

# TICS200: Proyecto Semestral

Universidad Adolfo Ibáñez

Justo Miguel Vargas  
[justo.vargas@edu.uai.cl](mailto:justo.vargas@edu.uai.cl)  
06 de junio de 2022

## Objetivos

- Aprender a estructurar una aplicación multiparadigma.
- Aplicar paradigma **procedural**, **orientado a objetos** y **funcional** en lenguaje **Python/JAVA/JavaScript**.
- Desarrollar un entregable funcional
- Se puede proponer un tema o utilizar el tema descripto abajo.

## 1. Medicamentos

Debido al excelente desempeño en las aplicaciones desarrolladas por Usted para la Biblioteca de la Universidad, tanto en paradigma procedural, como en paradigma orientado a objetos, y a su excelente solución para el Laberinto de Creta en paradigma funcional, la unidad de estudios de la **Universidad Adolfo Ibáñez** le encarga que desarrolle una aplicación **multiparadigma** que permita realizar el procesamiento de información de medicamentos a partir de sus principios activos, capturando para ello la información disponible en Internet. La aplicación deberá realizar lo siguiente:

- Leer un paquete de principios activos.
- Para cada principio activo del paquete, buscar los medicamentos disponibles en los sitios web de las cadenas de farmacias **Cruz Verde**, **Ahumada** y **Salcobrand**.
- Almacenar los resultados de todas las búsquedas en un archivo de salida.
- Generar estadísticas de la información procesada.

## 2. Requerimientos NO Funcionales

1. Desarrollar la aplicación en lenguaje **Python 3+/JAVA 17/JavaScript**.
2. Desarrollar en **paradigma procedural** la lectura del archivo de entrada y de los datos disponibles en Internet utilizando técnica **Web Scraping**.
3. Desarrollar en **paradigma orientado a objetos** las estructuras de datos para administrar todo lo relativo a información de medicamentos y precios (o lo relativo a su tema), utilizando herencia, getters, setters, y demás propiedades de la POO para todas las propiedades de las clases. La generación de archivos de salida será en POO.
4. Desarrollar en **paradigma funcional** las estadísticas solicitadas.

### 3. Requerimientos Funcionales

1. Realizar la captura del valor de la unidad de fomento (UF) desde el sitio web del **Banco Central de Chile** <https://www.bcentral.cl/inicio> y guardar la fecha y el valor de la unidad de fomento en el archivo de salida **parámetros.csv** (se adjunta ejemplo), el que contendrá solo un registro.
2. Leer paquete de principios activos desde el archivo de entrada **principios\_activos.txt** (se adjunta ejemplo) y por cada registro, hacer búsquedas de los medicamentos disponibles en los sitios web de las cadenas de farmacias:
  - Cruz Verde <https://www.cruzverde.cl/>
  - Ahumada <https://www.farmaciasahumada.cl/>
  - Salcobrand <https://salcobrand.cl/>
3. Por cada búsqueda realizada, deberá capturar todos los medicamentos disponibles, incluyendo aquellos que se encuentren paginados<sup>1</sup>. Los resultados de todas las apariciones deben quedar almacenadas en el archivo de salida **medicamentos.csv** (se adjunta ejemplo), el que debe contener los siguientes campos:
  - Principio activo.
  - Farmacia (Cruz Verde, Ahumada o Salcobrand).
  - Descripción del medicamento.
  - Precio en pesos chilenos, sin decimales, que corresponde al precio final al público publicado (incluida oferta si es que la hay).
  - Precio en UF, con 2 decimales, que corresponde a la conversión del precio en pesos chilenos a unidades de fomento.
4. Generar 2 estadísticas por integrante de grupo. Les paso algunos ejemplos, cada punto se considera una estadística:
  - **Precios mínimos por principio activo:** Principio activo, farmacia, descripción del medicamento, precio **mínimo** en pesos y UF (1 registro por cada principio activo).
  - **Precios máximos por principio activo** Principio activo, farmacia, descripción del medicamento, precio **máximo** en pesos y UF (1 registro por cada principio activo).
  - **Precios promedios por principio activo y farmacia:** Principio activo, farmacia, precio **promedio** en pesos y UF (1 registro por cada combinación de **principio activo y farmacia**).
  - **Precios por principio activo:** Principio activo, precio **mínimo, máximo y**

---

<sup>1</sup> Por ejemplo, al buscar el principio activo **metformina** en el sitio web de farmacias **Cruz Verde** se realiza paginación, para lo cual aparecen **7 páginas** de información y en cada una de ellas **hasta 12 medicamentos**. Usted deberá captura la información de todos los medicamentos de todas las páginas.

**promedio** en pesos y UF (1 registro por cada principio activo).

#### 4. Sobre la entrega

- La aplicación debe ser implementada en lenguaje **Python/Java/Javascript** con **paradigma procedural, orientado a objetos y funcional**, según corresponda a lo indicado en los **Requerimientos NO Funcionales**.
- Los equipos de trabajo pueden estar integrados por 4 o 5 personas.
- El plazo para entrega es el martes 28 de junio a las 23:59.
- Se debe entregar el repo completo por medio de github.
- El entregable señalado, debe incluir un comentario con los integrantes del grupo y solo uno de ellos debe realizar la entrega en el buzón habilitado en la plataforma WebCursos.
- Se aplicará 1 punto de descuento por cada día de atraso o fracción.

#### 5. Pauta de Evaluación

La aplicación será revisada utilizando la siguiente pauta de evaluación:

Concepto de evaluación	Descripción	Puntos	Descuento
Punto base	Punto base	1.0	No aplica
Aplicación de los paradigmas	<b>Paradigma procedural:</b> <ul style="list-style-type: none"> <li>- NO usar paradigma imperativo</li> <li>- Usar un orquestador</li> <li>- Usar procedimientos/funciones con propósitos específicos a cada uno</li> </ul> <b>Paradigma orientado a objetos:</b> <ul style="list-style-type: none"> <li>- Herencia.</li> <li>- Setters &amp; Getters</li> <li>- Encapsulamiento y ABstraccion</li> <li>- Principio de Responsabilidad Unica</li> </ul> <b>Paradigma funcional:</b> <ul style="list-style-type: none"> <li>- NO usar bucles nativos (for, while u otros), en su reemplazo debe utilizar funciones de orden superior (map, filter o reduce)</li> <li>- NO usar variables mutables. Excepcionalmente se permite uso reducido de variables inmutables, es decir, como constantes (en mayúsculas según norma) y sin realizar modificación de ellas después de ser definidas.</li> </ul>	4.0	0.5 por cada desviación
Funcionalidades	Cumplimiento del <b>Requerimientos Funcionales</b>	1.0	0.5 por cada incumplimiento funcional
Lógica y lenguaje	Uso adecuado del lenguaje y lógica empleada: <ul style="list-style-type: none"> <li>- Optimización códigos fuentes</li> <li>- Uso de comentarios</li> <li>- Orden del código fuente</li> <li>- Uso adecuado de las estructuras de datos y sentencias del lenguaje</li> </ul>	1.0	0.3 por cada desviación 0.5 por cada caída del programa