

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 7

**Acciones y funciones anidadas,
Reglas de alcance.**



2017 Lic. Ariel Ferreira Szpiniak

Novedades Escudo



Simbolismo

El escudo que identifica a la UNRC fue seleccionado en un concurso nacional.

- La franja superior es una guarda indigenista que representa la tradición.
- Las iniciales de la Universidad sobre el cielo simbolizan la concreción de las aspiraciones de la juventud riocuartense y de los hombres y mujeres que participaron en la obtención de una casa de altos estudios para la ciudad y zona de influencia.
- Las montañas representan el empuje de la juventud y la solidez y aplomo de los mayores que la guían.
- El río (franja celeste) el origen del nombre de la ciudad.

Noticias

1 de mayo: Día del Trabajador

Revuelta de Heymarket (1886, Chicago):

Dos recibieron cadena perpetua, uno 15 años de cárcel, cinco pena de muerte. A esos cinco les espera la horca, pero Lingg madrugó a la muerte haciendo estallar entre sus dientes una cápsula de dinamita. Fischer se viste sin prisa, tarareando "La Marsellesa". Parsons, el agitador que empleaba la palabra como látigo o cuchillo, aprieta las manos de sus compañeros antes de que los guardias se las aten a la espalda. Engel, famoso por la puntería, pide vino de Oporto y hace reír a todos con un chiste. Spies, que tanto ha escrito pintando a la anarquía como la entrada en la vida, se prepara, en silencio, para entrar en la muerte.



**"Mártires
de
Chicago"**

Noticias

1 de mayo: Día del Trabajador

Chicago está llena de fábricas, hay fábricas hasta en pleno centro de la ciudad, en torno al edificio más alto del mundo. Chicago está llena de fábricas, Chicago está llena de obreros.

Al llegar al barrio de Heymarket, pido a mis amigos que me muestren el lugar donde fueron ahorcados, en 1886, aquellos obreros que el mundo entero saluda cada primero de mayo.

-Ha de ser por aquí- me dicen. Pero nadie sabe.

Ninguna estatua se ha erigido en memoria de los mártires de Chicago en la ciudad de Chicago.

Ni estatua, ni monolito, ni placa de bronce, ni nada.

El primero de mayo es el único día verdaderamente universal de la humanidad entera, el único día donde coinciden todas las historias y todas las geografías, todas las lenguas y las religiones y las culturas del mundo; pero en los Estados Unidos, el primero de mayo es un día cualquiera.

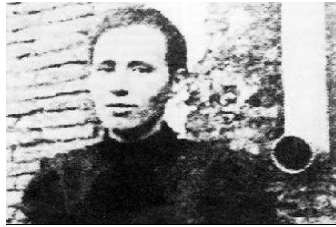
Ese día la gente trabaja normalmente, y nadie, o casi nadie, recuerda que los derechos de la clase obrera no han brotado de la oreja de una cabra, ni de la mano de Dios o del amo (Eduardo Galeano).

Noticias

1 de mayo: Día del Trabajador

Virginia Bolten (nació en San Luis en 1876), una luchadora por los derechos laborales femeninos a fines del siglo XIX, se atrevió a arengar a los obreros el 1 de mayo de 1890 en Rosario, cuando se celebró por primera vez el Día del Trabajador. Ella subió al escenario vestida de negro portando la bandera del anarquismo y denunciando la explotación laboral de las mujeres.

También fundó un periódico, cuyo slogan fue: "Ni Dios, ni patrón, ni marido", en tiempos en que las mujeres no tenían ni voz ni voto.



Trabajaba en una Refinería de Azúcar, donde se reveló contra la explotación de las mujeres.

Comenzó a agitar a todo el personal para cambiar las condiciones laborales hasta que finalmente la expulsaron del trabajo.

Abstracciones anidadas

- Las abstracciones anidadas se refieren a la posibilidad de declarar funciones y acciones dentro de otras funciones y acciones.
- El anidamiento también puede ser mixto, es decir, una acción puede tener definida una función dentro de ella (interna) y una función puede tener definida una acción dentro de ella (interna).
- La definición de las acciones o funciones internas deben aparecer en el léxico, es decir, junto a la declaración de las constantes y variables locales de la acción o función que las incluye.
- Es importante notar que las acciones y funciones internas son de uso **exclusivo** de la acción o función en que fueron declarados.



2017 Lic. Ariel Ferreira Szpiniak 6

Abstracciones anidadas Ejemplo

- Deseamos escribir algoritmo que contenga una acción **CalcSueldo**, que calcule el salario neto en pesos (\$) de un empleado. La acción lee de la entrada estándar: (a) el salario bruto de un empleado, dado por un real con un signo (\$, D o E), (b) el tipo de empleado, dado por los valores 1 o 2.
- Si el tipo de empleado es 1, el valor neto del salario se obtiene de aplicarle un único descuento al salario bruto, mientras que si el tipo es 2 se le aplican dos descuentos al salario bruto.
- Los porcentajes de descuento son leídos por el algoritmo, desde la entrada estándar, y pasados como parámetros a la acción **CalcSueldo**.



2017 Lic. Ariel Ferreira Szpiniak 7

Abstracciones anidadas Ejemplo (cont.)

Supongamos ya realizadas las etapas de análisis y diseño.

Algoritmo CalcularSueldo

Lexico

```
TMoneda =(P,D,E) {tipo de moneda Pesos, Dolares o Euros}
desc1, desc2 ∈ R {descuentos}
tipoMoneda ∈ TMoneda {moneda en que cobra el sueldo}
salBruto, salNeto ∈ R {salarios bruto y neto}
Acción LeerDescuentos(resultado d1, d2 ∈ R)
Acción CalcSueldo(dato d1,d2 ∈ R, resultado bruto,neto ∈ R)
```

Inicio {CalcularSueldo}

```
LeerDescuentos(desc1, desc2)
```

```
CalcSueldo(desc1, desc2, salBruto, salNeto)
```

Fin {CalcularSueldo}



2017 Lic. Ariel Ferreira Szpiniak 8

Abstracciones anidadas Ejemplo (cont.)

Las acciones estarían definidas de la siguiente manera:

Acción LeerDescuentos(resultado d1, d2 ∈ R)

Inicio

```
Escribir('Ingrese los dos porcentajes de descuento')
Leer(d1, d2)
```

Fin

Acción CalcSueldo(dato d1, d2 ∈ R, resultado bruto, neto ∈ R)

Lexico Local

Acción SignoMonetario(resultado signo ∈ TMoneda)

Acción LeerSalBruto(dato signo1 ∈ TMoneda, resultado bruto1 ∈ R)

Acción CalcNeto(dato d1, d2, bruto2 ∈ R, resultado neto1 ∈ R)

Inicio {CalcSueldo}

```
SignoMonetario(tipoMoneda)
LeerSalBruto(tipoMoneda, bruto)
CalcNeto(d1, d2, bruto, neto)
```

Fin {CalcSueldo}



Abstracciones anidadas Ejemplo (cont.)

Acción SignoMonetario(resultado signo ∈ TMoneda)

Lexico Local

moneda ∈ Caracter

Inicio

```
Escribir('Ingrese el tipo de moneda del sueldo: 'P'
para pesos, 'D' para dolares, 'E' para euros')
```

Leer(moneda)

según

(moneda='P')o(moneda='p'): signo ← P

(moneda='D')o(moneda='d'): signo ← D

(moneda='E')o(moneda='e'): signo ← E

otros: signo ← P

fsegún

Fin



Abstracciones anidadas Ejemplo (cont.)

Acción LeerSalBruto(dato signo1 ∈ TMoneda, resultado bruto1 ∈ R)

Lexico Local

bruauux ∈ R

Función Convertir(dato b ∈ R, s ∈ TMoneda) · R

Lexico

val ∈ R

Inicio

según

s = P: val ← b

s = D: val ← b * 15,60

s = E: val ← b * 17,20

fsegún

← val

Fin

Inicio {LeerSalBruto}

```
Escribir('Ingrese el salario bruto')
```

```
Leer(bruauux)
```

```
bruto1 ← Convertir(bruauux, signo1)
```

Fin {LeerSalBruto}



Abstracciones anidadas Ejemplo (cont.)

Acción CalcNeto(dato d1, d2, bruto2 ∈ R, resultado neto1 ∈ R)

Lexico Local

tipoEmp ∈ 1..2

Inicio

```
Escribir('Ingrese el tipo de empleado (1 o 2)')
```

Leer(tipoEmp)

según

tipoEmp = 1: neto1 ← bruto2 - bruto2 * d1 / 100

tipoEmp = 2: neto1 ← bruto2 - bruto2 * (d1+d2) / 100

fsegún

Fin



Identificadores locales

- Las acciones y las funciones pueden declarar identificadores de uso local.
- Estos identificadores son locales en el sentido de que son vistos solamente dentro de la acción o la función. O sea, fuera de ellos nadie sabe de su existencia y su utilización provocaría un fallo.



Reglas de alcance

- La existencia de identificadores globales y locales, así como la presencia de acciones y funciones anidadas, hace necesario definir con claridad las reglas de alcance de cada identificador.
- El **alcance** de un identificador es aquella porción del algoritmo en que se conoce al identificador.
- Veamos esto en el ejemplo anterior.



```
Acción CalcSueldo(dato d1, d2 ∈ R, resultado bruto, neto ∈ R)
  Acción SignoMonetario(resultado signo ∈ TMoneda)
  ...

  Acción LeerSalBruto(dato signo1 ∈ TMoneda, resultado bruto1 ∈ R)
  Lexico Local
    bruaux ∈ R
    Función Convertir(dato b ∈ R, s ∈ TMoneda) → R
    ...

  Acción CalcNeto(dato d1, d2, bruto2 ∈ R, resultado neto1 ∈ R)
  Lexico Local
    tipoEmp ∈ 1..2
    ...

Inicio {CalcSueldo}
...
Fin {CalcSueldo}
```



Reglas de alcance

¿Puedo usar **tipoEmp** en la acción **CalcularSueldo**?

- La variable **tipoEmp** es local a la acción **CalcNeto** y por lo tanto su alcance se reduce a esta acción. O sea, **tipoEmp** no se conoce ni en la acción **CalcSueldo** ni en el algoritmo principal **CalcularSueldo**.

¿Puedo utilizar **bruaux** y **Convertir** en otra acción que no sea **LeerSalBruto**?

- No, porque son locales a la acción **LeerSalBruto** y por lo tanto son conocidas únicamente por esta acción.



Reglas de alcance

- Los parámetros formales de **CalcSuelto** juegan el papel de identificadores:
 - locales a la acción, y
 - globales a las acciones anidadas.
- Por ejemplo: el parámetro formal **bruto** (que es de salida) es conocido en la acción **CalcNeto**. Por lo tanto podría usarse sin necesidad de pasarlo como parámetro.



Reglas de alcance

- La situación de los parámetros formales **d1** y **d2** de **CalcSuelto** es diferente en lo que respecta a **CalcNeto**.
- Dado que esta acción usa esos mismos nombres para sus parámetros formales, los identificadores **d1** y **d2** de **CalcSuelto** no son visibles dentro de **CalcNeto**.



Reglas de alcance

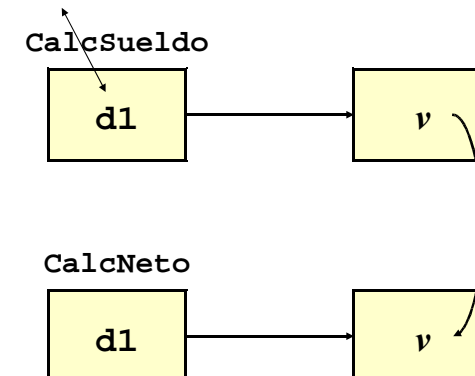
- Por ejemplo, las ocurrencias de **d1** en **CalcNeto**, se refieren al parámetro formal **d1** del mismo y **no** a la variable global **d1** (parámetro formal de **CalcSuelto**).
- A pesar de tener nombres iguales, y que la invocación a **CalcNeto** en el algoritmo principal de **CalcSuelto** tiene a estos mismos nombres como argumentos, **d1** y **d2** en una y otra acción denotan variables **diferentes**.



Reglas de alcance

Llamada desde **CalcSuelto** a

CalcNeto(d1,d2,bruto,neto)



Reglas de alcance

Ejemplo en Pascal

- Continuemos con el análisis de las reglas de alcance viendo otro ejemplo pero en **Pascal**.
- Al igual que en el ejemplo anterior, presentamos un dibujo donde para cada procedimiento mostramos su nombre, sus parámetros formales, sus variables locales y procedimientos internos.
- El tipo de las variables (y parámetros), así como el cuerpo de los procedimientos no son especificados para simplificar el dibujo.



```
Procedure Externo(x; VAR y);  
  VAR a,b; CONST c;  
  Procedure Interno1(x,y);  
    VAR a, z;  
    BEGIN  
      ...  
    END;  
  Procedure Interno2(VAR a);  
    VAR v, w;  
    Procedure Masinterno(VAR x,y);  
      VAR a;  
      BEGIN  
        ...  
      END;  
    BEGIN  
      ...  
    END;  
  BEGIN  
    ...  
  END;
```



Reglas de alcance

- La constante **c** y las variables **a** y **b** que se declaran en el procedimiento **Externo** son identificadores locales a este procedimiento.
- Su alcance abarca todo el cuadro principal, incluyendo los procedimientos **Interno1**, **Interno2** y **Masinterno** (para quienes estos identificadores son globales).
- Lo mismo puede decirse de los parámetros formales **x** y **y** del procedimiento **externo**, o sea, son locales a **externo** y globales a los otros tres procedimientos internos.



Reglas de alcance

- En lo que respecta al procedimiento **Interno1**, el alcance de las **variables** **a** y **z** y sus parámetros formales **x** e **y** es únicamente el cuerpo de este procedimiento, ya que son variables locales.
- Observar que el nombre de variable **a** aparece tanto en el procedimiento principal (**Externo**) como en el procedimiento **Interno1**. Cualquier referencia a la variable **a** en el cuerpo de **Interno1** será a la variable local **a** y no a la global.



Reglas de alcance

- Ahora analicemos el procedimiento **Interno2**. Las variables locales **v** y **w** y el parámetro formal **a** se conocen únicamente dentro de este procedimiento.
- Estas son variables globales para el procedimiento **Masinterno**, dado que **Masinterno** está contenido dentro de **Interno2** y su declaración es posterior a la declaración de las variables **v**, **w**, y el parámetro **a**.



Reglas de alcance

- Las variables **a** y **b** y la constante **c** declaradas en el procedimiento principal (**Externo**) son también identificadores globales para **Masinterno**.
- Sin embargo, notar que **Masinterno** tiene una variable local de nombre **a**. Eso significa que cualquier referencia a la variable **a** dentro de **Masinterno** será a la variable **a** declarada localmente dentro del procedimiento.



Reglas de alcance Procedimientos y Funciones

- Los nombres de procedimientos y funciones se rigen por las mismas reglas de alcance que el resto de los identificadores.
 - Por ejemplo, el procedimiento **Interno2** puede invocar al procedimiento **Interno1** ya que fue declarado después que este último. En cambio, **Interno1** no puede invocar a **Interno2**.*
- * En realidad en Pascal se puede, pero no es conveniente hacerlo salvo un caso de extrema necesidad (En Pascal se coloca la palabra **forward** en el encabezado de la función o el procedimiento).



Reglas de alcance Procedimientos y Funciones

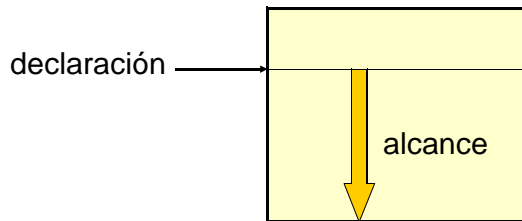
- El procedimiento **Masinterno** puede llamar al procedimiento **Interno1**.
- El procedimiento **Masinterno** sólo puede ser invocado dentro de **Interno2**



Reglas de alcance

En resumen...

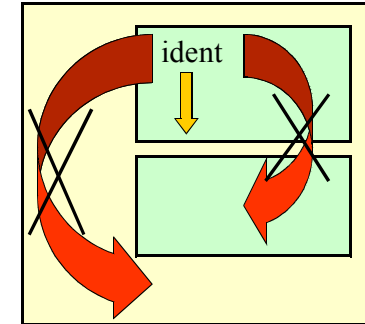
- El alcance de todo identificador es la porción de texto (declaraciones y sentencias) del algoritmo, acción o función (en Pascal, programa, procedimiento o función) que sigue a la declaración del identificador; y finaliza donde termina el algoritmo, acción o función que lo contenga (en Pascal, programa, procedimiento o función).



Reglas de alcance

En resumen...

- Esto significa que todo identificador es sólo conocido en el bloque en que es declarado (y en los bloques internos a este), pero **no** lo es ni en los bloques hermanos ni en ningún bloque externo.



Reglas de alcance

En resumen...

- Los parámetros formales de una acción o función (procedimiento o función) son tratados como variables locales del mismo. Por lo tanto, su alcance es el bloque formado por el procedimiento.
- Los identificadores locales que tienen el mismo nombre que identificadores globales tienen prioridad sobre los globales. Por lo tanto, para hacer referencia a una variable global en tal condición, ésta debe ser pasada como parámetro.

En anidamiento no es muy utilizado porque dificulta el reuso de las abstracciones.



Bibliografía

- Scholl, P. y Peyrin, J.-P. “Esquemas Algorítmicos Fundamentales: Secuencias e iteración”. (pags. 1 - 34)
- Biondi, J. y Clavel, G. “Introducción a la Programación. Tomo 1: Algorítmica y Lenguajes”. (pags. 1 – 12, 13 – 34, 127 - 140)
- Wirth, N. “Algoritmos + Estructuras de Datos = Programas”. (pags. 1 – 12).
- Watt, David: Programming Language Concepts and Paradigms, Prentice-Hall International Series in Computer Science (1990). Cap 4.
- Quetglás, Toledo, Cerverón. “Fundamentos de Informática y Programación”. Capítulo 3. <http://robotica.uv.es/Libro/Indice.html>
 - Programación Modular (pags 110 – 125)
- Pascal
 - introturbopascal.pdf (aula virtual)
 - laprogramacionenlenguajepascal.pdf (aula virtual)
 - pascalyturbopascal.pdf (aula virtual)
 - Biondi, J. y Clavel, G. “Introducción a la Programación. Tomo 1: Algorítmica y Lenguajes”: (pags. 243 - 252)
 - Joyanes Aguilar, L., “Programación en Turbo Pascal”. Mc Graw Hill, 1993.
 - Wirth, N. and K. Jensen, “Pascal: User Manual and Report”, 4ª ed., New York, Springer-Verlag, 1991 (traducción de la primera edición “Pascal: Manual del Usuario e Informe”, Buenos Aires, El Ateneo, 1984).



Citar/Atribuir: Ferreira, Szpiniak, A. (2017). Teoría 7: Acciones y funciones anidadadas, Reglas de alcance. 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 licenciente 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 licenciente.



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/>

