

v.7.6

# Trabajo Práctico 2: Listas

## Programación I

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

---

### 1 Enunciado

Periódicamente, los países recuentan sus habitantes y registran diversas características de su población. Cada país censa con diferente frecuencia. En Argentina, se censa a la población en promedio cada 10 años desde 1969.

---

#### 1.1 Datos

El conjunto de datos que usaremos para este trabajo lo provee el banco de datos [WorldBank](#). Al ser un conjunto de datos **libre**, es posible utilizarlo, modificarlo y distribuirlo con modificaciones siempre y cuando se le de crédito a *WorldBank*. Para consultar más detalles sobre esto, ver la [licencia](#) con la que está publicado.

El documento que utilizaremos como base de datos o *dataset* lo pueden encontrar adjunto bajo el nombre BDPoblacional.csv

---

#### 1.2 Familiarizandose con el dataset

Las primeras filas del dataset muestran lo siguiente:

---

Country Name	1960	1961	1962	...	2019
Aruba	54211	55438	56225	...	106314
Afghanistan	8996973	9169410	9351441	...	38041754

Angola	5454933	5531472	5608539	...	31825295
...	...	...	...	...	...

La primer columna contiene los nombres de todos los países con al menos 1 censo registrado entre el año 1960 y 2019. Y a partir de las segunda columna y hasta la columna número 60 (contando desde el 0) se encuentra la información recabada en dichos censos por año.

Observar que algunas celdas están completas con el número -1. Dicho valor representará que no hay registro para el par (pais, año) al que haga referencia la celda.

---

## 1.3 Preprocesamiento de los datos

Para poder trabajar con los datos, se debe leer el dataset. Eso se puede conseguir utilizando la función `read-csv-file` provista por `racket`.

`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.

En particular para `BDpoblacional.csv` tenemos

```
(define INPUT-PAISES (read-csv-file "BDPoblacional.csv"))
```

Notar que esta lectura solamente funciona si la base de datos está en el mismo directorio que el archivo extensión `rkt` que incluye esta línea.

Los primeros 4 elementos de cada uno de los 3 primeros elementos de `INPUT-PAISES` se ven así:

```
(list (list
      "Country Name"
      "1960"
      "1961"
      "1962"
```

```

    "...")
  (list
    "Aruba"
    "54211"
    "55438"
    "56225"
    "...")
  (list
    "Afghanistan"
    "8996973"
    "9169410"
    "9351441"
    "...")
  ...)

```

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

```
(define DATOS-PAISES (rest INPUT-PAISES))
```

**Importante:** se debe trabajar con el lenguaje "Estudiante Intermedio" y agregar el paquete de enseñanza "batch-io.rkt".

---

### 1.3.1 Diseño de datos

Representamos los nombres de los países con String y las fechas (años) y el número de habitantes de un país con Number.

Definimos la estructura **Registro**

```

(define-struct Registro [Fecha Poblacion])
; Registro es (Number, Number)
; Interpretación: un elemento en Registro representa el número
; de habitantes de un país (Poblacion) en una determinada fecha (Fecha).

```

y la estructura **Pais**

```

(define-struct Pais [Nombre Registros])
; Pais es (String, List(Registro))
; Interpretación: un elemento en Pais representa un pais de nombre
; Nombre con su lista de registros de censo asociado Registros.

```

## 1.3.2 Procesamiento sobre la entrada

Queremos procesar la lista INPUT-PAIS de forma que se ajuste a nuestro diseño de datos.

Para comenzar, definimos las constantes

```
; Primer registro censo
(define FECHA-INICIO 1960)
; Último registro censo
(define FECHA-FIN 2019)
; Años en los que se realizaron censos [1960,2019]
(define RANGO-FECHA (range FECHA-INICIO (+ FECHA-FIN 1) 1))
```

Luego definimos algunas funciones para estructurar los datos en Registro

```
; armar-registros: List(Number) List(String) -> List(Registro)
; armar-lista-registros: List(List(String)) List(Number) -> List(List(Registro))
```

armar-registros creará registros a partir de una lista de fechas y una lista de cantidad de habitantes y armar-lista-registros utilizará armar-registros para combinar listas de registros poblacionales con el rango de fechas en el que se registraron.

De esta forma, conseguimos organizar los registros poblacionales en estructuras Registro

```
; Datos para construir registros (sin nombres de países)
(define LISTA-DATOS-REGISTRO (map rest DATOS-PAISES))
; Lista de registros de cada país
(define LISTA-REGISTROS (armar-lista-registros LISTA-DATOS-REGISTRO RANGO-FECHA))
```

Resta combinar la información estructurada en registros con los nombres de los países. Para esto, definimos la función armar-paises

```
; armar-paises: List(String) List(Registro) -> List(Pais)
```

Finalmente, definimos LISTA-PAISES que es la constante que vamos a utilizar en las siguientes etapas del trabajo.

```
; Nombres de países
(define LISTA-NOMBRE-PAISES (map COMPLETAR DATOS-PAISES))
; Lista de países
```

```
(define LISTA-PAISES (armar-paises LISTA-NOMBRE-PAISES LISTA-REGISTROS))
```

---

## 1.4 Actualizaciones y consultas sobre el conjunto de datos

---

### 1.4.1 Funciones de alto orden para listas de países

En la plantilla se encuentran definidas 3 funciones de alto orden para usar sobre listas de estructura Pais. Verificar que se entienda qué hacen y su signatura. En caso de necesitarlo, escribir algunos casos de prueba para verificar que hacen lo que creen.

---

### 1.4.2 Parte 1 - Datos incompletos y datos incorrectos

#### Datos incompletos

Algunos países no tienen datos oficiales completos. En la base de datos esto está indicado con el valor -1. Cada celda con valor -1 significa que el país de esa fila en el año indicado por la columna no registró valores de un censo. Definir una constante LISTA-PAISES-REGISTRO-COMPLETO que contenga todos los datos en LISTA-PAISES excepto por los datos de los países con registros incompletos, en el mismo orden. Los países en el dataset cuyas filas incluyen al menos una celda con el valor -1 se consideran países con registro incompleto.

Para satisfacer este requerimiento ya se encuentra en la plantilla la definición

```
(define LISTA-PAISES-REGISTRO-COMPLETO  
  (COMPLETAR LISTA-PAISES COMPLETAR))
```

Se **deben** utilizar los patrones de alto orden definidos en la sección anterior. Para esto, definir una de las siguientes funciones auxiliares: 1) transformacion-registro-incompleto (para ser pasada como argumento a transformar-paises) 2) predicado-registro-incompleto (para pasarle a filtrar-paises) o 3) operacion-registro-incompleto (para ser pasada

como argumento a operar-sobre-paises). La función auxiliar que definas puede utilizar los patrones de alto orden generales presentados en teoría.

### Datos incorrectos

Los censos del 2014 de todos los países no se calcularon correctamente. Todos ellos tienen una diferencia del 10% por debajo de su valor real. Vamos a corregir estos valores y generar una nueva lista llamada LISTA-PAISES-RECALCULADA.

A diferencia del ejercicio anterior, la plantilla trae definida una función transformacion-recalcular. Completar todo lo que se pide al respecto.

---

### 1.4.3 Parte 2 - Países superpoblados

Ahora nos interesa generar estadísticas sobre países superpoblados. Para este trabajo, los países superpoblados son aquellos cuya población supera los mil millones de habitantes. Esto es una simplificación de la definición real.

Se pide

1. Obtener una lista de todos los países superpoblados con registros completos y toda su información poblacional
2. Obtener una lista de todos los nombres de los países del apartado anterior
3. Obtener la cantidad de países superpoblados
4. Calcular la proporción de habitantes en países superpoblados respecto a la población mundial. A esta medida la llamaremos TASA-POBLACION-SUPERPOBLADOS.

Completar la plantilla provista con los ejercicios referidos a esta sección.

---

## 2 Formato de la entrega y consideraciones generales

- El trabajo práctico debe resolverse en grupos de tres integrantes. Cada persona puede participar en un único grupo.
- Exactamente una persona por grupo debe realizar la entrega en el sitio.
- 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 (mínimo 2 casos representativos por función) y código. En caso contrario, se considerará que el TP no fue entregado.
- La entrega consiste en un único archivo por grupo. Para unificar el formato del mismo, se provee el archivo *TP1-Apellido1-Apellido2-Apellido3.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. Para verificar que no les haya quedado nada sin completar pueden realizar una búsqueda (Ctrl + F) con la cadena `#|COMPLETAR|#`.
- Cambiar el nombre del archivo reemplazando Apellido1, Apellido2 y Apellido3 por los apellidos de los integrantes, en orden alfabético ascendente.
- **No** cambiar las signaturas de las funciones provistas en el archivo.
- Se sugiere utilizar espacios y tabulaciones de forma de que el código entregado sea legible. Se valorará que sean consistentes con la elección de espacios y tabulaciones elegida.
- Se recomienda utilizar las constantes incluidas en la sección "Constantes para casos de prueba" para testear.