



EBook Gratis

APRENDIZAJE concurrency

Free unaffiliated eBook created from
Stack Overflow contributors.

#concurr

y

Tabla de contenido

Acerca de.....	1
Capítulo 1: Empezando con la concurrencia.....	2
Observaciones.....	2
Examples.....	2
Instalación o configuración.....	2
Ejemplo de ejecución concurrente en Java.....	2
Creditos.....	4

Acerca de

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: [concurrency](#)

It is an unofficial and free concurrency ebook created for educational purposes. All the content is extracted from [Stack Overflow Documentation](#), which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official concurrency.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Capítulo 1: Empezando con la concurrencia

Observaciones

Esta sección proporciona una descripción general de qué es la concurrencia y por qué un desarrollador puede querer usarla.

También debe mencionar cualquier tema grande dentro de la concurrencia, y vincular a los temas relacionados. Dado que la Documentación para la concurrencia es nueva, es posible que deba crear versiones iniciales de los temas relacionados.

Examples

Instalación o configuración

Instrucciones detalladas sobre cómo configurar o instalar la concurrencia.

Ejemplo de ejecución concurrente en Java.

```
import java.util.stream.IntStream;

public class Concurrent {
    public static void printAndWait(String s) {
        System.out.println(s);
        try {
            Thread.sleep(1000);
        } catch (Exception e) {}
    }

    public static void main(String[] args) {
        Thread myThread = new Thread() {
            public void run() {
                IntStream.range(1, 32)
                    .forEach(x -> printAndWait(""+x));
            }
        };
        myThread.start();
        IntStream.range('a', 'z').forEach(x -> printAndWait(""+(char)x));
    }
}
```

Esto producirá una salida de algo similar a

```
a
1
b
2
c
3
```

y así sucesivamente, aunque los resultados pueden variar. Esto se debe a que el código en `myThread` se ejecuta simultáneamente, en un hilo diferente, como el flujo principal. Es decir, el rango 1-32 es manejado por un hilo, y el rango az es manejado por otro.

Dado que no hay sincronización entre los subprocesos, no hay garantía de que uno ejecutará primero o incluso que produzcan un resultado que esté perfectamente entrelazado.

Lea [Empezando con la concurrencia en línea](https://riptutorial.com/es/concurrency/topic/4563/empezando-con-la-concurrencia):

<https://riptutorial.com/es/concurrency/topic/4563/empezando-con-la-concurrencia>

Creditos

S. No	Capítulos	Contributors
1	Empezando con la concurrencia	Bex , Community