

Alma Mater Studiorum-Università di Bologna

Scuola di Ingegneria

Dai Linguaggi alle Infrastrutture Software

Corso di Laurea in Ingegneria Informatica

Anno accademico 2021/2022

Prof. ENRICO DENTI

Dipartimento di Informatica – Scienza e Ingegneria (DISI)



“PROGRAMMARE” ... ?

- In passato, “occuparsi di informatica” è stato a lungo sinonimo di “programmare computer”
- Fa pensare a una *attività poco stimolante*, dove le *fasi creative – analisi e progetto* – sono già avvenute
 - tipico della costruzione di *algoritmi* di medio/piccola dimensione, normalmente già noti e descritti in letteratura
- Anni '80: “crisi del software”
insufficienza di metodologie e strumenti nati per una *visione algoritmica* dell’informatica (*programmazione in-the-small*) rispetto a progettazione, sviluppo e manutenzione di *sistemi software complessi* (*progettazione in the large*)



PROGETTARE SISTEMI SOFTWARE

- Il termine “programmare” richiama una *visione obsoleta* dell’informatica
- Nuovo obiettivo:
saper progettare e costruire sistemi software
con il *rigore* e le *garanzie* che si richiedono
a tutti gli altri prodotti dell’ingegneria
 - qualità
 - affidabilitàNecessità di *uscire dall’artigianato*
Concepire il *software* come un *prodotto industriale*



UN CAMBIO DI PARADIGMA

- Uscire dall'artigianato
 - non si può reinventare la ruota ogni giorno
 - impossibile fare le cose da soli "nel proprio sottoscala"
 - pieno supporto al *lavoro di squadra*
- Software come prodotto industriale
 - strutturazione precisa del *processo di produzione*
 - supporto al *versioning* del software
 - *collaudo sistematico* come parte integrante del processo
 - *automatizzazione* degli step chiave del processo
 - cura degli *aspetti legali*: *privacy by design and by default*
(General Data Protection Regulation – GDPR, 25/5/2018)



GESTIRE IL PROCESSO DI SVILUPPO

- Scrivere il software è solo una parte del problema
- Gestire il *processo di sviluppo* è altrettanto importante, nonché *critico* e costoso
- Un aspetto critico: *il controllo di versione*
 - lo sviluppo del software passa attraverso *molti file*
 - si fanno modifiche, poi si torna indietro, ci si pente
 - più persone lavorano contemporaneamente su più parti
 - pensare di farlo col *copia & incolla* significa *avviarsi al disastro*
 - non si sa più cosa c'è in ogni file
 - non si riescono a mantenere coerenti versioni successive
 - grande spreco di tempo e di soldi
 - occorre affidarsi a un *sistema di versioning*



SISTEMI DI VERSIONING

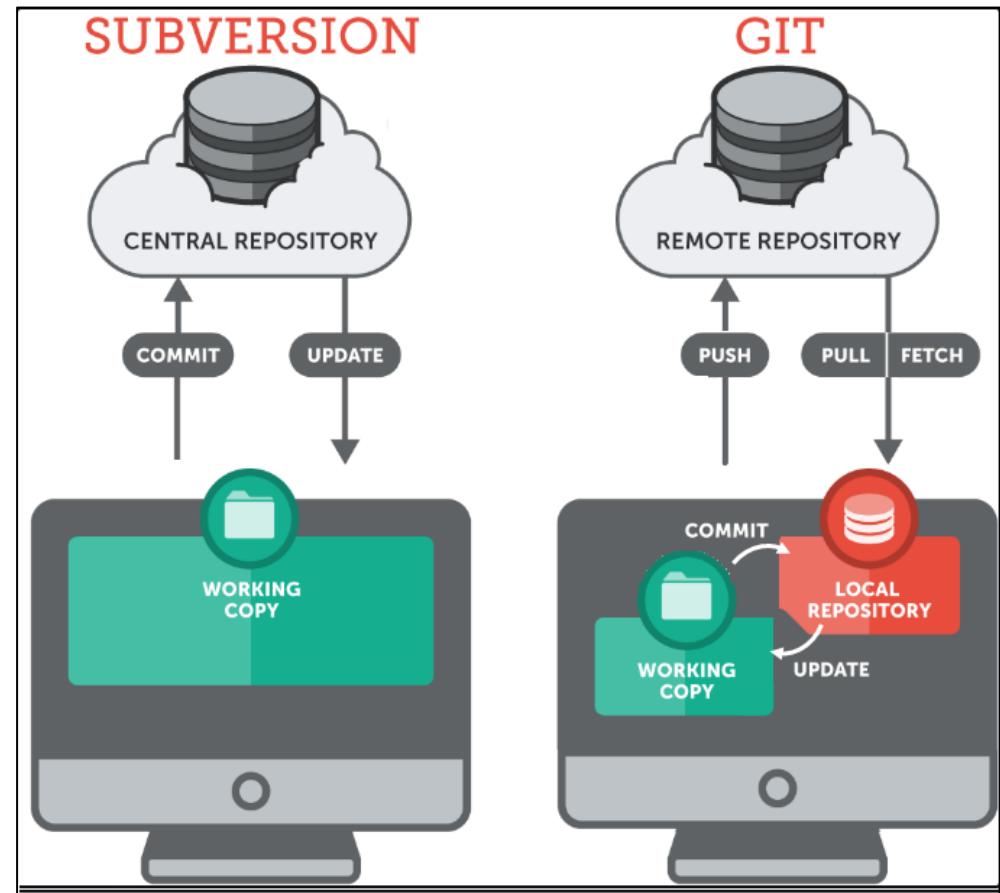
- Un **sistema di versioning** permette di:
 - tenere traccia di ogni versione di **ogni singolo file** del progetto
 - tornare indietro a versioni precedenti, senza mai perdere niente anche in caso di sbagli
 - gestire **diramazioni (branch)** sperimentali di un software, senza "pasticciare" sulla versione principale
 - riunire (**merge**) file di versioni diverse in una versione unica
 - ottenere **report** sulle differenze fra file di versioni diverse
 - gestire le **dipendenze** fra progetti diversi o fra versioni per piattaforme diverse
 - supportare il lavoro di **squadra** (chi ha fatto cosa e quando), anche nel distribuito, anche offline

SISTEMI DI VERSIONING

- I sistemi di versioning più diffusi sono **SVN** e **GIT**

- Idea base

- i file del progetto locale sono copiati, a richiesta, su un **repository** che traccia le versioni
- in **SVN**, il repository è unico e **centralizzato**
- in **GIT**, ogni pc ha il suo repository (**distribuito**), che solo a richiesta è copiato nel **repository centrale remoto**
→ *oggi più diffuso*





SISTEMI DI VERSIONING

- Esistono molti siti che offrono *hosting SVN e/o GIT*
 - per GIT,
i più noti sono GitHub, GitLab, BitBucket
 - per SVN (oggi meno usato),
GitHub, RiouxSVN, Atlassian, etc.



The screenshot shows a detailed view of a GitLab project named "2P-Kt". The left sidebar lists branches (master, 17 branches, 53 tags) and pull requests. The main area displays a list of commits, including:

- gciatto-unibo fix halt solution rendering in side (commit e2c2cc3)
- .github/workflows use s01 server for central upload
- .img update README.md
- core fix js names in Scope
- documentation use de.fayard.dependencies instead of de.fayard.buildSrcVersion
- dsl-core add common directives in PrologStdLibScope
- dsl-solve make standard channels non-removable
- dsl-theory add further theoryOf methods in PrologScopeWithTheories
- dsl-unify format code in :dsl-unify
- examples Merge branch 'develop' into feature/kt-mpp++
- gradle/wrapper use gradle 6.8

The right sidebar provides an overview of the project's statistics: 2,420 Commits, 13 Branches, 53 Tags, 2.9 MB Files, 760.6 MB Storage. It also shows a list of files and their last commits, such as README (gradle instead of gradlew in github ci ...), .img (update README.md), core (Merge branch 'feature/utils-module' into 'de...'), and documentation (use de.fayard.dependencies instead of de.fay...).



SISTEMI DI VERSIONING

- Serve poi un *client* sul proprio pc
 - Windows: TortoiseSVN, TortoiseGIT, SourceTree,...
 - Mac: SourceTree, Versions,...
 - spesso integrati negli strumenti (es. Eclipse, IntelliJ, ...)

TortoiseSVN the coolest interface to (Sub)version control

Home | About | Downloads | Translations | Support/Docs | Other tools | Support this project

 **Tortoise SVN**

Info

About
About TortoiseSVN
Screenshots
Screenshots of various dialogs
Testimonials
What users say about TortoiseSVN
News Archive
News archive

The Power of Git – in a Windows Shell

TortoiseGit provides overlay icons showing the file status, a powerful context menu for Git and much more!

Learn more [about TortoiseGit](#).

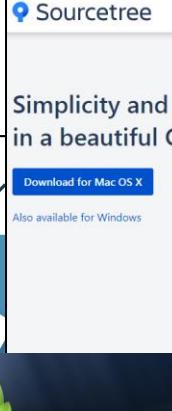


Sourcetree

Simplicity and power in a beautiful Git GUI

[Download for Mac OS X](#)

Also available for Windows



A free Git client for Windows and Mac





ESEMPIO: Subversion (SVN)

The screenshot shows the TortoiseSVN interface over a Windows file explorer window. The left pane lists files with their last modified dates:

- .svn 29/01/2013 19:58
- 2p 14/01/2013 15:47
- 2p-android 07/03/2012 17:48
- 2p-net 03/07/2012 11:49
- 2p-plugin 07/03/2012 17:53
- libraries 07/03/2012 17:52
- Manual 10/01/2013 17:44
- wiki 14/01/2013 15:43

A context menu is open on the '2p' file, with several items highlighted with red boxes:

- SVN Update
- SVN Commit...
- Sincronizzazione cartella condivisa
- Ripristina versioni precedenti
- Includi nella raccolta
- Invia a
- Taglia
- Copia
- Crea collegamento
- Elimina
- Rinomina
- Proprietà
- Check for modifications
- Revision graph
- Resolved...
- Update to revision...
- Rename...
- Delete
- Revert...
- Clean up...
- Get lock...
- Release lock...
- Branch/tag...
- Switch...
- Merge...
- Export...
- Add...
- Unversion and add to ignore list
- Create patch...

Yellow callout boxes provide additional context:

- File coerenti con la versione presente nel repository remoto (files in sync with remote repository)
- File modificato localmente ma non "committed" nel repository remoto (file modified locally but not committed to remote repository)
- Possibilità di aggiornare DAL repository o di fare il "commit" di file NEL repository (possibility to update the local repository or commit files to the remote repository)
- Molti strumenti di ausilio per gestire le versioni (many tools available to manage versions)

01-OOP intro.pptx	30/01/2013 14:44	Presentazione di ...	5.602 KB
01-OOP xfaq.pptx	24/01/2013 10:43	Presentazione di ...	147 KB
02-Array.pptx	24/01/2013 10:43	Presentazione di ...	203 KB
03-Package e namespace.pptx	24/01/2013 10:43	Presentazione di ...	823 KB



ESEMPIO: Subversion (SVN)

Differenze fra due versioni successive
dello stesso file

The screenshot shows the TortoiseMerge application interface comparing two versions of the Java file `Esempio2.java`.

Left View (Revision 10):

```
1 public class Esempio2 {  
2     public static void main(String[] args) {  
3         Counter c1 = new Counter();  
4         c1.reset(); c1.inc();  
5         System.out.println(c1.getValue());  
6         Counter c2 = c1;  
7         c2.inc();  
8         System.out.println(c1.getValue());  
9         System.out.println(c2.getValue());  
10    }  
11 }
```

Right View (Working Copy):

```
1 public class Esempio2 {  
2     public static void main(String[] args) {  
3         Counter c1 = new Counter();  
4         c1.reset(); c1.inc();  
5         System.out.println(c1.getValue());  
6         Counter c2 = c1;  
7         c2.inc();  
8         System.out.println(c1.getValue());  
9         System.out.println(c2.getValue());  
10    }  
11 }
```

Red circles highlight the title bars of both windows: "Esempio2.java Revision 10" and "Esempio2.java : Working Copy".

At the bottom of the interface, status messages include:

- For Help, press F1. Scroll horizontally with Ctrl-Scrollwheel
- Left View: ASCII CRLF
- Right View: ASCII CRLF / + 1
- Conflicts: 0
- CAP NUM SCRL



ESEMPIO: Git

The screenshot shows the SourceTree interface. On the left, the sidebar displays the repository structure: **tuprolog** (local) and **tuprolog** (remote at `E:\Home\Techni`). The local branch is `master`. The log pane shows a timeline of commits from `origin/BR-ANDRZEJ` to `REL-2.9.1`, with a focus on the commit "Merged TH-COLAMEO into master". A yellow callout bubble points to this commit with the text: "Modifiche tracciate: chi ha fatto cosa, dove l'ha fatto". The bottom right pane shows a diff view for the file `2p/trunk/.classpath`, comparing two versions of the XML code.

Modifiche tracciate:
chi ha fatto cosa,
dove l'ha fatto

```
<?xml version="1.0" encoding="UTF-8"?>
<classpath>
- <classpathentry kind="src" path="src"/>
<classpathentry kind="src" path="test/fit/fixtures">
<classpathentry excluding="TestURLClassLoader.class" kind="con" path="org.eclipse.jdt.junit4.runtime.jar"/>
<classpathentry kind="lib" path="lib/fit.jar"/>
```



PROGETTARE (PER) IL COLLAUDO

- Le garanzie di qualità di un prodotto dell'ingegneria richiedono una **attività di collaudo integrata e sistematica**

IL COLLAUDO

- **non si improvvisa**: va *accuratamente progettato*
 - non dopo: *prima ancora di costruire il sistema*
- non è un'attività accessoria: **è parte integrante del TUO lavoro**
- come tale, va **strumentato e supportato** in modo completo
- è espresso dal *piano di collaudo*
 - il piano *guida la validazione* del sistema
- non è fare "qualche stampa" (chi le legge??)



TIPI DI COLLAUDO

Esistono diversi *approcci*, con fini diversi.

BLACK-BOX TESTING

- collaudo *progettato e svolto "ai morsetti"* di un componente software, *senza sapere com'è fatto dentro*
- scopo: verificare la correttezza del comportamento esterno, indipendentemente dalla realizzazione

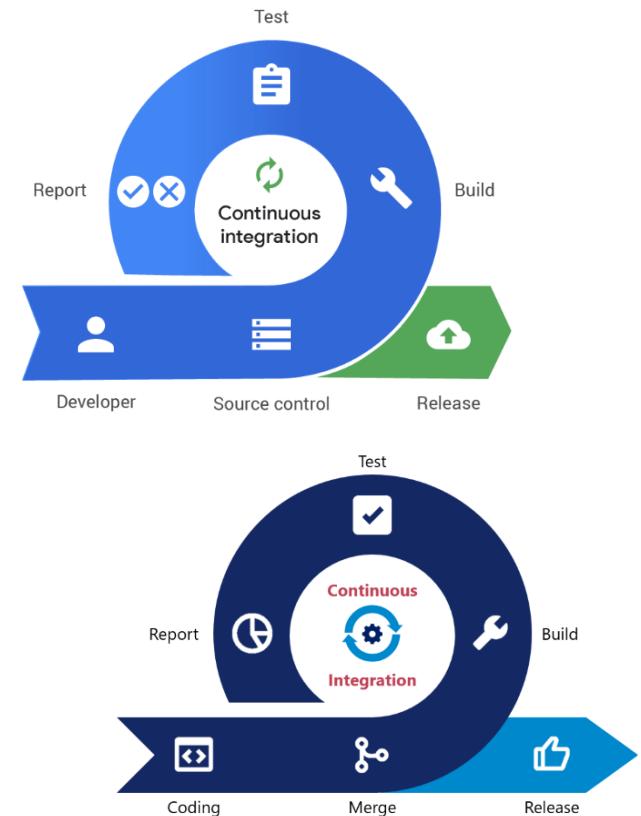
WHITE-BOX TESTING

- collaudo *progettato e svolto tenendo esplicitamente in conto com'è fatto dentro* un componente
- scopo: verificare la corretta implementazione delle varie funzioni, *stressandole una ad una* anche nei casi critici



CONTINUOUS INTEGRATION

- Il collaudo è parte integrante della strategia di sviluppo, oggi molto diffusa, detta *Continuous Integration*
 - il codice è sotto controllo di versione
 - a ogni modifica, si ri-eseguono i test e si ottengono report dettagliati
 - solo se OK, la modifica è approvata; altrimenti, si corregge o si ripristina la versione precedente
 - di tanto in tanto, si rilascia una versione ritenuta consistente (*release*)
- Tutto è sempre sotto controllo

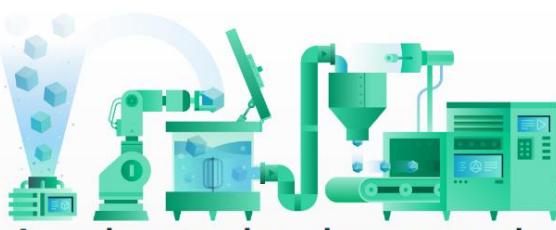




STRUMENTARE IL PROCESSO

- Per questo esistono strumenti che gestiscono / automatizzano il ciclo di sviluppi: i cosiddetti *Build tools*
- Esempi: Gradle, Maven, ...
 - in Gradle si specifica, con un apposito linguaggio, il grafo delle dipendenze fra più progetti + il relativo ciclo di sviluppo

 **Gradle Build Tool**



Accelerate developer productivity

From mobile apps to microservices, from small startups to big enterprises, Gradle helps teams build, automate and deliver better software.

Docs About Training News

 **Apache Maven Project**
<http://maven.apache.org/>

Apache / Maven / Welcome to Apache Maven

Welcome

Licence
ABOUT MAVEN
What is Maven?
Features
Download
Use
Release Notes

DOCUMENTATION

Maven Plugins
Index (category)
User Centre
Plugin Developer Centre
Maven Central Repository
Maven Developer Centre

Use

Download, Install, Run Maven
Information for those needing to build a project that uses Maven

Extend

Write Maven Plugins
Information for developers writing Maven plugins

Contribute

Help Maven
Information if you'd like to get involved. Maven is an open source community and welcomes contributions.

Configure, Use Maven and Maven Plugins

Information for developers using Maven to build their project, including a "10 minute test" that gives a practical overview of Maven's main features and a plugin list for more information on each plugin.

Improve the Maven Central Repository

Information for those who may or may not use Maven, but are interested in getting project metadata into the central repository.

Develop Maven

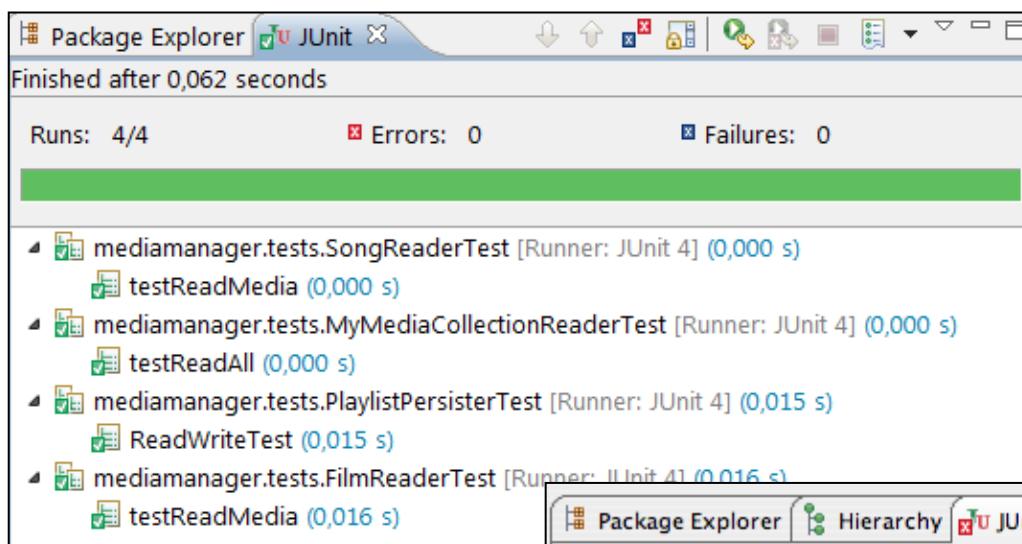
Information for those who are currently Maven developers, or who are interested in contributing to the Maven project itself.

Maven™
Download | Get Sources | Last Published: 2021-01-15



STRUMENTARE IL COLLAUDO

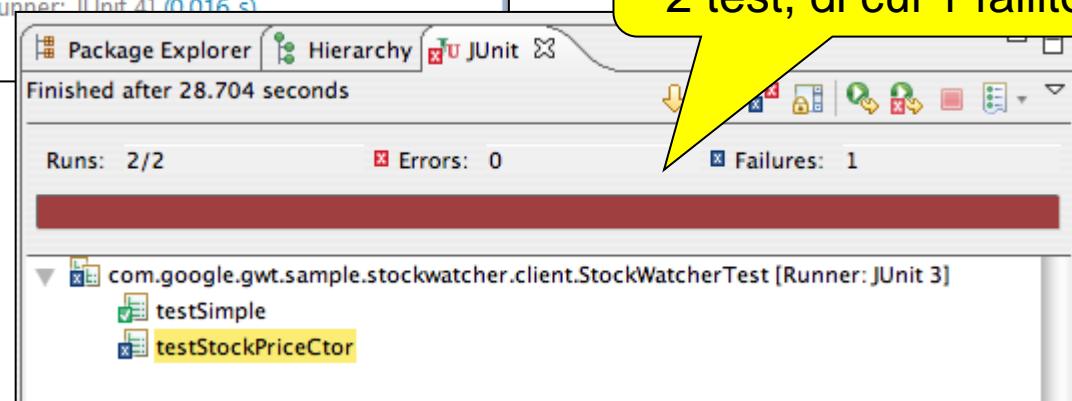
ESEMPIO: lo strumento JUnit esegue in automatico i test progettati, riportandone i risultati in forma grafica



A **failure** is when one of your assertions fails.

An **error** is when some other exception occurs, one you haven't tested for and didn't expect.

Esecuzione di un piano di collaudo:
4 test positivi su 4

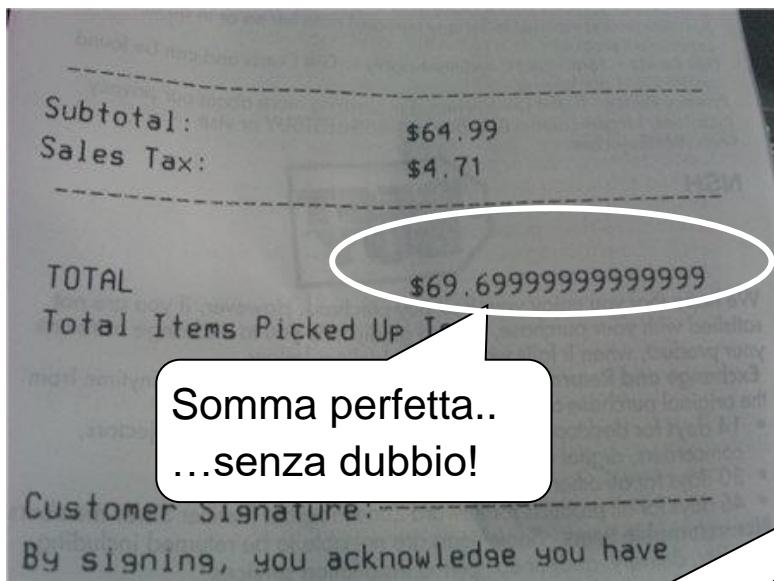


Controesempio:
2 test, di cui 1 fallito



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si richiedono per automobili, ascensori, case...
si eviterebbero "figuracce" (e danni) come questi



E questo sarebbe il prezzo più basso?? [3/2/2011]

Price	Departure	Arrival	Travel time	Operated by	Flight number
EUR 574	06:45 Bologna 10:20 Amsterdam 16:17 Atlanta	09:05 Amsterdam 14:10 Atlanta 18:06 Miami	17h21		+ Details
EUR 1,760	07:10 Bologna 09:20 Rome	08:10 Rome 14:45 Miami	13h35		+ Details
EUR 1,711	07:15 Bologna 10:50 Paris (CDG)	09:00 Paris (CDG) 10:35 Miami	13h20		+ Details
Please note: only 2 seat(s) available at this price					
Show all flights Taxes, surcharges and booking fee included					
<input checked="" type="checkbox"/> Next day Lowest price on this page					
Return Tuesday 22 February 2011					
Price	Departure	Arrival	Travel time	Operated by	Flight number
EUR 616	13:32 Miami 17:40 Detroit 09:20 Amsterdam	16:40 Detroit 07:45 Amsterdam 11:10 Bologna	15h30		+ Details
EUR 401	17:46 Miami 21:35 Atlanta 15:00 Amsterdam	19:49 Atlanta 11:50 Amsterdam 16:50 Bologna	17h04		+ Details
EUR 483	18:00 Miami 13:10 Paris (CDG)	08:20 Paris (CDG) 14:50 Bologna	14h50		+ Details
Show all flights Taxes, surcharges and booking fee included					
<input checked="" type="checkbox"/> Next day Lowest price on this page					



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

Leave	Thu, Feb 17	Hawaiian Airlines 451
	Depart:11:50am	Honolulu/Oahu, HI (HNL)
	Arrive:7:20pm	Sydney, Australia (SYD)
	Non-stop	Economy 10hr 30min View

This is an overnight flight.

Come? Parte a mezzogiorno, arriva alla sera.. ed è overnight?? Ignoranza + bug: l'algoritmo non tiene conto della linea del cambiamento di data.

IRRILEVANTE? aspetta di avere una coincidenza da prendere, e scoprire che è stata prenotata il 17 febbraio invece del 18..! Torni indietro nel tempo a prenderla?

Return	Sun, Feb 27	Hawaiian Airlines 452
	Depart:9:20pm	Sydney, Australia (SYD)
	Arrive:10:10am	Honolulu/Oahu, HI (HNL)

Non-stop

DEPARTING FLIGHT

Oahu - Honolulu (HNL) to Sydney, Australia (SYD)
Hawaiian Coach

FEB
17

HA451 Thursday, February 17, 2011 11:50 AM–
Friday, February 18, 2011 7:20 PM (next day)
Nonstop

La versione corretta

RETURN FLIGHT

Sydney, Australia (SYD) to Oahu - Honolulu (HNL)
Hawaiian Coach

FEB
27

HA452 Sunday, February 27, 2011 9:20 PM–10:10 AM
Nonstop

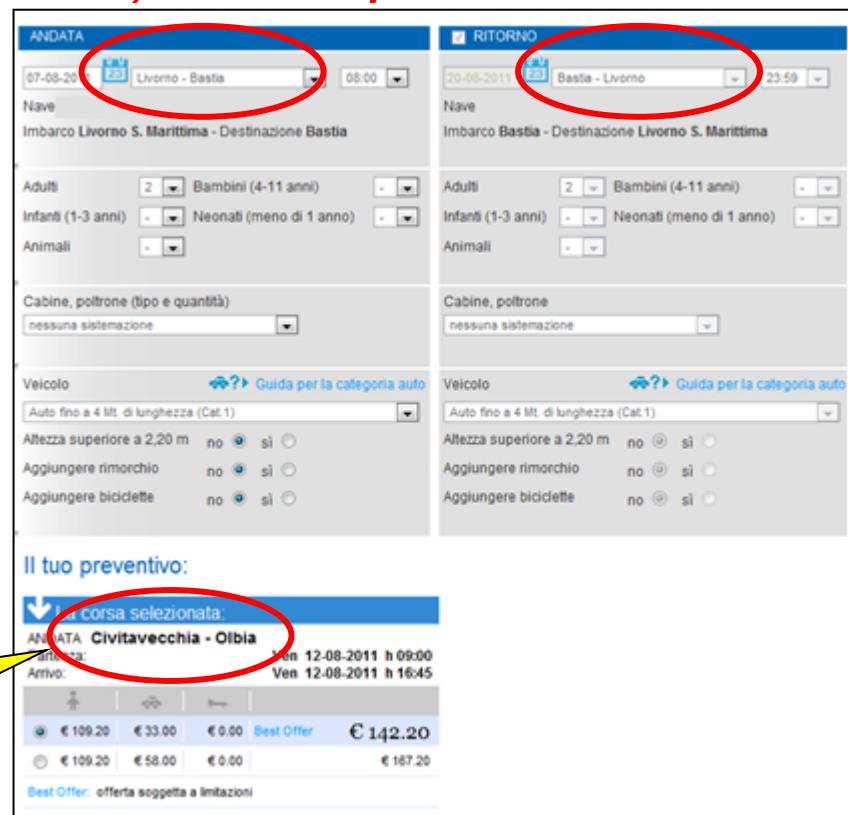
COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi



Non male il treno "not found"..
Che provenga dal binario 9 ¾ ? ☺

Tu chiedi Livorno-Bastia A/R...
..e lui ti conferma un preventivo per
Civitavecchia-Olbia??



The screenshot shows a travel booking application interface. On the left, under 'ANDATA', it lists 'Livorno - Bastia' as the departure route. On the right, under 'RITORNO', it lists 'Bastia - Livorno' as the return route. Both sections include fields for passengers (Adults, Infants, Animals), cabin/pool requests, vehicle type (auto), and height restrictions. Below these, a section titled 'Il tuo preventivo:' (Your quote) shows a summary for a 'corsa selezionata' (selected route) from Civitavecchia to Olbia. The quote details: Departure: Ven 12-08-2011 h 09:00; Arrival: Ven 12-08-2011 h 16:45; Best Offer: € 142.20. There are also other price options listed: € 109.20, € 33.00, € 0.00, € 167.20.



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

OCIAL 20/11/0731: 2.320

Vogliamo parlare della data 31/07/2011
formattata come GG/MM/AAAA ma coi
dati invertiti, sugli estratti conto di decine
di migliaia di clienti ?

E nessuno se ne accorge per mesi???

..per non dire di quando un'applicazione
crolla rovinosamente e nessuno ha pensato
alle conseguenze..





COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

..o come queste
a Roma Termini ☺





COLLAUDARE SISTEMI SOFTWARE...?

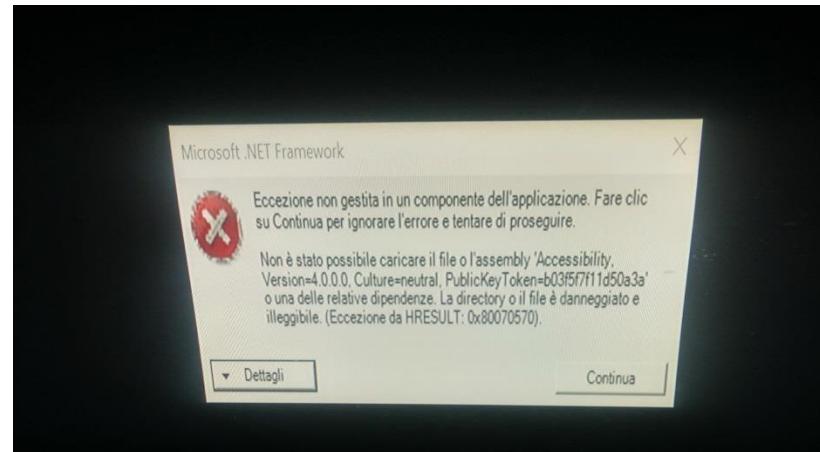
- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi





COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi



Ah, gli sportelli automatici...



COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

Per l'occasione, abbiamo riservato per te una speciale offerta:

Che dire? Un'offerta DAVVERO speciale... 😊

- cambio olio e filtro olio a 99€ → la tua vettura e 129€ → per il tuo ve...
- da oggi saremo aperti con orario continuato

Affrettati a cogliere questa opportunità!

COLLAUDARE SISTEMI SOFTWARE...?

- Se il software fosse collaudato con la stessa cura e lo stesso rigore che si usano per automobili, ascensori, ponti...
si eviterebbero "figuracce" (e danni) come questi

Il famoso **PULSATE** bianco..

Mica male due errori nella stessa macchinetta.. e sì che si vedono! ☺

..e l'altrettanto famoso
CODICEE





COLLAUDO vs PROGETTO...?

- A volte invece sono le *assunzioni di base* non rispettate a generare assurdità al momento dell'uso..
...come in questa compagnia aerea europea

16 Posso prenotare un volo per un passeggero minorenne con il Codice di Prenotazione per due persone?

Se hai un Codice di Prenotazione per due persone uno dei due passeggeri può essere minorenne.

Il sito di prenotazione ti chiederà di inserire però la data di due passeggeri adulti. Per far sì che uno dei due passeggeri sia impostato come bambino, dovrai inserire una data falsa di nascita e indicare la data di nascita corretta nel campo "Esigenze speciali di trasporto".

Stiamo lavorando per agevolare la prenotazione anche per i passeggeri minorenni, a breve sarà possibile procedere in automatico.

Per maggiori informazioni [CLICCA QUI](#).

Ma certo! Chi avrebbe mai potuto pensare che *anche un minorenne potesse prendere un aereo..???*

Dominio del problema
mal analizzato

Rimedio anche peggiore: MENTIRE nei moduli e poi usare (malamente) un campo pensato per qualcos'altro!



COLLAUDO vs PROGETTO...?

- A volte invece sono le *assunzioni di base* non rispettate a generare assurdità al momento dell'uso..
...come in questo caso molto "bolognese" ☺

09:28	par	Bologna, ALDINI		Autobus 33
09:31	arr	Bologna, PORTA SAN MAMOLO		Bologna, PORTA SAN MAMOLO
09:39	par	Bologna, PORTA SAN MAMOLO		Autobus 33
09:45	arr	Bologna, CARDUCCI		Bologna, BOLOGNA STAZIONE CENTRALE

PIANIFICATORE 2016

Per andare da ALDINI a CARDUCCI basterebbe restare sul 33... e invece veniva suggerito di scendere a porta S.Mamolo e aspettare il successivo!

Che fosse per socializzare..? ☺

IN REALTÀ...

- chi ha scritto il software ha ipotizzato che tutte le linee abbiano due capilinea..
- *ma una linea circolare viola tale assunzione* ed è stata quindi modellata *come due tronconi* "andata": Stazione/Porta S.Mamolo
"ritorno": Porta S.Mamolo/Stazione
- il software, tarato (giustamente) sull'assunzione di base, vede un itinerario composto.



COLLAUDO vs PROGETTO...?

- A volte invece sono le *assunzioni di base* non rispettate a generare assurdità al momento dell'uso..
...come in questo caso molto "bolognese" 😊

Calcola percorso

Partenza
Viale Aldini, Bologna, BO, Italia

Arrivo
Piazza di Porta Santo Stefano, Bolo

Calcola il percorso

Per ulteriori opzioni di ricerca consulta direttamente [Google Map](#)

Percorsi suggeriti

33 15 min	12:51-13:06
33 33 11 min	12:55-13:06
11 16 min	12:57-13:13
11 16 min	13:05-13:21

Legenda

Mappa Satellite

PIANIFICATORE 2020

Ora ti suggerisce di.. Andare a piedi fino al parcheggio Staveco 😊

oppure, «ovviamente», scendere a porta S.Mamolo (ma almeno ti dà la coincidenza dopo 2 minuti..)



General Data Protection Regulation

- Dal 25/5/2018 ha sostituito la Data Protection Directive
- *Obbligo* di seguire i principi della GDPR
 - **privacy by design & by default**
(non solo misure tecniche, ma anche organizzative)
 - minimalità, proporzionalità
 - **anonimizzazione, pseudonimizzazione**
 - trasferimento dati fuori dalla EU (occhio al cloud!)
 - adeguatezza delle misure di sicurezza
- Non è qualcosa che si possa "aggiungere dopo", a sistema già progettato: *va considerato fin dall'inizio*
 - ma non è (solo) un vincolo: è *un'opportunità per creare valore*



IL VOSTRO FUTURO...

- Saper progettare **sistemi di qualità** è e sarà sempre più necessario per la vostra professionalità e il vostro futuro

Assunzioni non stagionali fino a 29 anni prime 32 figure professionali ⁽¹⁾ p				24 Dicembre 2012	IL SOLE 24 ORE	estratto
Professione	Cat. ⁽²⁾	Valore assoluto 2012	In ass.			
Comessi delle vendite al minuto	PCS	23.520				
Addetti alla gestione dei magazzini e professioni assimilate	I	5.740				
Camerieri e professioni assimilate	PCS	5.560				
Addetti agli affari generali	I	5.270				
Addetti a funzioni di segreteria	I	5.150				
Personale non qualif. servizi pulizia uffici ed esercizi commerciali	PNQ	3.410				
Baristi e professioni assimilate	PCS	3.270				
Cuochi in alberghi e ristoranti	PCS	3.130				
Analisti e progettisti di software	PIS	2.890				
Addetti alla preparazione, alla cottura e alla distribuzione di cibi	PCS	2.860				
Muratori in pietra, mattoni, refrattari	OS	2.620				
Contabili e professioni assimilate	PT	2.530				
Facchini, addetti allo spostamento merci	PNQ	2.500				
Addetti sportelli assicurativi, bancari e altri intermed. finanziari	I	2.350				
Tecnici della vendita e della distribuzione	PT	2.240				

Il bilancio

Le caratteristiche delle assunzioni under 30 programmate dalle imprese per il 2012

200 mila

Le assunzioni di under 30 Sono gli inserimenti di giovani programmati dalle imprese per il 2012, in fleSSIONE di circa il 30% rispetto al 2011

33%

Non stagionali Più di un'assunzione su tre non stagionale (circa 144 mila) è rivolta ad under 30, ma il numero si è dimezzato rispetto al periodo pre-crisi

40%

Primo del Centro
Nel Centro Italia 4 assunzioni su dieci sono per giovani; nel Settentrione si registrano, invece, le dinamiche peggiori rispetto al 2011 (-32%)

13 mila

Milano e Roma al top
Nelle province di Milano e Roma si registrano più assunzioni in valore assoluto (entrambe oltre quota 13 mila); all'opposto Ragusa, Prato e Pisa

24 mila

Il record dei commessi
I profili più richiesti sono i commessi, seguiti da magazzinieri, camerieri, impiegati e segretari (tutti con oltre 5 mila assunzioni)

2.900

Progettisti software
Tra le figure «high skilled», si distinguono progettisti software (2.900), contabili (2.500) e tecnici per la vendita e la distribuzione (2.200)



GLI INTROVABILI (2012)

Indagine Unioncamere

Lavoro: ecco le figure più ricercate ma che le aziende non trovano. Situazione nelle varie Regioni

I LAVORI PIU' RICERCATI

Secondo una ricerca di Unioncamere sono oltre 65 "introvabili" nel 2012, il 16,1% del totale delle assunzioni previste per quest'anno, le quali le imprese italiane intendono fare nell'anno (406.000).

Tra i livelli di istruzione, la difficoltà di reperimento è maggiore sui laureati. Dei 58.890 laureati che le imprese intendono assumire con un contratto non stagionale, uno su cinque è introvabile (pari a 11.770 unità).

Per coloro che hanno un diploma la difficoltà di reperimento è minore (27.000 unità su 166.000 assunzioni previste).

Tra i laureati è difficile per le aziende trovare programmatore e analista software, progettisti di impianti industriali e revisori contabili. Ma non solo per questi professionisti. La ricerca guarda alla professioni con maggiore possibilità di inserimento sul mercato del lavoro: infermieri con 3.450 assunzioni previste, 950 dei quali difficili da reperire (27,4% del totale).

Ha molte possibilità di sbocco anche l'addetto all'assistenza sanitaria (1.200 assunzioni previste tra i laureati) con 2.950 assunzioni previste e il 24,5% di difficoltà di reperimento. Molto facile per le aziende invece tro-

Le professioni dei laureati: le più introvabili e quelle di maggiore sbocco

Nonostante la crisi e il crescente numero di soggetti in cerca di occupazione, di fronte alle assunzioni di laureati, ad esempio, le imprese incontrano elevate difficoltà di reperimento quando si trovano ad assumere determinate figure professionali. E' il caso del Progettista di sistemi informatici, professione per la quale i candidati ideali sono i laureati in Informatica: sono 900, delle mille circa previste per quest'anno, le assunzioni difficili da reperire, pari a quasi l'85% del totale. Nel campo dell'informatica, tra le prime venti professioni "introvabili" si incontrano, con tassi di difficoltà attorno al 30% delle assunzioni previste per quest'anno (e tutte con la prevalente richiesta di una laurea in Ingegneria informatica), anche il Consulente di software (un centinaio gli introvabili), l'Analista programmatore (circa 150 difficili da reperire) e il Programmatore informatico (più di 300 gli introvabili). Infine, con il 22% di difficoltà di reperimento, rientra anche lo Sviluppatore di software (più di 300 gli "introvabili").

Progettisti, ma non solo di sistemi informatici, perché quest'anno le imprese hanno difficoltà a trovare anche Progettisti meccanici (per cui si richiedono soprattutto ingegneri meccanici e navali), introvabili per quasi un terzo delle richieste, pari ad oltre 500 unità sulle 1.800 circa assunzioni previste. E sebbene i valori assoluti siano più bassi, piuttosto "irreperibili" sono anche altre figure di progettista, come quello di impianti industriali (sempre con preferenza per gli ingegneri meccanici e navali) e quello nel settore dei servizi (più frequentemente richiesti i laureati in economia bancaria, finanziaria e assicurativa), dove circa il 40% delle assunzioni sono difficili da reperire (circa un centinaio in entrambi i casi).

Tra le prime posizioni per difficoltà di reperimento si collocano anche i Revisori contabili (per i quali le imprese cercano soprattutto laureati in scienze economico-aziendali, del marketing e dell'amministrazione), introvabili in oltre un terzo dei casi, quasi 300 in valori assoluti su un totale di ben oltre 700 assunzioni previste per quest'anno.

I laureati nei diversi indirizzi di Ingegneria, comunque, rappresentano i candidati ideali per ben 9 delle prime 20 professioni in cui i "dottori" sono ritenuti introvabili, a dimostrazione che questo titolo di studio offre altissime opportunità di assunzione, a patto sempre di un'adeguata preparazione dei candidati, meglio se arricchita da un'esperienza, unita anche ad una flessibilità nello svolgere le professioni per cui sono chiamati a ricoprire.

Considerando invece le professioni di maggiore sbocco occupazionale, sempre tra i "dottori",



GLI INTROVABILI (2013)

Commissione Europea & Il Sole24ore

About this Site | Contact | FAQ | Cookies | Search | Legal notice

 European Commission

EMPLOYMENT, SOCIAL AFFAIRS & INCLUSION

European Commission > Employment, Social Affairs & Inclusion > ... > News

 POLICIES AND ACTIVITIES | NEWS AND MULTIMEDIA | PUBLICATIONS AND DOCUMENTS

The top 5 occupations with best employment opportunities were:

1. Personal care workers in health services
2. Software applications developers and analysts
3. Administrative and specialised secretaries
4. Mining, manufacturing and construction supervisors
5. Primary school and early childhood teachers

GRAFICI
Assunzioni: consistenza e andamento



Le prime cinque professioni tradizionali in crescita
L'Osservatorio europeo dei posti di lavoro vacanti (Evm) segnala **cinque occupazioni** tradizionali tra le più richieste e in crescita di assunzioni: servizio alla persona nell'ambito della sanità, **sviluppatore di applicazioni software e analista**, segretario amministrativo e specializzato, supervisore di miniera, di produzione e di costruzione e insegnante elementare e per la prima infanzia. L'agenda per nuove competenze e nuove occupazioni nell'ambito di Europa 2020 punta il faro sul numero insufficiente di infermieri formati e un numero crescente di infermieri e ostetrici vicini all'età pensionabile. Oltre ai tecnici medici e farmaceutici sono professioni in espansione nell'ambito sanitario anche dentisti, farmacisti e fisioterapisti.



GLI INTROVABILI (2014)

Indagine Unioncamere (*Il Sole24ore*, 21/11/2014)

Dall'esperto software allo stuccatore: introvabile una professione su 10

di Alessia Tripodi 21 novembre 2014 Commenti (15)

Tweet 15

Consiglia 213

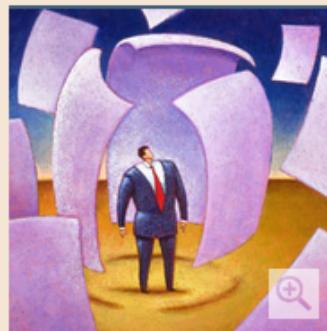
g+1 4

My24



A

A



Ingegneri in Lombardia, stuccatori edili e musicisti in Veneto, accompagnatori turistici nel Lazio, organizzatori di eventi in Piemonte. È questa la mappa delle figure “introvabili” per le imprese italiane secondo i dati dell’Osservatorio Excelsior di Unioncamere presentati ieri a Verona in occasione di «Job&Orienta».

DOCUMENTI

4

▼ [I numeri del gap](#)

Secondo i dati, nel 2014 le lauree più ricercate ma più difficili da reperire sono quelle in ingegneria, soprattutto in Lombardia. Delle 530 assunzioni di ingegneri esperti di software in tutta Italia, infatti, 210 (circa il 40%) sono considerate introvabili, così come i 300 profili di analista programmatore sui 930 ricercati in tutto il Paese nel 2014 (32%). Nella classifica delle



GLI INTROVABILI (2015)

Unioncamere (*Il Sole24ore*, 27/11/2015)

In Italia ci sono 76mila profili professionali introvabili

Statisticci, ingegneri, analisti e progettisti, ma anche tanti operatori nel settore del mobile e arredamento, abbigliamento, moda ed elettronico

Adriano Palazzolo - Ven, 27/11/2015 - 15:50



Se gli statisticci, gli **ingegneri** e i laureati in materie scientifiche "sono i più difficili da reperire" tra i dottori, problemi non trascurabili sono evidenziati anche tra i diplomati e quanti hanno conseguito una qualifica professionale. Sebbene per questi ultimi le difficoltà di reperimento siano mediamente inferiori a quella dei laureati (rispettivamente, il 10 e il 9,1% del totale delle assunzioni), per alcuni indirizzi le imprese segnalano quote anche consistenti di "introvabili".

Tra i diplomati, le difficoltà salgono decisamente sopra la media nel caso dell'indirizzo **telecomunicazioni** (20,4%, pari a 150 delle 760 unità programmate), dell'indirizzo informatico (19,1%, per 970 unità delle 5.060) e dell'indirizzo grafico-pubblicitario (18,8%, per 230 delle 1.230 assunzioni previste).

Tra i candidati con qualifica di formazione professionale, i problemi nel reperimento sono particolarmente sensibili nel caso dell'indirizzo **tessile**, abbigliamento e moda, arrivando a quasi un'assunzione su tre, mentre si attestano intorno a una su 6 nel caso dell'indirizzo legno-mobile-arredo e di quello elettrotecnico.

Riguardo le professioni dei laureati, Unioncamere registra oltre 2mila opportunità per gli analisti informatici (2.110 i ricercati, la metà dei quali introvabili), seguiti dai **progettisti** di impianti per l'automazione industriale (670 totali, 210 di difficile reperimento), quindi i tecnici specialisti di applicazioni informatiche (3.130, 540 dei quali ritenuti difficili da reperire).

(*Panorama*, 16/2/2015)

I dati

Secondo le ricerche condotte dall'agenzia di selezione del personale Page Personnel, i profili professionali con 3 – 5 anni di esperienza più richiesti nel nostro Paese trovano sbocco innanzitutto nell'area trasporti e spedizioni, che si rivela un mercato in continua espansione. Ottime prospettive anche per tecnici e ingegneri, con un focus su **ingegneri informatici** e **ingegneri meccanici** che possono aspirare a professioni ben retribuite, ben tutelate e che assicurano esperienze di lavoro stimolanti ed interessanti, ma ancora molto di nicchia e poco conosciute. Infine, il settore digital si fa notare perché è caratterizzato da una **disoccupazione prossima allo zero**.



GLI INTROVABILI (2017)

Unioncamere (*Il Sole24ore*, 2017)

24 ORE

IMPRESA & TERRITORI | Giovani e lavoro: introvabili



INTROVABILI

Professione	Numeri
Operai metalmeccanici ed elettromeccanici	14.990
Tecnici informatici e della produzione	14.430
Operatori della cura estetica	6.580
Progettisti e ingegneri	5.540
Formatori	4.590
Informatici, fisici e chimici	3.820

Le "primule rosse" sono gli informatici, insieme ai formatori e agli ingegneri. Ma anche i laureati in lingue e matematica sono spesso introvabili. In barba alla disoccupazione giovanile oltre i livelli di guardia (35,4% a giugno), ci sono profili che le imprese faticano a selezionare, con il rischio di posti di lavoro destinati a restare vuoti. A segnalare il sistema informativo Excelsior di Unioncamere e Anpal (Agenzia nazionale politiche attive del lavoro), che evidenzia come su 339mila giovani che le aziende puntano a inserire tra luglio e settembre di quest'anno, il 23% è considerato difficile da "rintracciare" sul mercato.

sia sulle chance lavorative in sé, visto che i neolaureati in ingegneria godono di tassi di occupazioni e livelli retributivi superiori alla media. Secondo i numeri di AlmaLaurea, i dottori magistrali in ingegneria che lavorano, a cinque anni dal titolo, sono l'88% del totale, dietro solo a medicina (93%). Anche le retribuzioni mensili nette sono tra le più alte in assoluto: 1.717 euro, contro i 1.405 euro di media degli altri gruppi disciplinari.



GLI INTROVABILI (2018)

Unioncamere (2018)



Most sought-after jobs in Germany

The ten most frequently sought jobs in Germany:

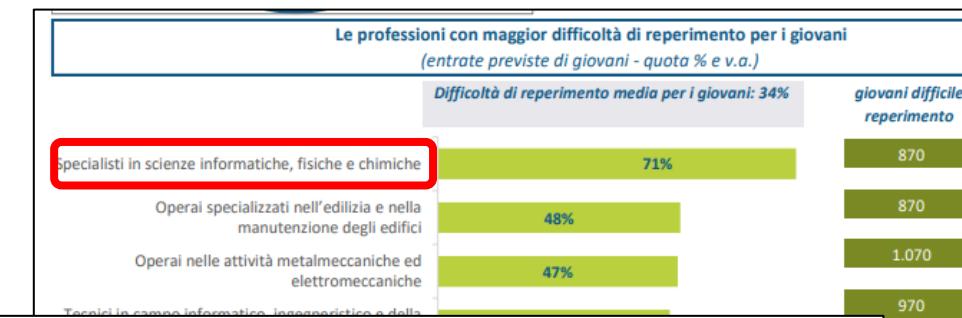
- 1 Software developer and programmer
- 2 Electronics engineer, electrician
- 3 Healthcare worker and nurse
- 4 IT consultant, IT analyst
- 5 Economist, business administrator
- 6 Account manager, client consultant
- 7 Production assistant
- 8 Sales representative, sales assistant
- 9 Sales manager, product manager
- 10 Architect, structural engineer

Source: DEKRA Akademie 2018



GLI INTROVABILI (2020)

Unioncamere (2020)



Le professioni più richieste nell'ambito tech

La ricerca di Experis invece si è focalizzata sulle figure tech più ricercate del 2021. Si tratta di professionisti diventati fondamentali per la transizione tecnologiche delle imprese, e che possono percepire stipendi molto generosi, proprio per la loro specializzazione. Sono i **developer, machine learning specialist, cloud architect, cyber security manager, data scientist/data analyst, e system administrator**. Se vi state chiedendo cosa siano e in cosa consistano, sappiate che hanno a che fare con la crescita impetuosa dell'**intelligenza artificiale** e del **machine learning**, i nuovi metodi di programmazione che permettono a computer e robot di apprendere informazioni partendo dai dati. I compatti che si stanno dotando maggiormente di questi profili sono quelli dei **servizi, finanza, telecomunicazioni, pharma e sanità**.

Le 5 professioni più richieste, secondo Union Camere

L'analisi del Sistema Informativo Excelsior promosso da Union Camere e Ampal evidenzia che le professioni più richieste sono cinque e sono ben distribuite in tutti i settori. La prima riguarda il settore della programmazione e sviluppo è in assoluto l'ambito lavorativo più dinamico degli ultimi anni con il digitale che ha acquisito un ruolo strategico essenziale in tutte le aziende. Il **Developer**, l'**ingegnere informatico** che si occupa di progettare, sviluppare e gestire non solo siti internet e app, ma anche codici per sistemi operativi e software di uso interno all'azienda, andando a supportare così il percorso di Digital Transformation che tutte le imprese stanno affrontando.

In seconda posizione c'è il **Web Marketing Specialist** che assume un ruolo strategico sempre più decisivo, pianificando e realizzando la presenza online delle aziende. Spazio anche nell'ambito delle risorse umane con l'**HR Specialist**, professionista nella ricerca, selezione e gestione del personale. In quarta posizione ci sono le figure professionali di **qualità e controllo nella grande distribuzione** e in quinta le professioni relative alla **sicurezza digitale** come **AI Specialist, Data Protection Officer, Big Data Analyst**.

Gli indirizzi di studio più richiesti					
	Entrate previste (v.a.)	% sul totale	% diff. reperimento	% con esperienza	
Indirizzo ingegneria	31.510	16%	40%	84%	
Indirizzo economico	8.520		45%	80%	
Segnamento e formazione	8.020		31%	78%	
Indirizzo sanitario e paramedico	3.840		40%	91%	
Indirizzo scienze matematiche, fisiche e informatiche	2.930		44%	93%	
	1.820		56%	91%	



GLI INTROVABILI (2021)

Unioncamere (2021-22)

Gli indirizzi di studio più richiesti				
	Entrate previste (v.a.)	% sul totale	% diff. reperimento	% con esperienza
UNIVERSITARIO	88.330	19%	43%	87%
Indirizzo economico	25.360	27%	82%	
Indirizzo ingegneria	21.670	53%	90%	
Indirizzo insegnamento e formazione	9.580	42%	92%	
Indirizzo sanitario e paramedico	6.960	60%	91%	
Indirizzo scienze matematiche, fisiche e informatiche	6.560	70%	86%	

I profili STEM maggiormente difficili da reperire

Q: Di quali figure in ambito STEM in particolare avrebbe bisogno la sua azienda?



Imprese che hanno cercato risorse STEM ma non le hanno trovate tutte, in collaborazione con SWG e grazie al contributo di Monitor Deloitte



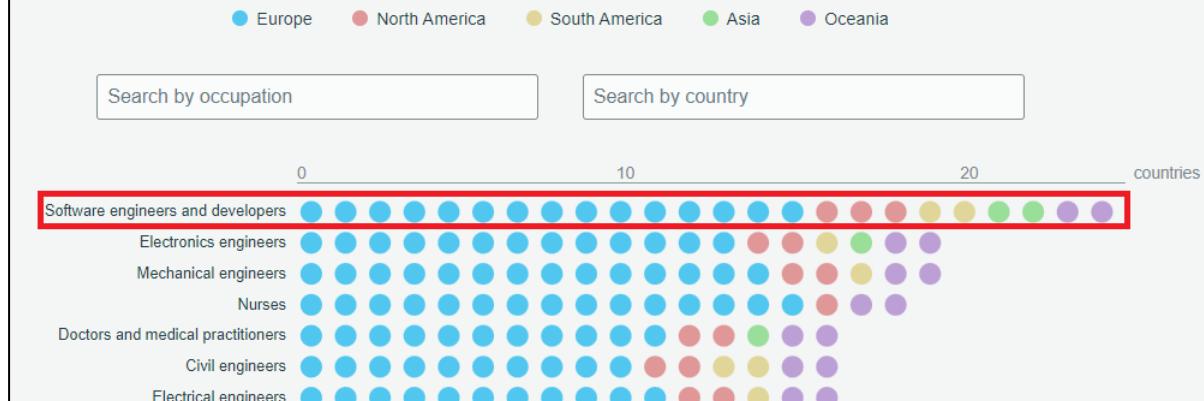


GLI INTROVABILI (2022)

LE PROFESSIONI PIÙ RICHIESTE NEL MESE DI GENNAIO 2022 *				
secondo la classificazione delle professioni - ISTAT 2011 (3 digit)				
	ENTRATE PREVISTE GENNAIO 2022	Ripartiz x 1000	Difficoltà a reperire (%)	ENTRATE PREVISTE gennaio - marzo 2022
ENTRATE	457.650	1.000	38,6	1.158.150
DIRIGENTI	1.340	2,9	54,0	2.510
PROFESSIONI INTELLETTUALI, SCIENTIFICHE E DI ELEVATA SPECIALIZZAZIONE	35.580	77,7	43,2	77.350
Specialisti in scienze matematiche, informatiche, chimiche, fisiche e naturali	5.920	12,9	61,9	13.470
Ingegneri e professioni assimilate	7.400	16,2	52,9	15.280
Specialisti delle scienze gestionali, commerciali e bancarie	7.490	16,4	33,8	14.550
Specialisti in discipline artistiche-espresive	2.100	4,6	17,0	4.820
Altri specialisti dell'educazione e della formazione				
PROFESSIONI TECNICHE				
Tecnici informatici, telematici e delle telecomunicazioni				
Tecnici in campo ingegneristico				

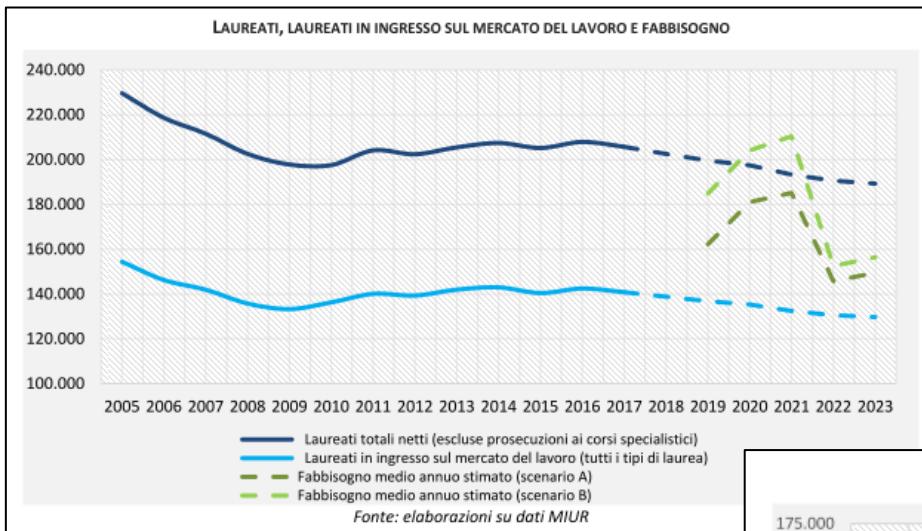
THE WORLD'S MOST IN DEMAND PROFESSIONS

Across the developed countries of the world, skilled professionals are in high demand. Software engineers are needed in 24 countries, nurses are needed in 18, while 11 countries report a shortage of accountants. Explore the chart below to see which occupations are most in demand across the world and discover which skills are needed.

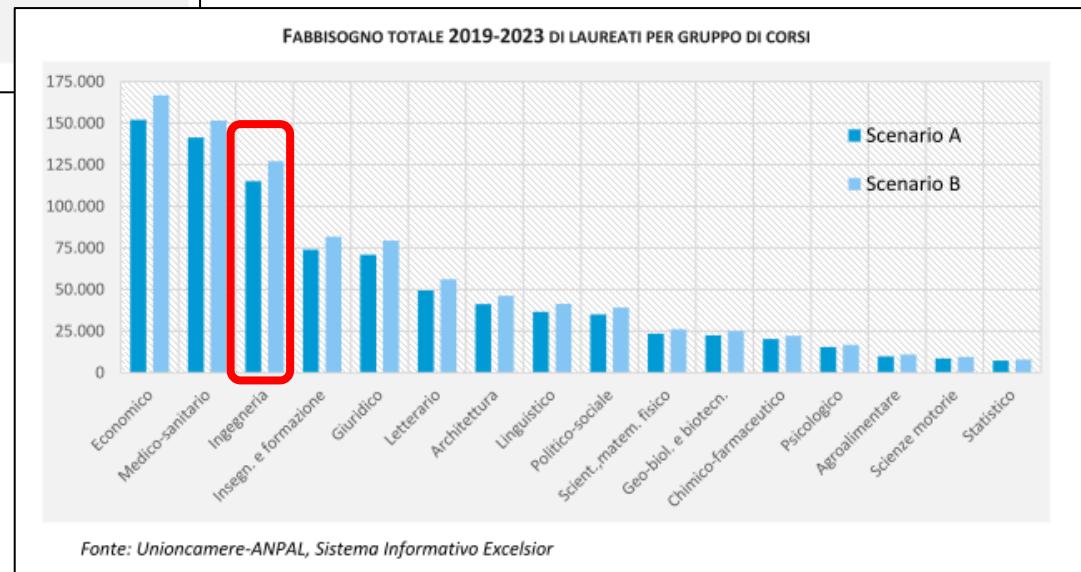




GLI INTROVABILI (previsione al 2023)



INDIRIZZO DI STUDIO	OFFERTA PREVISTA	FABBISOGNO TOTALE (V.A.)		FABBISOGNO/OFFERTA	
		SCENARIO A	SCENARIO B	SCENARIO A	SCENARIO B
Totale laureati	664.700	823.400	908.100	1,24	1,37
Economico-statistico	92.600	159.300	174.600	1,72	1,89
Medico-sanitario	100.000	140.000	160.000	1,00	1,00
Ingegneria	92.800	115.200	127.100	1,24	1,37
Insegnamento-formazione	42.500	62.500	52.500	1,95	2,10
Giuridico	48.700	70.900	79.400	1,46	1,63
Politico-sociale	57.500	35.000	39.100	0,61	0,68





DIVENTARE INGEGNERI INFORMATICI

- Per diventare validi ingegneri informatici bisogna "solo" *saper programmare bene*, in linguaggi moderni e di moda?
- NO !**
- Serve molto di più:
 - *saper pensare*, riflettere sulle cose (anche filosoficamente)
 - *saper impostare l'attività di progetto* di un sistema complesso
 - comprendere che il progetto non è il codice
 - comprendere che progettare non è implementare
 - interiorizzare l'idea che **il collaudo è parte integrante della propria attività**, non qualcosa "da fare dopo" o delegabile ad altri



IL PROBLEMA DEL PROGETTO

Problema:

spesso si pensa (solo) in base ai linguaggi conosciuti

MA

i linguaggi "classici" (C, ...) non offrono strumenti mentali tarati per affrontare la progettazione di sistemi software complessi

Problema del "foglio bianco":

- da dove si parte?
- come si comincia a "immaginare" la soluzione ?
- le metodologie, gli approcci usati per affrontare problemi "piccoli" (algoritmi) scalano su problemi "grandi" (sistemi software)?
Si riescono ancora a usare con efficacia?



IL TEMA DELLA SCALABILITÀ

- L'essere umano ha la naturale tendenza a pensare che tutto sia *scalabile*
 - ha funzionato bene su 3 elementi, perché non dovrebbe funzionare altrettanto bene con 300 o 3000 o 300000..?
 - ho sempre usato questo approccio per progettare finora, perché non dovrebbe reggere anche più in grande domani?
 - ho sempre fatto così, perché mi vuoi costringere a cambiare?
- Purtroppo, ciò è spesso una *illusione mentale*
 - l'essere umano è *intrinsecamente pigro*: ripensare ai propri approcci, cambiare abitudini e metodi di lavoro *costa fatica*
 - ..ma nei fatti *gli strumenti (fisici e mentali) quasi mai scalano all'aumentare della complessità!*
 - per avvitare cinque viti basta un cacciavite.. *ma per la Tour Eiffel?*



LINGUAGGI & CONTESTO

I linguaggi "classici", come il C:

- sono nati quando l'informatica era allo *stadio artigianale*
 - l'attenzione era sul definire un "linguaggio per codificare algoritmi"
 - i programmi erano piccoli sistemi, gestiti tipicamente da una singola persona (o poche persone) secondo il proprio stile di lavoro
 - l'hardware non era certo quello di oggi..!
- non erano pensati per supportare *un'attività di progetto*
 - il progetto "era" il codice (o al massimo un diagramma di flusso..)
- non erano pensati per supportare *una attività di collaudo sistematica di carattere ingegneristico - industriale*
 - il collaudo (quando c'era..) era un'attività "*fai da te*", non strumentata
 - i test (quando esistevano) erano tipicamente *mischiati nel codice*



LINGUAGGI & CONTESTO

Un tale approccio oggi **non può reggere**:

- è *impensabile* progettare e mantenere col "fai da te" un sistema complesso, costoso e spesso "mission critical"
- il linguaggio diviene solo *uno* degli strumenti di una *attività ben più ampia di progetto e sviluppo*, che va organizzata
- soprattutto, *il linguaggio non può più limitarsi a fornire solo gli elementi linguistici*, lasciando al progettista l'onere di affrontare tutta la complessità:
deve fornire concetti, metafore e strumenti adatti a governare la complessità dei sistemi software



VERSO NUOVI LINGUAGGI ED ECOSISTEMI

I modelli di progettazione e i linguaggi moderni sono nati per cercare di superare questo limite

- al di là delle differenze, essi introducono *nuovi concetti, nuove metafore e nuovi strumenti mentali* per riuscire a *concepire e governare il progetto, lo sviluppo e il collaudo* di sistemi software complessi
- la varietà di metafore e concetti è *la ragione di fondo per l'esistenza di tanti linguaggi* e il loro continuo proliferare

I modelli e linguaggi A OGGETTI costituiscono uno dei *maggiori casi di successo* degli ultimi decenni
ultimamente sempre più spesso *BLENDED* con concetti chiave provenienti dai *LINGUAGGI FUNZIONALI*



NEW LIFE TO OLD IDEAS

Paradigmi funzionali alla riscossa

- Per decenni, i paradigmi di programmazione e i linguaggi *funzionali*, nonostante il notevole contributo di idee "forti", sono stati penalizzati dalla *sintassi poco user-friendly, inadatta al "grande pubblico"*
 - spesso confinati nell'accademia, o in gruppi di "affezionati utenti"
 - spesso visti "con sospetto" dai "veri programmati" (...)
- Ma nell'ultimo decennio *l'aumento della complessità dei sistemi* ha dato loro nuova linfa in nuove, moderne forme
 - **LINGUAGGI BLENDED:** *idee originate dal paradigma funzionale re-innestate e adattate sui moderni linguaggi "main stream"* – sfruttando le infrastrutture esistenti ☺



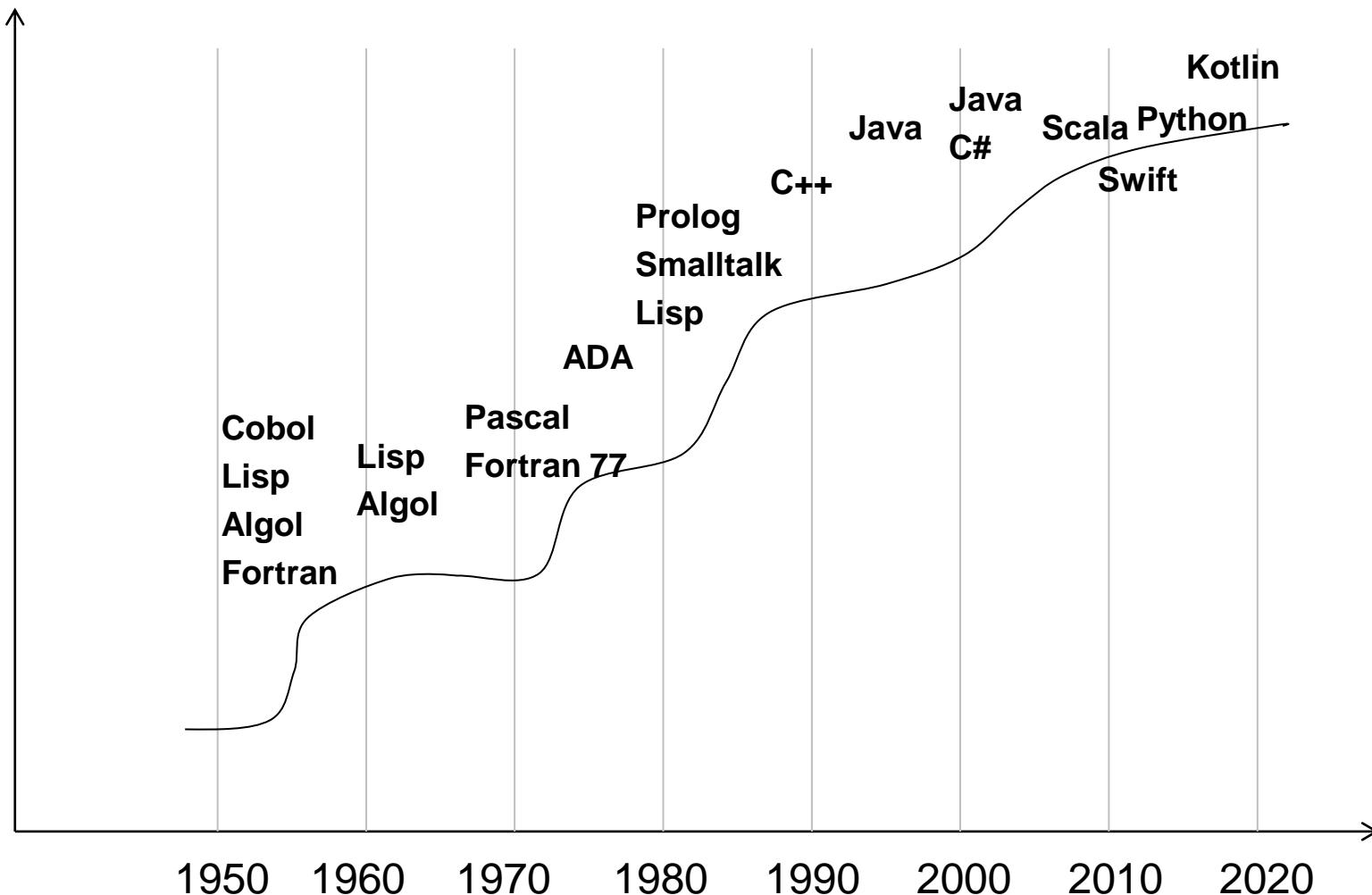
NUOVI PUNTI DI VISTA, NUOVE ISPIRAZIONI

Alcune idee "from the functional programming world"

- distinzione variabili / valori (**var** vs **val**)
- abolizione dei tipi primitivi: "*everything is an object*"
- costrutti come espressioni: "*everything is an expression*"
- strutture dati immodificabili: "*compute by synthesis*"
- funzioni come first-class entities
 - *chiusure, lambda expression*
 - *operatori come funzioni, operatori come metodi*
 - *lazy evaluation*
- stile più conciso
- ...



VERSO NUOVI LINGUAGGI





VERSO NUOVI LINGUAGGI

Scala

Scala 1.0	2004
Scala 2.0	2006
...	
Scala 2.13	2019
...	
Scala 3	2021

C# history

C# 1.0	2002
C# 2.0	2005
C# 3.0	2007
C# 4.0	2010
C# 5.0	2012
C# 6.0	2015
C# 7.0	2017
C# 8.0	2019
C# 9.0	2020
C# 10.0	2021

Kotlin

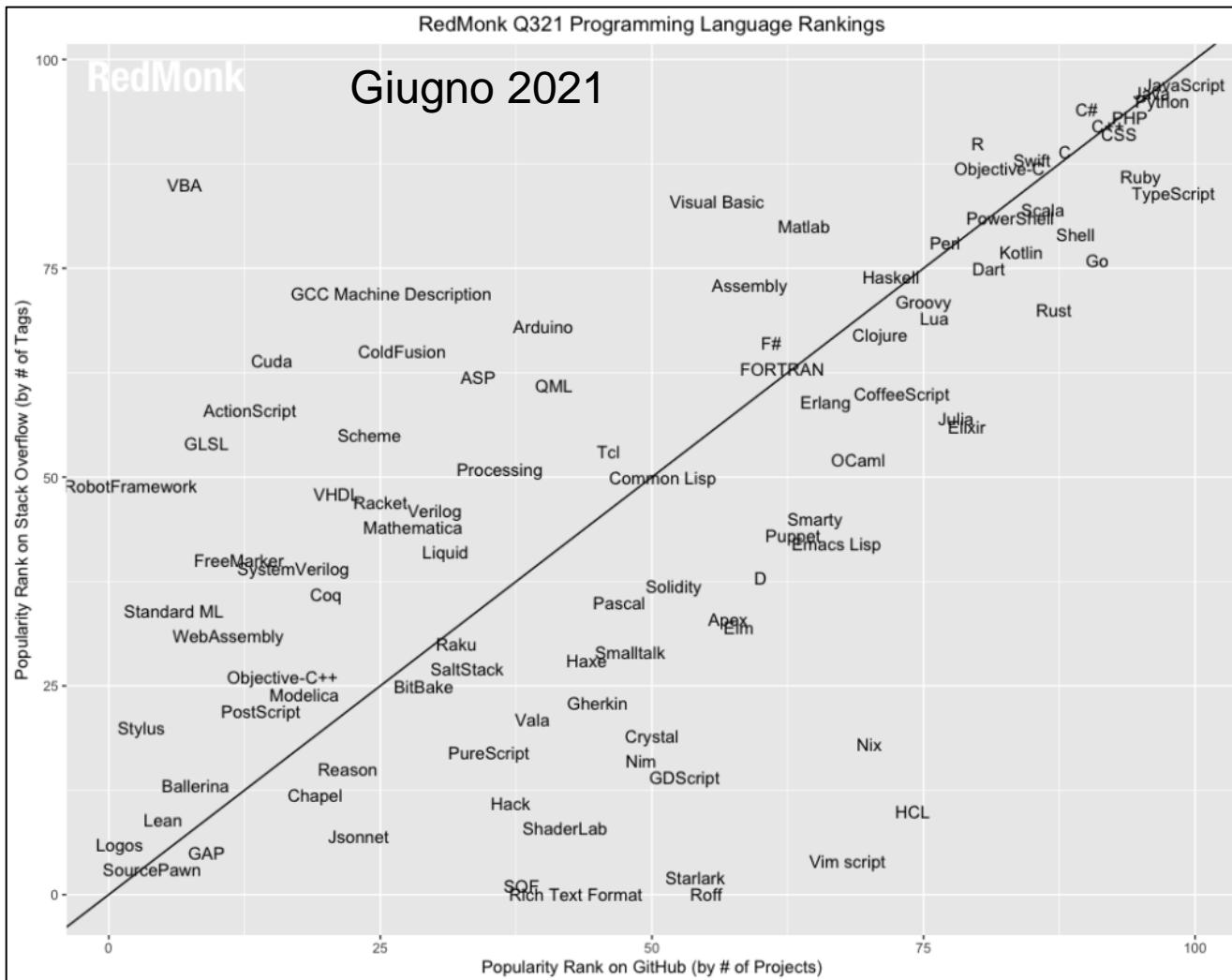
kotlin 1.0	2016
kotlin 1.2	2017
kotlin 1.3	2018
kotlin 1.4	2020
...	
Kotlin 1.6	2021

Java history

JDK 1.0 (1996)
JDK 1.1 (1997)
J2SE 1.2 (1998)
J2SE 1.3 (2000)
J2SE 1.4 (2002)
J2SE 5.0 (2004)
Java SE 6 (2006)
Java SE 7 (2011)
Java SE 8 (2014)[LTS]
Java SE 9 (2017)
Java SE 11 (2018)[LTS]
...
Java SE 17 (2021) [LTS]
...

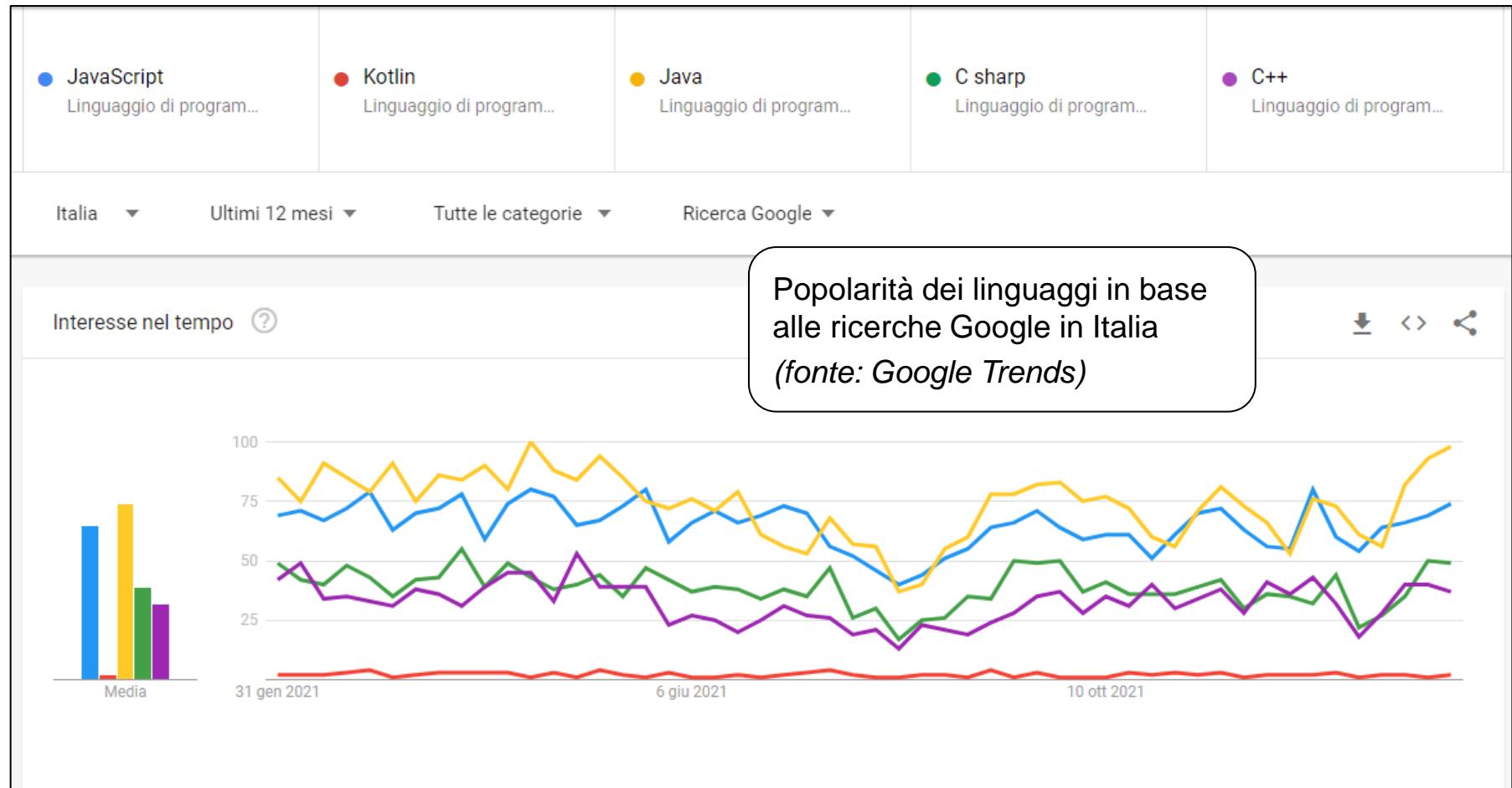


VERSO NUOVI LINGUAGGI



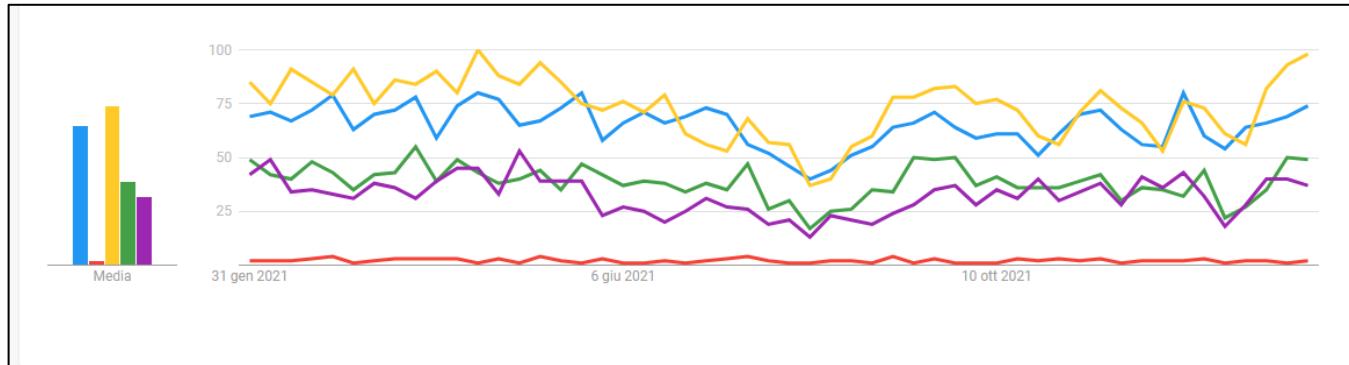


VERSO NUOVI LINGUAGGI

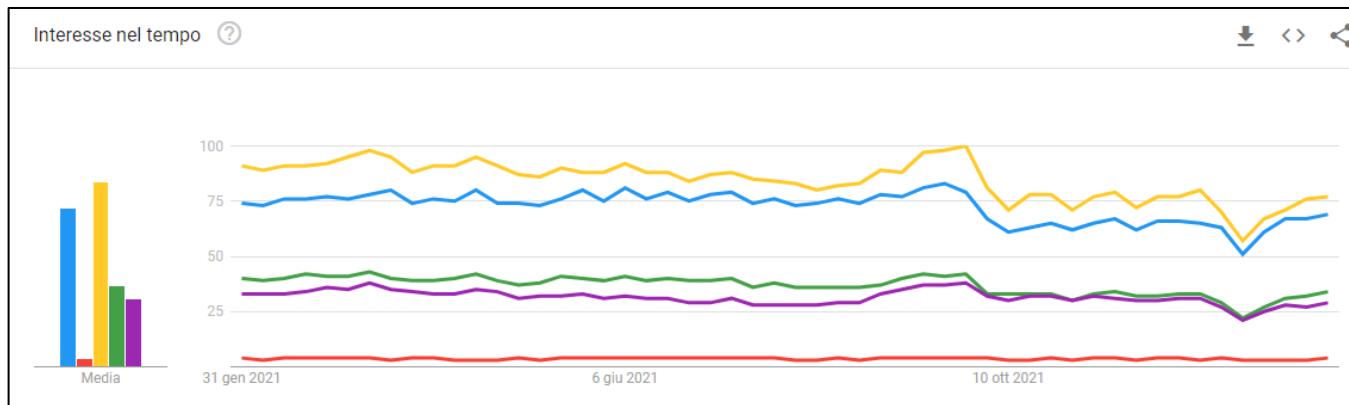




VERSO NUOVI LINGUAGGI



Italia



mondo

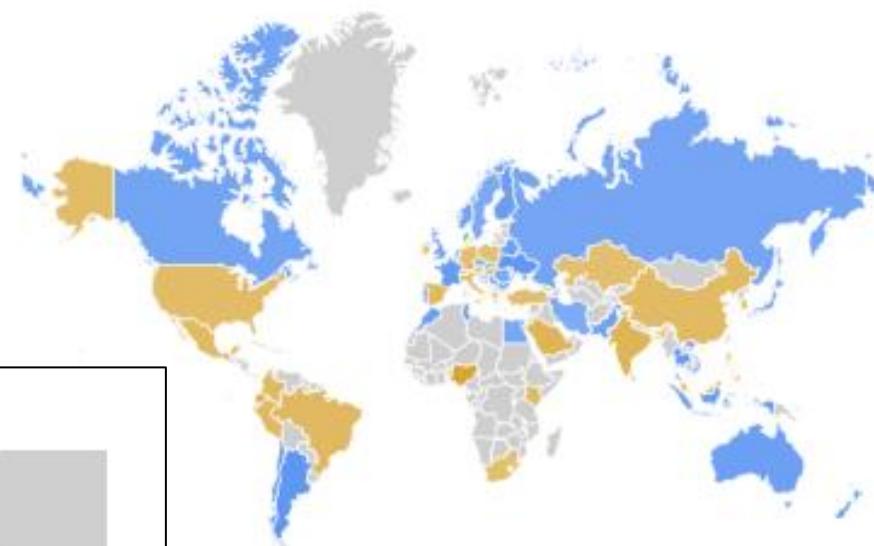
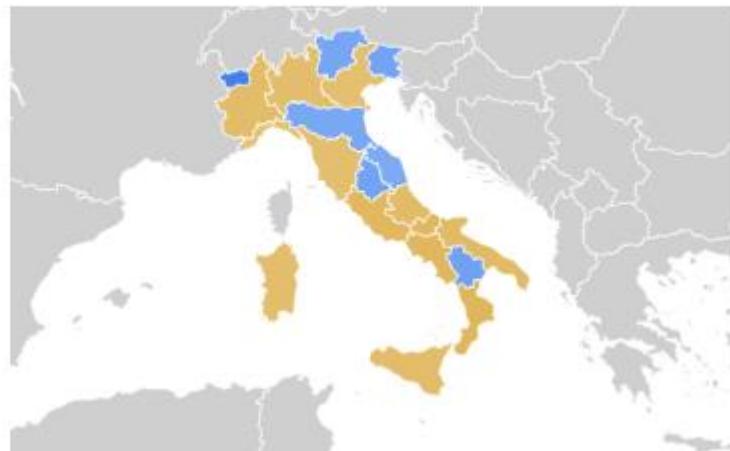


VERSO NUOVI LINGUAGGI

● JavaScript ● Kotlin ● Java ● C sharp ● C++

Popolarità dei linguaggi in base
alle ricerche Google
(fonte: *Google Trends*)

● JavaScript ● Kotlin ● Java ● C sharp ● C++





VERSO NUOVI LINGUAGGI

PYPL PopularitY of Programming Language

Jan 2022

The PYPL PopularitY of Programming Language Index is created by analyzing how often language tutorials are searched on Google.

The more a language tutorial is searched, the more popular the language is assumed to be. It is a leading indicator. The raw data comes from Google Trends.

If you believe in collective wisdom, the PYPL Popularity of Programming Language index can help you decide which language to study, or which one to use in a new software project.



Rank	Change	Language	Share	Trend
1		Python	28.74 %	-1.8 %
2		Java	18.01 %	+1.2 %
3		JavaScript	9.07 %	+0.6 %
4	↑	C/C++	7.4 %	+1.1 %
5	↓	C#	7.27 %	+0.7 %
6		PHP	6.06 %	+0.0 %
7		R	4.19 %	+0.3 %
8		Objective-C	2.27 %	-1.4 %
9		Swift	1.91 %	-0.2 %
10		TypeScript	1.74 %	-0.0 %
11		Matlab	1.74 %	+0.0 %
12		Kotlin	1.71 %	+0.0 %
13		Go	1.19 %	-0.1 %
14		VBA	1.14 %	-0.0 %
15		Ruby	1.07 %	-0.0 %
16		Rust	0.98 %	-0.0 %
17	↑↑↑	Ada	0.75 %	+0.1 %
18	↑↑↑	Dart	0.72 %	+0.2 %
19	↑↑↑	Abap	0.66 %	+0.2 %
20	↓↓	Visual Basic	0.64 %	-0.0 %



VERSO NUOVI LINGUAGGI

Jan 2017

Jan 2021

Jan 2022

Worldwide, Jan 2017 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Java	23.1 %	-0.9 %
2		Python	14.4 %	+2.7 %
3		PHP	9.7 %	-1.0 %
4		C#	8.4 %	-0.1 %
5	↑↑	Javascript	7.7 %	+0.5 %
6		C	7.1 %	-0.3 %
7	↓↓	C++	7.0 %	-0.6 %
8		Objective-C	4.4 %	-0.6 %
9	↑	R	3.4 %	+0.5 %
10	↓	Swift	3.0 %	+0.1 %
11		Matlab	2.6 %	-0.3 %
12		Ruby	2.1 %	-0.1 %
13		Visual Basic	1.5 %	-0.2 %
14		VBA	1.5 %	+0.0 %
15	↑	Scala	1.2 %	+0.4 %
16	↓	Perl	0.9 %	-0.1 %
17		lua	0.5 %	+0.0 %

Worldwide, Jan 2021 compared to a year ago:

Rank	Change	Language	Share	Trend
1		Python	30.44 %	+1.2 %
2		Java	16.76 %	-2.0 %
3		JavaScript	8.44 %	+0.3 %
4		C#	6.53 %	-0.7 %
5	↑	C/C++	6.33 %	+0.3 %
6	↓	PHP	6.05 %	-0.2 %
7		R	3.87 %	+0.1 %
8		Objective-C	3.71 %	+1.2 %
9		Swift	2.14 %	-0.3 %
10		TypeScript	1.78 %	-0.0 %
11		Matlab	1.74 %	-0.1 %
12		Kotlin	1.7 %	+0.0 %
13	↑↑	Go	1.33 %	+0.1 %
14	↓	VBA	1.2 %	-0.2 %
15	↓	Ruby	1.12 %	-0.2 %
16	↑↑	Rust	1.03 %	+0.3 %
17	↓	Scala	0.72 %	-0.3 %

Worldwide, Jan 2022 compared to a year ago:

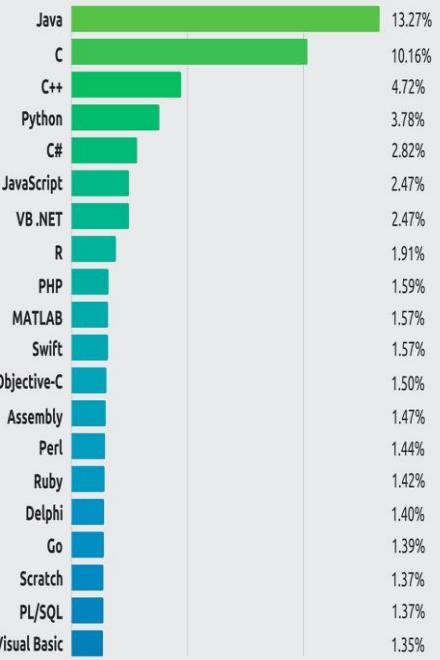
Rank	Change	Language	Share	Trend
1		Python	28.74 %	-1.8 %
2		Java	18.01 %	+1.2 %
3		JavaScript	9.07 %	+0.6 %
4	↑	C/C++	7.4 %	+1.1 %
5	↓	C#	7.27 %	+0.7 %
6		PHP	6.06 %	+0.0 %
7		R	4.19 %	+0.3 %
8		Objective-C	2.27 %	-1.4 %
9		Swift	1.91 %	-0.2 %
10		TypeScript	1.74 %	-0.0 %
11		Matlab	1.74 %	+0.0 %
12		Kotlin	1.71 %	+0.0 %
13		Go	1.19 %	-0.1 %
14		VBA	1.14 %	-0.0 %
15		Ruby	1.07 %	-0.0 %
16		Rust	0.98 %	-0.0 %
17	↑↑↑	Ada	0.75 %	+0.1 %



VERSO NUOVI LINGUAGGI – 2022

Top Programming Languages

Tiobe Index - December 2017

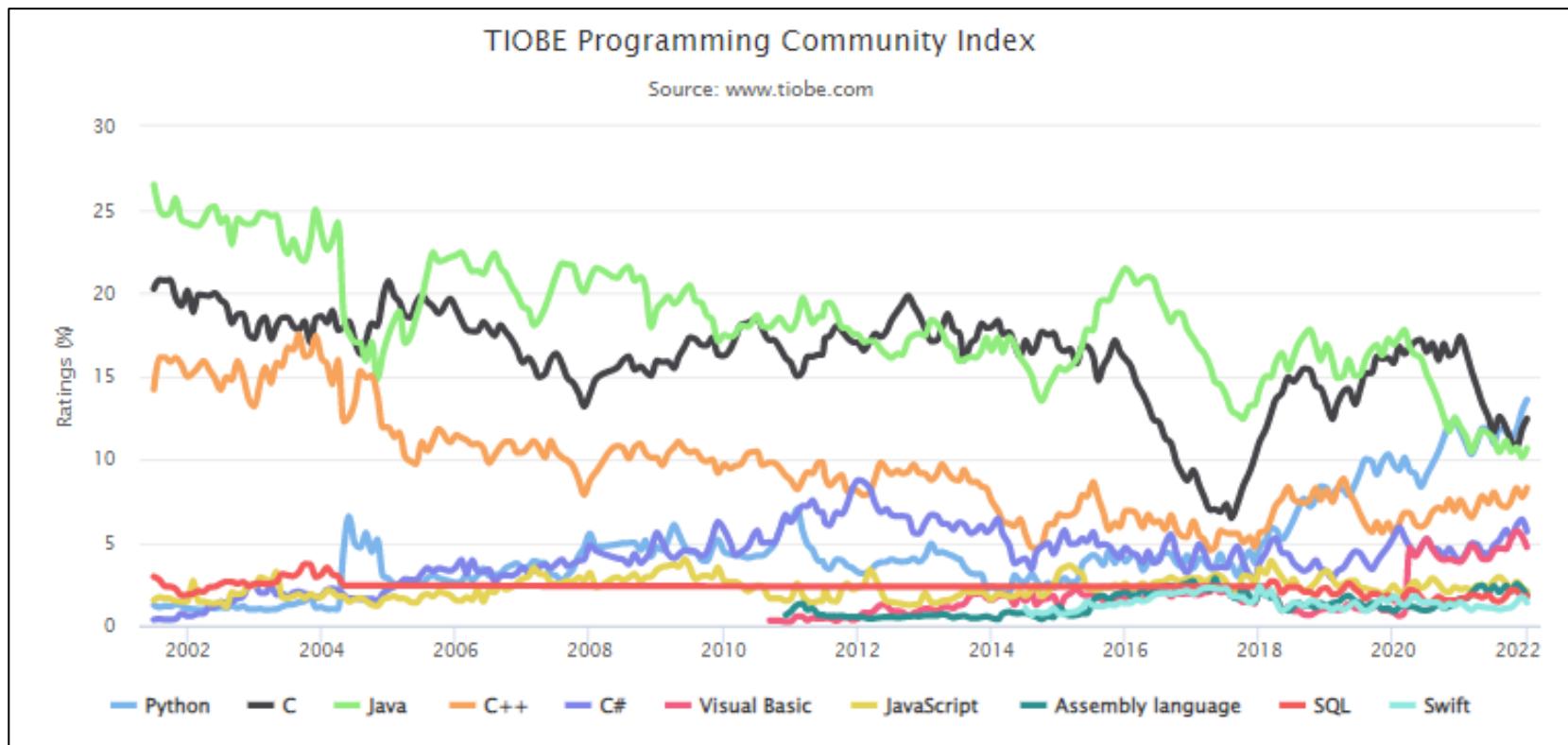


Jan 2021	Jan 2020	Change	Programming Language
1	2	▲	C
2	1	▼	Java
3	3		Python
4	4		C++
5	5		C#
6	6		Visual Basic
7	7		JavaScript
8	8		PHP
9	18	▲	R
10	23	▲	Groovy
11	15	▲	Assembly language
12	10	▼	SQL
13	9	▼	Swift
14	14		Go
15	11	▼	Ruby
16	20	▲	MATLAB
17	19	▲	Perl
18	13	▼	Objective-C

Jan 2022	Jan 2021	Change	Programming Language	Ratings
1	3	▲	Python	13.58%
2	1	▼	C	12.44%
3	2	▼	Java	10.66%
4	4		C++	8.29%
5	5		C#	5.68%
6	6		Visual Basic	4.74%
7	7		JavaScript	2.09%
8	11	▲	ASM Assembly language	1.85%
9	12	▲	SQL	1.80%
10	13	▲	Swift	1.41%
11	8	▼	PHP	1.40%
12	9	▼	R	1.25%
13	14	▲	Go	1.04%
14	19	▲	Delphi/Object Pascal	0.99%
15	20	▲	Classic Visual Basic	0.98%
16	16		MATLAB	0.96%
17	10	▼	Groovy	0.94%
18	15	▼	Ruby	0.88%



VERSO NUOVI LINGUAGGI





VERSO NUOVI LINGUAGGI

Language Rank	Types	Spectrum Ranking
1.	Python	100.0
2.	C	99.7
3.	Java	99.5
4.	C++	97.1
5.	C#	87.1
6.	R	87.7
7.	JavaScript	85.6
8.	PHP	81.2
9.	Go	75.1
10.	Swift	73.7

(dicembre 2019)

Language Rank	Types	Spectrum Ranking
1.	Python	100.0
2.	C++	99.7
3.	Java	97.5
4.	C	96.7
5.	C#	89.4
6.	PHP	84.9
7.	R	82.9
8.	JavaScript	82.6
9.	Go	76.4
10.	Assembly	74.1

(dicembre 2018)

Classifica IEEE Spectrum
(dicembre 2020)

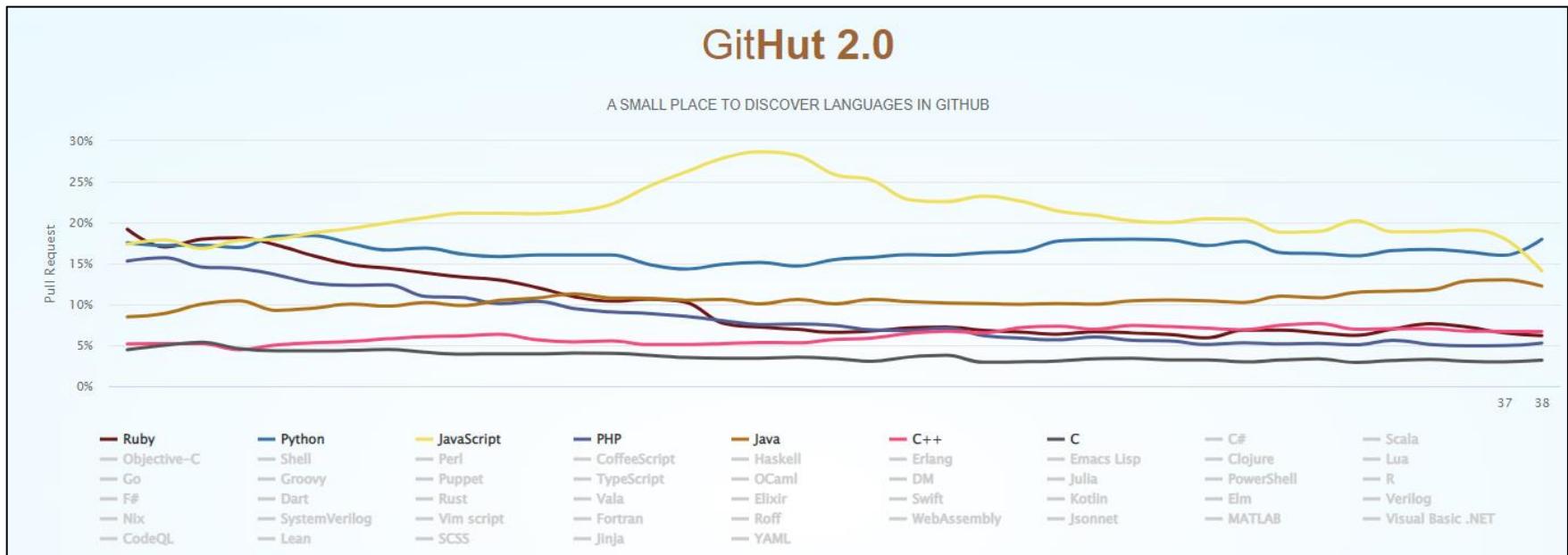
Rank	Language	Type	Score
1	Python▼	🌐💻⚙️	100.0
2	Java▼	🌐💻⚙️	95.3
3	C▼	💻⚙️	94.6
4	C++▼	💻⚙️	87.0
5	JavaScript▼	🌐	79.5
6	R▼	💻	78.6
7	Arduino▼	⚙️	73.2
8	Go▼	🌐💻	73.1
9	Swift▼	💻	70.5
10	Matlab▼	💻	68.4

Classifica IEEE Spectrum
(dicembre 2021)

Language Ranking: IEEE Spectrum			
Rank	Language	Type	Score
1	Python▼	🌐💻⚙️	100.0
2	Java▼	🌐💻⚙️	95.4
3	C▼	💻⚙️	94.7
4	C++▼	💻⚙️	92.4
5	JavaScript▼	🌐	88.1
6	C#▼	🌐💻⚙️	82.4
7	R▼	💻	81.7
8	Go▼	🌐💻	77.7
9	HTML▼	🌐	75.4
10	Swift▼	💻	70.4
11	Arduino▼	⚙️	68.4
12	Matlab▼	💻	68.3



VERSO NUOVI LINGUAGGI





NUOVI LINGUAGGI & NUOVE FIGURE

The 10 most in-demand tech jobs for 2021

Job	25th percentile	50th percentile	75th percentile	95th percentile
Security professional (information, systems, network, data, cloud)	\$124,250 (\$100,000 (information); \$105,500 (systems); \$112,500 (network); \$112,500 (data))	\$149,000 (\$120,500 (information); \$124,500 (systems); \$124,500 (network); \$134,000 (data))	\$179,000 (\$143,500 (information); \$148,250 (systems); \$148,250 (network); \$159,750 (data))	\$213,000 (\$168,500 (network); \$174,500 (data))
Cloud architect	\$121,000	\$146,000	\$174,500	\$202,250
Database administrator	\$79,750	\$100,750	\$120,500	\$160,000
Programmer analyst	\$88,750	\$109,500	\$131,000	\$154,500
Systems analyst	\$81,250	\$96,500	\$115,500	\$136,750
Mobile applications developer	\$112,750	\$135,750	\$161,750	\$189,750
Network administrator	\$78,500	\$95,250	\$113,500	\$133,500
Software developer	\$97,250	\$118,250	\$138,000	\$165,000
DevOps engineer	\$97,000	\$120,000	\$143,000	\$184,750
Help desk and desktop support professionals	\$35,250 (tier 1); \$41,500 (tier 2); \$50,750 (tier 3)	\$40,500 (tier 1); \$50,000 (tier 2); \$60,500 (tier 3)	\$47,500 (tier 1); \$56,500 (tier 2); \$72,250 (tier 3)	\$57,250 (tier 1); \$67,000 (tier 2); \$82,000 (tier 3)

Fonte: Cio.com

The 20 Best Tech Jobs for the Future

1. Artificial Intelligence/Machine Learning Engineer
2. Data Scientist
3. Information Security Analyst
4. Software Engineer/Software Developer
5. Computer Research Scientist
6. Data Analyst
7. IT Manager
8. Database Administrator
9. Web Developer
10. Computer Hardware Engineer
11. Computer Systems Analyst
12. DevOps Engineer
13. Computer Network Architect
14. Java Developer
15. Tech Sales Engineer
16. PHP Developer
17. Python Developer
18. Network and Systems Administrator
19. Mobile Application Developer
20. Web Designer



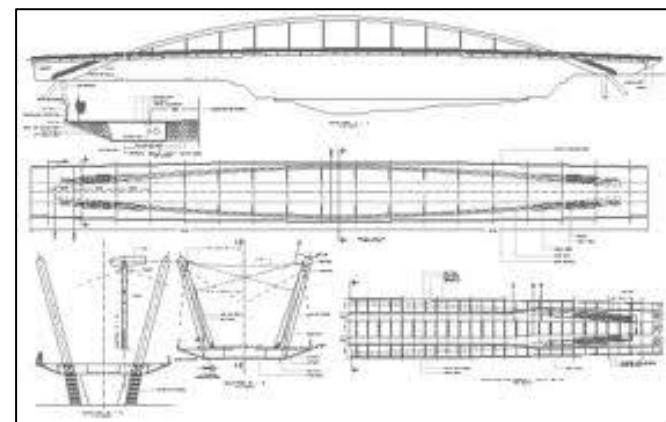
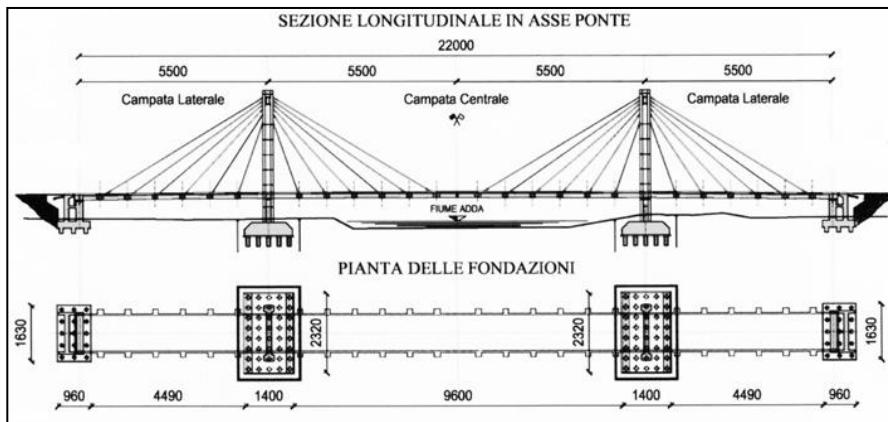
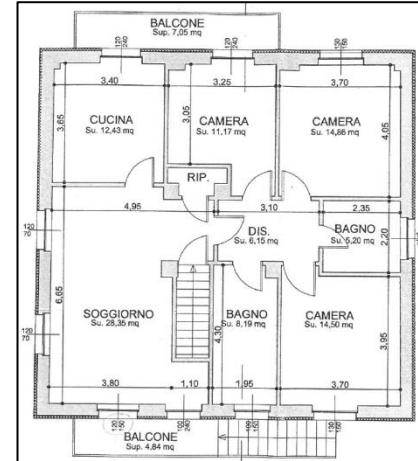
IL PROGETTO NON È IL CODICE

L'attività di progetto dev'essere riconosciuta come tale

- Il suo risultato – il **progetto** – dev'essere *un artefatto esistente e documentato in quanto tale*, come avviene da sempre in tutte le branche dell'ingegneria
 - il progetto di un sistema software non è il codice
 - esattamente come il *progetto di una casa* non è *la casa*, il *progetto di un ponte* non è *il ponte stesso!*
- Solitamente, nell'ingegneria il progetto è espresso tramite *disegni, tavole, diagrammi* che mirano a trasmettere
 - la struttura d'insieme
 - le parti coinvolte e le loro relazioni



IL PROGETTO NON È IL CODICE





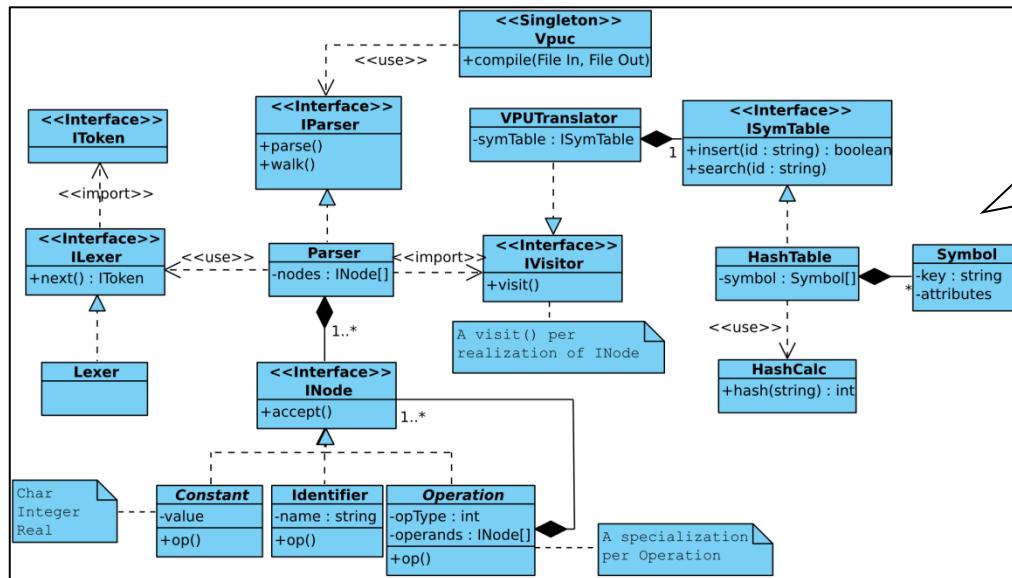
IL PROGETTO NON È IL CODICE

L'attività di progetto dev'essere riconosciuta come tale *anche nell'ingegneria informatica*

- L'attività di progetto dev'essere precedente e *totalmente disaccoppiata dall'implementazione*
- La relativa "giovinezza" dell'ingegneria informatica ha portato molti "artigiani del fai da te" e pensare erroneamente di poter far senza questi strumenti..
- ..ma non si gestisce la complessità senza poter operare a *un adeguato livello di astrazione – che non è il codice*
- *Servono metodologie, standard e linguaggi grafici per esprimere il progetto di un sistema software*

ESEMPIO: PROGETTO IN UML

- UML (Unified Modelling Language) è uno standard internazionale per esprimere modelli di sistemi (non necessariamente informatici)
- Fra i molti diagrammi che UML introduce, il *diagramma di struttura* esprime la *struttura di un sistema a oggetti*

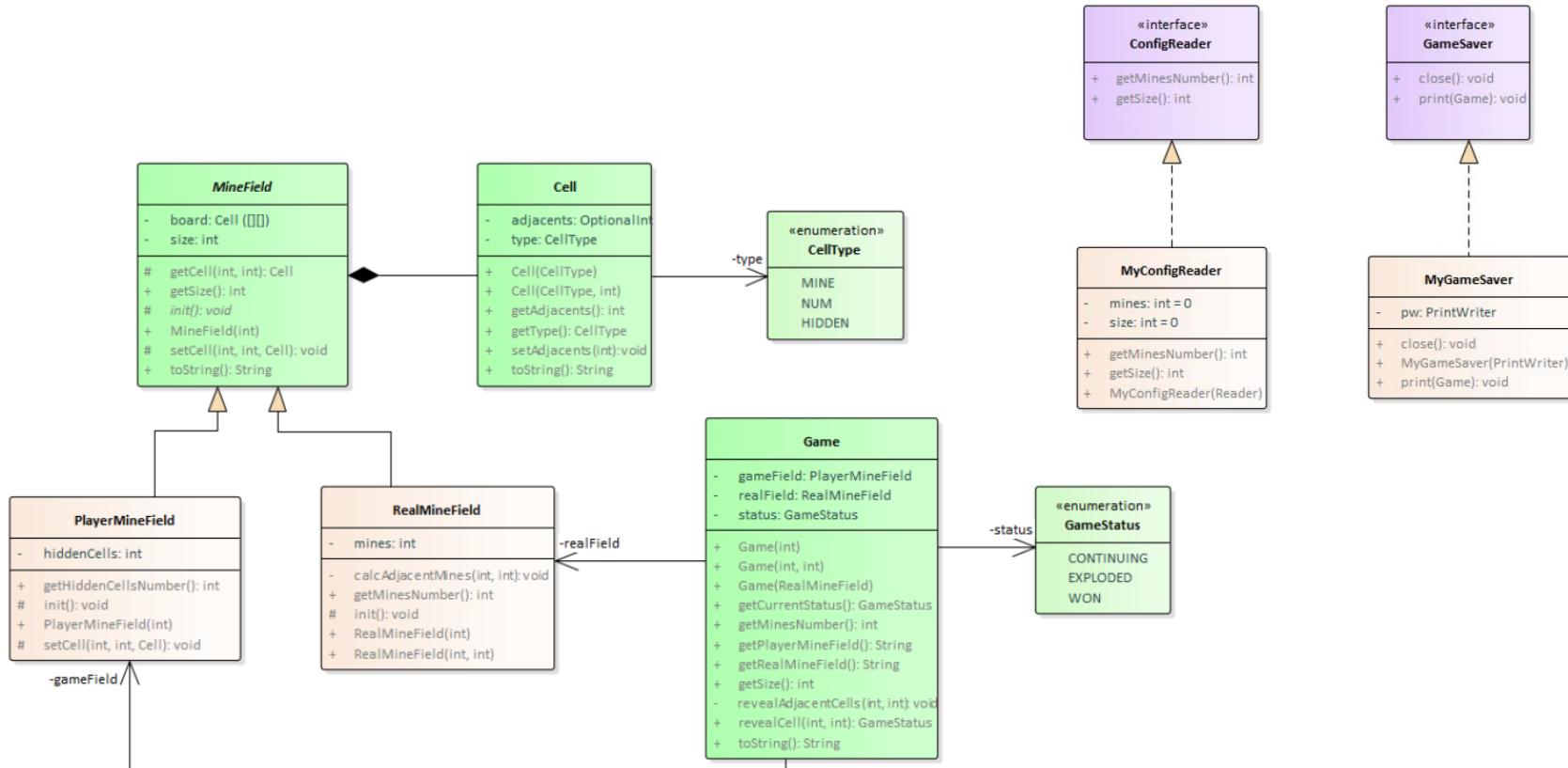


Esempio di progetto di un sistema di media complessità in UML



ESEMPIO: PROGETTO IN UML

- ESEMPIO: parte di un diagramma da un compito d'esame





SOLO NUOVI LINGUAGGI..?

MA un buon linguaggio non basta..

- Per avere successo occorre *non lasciarlo solo*
- Intorno a esso deve essere previsto un *ecosistema* pensato secondo una *precisa visione infrastrutturale*
 - l'esperienza (C++..) insegna che poter creare propri componenti e oggetti è importante, ma *standardizzare il linguaggio non basta*
- Non si può lasciare che ogni azienda (o strumento) realizzi "a proprio gusto" *tutto quello che ci sta intorno*
 - se ognuno realizza le proprie librerie (circa simili, ma non identiche) si creano *entità e comunità incomunicabili*
 - se ognuno copre solo alcune piattaforme, si finisce per *rifare le stesse cose N volte, con spreco di tempo e denaro*



DAI LINGUAGGI ALLE INFRASTRUTTURE

Occorre una *infrastruttura standard*

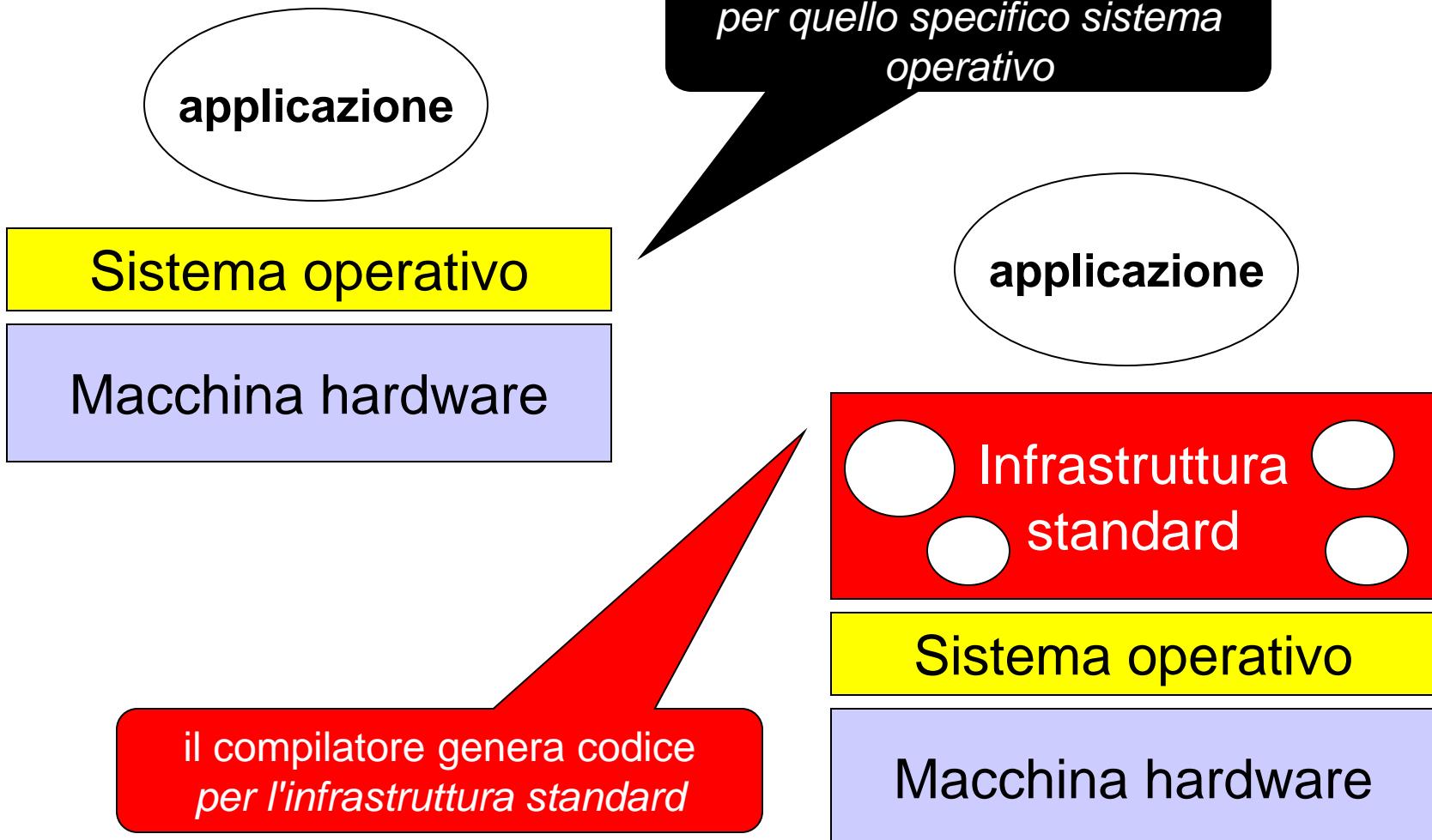
- un ambiente che offre *componenti pronti standardizzati, certamente presenti su ogni installazione*
- uno *strato di sicura affidabilità* su cui contare e costruire
- disponibile per *più sistemi operativi* e *diversi hardware*
- uno strato che *renda indipendente chi sta sopra dalla specifica configurazione hardware/software sottostante*

Se ci si riesce, *cambia totalmente lo scenario*

- cambia il processo di compilazione ed esecuzione
- *cambia il modo stesso di vedere "il software"*
- il sogno di ogni softwarista: *write once, run everywhere*



DAI LINGUAGGI ALLE INFRASTRUTTURE



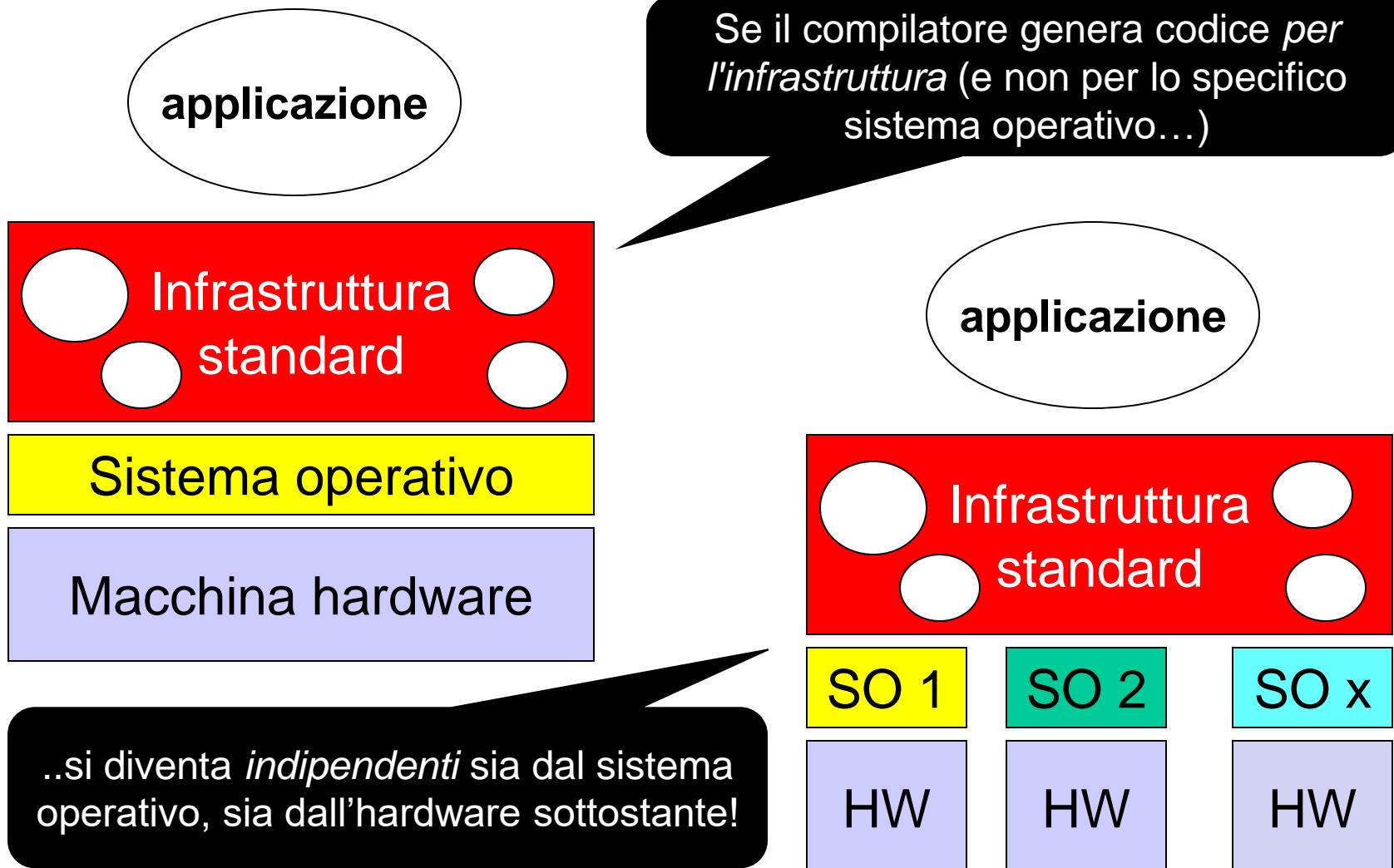


DAI LINGUAGGI ALLE INFRASTRUTTURE

- Poiché costruire software richiede tempo e risorse, si cerca di *massimizzare la resa* di tale investimento
- Perciò il software dovrebbe essere il più possibile *portabile e inter-piattaforma*
- Per questo interessano ***INFRASTRUTTURE PORTABILI*** e ***MULTI-PIATTAFORMA***
 - installabili su diversi sistemi hardware
 - con diversi sistemi operativi e configurazioni software
 - in grado di offrire alle applicazioni costruite sopra di esse un *ambiente noto, ricco e dal funzionamento uniforme*



INFRASTRUTTURE MULTI-PIATTAFORMA





COMPILAZIONE ED ESECUZIONE

applicazione

PRIMA il compilatore generava codice oggetto, lo *collegava (link) STATICAMENTE* alle librerie e produceva un *eseguibile AUTOCONTENUTO*

Sistema operativo

Macchina hardware

applicazione

ORA il compilatore genera codice *nel formato intermedio dell'infrastruttura* e non collega librerie staticamente:
si caricano e collegano DINAMICAMENTE, a run-time, i componenti che servono

Infrastruttura
standard

SO 1

SO 2

SO x

HW

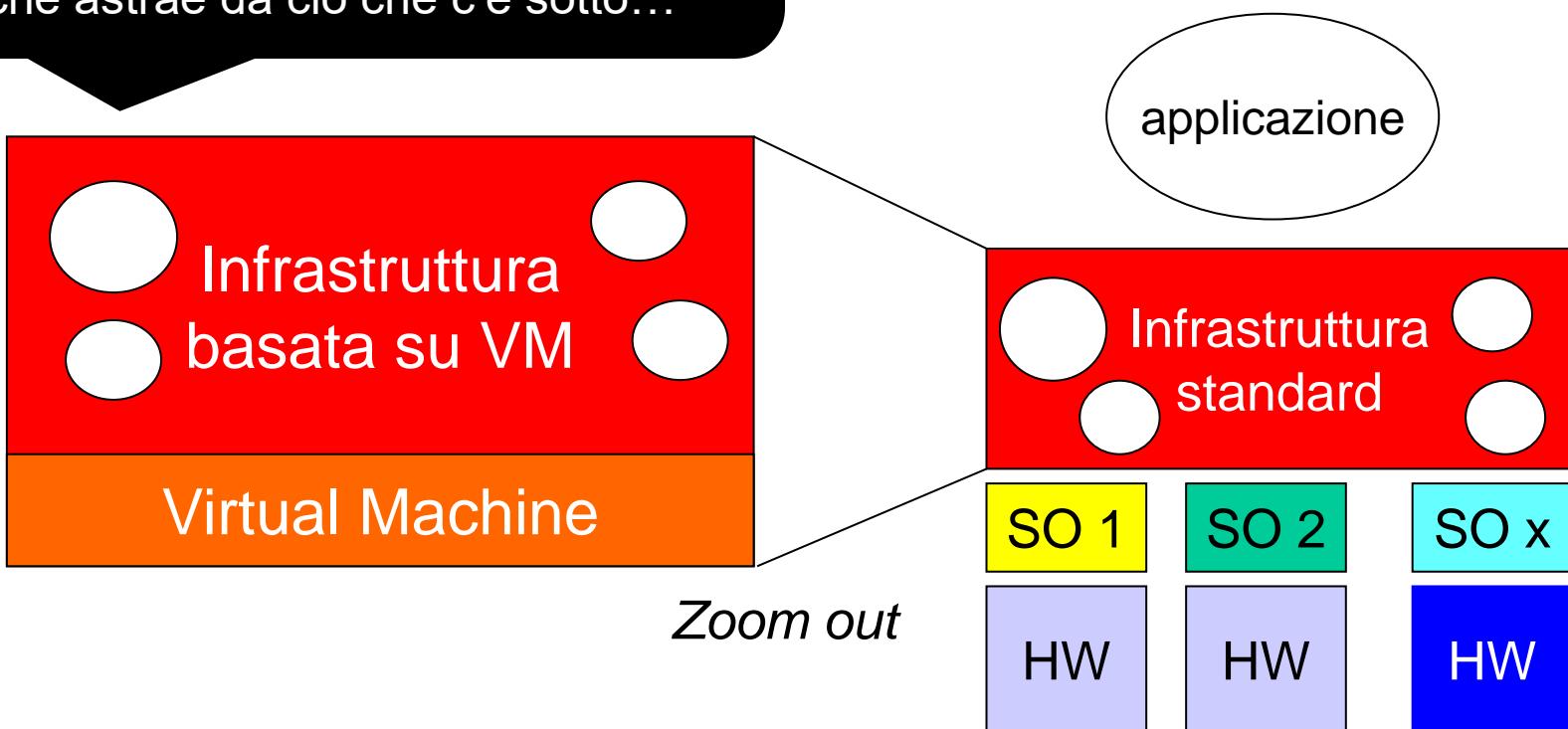
HW

HW



INFRASTRUTTURE MULTIPLE MULTI-PIATTAFORMA

Se poi l'infrastruttura a sua volta si appoggia su uno strato-base di MACCHINA VIRTUALE, che astrae da ciò che c'è sotto...



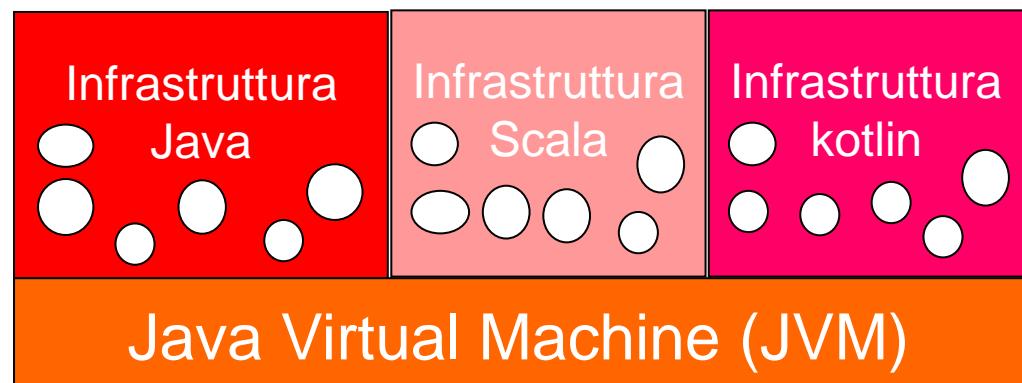


INFRASTRUTTURE MULTIPLE MULTI-PIATTAFORMA

Se poi l'infrastruttura a sua volta si appoggia su uno strato-base di MACCHINA VIRTUALE, che astrae da ciò che c'è sotto...



...si possono avere anche *DIVERSE INFRASTRUTTURE* sulla stessa VM, in grado di *interagire fra loro!*



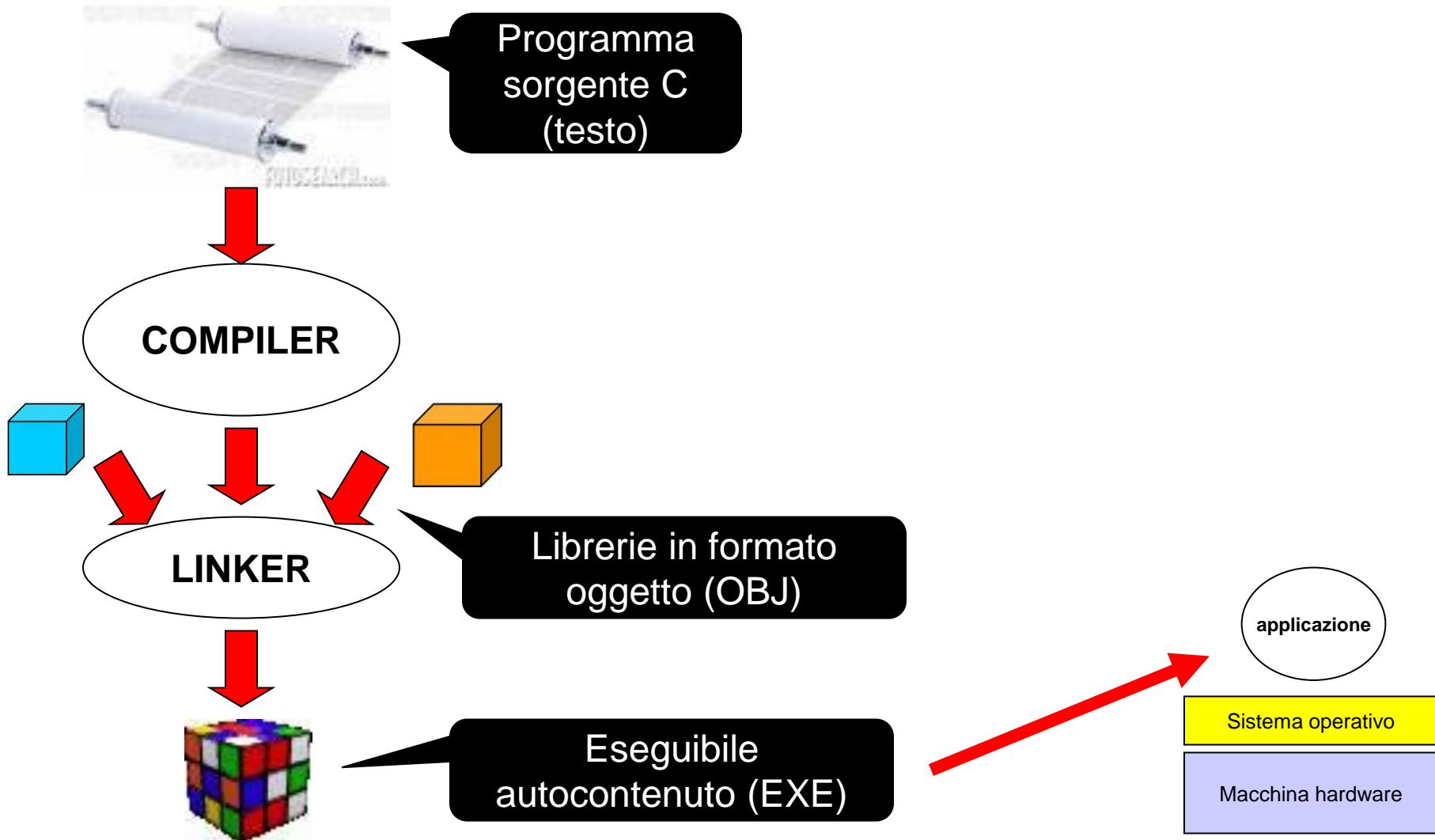


RIASSUMENDO

- Cambia il *processo di costruzione del software*
 - l'obiettivo non è più generare un programma per un certo processore e versione di sistema operativo..
 - ..ma generare qualcosa di *massimamente riusabile*
- Cambia il *processo di esecuzione del software*
 - non si esegue più un eseguibile (EXE) prodotto apposta per un certo processore e sistema operativo..
 - ..ma si fa eseguire *all'infrastruttura* un "prodotto" che non sarebbe eseguibile dal sistema operativo stand alone, poiché non è "autocontenuto": *un eseguibile portatile, inter-piattaforma*
- Si paga un (piccolo) prezzo per questo passo extra, ma i *vantaggi sono molto superiori al costo*
 - processo di sviluppo adatto a un *mondo distribuito e in divenire*

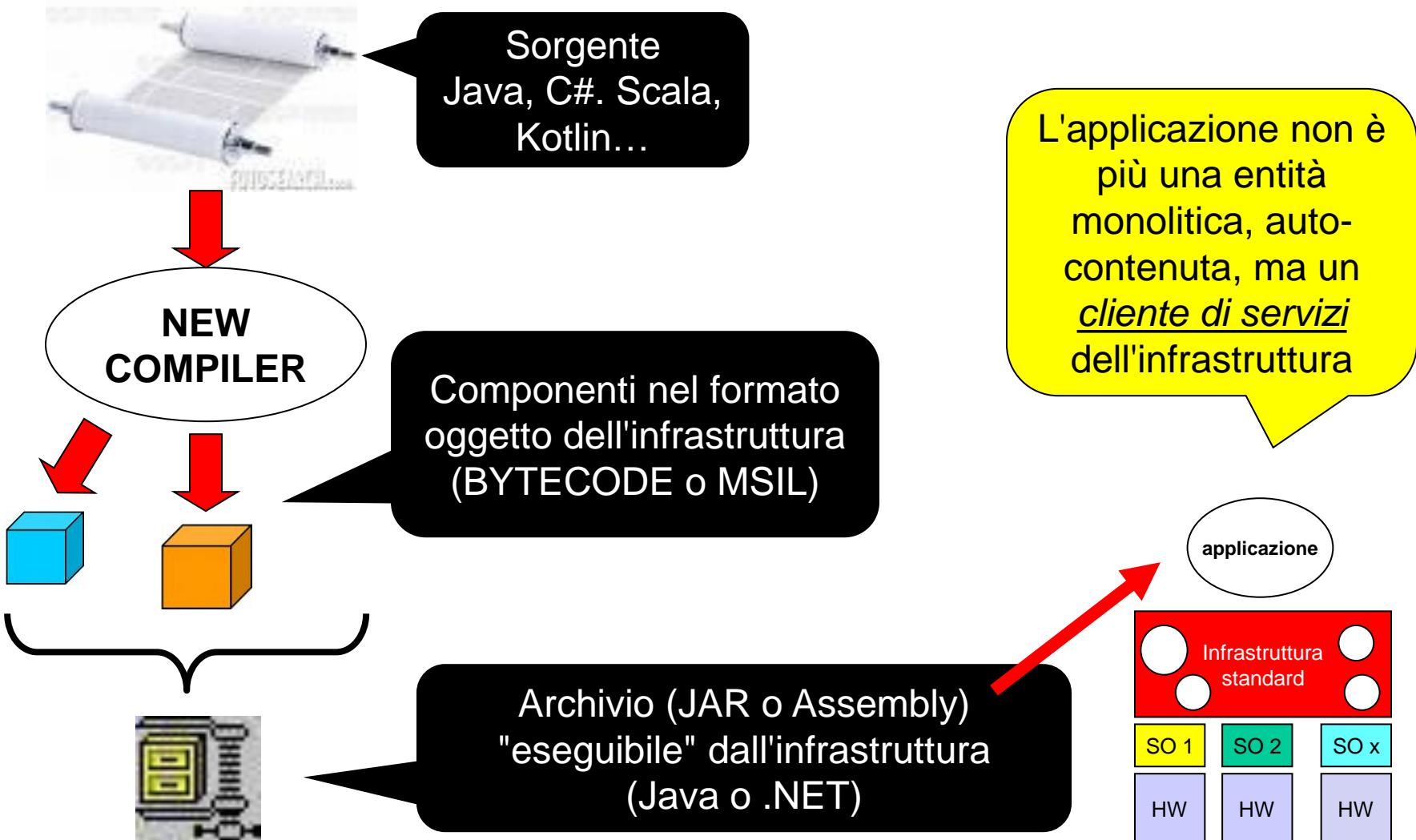


DAL "MONDO C" ...

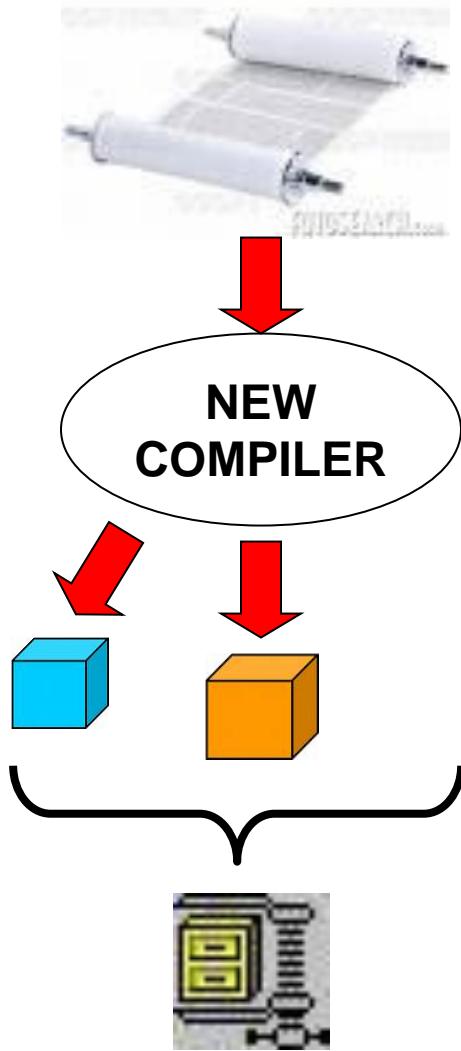




... AL "NUOVO MONDO"



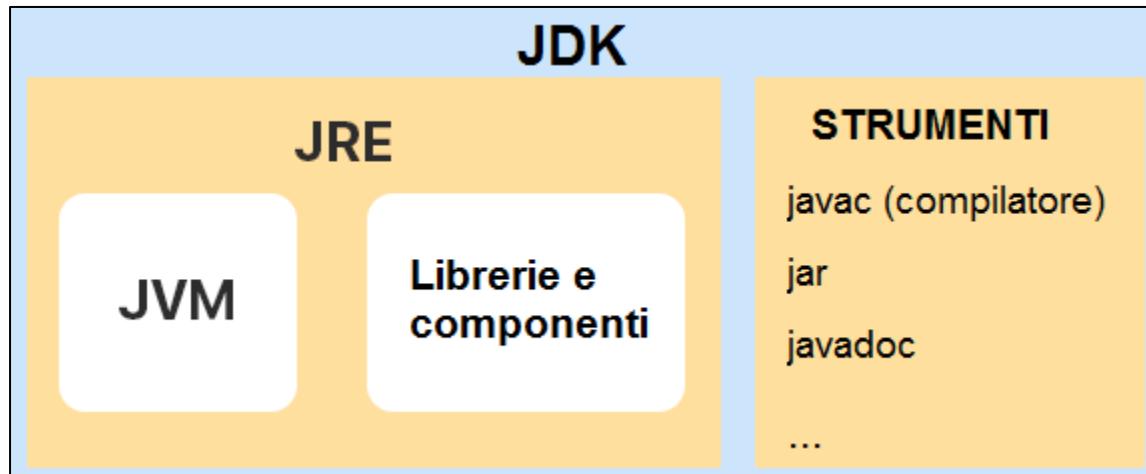
IL MONDO DI JAVA e C#



- Il sorgente è strutturato in una serie di componenti, i più importanti dei quali si chiamano **classi**
 - in Scala e kotlin, anche oggetti (*singleton*)
- Ogni componente viene compilato in un **formato portatile**, *indipendente da hw e sistema operativo*
- L'eseguibile può essere compilato su una sistema operativo **e funzionare su un altro**, purché la macchina sia *dotata della stessa infrastruttura*
- Per consentire ciò, cambia il senso di "eseguibile"
 - non è più un "eseguibile" in senso classico
 - un EXE di ".NET" o un JAR di Java non funziona su una macchina "nuda", col solo sistema operativo
- Occorre uno **STRATO-PONTE nell'infrastruttura**, che **interpreti** il formato portatile adattandolo alla specifica macchina
 - È l'unico strato *dipendente dalla piattaforma*.

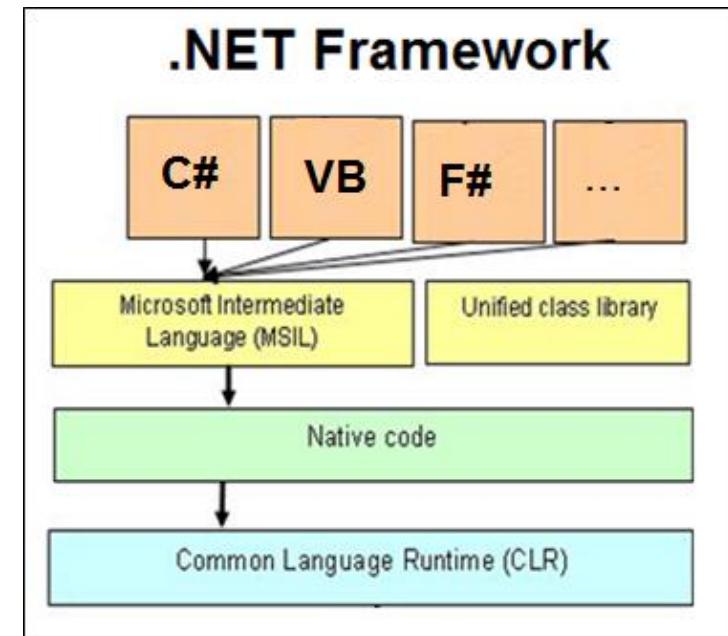
IL RUNTIME NELL'INFRASTRUTTURA

- Lo strato infrastrutturale di **Java** si chiama **JRE** (*Java Runtime Environment*)
 - Due livelli: Java Virtual Machine (JVM) + JRE vero e proprio
 - per **eseguire** applicazioni Java dev'essere installato sul computer
 - leggero, gratuito, multi-piattaforma (Windows, Mac, Linux, Android*)
 - per **sviluppare** applicazioni serve un set di strumenti più ampio, che include compilatore e altri strumenti: il **JDK** (*Java Development Kit*)



IL RUNTIME NELL'INFRASTRUTTURA

- Lo strato infrastrutturale di C# si chiama **.NET**
 - Articolazione simile: **Common Language Runtime (CLR)** + altro
 - **pre-installato** come **.NET framework** nei sistemi Windows (7, 10 ...) *incluso (fino alla versione 4.0) il compilatore a riga di comando (csc)*
 - Versione **.NET Core** multiplattaforma anche per MacOS e Linux (<https://dotnet.microsoft.com>)
 - Progetto open source, **MONO** (<https://www.mono-project.com/>)
- Novità da **.NET 5.0**
 - Multiplattaforma (è l'ex **.NET Core** rinominato)
 - **.NET 6.0**: rel. novembre 2021





LA PIATTAFORMA JAVA

Scaricamento e installazione (dettagli sul sito e in altre slide)

[jdk.java.net](#)

JDK 17.0.2 General-Availability Release

This page provides production-ready open-source builds of the Java Development Kit, version 17, an implementation of the Java SE 17 Platform under the GNU General Public License, version 2, with the Classpath Exception.

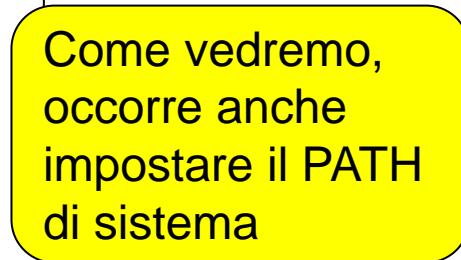
Commercial builds of JDK 17.0.2 from Oracle, under a non-open-source license, can be found at the Oracle Technology Network.

Documentation

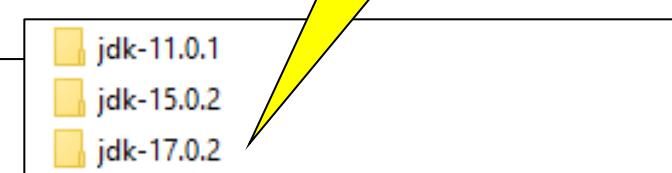
- Features
- Release notes
- API Javadoc

Builds

Linux/AArch64	tar.gz (sha256)	185776417 bytes
Linux/x64	tar.gz (sha256)	187033144
macOS/AArch64	tar.gz (sha256)	182209404
macOS/x64	tar.gz (sha256)	184480668
Windows/x64	zip (sha256)	186216309



Come vedremo,
occorre anche
impostare il PATH
di sistema

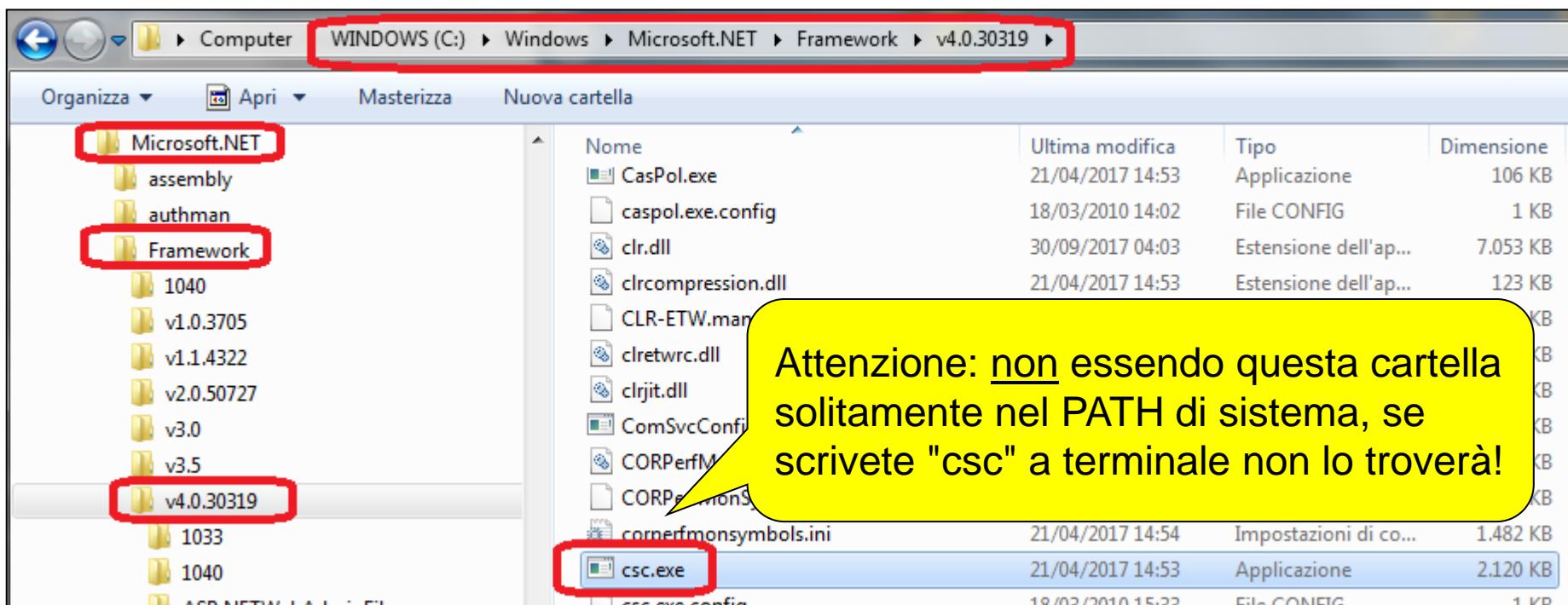


- jdk-11.0.1
- jdk-15.0.2
- jdk-17.0.2



LA PIATTAFORMA .NET

- Nel caso Windows, il compilatore è nella cartella del **.NET Framework** (una sottocartella di C:\Windows)
 - potreste averne anche più versioni: 3, 3.5, 4.0, ...
 - es. C:\Windows\Microsoft.NET\Framework\v4.0.30319 [e/o Framework64]





LA PIATTAFORMA .NET

- Dal 2019 il **.NET Framework** (solo per Windows) è in via di sostituzione col **.NET Core** (multiplattforma), rinominato semplicemente **.NET** (ora versione **6.0**)
 - allineato a C# 10.0 e Visual Studio 2022
 - sito: <https://dotnet.microsoft.com>

The screenshot shows the official .NET website homepage. The top banner features the .NET logo and the text "Free. Cross-platform. Open source. A developer platform for building all your apps." Below the banner are two buttons: "Get Started" and "Download". A note below the buttons states "Supported on Windows, Linux, and macOS". The main content area is divided into eight cards, each representing a different development category:

- Web**: Build web apps and services for Windows, Linux, macOS, and Docker.
- Mobile**: Use a single codebase to build native mobile apps for iOS, Android, and Windows.
- Desktop**: Create beautiful and compelling desktop apps for Windows and macOS.
- Microservices**: Create independently deployable microservices that run on Docker containers.
- Cloud**: Consume existing cloud services, or create and deploy your own.
- Machine Learning**: Add vision algorithms, speech processing, predictive models, and more to your apps.
- Game Development**: Develop 2D and 3D games for the most popular desktops, phones, and consoles.
- Internet of Things**: Make IoT apps, with native support for the Raspberry Pi and other single-board computers.



LA PIATTAFORMA .NET

- Da novembre 2021 è disponibile **.NET 6**
 - tipicamente non si trova nella cartella di Windows
 - si scarica da <https://dotnet.microsoft.com/download>

Download .NET

.NET is a free, cross-platform, open-source developer platform for building many different types of applications.

ⓘ Not sure what to download? [See recommended downloads for the latest version of .NET](#).

Supported versions

Version	Status	Latest release	Latest release date
.NET 6.0 (recommended)	LTS ⓘ	6.0.1	December 14, 2021
.NET 5.0	Current ⓘ	5.0.13	December 14, 2021
.NET Core 3.1	LTS ⓘ	3.1.22	December 14, 2021

Download and install

To start building .NET apps, download and install the .NET SDK (Software Development Kit).

[Download .NET SDK \(64-bit\)](#)

[32-bit download](#) | [Arm64 download](#)

⚠️ You need to install additional dependencies on older versions of Windows.

Check everything installed correctly

Once you've installed, open a **new** command prompt and run the following command:

```
Command prompt
> dotnet
```

If the installation succeeded, you should see an output similar to the following:

```
Command prompt
Usage: dotnet [options]
Usage: dotnet [path-to-application]

Options:
-h|--help      Display help.
--info         Display .NET information.
--list-sdks    Display the installed SDKs.
--list-runtimes Display the installed runtimes.
```



LA PIATTAFORMA .NET

- Per usarlo basta aggiungere la cartella al path corrente:

```
C:\Windows\System32\cmd.exe
C>csc
"csc" non è riconosciuto come comando interno o esterno,
un programma eseguibile o un file batch.

C>path=%PATH%;C:\Windows\Microsoft.NET\Framework\v4.0.30319
C>csc
Microsoft (R) Visual C# Compiler version 4.7.2053.0
for C# 5
Copyright (C) Microsoft Corporation. All rights reserved.

This compiler is provided as part of the Microsoft (R) .NET Framework, but only
supports language versions up to C# 5, which is no longer the latest version. For
compilers that support newer versions of the C# programming language, see http
://go.microsoft.com/fwlink/?LinkId=533240

warning CS2008: Nessun file di origine specificato
error CS1562: Per gli output senza origine occorre specificare l'opzione /out.

C>_
```

Ora lo trova ☺



LA TIPICA OBIEZIONE (1/2)

MA.. così non ho un “vero eseguibile”!

- Occhio alle parole: cosa vuol dire “eseguibile”?
- Ogni file “eseguibile” lo è solo *nel contesto di una certa infrastruttura*, che gli fornisce il necessario contorno
 - un “eseguibile” Windows non funziona in MacOS
 - un “eseguibile” .NET non funziona in Windows senza .NET
 - un file Word non si apre senza Microsoft Word installato..
- Una entità è “eseguibile” (apribile) solo se è presente lo strato software che la interpreta, *non perché il suo nome finisce per “EXE”*
 - un JAR di Java è dunque “eseguibile” quanto un EXE di C#: ciò che conta è che sia presente l’infrastruttura di supporto



LA TIPICA OBIEZIONE (2/2)

Sarà anche vero, ma... *gli eseguibili “veri” si aprono facendoci sopra “doppio clic”!*

E INFATTI...

- con la giusta configurazione del JRE (vedere slide!) si rendono eseguibili con doppio clic le applicazioni Java (Scala, Kotlin) distribuite come archivio JAR
 - Android: archivio APK
- l'installazione del .NET framework rende eseguibili le applicazioni .NET distribuite come archivio EXE
 - Occhio al nome: gli "EXE di .NET non sono i classici eseguibili di Windows... tanto è vero che su Windows "puro" non girano!
 - con gli EXE "classici" hanno in comune solo la desinenza.

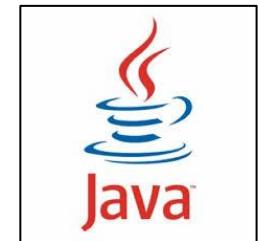


JAVA e C#: L'INFRASTRUTTURA

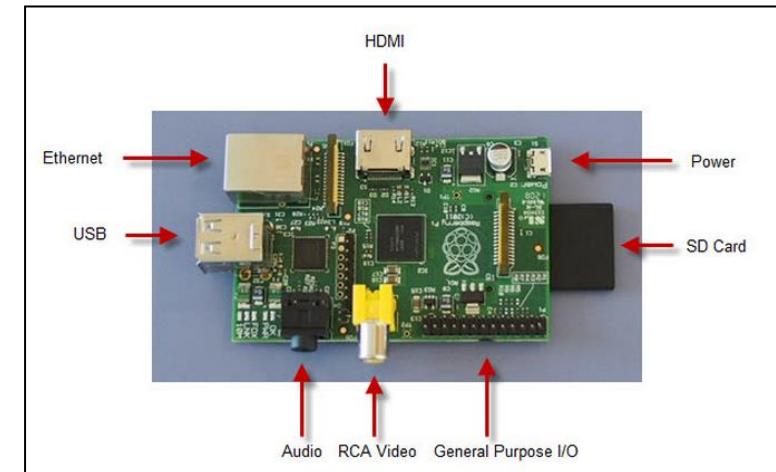
- A differenza di quanto accadeva in C (e C++), Java e C# nascono dotati da una *infrastruttura standard portatile*
 - indipendenza dalla piattaforma hardware / software
 - write once, run everywhere
- L'infrastruttura offre servizi per *coprire praticamente ogni esigenza del moderno sviluppo di software*
 - supporto per la grafica (JavaFX, Swing, Java2D, 3D..)
 - programmazione a eventi
 - supporto di rete: URL, Socket, connessioni...
 - supporto per il multi-threading e la concorrenza
 - interfacciamento con database (JDBC)
 - supporto per la sicurezza (cifratura, autenticazione..)
 - esecuzione remota (RMI)...

PORTABILITÀ: JAVA

- L' **infrastruttura Java** è disponibile per praticamente *tutte le piattaforme hardware/software*
- Spesso *embedded* in lettori DVD/BlueRay, TV,...
 - *guarda dietro se trovi il logo!*

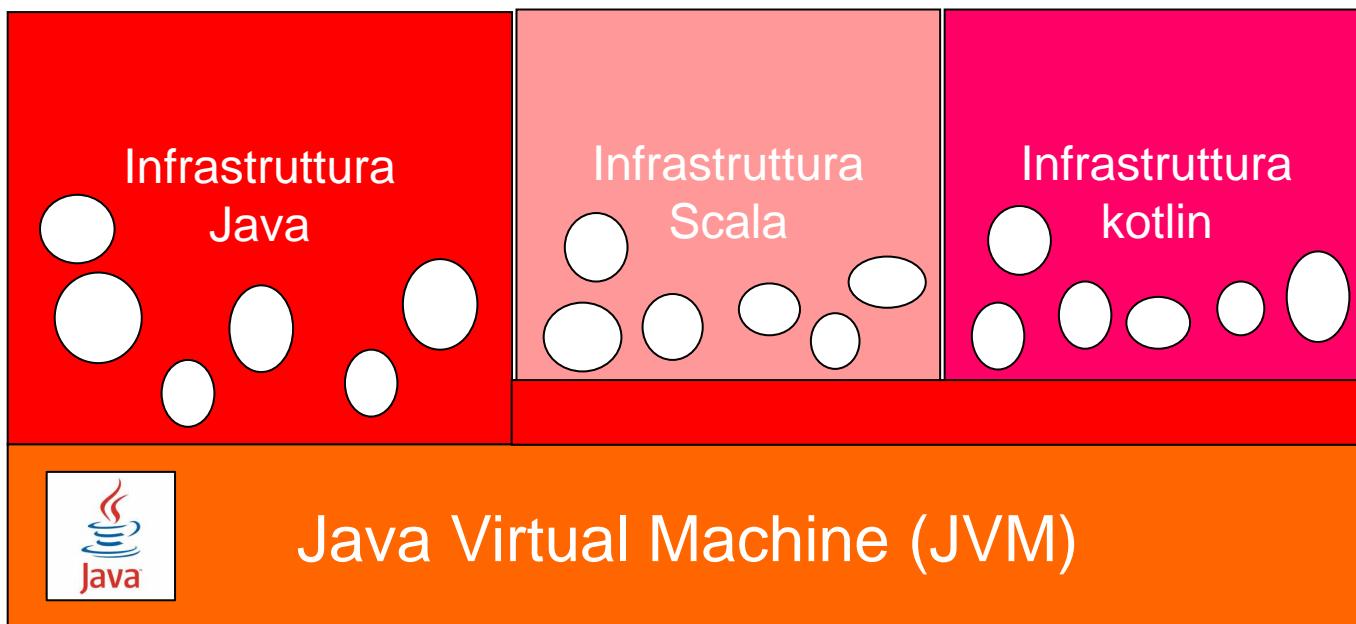


- Disponibile anche su micro-computer low-cost e all-in-one come i Raspberry Pi



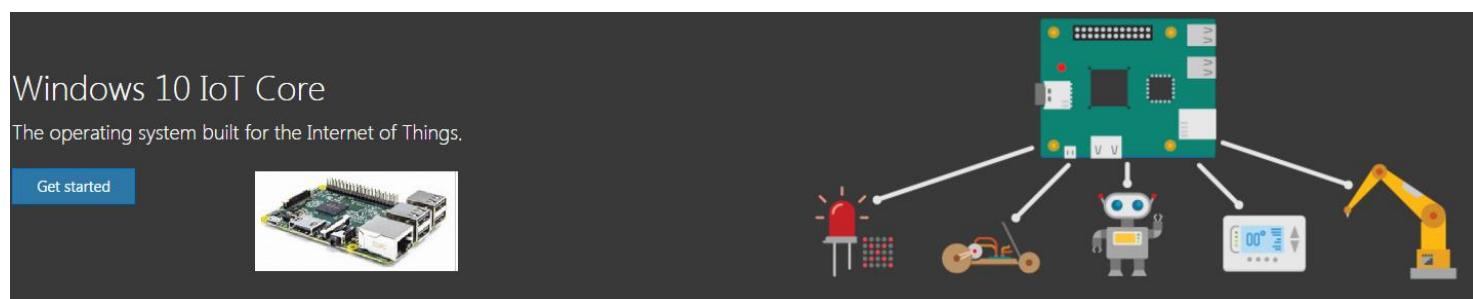
PORTABILITÀ: JAVA

- L' **infrastruttura Java** funge anche da base per molte altre infrastrutture e linguaggi, come *Scala* e *Kotlin*
 - di conseguenza, componenti Java, Scala, Kotlin sono interoperabili



PORTABILITÀ: .NET

- L' **infrastruttura .NET** è disponibile su più piattaforme
 - nativamente, sui sistemi Windows
 - tramite installazione separata per sistemi Linux/Mac
- Molti linguaggi supportati (ma non Java)
- Da Windows10, concetto di *Universal App*
 - unica app per desktop e mobile
- *Windows 10 IoT Core* disponibile per diverse schede (inclusi Raspberry PI)





JAVA e C#: IL SUPPORTO

- Java e C# sono dotati da un *compilatore ottimizzato* che genera codice *per l'infrastruttura*
- A differenza di C e C++, però, qui **il compilatore è solo uno degli strumenti**: il kit include anche strumenti per:
 - produrre *adeguata documentazione* sempre aggiornata
→ *estrazione automatizzata di MANUALI HTML*
 - supportare attività di *collaudo sistematico*
→ *esecuzione automatizzata di SUITE DI TEST*
 - progettare *la distribuzione dell'applicazione* (o *libreria*), eventualmente anche con firma digitale (per garantirne autenticità)
→ *impatto sul processo produttivo*
→ *impatto sull'organizzazione dell'applicazione*



IL PROBLEMA DEL DEPLOYMENT

Deployment = distribuzione di un software

- Non può essere lasciata al "fai da te":
anche questa attività va organizzata e progettata
 - è intrinsecamente non scalabile: 10 installazioni sono una cosa, 1000 o 10000 sono tutt'altra faccenda!
- **Cosa distribuire?**
 - supportare N piattaforme hardware & software *non può significare dover distribuire N eseguibili diversi...* è ingestibile!
 - ideale: distribuire un solo file, per tutte le possibili piattaforme
- **Un archivio di componenti compattati, simile a uno ZIP**
 - magari usabile senza bisogno neanche di scompattarlo...😊