

Ejercicio U3_B6_1:

VERSIÓN CON BREAK

```
class Unidad3 {  
    public static void main(String[] args) {  
        String s = "1110 001";  
        int i=0;  
        while (i < s.length() ) {  
            if (s.charAt(i) == ' ')  
                break;  
            i++;  
        }  
        if (i == s.length()) {  
            System.out.println("no existe ningún espacio");  
        } else {  
            System.out.println("Indice buscado : " + i);  
        }  
    }  
}
```

VERSIÓN SIN BREAK

```
class Unidad3 {  
    public static void main(String[] args) {  
        String s = "1110 001";  
        int i=0;  
        while( i < s.length() && s.charAt(i) != ' ' ) {  
            i++;  
        }  
  
        if (i == s.length()) {  
            System.out.println("no existe ningún espacio");  
        } else {  
            System.out.println("Indice buscado : " + i);  
        }  
    }  
}
```

Ejercicio U3_B6_2:

Puedo simplemente quitar el break pero tengo que tener en cuenta que si analizo un número más largo de 5 cifras el valor de cifras puede ser mayor que 5, por esto cambié el if del final poniendo >=

```
class Unidad3 {  
  
    public static void main(String[] args) {  
        int num = 1234567891;
```

```

int cifras = 0;
while (num > 0) {
    num /= 10;
    cifras++;
}
if (cifras >= 5) {
    System.out.println("El número tiene 5 o más dígitos");
} else {
    System.out.println("El número tiene menos de 5 dígitos");
}
}
}

```

El programa no es del todo eficiente ya que una vez que detecto que el número tiene 5 cifras debería abandonar el bucle. Puedo mejorar esto de la siguiente forma:

```

class Unidad3 {
    public static void main(String[] args) {
        int num = 1234;
        int cifras = 0;
        while(num > 0 && cifras<5) {
            num/=10;
            cifras++;
        }
        if (cifras ==5)
            System.out.println("El número tiene 5 o más dígitos");
        else
            System.out.println("El número tiene menos de 5 dígitos");
    }
}

```

Observa que en la condición del while aparece una condición compuesta. Las condiciones compuestas son peligrosas ya que es fácil cometer un error de lógica. En este caso la condición no es difícil de razonar apropiadamente pero en otros casos la condición es tan liosa (con 3 operandos, con not, etc...) que el break es “mano de santo” para simplificar y mejorar el código.

Ejercicio U3_B6_3:

Para registrar que se ha detectado una t se suelen usar una variable extra que por su valor avisa de que algo ocurrió. A estas variables se les llama en la literatura de programación con muchos nombres: flag, bandera, centinela, chivato, ...

En este ejemplo el flag *hayT* es una simple variable booleana ya que con dos valores que pueda tomar me resulta suficiente, en otros casos el flag sería un int.

```

class Unidad3 {
    public static void main(String[] args) {
        String frase = "que frase tan bonita";
        boolean hayT=false;
    }
}

```

```

for( int i = 0 ; i < frase.length(); i++ ){
    char caracter=frase.charAt(i);
    if( caracter == 't' ){
        hayT=true;
    }
}
if(hayT)
    System.out.println("si que hay T");
}
}

```

La solución con break es más eficiente, ya que la primera T con la que se encuentra hace abandonar el bucle. Para evitar hacer iteraciones necesarias podemos hacer una condición de salida más sofisticada, y de nuevo, en este caso es sencilla y por tanto el break es prescindible, pero si fuera más compleja el break "es más fácil".

```

class Unidad3 {

    public static void main(String[] args) {
        String frase = "que frase tan bonita";
        boolean hayT = false;
        for (int i = 0; i < frase.length() && !hayT; i++) {
            char caracter = frase.charAt(i);
            if (caracter == 't') {
                hayT = true;
                System.out.println("si que hay T");
            }
        }
    }
}

```

Ejercicio U3_B6_4:

en esta solución observa como j no es variable local del bucle interno, si no variable local del externo.

```

class Unidad3 {
    public static void main(String[] args) {

        for(int i=1; i<=3;i++){
            System.out.println("Bucle externo, COMIENZA iteración:"+i);
            int j;
            for(j=1;j<=5;j++){
                if(j==3)

```

```
        break ;
        System.out.println("\tBucle interno en iteración:"+j);
    }
    if(j==3)
        break;
    System.out.println("Bucle externo, FINALIZA iteración:"+i +"\n");
}

}

}
```

y sin break, todavía más rollo ... habría que meter variables flag