

Introducción a la Algorítmica y Programación (3300)

Prof. Ariel Ferreira Szpiniak - aferreira@exa.unrc.edu.ar

Departamento de Computación
Facultad de Cs. Exactas, Fco-Qcas y Naturales
Universidad Nacional de Río Cuarto

Teoría 15

Secuencias

Esquemas de Tratamiento de Secuencias – R1, R2, R3, B, RP



2017 Lic. Ariel Ferreira Szpiniak

Secuencias Motivación

- Desarrolle un Algoritmo que cuente y informe por pantalla la cantidad de números múltiplos de 2, de 3 y de 5 (por separado) que existen entre los números enteros 2 y 20.
- Desarrolle un Algoritmo para una central de carga virtual que solicite al usuario cada uno de los números (PIN) de la tarjeta e informe por pantalla si la cantidad de dígitos es la correcta (12). El algoritmo debe dejar de pedirle números al usuario cuando ésta haya presionado la tecla #.
- Desarrolle un Algoritmo que solicite caracteres al usuario, cuente los que son vocales e informe la cantidad por pantalla. El algoritmo debe dejar de pedirle caracteres al usuario cuando ésta haya ingresado el carácter @.
- Desarrolle un Algoritmo que solicite números enteros positivos al usuario (sin el 0) y los vaya sumando. El algoritmo debe dejar de pedir números cuando el usuario haya ingresado el número 0.



2017 Lic. Ariel Ferreira Szpiniak 2

Secuencias Motivación

Hay muchas maneras de resolver estos problemas.

Sin embargo, ¿qué cosas tienen en común las distintas soluciones?



2017 Lic. Ariel Ferreira Szpiniak 3

Secuencias Noción de Secuencias

Diremos que un conjunto de objetos está organizado en forma de secuencia si es posible definir las siguientes nociones:

1. **Primer elemento de la secuencia:** el acceso a este elemento permite el acceso ulterior a todos los demás elementos de la secuencia.
2. **Relación de sucesión entre elementos:** todo elemento de la secuencia (excepto el último) precede a uno de los demás elementos (su siguiente). Esta relación entre elementos permite, a partir del primero, acceder (recorrer) a todos los elementos de la secuencia, una vez y solo una vez. Este recorrido se termina con el acceso al elemento final.
3. **Caracterización del fin de la secuencia:** debe estar definido un indicador de fin de secuencia, el elemento final, que permite detener el recorrido de la secuencia por observación de la característica del último elemento.



2017 Lic. Ariel Ferreira Szpiniak 4

Secuencias

Noción de Secuencias

Las secuencias así definidas solo autorizan el acceso a un elemento a través del elemento que le precede. Se habla de **acceso secuencial** a los elementos de la secuencia, y por este motivo utilizamos el término **tratamiento secuencial** para referirnos al tratamiento de secuencias.



Esquemas de tratamiento

Introducción

- Las estructuras iterativas vistas anteriormente (**mientras**, **repetir**, **iterar**, **para**) pueden utilizarse para múltiples propósitos.
- **Un uso muy común de las estructuras iterativas**, sobre todo en los que respecta a la programación básica, es para el **tratamiento de secuencias de cosas**.
- Debido a que es **nuestro objetivo**, de ahora en más, es **aprender a tratar secuencias correctamente**, nos centraremos en el **estudio de las secuencias** y de las “**buenas prácticas**” en el **uso de ciclos** para tratar secuencias, a los efectos de minimizar los errores (muy frecuentes cuando se usan iteraciones)
- En un principio nos ocuparemos del tratamiento de las **secuencias sin importarnos como se representan** en una computadora. Más adelante veremos alternativas para representarlas (arreglos, archivos, estructuras dinámicas, etc.)



Esquemas de tratamiento

Introducción

- Existe un gran universo de problemas que involucran la manipulación de secuencias de objetos o el tratamiento secuencial de dichas secuencias. Podemos nombrar entre ellos a las bases de datos, los archivos (ficheros) de almacenamiento de datos, las cadenas de caracteres, los arreglos, los datos ingresados por la entrada estándar (teclado por ejemplo), entre otros.
- Para resolver este tipo de problemas, en general, se debe hacer uso de las estructuras repetitivas (composición iterativa).
- Por esta razón nos dedicaremos a estudiar cuales son las alternativas para resolver adecuadamente este tipo de problemas, tratando de obtener soluciones que se basen en el principio de la programación estructurada y minimicen las posibilidades de error.



Esquemas de tratamiento

Introducción

- La construcción de la iteración se basa en el **descubrimiento** de las **acciones** involucradas en su cuerpo y de la **condición** de terminación (o continuación).
- Es una **labor delicada**, origen de numerosos errores.
- Por ello proponemos una **método sistemático** para **minimizar los riesgos** que se basa en la capacidad de reconocer los problemas que trabajan sobre una **secuencia de objetos** y aplicarle a la misma un...
esquema de tratamiento!!!
esquema de tratamiento!!!
esquema de tratamiento!!!
esquema de tratamiento!!!



Secuencias y Esquemas

En general es posible identificar dos modelos de secuencias:

- con **marca final**
- con **marca inicial**

Modelo de secuencias con Marca Final



Modelo de secuencias con Marca Inicial



Esquemas de Tratamiento

Con el objeto de sistematizar y reducir la posibilidad de errores, se aplican algoritmos genéricos que tratan secuencias, ya preconcebidos.

Estos algoritmos funcionan como plantillas que hay que completar. Por ello se los conoce como **esquemas**.

- **Estos esquemas imponen una partición del problema.**
- **Facilitan la aplicación del método descendente.**



Esquemas de Tratamiento

Los esquemas son un método para construir algoritmos que se fundamentan en el hecho de que toda iteración, leída a posteriori, puede ser interpretada como la enumeración de alguna secuencia.

En tal enumeración se accede a cada uno de los objetos de la secuencia y para cada uno de ellos se aplica un tratamiento único, que depende del problema.

Los esquemas se basan en las siguientes **hipótesis**:

- el acceso a los elementos puede ser descrito en términos de las acciones: *inicialización de la adquisición de los elementos* y *obtener siguiente elemento*.
- el tratamiento de los elementos viene descrito en términos de las siguientes acciones: *inicialización del tratamiento*, *tratamiento del elemento corriente* y *tratamiento final*.



Esquemas de Tratamiento

- **Acceso a los elementos**
 - **inicialización de la adquisición:** todas las acciones necesarias para poder acceder a la secuencia que se va a tratar.
 - **obtener siguiente elemento:** todas las acciones necesarias para acceder al próximo elemento de la secuencia.
- **Tratamiento de los elementos**
 - **inicialización del tratamiento:** todas las acciones necesarias para dotar de sus valores iniciales a las variables y demás estructuras que serán utilizadas en el algoritmo.
 - **tratamiento del elemento corriente:** todas las acciones necesarias para realizar el procesamiento del elemento actual de la secuencia.
 - **tratamiento final:** todas las acciones necesarias para culminar con el trabajo, hacer cálculos finales, informar al usuario de algún resultado, etc.



Esquemas de Tratamiento

Cada modelo secuencial (con marca final y con marca inicial) posee sus propios esquemas de tratamiento de recorrido (R1, R2 y R3), búsqueda (B) y recorrido parcial (RP).

A continuación centraremos el análisis en los esquemas para el modelo secuencial de marca final.

Modelo de secuencias con Marca Final



Modelo de secuencias con Marca Inicial



2017 Lic. Ariel Ferreira Szpiniak 13

Secuencias y Esquemas

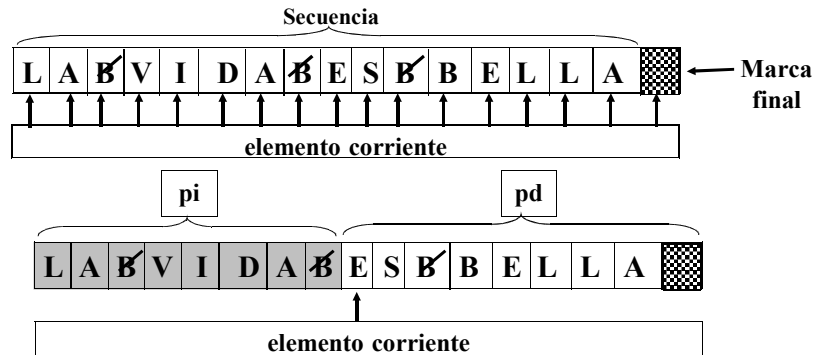
Modelo de secuencias con Marca Final



2017 Lic. Ariel Ferreira Szpiniak 14

Modelo de secuencias con Marca Final

La secuencia de caracteres “LA VIDA ES BELLA” podemos representarla como una secuencia con marca final de la siguiente manera:



Toda secuencia más la marca final puede descomponerse en dos subsecuencias denominadas **pi** (parte izquierda) y **pd** (parte derecha). En **pi** se encuentran los elementos ya analizados y en **pd** los que faltan analizar.




2017 Lic. Ariel Ferreira Szpiniak 15

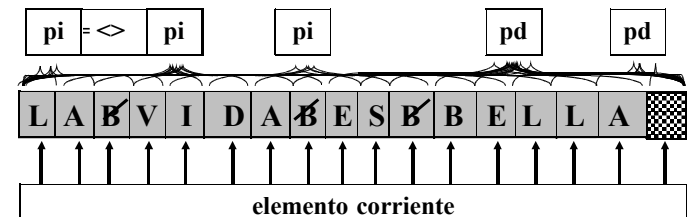
Modelo de secuencias con Marca Final

Las subsecuencias **pi** (parte izquierda) y **pd** (parte derecha) se utilizan frecuentemente para definir el invariante de los ciclos, fundamentalmente la parte izquierda **pi**.

La concatenación de **pi** con **pd** forma la secuencia original con la marca final. Por ello referirse a **pi** y **pd** es lo mismo que referirse a la secuencia con la marca.

Formalmente podemos expresarla como: secuencia  = **pi**&**pd**

El elemento corriente (EC) siempre cumple que: EC = prim(**pd**)



2017 Lic. Ariel Ferreira Szpiniak 16

Modelo de secuencias con Marca Final

inicialización de la adquisición: todas las acciones necesarias para poder acceder a la secuencia que se va a tratar.

{Pre-condición: True }

inicialización de la adquisición

{Pos-condición:

pi = <>

pd

L A B V I D A B E S B B E L L A

elemento corriente

elemento corriente = prim(pd)

}



2017 Lic. Ariel Ferreira Szpiniak 17

Modelo de secuencias con Marca Final

obtener siguiente elemento: todas las acciones necesarias para acceder al próximo elemento de la secuencia.

{Pre-condición:

pi

pd

pd ≠ []

L A B V I D A B E S B B E L L A

elemento corriente

}

obtener siguiente elemento

{Pos-condición:

pi

pd

L A B V I D A B E S B B E L L A

elemento corriente

}



2017 Lic. Ariel Ferreira Szpiniak 18

Modelo de secuencias con Marca Final

¿Cuándo se termina la secuencia?

Condición de fin de secuencia

pi

pd = []

L A B V I D A B E S B B E L L A

elemento corriente

Formalmente podemos expresarla como:

secuencia = pi

pd = []

elemento corriente = prim(pd) = marca



2017 Lic. Ariel Ferreira Szpiniak 19

Esquemas de Tratamiento

Se reconocen 3 tipos de problemas sobre secuencias:

- Recorrido
- Búsqueda
- Recorrido Parcial (búsqueda + recorrido)

Cualquier problema o subproblema sobre secuencias puede encajarse dentro de alguno de estos 3 tipos.

Hay casos donde el problema es de un tipo y posee algún subproblema que es de otro tipo. Hay otros casos donde el problema general no es de tratamiento de secuencias, pero algún/os subproblema/s sí.



2017 Lic. Ariel Ferreira Szpiniak 20

Esquemas de Tratamiento

Para cada tipo de problemas hay esquemas generales de diseño:

• Recorrido

- **Esquema R1:** tratamiento integrado de la secuencia vacía y del primer elemento.
- **Esquema R2:** tratamiento especial de la secuencia vacía.
- **Esquema R3:** tratamiento especial del primer elemento.

• Búsqueda

- **Esquema B:** búsqueda del primer elemento que verifica cierta propiedad.

• Recorrido Parcial

- **Esquema RP:** tratamiento de todos los elementos hasta cierta propiedad.



2017 Lic. Ariel Ferreira Szpiniak 21

Modelo secuencial de marca final Recorrido: Esquema R1

Tratamiento integrado de la secuencia vacía y del primer elemento (trata todos los elementos de la secuencia sin diferenciar el caso de la secuencia vacía).

Esquema R1 General

<inicialización de la adquisición>

<inicialización del tratamiento>

mientras <no fin de secuencia> **hacer**

<tratamiento del elemento corriente>

<obtener siguiente elemento>

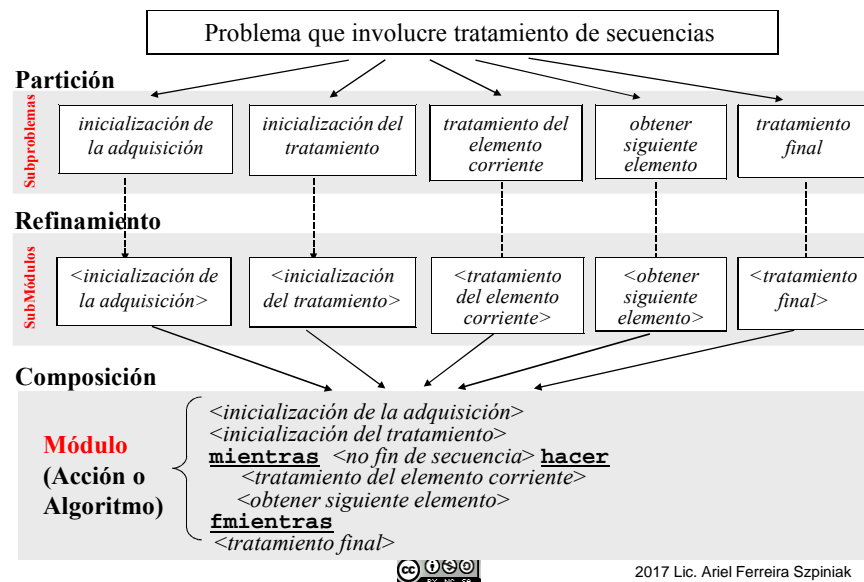
fmientras

<tratamiento final>



2017 Lic. Ariel Ferreira Szpiniak 22

Recorrido: Esquema R1



2017 Lic. Ariel Ferreira Szpiniak 23

Modelo secuencial de marca final Recorrido: Esquema R1 - Ejemplo

Algoritmo Sumar

Lexico

acum, $a \in \mathbb{Z}$

Inicio

Leer(a) inicialización de la adquisición

acum $\leftarrow 0$ inicialización del tratamiento

mientras $a < 99$ **hacer**

acum \leftarrow acum + a tratamiento del elemento corriente

Leer(a) obtener siguiente elemento

fmientras

Escribir("La suma de los números es: ", acum) tratamiento final

Fin



2017 Lic. Ariel Ferreira Szpiniak 24

Modelo secuencial de marca final

Recorrido: Esquema R2

Trata todos los elementos de la secuencia diferenciando el caso de la secuencia vacía.

Esquema R2 General

<inicialización de la adquisición>

según

<fin de secuencia> : <tratamiento secuencia vacía>

<no fin de secuencia> :

<inicialización del tratamiento>

mientras <no fin de secuencia> hacer

<tratamiento del elemento corriente>

<obtener siguiente elemento>

fmientras

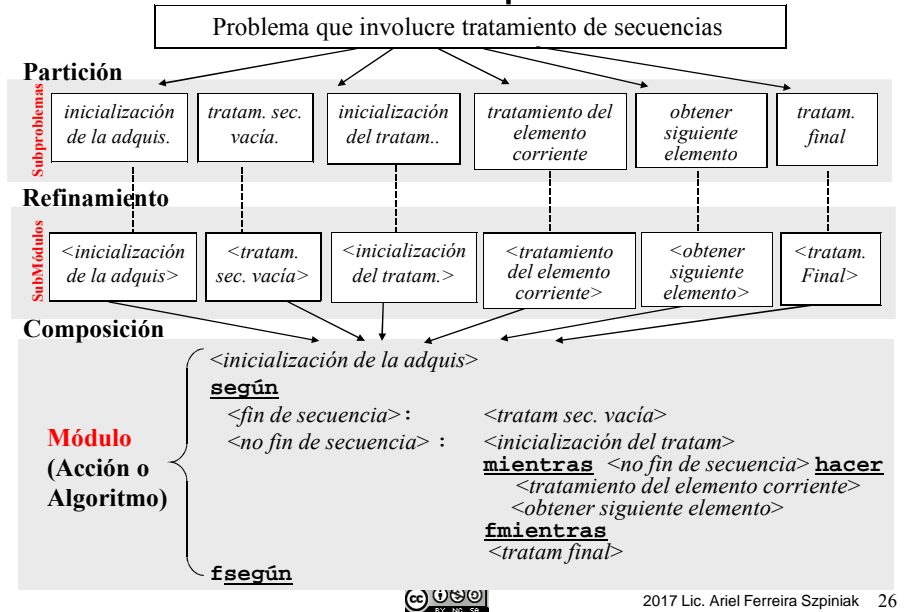
<tratamiento final>

fsegún



2017 Lic. Ariel Ferreira Szpiniak 25

Recorrido: Esquema R2



2017 Lic. Ariel Ferreira Szpiniak 26

Modelo secuencial de marca final

Recorrido: Esquema R2 - Ejemplo

Algoritmo Sumar

Lexico

acum, a ∈ Z

Inicio

Leer(a) inicialización de la adquisición

según

a=99: Escribir('No hay números para sumar') tratamiento secuencia vacía

a <> 99: acum ← 0 inicialización del tratamiento

mientras a <> 99 hacer

acum ← acum + a tratamiento del elemento corriente

Leer(a) obtener siguiente elemento

fmientras

Escribir('La suma de los números es: ', acum) tratamiento final

fsegún

Fin



2017 Lic. Ariel Ferreira Szpiniak 27

Modelo secuencial de marca final

Recorrido: Esquema R3

Trata todos los elementos de la secuencia diferenciando el primero.

Esquema R3 General

<inicialización de la adquisición>

según

<fin de secuencia> : <tratamiento secuencia vacía>

<no fin de secuencia> :

<inicialización del tratamiento>

<tratamiento del primer elemento>

iterar

<obtener siguiente elemento>

parada <fin de secuencia>

<tratamiento del elemento corriente>

fiterar

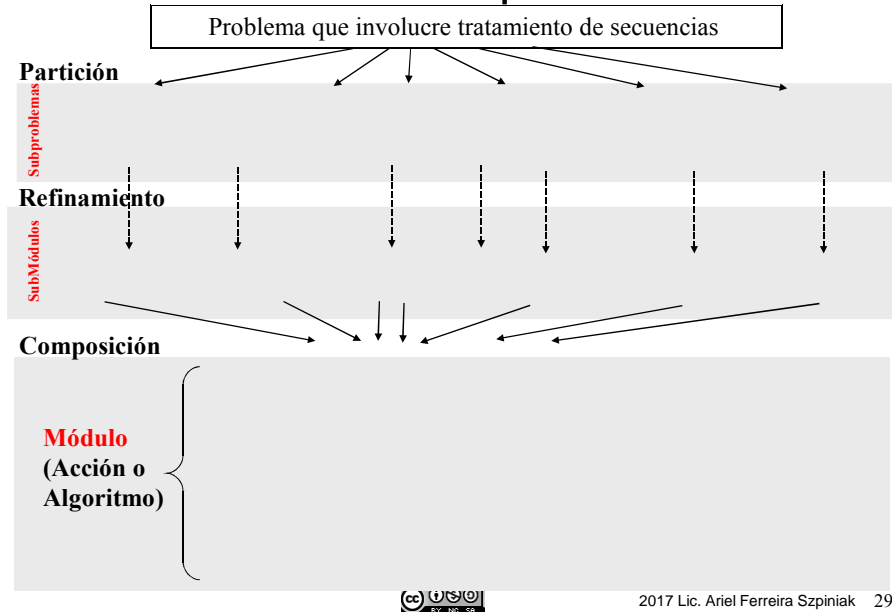
<tratamiento final>

fsegún



2017 Lic. Ariel Ferreira Szpiniak 28

Recorrido: Esquema R3



Modelo secuencial de marca final

Recorrido: Esquema R3 - Ejemplo

Algoritmo ContarElPrimero

Lexico
cont, a, primero $\in \mathbb{Z}$

Inicio
Leer(a) *inicialización de la adquisición*

según
a=99: Escribir('No hay números para verificar') *tratamiento secuencia vacía*

a<>99: cont \leftarrow 0 *inicialización del tratamiento*
primero \leftarrow a *tratamiento del primer elemento*

iterar
Leer(a) *obtener siguiente elemento*

parada a=99

si a=primero *tratamiento del elemento corriente*
entonces
cont \leftarrow cont + 1

fsi

fiterar
Escribir('Hay ', cont, 'repeticiones de ', primero) *tratamiento final*

fsegún

Fin

2017 Lic. Ariel Ferreira Szpiniak 30

Modelo secuencial de marca final

Búsqueda: Esquema B

Recorre la secuencia hasta que encuentra un elemento que cumple cierta propiedad. Si no hay elementos que cumplan con la propiedad buscada, recorre la secuencia hasta el final.

Esquema B General

<inicialización de la adquisición>

según

<fin de secuencia>: <tratamiento secuencia vacía>

<no fin de secuencia> :

<inicialización del tratamiento>

mientras <no fin de secuencia> **y** $\neg P(elem)$ **hacer**

<obtener siguiente elemento>

fmientras

según

<no fin de secuencia>: <trat.final por elem hallado>

<fin de secuencia>: <trat.final por elem NO hallado>

fsegún

fsegún

Dependiendo del problema la propiedad P podría aplicarse a la parte ya recorrida de la secuencia

Atención!!!



Forma de evaluación de condiciones

Existen diferentes modos de evaluación de las condiciones.

En notación algorítmica consideraremos que la evaluación de las condiciones es una especie de evaluación perezosa, donde los argumentos de una condición los evaluamos de izquierda a derecha y solo completamos la evaluación si es necesario para poder determinar el valor de verdad.

Por ejemplo, en el Esquema B General, la condición (<no fin de secuencia> y $\neg P(elem)$) evaluará primero <no fin de secuencia> y si es necesario evaluará $\neg P(elem)$.

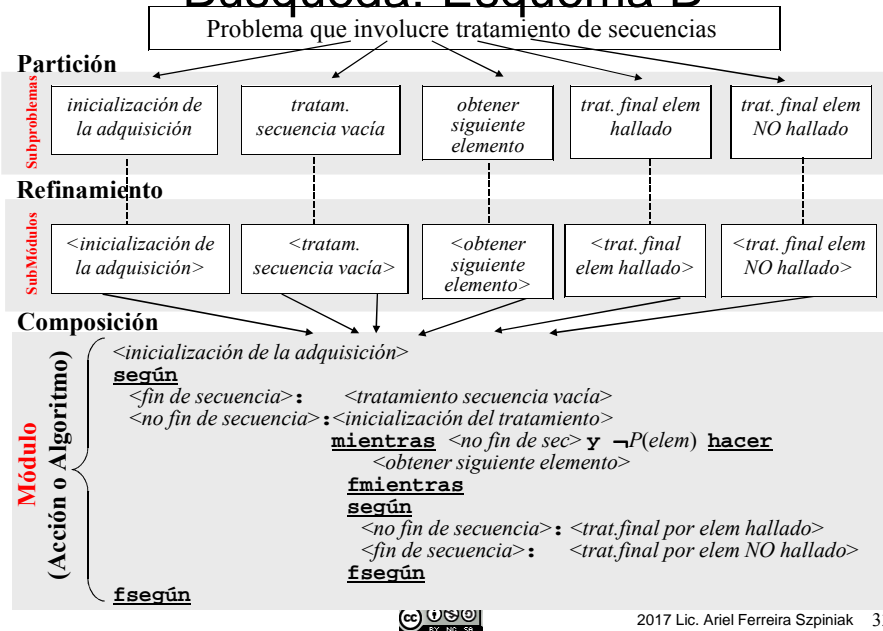
Este tipo de evaluación evita cierto tipo de problemas como el caso donde el elemento a tratar (elem) sea el que sirve para determinar el fin de secuencia, que en muchos casos se trata de un elemento "virtual" o de un tipo diferente al resto, con lo cual no puede determinarse el valor de verdad de $P(elem)$.

Cuando implementamos en un lenguaje de programación debemos tener especial cuidado con este tipo de situaciones. Tal es el caso de Pascal.

Más adelante veremos la manera de expresar los esquemas en Pascal y como modificarlos para superar estos problemas.

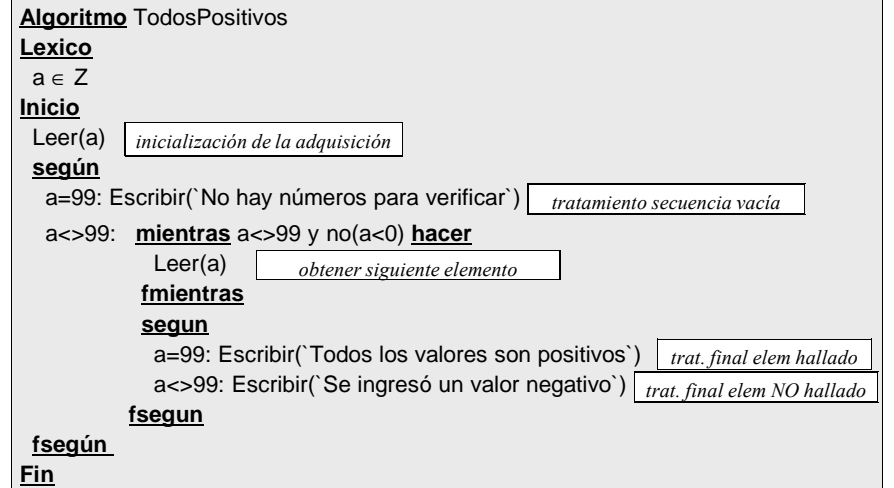


Búsqueda: Esquema B



Modelo secuencial de marca final

Recorrido: Esquema B - Ejemplo

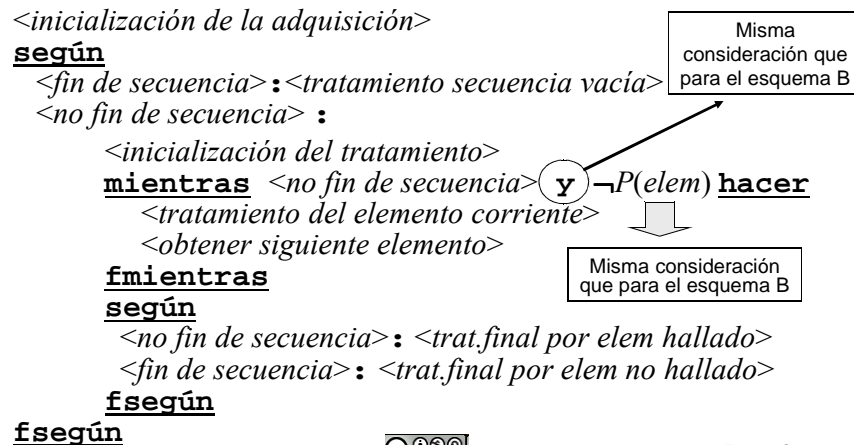


Modelo secuencial de marca final

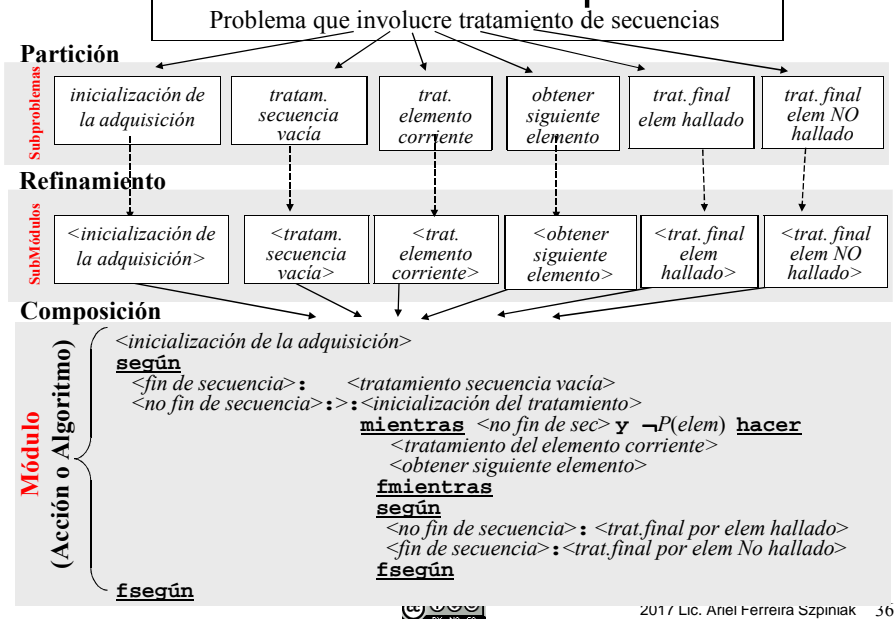
Recorrido Parcial: Esquema RP

Trata todos los elementos hasta que se cumpla cierta propiedad.

Esquema RP General



Recorrido Parcial: Esquema RP



Modelo secuencial de marca final

Recorrido: Esquema RP - Ejemplo

Algoritmo CuadradoMenoresQue100

Lexico

$a \in \mathbb{Z}$

Inicio

Leer(a) *inicialización de la adquisición*

según

$a=99$: Escribir(' No hay números para verificar') *tratamiento secuencia vacía*

$a < 99$: **mientras** $a < 99$ y $\neg(a \geq 100)$ **hacer**

Escribir(' El cuadrado de ', a, ' es: ', a^2) *tratamiento del elemento corriente*

Leer(a) *obtener siguiente elemento*

fmientras

según

$a=99$: Escribir(' Todos los valores son < que 100') *trat. final elem hallado*

$a < 99$: Escribir(' Se ingresó un valor $\geq a$ 100') *trat. final elem NO hallado*

fsegún

fsegún

Fin



2017 Lic. Ariel Ferreira Szpiniak 37

Notación de Secuencias

En lo sucesivo, para poder especificar sobre secuencias, adoptaremos la notación vista en la teoría 14 (Especificación. Secuencias. Notación) que nos permitirá darle mayor precisión a nuestras especificaciones.



2017 Lic. Ariel Ferreira Szpiniak 38

Resumen de esquemas

Esquema R1 General

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

Esquema R3 General

<inicialización de la adquisición>
según
<fin de secuencia>: <tratamiento secuencia vacía>
<no fin de secuencia> :
<inicialización del tratamiento>
<tratamiento del primer elemento>
iterar
<obtener siguiente elemento>
parada <fin de secuencia>
<tratamiento del elemento corriente>
fiterar
<tratamiento final>

fsegún

Esquema R2 General

<inicialización de la adquisición>
según
<fin de secuencia>: <tratamiento secuencia vacía>
<no fin de secuencia> :
<inicialización del tratamiento>
mientras <no fin de secuencia> **hacer**
<tratamiento del elemento corriente>
<obtener siguiente elemento>
fmientras
<tratamiento final>
fsegún



2017 Lic. Ariel Ferreira Szpiniak 39

Resumen de esquemas

Esquema B General

<inicialización de la adquisición>
según
<fin de secuencia>: <tratamiento secuencia vacía>
<no fin de secuencia> :
<inicialización del tratamiento>
mientras <no fin de secuencia> y $\neg P(\text{elem})$ **hacer**
<obtener siguiente elemento>
fmientras
según
<no fin de secuencia>: <trat. final por elem hallado>
<fin de secuencia>: <trat. final por elem NO hallado>
fsegún
fsegún

Esquema RP General

<inicialización de la adquisición>
según
<fin de secuencia>: <tratamiento secuencia vacía>
<no fin de secuencia> :
<inicialización del tratamiento>
mientras <no fin de secuencia> y $\neg P(\text{elem})$ **hacer**
<tratamiento del elemento corriente>
<obtener siguiente elemento>
fmientras
según
<no fin de secuencia>: <trat. final por elem hallado>
<fin de secuencia>: <trat. final por elem no hallado>
fsegún
fsegún



2017 Lic. Ariel Ferreira Szpiniak 40

Bibliografía

– Scholl, P. y Peyrin, J.-P. “Esquemas Algorítmicos Fundamentales: Secuencias e iteración”.

– Peyrin, J.-P. “Material del Seminario sobre la enseñanza de la programación: Todavía no aprendí a enseñar programación”. Escuela de Verano de Ciencias Informáticas, RIO 2006.



2017 Lic. Ariel Ferreira Szpiniak

Citar/Atribuir: Ferreira, Szpiniak, A. (2017). Teoría 15: Secuencias. Esquemas de Tratamiento de Secuencias – R1, R2, R3, B, RP. 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/>

