

# Sobre los Challenges

- ❖ Qué es un Challenge
- ❖ Preguntas Frecuentes

## Preguntas Frecuentes

**¿Qué son los Challenges ONE?**

Es una forma de implementar el Challenge Based Learning o sea, aprendizaje basado en desafíos que Apple ayudó a crear. Es un mecanismo donde podrás comprometerte en la resolución de un problema para después investigar soluciones con cursos, contenido y charlas; ¡O incluso con el conocimiento que ya tienes! Finalmente vas a actuar y colocar tu proyecto en el aire. Todo esto mientras comentas y ayudas en proyectos de otros alumnos y alumnas.

**¿Qué necesito saber, cuales son los lenguajes y cuál es el nivel de conocimiento?**

Necesitas haber realizado cuatro o más cursos de la formación de Back-End para conseguir realizar los desafíos. Iniciaremos con un desafío básico pasando por un desafío intermedio hasta llegar a un desafío avanzado.

### **¿Si tengo dudas con quien puedo hablar?**

En nuestro grupo oficial de Discord tenemos canales específicos para los Challenges. Allí podrás consultar tus dudas con los instructores e instructoras y con el resto de participantes. También tendremos Lives de mentoría "Alura te responde" durante los desafíos para consultar las principales dudas.

### **¿Obtendré un certificado?**

Una vez realizada la entrega de tu proyecto, podrás descargar tu certificado y la insignia referente al mismo, que deberás compartir en tus redes profesionales como LinkedIn etiquetando a Oracle y Alura Latam. Recuerda que los Challenges son parte integral del programa ONE y su objetivo es crear proyectos increíbles e incluirlos en tu portafolio para demostrar tus habilidades.

### **¿Qué materiales o herramientas necesito instalar para realizar el curso?**

Todas las etapas de los Challenges tendrán especificaciones diferentes que iremos avisando conforme los desafíos sean liberados. Te guiaremos durante todo el proceso para que puedas conseguir finalizar tu proyecto sin preocupaciones.

## ¿Cómo sabré sobre los nuevos retos?

Enviaremos e-mails de recordatorio cada vez que un nuevo desafío o Live esté próximo a iniciar. Es importante revisar con frecuencia las bandejas de Spam, Promociones y Social en tu e-mail. Algunas veces los e-mails pueden ir allá.

## ❖ Importancia del Desafío

La creación de un portafolio de proyectos es un paso esencial para consolidar los conocimientos adquiridos durante tus estudios. Aquí te presentamos algunas razones fundamentales para participar en este desafío:

### 1. **Aplicación Práctica del Conocimiento:**

Este challenge te brinda la oportunidad de aplicar en la práctica lo aprendido en los cursos. Al trabajar en proyectos reales, podrás consolidar tus habilidades y demostrar tu capacidad para llevar a cabo tareas concretas.

### 2. **Diferenciación en el Mercado Laboral:**

En un mercado laboral cada vez más competitivo, un portafolio bien elaborado es tu carta de presentación. Muestra a los reclutadores no solo tus habilidades teóricas, sino también tu capacidad para resolver problemas y abordar desafíos del mundo real.

### 3. **Reflejo de tu Creatividad y Construcción de una Marca Personal:**

Esta es tu oportunidad para construir tu marca personal. Refleja quién eres como profesional, tus valores y tus logros. Una marca personal sólida puede influir positivamente en las decisiones de contratación.

Es crucial resaltar la importancia de contar con un portafolio de proyectos, ya que este constituye tu tarjeta de presentación. En él, puedes destacar tus habilidades, proyectos previos y demostrar tu capacidad. Un portafolio bien elaborado puede marcar una significativa diferencia al buscar oportunidades laborales.

Tips:

- ¡Sé creativo! Hay muchas formas de llegar a la solución del desafío.
- La organización es clave: aprovecha herramientas como Trello para apoyarte en esta habilidad.
- Profundiza en los temas abordados en el desafío y, siempre que lo consideres necesario, regresa a los conceptos iniciales para asegurarte de que la teoría y la práctica estén en perfecta sintonía.
- Crea tu repositorio en GitHub y comparte tus proyectos y avances con la comunidad.
- No olvides crear un README impresionante para tu proyecto en GitHub, ya que esto ayuda a documentar y presentar tus soluciones de manera clara y atractiva."
- No olvides poner un enlace o video de tu proyecto en LinkedIn! Etiqueta a #Oracle y #AluraLatam

❖ [Oracle](#)

❖ [Alura Latam](#)

## ❖ Pre-requisitos

El objetivo principal de este challenge es que vivas la experiencia de realizar un proyecto real, tal y como ocurre en el día a día de una persona desarrolladora; tendrás la oportunidad de practicar los conceptos aprendidos en los cursos realizados hasta ahora.

Este desafío está diseñado pensando en la importancia de aplicar de manera práctica los aprendizajes obtenidos en los siguientes cursos:

- [Java: trabajando con lambdas, streams y Spring Framework](#)
- [Java: persistencia de datos y consultas con Spring Data JPA](#)

# Construye el Literalura, tu catálogo de libros

## ❖ ¡Acepta el Desafío!

¡Hola! ¡Claro que sí! Aquí tienes un resumen estructurado de la clase, con conceptos clave y puntos importantes para que puedas estudiar de manera eficiente. Resumen Estructurado de la Clase: "Construye el Literalura, tu catálogo de libros" Objetivo del Desafío: Crear una aplicación de consola en Java con Spring y Postgres que permita a los usuarios registrar libros en una base de datos y obtener información sobre ellos.

Tecnologías Clave:

Java

Spring Boot

Spring Data JPA

PostgreSQL

Requisitos Básicos de la Aplicación:

Buscar libro por título:

El usuario ingresa el nombre del libro.

La aplicación busca la información del libro en la API Gutendex y lo registra en la base de datos.

Se muestra el título, autor (apellido, nombre), idioma y número de descargas.

Listar libros registrados:

Muestra todos los libros almacenados en la base de datos.

Listar autores registrados:

Muestra todos los autores almacenados en la base de datos.

Listar autores vivos en un año determinado:

El usuario ingresa un año y se muestran los autores que estaban vivos en ese año.

Listar libros por idioma:

El usuario ingresa un código de idioma (ES, EN, FR, PT) y se muestran los libros en ese idioma.

Manejo de Errores:

Mostrar un mensaje si el libro no se encuentra en la API.

Evitar insertar el mismo libro más de una vez en la base de datos.

Pasos para Crear el Proyecto:

Spring Initializer (start.spring.io):

Proyecto Maven con Java.

Versión de Spring Boot: la más reciente sin "snapshots" o "M3".

Dependencias:

Spring Data JPA

PostgreSQL Driver

API Gutendex (gutendex.com):

Documentación en inglés (traducible con Google Chrome).

Biblioteca digital con más de 70,000 libros gratuitos.

Recursos Adicionales:

Tablero de Trello:

Tareas divididas en tarjetas.

Descripción del desafío y contenido adicional.

Detalle de cada tarea (desde crear el proyecto en GitHub hasta funcionalidades adicionales).

Servidor de Discord:

Resolver dudas, ayudar a compañeros y compartir resultados.

Ejemplo Detallado: Buscar Libro por Título

Entrada del Usuario:

El usuario ingresa el título del libro que desea buscar, por ejemplo, "Pride".

Búsqueda en la API Gutendex:

La aplicación utiliza la API para buscar el libro "Pride".

Respuesta de la API:

La API devuelve la información del libro "Pride and Prejudice" de Jane Austen, con idioma EN (inglés) y 6493 descargas.

Registro en la Base de Datos:

La aplicación guarda esta información en la base de datos.

Salida al Usuario:

La aplicación muestra los detalles del libro:

Título: Pride and Prejudice

Autor: Austen, Jane

Idioma: EN

Descargas: 6493

Consideraciones Adicionales:

Creatividad: Implementar funcionalidades adicionales.

Documentación: Consultar la documentación de la API para entender cómo obtener y trabajar con los datos.

Espero que este resumen te sea de gran ayuda. ¡Mucho éxito con tu estudio y en el desarrollo del LiterAlura!

## ❖ Trello del desafío

Haz clic [AQUÍ](#) para acceder al tablero del desafío

<https://trello.com/b/WDyMPDMb/literalura-challenge-java>

---

Trello es una herramienta de colaboración o gestión de proyectos que contribuye al éxito empresarial y a la organización personal. Te permite organizar tus tareas en tableros, además de poder crear un tablero y asignarlo a cada miembro de tu equipo.

Para ser más productivo, estar en contacto con una herramienta de gestión de proyectos, controlar tu tiempo y dividir tus tareas en actividades más pequeñas, te

sugerimos utilizar Trello como ayuda.



## Sobre el Desafío

### Descripción

En este emocionante desafío de programación, te invitamos a construir tu propio catálogo de libros: el LiterAlura. Aprenderás a realizar solicitudes a una API de libros, a manipular datos JSON, guardarlos en una base de datos y, finalmente, a filtrar y mostrar los libros y autores de interés.

**OBJETIVO:** Desarrollar un Catálogo de Libros que ofrezca interacción textual (vía consola) con los usuarios, proporcionando al menos 5 opciones de interacción. Los libros se buscarán a través de una API específica. La información sobre la API y las opciones de interacción con el usuario se detallará en la columna "Backlog"/"Listo para iniciar".

Los pasos para completar este desafío se detallarán a continuación y estarán disponibles en la sección adyacente:

Configuración del Ambiente Java;

Creación del Proyecto;

Consumo de la API;

Análisis de la Respuesta JSON;

Insertión y consulta en la base de datos;

Exhibición de resultados a los usuarios;

¡Prepárate para sumergirte en una experiencia práctica y emocionante en el desarrollo Java!



Vamos a trabajar con el sistema ágil de desarrollo, utilizando el Trello de la siguiente forma:

La columna Listos para iniciar presenta las tarjetas con elementos que aun no fueron desarrollados.

En la columna En Desarrollo estarán los elementos que estés desarrollando en el momento. Al iniciar una tarea, podrás mover la tarjeta que contiene dicha tarea para esta columna.

En la columna Pausado estarán los elementos que comenzaste a desarrollar, pero necesitaste parar por algún motivo.

Por fin, en la columna Concluido estarán los elementos ya concluidos.

El Trello es una herramienta de uso individual para que puedas controlar el progreso de tus actividades, pero no será evaluada.

¡Buen proyecto!

Para saber más: cursos, artículos y videos

#### **Descripción**

Separamos algunos materiales de Alura (como cursos, artículos y videos) que pueden ayudarte cuando tengas alguna duda teórica.

## Cursos (cursos latam aún en gravación)

- [Curso Online Java: trabajando con lambdas, streams y Spring Framework | Alura](#)
- [Curso Online Java: persistencia de datos y consultas con Spring Data JPA | Alura](#)

## Artículos

- [¿Qué es Maven? | Alura Cursos Online](#)
- [Spring: Conozca este framework de Java | Alura Cursos Online](#)
- [Empezando con Spring Framework | Alura Cursos Online](#)
- [¿Que es Json? | Alura Cursos Online](#)
- [IntelliJ IDEA para principiantes | Alura Cursos Online](#)
- [Evite NullPointerException en Java | Alura Cursos Online](#)

- [Entendiendo el Lazy y el Eager Load de JPA | Alura Cursos Online](#)
- [Bases de datos relacionales | Alura Cursos Online](#)

## Alura +

- ❖ [¿Qué es la Programación Orientada a Objetos?](#)
- ❖ [Camino hacia el éxito como desarrollador Java #Aluratips](#)
- ❖ [Spring Framework. ¿Qué es ? #AluraMás](#)

Enlace para el curso del desafío en la plataforma

### Descripción

<https://app.aluracursos.com/course/challenge-spring-boot-literalura>

Listo para iniciar

Crear el repositorio de tu proyecto en GitHub

Git y GitHub son herramientas muy útiles para cualquier desarrollador, por lo que deberías acostumbrarte a trabajar con ellas.

En este desafío queremos proponerte que el primer paso a realizar sea la creación de este repositorio en GitHub, aunque aún no hayas desarrollado ningún código, lo importante es que tengas al menos una carpeta específica para tu proyecto. y puedes actualizarlo gradualmente. El siguiente material te ayudará con esta actividad:

- [Git y GitHub: Herramientas Esenciales para el Control de Versiones y Colaboración en Desarrollo](#)
- [Git - Configurando Git por primera vez](#)
- [como subir mi proyecto en GitHub](#)

Vale la pena destacar que en esta oportunidad no podremos ver nuestro proyecto funcionando con GitHub Pages, sin embargo, te traemos un desafío diferente, usemos nuestro [README.md](#) para dejar bien detalladas las funcionalidades de nuestro sistema, capturas de pantalla e inclusive un video de tu aplicación funcionando de esta manera aún conseguirás mostrar tu proyecto a la comunidad 😊

Además, tenemos algunos **artículos** que pueden ayudarte a crear y manejar un repositorio en github:

- ❖ [Git y Github: que son y primeros pasos | Alura Cursos Online](#)
- ❖ [Comenzando con Git: aprendiendo a versionar | Alura Cursos Online](#)
- ❖ [Iniciando un repositorio con Git | Alura Cursos Online](#)
- ❖ [Creando un repositorio remoto en GitHub | Alura Cursos Online](#)

## 1) Configurando el entorno Java y Spring

### Descripción

En esta primera fase, nos sumergiremos en la configuración del entorno de desarrollo Java para nuestro desafío de construcción del **LiterAlura** en un proyecto Spring. Asegúrate de contar con los siguientes programas, archivos y versiones:

- Java JDK: *versión: 17 en adelante* - [Download the Latest Java LTS Free](#)
- Maven: *versión 4 en adelante*
- Spring: *versión 3.2.3* - <https://start.spring.io/>
- Postgres: *versión 16 en adelante* - [PostgreSQL: Downloads](#)
- IDE (Entorno de desenvolvimiento integrado) IntelliJ IDEA- *opcional* - [Descargar IntelliJ IDEA](#)

Configuración al crear el proyecto en [Spring Initializr](#):

- Java (versión 17 en adelante)
- Maven (Initializr utiliza la versión 4)
- Spring Boot (versión 3.2.3)
- Proyecto en JAR

Dependencias para agregar al crear el proyecto en [Spring Initializr](#):

- ❖ Spring Data JPA
- ❖ Postgres Driver

## 2) Conociendo la API para traer datos

### Descripción

La API [Gutendex](#) es un catálogo de información de más de 70.000 libros presentes en [Project Gutenberg](#) (biblioteca en línea y gratuita).

En este paso crucial, es fundamental comprender mejor la API de los libros, revisar su documentación y aprender cómo realizar las consultas en la API. En este desafío no es necesario obtener una clave de acceso, solo realizar consultas como se describe en el sitio web oficial.

Asegúrate de seguir cuidadosamente las instrucciones proporcionadas por la API Gutendex.

- Enlace de API: [Gutendex](#)
- Repositorio de API\*: [GitHub - garethbjohnson/gutendex: Web API for Project Gutenberg ebook metadata](#)

\* Dejamos el código oficial (el repositorio) de la API si tienes curiosidad sobre cómo se hizo.

### 3) Construyendo una solicitud de API

#### Descripción

- **Construyendo el Cliente para Solicitudes (HttpClient)**

En la tercera fase de nuestro desafío, empleamos la clase HttpClient para realizar solicitudes a la API de libros y obtener datos esenciales. El uso de HttpClient en Java facilita la conexión y la obtención de respuestas de manera eficiente. Además, proporciona una base sólida para realizar operaciones HTTP de manera más estructurada y versátil. Explorar y entender el manejo de esta clase en Java optimiza el proceso de solicitud.

- **Construyendo la Solicitud (HttpRequest)**

Además, nos sumergimos en el uso de la clase HttpRequest para configurar y personalizar nuestras solicitudes a la API de libros. La clase HttpRequest en Java nos brinda un control detallado sobre los parámetros de nuestras solicitudes, lo que resulta esencial para adaptar la consulta a nuestras necesidades específicas.

Aprender a utilizar la clase HttpRequest no solo es crucial para el éxito de nuestro proyecto, sino que también proporciona una comprensión más profunda de cómo interactuar eficientemente con APIs en Java.

- **Construyendo la Respuesta (HttpResponse)**

Finalmente, nos enfocamos en el uso de la interfaz HttpResponse para gestionar las respuestas recibidas de la API. La interfaz HttpResponse en Java ofrece una estructura que permite analizar y acceder a los diferentes elementos de una

respuesta HTTP. Al entender cómo trabajar con esta interfaz, podrás extraer información significativa de las respuestas, como códigos de estado, encabezados y el cuerpo de la respuesta, que normalmente se presenta en formato JSON.

#### 4) Analizando la respuesta en formato JSON

##### Descripción

En la cuarta fase de nuestro desafío, nos sumergimos en el análisis de la respuesta **JSON** utilizando la biblioteca Jackson en Java. La manipulación de datos JSON es esencial, ya que la mayoría de las respuestas de las API se presentan en este formato.

→ Para facilitar el análisis de los datos que se obtendrán de la API, recomendamos el uso del [sitio de API](#) para realizar consulta de libros o autores.

Con la biblioteca Jackson, puedes realizar el mapeo eficiente de los datos JSON a objetos Java, facilitando así la extracción y manipulación de la información necesaria.

→ **No olvides agregar la biblioteca Jackson al proyecto** como dependencia del archivo POM.xml - sugerimos usar la versión 2.16.

Recuerda utilizar las clases proporcionadas por Jackson, como **ObjectMapper**, para acceder a las distintas propiedades de la respuesta JSON.

#### 5) Conviertiendo los datos

##### Descripción

En esta etapa, llevaremos a cabo las conversiones con los datos de libros y autores, ahora que contamos con la información en nuestro poder. 🙌

Experimenta utilizando clases java para recibir los datos obtenidos mediante la API, transformar los atributos del cuerpo JSON a una clase Java y mostrar los resultados.

Es fundamental crear métodos específicos para manejar estos datos, lo que hará que el código sea más modular y fácil de entender, como `getters`, `setters` and `toString()`.

→ No olvides utilizar las anotaciones `@JsonIgnoreProperties` y `@JsonAlias` para obtener los atributos deseados del cuerpo de respuesta json.

## 6) Interactuando con el usuario

### Descripción

En esta etapa del desafío, nos adentraremos en la interacción con el usuario. El método `Main` debe implementar la interfaz `CommandLineRunner` y su método `run()` donde deberás llamar un método para exhibir el menu. En este método, debes crear un **bucle** para presentar a tu usuario las opciones de insercion y consulta. El usuario deberá seleccionar un número que corresponderá a la opcion numérica y proporcionar los datos que la aplicación recibirá, utilizando la clase **Scanner** para capturar la entrada do usuáio.

Recuerda exhibirle al usuario las informaciones que han sido demandadas y informarle cuando hay un error. Además, es fundamental realizar **pruebas** exhaustivas para garantizar el correcto funcionamiento del programa, simulando diversas situaciones y recorridos para identificar y corregir posibles errores.

## 7) Consultar libros

### Descripción

Bien, ahora vamos a empezar a construir nuestro catálogo de libros y autores. En esta etapa realizamos consulta de libro en la API, con la ayuda de la *URL base* "`https://gutendex.com/books/`" (sugerimos ver la [documentación oficial](#) para diferentes ejemplos de consulta).

En este caso, realizamos la **consulta por título del libro** en la API para retener el primer resultado obtenido. Un libro debe tener los siguientes atributos:

- Título;
- Autor;
- Idiomas;
- Número de Descargas.

Con esta funcionalidad lista, será posible presentar en la consola un **listado de todos los libros** que ya fueron buscados.

Además, también debes posibilitar al usuario ver un **listado con base en el idioma** que uno o más libros fueron escritos, con la ayuda de las *derived queries*.

→ En este desafío vamos a considerar que un libro posee solo un idioma, para que las consultas hechas con tal objeto sean más comprensibles y sencillas. En otras palabras, nos quedaremos solo con el primer resultado de idioma de la lista de idiomas recibida.

En **resumen** tenemos estas **dos funcionalidades obligatorias** en el proyecto:

- ❖ **Búsqueda de libro por título**
- ❖ **Lista de todos los libros**

## 8) Consultar autores

### Descripción

Como podemos ver en el sitio web de la API, cada libro tiene datos relacionados con sus autores, en este caso el cuerpo de json recibe una lista de autores por libro, donde cada autor tiene tres características:

- Nombre;
- Año de nacimiento;
- Año de fallecimiento.

→ En este desafío vamos a considerar que un libro posee solo **un autor**, para que las consultas hechas con tal objeto sean más comprensibles y sencillas. En otras

palabras, nos quedaremos solo con el primer resultado de autor de la lista de autores recibida.

Al guardar los datos de los autores, tendremos la opción de **listado de los autores** de los libros buscados.

Además, pensando en los años de nacimiento y fallecimiento, es posible incluso realizar una **consulta de autores vivos en un determinado año**. ¿Qué te parece eso?

En **resumen** tenemos estas **dos funcionalidades obligatorias** relacionadas a los autores:

- ❖ **Lista de autores**
- ❖ **Listar autores vivos en determinado año**

## 9) Persistencia de datos

### Descripción

Ha llegado el momento de enfocarnos en construir una base de datos, con tablas y atributos relacionados a nuestros objetos de interés: **Libro y Autor**.

En este desafío vamos a utilizar la base de datos llamada [PostgreSQL](#), una de las bases de datos *open source* más utilizadas en el mercado.

Sugerimos la creación de clases de **entidad/modelo** para Libro y Autor, así como también sus respectivas interfaces de **repositorio** para manejar inserción y consultas en la base de datos.

→ No olvides usar las anotaciones correctas y de importar `JpaRepository`, porque estamos trabajando con un proyecto [Spring](#) con **Spring Data JPA**, así que puede manejar las funciones necesarias para nuestro desafío java con persistencia de datos.



Al crear los repositorios de libros y autores, recuerda realizar la conversión de los atributos del libro presentes en el resultado json para un objeto java correspondiente al libro, así quedará más fácil manejar los datos obtenidos en tu proyecto.

**[IMPORTANTE]** Al insertar un libro en la base también deberás insertar su autor y así mantener una **relación entre los dos objetos** vía atributo de identificación (o como lo llamamos, el famoso **ID**).

#### 10) Listando libros por idiomas

##### Descripción

Una vez que ya tienes libros y autores guardados en tu base de datos, ¿qué tal exhibir estadísticas sobre ellos a tu usuario? Aprovecha los recursos de **Streams** de Java y **derived queries** para brindar tu usuario con estadísticas sobre la cantidad de libros en un determinado idioma en la base de datos.

**No es necesario** crear opciones para todos los idiomas. Elija como mínimo dos idiomas.

En resumen tenemos esta **funcionalidad obligatoria** en el proyecto:

- ❖ Exhibir cantidad de libros en un determinado idioma

#### 11) Listando autores vivos en determinado año

##### Descripción

Ahora que ya has avanzado en el uso de la base de datos, te invitamos a cambiar el método para listar los autores vivos en determinado año. Para eso, debes utilizar las **derived queries** para recuperar todos los autores que estaban vivos en el año que el usuario te informará.

→ Recuerda hacer **pruebas** para recibir adecuadamente los datos, porque esto te ayudará a aprender a lidiar con los **valores inválidos** que el usuario pueda ingresar en el sistema.

Haz un README

### Descripción

Uno de los pasos más importantes al participar en una selección de trabajo es resolver un desafío propuesto por la empresa con la información de la resolución, y generalmente esto debe estar en el README. ¿Y qué es el **README**? Es un archivo con extensión **.md** y es un documento con la descripción del proyecto.

Ahora que estamos en la recta final del programa ONE, comencemos a desarrollar archivos README para nuestros últimos desafíos.

Aquí hay un artículo con los pasos para crear un README asombroso:

❖ [Cómo escribir un README increíble en tu Github | Alura Cursos Online](#)

Terminar el Curso

### Descripción

Ahora que tu proyecto ya está listo, recuerda **enviar el enlace** del repositorio en GitHub al curso del challenge y descargar tu certificado.

<https://app.aluracursos.com/course/challenge-spring-boot-literalura/task/87631>

Extra (Opcional)

¡Excelente!  ¡Has finalizado tu proyecto!

En caso de que quieras desafiarte aún más y proporcionar a los usuarios una experiencia más rica y personalizada, hay diversas funcionalidades interesantes que puedes explorar:

- **Generando estadísticas:** Iniciamos las sugerencias de funcionalidades opcionales recordando la clase `DoubleSummaryStatistics`, utilizada para obtener datos estadísticos de un objeto Java. Es posible obtener dichos datos ya sea de consultas de la API o base de datos.

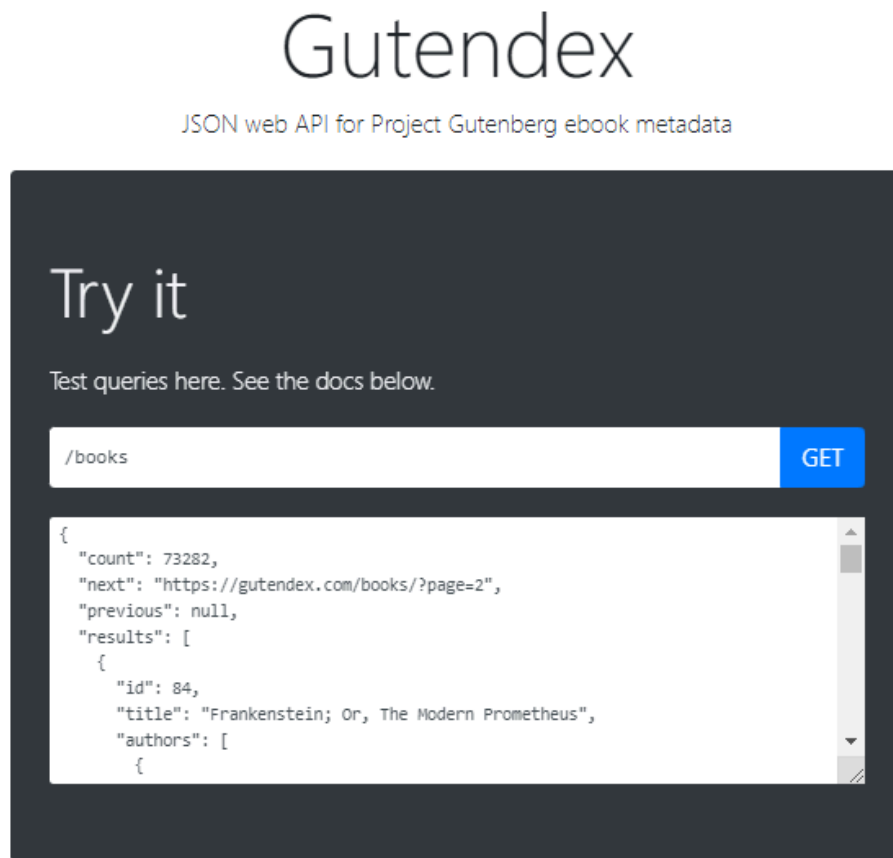
- **Top 10 libros más descargados:** Así como se presentó en el curso *Java: trabajando con lambdas, streams y Spring Framework*, es posible presentar los datos de los 10 libros más descargados, siendo una consulta directamente hecha en la API o en la base de datos.
- **Buscar autor por nombre:** Si has echado un vistazo al sitio de la API es posible realizar la búsqueda de libro o autor con la consulta hecha con *search?* - sin embargo, en este caso te desafiamos a realizar la consulta por nombre de autor en la base de datos creada para nuestro proyecto.
- **Listar autores con otras consultas:** Aquí dejamos como sugerencia implementar otras consultas con los atributos de año de nacimiento y fallecimiento de los autores. Siéntete libre de explorar e implementar estas características adicionales.

¡Desafíate a implementar estas características y convierte tu proyecto en una herramienta aún más poderosa y versátil!

## ❖ Comenzando con Gutendex

Para nuestro desafío actual, hemos elegido la API "[Gutendex](#)", la cual conserva datos de más de 70 mil libros de la biblioteca en línea y gratuita conocida como Proyecto Gutenberg. Esta API no solo proporciona el título y los autores, sino también información como idiomas, formato del texto y cantidad de descargas. ¡Y lo mejor de todo es que **es gratuita, fácil de usar y no requiere una clave de acceso!**

Esta opción nos asegura un acceso confiable y eficiente a los datos esenciales para el proyecto.



## ❖ Para saber Más

Separamos algunos materiales que pueden ayudarte cuando tengas alguna duda teórica =]

# Documentación Oficial:

- [Class HttpClient](#)
- [Class HttpRequest](#)
- [Interfaz HttpResponse](#)
- [Class Scanner](#)

## Artículos:

- [¿Qué es Maven?](#)
- [Spring: Conozca este framework de Java](#)
- [Empezando con Spring Framework](#)
- [¿Que es Json?](#)
- [IntelliJ IDEA para principiantes](#)
- [Evite NullPointerException en Java](#)
- [Entendiendo el Lazy y el Eager Load de JPA](#)
- 

## Entrega del Challenge

❖ Sube tu proyecto al GitHub

Git y GitHub son herramientas muy útiles para cualquier desarrollador, por lo que deberías acostumbrarte a trabajar con ellas.

En este desafío debes crear un repositorio para tu proyecto en GitHub. El siguiente contenido te ayudará con esta actividad:

- [Git - Configurando Git por primera vez](#)
- [Cómo subir mi proyecto en GitHub](#)
- [Cómo escribir un README increíble en tu Github](#)

En esta ocasión, te traemos un reto adicional: usemos nuestro README.md para dejar bien detalladas las funcionalidades de nuestro sistema, con capturas de pantalla si lo deseas, de esta manera ¡conseguirás mostrar tu proyecto a la comunidad!

## ❖ Entrega del desafío

Felicidades por completar tu challenge, ¡estamos seguros que lograste desarrollar un proyecto increíble!

## ¡IMPORTANTE!

- Verifica la url de tu proyecto antes de enviar.
- El sistema **SOLO** ACEPTA url de github.
- **Una vez colocada la url del challenge, debes descargar la insignia y solo después enviar tu proyecto.**
- Tienes 5 tentativas para entregar tu proyecto.

Una vez descargada tu insignia comparte en LinkedIn y en todas tus redes etiquetando #AluraLatam y #oraclenexteducation!

No olvides de darle un vistazo a este video que hemos preparado para ti con el paso a paso descrito anteriormente