**Laboratorio - Parcial 2**

**Programación Orientada a Objetos en Java**

**Generales**

* Para codear el parcial solo podrán utilizar algún editor de texto plano o IDE (sin asistencia). No utilizar teléfonos durante el examen. Por favor, ser honestos intelectualmente y no poner al docente en una situación incómoda.🙏
* 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 el código fuente del examen deben submitear.
* NO compila => Reprobado

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.

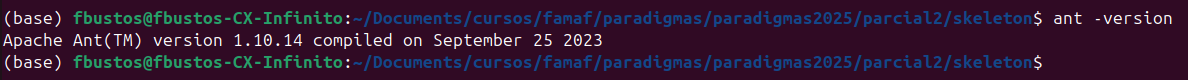
**Instrucciones para Compilar y Correr**

Para compilar y ejecutar el código del examen, debemos tener instalado una herramienta llamada “Ant” la cual facilita la gestión de proyectos en java con muchos archivos. Para instalarlo, ejecutamos el siguiente comando en la terminal de linux:

$ sudo apt install ant.

Y luego, verificamos si la instalación fue exitosa:

$ ant -version



Ant mediante el seteo de un archivo de configuración (build.xml) se encarga de compilar y linkear todo el proyecto llamando “por detrás” al comando “javac” y “java” que deben estar ya instalados en el sistema cuando instalaron una JDK.

La cátedra provee en la raíz del skeleton el archivo “build.xml” ya configurado para compilar y correr todo el parcial (o sea, no deben preocuparse por su configuración).

Para chequear que esta todo OK para empezar a codear el parcial nos posicionamos en el directorio donde se encuentra “build.xml” y ejecutamos en la terminal:

$ ant run\_ok

Y si vemos al final de la terminal el mensaje **“TODO OK PARA EL PARCIAL, gracias.”** entonces tenemos todo listo para empezar a trabajar.

Más tarde, para compilar y correr el código fuente del ejercicio X nos posicionamos en el directorio donde se encuentra “build.xml” y ejecutamos en la terminal:

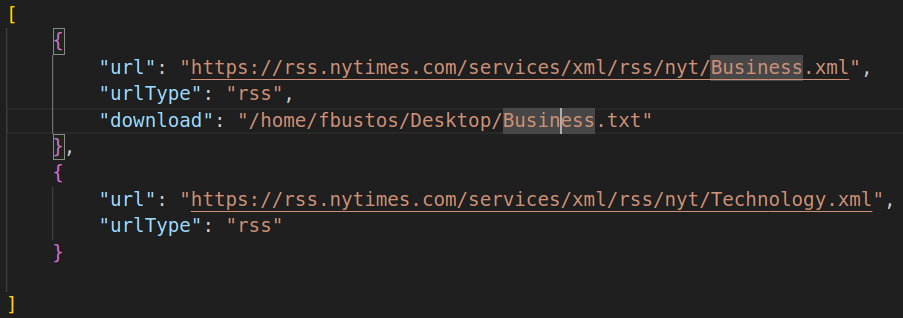
$ ant run\_ejX

Para eliminar los binarios generados por una compilación ejecutamos en la terminal:

$ ant clean

**Ejercicio 1. Persistencia en disco de los Feeds**

En el fichero “config/*subscriptions\_ej1.json”* vamos a agregar un tercer y nuevo “feature” a cada suscripción llamado “download”, cuyo seteo es opcional (puede estar definido o no) y su valor es una ruta absoluta a un fichero. En caso de estar seteado, la aplicación debe guardar en el archivo especificado por la ruta los artículos del feed respectivo, en lugar de imprimirlos por pantalla. En caso de no estar seteado el “download”, imprimir los artículos del feed por pantalla. Por ejemplo, si tenemos:

****

Entonces, los artículos del feed “Business” deben ser mostrados en el archivo “Business.txt”. Mientras que los artículos del feed “Technology” deben ser mostrados por pantalla, ya que, no tiene definido su feature “download”.

Para compilar y ejecutar el ejercicio 1, hacemos:

$ ant run\_ej1

Solo veremos por pantalla los artículos del feed “Technology”, y si abrimos el archivo “Business.txt” veremos los artículos de dicho feed.

**Ejercicio 2. Nueva Heurística para detectar Entidades Nombradas**

Se requiere implementar una nueva heurística para detectar entidades nombradas, llamada “ParityHeuristic” que clasifica una palabra como entidad nombrada si y sólo si su longitud es un número par.

Para compilar y ejecutar el ejercicio 2, hacemos:

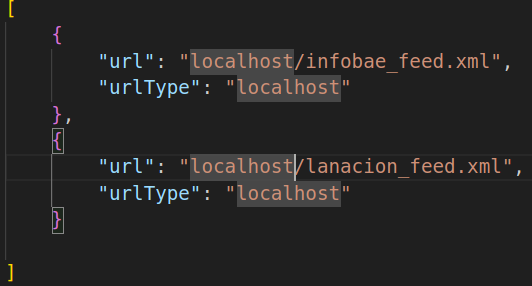
$ ant run\_ej2

Veremos por pantalla las entidades nombradas del feed “Business” las cuales serán todas palabras de longitud par y su cantidad de ocurrencias:

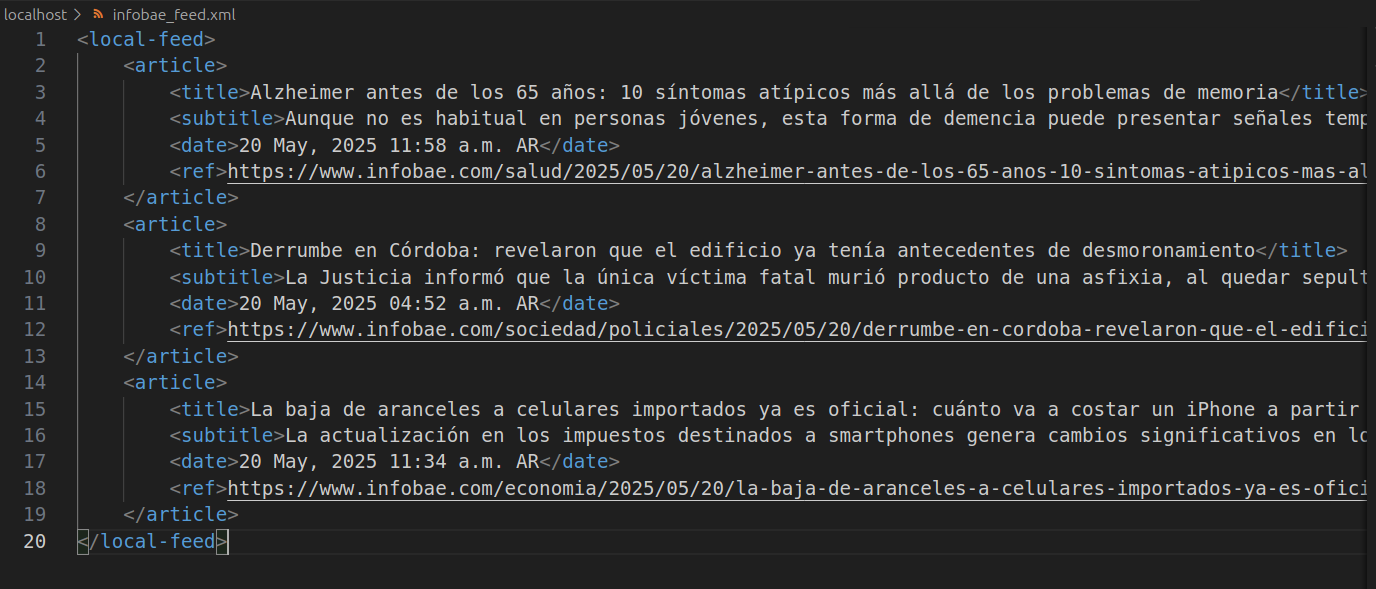


**Ejercicio 3. Servidor de Feed Local**

Implementaremos un nuevo tipo de servidor de feed “localhost”:



cuya respuesta es un xml con el formato que se muestra a continuación :

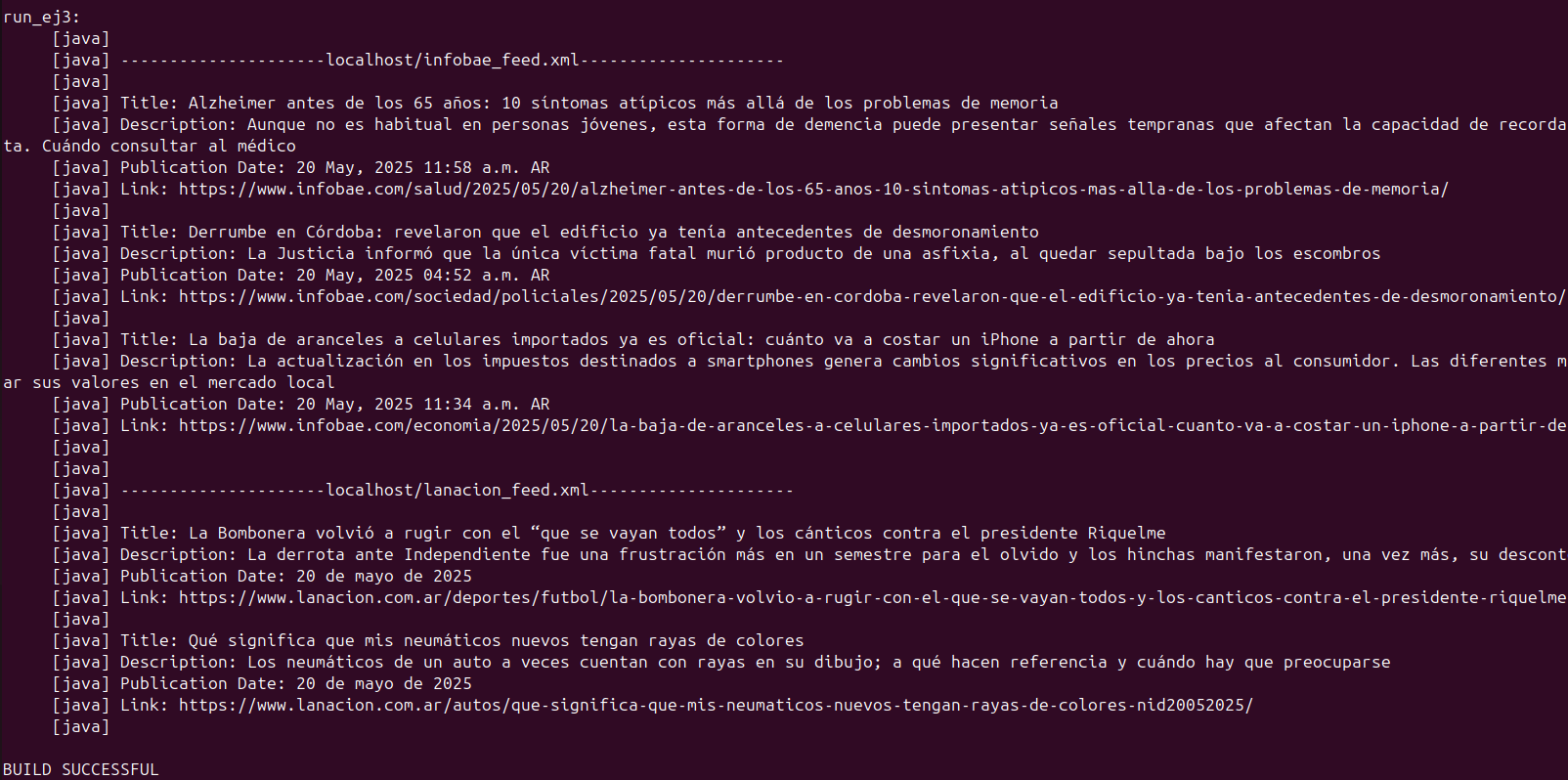


Realizar todo lo necesario para que este nuevo tipo de feed “localhost” pueda ser consumido por nuestra aplicación.

Para compilar y ejecutar el ejercicio 3, hacemos:

$ ant run\_ej3

Veremos por pantalla los artículos de los feed locales de Infobae y LaNacion:



**Entrega**

Comprimir el directorio raíz de su examen en un archivo “[tar.xz](http://tar.xz)” con el siguiente formato “Apellido\_Nombre[.tar.xz](http://.tar.xz)”. Para que quede con el formato adecuado, usar los siguientes comandos:

| 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.