

# Programación Web Full Stack



## Java Excepciones

Total de puntos **100/100** ?

Al finalizar la guía, **es el momento de validar lo que has aprendido hasta ahora**. Por eso, es importante responder el Check de conocimiento. ¡Aprovecha este espacio para chequear qué tal vienes hasta aquí!

Antes

de resolver este cuestionario te recomendamos haber visto con atención el/los videos, la guía de este tema y haber resuelto los ejercicios. Si ya lo hiciste, ¡adelante!

*Tendrás un único intento*  
¡Buena suerte!

**Correo electrónico \***

brogno.micaela@hotmail.com

100 de 100 puntos

**Nombre \***

Micaela

**Apellido \***

Brogno

**Documento de Identidad \***

38859739

**Correo electrónico con el que estás registrado en Egg \***

brogno.micaela@hotmail.com

✓ **1) La clase Error maneja errores: \***

10/10

- ☐ a) Del código
- ☒ b) De la Maquina Virtual de Java ✓
- ☐ c) De Netbeans
- ☐ d) Ninguna de las anteriores

✓ **2) La clase Exception maneja errores: \***

10/10

- ☒ a) Del código ✓
- ☐ b) De la Maquina Virtual de Java
- ☐ c) De Netbeans
- ☐ d) Ninguna de las anteriores

✓ **3) La clase padre de todas las excepciones y errores es la clase \*** 10/10

- ☐ a) Error
- ☐ b) Exception
- ☒ c) Throwable ✓
- ☐ d) Ninguna de las anteriores



✓ **4) Las excepciones se manejan con el bloque: \***

10/10

- ☐ a) Finally
- ☐ b) Throws
- ☒ c) Try Catch
- ☐ d) Throw



✓ **5) Para cerrar recursos después de una operación usamos el bloque:**

\*10/10

- ☒ a) Finally
- ☐ b) Throws
- ☐ c) Try Catch
- ☐ d) Throw



✓ **6) Cuando queremos informar que un método puede tirar una excepción usamos la palabra reservada:**

\*10/10

- ☐ a) Finally
- ☒ b) Throws
- ☐ c) Try Catch
- ☐ d) Throw



✓ 7) Cuando queremos lanzar una excepción de manera explícita usamos el bloque: \*10/10

- ☐ a) Finally
- ☐ b) Throws
- ☐ c) Try Catch
- ☒ d) Throw ✓

✓ 8) En Java la diferencia entre throws y throw es: \* 10/10

- ☐ a) throws arroja una excepción y throw indica el tipo de excepción que no maneja el método
- ☐ b) throws se usa en los metodos y throw en los constructores
- ☒ c) throws indica el tipo de excepción que maneja el método y throw arroja una excepción ✓
- ☐ d) Ninguna de las anteriores

✓ 9) Teniendo el siguiente código, indicar que excepción se debería manejar \*10/10

```
public static void main(String[] args) {  
    int division = 10 / 0;  
}
```

- ☐ a) NullPointerException
- ☒ b) ArithmeticException ✓
- ☐ c) RuntimeException
- ☐ d) ArrayIndexOutOfBoundsException

✓ 10) Teniendo el siguiente código, indicar que excepción se debería manejar

\*10/10

```
public static void main(String[] args) {  
    int[] vector = new int[5];  
    for (int i = 0; i < 10; i++) {  
        vector[i] = 1;  
    }  
}
```

- ☐ a) NullPointerException
- ☐ b) ArithmeticException
- ☐ c) RuntimeException
- ☒ d) ArrayIndexOutOfBoundsException



Te invitamos a responder algunas preguntas sobre el material de estudio 0 de 0 puntos

¿Te resultó clara la guía teórica de trabajo? \*

- ☒ Sí
- ☐ No

¿Te resultaron útiles los videos? \*

- ☒ Sí
- ☐ No



**Si respondiste que No a alguna de las anteriores, cuéntanos el motivo:**

---

**¿Los ejercicios propuestos te ayudaron a aplicar la teoría de la guía y/o videos?** \*

☒ Sí

☐ No

**¿Qué tan desafiantes te resultaron los ejercicios?** \*

1 2 3 4 5 6 7 8 9 10

Poco desafiantes

☐☐☐☐☐☒☐☐☐☐

Muy desafiantes

**En Egg agradecemos cada oportunidad de mejora. ¿Quisieras dejar algún comentario constructivo al respecto?** \*

**En caso de ser necesario, el equipo de contenidos te contactará vía email para solicitar profundizar tu idea.**

No

---

El formulario se creó en Egg Cooperation.

Google Formularios

