

# Laboratorio 2: Programación Orientada a Objetos

Paradigmas 2023 - FAMAF

**En este laboratorio vamos a implementar un lector automático de feeds, como aplicación de consola, bajo el paradigma orientado a objetos, utilizando como lenguaje de programación Java.**

## Instalación

Instalar un java development kit (JDK): `sudo apt install openjdk-8-jdk`.

Instalar algún IDE de su preferencia: eclipse, netbeans, etc.

Clonar de Bitbucket el skeleton provisto por la cátedra: `grupo00_lab02_2023`

Push del skeleton al repositorio del grupo: `grupo??_lab02_2023`

## Funcionalidad

El usuario de nuestro lector establece, mediante un archivo en formato .json, los distintos sitios (ej: new york times, etc) y sus respectivos tópicos (sports, business, etc) de los cuales obtener los diferentes feeds a mostrar por pantalla en forma legible y amigable para el usuario. Además, se agrega una funcionalidad a nuestro lector para computar heurísticamente las entidades nombradas más mencionadas en la lista de feeds.

## Pipeline de Trabajo Sugerido (y aproximado)

1. **Obtener las url's concretos de los distintos feeds del archivo de suscripción** (o sea, obtener qué feeds tenemos que consumir). Para ello van a tener que parsear un archivo en formato json, utilizando alguna librería ya disponible para ello: <https://www.w3docs.com/snippets/java/how-to-parse-json-in-java.html>

```

1 {
2   {
3     "url": "https://www.chicagotribune.com/arcio/rss/category/business/?query=display_date:[now-2d+T0+now]&sort=display_date:desc",
4     "urlParams": [],
5     "urlType": "rss"
6   },
7   {
8     "url": "https://rss.nytimes.com/services/xml/rss/nyt/%s.xml",
9     "urlParams": ["Business", "Technology"],
10    "urlType": "rss"
11  },
12  {
13    "url": "https://www.reddit.com/r/%s/hot/.json?count=100",
14    "urlParams": ["Marketing", "Sales", "Entrepreneur", "Startups"],
15    "urlType": "reddit"
16  }
17 }

```

2. Luego realizar la consulta “http request” al servidor del feed. El servidor en caso de éxito, devolverá un feed RSS, que es simplemente una interfaz (en formato xml) para obtener el contenido de los diferentes artículos del feed. Por ejemplo, si queremos obtener el feed de “business” del “sitio New Yor Times”, en el navegador ponemos la siguiente url:

<https://rss.nytimes.com/services/xml/rss/nyt/Business.xml>

```

<?xml version="1.0" encoding="UTF-8" ?>
<rss xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:media="http://search.yahoo.com/mrss/" xmlns:atom="http://www.w3.org/2005/Atom" xmlns:nyt="http://www.nytimes.com/namespaces/rss/2.0" version="2.0">
  <channel>
    <title>NYT > Business</title>
    <link>https://www.nytimes.com/section/business</link>
    <atom:link href="https://rss.nytimes.com/services/xml/rss/nyt/Business.xml" rel="self" type="application/rss+xml"/>
    <description>
    </description>
    <language>en-us</language>
    <copyright>Copyright 2023 The New York Times Company</copyright>
    <lastBuildDate>Mon, 13 Mar 2023 17:10:40 +0000</lastBuildDate>
    <pubDate>Mon, 13 Mar 2023 17:08:47 +0000</pubDate>
  </channel>
  <image>
    <title>NYT > Business</title>
    <url>https://static01.nyt.com/images/misc/nyt_logo_250x40.png</url>
    <link>https://www.nytimes.com/section/business</link>
  </image>
  <item>
    <title>SVB Collapse Upsets Expectations for Federal Reserve's Rate Decision</title>
    <link>https://www.nytimes.com/2023/03/13/business/economy/federal-reserve-interest-rates.html</link>
    <guid isPermaLink="true">https://www.nytimes.com/2023/03/13/business/economy/federal-reserve-interest-rates.html</guid>
    <atom:link href="https://www.nytimes.com/2023/03/13/business/economy/federal-reserve-interest-rates.html" rel="standout"/>
    <description>The Federal Reserve has been rapidly raising interest rates to fight inflation. But making big moves could be trickier amid instability.</description>
    <dc:creator>Jeanna Smialek</dc:creator>
    <pubDate>Mon, 13 Mar 2023 14:16:05 +0000</pubDate>
    <category domain="http://www.nytimes.com/namespaces/keywords/des">Banking and Financial Institutions</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/des">Regulation and Deregulation of Industry</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/nyt_org">Federal Deposit Insurance Corp</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/nyt_org">Federal Reserve System</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/nyt_org">Treasury Department</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/nyt_per">Powell, Jerome H</category>
    <media:content height="151" medium="image" url="https://static01.nyt.com/images/2023/03/13/multimedia/13dc-fed-pfjh/13dc-fed-pfjh-moth.jpg" width="151"/>
    <media:credit>Julia Hinkinson for The New York Times</media:credit>
    <media:description>Jerome Powell, Chairman of the Federal Reserve, testifies before the House Committee on Financial Services, last week.</media:description>
  </item>
  <item>
    <title>Live Updates: Biden Offers Assurances on the Safety of the Banking System</title>
    <link>https://www.nytimes.com/live/2023/03/13/business/silicon-valley-bank</link>
    <guid isPermaLink="true">https://www.nytimes.com/live/2023/03/13/business/silicon-valley-bank</guid>
    <atom:link href="https://www.nytimes.com/live/2023/03/13/business/silicon-valley-bank" rel="standout"/>
    <description>Markets wobbled as regulators tried to contain the damage from Silicon Valley Bank's collapse, closing another bank – Signature Bank in New York – and opening an emergency lending program. Regional bank stocks tumbled in early trading.</description>
    <dc:creator>The New York Times</dc:creator>
    <pubDate>Mon, 13 Mar 2023 17:08:47 +0000</pubDate>
    <media:content height="151" medium="image" url="https://static01.nyt.com/images/2023/03/13/multimedia/13svb-promo-930/13svb-promo-930-moth.jpg" width="151"/>
    <media:credit>Doug Mills/The New York Times</media:credit>
  </item>
  <item>
    <title>Colleges Have Been a Small-Town Lifeline. What Happens as They Shrink?</title>
    <link>https://www.nytimes.com/2023/03/13/business/economy/college-towns-economy.html</link>
    <guid isPermaLink="true">https://www.nytimes.com/2023/03/13/business/economy/college-towns-economy.html</guid>
    <atom:link href="https://www.nytimes.com/2023/03/13/business/economy/college-towns-economy.html" rel="standout"/>
    <description>Declining student enrollment is hitting the rural areas that rely on universities. They're trying to adapt to survive.</description>
    <dc:creator>Lydia DePillis</dc:creator>
    <pubDate>Mon, 13 Mar 2023 09:00:20 +0000</pubDate>
    <category domain="http://www.nytimes.com/namespaces/keywords/nyt_geo">Pennsylvania</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/des">Colleges and Universities</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/des">Rural Areas</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/des">Labor and Jobs</category>
    <category domain="http://www.nytimes.com/namespaces/keywords/des">United States Economy</category>
  </item>
</rss>

```

3. Una vez conseguidos la lista de artículos de un feed, van a extraer sólo aquellos atributos del artículo (item) que nos interesa (title, description, pubDate, link ) que luego serán mostrados por pantalla en forma legible para el usuario. Para ello van a tener que parsear formato xml con alguna libreria para ello:

[https://www.tutorialspoint.com/java\\_xml/java\\_dom\\_parse\\_document.htm](https://www.tutorialspoint.com/java_xml/java_dom_parse_document.htm)

Notar que deben parsear archivos json y xml, por lo tanto, deben tratar de **generalizar la noción de parser en la aplicación** (hint: pensar en herencia).

4. Por último, el lector tendrá una funcionalidad extra (via parámetro por consola, por ejemplo, `FeedReader -ne`), que computa mediante una heurística, un conjunto de **entidades nombradas** (*named entities*) de los feeds y su cantidad de ocurrencias, global, por clase y por subclase. La cátedra ofrecerá dos heurísticas para ello, **"QuickHeuristic"** y **"RandomHeuristic,"** muy simples, para detectar entidades nombradas. Ustedes deben programar la aplicación, de tal manera, que cambiar de una a otra, no implique un gran cambio en el código de la aplicación. (hint: pensar en herencia).

#### ATENCIÓN

Las entidades nombradas están organizadas en una **jerarquía de clases**, como sigue:

entidad nombrada → persona → apellido  
→ nombre  
→ título  
→ lugar → país  
→ ciudad  
→ dirección  
→ otro  
→ organización  
→ producto  
→ evento  
→ fecha  
→ otro

los atributos específicos **mínimos** (pueden añadir otros que los ayuden en el conteo) para cada una de estas clases son:

- entidad nombrada: string, nombre canónico
- persona: ID de la persona
- apellido: forma canónica, origen
- nombre: forma canónica, origen, formas alternativas

- título: forma canónica, profesional
- país: población, lengua oficial
- ciudad: país, capital, población
- dirección: ciudad
- organización: forma canónica, número de miembros, tipo de organización
- producto: comercial, productor
- evento: forma canónica, fecha, recurrente
- fecha: precisa, forma canónica
- otro: comentarios

Adicionalmente, existe otra jerarquía que también se aplica a las entidades nombradas, de forma ortogonal a la primera, es decir, conceptualmente como si fuera herencia múltiple:

```

tema → deportes → fútbol
                        → básquet
                        → tenis
                        → fórmula 1
                        → otros
→ cultura → cine
                        → música
                        → otros
→ política → nacional
                        → internacional
                        → otros
→ otros

```

## Recomendaciones

- Leer un tutorial o libro de Java antes de comenzar.
- Identificar en su aplicación cada mini-funcionalidad y mapearla a una clase que la abstraiga y la encapsule. “Mientras más clases, mejor”. No importa si una clase tiene un solo atributo y/o método, ya que, ésta inicialmente podría ser simple pero en el futuro podría crecer producto de nuevas funcionalidades.

- Testear cada mini-funcionalidad o clase por separada y a medida que las van implementado, es decir, construir una solución entera, conlleva construir pequeñas soluciones paso a paso y no pretender que todo funcione correctamente al último sin haber testeado cada parte individual de la aplicación.
- Identificar qué mini-funcionalidades son del mismo tipo o naturaleza para poder generalizarlas en una superclase utilizando herencia. En una superclase se modela aquellos atributos y métodos comunes a todas las subclases, quedando en cada subclase exclusivamente aquellos atributos y métodos propios o particulares de una subclase.
- No dar por sentado que el código del skeleton provisto por la cátedra está libre de bugs. Además pueden modificarlo libremente si ustedes lo consideran necesario.

### Servidores de Feeds

A continuación presentamos una tabla de distintos feeds para utilizar durante el desarrollo de la aplicación:

Template	Parámetro	URL final
https://rss.nytimes.com/services/xml/rss/nyt/%s.xml ( <a href="#">Más info</a> )	Business	https://rss.nytimes.com/services/xml/rss/nyt/Business.xml
	Technology	https://rss.nytimes.com/services/xml/rss/nyt/Technology.xml
https://www.chicagotribune.com/arcio/rss/category/%s/?query=display_date:[now-2d+TO+now]&sort=display_date:desc	sports	https://www.chicagotribune.com/arcio/rss/category/sports/?query=display_date:[now-2d+TO+now]&sort=display_date:desc
	business	https://www.chicagotribune.com/arcio/rss/category/business/?query=display_date:[now-2d+TO+now]&sort=display_date:desc

<a href="https://www.reddit.com/r/%s/hot/.json?count=10">https://www.reddit.com/r/%s/hot/.json?count=10</a>	Android	<a href="https://www.reddit.com/r/Android/hot/.json?count=10">https://www.reddit.com/r/Android/hot/.json?count=10</a>
	OddlySatisfaying	<a href="https://www.reddit.com/r/OddlySatisfaying/hot/.json?">https://www.reddit.com/r/OddlySatisfaying/hot/.json?</a>

### Puntos Extras

- Implementar el parser para leer feeds reddit.
- Proponer una nueva heurística para detectar entidades nombradas y comparar su efectividad contra las ofrecidas por la cátedra.
- Agregarle al lector de feeds alguna funcionalidad que a ustedes se les ocurra.

### Entrega y Defensa

Fecha límite de entrega: Jueves 11 de Mayo de 2023 a las 23:59:59 hs. Entrega del código fuente + informe mediante push en bitbucket en el repositorio asignado por la cátedra al grupo. Día y horario de la defensa a coordinar con el docente asignado a su grupo.