

Examen Final: Laboratorio 2

Paradigmas de la Programación - 1/07/2025

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, ant, etc).
- Pueden tener el código fuente del laboratorio 2 para consulta pero **NO** deben entregarlo. Solo entregar el código con la solución del examen.
- **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.

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 dict"
```

El resultado es similar a:

```
$ make run ARGS="-h quick"
java -cp "bin:lib/json-20231013.jar:" FeedReaderMain -h quick
***** FeedReader version 2.0 *****
Named Entities Found:
Top 20 Named Entities by Frequency:
Engineering (OTHER): 101
Daily</a></p> (OTHER): 100
GPIO (OTHER): 23
AI (OTHER): 18
TI (OTHER): 18
GitLab (OTHER): 16
However (OTHER): 15
Reolink (OTHER): 14
Linux (OTHER): 13
Allwinner (OTHER): 13
USB (OTHER): 13
Wi-Fi (OTHER): 13
APIs (OTHER): 12
Software (OTHER): 12
Pi&rsquos (OTHER): 12
API (OTHER): 11
Programmer</a></p> (OTHER): 10
Contents</label> (OTHER): 10
Crazy (OTHER): 10
Best (OTHER): 10
```

Ejercicio 1. Heurística de clasificación usando diccionarios

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`

El ejercicio consiste en agregar una nueva heurística `DictionaryHeuristic` que devuelva una categoría (PERSON, COMPANY, LOCATION, OTHER) para una palabra si dicha palabra está en un diccionario predefinido de entidades nombradas. Si la palabra no está en el diccionario, el método debe devolver null.

Requerimientos:

1. La heurística se debe poder seleccionar con la opción `-h dict` al ejecutar el programa.
2. El mapa de palabras a categorías puede estar hardcodeado en el código.
3. La comparación para determinar si una palabra está en el diccionario debe ser **sin distinguir mayúsculas de minúsculas**. Es decir, tanto “Argentina” como “argentina” son consideradas entidades nombradas con categoría LOCATION.
4. Deben eliminarse las **stopwords** de la misma forma que en las otras heurísticas (no deben clasificarse ni contarse si aparecen en el texto).

Deben utilizar este mapa de palabras a categorías en su solución:

```
None
// Company
"Microsoft": "COMPANY"
"Apple": "COMPANY"
"Google": "COMPANY"
"DoorDash": "COMPANY"
"OpenAI": "COMPANY"
"NVIDIA": "COMPANY"
"GitLab": "COMPANY"
// People
"Torvalds": "PERSON"
"Marx": "PERSON"
"Turing": "PERSON"
// Locations
"USA": "LOCATION"
"Germany": "LOCATION"
"India": "LOCATION"
// Other
"Linux": "OTHER"
"Python": "OTHER"
"Rust": "OTHER"
"Ethereum": "OTHER"
```

Para compilar y ejecutar el ejercicio 1 se pasa un nuevo argumento al comando make:

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

El resultado es similar a:

```
└$ make run ARGS="-h dict"
java -cp "bin:lib/json-20231013.jar:" FeedReaderMain -h dict
***** FeedReader version 2.0 *****
Named Entities Found:
Top 13 Named Entities by Frequency:
GitLab (COMPANY): 16
Linux (OTHER): 13
Google (COMPANY): 8
India (LOCATION): 7
NVIDIA (COMPANY): 6
Python (OTHER): 6
Rust (OTHER): 4
Microsoft (COMPANY): 4
Apple (COMPANY): 3
Ethereum (OTHER): 3
OpenAI (COMPANY): 3
DoorDash (COMPANY): 1
Germany (LOCATION): 1
```

Ejercicio 2. Imprimir conteo de entidades por categoría (solo alumnos libres)

El esqueleto cuenta las entidades nombradas de todos los artículos en todos los feeds e imprime su frecuencia agrupandolas por nombre. Por ejemplo:

```
None
Torvalds (PERSON): 3
Marx (PERSON): 3
Argentina (LOCATION): 5
GitLab (COMPANY): 2
```

Se pide modificar el sistema para que imprima la frecuencia de cada categoría de entidades nombradas. Por ejemplo, en el caso anterior el resultado sería:

```
None
PERSON: 6
LOCATION: 5
COMPANY: 2
```

Luego de compilar el ejercicio 2, el resultado es similar al siguiente ejemplo:

```
└$ make run ARGS="-h quick"
mkdir -p bin
javac -d bin -cp "lib/json-20231013.jar:" src/FeedReaderMain.java src/
ser.java src/subscription/Subscription.java src/feed/Feed.java src/fe
dParser.java src/namedEntity/heuristic/Heuristic.java src/namedEntity
ndomHeuristic.java src/namedEntity/heuristic/QuickHeuristic.java src/
java -cp "bin:lib/json-20231013.jar:" FeedReaderMain -h quick
***** FeedReader version 2.0 *****
Named Entities Found:
OTHER: 4202
```

Notar que con cada heurística, el resultado es distinto:

```
└$ make run ARGS="-h random"
java -cp "bin:lib/json-20231013.jar:" FeedReaderMain -h random
***** FeedReader version 2.0 *****
Named Entities Found:
OTHER: 6673
PERSON: 6751
COUNTRY: 6666
```

Ejercicio 3. Limpiar los tags `html` de los artículos (solo para alumnos libres)

La consigna de este ejercicio es limpiar los tags HTML de los artículos antes de procesar su contenido para extraer entidades nombradas.

El ejercicio se resuelve de forma sencilla utilizando las líneas:

```
None
import org.jsoup.Jsoup;
Jsoup.parse(htmlText).text();
```

Esto eliminará todos los tags y dejará solamente el texto plano, evitando problemas de tokens como ``, `
`, ` `, etc.

Lo más importante del ejercicio no es la línea de código en sí, sino decidir en qué clase y/o método colocar esta limpieza de HTML de forma consistente con los principios de POO. Heuristic? Parser? Article? Feed? FeedReaderMain? En el formulario de entrega del examen, debe responder 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:

```
[base] [~] addemilagros MacBook Pro ~ skeleton
└─$ make run ARGS="-h quick"
mkdir -p bin
javac -d bin -cp "lib/jsoup-1.21.1.jar:lib/json-20231013.jar:" src/Fe
ption/SubscriptionParser.java src/subscription/Subscription.java src/
va src/feedParser/FeedParser.java src/namedEntity/heuristic/Heuristi
edEntity/heuristic/RandomHeuristic.java src/namedEntity/heuristic/Qu
java -cp "bin:lib/jsoup-1.21.1.jar:lib/json-20231013.jar:" FeedReader
***** FeedReader version 2.0 *****
Named Entities Found:
Top 20 Named Entities by Frequency:
Software (OTHER): 105
Engineering (OTHER): 101
Daily (OTHER): 100
GPIO (OTHER): 29
TI (OTHER): 19
GitLab (OTHER): 18
AI (OTHER): 18
However (OTHER): 16
Linux (OTHER): 14
Reolink (OTHER): 14
Allwinner (OTHER): 14
APIs (OTHER): 13
USB (OTHER): 13
Pis (OTHER): 12
RAM (OTHER): 12
Wi-Fi (OTHER): 12
API (OTHER): 12
Plus (OTHER): 11
RTSP (OTHER): 11
Ghidra (OTHER): 11
```

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.