# Examen Final: Laboratorio 2

Paradigmas de la Programación - 11/02/2026

## Generales

- Para codear el parcial solo podrán utilizar algún editor de texto plano o IDE sin asistencia de IA. No utilizar teléfonos durante el examen. No acceder a sitios que permitan comunicarse con otras personas como whatsapp, discord, zulip, etc. **No respetar esta regla implica desaprobación inmediata del final y será calificada con 0 en el acta del examen.**
- Pueden utilizar sus propias laptops o alguna máquina del laboratorio que tenga instalado todo lo necesario para el lab2 (openjdk v21, etc).
- Pueden tener el código fuente del laboratorio 2 para consulta pero **NO** deben entregarlo. Solo entregar el código con la solución del examen. Resta puntos entregar el código de los laboratorios.
- **Verificar que se haya recibido el correo de confirmación** luego de la entrega**.** Si no llega, deben avisar antes de retirarse del aula. **No se aceptarán reclamos posteriores por entregas incorrectas o incompletas.**

El examen consiste en ir agregando con cada ejercicio, una nueva funcionalidad a nuestro FeedReader. Pero al ser un examen de POO, no solo es importante lograr la funcionalidad que se requiere (condición necesaria pero no suficiente) sino como modelan su solución a través de los conceptos de POO (clases, herencia, interfaces, modificadores de scope, etc), es decir, es importante la calidad de su solución.

## Esqueleto

Descargar del siguiente link:

Si al descomprimir se encuentran con archivos que comienzan con "._" deben borrarlos a todos, incluso en sub-directorio. Piensen y busquen un script que lo haga por ustedes

11-02-26-lab2-skeleton.tar.xz

Otras librerías que pueden ser útiles

json-20231013.jar

jsoup-1.21.1.jar

## Instrucciones para Compilar y Correr

Para compilar y ejecutar el código del examen, solo es necesario utilizar el Makefile provisto desde la carpeta del esqueleto:

```
None
$ make run ARGS="-h quick"
```

El resultado es similar a:

```
************* FeedReader version 2.0 *************
Named Entities Found:
Top 20 Named Entities by Frequency:
US (OTHER): 18
Tuesday (OTHER): 9
Trump (OTHER): 8
Gaza (OTHER): 6
Bloomberg (OTHER): 6
January (OTHER): 6
Heres (OTHER): 6
Stocks (OTHER): 5
Israeli (OTHER): 5
New (OTHER): 5
House (OTHER): 5
Dominion (OTHER): 5
Source (OTHER): 5
Street (OTHER): 4
AI (OTHER): 4
Jobs (OTHER): 4
GOP (OTHER): 4
Monday (OTHER): 4
Fox (OTHER): 4
Hostages (OTHER): 4
```

Notar que el esqueleto del examen difiere del código del laboratorio en varios puntos. En particular:

- Para asignar a cada palabra una categoría de entidad nombrada se utiliza el método Heuristic.getCategory. Si la palabra no es una entidad nombrada, entonces devuelve null.
- Para obtener las entidades nombradas de un artículo, se utiliza el método Heuristic.getNamedEntities

# News Topic Classifier

El objetivo del examen es convertir el `FeedReader` en un clasificador temático de noticias globales. El sistema debe ser capaz de identificar si un titular pertenece a secciones como Economía, Política, Tecnología o Mundo, basándose en un diccionario de términos extraídos de portales como CNN, Bloomberg y WSJ.

## Ejercicio 1. Heurística de Clasificación Temática (Map Lookup)

Crear una nueva heurística llamada `TopicHeuristic`. La detección debe hacerse mediante comparación exacta de cadenas (ignorando mayúsculas/minúsculas) contra una base de conocimientos en memoria.

**Requerimientos:**

- La heurística se selecciona con la opción `-h topic` al ejecutar el programa.
- Deben eliminarse las *stopwords* de la misma forma que en las otras heurísticas.
- Utilizar un `Map<String, String>` donde la Clave es la palabra y el Valor es la categoría (ej: "fed" -> "ECONOMY").
- Al instanciar la clase, se deben cargar (hardcodeado) las siguientes asociaciones:
    - **ECONOMY**: "fed", "stocks", "market", "inflation", "bank", "trade".
    - **POLITICS**: "trump", "harris", "election", "biden", "voters", "court".
    - **TECH**: "nvidia", "ai", "apple", "google", "data", "cyber".
    - **WORLD**: "china", "israel", "gaza", "beijing", "ukraine", "war".

**Salida esperada:**


## Ejercicio 2. Reporte de Titulares Destacados (Top 3)

Modificar el reporte para que se impriman únicamente las 3 entidades con mayor frecuencia por categoría detectada. Se pide imprimir la Categoría en mayúsculas, seguida de las entidades con su posición, nombre y cantidad.

```
************* FeedReader version 2.0 *************
Named Entities Found:

TECH
- data (3)
- ai (3)
- apple (2)

POLITICS
- trump (7)
- court (4)
- election (3)

WORLD
- gaza (6)
- war (3)
- china (2)

ECONOMY
- bank (4)
- trade (3)
- stocks (3)

Total Topic Tokens: 54
```

### Ejercicio 3. Carga de Diccionario desde JSON

En lugar de hardcodear el mapa, leer el archivo `config/topic_dictionary.json`. El alumno debe utilizar la librería `org.json` (proporcionada en `lib/`) para parsear el archivo.

**Formato del archivo `topic_dictionary.json`:**

```json
{
  "ECONOMY": ["fed", ...],
  "POLITICS": ["trump", "harris", ...],
  "TECH": ["nvidia", ...],
  "WORLD": ["china", ...]
}
```

**Nota:** Se debe transformar la estructura JSON (Categoría -> Lista) a la estructura del Mapa interno (Palabra -> Categoría). Al finalizar el reporte, el sistema debe imprimir el total de entidades clasificadas: `Total Topic Tokens: X`

Luego de compilar el ejercicio 3, el resultado es similar al siguiente ejemplo:

```
************* FeedReader version 2.0 *************
Named Entities Found:
Top 20 Named Entities by Frequency:
Trump (POLITICS): 8
Gaza (WORLD): 6
market (ECONOMY): 6
Stocks (ECONOMY): 5
data (TECH): 4
Data (TECH): 4
court (POLITICS): 4
stocks (ECONOMY): 4
election (POLITICS): 3
Trade (ECONOMY): 3
Bank (ECONOMY): 2
Ukraine (WORLD): 2
Market (ECONOMY): 2
Court (POLITICS): 2
Beijing (WORLD): 2
Apple (TECH): 2
Biden (POLITICS): 2
China (WORLD): 2
voters (POLITICS): 1
Israel (WORLD): 1
```

Notar que con cada heurística, el resultado es distinto, por ejemplo con quick:

```
************* FeedReader version 2.0 *************
Named Entities Found:
Top 20 Named Entities by Frequency:
US (OTHER): 19
Tuesday (OTHER): 9
Trump (OTHER): 8
Gaza (OTHER): 6
Bloomberg (OTHER): 6
January (OTHER): 6
Heres (OTHER): 6
Stocks (OTHER): 5
Street (OTHER): 5
Israeli (OTHER): 5
New (OTHER): 5
Wall (OTHER): 5
House (OTHER): 5
Dominion (OTHER): 5
Source (OTHER): 5
AI (OTHER): 4
Jobs (OTHER): 4
GOP (OTHER): 4
Monday (OTHER): 4
Fox (OTHER): 4
```

Nota: no deben eliminar el código existente.

## Ejercicio 4: Filtro de Exclusión de Dominio (Solo Alumnos Libres)

A diferencia de las *stopwords* gramaticales ya existentes (como "the", "is"), el sistema debe filtrar el "ruido" propio del dominio periodístico.

**Consigna:** Implementar un filtro que ignore términos genéricos de noticias, **a menos** que dicha palabra esté definida explícitamente en el diccionario de la heurística como una palabra clave.

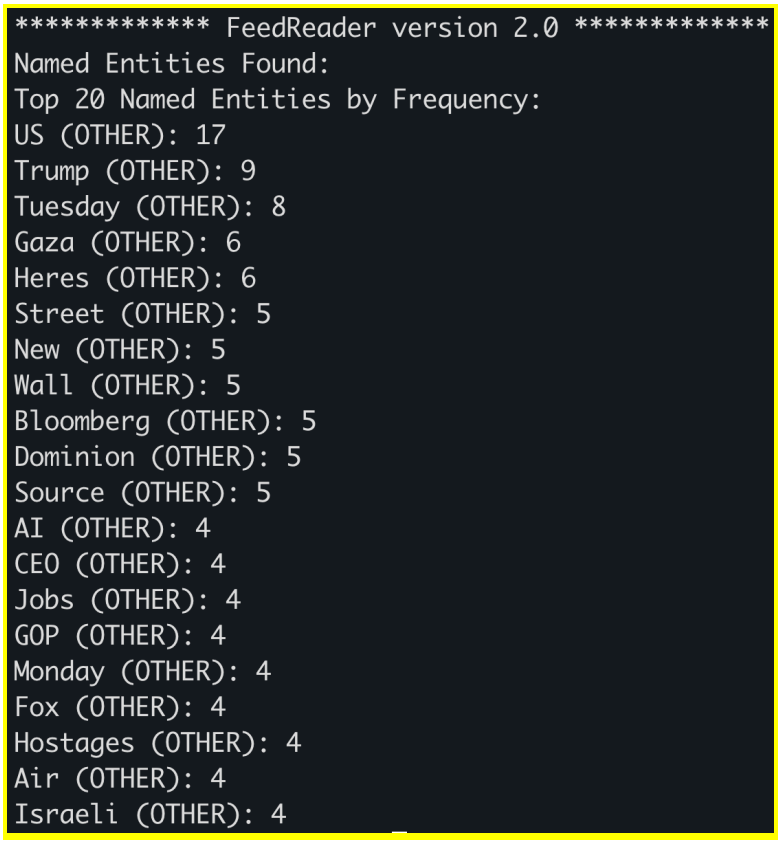
- **Términos genéricos a filtrar:** `news, update, video, report, today, latest, breaking`.
- **Ejemplo de comportamiento:**
    - Si la descripción dice *"Breaking news from the Fed"*:
        - `Breaking` y `news` son filtrados por ser términos genéricos.
        - `Fed` se mantiene porque es una **clave** en el diccionario de ECONOMY.

El ejercicio se resuelve de forma sencilla utilizando la lista de los términos genéricos mencionada y filtrando al obtener la categoría. Lo más importante del ejercicio no

son las líneas de código en sí, sino decidir en qué clase y/o método hacer la modificación de forma consistente con los principios de POO. Heuristic? Parser? Article? Feed? FeedReaderMain? Deberá dejar un comentario en el código respondiendo la siguiente pregunta:

> **¿Dónde se agregó el código? Utilice los principios de POO para justificar su elección**

Luego de implementar el ejercicio, el resultado (sin la modificación del ejercicio 1) se verá como:




Nota: recordá usar la heurística quick y el método original de reporte.

## Entrega

Comprimir el directorio raíz de su examen en un archivo “tar.xz” con el siguiente formato “Apellido_Nombre.tar.xz”. Para que quede con el formato adecuado, usar los siguientes comandos:

```shell
mv skeleton Juarez_MatiasAlfonso
tar -czvf Juarez_MatiasAlfonso.tar.xz Juarez_MatiasAlfonso
```

Adjuntar el comprimido en el formulario de entrega, llenar todos los campos del mismo y submitear con su correo institucional. Por último, chequear en su bandeja de entrada si recibieron el mail de confirmación de entrega realizada, gracias.