

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|----------------------|-------------------|-------------|
| Harold Matto | Programa matrices | Carlos Richard | 10/11/22 |

Title *Funciones*

| Keyword | Topic |
|--|---|
| <ul style="list-style-type: none"> • Percepción • parte • conjunto • actividad • problema | <p><i>Reducción de problemas</i></p> <p>Mediante este concepto el problema de la computación se convierte en un subproblema, hasta que el original se reduce a un conjunto de actividades básicas.</p> <p>En el lenguaje C, la solución de un problema se expresa mediante un programa y la solución de un subproblema, por medio de una función.</p> |
| <p>Questions</p> <ul style="list-style-type: none"> ¿cómo se compo- ne un pro- grama en C? ¿qué es un subproble- ma? | <p>Un programa en C está constituido por un pro- grama principal y un conjunto de funciones.</p> |

Summary: *La reducción consiste en dividir un gran problema, en varios más pequeños.*

| NAME | CLASS | SPEAKER | DATE & TIME |
|------------|--------------|---------|-------------|
| varaldmatw | Programación | Richard | 10/11/22 |

Title Funciones

| Keyword | Topic |
|---|---|
| • objeto • cambio • Ejemplo | Variables locales, globales y estáticas. Las variables son objetos que pueden cambiar su valor durante la ejecución de un programa. Las locales son objetos definidos en el programa en las funciones. Las globales son objetos definidos antes del inicio del programa principal. Las variables estáticas y locales no cambian su valor durante la ejecución del programa, y cambian al llamar la función. |
| Questions ¿cuál tipo de variable es esto? ¿cómo se declara? | |

Summary: Las variables son funciones de varios tipos, capaces de cambiar su valor en distintas etapas de la ejecución del programa.

| | | | |
|---------------------|------------------------------|------------------------|-------------------------|
| NAME Carla Irato | CLASS 3.º grado matrón | SPEAKER Carla Irato | DATE & TIME 10/14/22 |
|---------------------|------------------------------|------------------------|-------------------------|

Title *Funciones*

| Keyword | Topic |
|--|--|
| <i>Procedimiento</i> | <i>Conflicto entre nombres de variables</i> |
| <i>Definición</i> | <i>Las variables locales en una procedimiento están en el ámbito del mismo nombre, es decir, allí se define la variable y llaman "I", la cual interactúa en función de la local, manipulando directamente la variable, haciéndole uso del símbolo I delante de la función: I</i> |
| Questions | |
| <i>¿Por qué es importante usar nombres de distinto tipo?</i> | |

Summary: *Usar variables de distinto tipo, con el mismo nombre, puede haber confusión, por eso es del saber a cual llamas en cada momento.*

| NAME | CLASS | SPEAKER | DATE & TIME |
|-------------|--------------|---------|-------------|
| Harold nato | Programación | Richard | 10/11/22 |

Title Funciones

| Keyword | Topic |
|--------------|---|
| Comunicación | Por ausencia. |
| Paras | La comunicación entre las funciones y el programa. |
| Datos | Se lleva a cabo mediante variables y parámetros que son: |
| dirección | - Por valor: permite pasar dato entre el programa y la función, o entre las mismas funciones. |
| Questions | - Por referencia: también lo mismo que la anterior pero en lugar de escribir una copia de la variable en el programa, se escribe la dirección de la variable. |

Summary: El programa principal necesita comunicarse con las funciones, usando como medio a los parámetros.

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|---------------|-------------------|-------------|
| Harold Mateo | Prueba macton | Carlyton Richardo | 10/14/22 |

Title *Funciones*

| Keyword | Topic |
|--|--|
| <ul style="list-style-type: none"> • Acuntador • Dirección • operador | <p><i>Paso de funciones como parámetros.</i></p> <p><i>En el lenguaje C se puede realizar esto por medio de un acuntador que contiene la dirección de otra variable función y se representa por medio de los operadores de dirección "&" e indirecta "*", permitiendo esto que una función se pueda tratar como si fuera una variable.</i></p> |
| <p>Questions</p> <p><i>¿qué me dio el primer mi función?</i></p> | |

Summary: *Los acuntadores y sus operadores permiten pasar funciones como variables.*

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|------------|---------|-------------|
| Harald Matto | Pragmatics | Richard | 10/11/22 |

Title *A reglas unidimensionales*

| Keyword | Topic |
|-------------|----------------------------|
| Problema | Tipo estructurado de datos |
| Difícil | En la práctica podemos |
| Simple | encontrar como |
| Memoria | problema muy difícil |
| Identificar | de seleccionar una |
| | mejor tipo simple de da |
| | tos, o datos que ocupan |
| | una sola celda de me |
| | moría, pero aquí están |
| | los datos estructurados |
| | que ocupan varias ce |
| Questions | lillas de memoria. ¿ |
| ¿cómo | identificar con un nom |
| simple? | bre único. |
| ¿cómo | |
| identificar | |
| los? | |
| | |
| | |
| | |
| | |
| | |
| | |

Summary: *Los tipos de datos estructurados no permiten enfrentar problemas más grandes de forma simple.*

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|--------------|-----------------|-------------|
| Harold Mateo | Programación | Carlos Richardo | 10/11/22 |

Title *Arreglos unidimensionales*

| Keyword | Topic |
|--|---|
| <ul style="list-style-type: none"> Almacén Elemento Índice | <p><i>Arreglos unidimensionales.</i></p> <p><i>Esto permite almacenar</i></p> <p><i>N elementos del mismo tipo</i></p> <p><i>y acceder a ellos por medio de un índice.</i></p> <p><i>En esto se distinguen dos partes indispensables:</i></p> <p><i>El índice: Encapsula la forma de acceder a cada uno de los elementos.</i></p> |
| <p>Questions</p> <p><i>¿cómo se accede a ellos?</i></p> <p><i>¿A qué se le llama representación?</i></p> | <p><i>Los componentes: Hay una presencia a cada uno de los elementos que se almacenan en cada casilla.</i></p> |

Summary: *permite almacenar y acceder a datos del mismo tipo por su posición.*

| NAME | CLASS | SPEAKER | DATE & TIME |
|---------------|------------|---------|-------------|
| Harold Matteo | Primer año | Richard | 10/11/22 |

Title

Arreglo unidimensional

Keyword

- Espacio
- memoria
- Declarar

Topic

Declaración de arreglo unidimensional.
 En arreglo ocupan espacio memoria, y se le declara a la hora de declarar el arreglo.

| | | | | | |
|--|--|--|--|-----|--|
| | | | | ... | |
|--|--|--|--|-----|--|

↑

↑

↑

Questions

¿Qué espacio ocupan esto?

Primer elemento Segundo elemento

N-ésimo elemento

En la figura superior se muestra un arreglo con 10 elementos de tipo entero, mas, este arreglo pueden ser aun mas grandes.

Summary:

Estos solo ocupan espacio a la hora de ser declarados.

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|------------|------------------|-------------|
| Harold Mateo | Practica 1 | Carlo Richardson | 10/11/22 |

Title: *Arreglos unidimensionales*

| Keyword | Topic |
|--|---|
| <ul style="list-style-type: none"> • Parametro • Cambio • Valor • Original | <p><i>Arreglos y funciones</i></p> <p><i>El lenguaje C usa parámetros por referencia para pasar los arreglos a las funciones, cualquier cambio a los arreglos en las funciones afecta a su valor original.</i></p> <p><i>En la llamada a la función solo se debe incluir el nombre del arreglo que es un apuntador.</i></p> |
| <p>Questions</p> <p><i>¿que cambio puede realizar y en que situaciones?</i></p> | |

Summary: *Esto solo requiere en su nombre para ser llamado, mas cualquier cambio se refleja en los valores originales.*

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|------------------|-------------------|-------------|
| Harold Matto | Primer -mañan | Carlos Richard | 10/11/22 |

Title *Arreglo multidimensional*

| Keyword | Topic |
|---|---|
| <ul style="list-style-type: none"> • Accesos • Índices • Dimensiones | <p><i>Arreglo multidimensional</i></p> <p><i>Estos reciben su nombre debido a que se debe acceder a sus elementos utilizando múltiples índices.</i></p> <p><i>El número de dimensiones del arreglo depende de las características del problema a resolver, teniendo como ejemplo a los arreglos bidimensionales o matrices, los cuales son de los más utilizados.</i></p> |
| Questions | |
| <p><i>¿qué tipo existen?</i></p> <p><i>¿cómo se llaman?</i></p> | |

Summary: *Lo arreglo multidimensional acceden a sus elementos por medio de más de un índice.*

| | | | |
|----------------------|-----------------------|-------------------------|-------------------------|
| NAME Harold Mateo | CLASS Programación | SPEAKER Cano Ricardo | DATE & TIME 10/11/22 |
|----------------------|-----------------------|-------------------------|-------------------------|

Title *Arreglo multidimensional*

| | | | | | | | | | | | | | | | | | |
|--|--|-------------------|-----------------|----------------|--|-------------------|--|-----|--|--|--|-----|--|------------------|--|-----|--|
| Keyword • Tipo • Datos • Indices | Topic <i>Arreglo bidimensional</i> <i>- nales</i> <i>Un arreglo bidimensional</i> <i>permite almacenar N x M</i> <i>elementos del mismo tipo</i> <i>y acceder a ellos. Al igual</i> <i>que los unidimensionales,</i> <i>consta de componentes</i> <i>tres índices.</i> | | | | | | | | | | | | | | | | |
| Questions <i>¿cuántos</i> <i>datos al</i> <i>macenar?</i> | <table border="1"> <tr> <td>Primera Fila →</td> <td></td> <td>...</td> <td></td> </tr> <tr> <td>Segunda Fila →</td> <td></td> <td>...</td> <td></td> </tr> <tr> <td></td> <td></td> <td>...</td> <td></td> </tr> <tr> <td>Última Fila →</td> <td></td> <td>...</td> <td></td> </tr> </table> | Primera Fila → | | ... | | Segunda Fila → | | ... | | | | ... | | Última Fila → | | ... | |
| | Primera Fila → | | ... | | | | | | | | | | | | | | |
| | Segunda Fila → | | ... | | | | | | | | | | | | | | |
| | | | ... | | | | | | | | | | | | | | |
| | Última Fila → | | ... | | | | | | | | | | | | | | |
| <table border="1"> <tr> <td></td> <td>Primera columna</td> <td>Segunda columna</td> <td>Última columna</td> </tr> </table> | | Primera columna | Segunda columna | Última columna | | | | | | | | | | | | | |
| | Primera columna | Segunda columna | Última columna | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | |

Summary: *Esto permite almacenar*
un número de datos del mis-
mo tipo, ordenado

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|------------------------|-------------------|-------------|
| Harald Mator | Práctico multimedia | Carlos Richard | 10/11/22 |

Title *Aceptos multidimensionales*

| Keyword | Topic |
|-------------------|-----------------------------------|
| <i>memoria</i> | <i>Declaración de aceptos</i> |
| <i>defin</i> | <i>bidimensionales</i> |
| <i>definición</i> | <i>El espacio que ocupan los</i> |
| | <i>aceptos en memoria, el</i> |
| | <i>resulta al momento de</i> |
| | <i>realizar la declaración</i> |
| | <i>de los mismos.</i> |
| | <i>Una vez se definen los el-</i> |
| | <i>mentos pueden recibir sus</i> |
| | <i>valores a través de mul-</i> |
| | <i>tiples asignaciones, o por</i> |
| | <i>medio de un ciclo de lu-</i> |
| | <i>tura de valores.</i> |
| Questions | |
| <i>¿cuando</i> | |
| <i>ocupan</i> | |
| <i>memoria?</i> | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |
| | |

Summary: *¿alo ocupan memoria al*
se declarados? reciben sus valores
a través de asignaciones.

| NAME | CLASS | SPEAKER | DATE & TIME |
|--------------|--------------------|---------|-------------|
| Harold Motto | Reserva materna | Richard | 10/11/22 |

Title *Arreglos multidimensionales*

Keyword

*caras
varias
indices*

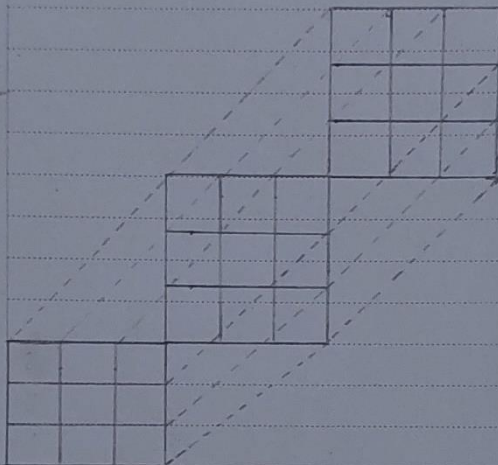
Topic

*Arreglos de mas de
de dimensiones.*

*Esto es similar a
las bidimensionales, con
la diferencia de estar
hecho con caras, y cada
elemento se debe referir
por tres o mas indi-
ces, siendo lo de tres, he-
cho "tridimensional".*

Questions

*¿cuantas
caras po-
den tener?*



Summary:

*Esto son varias caras de
arreglos de una dimension en
terior.*