60. Actividad

De forma individual investiga:

1. Palabras reservadas de C++

__abstract [2]	__alignof	__asm	__assume
__based	__box [2]	__cdecl	__declspec
__delegate [2]	__event	__except	__fastcall
__finally	__forceinline	__gc [2]	__hook [3]
__identifier	__if_exists	__if_not_exists	__inline
__int8	__int16	__int32	__int64
__interface	__leave	__m64	__m128
__m128d	__m128i	__multiple_inheritance	__nogc [2]
__noop	__pin [2]	__property [2]	__raise
__sealed [2]	__single_inheritance	__stdcall	__super
__try_cast [2]	__try/__except,	__unhook [3]	__uuidof

__value [2]	__try/__finally	__w64	bool
break	__virtual_inheritance	catch	char
class	case	const_cast	continue
default	const	deprecated [1]	dllexport [1]
dllimport [1]	delete	double	dynamic_cast
else	do	explicit	extern
false	enum	for	friend
goto	float	inline	int
long	if	naked [1]	namespace
new	mutable	noreturn [1]	nothrow [1]
novtable [1]	noinline [1]	private	property [1]
protected	operator	register	reinterpret_cast
return	public	short	signed
sizeof	selectany [1]	static_cast	struct
switch	static	this	thread [1]
throw	template	try	typedef
typeid	true	union	unsigned
using declaracion, using directiva	typename	virtual	void
volatile	uuid [1]	while	
	wchar t, wchar t		

2. Reglas para la construcción de Identificadores en C++

1. Debe empezar por una letra o carácter de subrayado.

   `natural` `_var2` `alumno_eii` `x1`

   A pesar de estar permitido, es recomendable no usar el subrayado al inicio, pues es utilizado por los desarrolladores de bibliotecas y compiladores y, eventualmente, podemos generar una colisión.

2. Las mayúsculas son caracteres distintos que las minúsculas.

   `interes` e `Interes` son variables distintas.

3. No pueden utilizarse las palabras reservadas del lenguaje, como `double`, `false`, etc.
4. No deben llevar comas, puntos, acentos, espacios en blanco, la letra `ñ`, etc.

   `tamaño` `var 1` `interés` `dia.24` no son válidos.

3. Operadores de C++

OPERADORES ARITMETICOS

\+ suma
– resta
* multiplicación
/ división
% mod o residuo

El simbolo / (slash) se utiliza para la division real y el operador % (mod) representa el resto de la división entera

OPERADORES RELACIONALES

< menor que
> mayor que
<= menor o igual
>= mayor o igual
== igual
!= diferente

OPERADORES LOGICOS

&& and (y lógica)
|| or (ó lógico)
! not (negación)

OPERADORES DE ASIGNACION

= igual
+= mas igual
-=menos igual
*= por igual
/= dividido igual

¿Qué representan las siguientes expresiones regulares?

L = [a-zA-Z_]+

- Matchea una vez o más, letras mayúsculas o minúsculas o guiones bajos.

D=[0-9]+

- Hace match de uno o más números.

E=[,\t,\r,\n]+

- Acepta uno o más espacios.

Notas: Las llaves “{ }” además de utilizarse con cuantificadores se utilizan para hacer referencia a una expresión regular previamente definida. Los paréntesis “( )” permiten agrupar y las comillas permite agregar un símbolo que no esta considerado

(L)({L}|{D})*

- Busca letras iniciales y después seguidas por cero o más letras o números.

("-"{D}+)|{D}+

- Matchea uno o más números positivos o negativos.

“+”

- Encuentra el caracter más.

while

- Encuentra las palabras que son while exactamente.

Utiliza las expresiones regulares para llenar la siguiente tabla:

Token	Lexema	Patrón	Reservada (SI/NO)
Letras	Hola, alo, jaja	[a-zA-Z_]+	No
Número	0, 12, 123	[0-9]+	No
Espacios	“ “, “ “	[,\t,\r,\n]+	No
Variable	Var, vel, vel1, vel10	(L)({L}|{D})*	No
Número	10, 1, -2, 5	("-"{D}+)|{D}+	No
Operador suma	+	“+”	Sí
while	while	while	Sí