

# Introducción a la Algorítmica y Programación (3300)

Prof. Ariel Ferreira Szpiniak - [aferreira@exa.unrc.edu.ar](mailto:aferreira@exa.unrc.edu.ar)  
Departamento de Computación  
Facultad de Cs. Exactas, Fco-Qcas y Naturales  
Universidad Nacional de Río Cuarto

## Teoría 21

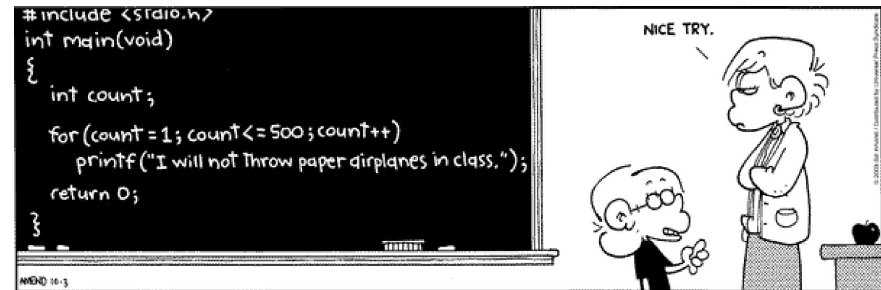
### Representación de Secuencias de marca final e inicial



2018 Lic. Ariel Ferreira Szpiniak

1

## Humor (solo para informáticos)



2018 Lic. Ariel Ferreira Szpiniak

2

### Representación de Secuencias de marca final e inicial

El concepto de secuencia se puede representar utilizando variadas alternativas según el contexto de uso.

- Arreglos
- Listas encadenadas
- Archivos



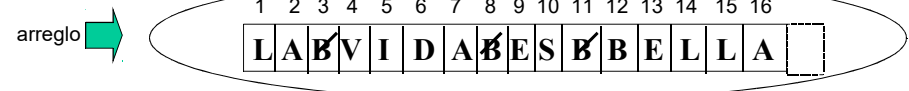
2018 Lic. Ariel Ferreira Szpiniak

3

## Secuencias y Arreglos

Los arreglos son estructuras de datos que permiten representar secuencias de marca final y de marca inicial.

**Veamos en primer término como representar el Modelo de secuencias con Marca Final con arreglos**



2018 Lic. Ariel Ferreira Szpiniak

4

# Secuencias y Arreglos

¿Cómo modelamos la marca final?

Hay varias alternativas

1. **Marca “virtual”**: llevar de alguna forma la información de cual es la longitud de la secuencia  $S + 1$ .
2. **Marca “real”**: colocar al final de la secuencia válida  $S$  un elemento del mismo tipo de los elementos de la secuencia pero normalmente no se utilice. ¡Es mucho más riesgoso!

Modelo de secuencias con Marca Final



2018 Lic. Ariel Ferreira Szpiniak

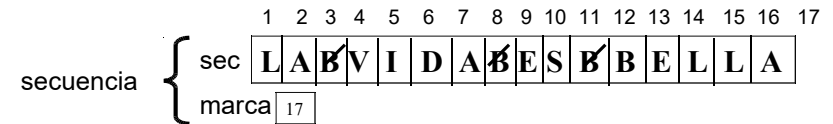
5

# Secuencias y Arreglos

¿Cómo modelamos la marca final?



**Marca “virtual”**: registro: arreglo + fin de secuencia



Pueden haber otras formas, como por ejemplo:

- Usar un registro donde, además del arreglo y el fin de secuencia (marca), incorporar también el índice permite acceder al componente donde está el elemento corriente (EC).
- Representar por separado el arreglo y el fin de secuencia (marca).



2018 Lic. Ariel Ferreira Szpiniak

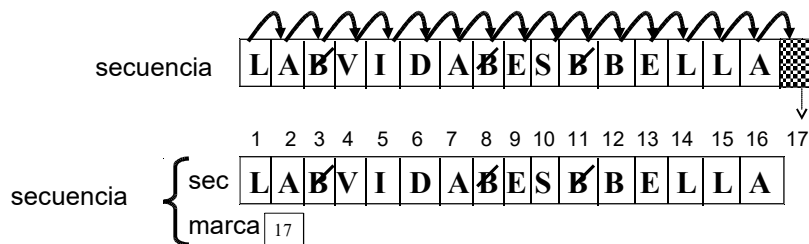
6

# Secuencias y Arreglos

¿Cómo modelamos la marca final?

**Marca “virtual”**

Modelo de secuencias con Marca Final



Long = 255

TData = <sec ∈ arreglo [1..Long] de Carácter, marca ∈ (1..Long+1)>

s ∈ TData

i ∈ (1..Long+1)

↖ O la longitud que  
necesitemos

Donde:  $i$  es igual a  $long(pi)+1$  y marca es igual a  $long(S)+1$



2018 Lic. Ariel Ferreira Szpiniak

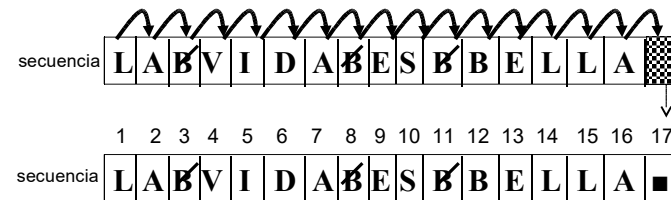
7

# Secuencias y Arreglos

¿Cómo modelamos la marca final?

**Marca “real”**

Modelo de secuencias con Marca Final



chr(254) u otro  
carácter “inusual”

@ # \* .

Si la secuencia  
fuera de números  
debería ser un  
número que no se  
use como el  
maxInt, minInt, -1  
(si la secuencia  
de es números  
positivos), etc.

Long = 254

TData = <sec ∈ arreglo [1..Long+1] de Carácter>

s ∈ TData

↖ O la longitud que necesitemos

Donde  $s.sec[cantDeCaracteres+1]$  es igual a chr(254) u otro



2018 Lic. Ariel Ferreira Szpiniak

8

# Ejemplo I Utilizando marca “virtual”

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

```
{Pre-cond: True}
{Pos-cond: lluvias.sec=lluvias.sec_0 ^ totalMilimetros=( $\sum_{j:1 \leq j < \text{lluvias.marca}}$  lluvias.sec[j])}
```

Algoritmo PrecipitAnual

Léxico

N = 365

TData = <sec ∈ arreglo [1..N] de R, marca ∈ (1..N+1)>

totalMilimetros ∈ R

i ∈ (1..N+1)

lluvias ∈ TData

Acción CargarPrecipAnual (resultado pAnual ∈ TData)

{Pre-cond: True} {Pos-cond: pAnual.sec=pAnual.sec\_0 ∧ pAnual.marca=pAnual.marca\_0}

Inicio

{Pre-cond: True}

CargarPrecipAnual(lluvias)

i ← 1

totalMilimetros ← 0

mientras i < lluvias.marca hacer

totalMilimetros ← totalMilimetros + lluvias.sec[i]

i ← i+1

fmientras

Salida: totalMilimetros

{Pos-cond: lluvias.sec=lluvias.sec\_0 ^ totalMilimetros=( $\sum_{j:1 \leq j < \text{lluvias.marca}}$  lluvias.sec[j])}

Fin



2018 Lic. Ariel Ferreira Szpiniak

9

Esquema RI  
<inicialización de la adquisición>  
<inicialización del tratamiento>  
mientras no fin de sec hacer  
<trat. elemento corriente>  
<obtener sig. elemento>  
fmientras  
<tratamiento final>

# Ejemplo I Utilizando marca “real”

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

```
{Pre-cond: True}
{Pos-cond: lluvias.sec=lluvias.sec_0 ^ totalMilimetros=( $\sum_{i:1 \leq i < w}$  Alluvias.sec[w]=-1 ∧ ( $\forall k:1 \leq k < w$ : lluvias.sec[k] ≠ -1): lluvias.sec[i])}
```

Algoritmo PrecipitAnual

Léxico

N = 365

Marca = -1 // -1 será usado como marca real

TData = <sec ∈ arreglo [1..N+1] de R>

totalMilimetros ∈ R

i ∈ (1..N+1)

lluvias ∈ TData

Acción CargarPrecipAnual (resultado pAnual ∈ TData)

{Pre-cond: True}

{Pos-cond: lluvias.sec=lluvias.sec\_0 ∧ ( $\exists j:1 \leq j \leq N+1$ : (lluvias.sec[j]=-1 ∧ ( $\forall k:1 \leq k < j$ : lluvias.sec[k] ≠ -1)))}

Inicio

CargarPrecipAnual(lluvias)

i ← 1

totalMilimetros ← 0

mientras lluvias.sec[i] ≠ Marca hacer

totalMilimetros ← totalMilimetros + lluvias.sec[i]

i ← i+1

fmientras

Salida: totalMilimetros

{Pos-cond: lluvias.sec=lluvias.sec\_0 ^ totalMilimetros=( $\sum_{i:1 \leq i < w}$  Alluvias.sec[w]=-1 ∧ ( $\forall k:1 \leq k < w$ : lluvias.sec[k] ≠ -1): lluvias.sec[i])}

Fin

Esquema RI  
<inicialización de la adquisición>  
<inicialización del tratamiento>  
mientras no fin de sec hacer  
<trat. elemento corriente>  
<obtener sig. elemento>  
fmientras  
<tratamiento final>



2018 Lic. Ariel Ferreira Szpiniak

10

## Ejemplo I bis

Solucionar el mismo problema anterior, en sus 2 versiones (marca virtual y marca real), pero utilizando el Esquema R2.



Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.



2018 Lic. Ariel Ferreira Szpiniak

11

## Ejemplo II Utilizando marca “virtual”

La maestra de sexto grado tiene un listado con el nombre de sus estudiantes y 3 notas (una por cada trimestre). La maestra necesita informar el nombre de cada estudiante y el promedio que obtuvo.

**Nota:** La cantidad de estudiantes es 45 como máximo. Las notas son números del 1 al 10.



2018 Lic. Ariel Ferreira Szpiniak

12

## Ejemplo II Utilizando marca “real”

La maestra de sexto grado tiene un listado con el nombre de sus estudiantes y 3 notas (una por cada trimestre). La maestra necesita informar el nombre de cada estudiante y el promedio que obtuvo.

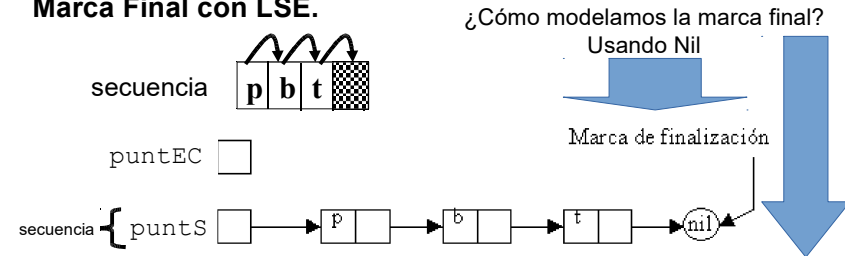
Nota: La cantidad de estudiantes es 45 como máximo. Las notas son números del 1 al 10.



## Secuencias y LSE

Las LSE son estructuras de datos que permiten representar secuencias de marca final y de marca inicial (las LDE también).

**Veamos como representar el Modelo de secuencias con Marca Final con LSE.**



También se podría usar un registro que contenga el puntero al comienzo de la lista y una marca “virtual” (cantidad de elementos +1), como hemos utilizado en arreglos. Pero al ser Nil una forma natural de indicar el fin de una lista (a ninguna parte), es decir, una marca “real”, lo usaremos como marca final.



## Secuencias y LSE

Solo veremos cómo trabajar con LSE (para LDE es igual).

Tenemos dos posibilidades:

1. Sin Usar Ficticio (desarrollaremos ésta)
2. Usando Ficticio



## Listas encadenadas Modelo de secuencias con Marca Final

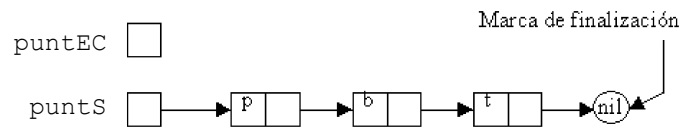
- El manejo de punteros es muy propenso a errores por parte del programador.
- Para manipular secuencias sobre listas encadenadas, el uso de los esquemas de tratamiento se hace fundamental a la hora de mitigar dichos errores, programando de manera disciplinada.
- Analizaremos a continuación la forma de implementar secuencias con marca final, y de que forma se pueden adaptar los esquemas de tratamiento. A tal efecto usaremos el Esquema R1, dejando el Esquema R2, R3, B, y RP como tarea, ya que nos aportan consideraciones adicionales.



# Listas encadenadas

## Modelo de secuencias con Marca Final

La lista simplemente encadenada debe permitir implementar una secuencia con marca de finalización. La marca de finalización la podemos representar mediante el valor **nil**, que es una dirección especial de memoria quiere decir “a ninguna parte”. Si lo comparamos con las opciones que vimos usando arreglos, **nil** sería como una marca “real” ya que está dentro de la estructura utilizada para implementar la secuencia.



2018 Lic. Ariel Ferreira Szpiniak

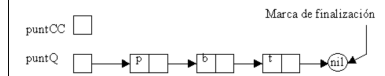
17

# Listas encadenadas

## Modelo de secuencias con Marca Final

Como mencionamos anteriormente nos centraremos en el **Esquema R1** de tratamiento de secuencia. Dicho esquema plantea la siguiente división en subproblemas:

```
<inicialización de la adquisición>
<inicialización del tratamiento>
mientras <no fin de secuencia> hacer
  <tratamiento de elemento corriente>
  <obtener el siguiente elemento>
fmientras
  <tratamiento final>
```



A continuación analizaremos soluciones generales a la condición **no fin de secuencia** y a los subproblemas **inicialización de la adquisición** y **obtener siguiente elemento**. Los subproblemas **inicialización del tratamiento**, **tratamiento del elemento corriente**, y **tratamiento final** son dependientes pura y exclusivamente del problema a resolver.



2018 Lic. Ariel Ferreira Szpiniak

18

# Listas encadenadas

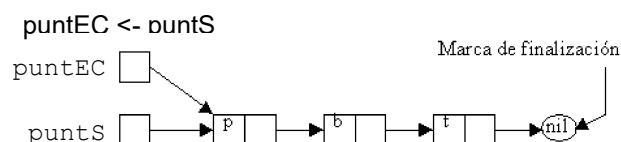
## Modelo de secuencias con Marca Final

### Obtener primer elemento

Para resolver este subproblema debemos posicionar el puntero **puntCC** en el primer doblete. Este puntero, **puntCC**, lo utilizaremos para recorrer la secuencia. Suponemos definida la estructura de la siguiente manera:

```
TNodo = <info ε Caracter, next ε puntero a TNodo>
TData = puntero a TNodo
puntEC, puntS ε TData
```

Por lo tanto, para inicializar la adquisición de los elementos debemos hacer lo siguiente:



2018 Lic. Ariel Ferreira Szpiniak

19

# Listas encadenadas

## Modelo de secuencias con Marca Final

### Obtener primer elemento

A los efectos de modularizar el esquema podríamos encapsular esta sentencia dentro de una acción que podríamos llamar **InicAdq**, es decir, arrancar con la adquisición de elementos.

**Accion** InicAdq (resultado p ε TData, dato q ε TData)

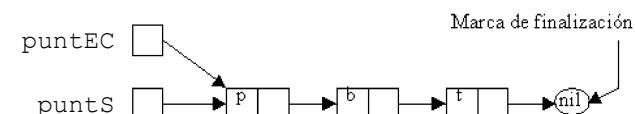
### Inicio

p <- q

### Invocación

InicAdq(puntEC, puntS)

### Fin



2018 Lic. Ariel Ferreira Szpiniak

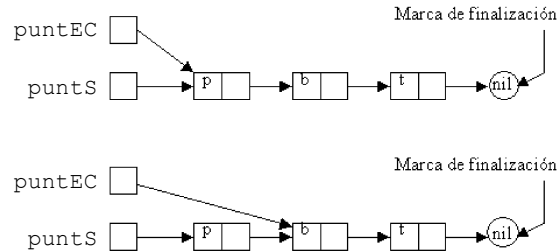
20

# Listas encadenadas

## Modelo de secuencias con Marca Final

### Obtener elemento siguiente

Para resolver este subproblema debemos mover el puntero **puntEC** desde su lugar actual al siguiente doblete. Por ejemplo, si **puntEC** apunta al primer doblete, debemos hacer que apunte al segundo, y así sucesivamente



Para lograr esto debemos hacer lo siguiente:

```
puntEC <- (^puntEC).next
```



2018 Lic. Ariel Ferreira Szpiniak

21

# Listas encadenadas

## Modelo de secuencias con Marca Final

### Obtener elemento siguiente

A los efectos de modularizar el esquema podríamos encapsular esta sentencia dentro de una acción que podríamos llamar **SigElem**, es decir, avanzar al siguiente elemento.

**Acción** SigElem (dato-resultado p  $\in$  TData)

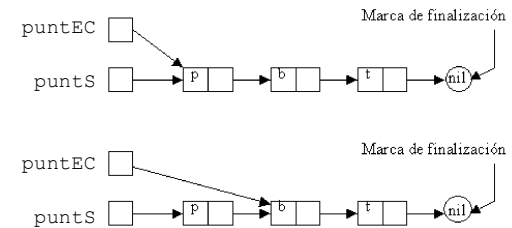
### Inicio

```
p <- (^p).next
```

### Fin

*Invocación*

SigElem(puntEC)



2018 Lic. Ariel Ferreira Szpiniak

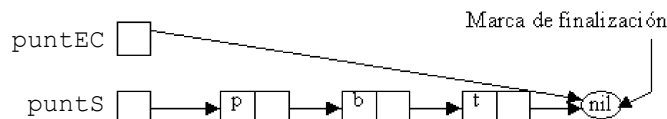
22

# Listas encadenadas

## Modelo de secuencias con Marca Final

### Condición de fin de secuencia

Esta condición debe ser verdadera cuando estemos posicionados sobre la marca de finalización, es decir, cuando **puntEC** sea igual a **nil**. Esto quiere decir que hemos tratado todos los elementos de dicha secuencia.



Para lograr esto debemos hacer lo siguiente:

```
(puntEC = nil)
```



2018 Lic. Ariel Ferreira Szpiniak

23

# Listas encadenadas

## Modelo de secuencias con Marca Final

### Condición de fin de secuencia

A los efectos de modularizar el esquema podríamos encapsular la condición de fin de secuencia en una función que podríamos llamar **FinSec**, que devuelva Verdadero cuando el puntero de recorrido halla llegado a **nil**.

**Funcion** FinSec (dato p  $\in$  TData) -> Logico

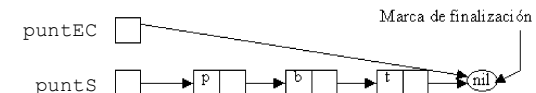
### Inicio

```
<- (p = nil)
```

### Fin

*Invocación*

**mientras** no FinSec(puntEC) **hacer**



2018 Lic. Ariel Ferreira Szpiniak

24

# Listas encadenadas

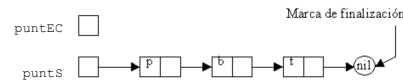
## Modelo de secuencias con Marca Final

Una vez definido cada uno de los subproblemas comunes a todos los algoritmos estamos en condiciones de enunciar el Esquema **R1**, particularizado para manipular secuencias con listas simplemente encadenadas:

```
puntEC <- puntS
<inicialización del tratamiento>
mientras puntSC≠nil hacer
  <tratamiento del elemento corriente>
  puntEC <- (^puntEC).next
fmientras
<tratamiento final>
```

### Léxico

TNodo = <info ε Caracter,  
next ε puntero a TNodo>  
TData = **puntero a** TNodo  
puntEC, puntS ε TData



*Aclaración:* suponemos que el puntero de recorrido se denomina **puntEC**, y el que apunta al primer elemento se llama **puntS**, aunque pueden tomar cualquier otro nombre.



2018 Lic. Ariel Ferreira Szpiniak

25

# Listas encadenadas

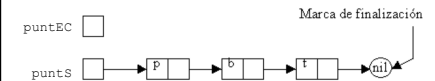
## Modelo de secuencias con Marca Final

Una vez definido cada uno de los subproblemas comunes a todos los algoritmos estamos en condiciones de enunciar el Esquema **R1**, particularizado para manipular secuencias con listas simplemente encadenadas:

```
InicAdq(puntEC, puntS)
<inicialización del tratamiento>
mientras no FinSec(puntEC) hacer
  <tratamiento del elemento corriente>
  SigElem(puntEC)
fmientras
<tratamiento final>
```

### Léxico

TNodo = <info ε Caracter,  
next ε puntero a TNodo>  
TData = **puntero a** TNodo  
puntEC, puntS ε TData



*Aclaración:* suponemos que el puntero de recorrido se denomina **puntEC**, y el que apunta al primer elemento se llama **puntS**, aunque pueden tomar cualquier otro nombre.



2018 Lic. Ariel Ferreira Szpiniak

26

## Ejemplo I Utilizando LSE sin ficticio

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

```
{Pre-cond: True}
{Pos-cond: totalMilimetros = (Σ x: x&A: x.info) }
Algoritmo PrecipitAnual
Léxico
TNodo = <info ε Z, next ε puntero a TNodo>
TData = puntero a TNodo
lluvias, aux ε TData
totalMilimetros ε R
Acción CargarPrecipAnual (resultado pAnual ε TData)
Inicio
CargarPrecipAnual(lluvias)
aux ← lluvias
totalMilimetros ← 0
mientras aux ≠ nil hacer
  totalMilimetros ← totalMilimetros + (^aux).info
  aux ← (^aux).next
fmientras
Salida: totalMilimetros
Fin
```

### Esquema R1

<inicialización de la adquisición>  
<inicialización del tratamiento>  
**mientras** no fin de sec **hacer**  
 <trat. elemento corriente>  
 <obtener sig. elemento>  
**fmientras**  
<tratamiento final>



2018 Lic. Ariel Ferreira Szpiniak

27

## Ejemplo I Utilizando LSE sin ficticio

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

```
{Pre-cond: True}
{Pos-cond: totalMilimetros = (Σ x: x&A: x.info) }
Algoritmo PrecipitAnual
Léxico
TNodo = <info ε Z, next ε puntero a TNodo>
TData = puntero a TNodo
lluvias, aux ε TData
totalMilimetros ε R
Acción CargarPrecipAnual (resultado pAnual ε TData)
Acción InicAdq(resultado p ε TData, dato q ε TData)
Acción SigElem(dato-resultado p ε TData)
Función FinSec(dato p ε TData)
Inicio
CargarPrecipAnual(lluvias)
InicAdq(aux, lluvias)
totalMilimetros ← 0
mientras no FinSec(aux) hacer
  totalMilimetros ← totalMilimetros + (^aux).info
  SigElem(aux)
fmientras
Salida: totalMilimetros
Fin
```

### Esquema R1

<inicialización de la adquisición>  
<inicialización del tratamiento>  
**mientras** no fin de sec **hacer**  
 <trat. elemento corriente>  
 <obtener sig. elemento>  
**fmientras**  
<tratamiento final>



2018 Lic. Ariel Ferreira Szpiniak

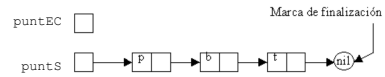
28

# Listas encadenadas

## Modelo de secuencias con Marca Final

Esquema R2 de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

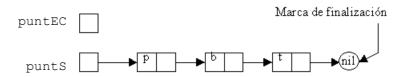
29

# Listas encadenadas

## Modelo de secuencias con Marca Final

Esquema R3 de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

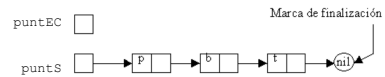
30

# Listas encadenadas

## Modelo de secuencias con Marca Final

Esquema B de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

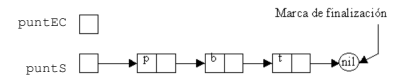
31

# Listas encadenadas

## Modelo de secuencias con Marca Final

Esquema RP de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

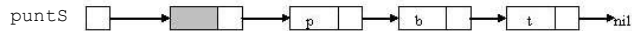
32



# Listas encadenadas

## Modelo de secuencias con Marca Final

¿Qué cambia si usamos ficticio?



```
puntEC <- (^puntS).next
<inicialización del tratamiento>
mientras puntEC≠nil hacer
  <tratamiento del elemento corriente>
  puntEC <- (^puntEC).next
fmientras
<tratamiento final>
```

### Léxico

TNodo = <info ε Caracter,  
next ε puntero a TNodo>  
TData = **puntero a** TNodo  
puntEC, puntS ε TData



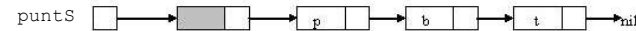
2018 Lic. Ariel Ferreira Szpiniak

33

# Listas encadenadas

## Modelo de secuencias con Marca Final

¿Qué cambia si usamos ficticio?



```
InicAdq(puntEC, puntS)
<inicialización del tratamiento>
mientras no FinSec(puntEC) hacer
  <tratamiento del elemento corriente>
  SigElem(puntEC)
fmientras
<tratamiento final>
```

### Léxico

TNodo = <info ε Caracter,  
next ε puntero a TNodo>  
TData = **puntero a** TNodo  
puntEC, puntS ε TData

**Accion** InicAdq (resultado p ε TData, dato q ε TData)

### Inicio

p <- (^q).next

### Fin



2018 Lic. Ariel Ferreira Szpiniak

34

## Ejemplo I utilizando LSE con ficticio

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

```
{Pre-cond: True}
{Pos-cond: totalMilímetros=(Σx: x&A: x.info)}
```

**Algoritmo** PrecipitAnual

### Léxico

TNodo = <info ε Caracter, next ε puntero a TNodo>  
TData= puntero a TNodo  
lluvias, aux ε TData  
totalMilímetros ε R

**Acción** CargarPrecipAnual (resultado pAnual ε TData)

### Inicio

```
CargarPrecipAnual (lluvias)
aux <- (^lluvias).next
totalMilímetros <- 0
```

**mientras** aux ≠ nil **hacer**

```
  totalMilímetros <- totalMilímetros + (^aux).info
```

```
  aux <- (^aux).next
```

### fmientras

Salida: totalMilímetros

### Fin

### Esquema R1

```
<inicialización de la adquisición>
<inicialización del tratamiento>
mientras no fin de sec hacer
  <trat. elemento corriente>
  <obtener sig. elemento>
fmientras
<tratamiento final>
```



2018 Lic. Ariel Ferreira Szpiniak

35

## Ejemplo I utilizando LSE con ficticio

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

```
{Pre-cond: True}
```

```
{Pos-cond: totalMilímetros=(Σx: x&A: x.info)}
```

**Algoritmo** PrecipitAnual

### Léxico

TNodo = <info ε Z, next ε puntero a TNodo>  
TData= puntero a TNodo  
lluvias, aux ε TData  
totalMilímetros ε R

**Acción** CargarPrecipAnual (resultado pAnual ε TData)

**Acción** InicAdq(resultado p ε TData, dato q ε TData)

**Acción** SigElem(dato-resultado p ε TData)

**Función** FinSec(dato p ε TData)

### Inicio

```
CargarPrecipAnual (lluvias)
InicAdq(aux, lluvias)
totalMilímetros <- 0
```

**mientras** no FinSec(aux) **hacer**

```
  totalMilímetros <- totalMilímetros + (^aux).info
```

```
  SigElem(aux)
```

### fmientras

Salida: totalMilímetros

### Fin

```
Accion InicAdq (resultado pεTData,dato qεTData)
Inicio
  p <- (^q).next
Fin
```

### Esquema R1

```
<inicialización de la adquisición>
<inicialización del tratamiento>
mientras no fin de sec hacer
  <trat. elemento corriente>
  <obtener sig. elemento>
fmientras
<tratamiento final>
```



2018 Lic. Ariel Ferreira Szpiniak

36

# Secuencias de Marca Inicial

Analizaremos a continuación la forma de representar secuencias con marca inicial usando arreglos, listas encadenadas y archivos. Veremos además que forma toman los esquemas generales para el tratamiento de secuencias con marca inicial sobre arreglos, y listas encadenadas y archivos.

## Modelo secuencial con Marca Inicial



También es llamado “Modelo secuencial sin marca de finalización”  
o “Modelo secuencial de último elemento”



2018 Lic. Ariel Ferreira Szpiniak

37

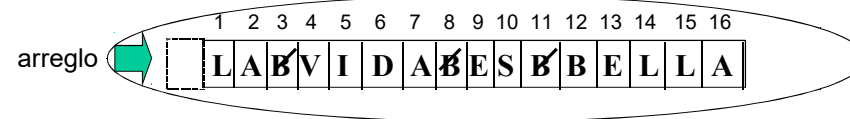
# Secuencias y Arreglos

Los arreglos son estructuras de datos estáticas que permiten representar secuencias bajo los 2 modelos. Ahora analizaremos el Modelo de Marca Inicial.

## Modelo secuencial con Marca Inicial



¿Hace falta destinar un lugar especial para la marca inicial?



2018 Lic. Ariel Ferreira Szpiniak

38

# Secuencias y Arreglos

¿Cómo modelamos la marca inicial?

Hay varias alternativas

1. **Marca “virtual”**: llevar de alguna forma la información de cual es la longitud de la secuencia S para saber la posición del último elemento.
2. **Marca “real”**: colocar al comienzo de la secuencia válida S un elemento del mismo tipo de los elementos de la secuencia pero que normalmente no se utilice.  
**¡Es mucho más riesgoso!**

## Modelo secuencial con Marca Inicial



2018 Lic. Ariel Ferreira Szpiniak

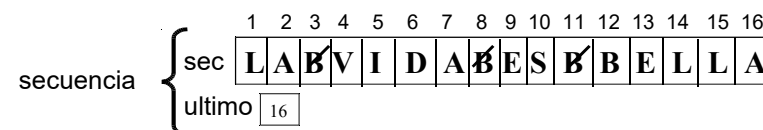
39

# Secuencias y Arreglos

¿Cómo modelamos la marca inicial?



## Marca “virtual”: registro: arreglo + fin de secuencia



Pueden haber otras formas, como por ejemplo:

- Usar un registro donde, además del arreglo y el fin de secuencia (posición del último elemento de la secuencia), incorporar también el índice permite acceder al componente donde está el elemento corriente (EC).
- Representar por separado el arreglo y la cantidad de elementos.



2018 Lic. Ariel Ferreira Szpiniak

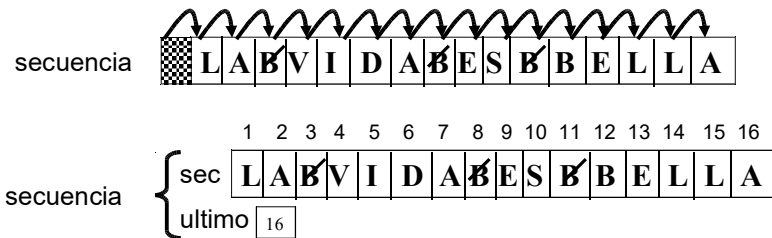
40

# Secuencias y Arreglos

¿Cómo modelamos la marca inicial?

Marca “virtual”

Modelo de secuencias con Marca Inicial



Long = 255  
TData = <sec ∈ arreglo [1..Long] de Carácter, ultimo ∈ (0..Long)>  
s ∈ TData  
i ∈ (0..Long)  
Donde: i es igual a long(pi) y último elemento es igual a long(S)  
O la longitud que necesitamos



2018 Lic. Ariel Ferreira Szpiniak

41

# Secuencias y Arreglos

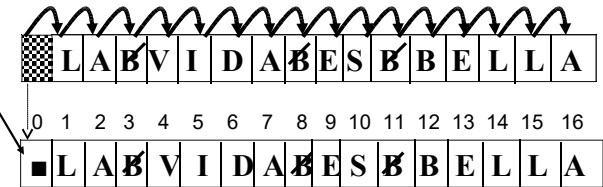
¿Cómo modelamos la marca inicial?

Marca “real”

Modelo secuencial con Marca Inicial

chr(254) u otro carácter  
“inusual” @ # \* .

Si la secuencia fuera de números debería ser un número que no se use como el maxInt, minInt, -1 (si la secuencia de es números positivos), etc.



Long = 254  
TData = <sec ∈ arreglo [0..Long] de Carácter, ultimo ∈ (0..Long)>  
s ∈ TData  
i ∈ (0..Long)  
Donde s[0] es igual a chr(254) u otro  
O la longitud que necesitamos



2018 Lic. Ariel Ferreira Szpiniak

42

## Ejemplo I Utilizando marca “virtual”

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

{Pre-cond: True}  
{Pos-cond: lluvias.sec=lluvias.sec, totalMilímetros=( $\sum_{j:1 \leq j \leq \text{lluvias.ultimo}}$  lluvias.sec[j])}

Algoritmo PrecipitAnual

Léxico

N = 365

TData = <sec ∈ arreglo [1..N] de R, ultimo ∈ (0..N)>

lluvias ∈ TData

i ∈ (0..N)

totalMilímetros ∈ R

Acción CargarPrecipAnual (resultado pAnual ∈ Tdata) //hacer acción CargarPrecipAnual

{Pre-cond: True} {Pos-cond: pAnual.sec=pAnual.sec, pAnual.ultimo=long(pAnual.sec)}

Inicio // inicio del algoritmo principal

{Pre-cond: True}

CargarPrecipAnual(lluvias)

i ← 0

totalMilímetros ← 0

mientras i < lluvias.ultimo hacer

i ← i+1

totalMilímetros ← totalMilímetros + lluvias.sec[i]

fmientras

Salida: totalMilímetros

{Pos-cond: lluvias.sec=lluvias.sec, totalMilímetros=( $\sum_{j:1 \leq j \leq \text{lluvias.ultimo}}$  lluvias.sec[j])}

Fin // fin del algoritmo principal

i se inicializa en 0  
(marca inicial, virtual)

Esquema RI

<inicialización de la adquisición>

<inicialización del tratamiento>

mientras no fin de sec hacer

<obtener sig. elemento>

<trat. elemento corriente>

fmientras

<tratamiento final>



2018 Lic. Ariel Ferreira Szpiniak

43

## Ejemplo I Utilizando marca “real”

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

{Pre-cond: True}  
{Pos-cond: lluvias.sec=lluvias.sec, totalMilímetros=( $\sum_{j:1 \leq j \leq \text{lluvias.ultimo}}$  lluvias.sec[j])}

Algoritmo PrecipitAnual

Léxico

N = 365

TData = <sec ∈ arreglo [0..N] de R, ultimo ∈ (0..N)>

lluvias ∈ TData

i ∈ (0..N)

totalMilímetros ∈ R

Acción CargarPrecipAnual (resultado pAnual ∈ TData)

{Pre-cond: True} //hacer acción CargarPrecipAnual

{Pos-cond: pAnual.sec=pAnual.sec, pAnual.ultimo=long(pAnual.sec) ∧ pAnual.sec[0]=-1}

Inicio // inicio del algoritmo principal

{Pre-cond: True}

CargarPrecipAnual(lluvias)

i ← 0

totalMilímetros ← 0

mientras i < lluvias.ultimo hacer

i ← i+1

totalMilímetros ← totalMilímetros + lluvias.sec[i]

fmientras

Salida: totalMilímetros

Fin // fin del algoritmo principal

Esquema RI

<inicialización de la adquisición>

<inicialización del tratamiento>

mientras no fin de sec hacer

<obtener sig. elemento>

<trat. elemento corriente>

fmientras

<tratamiento final>

Marca inicial, real (-1).  
La acción debe colocar la  
marca (-1) en el primer  
componente del arreglo



2018 Lic. Ariel Ferreira Szpiniak

44

## Ejemplo I bis

Solucionar el mismo problema anterior, en sus 2 versiones (marca virtual y marca real), pero utilizando el Esquema R2.



Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.



2018 Lic. Ariel Ferreira Szpiniak

45

## Ejemplo II Utilizando marca “virtual”

La maestra de sexto grado tiene un listado con el nombre de sus estudiantes y 3 notas (una por cada trimestre). La maestra necesita informar el nombre de cada estudiante y el promedio que obtuvo.

**Nota:** La cantidad de estudiantes es 45 como máximo. Las notas son números del 1 al 10.



2018 Lic. Ariel Ferreira Szpiniak

46

## Ejemplo II Utilizando marca “real”

La maestra de sexto grado tiene un listado con el nombre de sus estudiantes y 3 notas (una por cada trimestre). La maestra necesita informar el nombre de cada estudiante y el promedio que obtuvo.

**Nota:** La cantidad de estudiantes es 45 como máximo. Las notas son números del 1 al 10.



2018 Lic. Ariel Ferreira Szpiniak

47

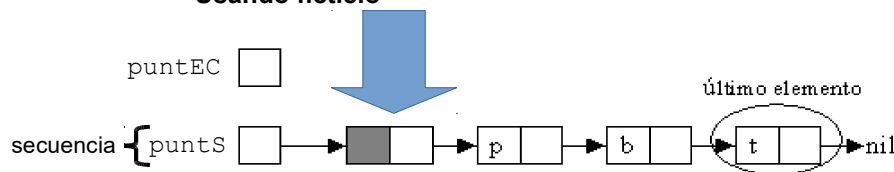
## Secuencias y LSE

Las LSE son estructuras de datos que permiten representar secuencias de marca inicial (las LDE también).

**Veamos como representar el Modelo de secuencias con Marca Inicial con LSE.**



¿Cómo modelamos la marca inicial?  
Usando ficticio



2018 Lic. Ariel Ferreira Szpiniak

48

# Secuencias y LSE

Solo veremos cómo trabajar con LSE (para LDE es igual).

Tenemos una sola posibilidad:

- ~~Sin Usar Ficticio~~
- Usando Ficticio

¿Porqué?



2018 Lic. Ariel Ferreira Szpiniak

49

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

- El manejo de punteros es muy propenso a errores por parte del programador.
- Para manipular secuencias sobre listas encadenadas, el uso de los esquemas de tratamiento se hace fundamental a la hora de mitigar dichos errores, programando de manera disciplinada.
- Analizaremos a continuación la forma de implementar secuencias con marca inicial, y de que forma se pueden adaptar los esquemas de tratamiento. A tal efecto usaremos el Esquema R1, dejando el Esquema R2, R3, B, y RP como tarea, ya que nos aportan consideraciones adicionales.



2018 Lic. Ariel Ferreira Szpiniak

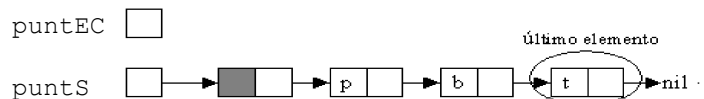
50

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

La lista simplemente encadenada debe permitir implementar una secuencia con marca inicial. La marca inicial la podemos representar mediante un **elemento ficticio**. Si lo comparamos con las opciones que vimos usando arreglos, el **elemento ficticio** sería como una marca “real” ya que está dentro de la estructura utilizada para implementar la secuencia.

Para saber que hemos finalizado el tratamiento debemos detectar cuando estamos apuntando al último elemento.



2018 Lic. Ariel Ferreira Szpiniak

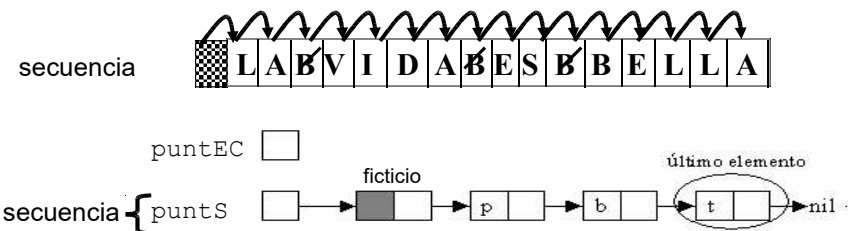
51

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Las listas encadenadas permiten representar secuencias bajo los 2 modelos. Ya hemos visto el Modelo de marca final, ahora analizaremos el Modelo de marca inicial.

### Modelo secuencial con Marca Inicial



TNodo = <info ε Caracter, next ε puntero a TNodo>  
TData = puntero a TNodo  
punteC, punteS ε TData



2018 Lic. Ariel Ferreira Szpiniak

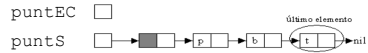
52

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Al igual que en el Modelo de Marca Final, a continuación nos centraremos en el **Esquema R1** de tratamiento de secuencia. Dicho esquema plantea la siguiente división en subproblemas:

```
<inicialización de la adquisición>
<inicialización del tratamiento>
mientras <no fin de secuencia> hacer
  <obtener el siguiente elemento>
  <tratamiento de elemento corriente>
fmientras
<tratamiento final>
```



A continuación analizaremos soluciones generales a la condición **no fin de secuencia** y a los subproblemas **inicialización de la adquisición** y **obtener siguiente elemento**. Los subproblemas *inicialización del tratamiento*, *tratamiento del elemento corriente*, y *tratamiento final* son dependientes pura y exclusivamente del problema a resolver.



2018 Lic. Ariel Ferreira Szpiniak

53

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

### Inicialización de la adquisición

- En secuencias de marca inicial no es posible obtener el primer elemento de la secuencia sin corroborar previamente que la secuencia no esté vacía.
- De allí que en lugar de obtener el primer elemento se realiza la **inicialización de la adquisición** de elementos, que consiste en dejar la secuencia en condiciones de ser recorrida (en la marca inicial).
- En caso de LSE esto significa que el puntero utilizado para recorrer la misma, **puntEC**, no debe quedar apuntando al doblete que contiene el primer elemento válido de secuencia, pero si preparar el terreno para que esto pueda hacerse previo chequeo de que la secuencia no está vacía.
- Por ello una de las mejores, y más simples, alternativas es utilizar un **elemento ficticio** al comienzo de la lista, que preceda al primer elemento válido de la secuencia. De ésta manera se deja apuntando el puntero de recorrido **puntEC** sobre éste elemento (marca inicial).



2018 Lic. Ariel Ferreira Szpiniak

54

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

### Inicialización de la adquisición

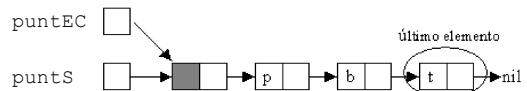
Entonces, para resolver este subproblema debemos posicionar el puntero **puntEC** en el elemento ficticio, que vendría a ser el primer doblete de la lista, pero no el primer elemento de la secuencia. Este puntero, **puntEC**, lo utilizaremos para recorrer la secuencia.

Suponemos definida la estructura de la siguiente manera:

```
TNodo = <info ε Caracter, next ε puntero a TNodo>
TData = puntero a TNodo
puntEC, puntS ε TData
```

Por lo tanto, para inicializar la adquisición de los elementos debemos hacer lo siguiente:

```
puntEC <- puntS
```



2018 Lic. Ariel Ferreira Szpiniak

55

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

### Inicialización de la adquisición

A los efectos de modularizar el esquema podríamos encapsular esta sentencia dentro de una acción que podríamos llamar **InicAdq**, es decir, arrancar con la adquisición de elementos.

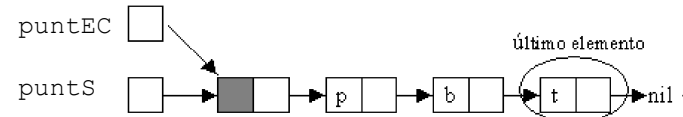
**Accion** InicAdq (resultado p ε puntero a TDobleteCar, dato q ε puntero a TDobleteCar)

**Invocación**  
InicAdq(puntEC, puntS)

### Inicio

```
p <- a
```

### Fin



2018 Lic. Ariel Ferreira Szpiniak

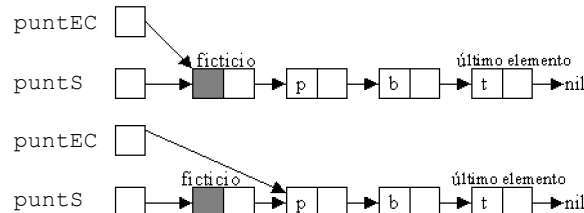
56

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

### Obtener elemento siguiente

Para resolver este subproblema debemos mover el puntero **puntEC** desde su lugar actual al siguiente doblete. Por ejemplo, si **puntEC** apunta al primer doblete, debemos hacer que apunte al segundo, y así sucesivamente.



Para lograr esto debemos hacer lo siguiente:

```
puntEC <- (^puntEC).next
```



2018 Lic. Ariel Ferreira Szpiniak

57

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

### Obtener elemento siguiente

A los efectos de modularizar el esquema podríamos encapsular esta sentencia dentro de una acción que podríamos llamar **SigElem**, es decir, avanzar al siguiente elemento.

**Accion** SigElem (dato-resultado p  $\in$  puntero a TDobleteCar)

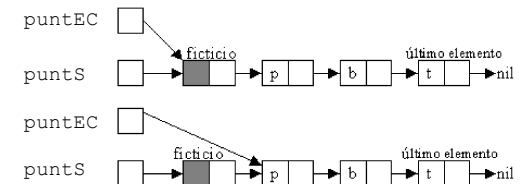
### Inicio

```
p <- (^p).next
```

### Fin

*Invocación*

**sigElem(puntEC)**



2018 Lic. Ariel Ferreira Szpiniak

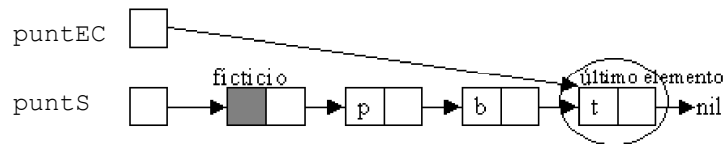
58

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

### Condición de no fin de secuencia

Esta condición debe ser verdadera cuando estemos posicionados sobre el último doblete, es decir, cuando **(^puntEC).next** sea igual a **nil**. Esto quiere decir que hemos tratado todos los elementos de dicha secuencia.



Para lograr esto debemos hacer lo siguiente:

```
((^puntEC).next = nil)
```



2018 Lic. Ariel Ferreira Szpiniak

59

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

### Condición de no fin de secuencia

A los efectos de modularizar el esquema podríamos encapsular la condición de fin de secuencia en una función que podríamos llamar **Ultimo**, que devuelva Verdadero cuando el puntero de recorrido en el campo next halla llegado a nil.

**Funcion** Ultimo (dato p  $\in$  puntero a TDobleteCar) -> Logico

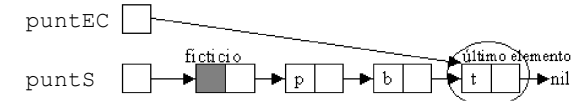
### Inicio

```
<- ((^p).next = nil)
```

### Fin

*Invocación*

**mientras** no Ultimo(puntEC) **hacer**



2018 Lic. Ariel Ferreira Szpiniak

60



# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Una vez definido cada uno de los subproblemas comunes a todos los algoritmos estamos en condiciones de enunciar el Esquema **R1**, particularizado para manipular secuencias con listas simplemente encadenadas:

```
puntEC <- puntS
<inicialización del tratamiento>
mientras (^puntEC).next ≠ nil hacer
  puntEC <- (^puntEC).next
  <tratamiento del elemento corriente>
fmientras
<tratamiento final>
```

### Léxico

TNodo = <info ∈ Caracter, next ∈ puntero a TNodo>  
TData = puntero a TNodo  
puntEC, puntS ∈ TData

*Aclaración:* suponemos que el puntero de recorrido se denomina **puntEC**, y el que apunta al primer elemento se llama **puntS**, aunque pueden tomar cualquier otro nombre.



2018 Lic. Ariel Ferreira Szpiniak

61

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Una vez definido cada uno de los subproblemas comunes a todos los algoritmos estamos en condiciones de enunciar el Esquema **R1**, particularizado para manipular secuencias con listas simplemente encadenadas:

```
InicAdqArr(puntEC, puntS)
<inicialización del tratamiento>
mientras no Ultimo(puntEC) hacer
  SigElem(puntEC)
  <tratamiento del elemento corriente>
fmientras
<tratamiento final>
```

### Léxico

TNodo = <info ∈ Caracter, next ∈ puntero a TNodo>  
TData = puntero a TNodo  
puntEC, puntS ∈ TData

*Aclaración:* suponemos que el puntero de recorrido se denomina **puntEC**, y el que apunta al primer elemento se llama **puntS**, aunque pueden tomar cualquier otro nombre.



2018 Lic. Ariel Ferreira Szpiniak

62

## Ejemplo I Utilizando LSE (ficticio)

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

{Pre-cond: True}

{Pos-cond: totalMilímetros = ( $\sum x: x \& a: x.info$ ) }

Algoritmo PrecipitAnual

### Léxico

TNodo = <info ∈ Caracter, next ∈ puntero a TNodo>

TData = puntero a TNodo

lluvias, aux ∈ TData

totalMilímetros ∈ R

Acción CargarPrecipAnual (resultado pAnual ∈ Tdata)

//hacer acción CargarPrecipAnual (poner ficticio al comienzo)

Inicio // inicio del algoritmo principal

CargarPrecipAnual(lluvias)

aux ← lluvias

totalMilímetros ← 0

**mientras** (^aux).next ≠ nil **hacer**

aux ← (^aux).next

totalMilímetros ← totalMilímetros + (^aux).info

**fmientras**

Salida: totalMilímetros

Fin // fin del algoritmo principal



2018 Lic. Ariel Ferreira Szpiniak

63

### Esquema R1

<inicialización de la adquisición>  
<inicialización del tratamiento>  
**mientras** no fin de sec **hacer**  
 <obtener sig. elemento>  
 <trat. elemento corriente>  
**fmientras**  
<tratamiento final>

## Ejemplo I Utilizando LSE (ficticio)

Una Estación Meteorológica registra diariamente la cantidad de milímetros llovidos en Río Cuarto desde comienzo de año hasta la fecha, es decir, hasta el día de hoy (secuencia de datos). Necesita calcular e informar el total de milímetros llovidos en Río Cuarto hasta la fecha.

{Pre-cond: True}

{Pos-cond: totalMilímetros = ( $\sum x: x \& a: x.info$ ) }

Algoritmo PrecipitAnual

### Léxico

TNodo = <info ∈ Caracter, next ∈ puntero a TNodo>

TData = puntero a TNodo

lluvias, aux ∈ TData

totalMilímetros ∈ R

Acción CargarPrecipAnual (resultado pAnual ∈ Tdata)

//hacer acción CargarPrecipAnual (poner ficticio al comienzo)

Acción InicAdq(resultado p ∈ TData, dato q ∈ TData)

Acción SigElem(dato=resultado p ∈ TData)

Función Ultimo(dato p ∈ TData)

Inicio // inicio del algoritmo principal

CargarPrecipAnual(lluvias)

InicAdq(aux, lluvias)

totalMilímetros ← 0

**mientras** no Ultimo(aux) **hacer**

SigElem(aux)

totalMilímetros ← totalMilímetros + (^aux).info

**fmientras**

Salida: totalMilímetros

Fin // fin del algoritmo principal



2018 Lic. Ariel Ferreira Szpiniak

64

### Esquema R1

<inicialización de la adquisición>  
<inicialización del tratamiento>  
**mientras** no fin de sec **hacer**  
 <obtener sig. elemento>  
 <trat. elemento corriente>  
**fmientras**  
<tratamiento final>

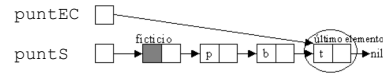


# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Esquema R2 de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

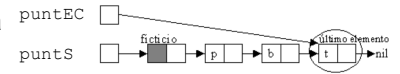
65

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Esquema R3 de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

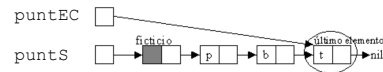
66

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Esquema B de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

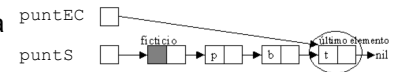
67

# Listas encadenadas

## Modelo de secuencias con Marca Inicial

Esquema RP de tratamiento de secuencia

Completar...



2018 Lic. Ariel Ferreira Szpiniak

68

# Secuencias y Archivos

- Los archivos permiten representar secuencias (y otras cosas también).
- Dependiendo de la clase de archivo que se trate, o de las primitivas existentes para manipularlo, podrá utilizarse el Modelo de marca final, inicial, o ambos.
- Pero, en general, cada lenguaje permite manipular los archivos usando ciertas primitivas predefinidas, como en **C** (fopen, feof, fread, fwrite, etc).



2018 Lic. Ariel Ferreira Szpiniak

69

# Secuencias y Archivos

## Archivos de texto. Modelo de marca inicial

Abrir	fopen
Leer	fgets, fgetc
EOF (fin de archivo)	feof
Cerrar	fclose

Mediante el uso repetido de la primitiva **Leer** (**fgets**, **fgetc**) podemos ir extrayendo la información de un archivo.

Esta primitiva, junto a **Abrir** (**fopen**) y el **EOF** (**feof**) y/o **EOLN** (no existe en C), se adapta al **Modelo de marca inicial**.



2018 Lic. Ariel Ferreira Szpiniak

70

# Secuencias y Archivos

Ejemplo de lectura de datos de un archivo de texto, línea por línea, usando Leer

Algoritmo LeerArchivoDeTexto

Lexico

$f \in \text{ARCHIVO de Texto}$

$\text{cad} \in \text{Cadena}$

Inicio

Abrir ("test.txt", f, l)

mientras not(EOF(f)) hacer

Leer (f, cad)

Salida: cad

fmientras

Cerrar (f)

Fin

**Esquema RI**  
<inicialización de la adquisición>  
<inicialización del tratamiento>  
mientras no fin de sec hacer  
<obtener sig. elemento>  
<trat. elemento corriente>  
fmientras  
<tratamiento final>



2018 Lic. Ariel Ferreira Szpiniak

71

# Secuencias y Archivos

Ejemplo de lectura de datos de un archivo de texto, línea por línea, usando gets

```
int main() {
    char cadena[128];
    FILE* archivo;
    archivo = fopen("test.txt", "r");
    if (archivo != NULL) {
        while (!feof(archivo)) {
            fgets(cadena, 128, archivo);
            printf("%s", cadena);
        }
        fclose(archivo);
    }
}
```

**Esquema RI**  
<inicialización de la adquisición>  
<inicialización del tratamiento>  
mientras no fin de sec hacer  
<obtener sig. elemento>  
<trat. elemento corriente>  
fmientras  
<tratamiento final>



2018 Lic. Ariel Ferreira Szpiniak

72

# Secuencias y Archivos

Ejemplo de lectura de datos de un archivo de texto, carácter por carácter, usando Leer

Algoritmo LeerArchivoDeTexto2

Lexico

f ∈ ARCHIVO de Texto

car ∈ Caracter

Inicio

Abrir("test.txt", f, l)

mientras not(EOF(f)) hacer

mientras not(EOL(f)) hacer

Leer(f, car)

Salida: car

fmientras

fmientras

Cerrar(f)

Fin



2018 Lic. Ariel Ferreira Szpiniak

73

**Esquema RI**  
<inicialización de la adquisición>  
<inicialización del tratamiento>  
mientras no fin de sec hacer  
<obtener sig. elemento>  
<trat. elemento corriente>  
fmientras  
<tratamiento final>

# Secuencias y Archivos

Ejemplo de lectura de datos de un archivo de texto, carácter por carácter, usando fgetc

```
int main() {
    char car;
    FILE* archivo;
    archivo = fopen("test.txt", "r");
    if (archivo != NULL) {
        while (!feof(archivo)) {
            car = fgetc(archivo);
            printf("%c", car);
        }
        fclose(archivo);
    }
}
```



2018 Lic. Ariel Ferreira Szpiniak

74

**Esquema RI**  
<inicialización de la adquisición>  
<inicialización del tratamiento>  
mientras no fin de sec hacer  
<obtener sig. elemento>  
<trat. elemento corriente>  
fmientras  
<tratamiento final>

# Secuencias y Archivos

Archivos binarios. Modelo de marca inicial

Abrir fopen

Leer fread

EOF (fin de archivo) feof

Cerrar fclose

Mediante el uso repetido de la primitiva Leer (fread) podemos ir extrayendo la información de un archivo.

Esta primitiva, junto a Abrir (fopen) y el EOF (feof), se adapta al Modelo de marca inicial.



2018 Lic. Ariel Ferreira Szpiniak

75

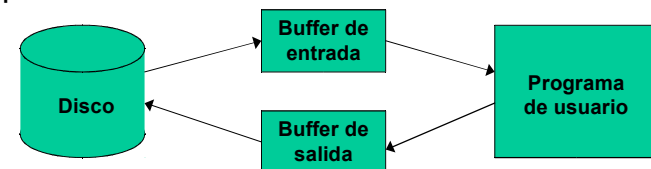
# Secuencias y Archivos

## Buffers

### Concepto

Posición intermedia entre un archivo y un programa, donde los datos residen provisoriamente hasta ser almacenados en memoria secundaria.

Ocupan una zona de la RAM.



2018 Lic. Ariel Ferreira Szpiniak

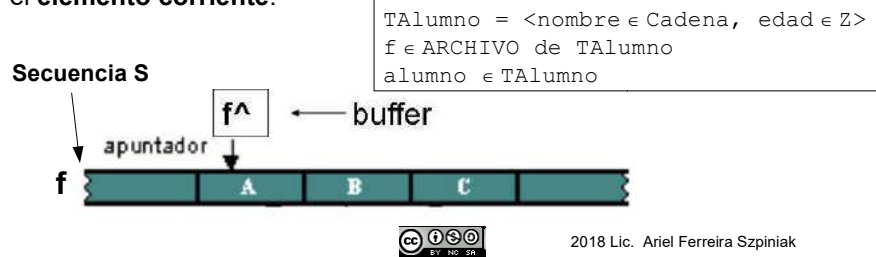
76

# Secuencias y Archivos

Las operaciones lectura y escritura se definen con el auxilio de una variable implícita que representa una memoria intermedia de amortiguación llamada "**buffer**".

Cuando se declara una variable de tipo ARCHIVO (FILE \*), se crea automáticamente una variable "**buffer**" del mismo tipo que el componente. El "**buffer**" no puede ser manipulado directamente sino a través de las operaciones de lectura y escritura.

Para una variable "**f**" de tipo ARCHIVO de TAlumno, el "**buffer**" se designa por "**f^**". El tipo de "**f^**" es de tipo TAlumno. Es decir que "**f**" sería similar a la secuencia **S** y el **buffer** el lugar desde donde se obtiene el **elemento corriente**.



# Secuencias y Archivos

## Archivos binarios. Modelo de marca inicial

### Algoritmo

Abrir("alumnos.dat", f, l)

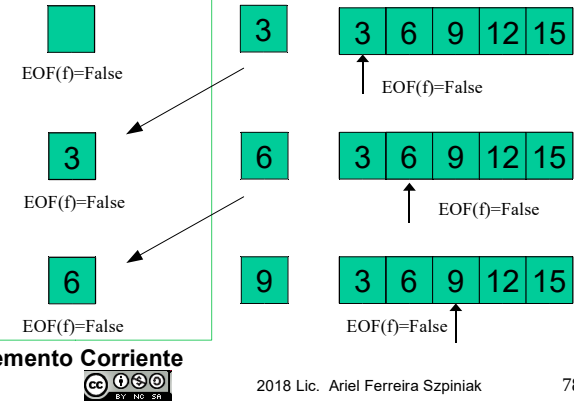
Leer(f, alumno)

Leer(f, alumno)

### Variable (alumno)

### Buffer(f^)

### Archivo



# Secuencias y Archivos

## Archivos binarios. Modelo de marca inicial

### Algoritmo

Leer(f, alumno)

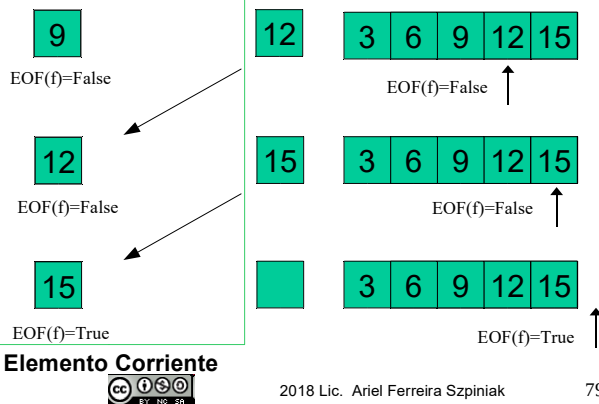
Leer(f, alumno)

Leer(f, alumno)

### Variable (alumno)

### Buffer(f^)

### Archivo



## Acceso secuencial – archivos binarios

### Lectura de datos

### Algoritmo LecturaDeArchivoBinario

### Lexico

**TAlumno** = <nombre ∈ Cadena, edad ∈ Z>

**f** ∈ ARCHIVO de TAlumno

**alumno** ∈ TAlumno

### Inicio

**Abrir** ("alumnos.dat", f, l)

**mientras** not (**EOF**(f)) **hacer**

**Leer**(f, alumno)

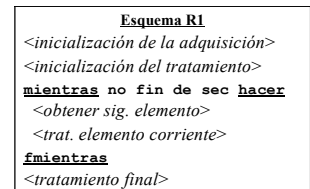
Salida: alumno

// Salida: alumno.nombre alumno.edad

**fmientras**

**Cerrar**(f)

### Fin



# Acceso secuencial – archivos binarios

## Lectura de datos

```
typedef struct { char nombre[30]; int edad;} TAlumno;
TAlumno alumno;
void main() {
    FILE *f;
    f = fopen("alumnos.dat","r");
    while (!feof(f)) {
        fread(&alumno,sizeof(alumno),1,f);
        printf(" NOMBRE = %s ",alumno.nombre);
        printf(" EDAD = %d ",alumno.edad);
        printf("\n");
    };
    fclose(f);
}
```

**Esquema R1**  
<inicialización de la adquisición>  
<inicialización del tratamiento>  
mientras no fin de sec hacer  
<obtener sig. elemento>  
<trat. elemento corriente>  
fmientras  
<tratamiento final>



2018 Lic. Ariel Ferreira Szpiniak

81

# Acceso secuencial - Tratamiento

De acuerdo a las primitivas analizadas podemos decir que los archivos de texto y binarios (cuando se los accede de manera secuencial) se pueden adaptar perfectamente al modelo de **secuencias con marca inicial**, y por lo tanto al tratamiento con los **esquemas correspondientes**.

Ello ocurre tanto en Notación Algorítmica como en lenguaje C.



2018 Lic. Ariel Ferreira Szpiniak

82

## Secuencias y Archivos Resumen

### Archivos de texto (.txt)

#### • Modelo de marca inicial:

Abrir, Leer, EOF (y/o EOLN), Cerrar  
fopen, fgetc, fgets, feof, fclose

### Archivos binarios (.dat)

#### • Modelo de marca inicial:

Abrir, Leer, EOF, Cerrar  
fopen, fread, feof, fclose



2018 Lic. Ariel Ferreira Szpiniak

83

## Bibliografía

- Scholl, P. y J.-P. Peyrin, “Esquemas Algorítmicos Fundamentales: Secuencias e iteración”, Barcelona, Ed. Masson, 1991.
- Lucas, M., J.-P. Peyrin y P. Scholl, “Algorítmica y Representación de Datos. Tomo 1: Secuencia, Autómata de estados finitos”, Barcelona, Ed. Masson, 1985.
- Watt, David, “Programming Language Concepts and Paradigms”, Prentice-Hall International Series in Computer Science (1990).
- Biondi, J. y G. Clavel, “Introducción a la Programación. Tomo 1: Algorítmica y Lenguajes”, 2º ed., Barcelona: Masson, 1985.
- Clavel, G. y Biondi, J., “Introducción a la Programación. Tomo 2: Estructuras de Datos”, 2º ed., Barcelona: Masson, 1985.



2018 Lic. Ariel Ferreira Szpiniak

84

Citar/Atribuir: Ferreira, Szpiniak, A. (2018). Teoría 21: Representación de Secuencias de marca final e inicial. Introducción a la Algorítmica y Programación (3300). Departamento de Computación. Facultad de Cs. Exactas, Fco-Qcas y Naturales. Universidad Nacional de Río Cuarto.

**Usted es libre para:**

Compartir: copiar y redistribuir el material en cualquier medio o formato.

Adaptar: remezclar, transformar y crear a partir del material.

El licenciante no puede revocar estas libertades en tanto usted siga los términos de la licencia.

Bajo los siguientes términos:



**Atribución:** Usted debe darle crédito a esta obra de manera adecuada, proporcionando un enlace a la licencia, e indicando si se han realizado cambios. Puede hacerlo en cualquier forma razonable, pero no de forma tal que sugiera que usted o su uso tienen el apoyo del licenciante.



**Compartir Igual:** Si usted mezcla, transforma o crea nuevo material a partir de esta obra, usted podrá distribuir su contribución siempre que utilice la misma licencia que la obra original.

<https://creativecommons.org/licenses/by-sa/2.5/ar/>

