



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Triennale in Informatica

TESI DI LAUREA

Un'Analisi Esplorativa sull'Evoluzione della Community Stack Overflow

RELATORE

Prof. Fabio Palomba

SECONDO RELATORE

Dott. Giammaria Giordano

Università degli studi di Salerno

CANDIDATO

Giorgio Iannello

Matricola: 0512107789

Anno Accademico 2021-2022

Credere in se stessi è uno dei mattoni più importanti nella costruzione di ogni impresa di successo.

Lydia Maria Child

Sommario

Conoscere come il mondo dell'ICT si evolve nel tempo è sicuramente interessante. Capire cosa la community di programmatori ha preferito negli anni, cosa ha dato loro maggiore difficoltà, quali linguaggi sono più amati, quando iniziano ad informarsi seriamente, sono tutti dati hanno permesso di capire quali cambiamenti sono avvenuti.

Mentre buona parte della ricerca si è focalizzata in questo ambito a livello universale, poco è stato fatto per valutare le differenze abitudinali dei programmatori in diverse regioni del mondo o sulla qualità della loro vita.

La tesi in esame propone dunque uno studio empirico sull'evoluzione di Stack Overflow ossia sull'evoluzione del mondo ICT basandosi sui sondaggi fatti dalla piattaforma e messi a disposizione sul sito.

In primo luogo, vengono osservati i dati poi analizzati, incrociati, omogenizzati e infine raggruppati. Dopo una sintesi chiara delle informazioni tramite tabelle, dei commenti porteranno alla luce ciò che maggiormente è cambiato negli anni. Si vedrà che l'utenza è stata molto varia. In alcuni ambiti è rimasta tendenzialmente la stessa ma in altri è stata capace di far notare l'evoluzione. Vedremo infatti come le differenze di genere (a livello di popolarità), da sempre esistite nel mondo dell'informatica, siano rimaste, vedremo come i giovani abbiano voglia di informarsi, di crescere e di sapere, vedremo che alcuni linguaggi un po' più moderni come Python, siano riusciti a diventare più popolari e usati di altri meno nuovi e già da tempo affermati. Inoltre vedremo che tecnologia dà maggiore difficoltà alla community e quale meno, quali sono le occupazioni più frequenti tra i lavoratori e anche altro.

Indice	ii
Elenco delle figure	iv
Elenco delle tabelle	v
1 Introduzione	1
1.1 Contesto applicativo	1
1.1.1 Lo sviluppo del software	1
1.1.2 Sistema pervasivo	2
1.1.3 Malfunzionamenti del software	3
1.1.4 Qualità del codice sorgente	3
1.1.5 Stack Overflow. Cos'è?	4
1.2 Motivazioni e obiettivi	4
1.3 Risultati	5
1.4 Struttura della tesi	6
2 Stato dell'arte	7
2.1 Background	7
2.1.1 Linguaggi di programmazione	7
2.1.2 Linguaggi di programmazione più diffusi	8
2.1.3 Storia della programmazione	8
2.1.4 L'evoluzione della programmazione	9

2.1.5	Età dell'utenza	10
2.1.6	Genere	12
2.2	Evoluzione dei linguaggi di programmazione	13
2.3	Mining di Stack Overflow	14
2.4	Limitazioni dello stato dell'arte	16
3	Metodologia	20
4	Risultati	24
5	Conclusioni	29
5.1	Risultati	29
5.2	Sviluppi futuri	31
	Ringraziamenti	32

Elenco delle figure

Elenco delle tabelle

3.1	Categorie considerate nel contesto dello studio	22
4.1	Andamento dell'età degli utenti di Stack Overflow	24
4.2	Differenze di popolarità tra uomini e donne presenti su Stack Overflow	25
4.3	Tecnologie più usate negli anni dagli utenti di Stack Overflow	25
4.4	Linguaggi che sono stati più ricercati negli anni su Stack Overflow	26
4.5	Sistemi operativi preferiti dagli utenti di Stack Overflow	27
4.6	Occupazione sul lavoro degli utenti di Stack Overflow	27
4.7	Situazione smart working prima dell'emergenza Covid-19	28

1.1 Contesto applicativo

1.1.1 Lo sviluppo del software

Sviluppo del software si riferisce a una serie di attività relative all'informatica dedicate al processo di creazione, progettazione, implementazione e supporto del software.

Il software stesso rappresenta la serie di istruzioni o programmi che indicano a un computer cosa fare. È indipendente dall'hardware e rende i computer programmabili. Esistono tre tipi di base di software:

- **Software di sistema** per fornire le principali funzioni come sistemi operativi, gestione dischi, programmi di utilità, gestione hardware e altre esigenze operative.
- **Software di programmazione** per fornire ai programmatori strumenti quali editor di testo, compilatori, linker, debugger e altri strumenti per la creazione del codice.
- **Software applicativo** (applicazioni o app) per aiutare gli utenti a eseguire attività. Le suite di produttività per uffici, il software di gestione dei dati, i lettori multimediali e i programmi di sicurezza rappresentano degli esempi.

Lo sviluppo del software viene eseguito principalmente da programmatori, ingegneri del software e sviluppatori del software. Questi ruoli interagiscono e si sovrappongono e le dinamiche tra loro variano notevolmente tra i reparti e le comunità di sviluppo.

I programmatori scrivono il codice sorgente per programmare i computer per attività specifiche come l'unione di database l'esecuzione di ricerche o la visualizzazione di testo e grafici. Gli ingegneri del software applicano principi di ingegneria per creare software e sistemi e risolvere i problemi. Usano il linguaggio di modellazione e altri strumenti per individuare soluzioni che spesso possono essere applicate ai problemi in modo generale, invece di limitarsi a risolvere un'istanza o un client specifico. La loro responsabilità è aumentata man mano che i prodotti sono diventati sempre più intelligenti con l'aggiunta di microprocessori e sensori. Gli sviluppatori di software hanno un ruolo meno formale degli ingegneri e possono essere strettamente coinvolti in aree di uno specifico progetto, inclusa la scrittura del codice. Allo stesso tempo, eseguono l'intero ciclo di vita dello sviluppo del software, compreso il lavoro con i team funzionali per trasformare i requisiti in caratteristiche, gestendo team e processi di sviluppo e conducendo i test e la manutenzione del software.

È importante distinguere tra sviluppo del software personalizzato e sviluppo del software commerciale. Il primo è il processo di progettazione, creazione, implementazione e manutenzione del software per un insieme specifico di utenti, funzioni o organizzazioni. Al contrario, il secondo è progettato per un ampio insieme di requisiti, il che consente di impacchettarlo e commercializzarlo e distribuirlo.

Lo sviluppo del software è importante perché è pervasivo.

1.1.2 Sistema pervasivo

Una prerogativa di un sistema pervasivo è che sia in grado di ottenere informazioni sull'utente e sull'ambiente in cui viene utilizzato, per esempio, la posizione o l'identità dei singoli utenti. Un aspetto fondamentale del sistema pervasivo, quindi, è quello di collezionare dati grezzi, trasformarli in informazioni di contesto e condividere le informazioni tra le diverse applicazioni in esecuzione su i vari dispositivi cercando comunque di garantire la sicurezza delle informazioni, inibendo accessi non autorizzati e rispettando la privacy individuale. In base a queste informazioni, le applicazioni possono adattare il loro comportamento in maniera diversa a seconda dei casi.

Un sistema pervasivo, quindi, dovrebbe essere capace di cambiare dinamicamente il proprio comportamento in base alle informazioni collezionate o sulle intenzioni dell'utente. Questi modelli, quindi, sono basati sulla capacità di comprendere quello che le persone stanno facendo e quello che desiderano fare. Per esempio, i modelli che tengono conto dell'attività dell'utente del passato rappresentano un importante strumento per un sistema pervasivo perché consentono di determinare, in accordo con i precedenti comportamenti, le azioni

ottimali che l'utente stesso deve eseguire in determinate situazioni.

Un sistema pervasivo generalmente gestisce grosse quantità di informazioni, molte delle quali acquisite tramite sensori e riguardanti gli utenti. Dal punto di vista dell'utente, è desiderabile che vengano rispettati i principi di privacy e che sia garantita la sicurezza di queste informazioni.

1.1.3 Malfunzionamenti del software

Il malfunzionamento del software è sostanzialmente quando lo stesso non si comporta come l'utente si aspetta.

In altre parole, l'essere umano può commettere un errore nel codice del programma che può provocare un difetto (fault, bug). Se il codice difettoso viene eseguito, il sistema può fallire nel fare quello che deve fare (o può fare qualcosa che non deve), causando un esito negativo chiamato 'failure'. I difetti nascono per vari motivi:

- Il cliente o la persona che fornisce i requisiti per conto dell'organizzazione del cliente potrebbe non sapere esattamente cosa è richiesto o potrebbe dimenticare di fornire alcuni dettagli, il che potrebbe portare a funzionalità mancanti.
- La persona che sta raccogliendo i requisiti può interpretare erroneamente o perdere completamente un requisito quando li documenta.
- problemi di progettazione.
- errori dei programmatori causati da mancanza di tempo o inesperienza.

Consegnare un prodotto mal funzionante può portare a problemi come rilavorazioni, guasti costosi o addirittura perdita del cliente che può preferire acquistare un prodotto concorrente rispetto a un prodotto di scarsa qualità.

Per questo motivo sorge la necessità da parte degli sviluppatori di curare la qualità del codice sorgente.

1.1.4 Qualità del codice sorgente

Uno degli strumenti più importanti per garantire la qualità del codice è sicuramente il testing il cui scopo è quello di rilevare i difetti tramite i malfunzionamenti per poterli correggere prima che venga utilizzato dall'utente finale. In pratica, il test del software è una parte importante dello sviluppo del software. Se il test del software non viene eseguito

correttamente, le applicazioni possono contenere errori.

Altro strumento fondamentale per la qualità del codice è sicuramente l'implementazione di meccanismi di riuso come i design pattern ossia "soluzioni progettuali generali a problemi ricorrenti in un contesto specifico". Utili anche a riconoscere anticipatamente un problema e quindi a fare meno fatica e risolverlo più velocemente oltre che rendere un progetto più chiaro e comprensibile per la manutenzione.

Importante sarebbe anche la tecnica del code review ossia la tecnica che prevede di far revisionare il codice da uno o più persone dove almeno una di queste non è l'autore del codice stesso.

Negli ultimi anni è diventato sempre più popolare il concetto di crowd-development, ovvero il meccanismo tramite il quale uno sviluppatore può fare domande ed ottenere risposte e/o commenti rispetto all'implementazione di nuove feature o alla risoluzione di difetti/vulnerabilità. Uno dei più popolari strumenti in questo senso è Stack Overflow.

1.1.5 Stack Overflow. Cos'è?

[1] Stack Overflow è un sito web di proprietà di Stack Exchange creato da Jeff Atwood e Joel Spolsky nel 2008, che funge da "forum" sulla programmazione. Infatti, nel suddetto sito, è possibile porre e rispondere a domande inerenti al mondo dell'informatica ed è possibile, da parte degli utenti iscritti, modificare le stesse in pieno stile Wikipedia. Inoltre è possibile votare le risposte considerate "migliori" dando punti, detti "reputazione", all'autore facendo salire di visibilità di quella determinata risposta per aiutare la community a trovare più velocemente la soluzione al problema in esame. Per lo stesso motivo si può votare contro una determinata risposta perchè considerata non pertinente e farle perdere visibilità.

1.2 Motivazioni e obiettivi

L'obiettivo di questa tesi è quello di osservare l'evoluzione della community di programmatori e dei linguaggi di programmazione. Capire quanti giovani si sono approcciati a questo mondo in passato e quanti lo stanno facendo oggi, le differenze di genere presenti nell'ICT sono sicuramente dati interessanti per conoscere come si sta muovendo la società in questo ambito. Inoltre sapere i programmatori, negli anni, quale linguaggio hanno usato maggiormente e quale hanno ricercato di più su internet farà capire in primis l'evoluzione delle tecnologie ossia farà capire se ci sono stati linguaggi che sono stati accantonati in favore di altri e se nuove risorse sono riuscite a pareggiare o magari superare quelle già da tempo

affermate. In secondo luogo, saremo in grado di conoscere cosa da maggiore e minore difficoltà ai programmatori quando sono a lavoro sui loro progetti.

Banalmente alcune di queste informazioni sono reperibili sul web. Una semplice ricerca potrà sicuramente dirci quale è il linguaggio di programmazione più usato, o quanti uomini e donne sono presenti in campo informatico e così via ma lo scopo di questo studio è quello di riportare anno per anno le variazioni delle percentuali delle varie categorie così da avere una panoramica ben più ampia e precisa dell'evoluzione e osservare come si è arrivato a un determinato risultato, tramite la crescita di uno o la decrescita dell'altro?

Per fare ciò si è pensato, quale modo migliore per ottenere statistiche annuali se non chiedendo proprio alla community stessa? Per questo motivo si è pensato di osservare i questionari che ogni anno Stack Overflow rilascia ai suoi utenti e i cui risultati vengono pubblicati sul sito. Nel corso degli anni, sono stati sviluppati pochi studi sui survey di Stack Overflow e quando è stato fatto ci si è focalizzato solo su un aspetto o su un settore. C'è chi si è occupato di analizzare i numeri riguardanti il genere dell'utenza, altri si sono occupati di analizzare le tecnologie più richieste sul forum e quindi di capire il mondo dei programmatori in quale ambito lavora di più o è più propenso ad imparare. Inoltre, ci sono stati studi che hanno tentato di analizzare anche i differenti comportamenti degli anziani e dei giovani in ambito ICT e infine c'è chi si è chiesto quali sono stati i problemi più affrontati in ambito mobile.

Osservando gli studi presenti in letteratura è stato notato che alcuni di questi sono abbastanza datati.

Nessuno, insomma, ha analizzato i survey di Stack Overflow nella sua interezza.

Questa tesi, quindi, si occuperà sicuramente di riprendere lavori già fatti in passato, di aggiornarli rispetto al presente e di ampliare ciò che è presente in letteratura con analisi del tutto nuove.

1.3 Risultati

Dallo studio emerge che l'utenza di Stack Overflow in alcuni ambiti si è evoluta se non addirittura capovolta mentre in altri settori ha avuto delle variazioni minime o quasi nulle. I problemi che hanno affrontato gli utenti sono stati vari e basati su molte tecnologie. Nuovi linguaggi di programmazione hanno scalato la classifica dei "linguaggi più usati" mentre altri sono scesi e alcuni quasi del tutto scomparsi. Le occupazioni degli utenti sono cresciute e pressoché costanti sono rimasti i sistemi operativi utilizzati e la quantità di lavoro da remoto prima della pandemia.

1.4 Struttura della tesi

Il resto della tesi è strutturato come descritto in seguito: Nel Capitolo 2 vengono presi in esame i lavori già presenti in letteratura che più si avvicinano al lavoro svolto per questa tesi, si spiega cosa è stato fatto nello specifico e vengono confrontati i risultati ottenuti dagli altri con quelli ottenuti in questo studio, evidenziando similitudini e differenze. Nel capitolo 3, invece, verrà mostrata la metodologia utilizzata per ottenere i dati. Nel capitolo 4, poi, verrà fatta una panoramica approfondita dei risultati ottenuti. Infine, nel capitolo 5, verranno riportate le conclusioni e i possibili spunti per il futuro.

2.1 Background

2.1.1 Linguaggi di programmazione

Chi si avvicina per la prima volta al mondo dell'informatica si sarà sicuramente chiesto cos'è di preciso un linguaggio di programmazione.

Aulab [2] si esprime dicendo che tutti noi parliamo almeno una lingua per esprimere pensieri, concetti astratti e per comunicare con gli altri. Possiamo pensare ai linguaggi di programmazione come la "lingua del computer". Più precisamente sono linguaggi che permettono ai programmatori di parlare con i computer. Si tratta, in concreto, di una serie di istruzioni impartite alla macchina per portare a termine un determinato compito. È piuttosto comune che i meno ferrati utilizzino in modo improprio il termine "linguaggio di programmazione" per indicare anche gli scripting e i markup. Vi sono però delle importanti differenze.

- **I linguaggi di programmazione** si usano per dare istruzioni a un computer, indicando che cosa deve fare. Sono linguaggi ad alto livello e si utilizzano comunemente per realizzare software e driver
- **I linguaggi di scripting** sono una sotto categoria dei linguaggi di programmazione e connettono un linguaggio a un altro linguaggio. Si usano spesso per creare siti e

applicativi web.

- **I linguaggi di markup** sono una categoria a parte, usati per creare la struttura dei dati su una pagina, oppure il suo aspetto visivo. Si tratta quindi di presentare informazioni in modo strutturato.

2.1.2 Linguaggi di programmazione più diffusi

Esistono centinaia di linguaggi di programmazione ognuno più o meno utile per svolgere un determinato lavoro. Di seguito verranno elencate le più utilizzate al giorno d'oggi secondo Matteo Moroni [3]:

- **JavaScript** è sicuramente uno dei linguaggi più utilizzati se non il più utilizzato perché permette di sviluppare siti internet con funzioni dinamiche, interattive o animate. Per questo, JavaScript è presente in gran parte del web.
Nato per codificare front-end, adesso si sta diffondendo anche per l'uso in back-end grazie a NodeJS che consente di eseguire codice JavaScript lato server.
- **Java** è incredibilmente diffuso, grazie alle caratteristiche che lo rendono uno dei linguaggi più stabili, completi ed affidabili per costruire sistemi complessi.
Gli innumerevoli frameworks Java offrono ai programmatori uno sviluppo rapido e solido di applicazioni di ogni tipo.
- **Python** è un linguaggio di programmazione semplice da imparare, ha un codice facilmente leggibile ed è molto versatile. Per queste sue caratteristiche, Python viene considerato da molti sviluppatori uno dei migliori linguaggi di programmazione con cui iniziare a programmare.

2.1.3 Storia della programmazione

Una panoramica generale della storia della programmazione, ci è stata fornita da Gabriella Greison in "Storia di Ada Lovelace, la donna che per prima immaginò il potere del computer" [4] la quale ci racconta che probabilmente possiamo collocare la nascita della programmazione negli anni 40 del 800', più precisamente nel 1842, molto prima dell'invenzione dei computer grazie ad una donna: Ada Lovelace.

Ada Byron nasce nel 1815 in una famiglia difficile, con una madre violenta e un padre che va via di casa quando lei aveva appena 1 anno.

In una situazione familiare difficile da affrontare, Ada trova il conforto e la tranquillità che le

mancavano nei libri di matematica. Finisce gli studi grazie a due professori da cui prendeva lezioni private nelle loro case dove si tratteneva anche oltre l'orario previsto.

Cresce e decide di sposarsi con il conte di Lovelace da cui ne acquisirà il cognome. Ed è successivamente al suo matrimonio che la sua vita cambierà, quando avverrà l'incontro con un matematico di nome Charles Babbage, che la convince a seguirlo nei suoi studi e nelle sue ricerche. Charles è l'inventore della macchina differenziale. Ada è molto incuriosita da lui e dalla sua macchina che decide di studiarla, crea calcoli ed algoritmi. Si rende conto ben presto che avere a che fare con i numeri è la cosa che le riesce meglio e che vuole continuare per tutta la vita. Charles è molto fiero del suo lavoro e grazie a lei riesce a sviluppare nuove forme di programmazione utili tutt'ora.

Man mano che il lavoro con Charles va avanti, ottiene risultati sempre più incredibili, ed è proprio lei che si accorge di avere di fronte un nuovo oggetto che presto cambierà le sorti del mondo.

In un articolo descrive il comportamento che può avere questo oggetto, che sta costruendo insieme al suo maestro: ne parla come "una macchina capace di essere uno strumento programmabile, con una intelligenza simile a quella dell'uomo". In un diario scrive anche di non ritenere plausibile che la macchina in questione "possa divenire con il tempo una macchina pensante, però ci arriverà vicino".

Si preoccupa anche di raccontare nel dettaglio quali sono i lavori che la macchina potrebbe svolgere in un futuro, e in quali compiti potrebbe presto sostituire l'uomo. Ada fa delle riflessioni molto lungimiranti, che neanche Charles riesce a eguagliare e per questo ancora oggi viene ricordata come la madre dei computer moderni.

Ad un certo punto Ada si ammala ma anche dal letto cerca di lavorare, infatti Babbage le propone un nuovo lavoro: la traduzione delle opere del matematico italiano Luigi Menabrea che affrontano proprio la macchina di Babbage. Ada fa molto più che tradurle, le amplia aggiungendo note e appunti, intuì l'idea di loop e di sequenza ripetuta di passi. Alla fine del libro inserisce un nuovo algoritmo per il calcolo dei numeri di Bernoulli. Quell'algoritmo venne considerato come il primo programma informatico della storia.

2.1.4 L'evoluzione della programmazione

Circa 100 anni più tardi iniziano ad apparire i primi calcolatori elettrici, macchine dalla memoria limitata e con velocità di esecuzione ancora molto scarna che rendeva qualunque tipo di operazione un compito molto oneroso, senza contare il fatto che bisognava programmare in linguaggio macchina che consiste nel settare ogni singolo bit a 1 o 0 rendendo il

procedimento molto difficile e destinato solo a pochi eletti.

Fabrizio Ciacchi ci informa in "Storia della programmazione"[5] che nel 1950 con il progresso tecnologico che portò l'abbassamento delle dimensioni e dei costi dei calcolatori, nacquero i primi due linguaggi di programmazione ad alto livello: il FORTRAN, il cui utilizzo era ed è prettamente quello di svolgere in maniera automatica calcoli matematici e scientifici, e l'ALGOL i quali misero le basi per rappresentare le regole dei vari linguaggi di programmazione che stavano nascendo.

Nel 1960 venne presentato il COBOL, ideato per applicazioni nei campi dell'amministrazione e del commercio, per l'organizzazione dei dati e la manipolazione dei file.

Nel 1964 fa la sua comparsa il BASIC, il linguaggio di programmazione per i principianti, che ha come caratteristica fondamentale quella di essere molto semplice e, infatti, diventa in pochi anni uno dei linguaggi più utilizzati al mondo.

Intorno al 1970, però, Niklaus Wirth pensò bene di creare il PASCAL per andare incontro alle esigenze di apprendimento dei neo-programmatori, introducendo però la possibilità di creare programmi più leggeri e comprensibili di quelli sviluppati in BASIC. Si può affermare con certezza che Wirth ha centrato il suo obiettivo, considerando che ancora oggi il Pascal viene usato come linguaggio di apprendimento nelle scuole.

Pochi anni più tardi fa la sua comparsa il C (chiamato così perché il suo predecessore si chiamava B), che si distingueva dai suoi predecessori per il fatto di essere molto versatile nella rappresentazione dei dati. Il C, infatti, ha delle solide basi per quanto riguarda la strutturazione dei dati, però può apparire come un linguaggio assai povero vista la limitatezza degli strumenti a disposizione. Invece la sua forza risiede proprio in questi pochi strumenti che permettono di fare qualsiasi cosa, non a caso viene considerato "il linguaggio di più basso livello tra i linguaggi ad alto livello", per la sua potenza del tutto paragonabile al linguaggio macchina, mantenendo però sempre una buona facilità d'uso.

Ma la vera rivoluzione si è avuta nel 1983 quando Bjarne Stroustrup inventò il C++ (o come era stato chiamato inizialmente "C con classi") che introduceva, sfruttando come base il C, la programmazione Orientata agli Oggetti.

2.1.5 Età dell'utenza

Nel corso dei vari anni sono stati proposti dei sondaggi all'utenza di Stack Overflow chiedendo agli iscritti alcune generalità e informazioni sugli argomenti da loro più richiesti sul sito e, analizzando questi dati, è emerso che i giovani fino a 30 anni risultano essere la fascia dominante dell'utenza attiva che fa uso del forum, raggiungendo il picco massimo

nell'età post scolastica per poi ricevere un costante calo man mano che l'età si alza fino ad arrivare ad una percentuale misera negli ultrasessantenni.

Ma a cosa è dovuto questo grande calo degli anziani nell'uso di Stack Overflow? Dalla loro maggiore esperienza nelle tecnologie informatiche o perché negli ultimi anni gli anziani sono ancora restii dall'uso di Internet e dal cercare informazioni sul web? Di contro ci potremmo porre le stesse domande per i giovani. Perché le loro percentuali raggiungono il picco massimo nell'età post scolastica? Ossia nell'età, presumibilmente, di maggiore studio di programmazione (che sia un percorso universitario o un corso specifico sulla materia), perché i giovani sono più propensi a cercare informazioni sul web? Forse perché preferiscono la strada più veloce, evitando la più lenta, seppur forse migliore, strada dei manuali e dei libri?

Cercheremo di rispondere a queste domande.

Mettendo da parte la situazione della "generazione Z" che viene chiamata anche generazione "social" che nasce e vive su internet, un punto di partenza c'è! Ed è lo studio fatto dall'Autorità per le Garanzie nelle Comunicazioni (AGCOM) in "L'informazione alla prova dei giovani" [6] che ci parla di una vera e propria "fuga delle nuove generazioni dai mezzi tradizionali con una tendenza verso una sola fonte di informazione, ovvero la rete, tramite i social network" (se così possiamo definire un forum come Stack Overflow). L'elevata domanda di informazione, si scontra con la limitata offerta dei metodi tradizionali, per il nostro caso si pensi alla continua e rapida evoluzione (ma anche generazione) di tecnologie di programmazione, che probabilmente non offre tempo materiale ai professionisti di scrivere e pubblicare manuali nuovi o aggiornare quelli vecchi che già il nuovo aggiornamento della tecnologia in esame è alle porte ed è proprio per questo che spesso troviamo delle vere e proprie documentazioni ufficiali online che vengono man mano cambiate.

Ma tornando a noi e allo studio preso in considerazione, emerge che i giovani si interfacciano ai metodi tradizionali in modo sfuggente e disinteressato anche perché vogliono accedere alle informazioni in modo semplice, gratuito e sbrigativo anche dal punto di vista del linguaggio utilizzato, denotando anche una scarsa capacità di apprendimento. C'è da dire che il guadagno di tempo, inoltre, va a discapito della qualità dell'informazione che si sta cercando e che sul web potrebbe essere più o meno veritiera, precisa o ambigua. Si pensi ad una risposta di un utente di Stack Overflow alla domanda di un altro utente. Quanto può essere preciso o ambiguo un programmatore che utilizza un forum rispetto ad un manuale? Quanto può essere veritiera la sua informazione? Certo spesso si parla di codice e un codice di programmazione o funziona o non funziona. Ma se dopo aver dato una risposta funzionante la tecnologia

in esame rilasciasse un aggiornamento che fa diventare obsoleta la precedente risposta? In questo caso si ritorna al discorso dell'assenza o quasi di manuali di programmazione (almeno per le tecnologie più nuove).

Quindi se da un lato abbiamo giovani più propensi a fruire news sul web, dall'altro abbiamo anche il problema che alcune documentazioni sono fruibili solo online. E allora la risposta alla domanda potrebbe essere proprio questa.

Per quanto riguarda gli anziani invece, è noto che, data la loro età, sono più propensi a carpire informazioni da metodi tradizionali (giornali, TV ecc) ma anche in questo caso prenderemo in considerazione uno studio, stavolta fatto dall'Auser che ci informa, appunto, che la quantità di anziani che utilizza internet negli ultimi anni è aumentata in modo considerevole (fino al 17%). Ma cosa si cerca su internet? Soprattutto informazioni, posta elettronica e curiosità prediligendo, però, i metodi tradizionali come la TV, i giornali, i libri etc.

Anche in questo caso la bilancia pende tutta da un lato dato che se è pur vero che la percentuale di anziani che usa internet è in aumento, essa rimane comunque la fetta minore rispetto alla totalità e c'è inoltre da considerare la loro preferenza nell'informazione tradizionale e la loro esperienza acquisita negli anni in campo di programmazione.

2.1.6 Genere

Tornando all'analisi dei sondaggi fatti all'utenza di Stack Overflow non si è potuto non notare una sostanziale differenza di genere nel mondo dell'informatica che vede gli uomini dominare la scena con percentuali che si aggirano sempre oltre il 90%.

Ma perché c'è così poca affluenza di donne in ambito informatico?

Uno studio commissionato da Cisco "Donne e ICT: perché le ragazze non entrano nel mondo della tecnologia" [7] indaga i motivi per cui la disparità di genere nelle professioni legate al mondo delle nuove tecnologie è così accentuata. L'interesse per l'informatica e l'uso quotidiano del PC non bastano se mancano modelli femminili a cui ispirarsi e persiste la convinzione che l'ICT sia "un mondo da uomini". Lo studio attesta che tra i ragazzi e le ragazze delle scuole superiori sussiste una sostanziale parità di interesse nelle scienze informatiche eppure la maggior parte delle ragazze non intende proseguire gli studi nel settore tecnologico. Il fattore che demotiva le studentesse è la convinzione che l'ambito ICT sia "di per sé più adatto agli uomini". Altro fattore determinante che spinge le ragazze a scegliere un percorso universitario differente è il modello di riferimento a cui ispirarsi per le scelte di studio e carriera, nel loro caso prevalentemente la madre o un insegnante di sesso femminile. Madri e insegnanti, però, non sono un buon modello per quanto riguarda

l'utilizzo delle tecnologie: usano il PC meno degli uomini, e lo usano per fare meno cose. Secondo lo studio preso in esame, questi modelli utilizzano il computer meno di 30 minuti al giorno e solo il 5% di loro ha dichiarato di avere competenze minime di networking (valutate come la capacità di risolvere piccoli problemi di connettività domestica autonomamente), un dato che assume rilievo se confrontato con quello relativo agli uomini.

Secondo Jane Margolis, co-autrice del libro "Unlocking the Clubhouse: Women in Computing" il fattore che più spinge le donne al mondo dell'ICT è la presenza di un familiare nel settore. Ma le donne sono sempre state presenti nel mondo dell'informatica fin dalle sue origini e hanno contribuito in modo sostanziale ad evolvere questa realtà ed hanno avuto un ruolo importante. Per citare alcuni grandi esponenti donna del settore abbiamo, oltre alla già citata Ada Lovelace:

- **Grace Murray Hopper:** creatrice del primo compilatore della storia, liberando i programmatori dallo scrivere codice in linguaggio binario rendendo il computer accessibile a più persone, ed ebbe un ruolo fondamentale nello sviluppo e nella progettazione del linguaggio COBOL. Famosa anche per aver coniato la parola "bug".
- **Rosza Peter:** la creatrice delle funzioni ricorsive. Ungherese, studia matematica all'università di Budapest e dopo la laurea comincia la sua carriera come tutor e ricercatrice.

2.2 Evoluzione dei linguaggi di programmazione

Uno degli studi più interessanti presenti in letteratura sull'evoluzione dei linguaggi di programmazione è "L'evoluzione dei linguaggi di programmazione: analisi e prospettive" [8] di Giancarlo Succi del 2003. L'autore inizia parlandoci un po' del passato e facendo una carrellata storica. Ci dice che inizialmente, nei primi calcolatori, i programmi erano definiti da circuiti elettrici, veri e propri collegamenti fisici passando poi ai codici binari e i linguaggi assemblatici, ognuno dei quali spesso era associato univocamente ad un calcolatore: cambiando elaboratore si cambiava linguaggio. Il primo sostanziale passo in avanti avvenne a cavallo degli anni 50 e 60, quando nacquero i linguaggi ad alto livello come il FORTRAN e il COBOL. Verso la fine degli anni 60 e l'inizio degli anni 70 ci si rese conto che i programmi diventavano sempre più grandi e complessi e richiedevano la collaborazione di più persone. L'idea che parve più ovvia fu quella di basarsi su modelli modulari, concentrati sulla suddivisione del lavoro in parti. Il primo modello fu chiamato "a cascata". Per questo motivo ci si focalizzò sullo sviluppo di linguaggi in grado di permettere la suddivisione di un sistema in sottosiste-

mi. Fu così che nacquero l'ALGOL il C e il PASCAL. La programmazione strutturata però acquisisce la sua massima espressione solo un po' più tardi, quando nacquero il MODULA 2 e l'ADA. Con l'avvento degli anni 80 il modello a cascata iniziò a mostrare i suoi limiti e fu introdotto il modello a V in cui si associava a ogni fase di sviluppo (ovvero analisi, progetto e codifica), una batteria di test prima della codifica, poi del progetto e poi dell'analisi. Dato questo movimento discendente nella fase di sviluppo dal più generale al più particolare e una fase ascendente dei test dal più particolare al più generale, questo modello prese il nome di V. Negli anni 80, di colpo, i modelli di divisione del lavoro divennero inadeguati perché se lo scopo di un sistema da sviluppare non era chiaro, una qualunque sua suddivisione rischiava di rendere ancora meno chiaro lo scopo del sistema. Allora si pensò a creare il lavoro con "piccoli pezzi". Si pensava al contesto applicativo e si creavano entità che sicuramente sarebbero esistite e che avrebbero comunicato con altre entità. Per esempio, si pensi ad un sistema per un aeroporto che doveva sicuramente tenere in considerazione le relazioni tra voli, passeggeri, biglietti e via dicendo. Queste entità vennero chiamate "oggetti" i quali avevano incorporati una serie di metodi per rispondere ai messaggi ed andarono a sostituire le chiamate a funzione. Nacque il C++ basato sul C che in realtà, inizialmente, non fu considerato un vero e proprio linguaggio ad oggetti ma un ibrido. Lo sviluppo a piccoli pezzi si concretizzò con l'arrivo dei modelli di produzione detti incrementali, prototipali e a spirale. I linguaggi di programmazione orientata agli oggetti furono un arma a doppio taglio perché è vero che da un lato si ebbe un sostanziale miglioramento dall'altro richiese una competenza decisamente superiore da parte degli sviluppatori e imparare linguaggi complessi ovviamente non era una cosa facile considerando che chi si avvicinava alla programmazione orientata agli oggetti veniva da anni di uso di linguaggi meno evoluti. Successivamente la domanda del mercato di prodotti in tempi brevi e la comparsa del web che aggiunse la necessità di un linguaggio che si adattasse a combinare pezzi di codice che si muovevano sulla rete fecero la fortuna e il successo di Java.

2.3 Mining di Stack Overflow

Il data mining è l'insieme di tecniche e metodologie che hanno la finalità di estrarre informazioni utili da una grande quantità di dati attraverso metodi automatici o semi-automatici.

Negli anni ci sono stati diversi lavori di mining sulla piattaforma di Stack Overflow, andiamo a vederne alcuni.

Nel 2014 Ponzanelli et al. attraverso il mining di Stack Overflow hanno trasformato un IDE in un suggeritore. In "Mining stackoverflow to turn the ide into a self-confident programming prompter" [9] scopriamo che gli autori si sono resi conto che spesso i programmatori si ritrovano dinanzi a problematiche che non riescono a risolvere tramite le loro conoscenze, pertanto si ritrovano costretti a consultare fonti di informazioni come documentazione API, forum, siti Web di domande e risposte, ecc. Sapere cosa cercare e come non è banale e formulare problemi sotto forma di domande richiede loro una gran dispendio di energie. Per questo motivo gli autori hanno proposto un approccio tramite plug-in su Eclipse chiamato Prompter che, dato un contesto nell'IDE, recupera automaticamente le discussioni da Stack Overflow, ne valuta la pertinenza e, se viene superata una determinata soglia di confidenza, notifica allo sviluppatore l'aiuto disponibile.

Nel 2018 Liu et al in "Searching stackoverflow questions with multi-faceted categorization" [10] hanno notato che la ricerca di domande pertinenti su Stack Overflow non è sempre semplice inserendo parole chiave nella barra di ricerca del sito. Infatti una domanda può essere caratterizzata da vari attributi come per esempio linguaggio di programmazione, sistema operativo e middleware, framework, libreria e tecnologia software coinvolti e tanti altri. Gli autori propongono un approccio sfaccettato e interattivo per la ricerca di domande StackOverflow chiamato MFISSO, che sfrutta proprio questi attributi. L'approccio inizia con una query iniziale basata su parole chiave ed estrae una categorizzazione sfaccettata da tutte le domande candidate utilizzando l'elaborazione del linguaggio naturale e il data mining. Consente quindi agli sviluppatori di perfezionare iterativamente i risultati della ricerca attraverso un processo interattivo. E' stata valutata un'implementazione di MFISSO controllando l'attività 20 studenti di informatica e sono state risolte dieci problematiche di sviluppo software utilizzando StackOverflow. L'esperimento mostra che MFISSO può aiutare gli sviluppatori a trovare domande pertinenti più velocemente e con maggiore precisione.

Nel 2012 Bogdan et. al. in "Gender, representation and online participation: A quantitative study of stackoverflow"[11] per recuperare i dati inerenti al genere degli utenti su Stack Overflow e dato che a quei tempi il forum non chiedeva esplicitamente agli utenti di che genere fossero hanno previsto due approcci basandosi sui survey della piattaforma: uno automatico e uno manuale usando quest'ultimo solo in caso il primo non avesse dato riscontro. L'approccio automatico prevedeva di dedurre il genere in base al nome della persone mentre il processo manuale prevedeva di dedurre questa informazione controllando l'immagine dell'avatar o controllando fonti di dati aggiuntive come Twitter, LinkedIn e GitHub.

In primo luogo per la risoluzione automatica, Bogdan et al, hanno pensato di abbinare ogni

nome ad un paese perché hanno considerato che ci sono casi in cui un nome come per esempio Andrea risulta essere la maggior parte delle volte maschile in Italia ma femminile in Germania. Per questo hanno previsto delle tuple nome,paese dove possibile. Per determinare, invece, i nomi di chi ha usato dei nickname, si sono rifatti alle pagine web collegate agli account. Hanno utilizzato uno strumento Python che accetta tuple nome,paese e restituisce in output "maschio", "femmina" oppure "x" cioè che non è stato possibile dedurre il genere. In casi come questo hanno proceduto con la tecnica manuale.

Rosen e Shihab, invece, in "What are mobile developers asking about? a large scale study using stack overflow" [12] per analizzare le maggiori sfide che affrontano gli sviluppatori in campo mobile, hanno scaricato il dump dei dati di Stack Overflow in formato XML e dato che i post su Stack Overflow possono essere di qualunque tipo, hanno filtrato quelli inerenti al campo mobile attraverso l'uso di parole chiave e tag. Una volta ottenuto il set di post di interesse, al fine di esplorare i problemi di alto livello che devono affrontare gli sviluppatori mobili, sono stati utilizzati modelli di argomenti basati su LDA per aggregare e scoprire cosa viene chiesto nei post. LDA è stata riconosciuta come una delle migliori tecniche per trovare argomenti di discussione nei documenti di testo in linguaggio naturale. Dato che l'obiettivo dell'utilizzo del modello LDA è scoprire cosa chiedono gli sviluppatori decidono di estrarre solo i titoli da ciascuna domanda poiché i titoli riassumono e identificano i concetti principali che vengono posti nel post e perché il corpo della domanda aggiunge informazioni non rilevanti. Utilizzando solo i titoli, si ottiene un corpus contenente i riepiloghi di ciò che gli sviluppatori chiedono. Quindi ogni parola è stata "tokenizzata" e sono stati rimossi tutti gli avverbi, le congiunzioni, i pronomi e le preposizioni inglesi comuni.

2.4 Limitazioni dello stato dell'arte

Nel mondo ci sono state diverse analisi sull'uso di Stack Overflow, una fra queste è sicuramente lo studio fatto nel 2012 da Bogdan et. al. in "Gender, representation and online participation: A quantitative study of stackoverflow"[11] dove ci informano del costante crescendo delle community online e hanno notato che queste ultime sono sostanzialmente "prive" di donne e proprio a causa della mancanza di evidenze empiriche in merito è stato portato avanti uno studio quantitativo del fenomeno, al fine di valutare la rappresentazione e l'impatto sociale del genere in Stack Overflow. Questo studio si colloca all'interno di mirate iniziative internazionali, lanciate dalla Commissione Europea per incoraggiare le donne nel campo delle scienze e della tecnologia. I risultati confermano che gli uomini rappresentano la

stragrande maggioranza dei contributori a Stack Overflow. Inoltre, gli uomini partecipano di più, guadagnano più reputazione e si impegnano nel "gioco" più delle donne. Gli autori, però, hanno studiato il fenomeno a causa della mancanza di evidenze empiriche e non per osservarne l'evoluzione tanto è vero che hanno esportato dati relativi solo ad un unico anno, il 2012. Seppur con motivazioni diverse da questa tesi, si può affermare che i risultati ottenuti da Bogdan et al. non smentiscono quelli ottenuti per la mia ricerca anzi confermano ed estendono il lavoro che si è focalizzato a seguire varie categorie di dati (tra cui anche il genere) negli anni dal 2015 al 2021, trovando sempre enormi disparità. C'è da dire che gli autori di questo studio hanno dovuto fare un lavoro più oneroso per ottenere i dati in quanto i sondaggi dell'epoca non raccoglievano informazioni inerenti al genere dell'utente mentre negli ultimi anni è sempre stato fatto, pertanto è bastata una semplice query al programma per ottenere le percentuali.

Moutidis et. al., invece, in "Moutidis I, Williams HTP (2021) Community evolution on Stack Overflow." [13] hanno analizzato tutte le domande, risposte, voti e tag di Stack Overflow tra il 2008 e il 2020, generato una serie di grafici di interazione utente-tecnologia e applicato algoritmi di rilevamento per identificare le più grandi comunità di utenti ogni anno, esaminare quali tecnologie incorporano tali comunità, come sono interconnesse e come si evolvono nel tempo.

E' emerso che la stragrande maggioranza degli utenti è legata allo sviluppo web e che non tende nel corso del tempo a diminuire, "evolversi" o spostarsi su altre tecnologie seppur sempre popolari e comunque presenti tra le Q/A di Stack Overflow. Sono stati analizzati, poi, i linguaggi più diffusi e richiesti ed è emerso dai grafici che i linguaggi in costante crescita sono Python e R. In particolare Python diventa il linguaggio al primo posto entro la fine del 2020, dopo essere cresciuto in modo significativo per sostituire JavaScript. Questo è uno degli studi che più si avvicina a questo lavoro di tesi che intende studiare l'evoluzione delle tecnologie, in primo luogo perché vengono analizzati dati di un periodo di tempo molto lungo e abbastanza recente. In secondo luogo il tipo di dati analizzati corrispondono proprio a categorie prese in considerazione anche nella mia analisi, ossia le "tecnologie più popolari" e "occupazione" dove si è osservato l'andamento dei linguaggi più utilizzati negli anni e il tipo di occupazione che svolgono i programmatori sul posto di lavoro, trovando anche una certa corrispondenza in termini di risultati. Ciò che differisce molto, in realtà è stata la metodologia perché in questo caso non si è andati ad osservare i sondaggi che Stack Overflow rilascia ogni anno bensì si è optato per un processo forse più oneroso, ossia quello di recuperare tutte le interazioni degli utenti con la piattaforma tramite il database della piattaforma stessa che è

comunque reso pubblico e accessibile a tutti ed estrapolando i dati tramite appositi algoritmi. Rosen e Shihab, invece, in "What are mobile developers asking about? a large scale study using stack overflow" [12] nel 2016, notando l'esponenziale crescita dei dispositivi mobile e che pochissimi lavori si sono concentrati sui problemi degli sviluppatori mobile, hanno esaminato ciò che chiedono tali sviluppatori sulla piattaforma ed i risultati mostrano che chiedono informazioni principalmente su distribuzione di app, API mobile, gestione dei dati, sensori e contesto, strumenti mobile e sviluppo dell'interfaccia utente. Questo studio risulta essere molto interessante perché va ad esplorare quella parte di community che si occupa della programmazione mobile e va a informarsi su quali sono gli aspetti di questo campo che danno maggiore difficoltà. Questo studio, in realtà si discosta molto dall'intento di questa tesi che vuole osservare l'evoluzione della community nel tempo. In realtà per fare ciò si è dato un occhio anche all'area mobile, più precisamente quando è stato osservata la categoria inerente all'occupazione dei dipendenti informatici sul lavoro e si è notato che negli ultimi anni gli sviluppatori mobile sono raddoppiati rispetto al 2015.

Si pensa che diventare un esperto di programmazione richieda circa 10.000 ore di pratica deliberata. Ma cosa succede dopo? Gli esperti di programmazione continuano a svilupparsi, si stabilizzano o c'è un declino a un certo punto? Esiste una diversità di opinioni su questo argomento, ma molti sembrano pensare che l'invecchiamento comporti un declino nell'adozione e nell'assorbimento di nuove conoscenze di programmazione. In "Is programming knowledge related to age? an exploration of stack overflow" [14] Morrison e Murphy-Hill, nel 2013, hanno attinto ai dati di Stack Overflow per rispondere a queste domande. L'obiettivo di questa ricerca è supportare la pianificazione della carriera e lo sviluppo del personale per i programmatori identificando le tendenze legate all'età nei dati della piattaforma. È stato osservato che i punteggi di reputazione dei programmatori aumentano rispetto all'età fino agli anni '50, che i programmatori sui 30 anni tendono a concentrarsi su meno aree rispetto a quelli più giovani o più anziani. Inizialmente sono stati esplorati i dati SO scaricando il dump dei dati PostgreSQL, sono state create query equivalenti per estrarre informazioni dal sito. La popolazione complessiva di utenti di Stack Overflow è di 1694981 programmatori, con un'età media di 30 anni. L'aspetto trattato da questo studio è stato già ampiamente osservato nel paragrafo 2.1.5 e approfondito nei capitoli successivi. Si può dire che il lavoro e i risultati ottenuti da Morrison e Murphy-Hill siano abbastanza congruenti con la ricerca svolta per questa tesi. Seppure gli obiettivi e le osservazioni fatte differiscano nella sostanza, perché gli autori di questo studio si sono soffermati anche sulle aree che gli utenti visitano in base all'età mentre per questa tesi è stata fatta una stima solo sulla popolarità degli utenti in base all'età,

i risultati trovano comunque qualche somiglianza seppure ci siano molti anni di differenza tra la ricerca di Morrison e Murphy-Hill e la mia. Entrambi i risultati, infatti, mostrano che la fascia che gira intorno ai 30 anni è la più popolare sul forum e che la fascia over 50 da un lato, vede che i punti reputazione ottenuti sul sito calano e dall'altro lato si è osservato che gli utenti a partire da quella fascia sono meno attivi.

L'obiettivo dello studio è osservare l'utenza di Stack Overflow, come si è evoluta nel tempo, quali sfide di programmazione si sono posti maggiormente gli utenti, quali tecnologie sono più usate e cosa fanno nella vita gli utilizzatori del sito.

Per fare ciò, la ricerca si è concentrata sull'analisi dei sondaggi che ogni anno il forum rilascia alla sua utenza e che pubblica sul sito lasciando libera la consultazione a tutti.

I survey vengono sottoposti agli utenti solitamente dal 25 maggio al 15 giugno dell'anno preso in considerazione. Gli intervistati vengono "reclutati" attraverso i canali di proprietà di Stack Overflow. Le principali fonti sono state la messaggistica in loco, post su blog e social media, e-mail e banner pubblicitari. Per questo motivo gli utenti molto coinvolti su Stack Overflow avevano maggiori probabilità di notare le richieste di partecipare al sondaggio per tutta la durata della promozione della raccolta. Come incentivo, gli intervistati che hanno completato il sondaggio hanno ottenuto la possibilità attivare un badge "Census". A causa delle sanzioni di trasporto/esportazione degli USA, i sondaggi non sono stati accessibili agli utenti in Crimea, Cuba, Iran, Corea del Nord e Siria. Sebbene alcuni intervistati abbiano utilizzato le VPN per aggirare il blocco, è necessario tenere presente la limitazione durante l'interpretazione dei risultati del sondaggio. Inoltre sono stati esclusi i risultati ottenuti dagli utenti che hanno impiegato meno di 3 minuti per concludere l'operazione.

Reperire i risultati di questi sondaggi è molto semplice in realtà, basta digitare su Google le parole chiave "survey Stack Overflow" e tra i primi risultati si accede facilmente a una pagina che prevede un elenco di anni (dal 2011 al 2021) e un apposito link che permette di

scaricare dei file CSV (formato che permettere di rappresentare semplicemente i dati in forma tabellare) e visualizzare tutte le domande e le risposte degli utenti.

Per adempiere al lavoro di questa tesi, in primo luogo, sono stati scaricati i tali file dal 2015 al 2021 i quali sono stati letti e osservati sul software di casa Apple "Numbers", utilizzabile tramite iCloud anche da chi non possiede il sistema operativo MacOS. Questo software è stato scelto per la sua enorme potenza in grado di gestire tabelle molto grandi (come quelle prese in esame) e aprire file CSV senza perdita di formattazione.

Sono state trovate molte domande inerenti la vita quotidiana degli utenti come per esempio la quantità d'uso di caffeina, il possesso o meno di una famiglia, l'ora in cui si svegliano, quanto sport praticano ed anche altro. Tutte queste categorie non sono state considerate ai fini di questo studio che intende osservare come il mondo della programmazione si è evoluto nel tempo. Pertanto sono state scelte le categorie che più di tutte possono esporre questo fenomeno: i linguaggi di programmazione più utilizzati dagli utenti per osservare come negli anni le percentuali sono cambiate, se c'è stato un linguaggio che mano mano è stato messo da parte o se qualcosa di nuovo è arrivato a compararsi con le tecnologie già presenti e affermate da tempo, i linguaggi più ricercati per capire cosa ha dato maggiore difficoltà ai programmatori, i dati anagrafici dell'utenza come l'età e il genere per capire come la società si è approcciata a questo mondo in passato e come lo sta facendo ora. Anche l'occupazione dei programmatori sul posto di lavoro può far comprendere come l'ICT si sta evolvendo in questo ambito e in cosa adoperano maggiormente le loro skills. Inoltre come piccolo extra si è voluto osservare che rapporto avesse la gente con lo smart working prima che la pandemia Covid-19 sdoganasse il lavoro da remoto.

Le categorie prese in considerazione, sono riepilogate nella tabella 3.1:

Categoria	Motivazione
Età	Per comprendere quanti giovani si stanno avvicinando a questo mondo e quando le persone smettono di informarsi sulla piattaforma.
Genere	Per capire se il mondo dell'ICT, da sempre considerato un ambiente prettamente maschile, è stato accolto con più entusiasmo anche dalle donne.
Tecnologie più popolari	Per sapere quale linguaggio è il più utilizzato dall'utenza, se sono stati accantonati i più datati e se qualcuno nuovo è riuscito ad affermarsi.
Tecnologie più ricercate	Per capire quale linguaggio crea maggiore difficoltà ai programmatori.
Sistemi operativi più utilizzati	Per sapere su quali macchine la gente preferisce lavorare.
Occupazione	Per conoscere cosa fanno nello specifico a lavoro i programmatori. In che ambito utilizzano di più le loro skills
Lavoro da remoto	Per studiare la situazione "smart working" prima che la stessa fosse sdoganata dall'emergenza Covid-19

Tabella 3.1: Categorie considerate nel contesto dello studio

I file CSV sono risultati spesso enormi, arrivando a toccare anche quasi i 100.000 intervistati.

Per estrarre i dati, si è fatto uso, su "Numbers", per ogni colonna (categoria) delle query ossia di "interrogazioni" fatte al programma che rispondeva fornendo il numero di intervistati che avevano scelto una determinata risposta. Una volta estratti questi numeri sono stati convertiti in percentuali.

Si è notato che ogni intervistato, in alcune categorie, ha potuto scegliere tramite spunte più di una preferenza per ogni domanda, motivo per cui la somma delle percentuali, alle volte, è risultata essere superiore al 100%. Per esempio quando è stato chiesto quale fossero i linguaggi più utilizzati o che occupazione svolgessero al lavoro.

Allo stesso modo si è notato che in alcune categorie il totale non raggiungeva il 100% perché gli utenti potevano anche decidere di non rispondere a un determinato quesito.

Alcune domande, invece, seppur uguali nei vari survey, hanno concesso di rispondere in

modo differente. Per esempio quando sono stati chiesti dei dati anagrafici come l'età, il sondaggio negli anni ha proposto come possibilità di risposta dei range differenti. E' stato deciso, in casi come questo, di unificare il tutto. Cioè mostrare dei range univoci per tutti gli anni presi in considerazione.

Un'altra piccola modifica che è stata apportata alle categorie, come si vedrà, è stata quella di unificare sotto un'unica voce HTML e CSS, originariamente divise, che essendo tecnologie che lavorano a stretto contatto hanno sempre ricevuto le stesse percentuali (o quasi).

Età							
	<20	20-24	25-29	30-34	35-39	40-49	50+
2015	8,80%	24,50%	28,50%	17,80%	9,10%	7,60%	2,40%
2016	7,10%	23,60%	28,50%	18,10%	10,20%	8,90%	3,80%
2017	-	-	-	-	-	-	-
2018	4,50%	21,60%	20,52%	28,70%	10,68%	9,10%	4,80%
2019	6,20%	21,10%	27,90%	19,30%	11,90%	9,60%	4,20%
2020	5,90%	20,30%	26,50%	19,40%	12,40%	10,60%	5,00%
2021	9,02%	21,97%	23,71%	18,81%	12,42%	10,64%	4,42%

Tabella 4.1: Andamento dell'età degli utenti di Stack Overflow

La Tabella 4.1 ci mostra i dati relativi all'età dell'utenza di Stack Overflow. Dai risultati ottenuti si evince che essa è rimasta sostanzialmente costante nel corso degli anni. Possiamo notare che la fascia predominante è quella dai 20 ai 29 anni. Subito seguita dalla fascia 30-39 e 40-49. Nei due gradini più bassi troviamo gli under 20 e gli over 50. Si noti come i meno influenti sulla piattaforma siano i ragazzi in età pre universitaria e gli anziani, (forse per esperienza o forse perché si preoccupano di cercare informazioni tramite altri strumenti). I più attivi invece sono proprio quelli che si avvicinano al mondo dell'università o del lavoro, proba-

bilmente perché più inesperti e con più voglia di apprendere informazioni o cercare soluzioni.

	Genere	
	Uomini	Donne
2015	92,10%	5,80%
2016	92,80%	5,80%
2017	88,60%	7,60%
2018	92,90%	6,90%
2019	87,60%	7,10%
2020	91,50%	8,00%
2021	91,67%	5,31%

Tabella 4.2: Differenze di popolarità tra uomini e donne presenti su Stack Overflow

La Tabella 4.2 ci mostra la differenza di genere su Stack Overflow. Che il mondo dell'ICT sia sempre stato un ambiente prettamente maschile è risaputo. Ebbene, i dati confermano che è tutt'altro che sbagliato. Infatti dal 2015 ad oggi oltre il 90% degli utenti di Stack Overflow ha dichiarato di essere uomo, un dato che forse ci dice che siamo ancora lontani dall'avere una crescita sostanziale di donne nelle aule universitarie informatiche e negli uffici di programmazione.

	Tecnologie più popolari								
	C	C++	C	Java	JavaScript	Python	PHP	SQL	HTML /CSS
2015	16,40%	20,60%	31,60%	37,40%	54,40%	23,80%	29,70%	48,00%	-
2016	15,50%	19,40%	30,90%	36,30%	55,40%	24,90%	25,90%	49,10%	-
2017	18,90%	22,10%	33,80%	39,30%	61,90%	31,70%	27,90%	50,80%	-
2018	23,00%	25,40%	34,40%	45,30%	69,80%	38,80%	30,70%	57,00%	66,80%
2019	20,60%	23,50%	31,00%	41,10%	67,80%	41,70%	26,40%	54,40%	63,50%
2020	21,80%	23,90%	31,40%	40,20%	67,70%	44,10%	26,20%	54,70%	63,10%
2021	21,01%	24,31%	27,89%	35,35%	64,96%	48,24%	21,98%	47,08%	56,07%

Tabella 4.3: Tecnologie più usate negli anni dagli utenti di Stack Overflow

La Tabella 4.3 mostra i dati inerenti alle tecnologie più utilizzate dove si evince che la

più popolare (almeno dagli utenti del forum) sembra essere JavaScript che si pone sempre al primo posto con il susseguirsi degli anni. JavaScript è il linguaggio di programmazione più popolare per progettare app e siti web interattivi e sembra che il 98% dei siti web utilizzino questo linguaggio. Per lo stesso motivo si colloca costantemente al secondo posto HTML/CSS posti sotto un'unica cella perché lavorano a stretto contatto tra loro. Si può dire che troviamo questi due linguaggi in praticamente tutti i siti web. L'HTML definisce la struttura di una pagina mentre il CSS si occupa della grafica (quindi dei colori, il font e le immagini). Successivamente troviamo SQL il linguaggio più diffuso per gestire i database relazionali. Lo scalino successivo, invece, è interessante da osservare perché possiamo notare che il famoso e ampiamente diffuso Java viene superato a partire dal 2019 in poi da Python che si afferma come linguaggio maggiormente in crescita con la sua facilità nell'essere appreso e la sua grande versatilità. Un dato molto interessante perché ci dice che nuove tecnologie si stanno affermando e si stanno sostituendo a quelle più vecchie.

Tecnologie più ricercate									
	C	C++	C	Java	JavaScript	Python	PHP	SQL	HTML /CSS
2015	-	-	8,00%	10,50%	15,40%	14,80%	-	-	-
2016	-	8,00%	-	-	11,90%	13,30%	-	-	-
2017	6,40%	11,80%	10,10%	11,70%	18,60%	20,60%	4,10%	6,50%	-
2018	5,90%	10,20%	8,00%	10,50%	19,00%	25,10%	4,10%	6,80%	15,20%
2019	5,00%	9,10%	7,00%	8,30%	17,80%	25,70%	3,50%	7,60%	6,60%
2020	4,30%	8,60%	7,30%	8,80%	18,50%	30,00%	3,50%	8,20%	6,50%
2021	4,52%	8,80%	6,45%	6,79%	14,59%	19,04%	2,55%	6,30%	4,67%

Tabella 4.4: Linguaggi che sono stati più ricercati negli anni su Stack Overflow

La Tabella 4.4 mostra i dati dei linguaggi più ricercati su Stack Overflow. La tecnologia che gli utenti hanno ricercato di più sul sito negli anni è Python, un dato che in effetti non sorprende se pensiamo che nella categoria precedente abbiamo potuto notare che è stato il linguaggio che è cresciuto maggiormente in termini di popolarità. Per lo stesso motivo, non sorprende che JavaScript sia sempre stato il secondo linguaggio più ricercato, essendo il più utilizzato. Java, poi, al terzo posto parte bene insieme a C++ con il quale procede di pari passo con le percentuali che si abbassano negli ultimi anni rispetto al 2015. Più basse e sostanzialmente costanti rimangono, invece, i numeri delle altre tecnologie. E' interessante

notare come tecnologie molto usate tra cui tra cui HTML/CSS e SQL siano molto poco ricercate rispetto alla loro popolarità, probabilmente perché non forniscono difficoltà tali da dover ricercare una risposta sul forum.

Sistemi operativi più utilizzati			
	Windows	Linux	MacOS
2015	53,30%	20,50%	21,50%
2016	51,70%	21,70%	26,20%
2017	-	-	-
2018	49,90%	23,20%	26,70%
2019	47,50%	25,60%	26,80%
2020	45,80%	26,60%	27,50%
2021	45,33%	25,32%	25,19%

Tabella 4.5: Sistemi operativi preferiti dagli utenti di Stack Overflow

Nella Tabella 4.5 possiamo osservare quali sono stati i sistemi operativi più in voga. E' noto che il sistema Windows è il preferito dalla community di utilizzatori del computer. Questo si evince anche dai programmatori che girano su Stack Overflow. Infatti dai dati emerge che la metà degli utenti usa proprio Windows mentre l'altra metà si alterna tra MacOS e Linux ma con una leggera preferenza verso il primo.

Occupazione						
	Sviluppatore web completo	Sviluppatore back-end	Sviluppatore front-end	Sviluppatore mobile	Sviluppatore desktop	Studente
2015	32,40%	10,10%	6,00%	9,10%	8,30%	13,60%
2016	28,00%	12,20%	5,80%	8,40%	6,90%	11,40%
2017	31,85%	12,20%	5,90%	9,08%	-	-
2018	48,20%	57,90%	37,80%	20,40%	17,20%	17,10%
2019	51,90%	50,00%	32,80%	18,10%	21,30%	14,70%
2020	54,90%	55,20%	37,10%	19,20%	23,90%	-
2021	49,47%	43,73%	27,42%	14,74%	16,60%	6,30%

Tabella 4.6: Occupazione sul lavoro degli utenti di Stack Overflow

Che impiego ha chi circola su Stack Overflow? Dalla Tabella 4.6 emerge che dal 2015 al 2018 gli utenti erano maggiormente coinvolti in lavori di sviluppatore web completo subito seguito dal solo sviluppatore back-end e dal 2018 anche dal solo front-end. E' interessante osservare come l'affermazione degli smartphone abbia portato progressivamente a raddoppiare il numero di sviluppatori mobile. Gli studenti, invece, sono sempre stati più o meno costanti.

Lavoro da remoto				
	Remoto a tempo pieno	Remoto part-time	Remoto raramente	Remoto mai
2015	10,40%	18,60%	58,10%	22,80%
2016	12,10%	17,60%	48,40%	22,00%
2017	11,10%	17,70%	35,10%	31,80%
2018	-	-	-	-
2019	12,00%	15,25%	33,50%	43,00%
2020	-	-	-	-
2021	-	-	-	-

Tabella 4.7: Situazione smart working prima dell'emergenza Covid-19

la Tabella 4.7 ci mostra la situazione di smart working prima dell'emergenza sanitaria. Con l'avvento del Covid nel 2019 e le quarantene nel 2020 e quindi con un lavoro totalmente da remoto, è stato interessante osservare questo fenomeno prima che la pandemia prendesse piede. Oggi il lavoro totalmente o in parte da remoto è sicuramente sdoganato ma come era la situazione prima? Dai risultati ottenuti scopriamo che il lavoro da remoto a tempo pieno era quello meno diffuso. Dal 2015 al 2019 è sempre oscillato tra il 10 e il 12%. Ma anche lavorare part-time da remoto era sostanzialmente poco diffuso con percentuali tra il 15% e il 18%. I più invece hanno risposto ai sondaggi che lavoravano da remoto "raramente" e la seconda fetta più grande della community invece non lavorava mai a distanza.

5.1 Risultati

In questa tesi sono stati analizzati i survey che ogni anno Stack Overflow rilascia ai suoi utenti e che pubblica sul suo sito lasciando la libera consultazione a tutti. L'osservazione di tali sondaggi ha portato a valutare l'evoluzione della community di programmatori che fanno uso della suddetta piattaforma.

In generale, si è notato, che per certi versi l'utenza è cambiata molto ma in altre occasioni è rimasta tendenzialmente la stessa. Si pensi all'età che hanno gli utenti su Stack Overflow e il loro genere, le cui percentuali negli anni sono rimaste sostanzialmente le stesse: la fascia dai 20 ai 30 è quella più corposa con un graduale ribasso nelle fasce successive fino ad arrivare ai picchi più bassi della fascia over 50. Inoltre gli uomini da sempre sono rimasti in netta maggioranza rispetto alle donne. Ritengo che quest'ultimo dato sia rilevante perché, come ampiamente visto nel paragrafo 2.1.6, la mancanza di donne non è dovuta da un disinteresse o da una mancanza di bravura da parte loro, bensì da una mancanza di modelli di riferimento. Questo dato, inoltre, è rilevante perché ci dimostra che nelle scuole da parte degli orientamenti non c'è il giusto incentivo, si dovrebbe fare di più per indirizzare le ragazze verso questo settore che rappresenta il presente e il futuro, offre tanto lavoro e garantisce crescita personale.

Abbiamo visto come alcuni linguaggi tra i più utilizzati in assoluto sono stati messi in secondo piano con la nascita di nuove tecnologie che ad oggi stanno ancora crescendo. Si pensi al

"moderno" Python che è arrivato negli ultimi anni ad essere più utilizzato del più famoso e già ampiamente affermato Java. Anche questo dato ritengo che sia rilevante perchè ci fa capire che, evidentemente, certi linguaggi nel corso degli anni sono risultati più potenti, versatili e semplici tanto da essere preferiti in aziende e luoghi di lavoro rispetto alla controparte più datata. Questo dato, insieme a quello delle "tecnologie più ricercate" che vede lo stesso Python al primo posto, fa pensare che forse è necessario un rinnovo nel piano di studi di molte Università che al giorno d'oggi ancora non propongono corsi su determinate tecnologie. Le occupazioni dei programmatori sul lavoro sono varie e ci si sta specializzando in sempre più ambiti. Si pensi ai soli sviluppatori back-end e front-end che negli ultimi anni hanno più che raddoppiato le percentuali, arrivando a pareggiare lo sviluppatore web completo. Anche gli sviluppatori mobile, pur rimanendo in minoranza dal 2015 ad oggi si sono raddoppiati. Questo può essere un buon punto di partenza per chi decide di intraprendere un percorso in questo settore. L'ambito ICT offre numerosi sbocchi lavorativi e anche se durante il percorso di studi si ha modo di capire cosa piace di più e cosa meno è da tenere in considerazione che la metà dei programmatori si occupano in generale del web il che potrebbe essere un incentivo, per i nuovi arrivati, a specializzarsi in altri campi in modo da trovare meno "concorrenza" nella ricerca successiva di un'occupazione.

Abbiamo visto che anche nel mondo dei programmatori, oltre che a quello dell'utilizzatore medio del computer, si preferisce usare il sistema Windows rispetto a Linux e MacOS.

Infine abbiamo visto come era la situazione del lavoro da remoto prima che l'emergenza Covid-19 sdoganasse questa soluzione e si è osservato che lo smart working è sempre stato presente ma veniva praticato di rado o mai. Erano pochi, insomma, quelli che in passato hanno lavorato a tempo pieno o part-time da casa. Seppur non sono stati trovati dati inerenti agli anni dal 2020 a seguire, è lecito pensare che l'utilizzo di questa pratica sia ben più che utilizzata rispetto al passato il che rende il lavoro in questo campo forse "più comodo" perché può aprire più strade agli aspiranti programmatori in quanto si può facilmente trovare lavoro all'estero o fuori dalla propria regione di residenza, si può continuare a dare il proprio contributo se contagiati e per qualunque tipo di inconveniente si può essere presenti pur non essendoci fisicamente.

Le domande che gli sviluppatori pongono su Stack Overflow sono di ogni genere ed entità: possono riguardare problematiche riguardanti configurazioni, incompatibilità di versioni, problemi logici specifici o anche di carattere generale e molto altro. Le tecnologie di programmazione sono in continua evoluzione: ciò che può essere vero oggi potrebbe risultare obsoleto in futuro, in più c'è da dire che uno stesso problema può essere causato da dina-

miche differenti. Pertanto, si ritiene, che lo sviluppo di un "suggeritore automatico" come implementato da qualcuno (discusso nel cap 2.2) non è un'idea che va a migliorare la vita del programmatore dato che la soluzione trovata per uno, non necessariamente può andare bene per l'altro. E' fondamentale, invece, che l'utente si cerchi una soluzione da solo, che analizzi come un altro utente sia arrivato ad un determinato problema, come l'abbia risolto e cosa ne pensa chi ha risposto al post senza dare la soluzione. Attraverso ricerche incrociate di questo tipo e basandosi sui problemi degli altri si può arrivare a risolvere il proprio. Si potrebbe pensare, in questa prospettiva, ad un software basato su una struttura ad albero in grado di suggerire una serie di post basandosi sulle risposte che l'utente fornisce a determinate domande a risposta multipla che il programma propone, domande fatte ad hoc per poter escludere determinati rami e indirizzarsi quanto più possibile verso la soluzione. Discorso diverso può essere fatto con problemi basilari come la sintassi, la logica, le librerie ecc. A quel punto sarebbe buona prassi prevedere una categoria dedicata sul sito dove il principiante o lo studente può reperire informazioni senza troppi indugi.

5.2 **Sviluppi futuri**

Si può affermare che lo studio ha portato alla luce informazioni interessanti che potranno sicuramente lasciare spunti per il futuro. In primo luogo si potrà continuare, nei prossimi anni, a tenere d'occhio i survey e in particolare le categorie prese in considerazione per questo lavoro di tesi, quindi continuare a seguire l'evoluzione dell'utenza. Si potrà, poi, pensare di osservare la community in base alla posizione geografica e non in modo universale come fatto in questo studio per capire cosa accomuna e in cosa differiscono i vari programmatori del mondo. Si potrà pensare di estendere questo studio osservando anche la qualità della vita di chi lavora in ambito ICT, per esempio indagando sugli orari lavorativi, sui guadagni medi annuali o sulla difficoltà a trovare impiego o meno.

Ciò significa che, da un lato, bisognerebbe aspettare alcuni anni per poter consultare i prossimi survey e continuare a seguire l'evoluzione ma dall'altro lato già si potrebbe cominciare a studiare le differenze tra diverse regioni del mondo e la vita dei lavoratori informatici.

Ringraziamenti

Ringrazio i miei genitori e mio fratello che, nonostante i mille fallimenti, non hanno mai smesso di credere in me, mi hanno sempre sostenuto, hanno sempre desiderato il meglio per me e non mi hanno mai fatto mancare niente.

Ringrazio nonna Annamaria, la donna che più si è entusiasmata quando ho iniziato questo percorso, quella che ha fatto i salti di gioia per ogni esame passato, quella che mi ha sempre offerto un posto in cui rifugiarmi e quella che mi ha aiutato in ogni situazione.

Ringrazio il mio amico Vincenzo, l'amico con la quale sono cresciuto, per essermi stato affianco fin dall'infanzia, per essermi stato vicino in qualunque momento, per aver sempre condiviso con me gioie e dolori.

Ringrazio tutti i miei compagni di corso che mi hanno aiutato ad affrontare difficoltà, ansie e paure. Per avermi aiutato nello studio e per i momenti che abbiamo passato insieme dentro e fuori l'università.

Inoltre, è doveroso ringraziare le persone che hanno contribuito, con il loro supporto, alla realizzazione di questa tesi: i miei relatori, il professore Fabio Palomba e il dottor Giammaria Giordano che con grande disponibilità hanno sempre seguito gli aggiornamenti e suggerito le giuste modifiche e integrazioni.

Questo traguardo lo dedico a tutti loro ma soprattutto a me stesso che sono riuscito, finalmente, a dedicare impegno e dedizione costante in qualcosa.

In particolar modo dedico questo traguardo al nonno Alberto, la persona più simpatica che abbia conosciuto in vita mia e, probabilmente, la persona con la quale più avrei desiderato condividere questo momento e che più avrei voluto rendere fiero ed orgoglioso di me.

Ad maiora.

Bibliografia

- [1] Wikipedia, “Stack overflow (sito) — wikipedia, l’enciclopedia libera,” 2022. [Online; in data 21-settembre-2022]. (Citato a pagina 4)
- [2] Aulab, “Cos’è un linguaggio di programmazione,” 2021. (Citato a pagina 7)
- [3] M. Moroni, “I 12 linguaggi di programmazione più usati nel 2022 (e consigli su quale imparare),” 2022. (Citato a pagina 8)
- [4] G. Greison, “Storia di ada lovelace, la donna che per prima immaginò il potere del computer,” 2018. (Citato a pagina 8)
- [5] F. Ciacchi, “Storia della programmazione,” 2006. (Citato a pagina 10)
- [6] AGCOM, “L’informazione alla prova dei giovani,” 2020. (Citato a pagina 11)
- [7] L. Pica, “Donne e ict: perché le ragazze non entrano nel mondo della tecnologia,” 2009. (Citato a pagina 12)
- [8] G. Succi, “L’evoluzione dei linguaggi di programmazione: analisi e prospettive,” *Mondo Digitale*, vol. 4, 2003. (Citato a pagina 13)
- [9] L. Ponzanelli, G. Bavota, M. Di Penta, R. Oliveto, and M. Lanza, “Mining stackoverflow to turn the ide into a self-confident programming prompter,” in *Proceedings of the 11th working conference on mining software repositories*, pp. 102–111, 2014. (Citato a pagina 15)
- [10] M. Liu, X. Peng, Q. Jiang, A. Marcus, J. Yang, and W. Zhao, “Searching stackoverflow questions with multi-faceted categorization,” in *Proceedings of the Tenth Asia-Pacific Symposium on Internetware*, pp. 1–10, 2018. (Citato a pagina 15)

- [11] B. Vasilescu, A. Capiluppi, and A. Serebrenik, "Gender, representation and online participation: A quantitative study of stackoverflow," in *2012 International Conference on Social Informatics*, pp. 332–338, IEEE, 2012. (Citato alle pagine 15 e 16)
- [12] C. Rosen and E. Shihab, "What are mobile developers asking about? a large scale study using stack overflow," *Empirical Software Engineering*, vol. 21, no. 3, pp. 1192–1223, 2016. (Citato alle pagine 16 e 18)
- [13] I. Moutidis and H. T. Williams, "Community evolution on stack overflow," *Plos one*, vol. 16, no. 6, p. e0253010, 2021. (Citato a pagina 17)
- [14] P. Morrison and E. Murphy-Hill, "Is programming knowledge related to age? an exploration of stack overflow," in *2013 10th Working Conference on Mining Software Repositories (MSR)*, pp. 69–72, IEEE, 2013. (Citato a pagina 18)