mensaje
  })
  .then(message => console.log(message.sid));
```

El SDK de Twilio permite integrar su API en diferentes lenguajes de programación.

### 3. Bibliografía

<https://openai.com/blog/chatgpt>

<https://www.redhat.com/es/topics/cloud-native-apps/what-is-SDK>

<https://www.themoviedb.org/documentation/api>