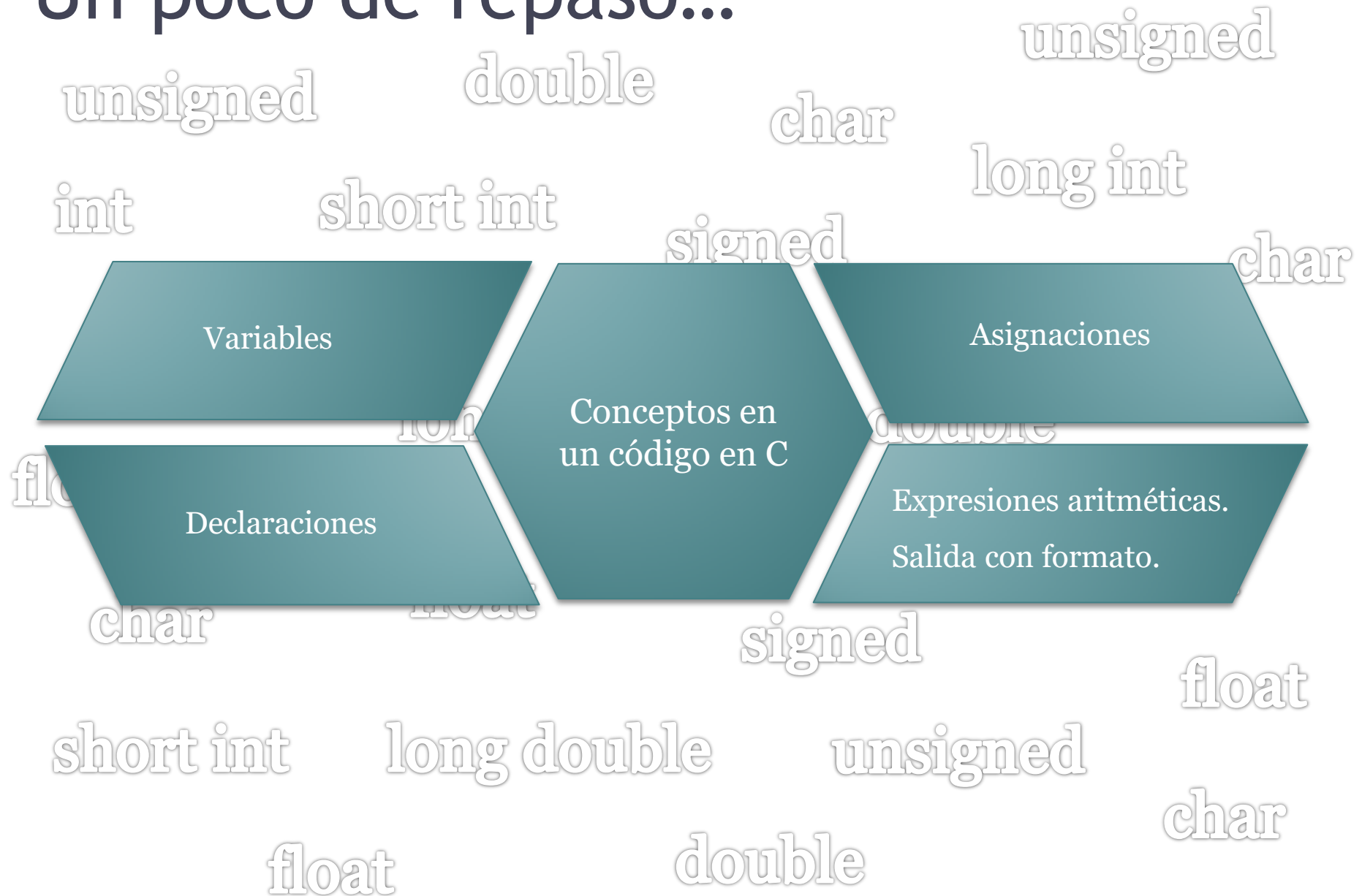


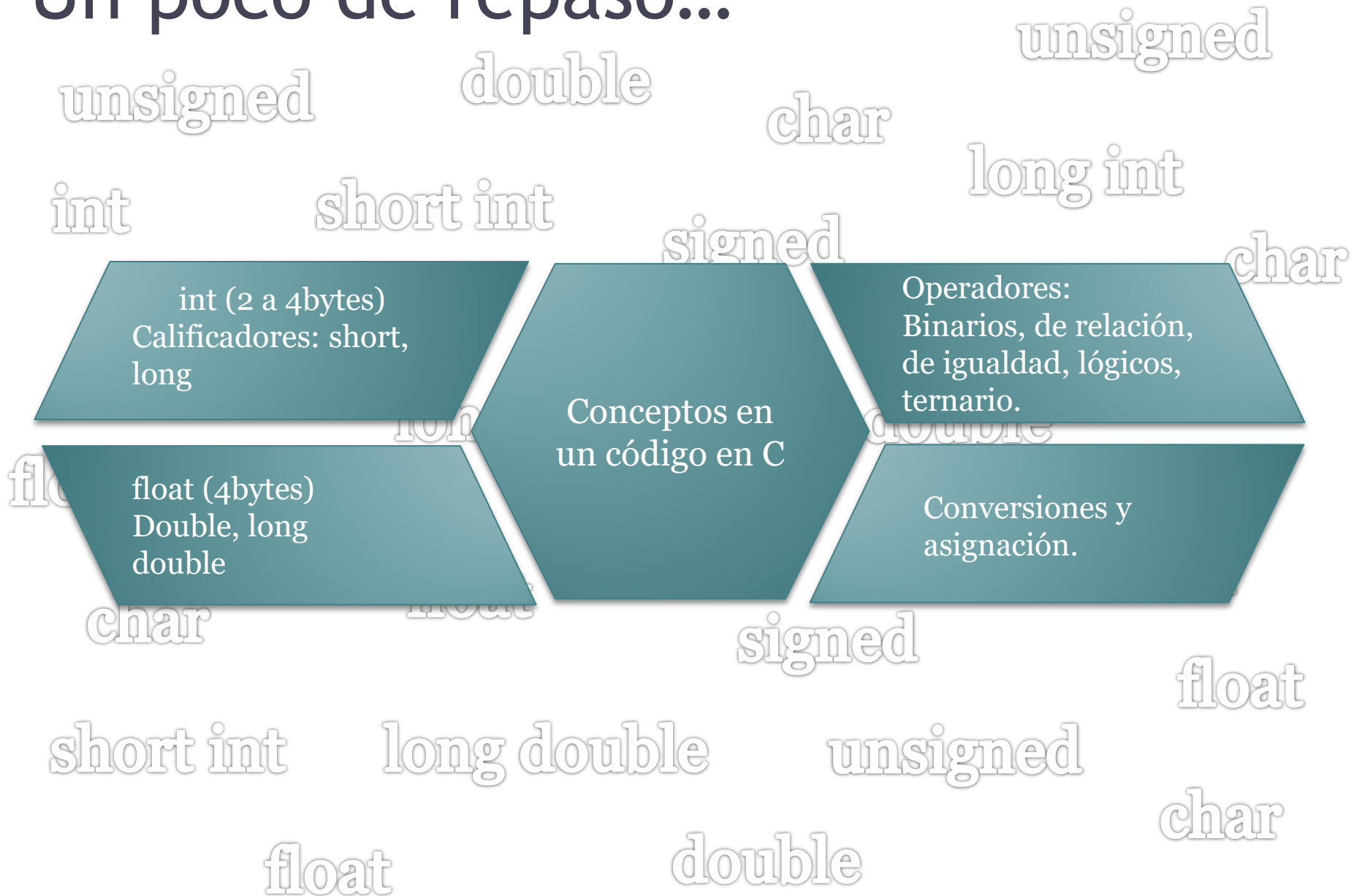
PROGRAMACIÓN

Unidad 3: Lenguaje C .Concepto de tipo de dato.
Tipos, operadores, expresiones, calificadores,
conversiones

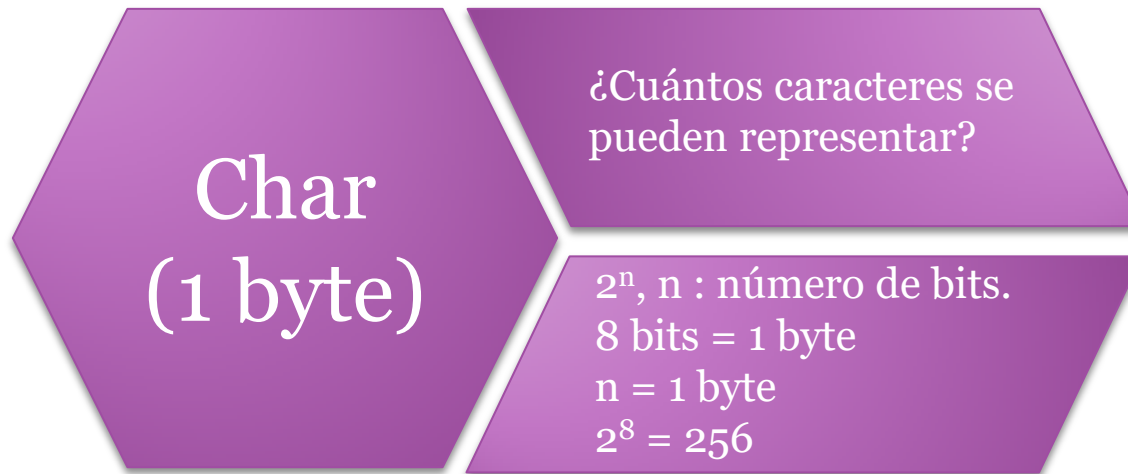
Un poco de repaso...



Un poco de repaso...



Tipo de dato carácter



Un poco de información previa



Código ASCII: código estándar para el intercambio de Información

ASCII (American Standard

Code for Information Interchange Estándar)

Creado en 1963 por el Comité Estadounidense de Estándares este organismo cambio su nombre en 1969 por "Instituto Estadounidense de Estándares Nacionales" o "ANSI" como se lo conoce desde entonces.

¿Por qué hace falta la conversión?

Cada procesador de palabras codifica textos, números y los datos necesarios para el formato de márgenes, paginación, etc.

En cambio, un texto en ASCII es simplemente un texto sin atributos de ninguna especie y se rige por una tabla universal.

La opción “Sólo texto” de los procesadores de palabras mas populares transforma el archivo en un texto ASCII.

El código ASCII

- Establece un criterio de orden:

- cada signo
- cada número
- cada letra
- cada carácter de control



Código numérico

Tabla de Código ASCII

Caracteres ASCII de control

¡	barra invertida (alt + 92)
@	arroba (alt + 64)
ñ	eñe minúscula (alt + 164)
'	comilla simple, apóstrofe (alt + 39)
#	signo numeral (alt + 35)
!	signo de admiración (alt + 33)
_	guión bajo, subrayado (alt + 95)
*	asterisco (alt + 42)
~	equivalencia, tilde (alt + 126)
-	guión medio (alt + 45)

32	espacio	64	@	96	`
33	!	65	A	97	a
34	"	66	B	98	b
35	#	67	C	99	c
36	\$	68	D	100	d
37	%	69	E	101	e
38	&	70	F	102	f
39	'	71	G	103	g
40	(72	H	104	h
41)	73	I	105	i
42	*	74	J	106	j
43	+	75	K	107	k
44	,	76	L	108	l
45	-	77	M	109	m
46	.	78	N	110	n
47	/	79	O	111	o
48	0	80	P	112	p
49	1	81	Q	113	q
50	2	82	R	114	r
51	3	83	S	115	s
52	4	84	T	116	t
53	5	85	U	117	u
54	6	86	V	118	v
55	7	87	W	119	w
56	8	88	X	120	x
57	9	89	Y	121	y
58	:	90	Z	122	z
59	;	91	[123	{
60	<	92	\	124	
61	=	93]	125	}
62	>	94	^	126	~
63	?	95	_		

Caracteres ASCII imprimibles

¿Cuál es el objetivo del uso de ASCII?

En concreto, el **ASCII** establece una correspondencia entre estas cadenas y los símbolos que representa (como letras y números). Esto hace que se **facilite el almacenamiento y el procesamiento de información en una computadora** y la comunicación entre diferentes aparatos digitales.

El uso de un código de caracteres en particular resulta extremadamente útil para evitar la pérdida de información como consecuencia de la incompatibilidad entre diferentes programas.

Retomemos: Tipo de dato carácter

2^n , n : número de bits.

8 bits = 1 byte

$n = 1$ byte

$2^8 = 256$

La tabla de
caracteres ASCII
básica tiene 256
caracteres.

Tipo de dato carácter

Las variables de tipo char almacenan 1 solo carácter.

Declaración de una variable de tipo carácter:

```
char letra;
```

Declaración e inicialización: `char letra ='a';`

ASCII, caracteres, valores enteros

Una constante de caracteres un entero, escrita entre ‘ ’ (comillas simples).

char letra = ‘a’; // 97 en ASCII

char digito = ‘0’; // 48 en ASCII

¿Operaciones permitidas ?

Las constantes de carácter pueden participar de operaciones aritméticas (por su “respaldo” ASCII), aunque se utilizan mas en comparaciones con otros caracteres.

Algunos caracteres se representan como constantes de cadena: \n
Son 2 caracteres que representan uno solo (Secuencias de escape).

Leer y escribir un carácter

Con formato %c, en una invocación a la función scanf.

```
char letra;  
scanf("%c",&letra);
```

```
Letra=getchar();  
putchar(letra);
```

Con formato %c, en una invocación a la función printf.

```
char letra;  
printf("%c",letra);
```

De las operaciones permitidas

```
int main(void)
{
    char indice;

    setvbuf(stdout, NULL, _IONBF, 0);

    for (indice=122; indice >=97; indice--)

        putchar(indice);

    return 0;
}
```

¿Qué resultado obtenemos de este programa?

Estructura de selección

Hasta ahora los programas se ejecutaban por un secuencia ordenada de proposiciones, cada una de ellas se ejecuta una vez o un número fijo de veces.

Cuando estudiaron diseño vieron la estructura Si-sino:

SI condición ENTONCES

Acción 1;

SINO

Acción 2;

Estructura de selección

En el lenguaje C se dispone de sentencias condicionales: if-else, switch que permite seleccionar una única alternativa entre varias para ser ejecutada.

Sentencia if-else

Esta sentencia permite seleccionar el grupo de sentencias o bloques que serán ejecutadas, de acuerdo al valor de una condición.

Formalmente, la expresión:

```
if(expresión)
{
    proposición 1;
}
else{
    proposición 2;
}
```

Sentencia if-else

Expresión: es una expresión que tiene que ir entre () y tiene que ser evaluada como VERDADERA (distinto de cero) o FALSO (CERO).

Proposición: puede ser simple o compuesta { }.

La semántica de una sentencia **if** es:

- **Con else:** se evalúa la expresión, si es verdadera, se lleva a cabo la proposición 1, sino si es falsa, se ejecuta la proposición 2.
- **Sin else:** se evalúa la expresión, si es verdadera se ejecuta la proposición 1

Ejercicio 1:

Escribir un algoritmo que lea una línea de texto que termina con un punto y determine la cantidad de consonantes minúsculas que hay en la misma. El algoritmo debe mostrar la cantidad obtenida. Usar código ASCII

Ejercicio 2:

Dado un conjunto de n estudiantes, se quiere saber el número de desaprobados y la nota promedio de los aprobados.

if anidados

```
if (expresión 1)
    proposición 1;
else
    if (expresión 2)
        proposición 2;
    else
        if (expresión 2)
            proposición 2;
        else
```

Es la forma general de escribir una decisión múltiple.

Las expresiones se evalúan en orden, si cualquier expresión es verdadera, se ejecuta la proposición asociada a ella.

Sentencia Switch

```
SEGÚN variable
    const 1: A1;
    const 2: A2;
    ...
SINO
    An
```

```
Switch(expresión)
{
    case const1: sent1;
                    break;
    case const2: sent2;
                    break;
    ...
    default: sentencias;
}
```


Sentencia Switch

Expresión: la expresión puede ser una expresión entera, un carácter alfanumérico, una constante o una variable.

Const.: puede ser una expresión entera, un carácter alfanumérico, una constante o una variable.

Sent.: puede ser cualquier bloque de sentencias de C. Si el bloque consta de una sola sentencia no es necesario incluir las llaves de apertura y cierre. De igual manera se recomienda incluirlas.

default (case predeterminado): es opcional. En general se inserta la línea default al final del cuerpo de switch cuando se estima que el case predeterminado va a ocurrir con mayor frecuencia.

Break: esta sentencia provoca una salida inmediata del switch. Puesto que los case sirven sólo como etiquetas, después de que se ejecutan el código para uno el código pasa al siguiente. Las formas más comunes de dejar un switch son **break** y **return**

Ejercicio:

Diseñar un algoritmo que determine que pida al usuario un número de día, un número de mes y un año. Evalúe si la entrada es correcta y en caso de que así sea calcule el día del año que es. Tenga en cuenta que el año puede ser bisiesto.

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    printf("Hasta la próxima clase!!\n");
```

```
    return 0;
```

```
}
```