



Introducción

A continuación se detalla la práctica final, fin de formación, de la III Edición del Swift Developer Program, desarrollada en el año 2023. La creación de una app que consuma una API REST con más de 64.000 mangas publicados, donde el usuario podrá gestionar su colección guardando qué mangas tiene, por qué tomo lleva la colección e incluso por qué tomo de los que tiene lleva la lectura.

Niveles de desarrollo

La app se plantea como un desafío a varios niveles, donde hay un mínimo a entregar (producto mínimo viable) y a partir de ahí, se puede implementar más funcionalidad en la app sobre las versiones propuestas.

Información general de los datos

Los mangas pueden tener distintos temas (*themes*) que definen la temática del mismo, como tipo *School*, *Parody*, *Mecha* (robots gigantes), *Vampires*, *Music* y muchos más.

También tienen géneros (*genres*) como *Action*, *Adventure*, *Sci-Fi*, *Romance*, *Comedy* y más.

De igual forma se clasifican por demografías en cuanto al público objetivo: *Shounen* (chicos jóvenes), *Shoujo* (chicas jóvenes), *Seinen* (adultos), *Kids* (niños) y *Josei* (mujeres adultas).

También hay autores, donde pueden aparecer asociados a su rol: si solo han escrito, solo han dibujado o han escrito y dibujado. Los mangas pueden tener 1 o n autores cada uno, por lo que siempre vendrán en una

colección asociada. También sucede así con los géneros, demografías y temáticas, que pueden ser varias por cada manga.

Cualquier manga consultado vendrá con toda la información en estructuras de datos anidadas, con uno o varios autores, demografías, temáticas y géneros. También incluye URLs de consulta y una URL para una portada.

La estructura podrá ir en una colección o dentro de una subestructura como colección también cuando se devuelvan paginados:

```
{  
    "titleJapanese": "ドラゴンボール",  
    "authors": [  
        {  
            "id": "998C1B16-E3DB-47D1-8157-8389B5345D03",  
            "firstName": "Akira",  
            "lastName": "Toriyama",  
            "role": "Story & Art"  
        }  
    ],  
    "themes": [  
        {  
            "id": "ADC7CBC8-36B9-4E52-924A-4272B7B2CB2C",  
            "theme": "Martial Arts"  
        },  
        {  
            "id": "472FB2AE-13C0-4EEE-9A45-A7B75AC5DC29",  
            "theme": "Super Power"  
        }  
    ],  
    "title": "Dragon Ball",  
    "id": 42,  
    "endDate": "1995-05-23T00:00:00Z",  
    "score": 8.41,  
    "status": "finished",  
    "demographics": [  
        {  
            "demographic": "Shounen",  
            "id": "5E05BBF1-A72E-4231-9487-71CFE508F9F9"  
        }  
    ],  
    "genres": [  
        {  
            "genre": "Action",  
            "id": "72C8E862-334F-4F00-B8EC-E1E4125BB7CD"  
        },  
        {  
            "genre": "Adventure",  
            "id": "BE70E289-D414-46A9-8F15-928EAFBC5A32"  
        },  
        {  
            "genre": "Comedy",  
            "id": "F974BCB6-B002-44A6-A224-90D1E50595A2"  
        },  
        {
```

```

        "genre": "Sci-Fi",
        "id": "2DEDC015-82DA-4EF4-B983-F0F58C8F689E"
    }
],
"startDate": "1984-11-20T00:00:00Z",
"titleEnglish": "Dragon Ball",
"chapters": 520,
"mainPicture": "\https://cdn.myanimelist.net/images/manga/1/2677931.jpg\"",
"synopsis": "Bulma, a headstrong 16-year-old girl, is on a quest to find the mythical Dragon Balls—seven scattered magic orbs that grant the finder a single wish. She has but one desire in mind: a perfect ...",
"background": "Dragon Ball has become one of the most successful manga series of all time, with over 230 million copies sold worldwide with 157 million in Japan alone...",
"url": "\https://myanimelist.net/manga/42/Dragon_Ball\",
"volumes": 42
}

```

Información de la API

Dada la alta cantidad de mangas en la base de datos, los *endpoints* generales devolverán los datos por paginación:

/list/mangas Devolverá 10 mangas por página, enviando la página 1.
/list/mangas?page=2&per=20 Devolverá la página 2, de 20 en 20.
/list/mangas?page=1&per=50 Devolverá 50 mangas para la página 1.

Hay que ser coherentes con la consulta para solicitar los datos que necesitamos y que tengan coherencia a lo que hemos solicitado, de forma que si pedimos la página 2 o 3, el parámetro **per** sea el mismo para todas las solicitudes para no devolver datos duplicados.

El JSON vendrá acompañado siempre, en estas consultas, de un árbol extra llamado **metadata** cuya estructura nos informará del total de datos de la consulta, la página devuelta y cuántos datos ha devuelto para esta.

```

"metadata": {
    "total": 64833,
    "page": 1,
    "per": 10
}

```

Por defecto, todas las consultas paginadas devuelven los datos con un parámetro **per** que es igual a 10.

Listados en endpoint (**/list**)

/mangas – Devuelve todos los mangas de la base de datos
/bestMangas – Mangas ordenados inversamente por puntuación
/authors – Todos los autores de mangas en la base de datos (no paginada)
/demographics – Array de cadenas con todas las demografías
/genres – Array de cadenas con todos los géneros
/themes – Array de cadenas con todas las temáticas

```
/mangaByGenre – Devuelve todos los mangas de un género (solo uno).  
    /mangaByGenre/romance  
/mangaByDemographic – Devuelve todos los mangas de una demografía  
    /mangaByDemographic/shoujo  
/mangaByTheme” – Devuelve todos los mangas de una temática.  
    /mangaByDemographic/school  
/mangaByAuthor” – Devuelve los mangas de un autor (por su ID).  
    /mangaByAuthor/998C1B16-E3DB-47D1-8157-8389B5345D03
```

Búsquedas en endpoint (`/search`)

```
/mangasBeginsWith – Devuelve los mangas cuyo título empieza por...  
    /mangasBeginsWith/dragon  
/mangasContains – Devuelve los mangas que contienen en el título...  
    /mangasContains/ball  
/author – Devuelve los autores que su primer nombre o último nombre...  
    /author/toriya  
  
/manga – Devuelve el manga que corresponde con un ID exacto  
    /manga/42  
/manga – Método POST que recibe el siguiente JSON:
```

```
struct CustomSearch: Codable {  
    var searchTitle: String?  
    var searchAuthorFirstName: String?  
    var searchAuthorLastName: String?  
    var searchGenres: [String]?  
    var searchThemes: [String]?  
    var searchDemographics: [String]?  
    var searchContains: Bool  
}
```

Se le puede pasar título, primer nombre de autor, último nombre de autor, colección de géneros (como cadenas), de temáticas y de demográficos. El valor Bool establece cuando es false la búsqueda de valores que empiezan por título y autor, y con true que incluyan la cadena.

Es una búsqueda multipropósito por todos los datos posibles y que devuelve por paginación los resultados.

Gestión de usuarios

/users – Método POST que recibe el siguiente JSON:

```
struct Users: Codable {  
    var email: String  
    var password: String  
}
```

El email debe tener un formato válido y el password debe tener al menos 8 caracteres de longitud. La llamada a este **endpoint** está supeditada a incluir un parámetro en la cabecera que identifique a la app pues es un **endpoint** abierto a cualquier solicitud en la red.

Dentro de los *headers* de la solicitud HTTP debe haber un parámetro:
“App-Token”: “sLGH38NhEJ0_anlIWwhsz1-LarCLEohiAHQqayF0FY”

Este **endpoint** devuelve un status 201: Created que establecerá que el usuario se ha creado correctamente. A partir de ahí, se podrá hacer login con el siguiente **endpoint**.

/users/login – Método POST que recibe el user/pass y devuelve un TOKEN.

La entrada a este **endpoint** está supeditada a un protocolo de autenticación básica con los datos de usuario y contraseña en base64, en el campo **Authorization** en las cabeceras de la solicitud HTTP.

```
let credentials = "\$(username):\$(password)"
if let encodedCredentials = credentials.data(using: .utf8) {
    let auth = "Basic \$(encodedCredentials.base64EncodedString())"
}
```

Cuando el usuario y clave son correctas devuelve un **TOKEN** que tendrá dos días de validez. A los dos días, este dejará de ser válido y habrá que volver a pedir al usuario de la app sus credenciales para obtener un nuevo **TOKEN**.

/users/renew – Método POST que recibe el token válido actual y lo renueva. Lo debe recibir en el campo **Authorization** en las cabeceras de la solicitud HTTP en la forma “Bearer \\$(token)”.

Si el **TOKEN** enviado es válido, se generará uno nuevo y se invalidará el anterior (un buen método para el inicio de la app cuando el usuario ya está logueado dentro del sistema).

Gestión de usuarios / mangas

ATENCIÓN: Todos estos métodos requieren obligatoriamente de la presencia del **TOKEN** para identificar al usuario.

/collection/manga – Método POST que crea o actualiza la inclusión de un manga en la colección de un usuario.

Recibe el siguiente JSON:

```
struct UserMangaCollectionRequest: Codable {
    var manga: Int
    var completeCollection: Bool
    var volumesOwned: [Int]
    var readingVolume: Int?
}
```

El campo manga es el código del manga a dar de alta o actualizar, completeCollection indica si el usuario tiene toda la colección del manga con todos sus volúmenes, en volumesOwned tenemos un array con el total de volúmenes que tiene el usuario de dicho manga y en readingVolume indicamos el volumen por el que va leyendo el usuario.

Si enviamos un manga que no existe lo da alta con sus datos asociados. Si enviamos un manga que ya existe, lo actualiza con los nuevos datos asociados al registro.

/collection/manga – Método GET que recupera toda la colección de mangas de un usuario con todos los datos de un usuario. Ojo: no es paginado.

/collection/manga/<mangaID> – Método GET que recupera un manga concreto de un usuario de su colección, que coincide con el ID del manga.
/collection/manga/42

Si no existe el manga en la colección devolverá un error.

/collection/manga/<mangaID> – Método DELETE para eliminar un manga de la colección de un usuario.
/collection/manga/42

Si no existe el manga en la colección devolverá un error. Si se borra devolverá un estado 200 OK.



Versión básica

En la versión de entrega obligada, el alumno deberá usar los **endpoints** que corresponden a los listados y búsquedas en la forma que considere oportuno, así como la persistencia en local de los datos de las colecciones de los usuarios, para ofrecer la siguiente funcionalidad:

- Consulta de cualquier referencia bibliografía de manga.
- Inclusión de al menos una categorización en los listados o filtros.
- Que el usuario de la app pueda guardar el manga en su colección.
- Mostrar la colección del usuario.

La app deberá tener un *layout* funcional para iPhone y iPad y deberá incluir siempre la imagen de la portada del manga que está en una URL.

Los datos que deberán guardarse sobre la colección del usuario son:

- Número de tomos comprados.
- Tomo por el que va leyendo.
- Si tiene o no la colección completa.

Versión media

Deberá de realizarse todo lo que incluye la versión básica, pero deberá de ofrecerse un conjunto de filtros completo para la app basado en todas las categorías de datos por los que se puede clasificar la información. La app deberá incluir, como mínimo: un listado, un detalle y un grid.

Versión avanzada

Deberá incluirse toda la gestión en la nube de la colección del usuario, así como el flujo de creación y login de usuarios. Podrá persistirse la información en local, igualmente, si se desea. Pero la información deberá estar de manera fija en la nube, en la API, para ese usuario. El token y/o credenciales de usuario deben ir almacenadas en la cartera de certificados del dispositivo.

Versión deluxe

Incluir versiones para otros dispositivos de Apple: al menos uno más.