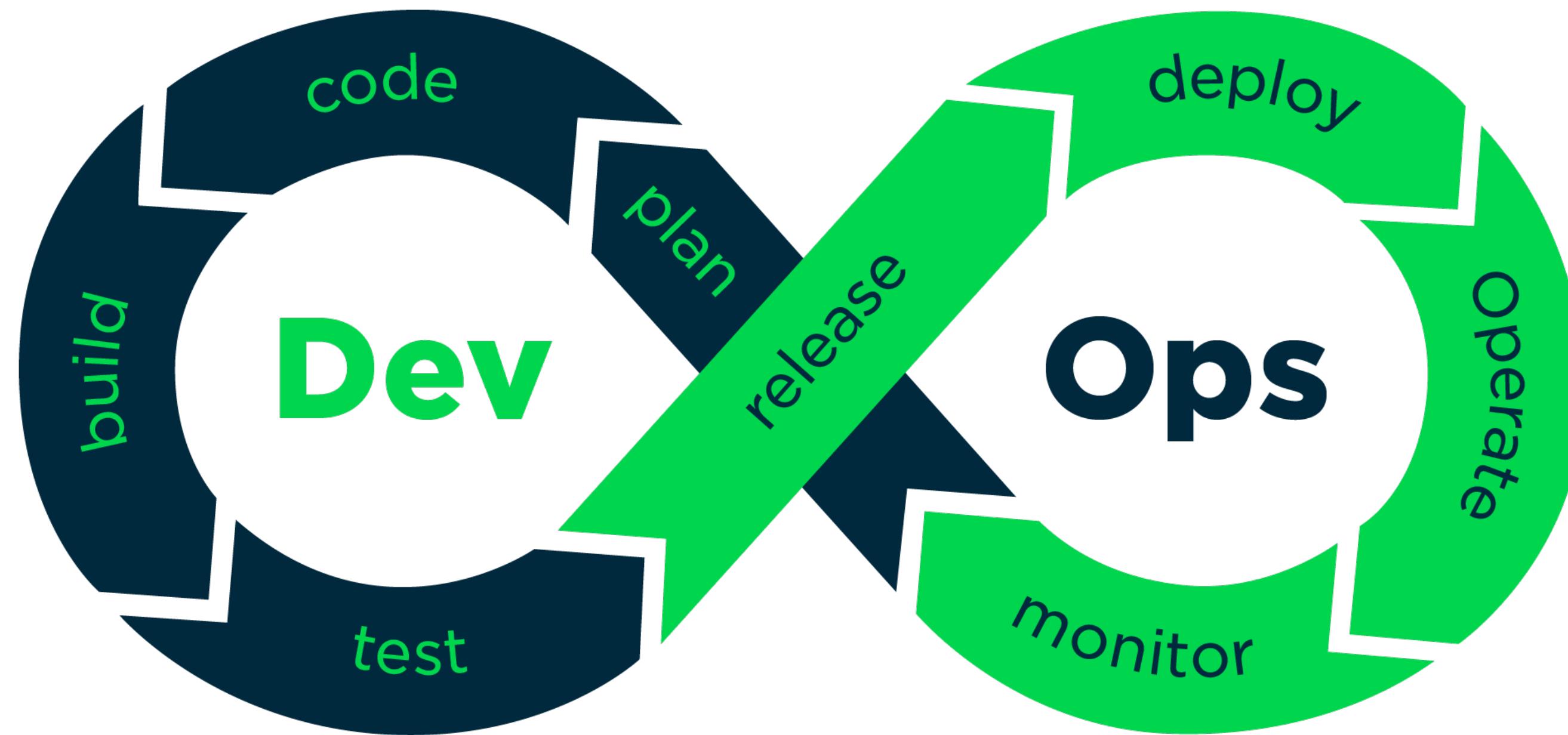


Introduzione alla Tecnologia DevOps



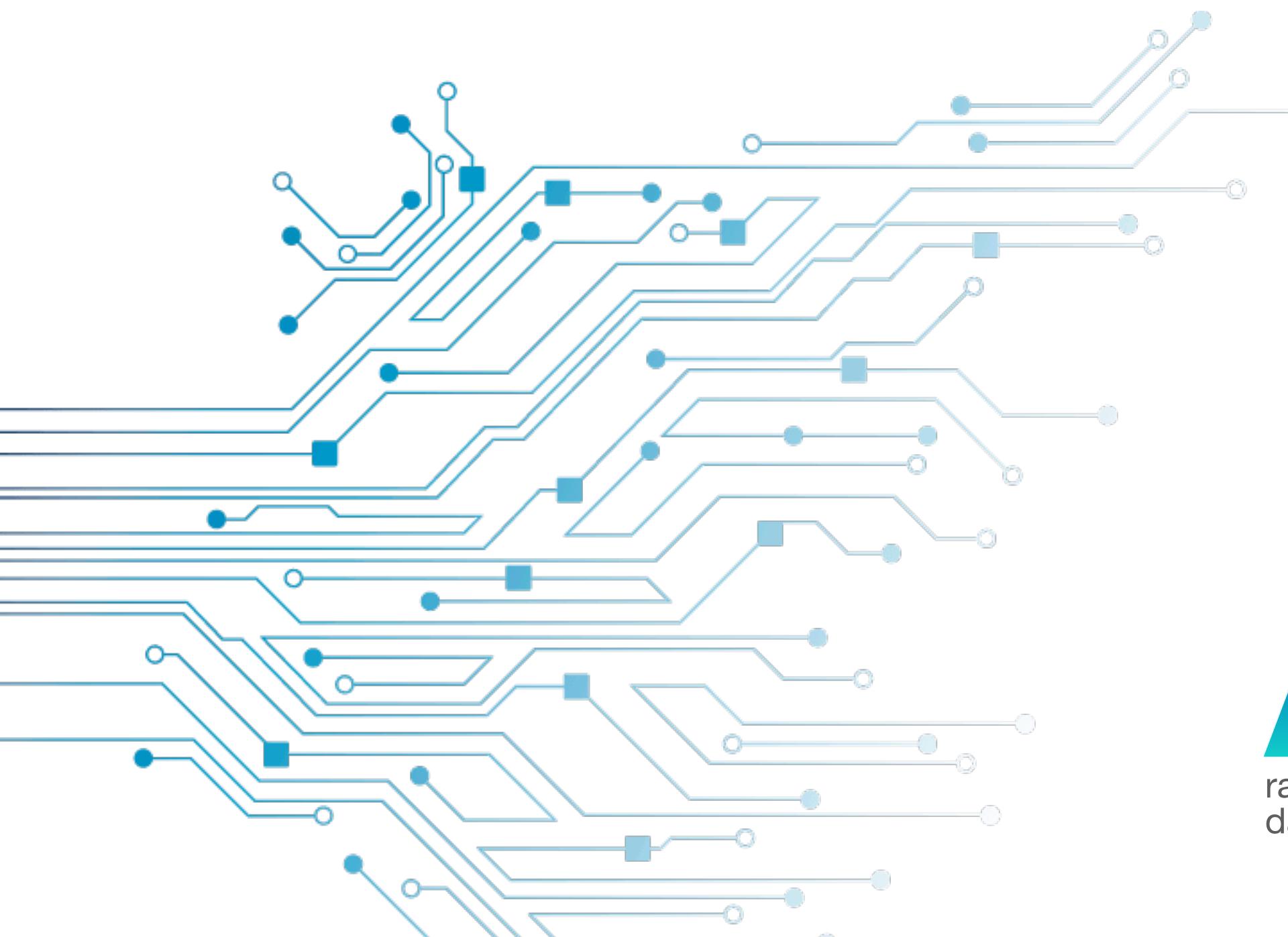
Ludovica P.

IT Consultant presso **Betacom Bari**
Java Full Stack Developer
DevOps Engineer

L'informatica è

la scienza che si occupa del trattamento dell'informazione
mediante procedure automatizzate

Sviluppo software



Cybersecurity

AI e machine learning

branca dell'intelligenza artificiale che raccoglie metodi come il riconoscimento di pattern

Ingegneria del Software

disciplina informatica che si occupa dei processi produttivi
e delle metodologie di sviluppo finalizzate alla realizzazione di sistemi software

Sistemi embedded e IoT

sistemi elettronici di elaborazione a microprocessore progettati appositamente per un determinato utilizzo

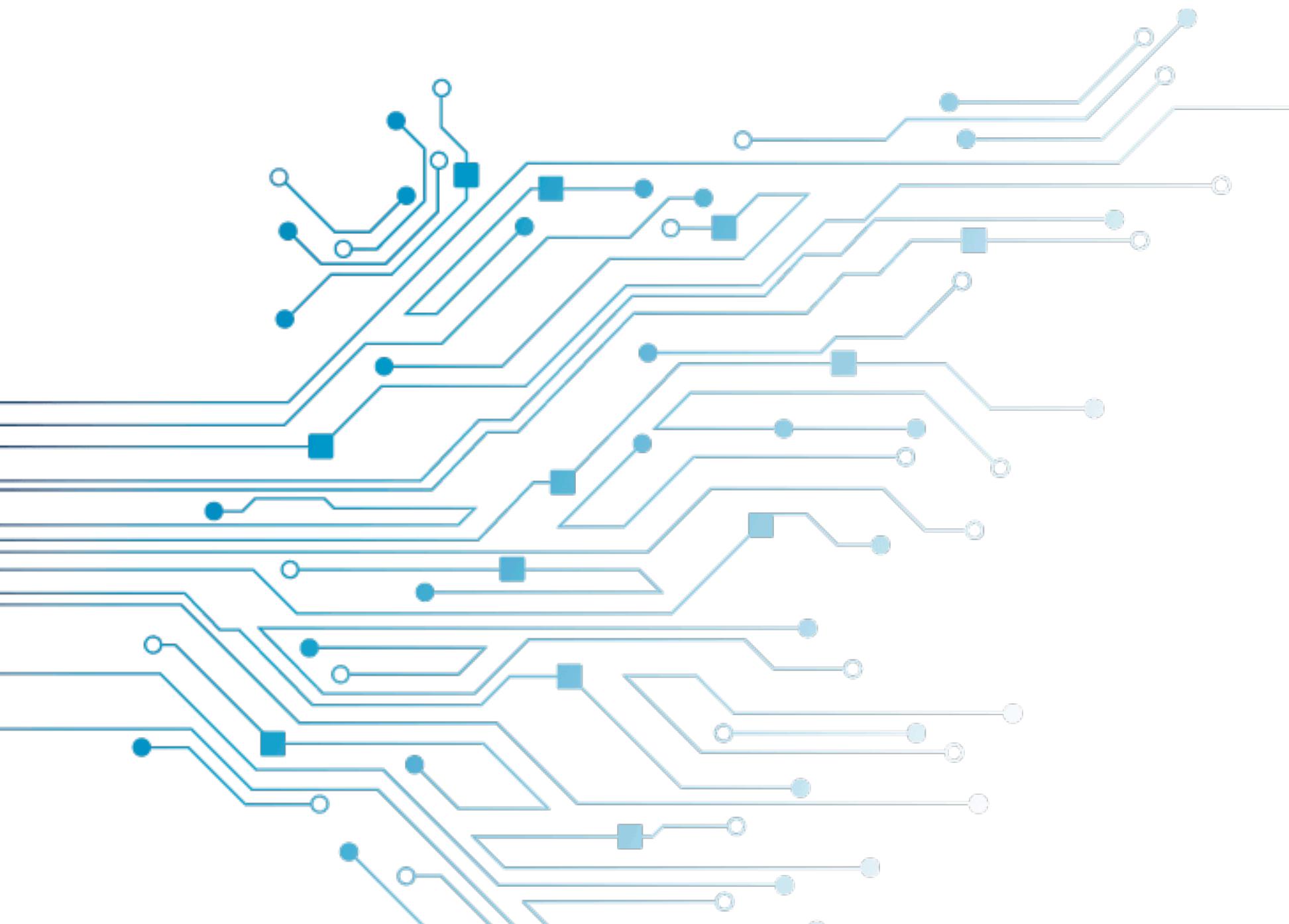
Analisi dei dati e Big Data

raccolta di **dati informatici** così estesa in termini di volume, velocità e varietà
da richiedere tecnologie e metodi analitici specifici per l'estrazione di valore o **conoscenza**

Le competenze chiave



APPRENDIMENTO CONTINUO



Pensiero critico

Problem Solving

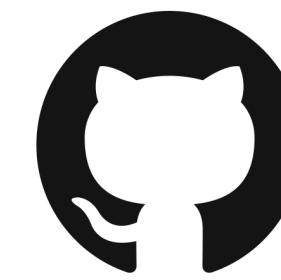
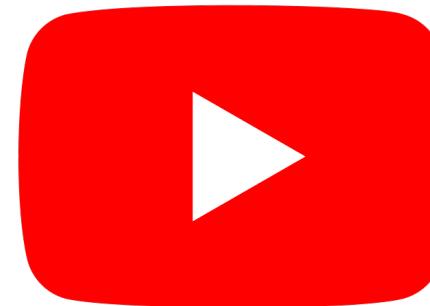
Apertura mentale

Disponibilità

Lavoro di Squadra

Padronanza dell'inglese

Strumenti per imparare

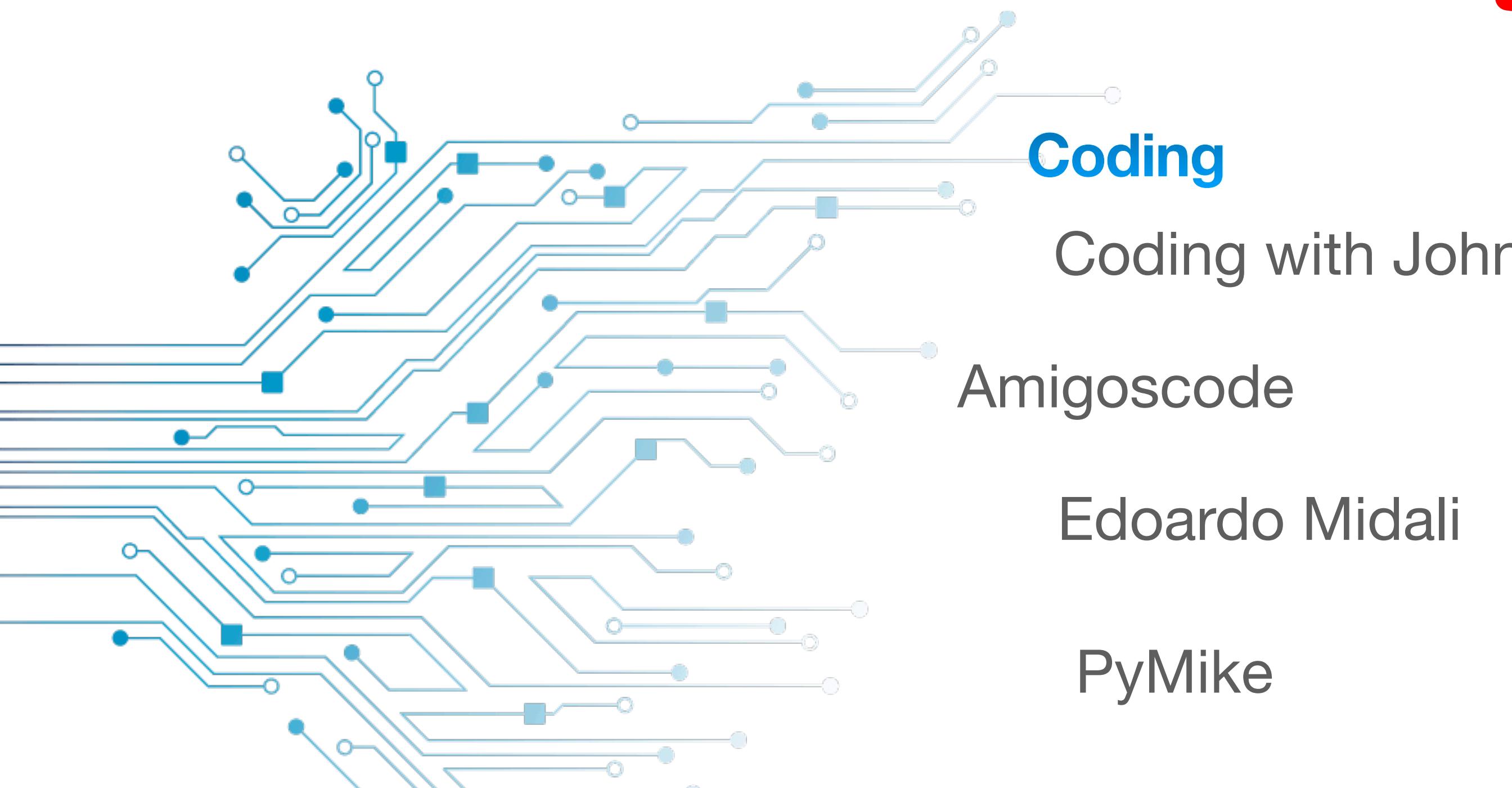


Risorse online:

www.w3schools.com

www.tutorialspoint.com

Content Creators:



Coding

Coding with John

Amigoscode

Edoardo Midali

PyMike

Linux

Learn Linux TV

Cybersec

Null Byte

Live Overflow

The Cyber Mentor

Coding for Discord and Minecraft

Work Off Keys

Prima di iniziare...

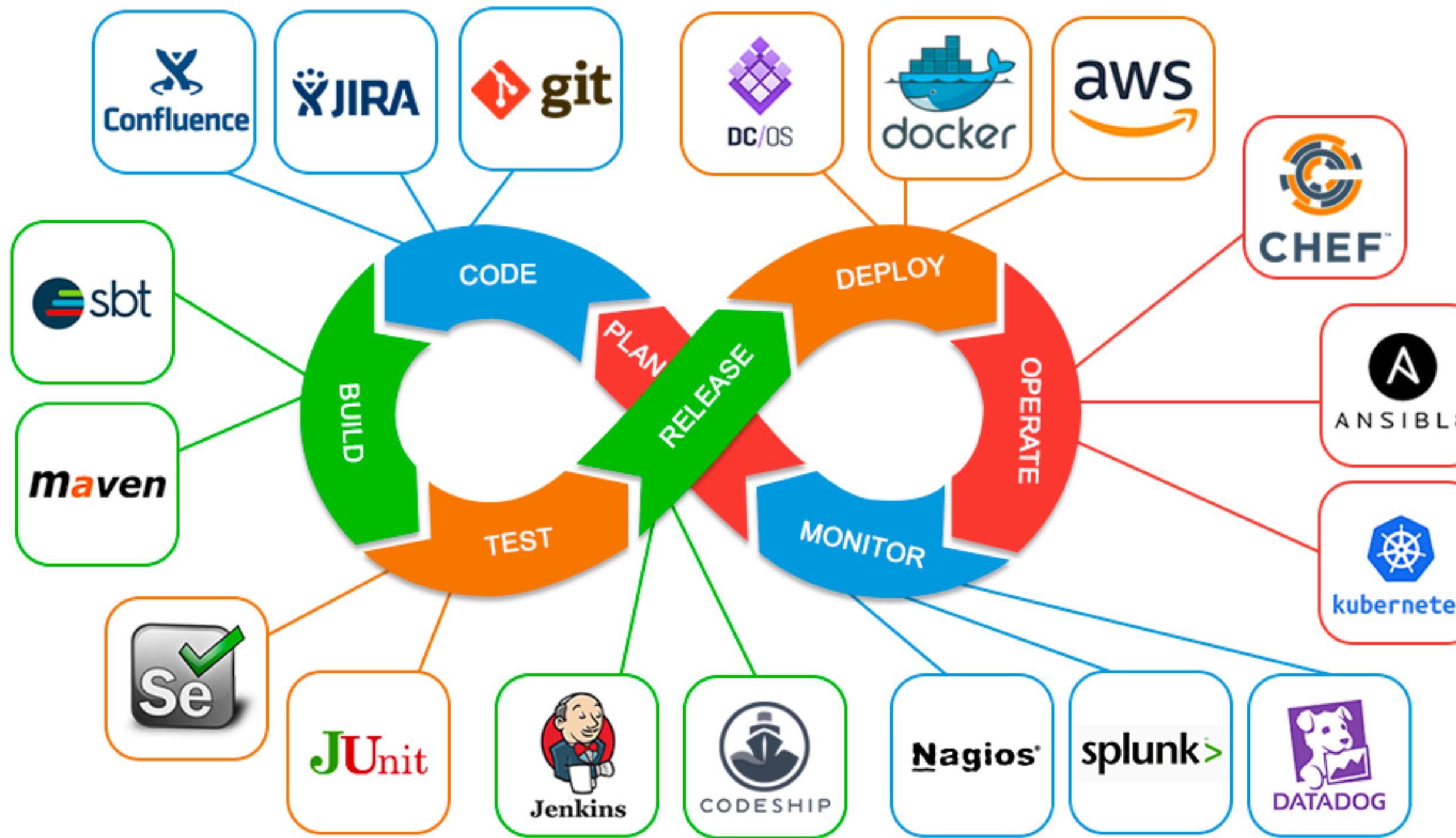
Un po' di definizioni utili

- **Software Deployment** - Il rilascio del software include tutte le fasi, i processi e le attività necessarie per rendere un sistema software o un aggiornamento disponibile agli utenti previsti.
- **Metodologia agile** - è un modo di sviluppare software in modo flessibile e collaborativo. Invece di pianificare tutto all'inizio, si lavora a piccoli passi, facendo miglioramenti continuativi.
- **Test Automation** - il processo di automatizzare l'esecuzione di test software attraverso l'uso di strumenti e script. Questo approccio consente di effettuare test in modo rapido e consistente, migliorando l'efficienza e riducendo gli errori umani.
- **Environment** - un contesto o una configurazione specifica in cui il software viene eseguito, testato o distribuito.

Che significa DevOps?

Dev: development - sviluppo (software)

Ops: operations - monitoraggio e manutenzione dell'ambiente di produzione



Pipelines CI / CD

flussi automatizzati che gestiscono il processo di integrazione, distribuzione e consegna del software.

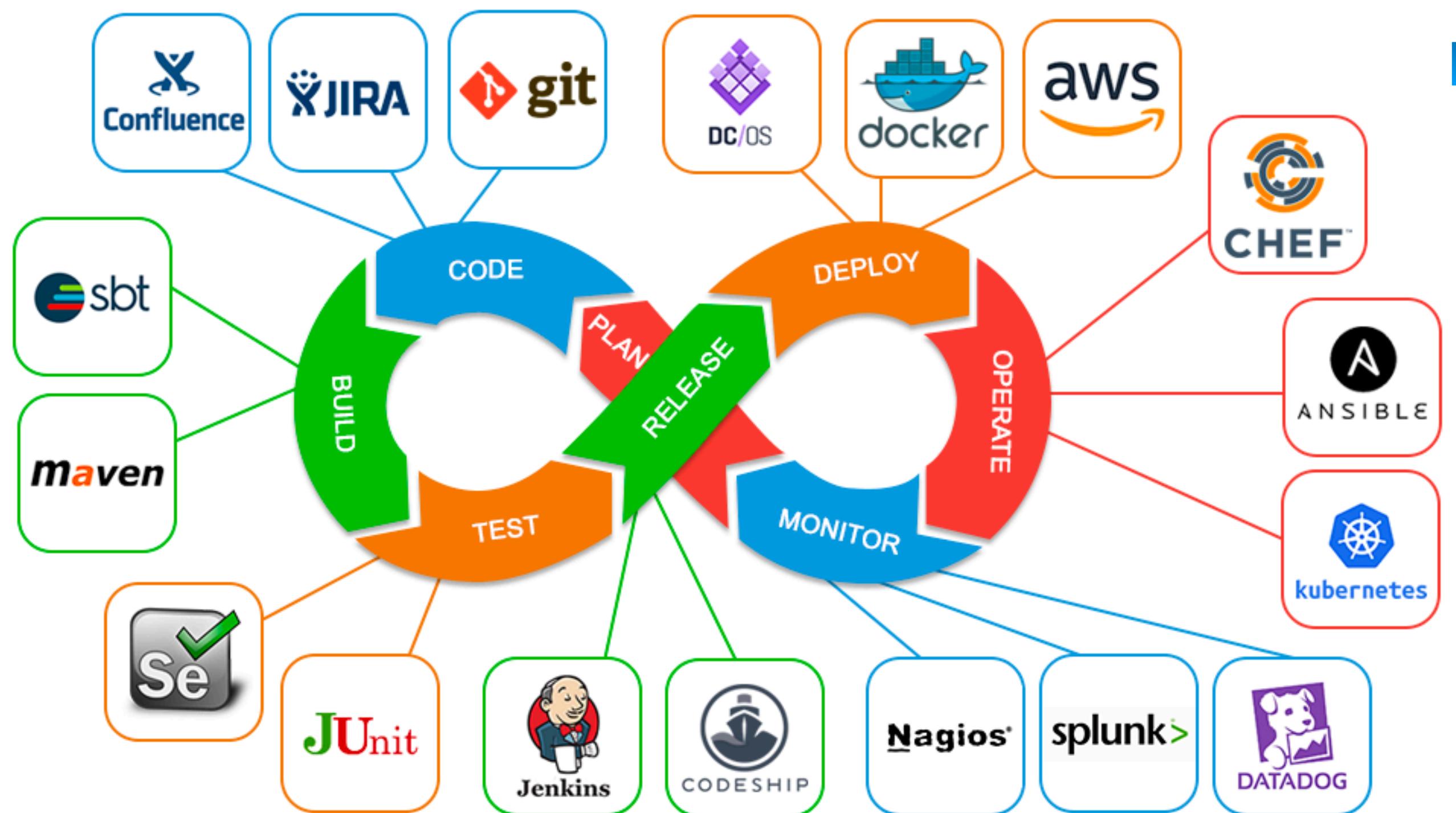
E' una metodologia aziendale che mira a:

- migliorare la **collaborazione** e la **comunicazione** tra i team di sviluppo e quelli di operations
- ridurre il **gap** tra le fasi di sviluppo e distribuzione del software
- promuovere l'**automazione** dei processi, la condivisione delle responsabilità e l'implementazione di **pratiche agili** per accelerare la consegna del software, migliorare l'efficienza operativa e garantire un ambiente più stabile.
- fornire prodotti software di **alta qualità** in modo **rapido** e **affidabile**

Pipelines CI / CD

flussi automatizzati che gestiscono il processo di integrazione, distribuzione e consegna del software.

Che significa DevOps?



PLAN ↔

In questa fase, si definiscono gli obiettivi del progetto, si identificano le risorse necessarie e si stabilisce un piano di sviluppo.

CODE ↔

I programmatore scrivono il codice sorgente del software in base ai requisiti specificati durante la fase di pianificazione.

BUILD ↔

Questa fase coinvolge la compilazione del codice sorgente in un eseguibile o in un pacchetto software utilizzabile.

TEST ↔

Vengono eseguiti test automatici e manuali sul software per individuare e correggere eventuali errori o bug.

RELEASE

Il software viene preparato per la distribuzione, comprensivo di documentazione, e reso disponibile per il rilascio al pubblico o in un ambiente di produzione.

DEPLOY

I software viene installato e configurato nell'ambiente di produzione, pronto per essere eseguito e utilizzato dagli utenti finali.

OPERATE

Durante questa fase, il software è in funzione nell'ambiente di produzione e gli operatori monitorano le prestazioni, gestiscono risorse e risolvono eventuali problemi.

MONITOR

Si utilizzano strumenti di monitoraggio per tracciare le prestazioni del software in tempo reale, raccogliere dati sull'utilizzo e rilevare eventuali anomalie o problemi operativi.

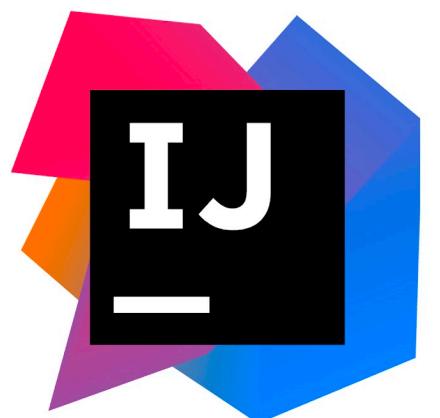
La figura del Consulente nell'Ambito DevOps



Nel prossimo episodio...

JUnit

Introduzione a JUnit e
realizzazione di
esercizi



teradata.

Cos'è JUnit?

JUnit è un framework di testing per il linguaggio di programmazione Java che fornisce annotazioni e metodi per definire, organizzare e eseguire i test.

`@Test`



`@AfterAll`

`@BeforeAll`

`assertEquals(expected, actual)`
`assertNotEquals(expected, actual)`

`assertTrue(condition)`
`assertFalse(condition)`

E così via...

Esempio di Utilizzo del Framework Junit

Calculator.java

```
1 package org.example;  
2  
2 usages  
3 public class Calculator {  
4     1 usage  
5         public int add(int a, int b) {  
6             return a + b;  
7         }  
7 }
```

CalculatorTest.java

```
1 package org.example;  
2 import org.junit.Test;  
3  
4 import static org.junit.Assert.assertEquals;  
5  
6 ► public class CalculatorTest {  
7     @Test  
8     ► public void testAdd() {  
9         Calculator calculator = new Calculator();  
10        int result = calculator.add( a: 2, b: 3);  
11  
12        // Verifica che il risultato sia corretto  
13        assertEquals( expected: 5, result);  
14    }  
15 }  
16
```

Qualche video per approfondire...

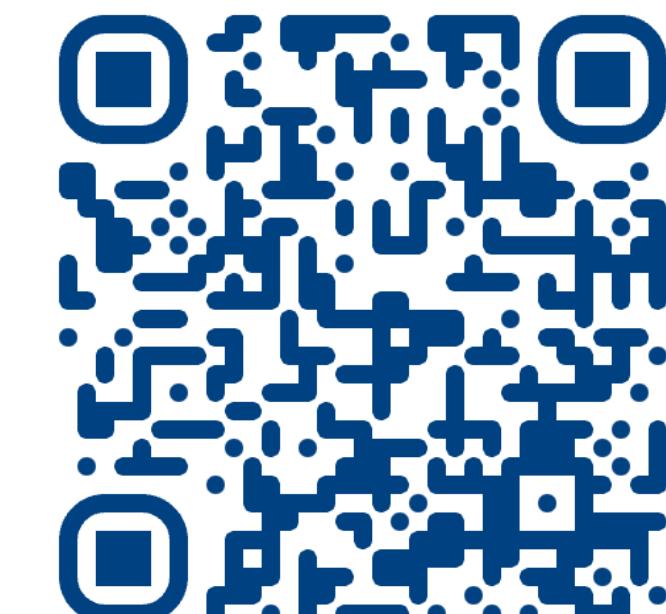
**Java Unit Testing with JUnit - Tutorial
- How to Create And Use Unit Tests**



JUnit 5 Tutorial - Crash Course



JUnit Tutorial - tutorialspoint



Esercizio guidato - Intro alla Test Automation

Crea un Test JUnit per verificare la connettività ad un Database e l'esecuzione corretta di una query
(caricare il codice su GitHub)

IDE -> IntelliJ IDEA, Visual Studio Code, Eclipse...

DB -> MySQL Community Server e MySQL Workbench

Maven !!!

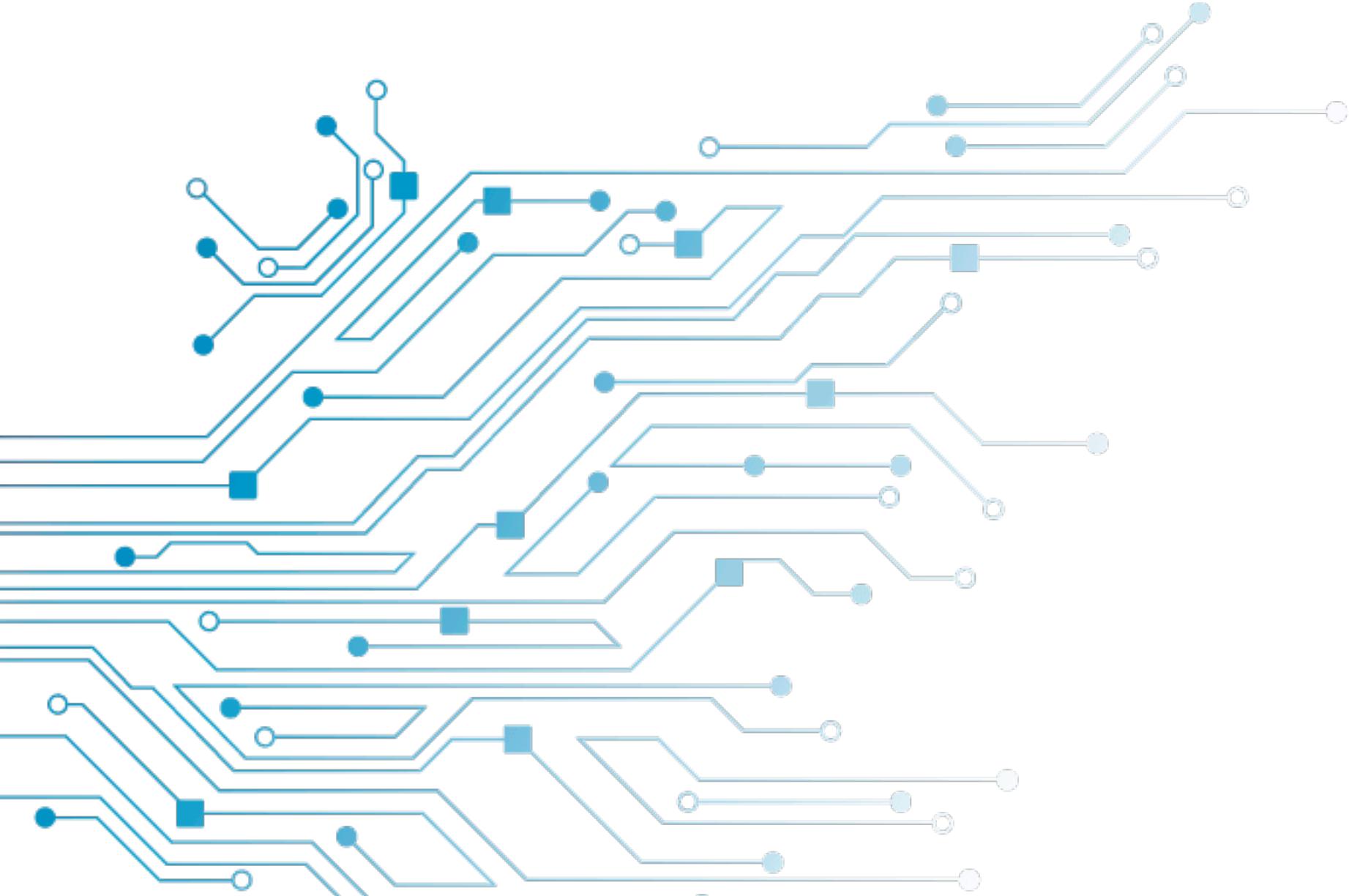


Download Links

<https://dev.mysql.com/downloads/mysql/>

<https://dev.mysql.com/downloads/workbench/>

<https://maven.apache.org/download.cgi>



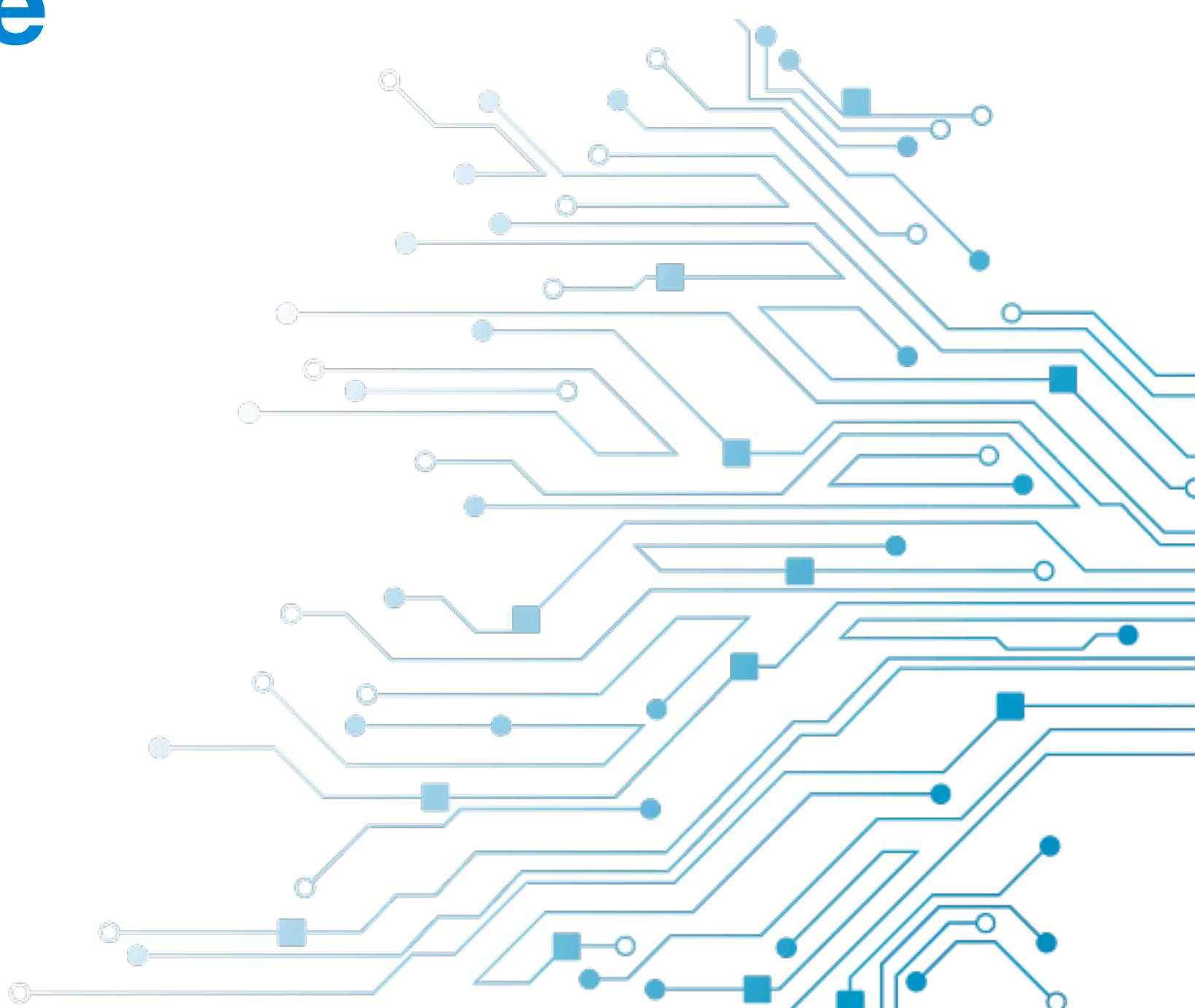
Esercizio guidato - Creazione DB

1) Set-up (installazione) di MySQL Community Server e di MySQL Workbench

Script .sql

2) Creazione e popolamento di un database

3) Scrittura di una semplice query SELECT che restituisca un intero



Esercizio guidato - Maven dependencies

- 1) Creazione di un progetto Java con Maven**
- 2) Modifica del file pom.xml per introdurre le dependencies**
- 3) Opzionale -> Installazione del plugin Lombok se si vuole implementare @Slf4j per le loggate**



Esercizio guidato - Maven dependencies

```
<dependencies>
    <dependency>
        <groupId>org.junit.jupiter</groupId>
        <artifactId>junit-jupiter-params</artifactId>
        <version>5.8.1</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>com.mysql</groupId>
        <artifactId>mysql-connector-j</artifactId>
        <version>8.2.0</version>
        <scope>test</scope>
    </dependency>
    <dependency>
        <groupId>org.projectlombok</groupId>
        <artifactId>lombok</artifactId>
        <version>1.18.22</version>
        <scope>provided</scope>
    </dependency>
    <dependency>
        <groupId>org.slf4j</groupId>
        <artifactId>slf4j-api</artifactId>
        <version>1.7.32</version>
    </dependency>
    <dependency>
        <groupId>ch.qos.logback</groupId>
        <artifactId>logback-classic</artifactId>
        <version>1.4.12</version>
    </dependency>
</dependencies>
```

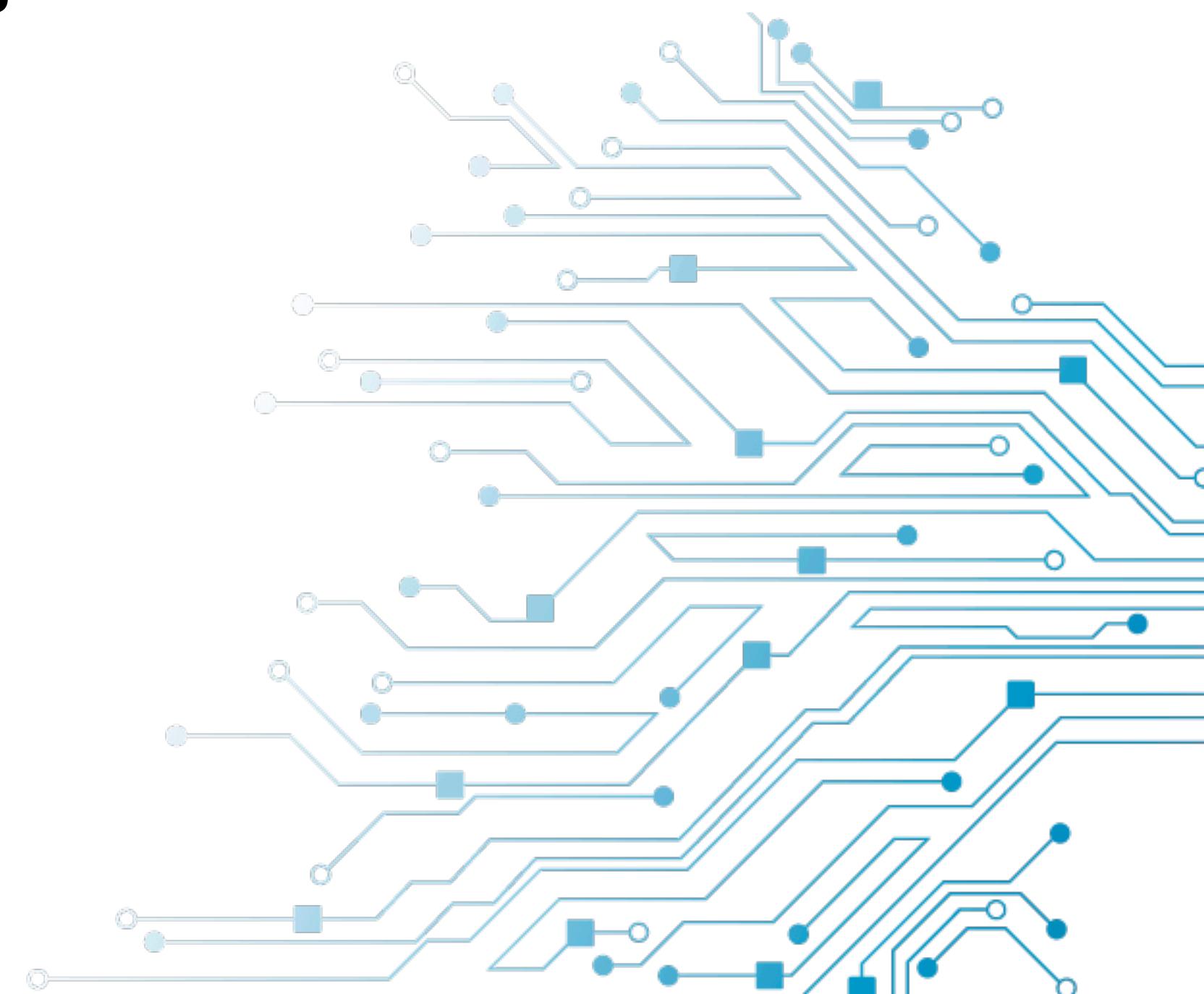
MySQL Connector (jdbc)

JUnit

Logback

Lombok

Slf4j



Esercizio guidato - Creazione classe Test

- 1) Creazione di una classe di Test (varia in base all'IDE)**
- 2) Creazione di un file .csv in cui inserire le query e di un file config.properties per inserire le proprietà del DB**
- 3) Scrivere un metodo per leggere le properties dal file config.properties**
- 4) Scrivere il metodo per la connessione al db**
- 5) Scrivere il metodo per la chiusura della connessione**



Esercizio guidato - Creazione test query

All'interno della classe di Test, creare un metodo che legga dal file .csv la query e la esegua (sfruttando la connessione al db pre-esistente).

Utilizzare assertEquals fra il valore ottenuto dalla query e fra quello che ci si aspetta di ottenere.

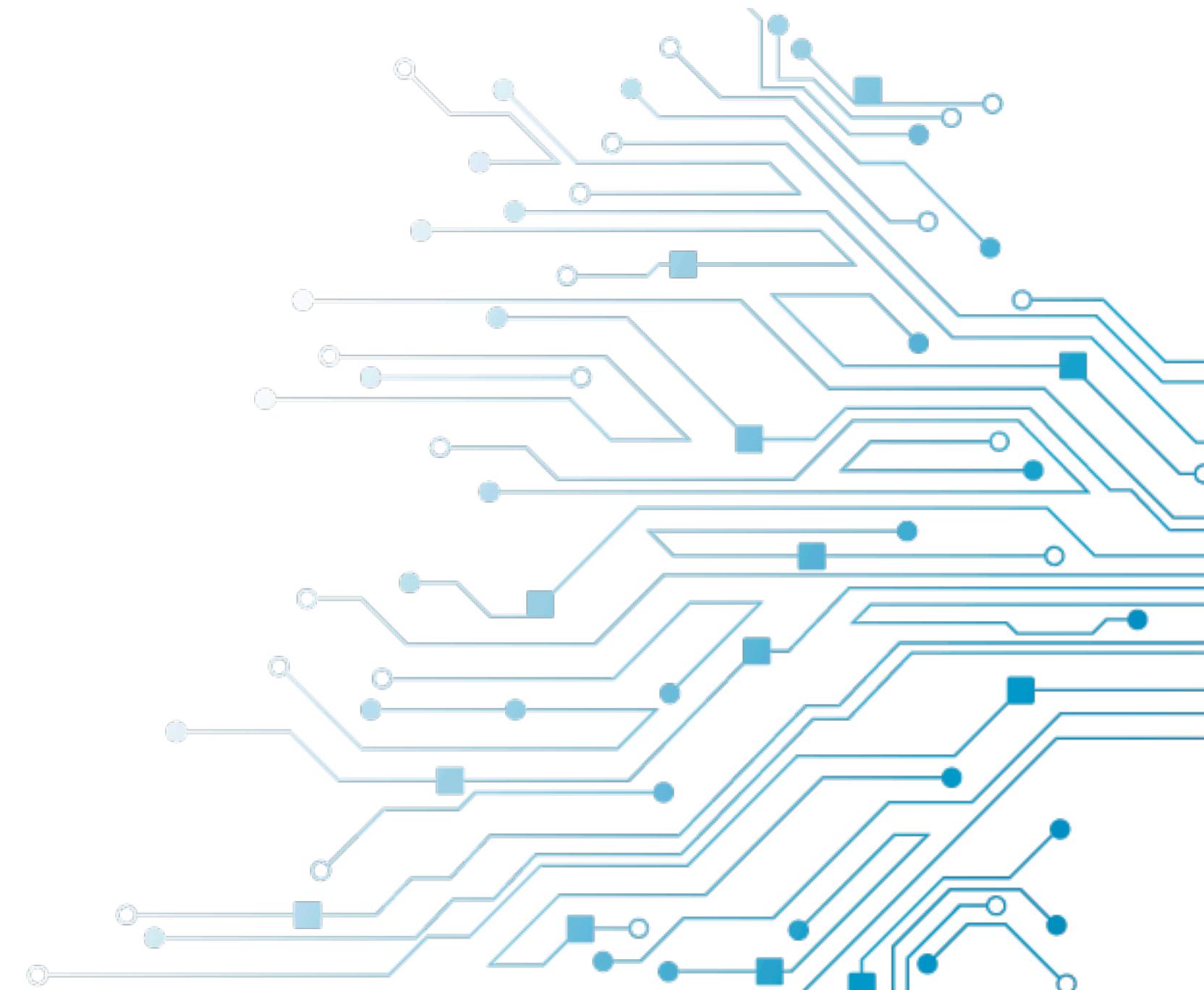
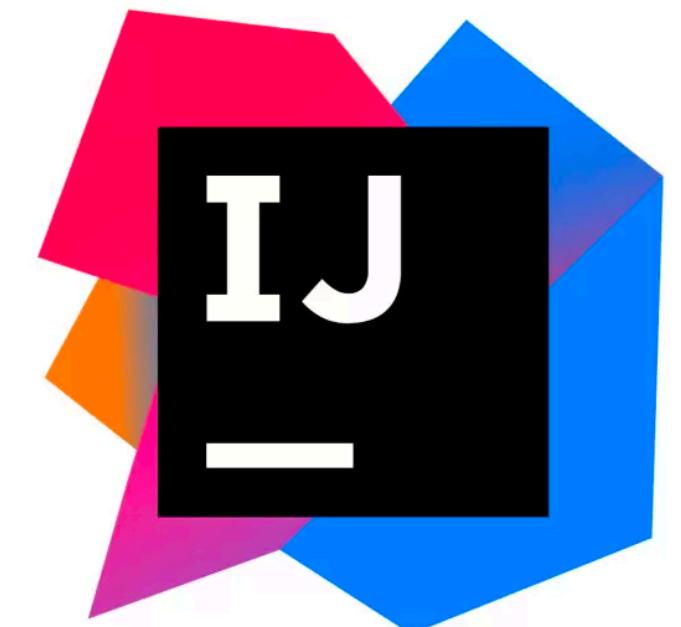
```
@ParameterizedTest  
@CsvFileSource(resources = "/queries.csv", numLinesToSkip = 1)
```

Si può fare in vari modi



Esercizio guidato - GitHub

Se volete potete creare una repository su GitHub e condividere lì il codice in modo da testare tutti insieme alla fine.



Dubbi, Domande, Incertezze?

