

Material para a Formación Profesional inicial

A02. Estructuras de control repetitivas

Familia profesional	IFC	Informática e comunicacións
Ciclo formativo	CSIFC02 CSIFC03	Desenvolvemento de aplicacións multiplataforma Desenvolvemento de aplicacións web
Grao		Superior
Módulo profesional	MP0485	Programación
Unidade didáctica	UD02	Uso de estruturas de control.
Actividade	A02	Estructuras de control repetitivas.
Autores		Silvia Framiñán Fondevila Marta Rey López
Nome do arquivo		CSIF02_MP0485_V000202_UD02_A02_Estructuras_control_repeti- tivas.odt

© 2017 Xunta de Galicia.

Consellería de Cultura, Educación e Ordenación Universitaria.

Este traballo foi realizado durante unha licenza de formación retribuída pola Consellería de Cultura, Educación e Ordenación Universitaria e ten licenza Creative Commons BY-NC-SA (recoñecemento - non comercial - compartir igual). Para ver unha copia desta licenza, visitar a ligazón <http://creativecommons.org/licenses/by-nc-sa/3.0/es/>.

1.	<u>Ficha técnica</u>	6
	<u>Contexto da actividade</u>	6
	<u>Título da actividade</u>	7
	<u>Resultados de aprendizaxe do currículo</u>	7
	<u>Obxectivos didácticos e título e descrición da actividade</u>	7
	<u>Criterios de avaliación</u>	7
	<u>Contidos</u>	7
	<u>Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación (exemplo)</u>	8
2.	<u>A02. Estruturas de control repetitivas</u>	9
2.1	<u>Introdución</u>	9
2.2	<u>Actividade</u>	9
2.2.1	<u>Definición de estrutura de repetición</u>	9
2.2.2	<u>Estruturas de control repetitivas</u>	9
2.2.2.1	<u>Bucle For</u>	9
	<u>Diagrama de fluxo</u>	9
	<u>Sintaxe en Java</u>	10
	<u>Exemplo</u>	11
2.2.2.2	<u>Bucle While</u>	11
	<u>Diagrama de fluxo</u>	12
	<u>Sintaxe en Java</u>	12
	<u>Exemplo</u>	12
2.2.2.3	<u>Do-while</u>	13
	<u>Diagrama de fluxo</u>	13
	<u>Sintaxe en Java</u>	13
	<u>Exemplo</u>	13
2.3	<u>Tarefas</u>	14
2.3.1	<u>Tarefa 2.1. Utilizar o bucle for</u>	14
	<u>Enunciado</u>	14
	<u>Solución</u>	14
2.3.2	<u>Tarefa 2.2. Utilizar o bucle while</u>	14
	<u>Enunciado</u>	14
	<u>Solución</u>	14
2.3.3	<u>Tarefa 2.3. Utilizar o bucle do-while</u>	15
	<u>Enunciado</u>	15
	<u>Solución</u>	15
2.3.4	<u>Tarefa 2.4. Combinar as estruturas repetitivas</u>	15
	<u>Enunciado</u>	15
	<u>Solución</u>	16
3.	<u>Materiais</u>	17
3.1	<u>Documentos de apoio ou referencia</u>	17
3.2	<u>Recursos didácticos</u>	17

4.	<u>Avaliación</u>	18
4.1	<u>Modelo de proba</u>	18
	<u>Enunciado</u>	18
	<u>Solución</u>	18
	<u>Táboa de indicadores</u>	19
	<u>OU 3. Táboa de indicadores sobre un proxecto Java no que se utilicen estruturas de repetición</u>	19
	<u>OU 4. Táboa de indicadores sobre un proxecto Java no que se recoñezan as posibilidades das sentenzas de salto</u>	19
	<u>OU 5. Táboa de indicadores sobre un proxecto Java no que se creen programas executables utilizando diversas estruturas de control repetitivas</u>	19

1. Ficha técnica

Contexto da actividade

Módulo	Duración	Unidade didáctica.	Sesións 60'	Actividades	Sesións 60'
MP0485. Programación.	240	UD01. Identificación dos elementos dun programa informático.	30	A01. Metodoloxía da programación.	5
				A02. Introducción á linguaxe de programación Java.	7
				A03. Elementos básicos dun programa Java.	12
		UD02. Uso de estruturas de control.	20	A01. Estruturas de control selectivas.	8
				A02. Estruturas de control repetitivas.	12
		UD03. Introducción á POO.	28	A01. Conceptos de POO.	8
				A02. Clases e obxectos.	20
		UD04. Conceptos avanzados de POO.	20	A01. Obxectos predefinidos e métodos estáticos.	10
				A02. Constantes e métodos estáticos.	5
				A03. Visibilidade e empaquetaxe.	5
		UD05. Lectura e escritura de información.	30	A01. Fluxos de entrada e saída.	4
				A02. Uso de ficheiros.	15
				A03. Interfaces gráficas de usuario.	15
		UD06. Aplicación das estruturas de almacenamento.	35	A01. Uso de cadeas.	8
				A02. Uso de arrays.	9
				A03. Uso de coleccións e outras estruturas	12
				A04. Manipulación de documentos XML.	8
		UD07. Xerarquías de clases e excepcións.	37	A01. Introducción á herdanza.	10
				A02. Uso avanzado da herdanza.	10
				A03. Interfaces.	10
				A04. Control de código. Excepcións.	7
		UD08. Mantemento da persistencia dos obxectos.	18	A01. Instalación dun SXBDOO e almacenamento básico de obxectos.	8
				A02. Almacenamento e recuperación de obxectos nun SXBDOO e operacións con datos complexos.	10
		UD09. Xestión dos datos almacenados nas bases de datos relacionais.	22	A01. Conexión con sistemas xestores de bases de datos relacionais.	6
				A02. Operacións de lectura nunha base de datos relacional.	7
				A03. Operacións de escritura nunha base de datos relacional e uso de transaccións.	9

NOTA: Esta actividade está vinculada á programación recollida no arquivo CSIFC02_MP0485_V000200_UD02_Uso_estructuras_control.pdf

Título da actividade

Nº	Título	Descrición	Duración
A02	Estruturas de control repetitivas.	Nesta actividade aprenderase a resolver problemas que impliquen a repetición de acción. Para elo planificarase o algoritmo e implementárase posteriormente empregando a estrutura máis apropiada.	12

Resultados de aprendizaxe do currículo

Resultados de aprendizaxe do currículo	Completo
▪ RA3. Escribe e depura código, para o que analiza e utiliza as estruturas de control da linguaxe.	NON

Obxectivos didácticos e título e descrición da actividade

Obxectivos específicos		Actividade		Descrición básica	Duración
O2.1	Utilizar estruturas de repetición.	A02	Estruturas de control repetitivas.	Nesta actividade aprenderase a resolver problemas que impliquen a repetición de acción. Para elo planificarase o algoritmo e implementárase posteriormente empregando a estrutura máis apropiada.	12
O2.2	Crear programas executables utilizando diversas estruturas de control.				
O2.3	Recoñecer as posibilidades das sentenzas de salto.				

Criterios de avaliación

Criterios de avaliación
▪ CA3.2. Utilizáronse estruturas de repetición.
▪ CA3.3 Recoñecéronse as posibilidades das sentenzas de salto.
▪ CA3.5.2. Creáronse programas executables utilizando diversas estruturas de control repetitivas.

Contidos

Contidos
▪ BC3. Uso de estruturas de control: <ul style="list-style-type: none">– Bloques de instrucións.– Estruturas de repetición.

Actividades de ensino e aprendizaxe e de avaliación, métodos, recursos e instrumentos de avaliación (exemplo)

Que e para que	Como			Con que	Como e con que se valora	Sesións
Actividade (título e descrición)	Profesorado (en termos de tarefas)	Alumnado (tarefas)	Resultados ou produtos	Recursos	Instrumentos e procedementos de avaliación	12
A02. Estructuras de control repetitivas. <ul style="list-style-type: none"> Nesta actividade aprenderase a resolver problemas que impliquen a repetición de acción. Para elo planificarase o algoritmo e implementárase posteriormente empregando a estrutura máis apropiada. 	<ul style="list-style-type: none"> Tp2.1. Introducción sobre as estruturas de repetición e as súas vantaxes na programación. Tp2.2. Explicación da estrutura repetitiva for. 	<ul style="list-style-type: none"> Ta2.1. Codificación do bucle for. 	<ul style="list-style-type: none"> Código con bucle for. 	<ul style="list-style-type: none"> Ordenador persoal con conexión a Internet. Contorno de desenvolvemento NetBeans. Java SE Development Kit. Apuntamentos da profesora. Proxector. 		9
	<ul style="list-style-type: none"> Tp2.3. Explicación da estrutura repetitiva while. 	<ul style="list-style-type: none"> Ta2.2. Codificación do bucle while. 	<ul style="list-style-type: none"> Código con bucle while. 			
	<ul style="list-style-type: none"> Tp2.4. Explicación da estrutura repetitiva do-while. Comparación coa estrutura while 	<ul style="list-style-type: none"> Ta2.3. Codificación do bucle do-while. Ta2.4. Combinar as estruturas repetitivas 	<ul style="list-style-type: none"> Código con bucle do-while. Proxecto Java 			
		<ul style="list-style-type: none"> Ta2.5 - Tarefa de avaliación, combinando: <ul style="list-style-type: none"> Táboa de indicadores sobre un proxecto Java no que se utilicen estruturas de repetición. Táboa de indicadores sobre un proxecto Java no que se creen programas executables utilizando diversas estruturas de control. Táboa de indicadores sobre un proxecto Java no que se proben e depuren os programas nos que se empregan control de código e excepcións 	<ul style="list-style-type: none"> Proxecto Java. 		<ul style="list-style-type: none"> OU 2. Táboa de indicadores sobre un proxecto Java no que se utilicen estruturas de repetición. OU 3. Táboa de indicadores sobre un proxecto Java no que se creen programas executables utilizando diversas estruturas de control. OU 5. Táboa de indicadores sobre un proxecto Java no que se proben e depuren os programas nos que se empregan control de código e excepcións 	3

2. A02. Estructuras de control repetitivas.

2.1 Introducción

Na actividade que nos ocupa aprenderanse os seguintes conceptos e manexo de destrezas sobre a Programación Orientada a Obxectos:

- Utilidade e tipos de estruturas repetitivas ou bucles
- Definición e uso da estrutura repetitiva **for**
- Definición e uso da estrutura repetitiva **while**
- Definición e uso da estrutura repetitiva **do-while**. Comparación co bucle *while*.

2.2 Actividade

2.2.1 Definición de estrutura de repetición

Durante a execución dun programa xurde a necesidade de variar o seu fluxo de control para repetir accións ou cálculos.

Unha estrutura de repetición ou bucle permítenos executar un número de veces determinado un bloque específico de instrucións. As instrucións repítense mentres ou ata que se cumpra unha determinada condición. Esta condición coñécese como condición de saída. A maior utilidade é evitar a repetición de liñas de código iguais.

Tipos de estruturas repetitivas:

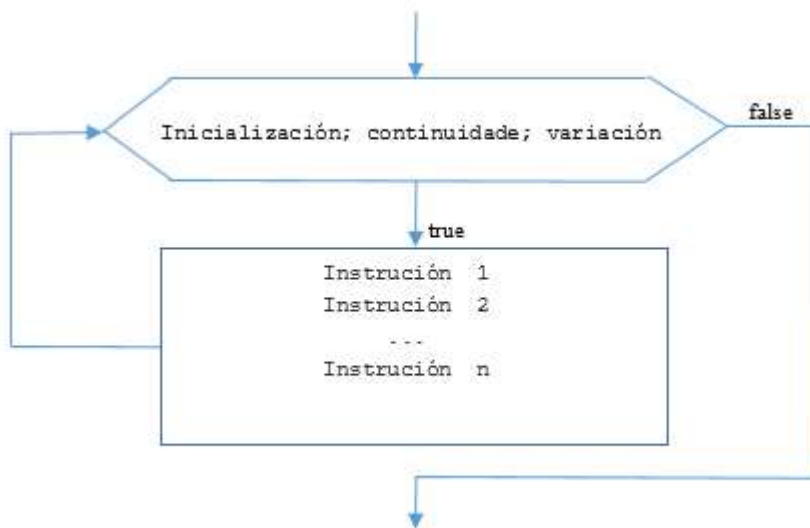
- bucle **for**
- bucle **while**
- bucle **do-while**

2.2.2 Estructuras de control repetitivas

2.2.2.1 Bucle For

Cando desexamos executar un grupo de instrucións un número determinado de veces, a instrución **for** é a que mellor se adapta a esta tarefa.

Diagrama de fluxo



Sintaxe en Java

A sintaxe desta instrución é:

```
for (iniciación; condición de continuidade; expresión de variación){
    Instrucións a executar de forma repetitiva
}
```

A semántica (significado) da instrución é a seguinte: inicialízase unha variable, avalíase a condición de continuidade e, se se cumpre, execútanse as instrucións situadas entre as chaves, finalmente execútase a expresión de variación e repítese o ciclo ata que a condición de continuidade avalíase como false. Cando saímos do bucle, a variable inicializada non existe.

Exemplo:

```
public class BucleFor {
    public static void main (String[] args) {
        int i;
        for (i=1;i<=4;i++)    System.out.println(i);
    }
}
```

No exemplo anterior (BucleFor1) declárase unha variable de tipo int na liña 3, que se utiliza na liña 4 (for). O bucle for contén as súas tres seccións obrigatorias:

- Inicialización: i=1
- Condición de continuidade: i<=4
- Expresión de incremento: i=i+1

Desta maneira o bucle execútase da seguinte forma:

- Primeira iteración i almacena o valor 1 i<=4 avalíase como true Imprímese: 1
- Segunda iteración i almacena o valor 2 i<=4 avalíase como true Imprímese: 2
- Terceira iteración i almacena o valor 3 i<=4 avalíase como true Imprímese: 3
- Cuarta iteración i almacena o valor 4 i<=4 avalíase como true Imprímese: 4
- Finalización i almacena o valor 5 i<=4 avalíase como false -----

Exemplo

Realiza unha aplicación que nos pida un número de vendas a introducir, despois pediranos tantas vendas por teclado como número de vendas indicáronse. Ao final mostrase a suma de todas as vendas.

```
public class VendasApp {
    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);
        System.out.println("Introduce o número de vendas: ");
        int numVendas=sc.nextInt();

        /* Declaramos unha variable para sumar as vendas,
        * se o declaramos dentro do bucle, non poderíamos
        * acceder fóra do bucle á suma das vendas.
        */
        int sumaVendas=0;

        for (int i=0;i<numVendas;i++){
            // indico o numero de venda
            System.out.println("Introduce o prezo da venda "+(i+1));
            int venda=sc.nextInt();

            // Acumulamos o valor da venda
            sumaVendas=sumaVendas+venda;
        }
        System.out.println(sumaVendas);
    }
}
```

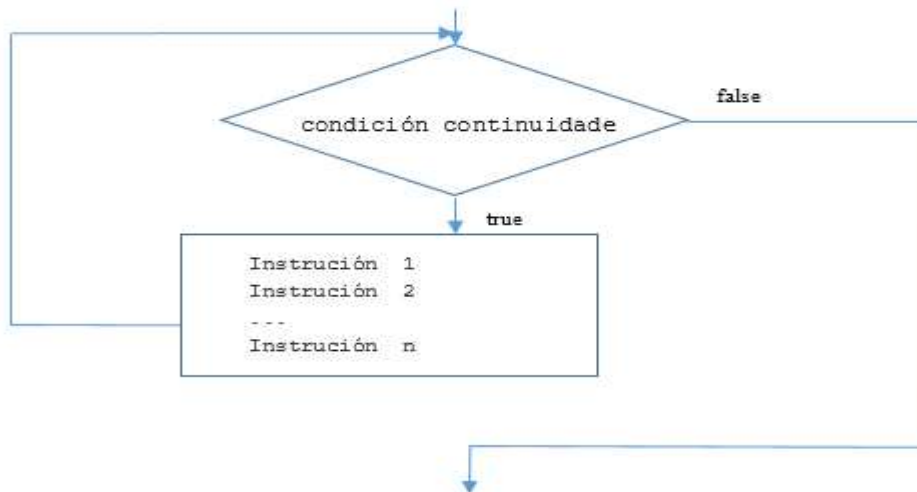


Realiza a Tarefa 2.1. **Utilizar o bucle for**, onde o alumnado resolverá o exercicio utilizando a estrutura de repetición *for*.

2.2.2.2 Bucle While

O bucle **while** permítenos repetir a execución dunha serie de instrucións mentres que se cumpra unha condición de continuidade. O seu uso resulta recomendable cando non coñecemos a priori o número de iteracións que debemos realizar.

Diagrama de fluxo



Sintaxe en Java

```
while (condición de continuidade) {
    Instrucciones a executar de forma repetitiva
}
```

Itérase mentres que a “condición de continuidade” se cumpra, abandonándose o bucle cando a condición avalíase como false. Pode ocorrer que as instrucións interiores do bucle nunca se executen, se a primeira vez que se avalía a condición resulta false.

É importante que a variable que estea na condición varíe para que non se produza un bucle infinito.

Exemplo

Imprimir os números naturais desde 1 ata un número introducido por teclado.

```
1 public class BucleWhile {
2     public static void main (String[] args) {
3         Scanner sc = new Scanner(System.in);
4         System.out.println("Introduce o número de vendas: ");
5         int num=sc.nextInt();
6         int i=1;
7         while (i<=num) {
8             System.out.println(i);
9             i++;
10        }
11    }
12 }
```

En BucleWhile declárase unha variable de tipo int na liña 6 e inicialízase a 1; esta variable actuará como contador de iteracións no bucle. Na liña 7 establécese a condición de continuidade do bucle. A liña 8 encárgase de imprimir o valor do índice e a liña 9 de incrementalo. Nótese como se están codificando as distintas expresións e condicións do bucle for: for (i=1; i<=num;i++), nas liñas 6, 7 e 9. Un erro moi frecuente cando se codifica un bucle while é esquecer incrementar o contador (liña 9), xerando un bucle infinito. No noso exemplo tamén crearíamos un bucle infinito se esquecésemos as chaves delimitadoras do ámbito do bucle.

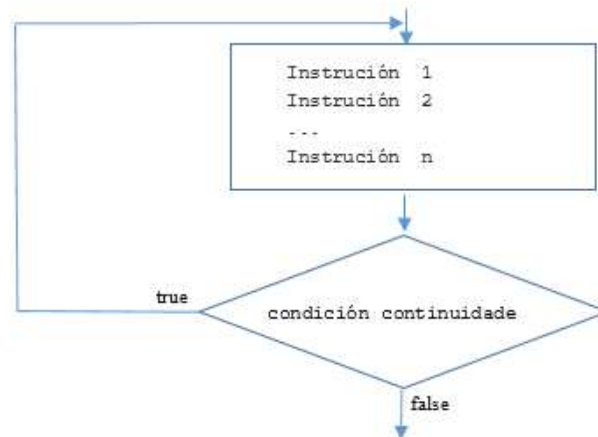


Realiza a Tarefa 2.2. **Utilizar o bucle while**, onde o alumnado resolverá o exercicio utilizando a estrutura de repetición *while*.

2.2.2.3 Do-while

É a forma alternativa de utilizar un bucle while. A diferenza en execución é que a condición de continuidade avalíase ao final no canto do comezo do bucle, polo que as instrucións interiores ao bucle execútanse polo menos unha vez.

Diagrama de fluxo



Sintaxe en Java

```
do {  
    Instruccións a executar de forma repetitiva  
} while (condición de continuidade);
```

Itérase mentres que a “condición de continuidade” se cumpra, abandonándose o bucle cando a condición avalíase como false. É importante que a variable que estea na condición varíe para que non se produza un bucle infinito.

Exemplo

Cantos números naturais (1, 2, 3, 4...) debemos sumar en secuencia para obter polo menos un valor de 100000?, é dicir: ata que valor chegará o sumatorio 1+2+3+4+5+..... para que a suma alcance o valor 100000?

```
1 public class BucleDoWhile {  
2     public static void main (String[] args) {  
3         int Suma=0;  
4         int i=0;  
5         do {  
6             i++;  
7             Suma = Suma + i;  
8         } while (Suma<100000);  
9         System.out.println(i);  
10    }  
11 }
```



Realiza a Tarefa 2.3. **Utilizar o bucle do-while**, onde o alumnado resolverá o exercicio utilizando a estrutura de repetición *do-while*.



Realiza a Tarefa 2.4. **Utilizar as estruturas repetitivas**, onde o alumnado resolverá o exercicio combinando as estruturas de repetición.

2.3 Tarefas

As tarefas propostas para esta actividade son as seguintes:

- Tarefa 2.1. **Utilizar o bucle for.** Nesta tarefa o alumnado resolverá un exercicio utilizando a estrutura de repetición *for*.
- Tarefa 2.2. **Utilizar o bucle while.** Nesta tarefa o alumnado resolverá un exercicio utilizando a estrutura de repetición *while*.
- Tarefa 2.3. **Utilizar o bucle do-while.** Nesta tarefa o alumnado resolverá un exercicio utilizando a estrutura de repetición *do-while*.
- Tarefa 2.4. **Combinar as estruturas repetitivas.** Nesta tarefa o alumnado resolverá un exercicio combinando as estruturas de repetición.

2.3.1 Tarefa 2.1. Utilizar o bucle for

Nesta tarefa o alumnado resolverá o exercicio utilizando a estrutura de repetición **for**.

Enunciado

Crea o código en Java que amose o produto dos 10 primeiros números impares utilizando un bucle **for**.

Solución

Unha posible solución sería:

```
public class Forproductoimpares {  
  
    public static void main(String[] args) {  
        int total = 1;  
        int imp = 1;  
        for (int i=0; i<10; i++ ) {  
            total *= imp;  
            imp += 2;  
        }  
        System.out.println(total);  
    }  
}
```

2.3.2 Tarefa 2.2. Utilizar o bucle while

Nesta tarefa o alumnado utilizará a estrutura repetitiva **while** para resolver o exercicio.

Enunciado

Codificar en Java un programa que permita pedir a altura de *n* persoas e mostre a súa media.

Solución

```
public class Alturas {  
    public static void main( String[] args) {  
        Scanner sc= new Scanner( System. in);
```

```

        int i=0;
        int n;
        int alturaf=0;
        int altura;
        float media=0;

        System.out.println("Cantas alturas queres introducir?");
        n= sc. NextInt();
        while (i!= n) {
            System.out.println("Introduce unha altura en cm");
            altura= sc. nextInt();
            alturaf= alturaf+altura;
            i++;
        }
        If (n!=0)
            media= alturaf/ n;

        System.out.println("Das "+ n+" alturas que metiches, a altura media é
        "+media);
    }
}

```

2.3.3 Tarefa 2.3. Utilizar o bucle do-while

Nesta tarefa o alumnado utilizará a estrutura repetitiva **do-while** para resolver o exercicio.

Enunciado

Realizar o código en Java para amosar os múltiplos de 8 ata un valor dado polo usuario. Debe aparecer en pantalla 8 - 16 - 24, etc.

Solución

```

import java.util.Scanner;

public class Multiplos8 {
    public static void main(String[] ar) {
        Scanner teclado=new Scanner(System.in);
        int mult8;
        mult8=8;
        System.out.print("Dime o valor máximo para os múltiplos de 8?");
        n=teclado.nextInt();
        do {
            System.out.print(mult8);
            System.out.print(" - ");
            mult8=mult8 + 8;
        }while (mult8<=500)
    }
}

```

2.3.4 Tarefa 2.4. Combinar as estruturas repetitivas

Nesta tarefa o alumnado resolverá o exercicio combinando as estruturas de repetición.

Enunciado

Realizar un proxecto Java que pida por teclado a data de nacemento dunha persoa (día, mes, ano) e calcule o seu número da sorte.

O número da sorte calcúlase sumando o día, mes e ano da data de nacemento e a conti-

nuación sumando as cifras obtidas na suma.

Por exemplo, se a data de nacemento é 12/07/1980, calculamos o número da sorte así:
 $12+7+1980 = 1999$ $1+9+9+9 = 28$

Número da sorte: 28

Solución

```
import java.util.Scanner;

public class Sorte{
    public static void main(String[] args) {
        Scanner sc = new Scanner( System. in);
        int dia=0; mes=0; ano=0;
        int suma1=0; //esta é a primeira suma( dia+mes+ano)
        int suma2=0; //esta é a suma final
        int inc=1; cont=1; sorte=0;

        do {
            System. out. println("Día de nacemento?");
            dia = sc. nextInt();
        } while ( dia<1 || dia>31 )
        do {
            System. out. println("Mes de nacemento?");
            mes = sc. nextInt();
        } while ( mes<1 || mes>12 )

        System. out. println("Ano de nacemento?");
        ano = sc. nextInt();

        suma1=( dia+mes+ year );

        //isto é para contar os díxitos
        while ( (suma1/ inc) > 9 ){
            inc*=10;
            cont++;
        }

        // isto é para a suma final
        for (int i=0 ;i< cont ; i++){
            suma2=(suma1/ inc);
            suma1=(suma2* inc);
            inc/=10;
            sorte+=suma2;
        }

        System. out. println("O teu numero da sorte é: "+sorte);
    }
}
```


3. Materiais

3.1 Documentos de apoio ou referencia

- Especificación da linguaxe Java: <http://docs.oracle.com/javase/specs/>
- Java SE 8 API Documentation: <https://docs.oracle.com/javase/8/docs/api/index.html>
- The Java Tutorials: <https://docs.oracle.com/javase/tutorial/>
- V. RESÚA EIRAS. POOJava. <http://iespazodamerce.es/wiki/index.php?title=POOJava>
- J. SÁNCHEZ. Manual de Java. <http://jorgesanchez.net/java>
- P.A. SZNAJDLEDER. Java a fondo. Alfaomega. 2ª edición.
- J. BOBADILLA SANCHO. Java a través de ejemplos. Ra-Ma. Ed. 2003.
- Wikipedia. <https://www.wikipedia.org>

3.2 Recursos didácticos

- Ordenador persoal con conexión a Internet.
- Contorno de desenvolvemento NetBeans.
- Java SE Development Kit.
- Apuntamentos da profesora.
- Proxector.

4. Avaliación

Criterios de avaliación seleccionados para esta actividade (exemplo)	Evidencia de aprendizaxe	Instrumento de avaliación (tomados da aplicación de programación)	Peso na cualificación da UD
▪ CA3.2. Utilizáronse estruturas de repetición.	▪ Proxecto Java	▪ OU 2. Táboa de indicadores sobre un proxecto Java no que se utilicen estruturas de repetición.	15%
▪ CA3.3 Recoñecéronse as posibilidades das sentenzas de salto.	▪ Proxecto Java	▪ OU 3. Táboa de indicadores sobre un proxecto Java no que se recoñezan as posibilidades das sentenzas de salto.	15%
▪ CA3.5.2. Creáronse programas executables utilizando diversas estruturas de control repetitivas.	▪ Proxecto Java	▪ OU 5. Táboa de indicadores sobre un proxecto Java no que se creen programas executables utilizando diversas estruturas de control repetitivas.	30%

4.1 Modelo de proba

Enunciado

Crear un proxecto Java que debuxa un rectángulo sólido de asteriscos. O número de filas e columnas pídesse por teclado.

Solución

```
import java.util.*;

public class Rectangulo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int filas, columnas;

        //ler número de filas ata que sexa un número > 0
        do {
            System.out.print("Introduce número de filas: ");
            filas = sc.nextInt();
        } while(filas<1);

        //ler número de columnas ata que sexa un número > 0
        do {
            System.out.print("Introduce número de columnas: ");
            columnas = sc.nextInt();
        } while(columnas<1);

        for( int i = 1; i<=filas; i++){ //filas
            for( int j = 1; j<=columnas; j++){ //columnas
                System.out.print(" * ");
            }
            System.out.println();
        }
    }
}
```

A saída deste programa para filas = 6 e columnas = 10 é:

```
* * * * *
* * * * *
* * * * *
```

* * * * *

* * * * *

* * * * *

Táboa de indicadores

OU 3. Táboa de indicadores sobre un proxecto Java no que se utilicen estruturas de repetición.

Nome:		Data:	
■ CA3.2. Utilizáronse estruturas de repetición.			
Indicadores	Puntuación base	Puntuación obtida	Observacións
■ Crea unha estrutura for	3		
■ Crea unha estrutura while	3		
■ Crea unha estrutura do-while	3		
■ Diferenza as estruturas repetitivas	1		

OU 4. Táboa de indicadores sobre un proxecto Java no que se recoñezan as posibilidades das sentenzas de salto.

Nome:		Data:	
■ CA3.3 Recoñecéronse as posibilidades das sentenzas de salto.			
Indicadores	Puntuación base	Puntuación obtida	Observacións
■ Recoñeceu o código que debía ir nun bucle	3		
■ Determinou o bucle a utilizar	1		
■ Estableceu correctamente as condicións de repetición	2		
■ Evitou os bucles infinitos	1		
■ Combinou as estruturas de repetición	3		

OU 5. Táboa de indicadores sobre un proxecto Java no que se creen programas executables utilizando diversas estruturas de control repetitivas.

Nome:		Data:	
■ CA3.5.2. Créanse programas executables utilizando diversas estruturas de control repetitivas.			
Indicadores	Puntuación base	Puntuación obtida	Observacións
■ Codificou correctamente un bucle para o control de datos de entrada	4		
■ Combinou dous bucles for anidados	2		
■ Determinou ben os índices dos dous bucles for anidados	3		
■ Comentou o código	1		