

# Trabajo Práctico 4: Listas

## Programación I

*El objetivo de este trabajo práctico es incorporar el uso de listas en el diseño de programas.*

---

### 1 Enunciado

Diariamente, el ministerio de Salud de la provincia de Santa Fe emite partes informativos sobre la evolución de los casos de COVID-19 en la provincia. Además, dichos datos se publican en formato abierto para que sean utilizados, reutilizados y redistribuidos libremente.

En este trabajo práctico, se propone usar esos datos para responder algunas consultas puntuales. Para ello, deberá resolver diversos ejercicios que podemos agrupar en tres etapas:

1. Obtención y preparación de los datos.
2. Formulación de consultas para analizar esos datos.
3. Presentación de resultados.

El archivo TP4-Apellido1-Apellido2.rkt contiene una implementación parcial del trabajo, que deberá completar adecuadamente según se indica en las siguientes secciones.

Trabaje con el lenguaje "Estudiante Intermedio" y agregue el paquete de enseñanza "batch-io.rkt".

---

#### 1.1 Datos

Para obtener los datos con los que deberá trabajar, diríjase al siguiente enlace <https://www.santafe.gob.ar/ms/covid19/datosabiertos/> y descargue el dataset.

También puede usar el dataset adjunto a este enunciado, que fue descargado de dicha página y corresponde al día 2 de Junio.

---

##### 1.1.1 Dataset

Un dataset es una colección de datos, usualmente organizada en tablas. Cada columna de cada tabla representa una variable, mientras que cada fila

representa a un miembro del conjunto de datos.

El dataset sobre COVID-19 provisto por la provincia de Santa Fe consiste de cuatro tablas:

- **santa\_fe\_departamento\_localidad.csv**
- **notificaciones\_localidad.csv**
- **notificaciones\_total\_provincial.csv**
- **notificaciones\_grupo\_etario.csv**

Para este trabajo práctico se usarán, únicamente, las primeras dos.

La tabla **santa\_fe\_departamento\_localidad**, contiene datos relativos a las localidades de la provincia de Santa Fe. Las primeras filas de esta tabla muestran lo siguiente:

| Departamento | Localidad     | Latitud           | Longitud          |
|--------------|---------------|-------------------|-------------------|
| 9 de Julio   | ESTEBAN RAMS  | -29.7714746269502 | -61.4892977634018 |
| 9 de Julio   | GATO COLORADO | -28.0221451470519 | -61.1868995400897 |
| ...          | ...           | ...               | ...               |

La tabla **notificaciones\_localidad.csv**, contiene datos sobre las notificaciones de COVID-19:

| Fecha      | Localidad    | Confirmados | Descartados | En estudio | Notificaciones |
|------------|--------------|-------------|-------------|------------|----------------|
| 2020-03-24 | ARMSTRONG    | 0           | 1           | 0          | 1              |
| 2020-03-24 | ARROYO LEYES | 0           | 0           | 1          | 1              |
| ...        | ...          | ...         | ...         | ...        | ...            |

Cada fila de la tabla corresponde al conjunto de las notificaciones registradas en una localidad hasta una fecha dada (columnas dos y uno, respectivamente). Las siguientes cuatro columnas refieren al número de casos:

- Confirmados indica el número de casos con resultado positivo de test para COVID-19.
- Descartados indica el número de casos que no cumplen con las condiciones de caso confirmado.
- En estudio indica el número de casos sospechosos según situación epidemiológica o en espera del resultado del test para COVID-19, y que por lo tanto aún no se pueden clasificar en alguna de las dos categorías anteriores.

- Notificaciones indica el número total de notificaciones al momento (es decir, es la suma de los casos confirmados, descartados y en espera).

Observar que, mientras no se registren notificaciones en una localidad, la misma no será incluida en esta tabla (con lo cual el valor de la última columna siempre será mayor a 0). Por ejemplo, al día 2 de Junio, las únicas filas correspondientes a la localidad Felicia son:

|            |         |   |   |   |   |
|------------|---------|---|---|---|---|
| 2020-05-27 | FELICIA | 0 | 0 | 1 | 1 |
| 2020-05-28 | FELICIA | 0 | 0 | 1 | 1 |
| 2020-05-29 | FELICIA | 0 | 0 | 1 | 1 |
| 2020-05-30 | FELICIA | 0 | 1 | 1 | 2 |
| 2020-05-31 | FELICIA | 0 | 2 | 0 | 2 |
| 2020-06-01 | FELICIA | 0 | 2 | 0 | 2 |
| 2020-06-02 | FELICIA | 0 | 2 | 0 | 2 |

dado que antes del 27 de mayo, no hubo notificaciones en esa localidad.

## 1.1.2 Preparación de los datos

Para poder trabajar con esos datos, primero es necesario leerlos, para lo cual se usa la función `read-csv-file`.

`read-csv-file` toma como argumento el nombre de un archivo de extensión `.csv` y devuelve una lista de listas de strings, de manera tal que:

- la  $i$ -ésima lista de strings corresponde a la  $i$ -ésima fila de la tabla,
- el  $j$ -ésimo elemento de la  $i$ -ésima lista de strings es un string que representa el valor que corresponde a la  $i$ -ésima fila y  $j$ -ésima columna de la tabla.

Por ejemplo, para leer los datos de la tabla **santa\_fe\_departamento\_localidad** procedemos de la siguiente manera:

```
;; (define INPUT-LOC (read-csv-file "dataset/santa_fe_departamento_localidad.csv"))
```

Si consideramos unicamente los primeros tres elementos de `INPUT-LOC` tenemos lo siguiente:

```
;; (list
;;   (list "Departamento" "Localidad" "Latitud" "Longitud")
;;   (list "9 de Julio" "ESTEBAN RAMS" "-29.77147462695020" "-61.48929776340180")
;;   (list "9 de Julio" "GATO COLORADO" "-28.02214514705190" "-61.18689954008970"))
```

Luego, descartamos el primer elemento de la lista, ya que corresponde al encabezado de la tabla:

```
;; (define DATOS-LOC (rest INPUT-LOC))
```

Por último, procesamos esa lista para eliminar los datos que no usaremos (en nuestro caso, no nos interesa conocer la latitud y longitud de las localidades). En particular, para este problema es útil tener una lista de los nombres de las localidades y una en la que además se detalle a qué departamento corresponden, con lo cual definimos:

```
(define LISTA-LOC (map second DATOS-LOC))
(define LISTA-DPTO-LOC (map tomar-dos DATOS-LOC))
```

Para el caso de la tabla **notificaciones\_localidad**, se usará una estructura para representar notificaciones:

```
(define-struct notificacion [fecha loc conf desc sosp notif])
; notificacion es (String, String, Number, Number, Number, Number)
; Interpretación: un elemento en notificacion representa el conjunto
; de notificaciones registradas en una localidad (loc) hasta un día
; (fecha), en donde:
; - hay conf casos confirmados de COVID-19
; - hay desc casos descartados de COVID-19
; - sosp casos estaban en estudio.
; El último elemento, notif, indica la cantidad total de notificaciones.
```

**Ejercicio 1.** Defina una constante LISTA-NOTIF que represente al conjunto de datos dado por notificaciones\_localidad.csv. Tenga en cuenta que LISTA-NOTIF deberá ser una lista de elementos de tipo notificacion.

---

## 1.2 Consultas

**Ejercicio 2.** Resulta de interés determinar qué localidades han superado un cierto número de casos confirmados de COVID-19. Por ello, se pide:

1. Diseñe una función localidades-limite-casos que dada una lista de notificaciones, devuelva la lista de localidades que presentan LIMITE-CASOS o más casos confirmados de COVID-19 hasta el día HOY.
2. Defina una constante LOCALIDADES-LIMITE-CASOS que represente la lista de localidades santafesinas con LIMITE-CASOS o más casos confirmados de COVID-19 hasta el día HOY.

Al resolver este ejercicio, utilice foldr, filter y/o map para cualquier función que requiera procesar los elementos de una lista.

**Ejercicio 3.** Se requiere calcular la cantidad de casos confirmados en cada departamento en una determinada fecha. Para ello, resuelva cada una de las siguientes consignas:

1. Defina una constante LISTA-DPTO que represente la lista de los departamentos santafesinos.
2. Diseñe una función confirmados-dpto-fecha que dada una lista de

notificaciones, un departamento y una fecha, calcule la cantidad de casos confirmados registrados en dicho departamento hasta la fecha en cuestión.

3. Diseñe una función `confirmados-por-dpto` que dada una lista de notificaciones y una fecha, devuelva una lista de listas de longitud dos: una por cada departamento santafesino, donde el primer elemento sea el nombre del departamento y el segundo sea el número de casos confirmados que se han registrado allí hasta la fecha en cuestión.
4. Defina constantes `CONFIRMADOS-DPTO-ANTES` y `CONFIRMADOS-DPTO-HOY` para representar el resultado de evaluar la función `confirmados-por-dpto` con los datos del dataset, en las fechas `ANTES` y `HOY`, respectivamente.

Al resolver este ejercicio, **no** utilice `foldr`, `filter` y/o `map`.

---

## 1.3 Presentación de resultados

Para finalizar, debe presentar los resultados obtenidos mediante tablas en archivos `.csv`. Para ello, utilice la función `write-file`.

Dado un string `s` y un contenido `cont`, en caso de éxito, `write-file` crea un archivo de nombre `s` cuyo contenido es `cont`.

Para generar tablas, el contenido `cont` debe ser un string donde las filas se separan por `"\n"` y los valores dentro de cada fila se separan con `","`.

Por ejemplo, para escribir la siguiente tabla:

---

| Nombre | Edad | Peso |
|--------|------|------|
| José   | 30   | 80   |
| Ana    | 35   | 61   |

en un archivo `"tabla.csv"`, escribimos el siguiente código:

```
⋮ (write-file "tabla.csv" "Nombre, Edad, Peso\n José, 30, 80\n Ana, 35, 61\n")
```

**Ejercicio 4.** Escriba código para generar los siguientes archivos de salida:

- `loc-lim-casos.csv`: debe contener una tabla (de una única columna) con todas las localidades que a la fecha `HOY` registran `LIMITE-CASOS` o más casos confirmados de COVID-19.
- `casos-por-dpto-hoy.csv`: debe contener una tabla indicando los casos confirmados de COVID-19 de cada departamento a la fecha `HOY`. La primer columna de la tabla debe indicar el nombre de los departamentos, mientras que la segunda corresponde al número de casos.
- `casos-por-dpto-antes.csv`: análogo a `casos-por-dpto-hoy.csv`, pero refiriendo a la fecha `ANTES`.

En todos los casos, incluya encabezados apropiados.

Al resolver este ejercicio, utilice `foldr`, `filter` y/o `map` para cualquier función que requiera procesar los elementos de una lista.

---

## 2 Formato de entrega

- El trabajo práctico debe resolverse en grupos de dos integrantes. Cada persona puede participar en un único grupo. Un grupo puede estar integrado por personas de distinta comisión.
- Exactamente una persona por grupo debe realizar la entrega en el sitio, escribiendo en el campo "Comentarios" los apellidos y nombres de cada integrante.
- No se aceptarán entregas en las que no se haya seguido la receta para el diseño. Para que la entrega del trabajo práctico sea válida, todas las funciones deben contar con diseño de datos, signatura, declaración de propósito, casos de prueba (si corresponde) y código. En caso contrario, se considerará que el TP no fue entregado.
- La entrega consiste en un único archivo por grupo: el archivo TP4-Apellido1-Apellido2.rkt que deberá ser editado apropiadamente. En concreto, se solicita:
  - Completar los datos pedidos para cada integrante del grupo según se indica en el archivo.
  - Completar el archivo con la resolución.
  - Cambiar el nombre del archivo reemplazando Apellido1 y Apellido2 por los apellidos de los integrantes, en orden alfabético ascendente.

No deben entregarse los archivos generados en el ejercicio 4 ni el dataset.