|  |
| --- |
| Itt “B. Pascal” |
| La comunicazione |
| Scambio di informazioni veicolato attraverso Internet |
|  |

Alessandro Lombardini Anno scolastico 2016/2017

**INTRODUZIONE**

Oggi viviamo in un mondo nuovo che è cambiato e che continuerà a cambiare negli anni a venire. Si è assistito all’avvento dell’informatica, del digitale e di Internet, che hanno contribuito a veicolare in maniera differente la comunicazione, intesa come scambio di informazioni.

Tra gli autori della letteratura contemporanea spicca la figura di Saviano, il quale si serve di Internet per comunicare ai giovani e parlare loro della mafia e dei suoi effetti sulla società. Nella *Paranza dei bambini,* uno dei suoi celebri romanzi, le piattaforme web vengono descritte come palestre virtuali nelle quali si impara la violenza.

L’uso di Internet implica la condivisione di una Netiquette, un complesso di regole di comportamento volte a favorire il reciproco rispetto tra gli utenti durante la comunicazione. La noncuranza di tali principi favorisce l’ incorrere del fenomeno del cyberbullismo, dilagante tra i giovani.

Internet ha aperto infinite possibilità: le persone sono in grado di effettuare acquisti on-line, lavorare a distanza, condurre transazioni finanziarie. Le reti digitali sono tuttavia intrinsecamente insicure, in quanto non sono state progettate per garantire la difesa contro eventuali abusi; la crittografia rappresenta la soluzione a tale problema.

Con la definizione del protocollo HTTP si è rivoluzionato profondamente il modo di effettuare le ricerche e di comunicare in rete. La facilità d’uso che ne è risultata, assieme ad una vasta diffusione dei computer anche per uso personale, hanno aperto l’uso di internet a milioni di persone, cambiando radicalmente la società.

QuickTreno, infine, è un programma che, interfacciandosi all’utente tramite un’applicazione di messaggistica istantanea, permette un accesso alle informazioni inerenti ai treni della nota compagnia Trenitalia; esso è il risultato delle competenze acquisite nel corso degli studi.

SOMMARIO

[1.0 LA COMUNICAZIONE OLTRE LA MAFIA: ROBERTO SAVIANO 5](#_Toc485649996)

[La paranza dei bambini 8](#_Toc485649997)

[2.0 NETIQUETTE AND CYBERBULLYING 10](#_Toc485649998)

[2.1 Social Network 10](#_Toc485649999)

[2.2 Netiquette 10](#_Toc485650000)

[2.3 Cyberbullying 11](#_Toc485650001)

[3.0 SICUREZZA DELLE COMUNICAZIONI: CRITTOGRAFIA 14](#_Toc485650002)

[3.1 Crittografia simmetrica 15](#_Toc485650003)

[3.2 Crittografia asimmetrica 16](#_Toc485650004)

[3.3 Crittografia ibrida 18](#_Toc485650005)

[3.4 Firma digitale 19](#_Toc485650006)

[3.5 Certificati digitali 20](#_Toc485650007)

[4.0 PROTOCOLLI PER LA TRASMISSIONE DI INFORMAZIONI: HTTP 22](#_Toc485650008)

[4.1 Storia 22](#_Toc485650009)

[4.2 Connessione 22](#_Toc485650010)

[4.3 Messaggi 25](#_Toc485650011)

[4.4 Metodi della richiesta 27](#_Toc485650012)

[4.5 Codici di stato 29](#_Toc485650013)

[4.6 Intestazioni 30](#_Toc485650014)

[4.7 Cookies 32](#_Toc485650015)

[4.8 HTTPS 33](#_Toc485650016)

5.0 [QuickTreno 34](#_Toc485650017)

[Presentazione 35](#_Toc485650018)

[5.1 Tecnologie in uso 36](#_Toc485650019)

[5.2 Bot 37](#_Toc485650020)

[5.3 Interfaccia utente 39](#_Toc485650021)

[5.5 Progettazione 43](#_Toc485650022)

[5.7 Sviluppo 47](#_Toc485650023)

[5.8 Esempio 53](#_Toc485650024)

# 1.0 LA COMUNICAZIONE OLTRE LA MAFIA: ROBERTO SAVIANO



Roberto Saviano è uno scrittore e saggista italiano nato a Napoli nel 1979. Figlio del medico Luigi Saviano e di Miriam Haftar, ebrea di origine sefardita, vive la sua infanzia ed adolescenza a Caserta dove consegue la maturità scientifica presso il Liceo Scientifico Statale “Armando Diaz”. Successivamente si trasferisce a Napoli per frequentare la facoltà di Filosofia all’Università degli Studi di Napoli Federico II. Nel 2002 inizia la sua carriera di giornalista scrivendo per riviste e quotidiani tra i quali *Pulp, Sud Diario, Il Manifesto*, per il sito web *Nazione Indiana* e per l’osservatorio sulla camorra del *Corriere del Mezzogiorno*.

Nel 2009, alla presenza di [Dario Fo](https://it.wikipedia.org/wiki/Dario_Fo), riceve il titolo di Socio Onorario dell'[Accademia di Brera](https://it.wikipedia.org/wiki/Accademia_di_Brera) e il Diploma di Secondo Livello in Comunicazione e Didattica dell'Arte [honoris causa](https://it.wikipedia.org/wiki/Honoris_causa). La laurea honoris causa in Giurisprudenza giunge nel 2011 "per l'importante contributo alla lotta contro la criminalità e alla difesa del principio di legalità nel nostro Paese".

Nel Marzo 2006 pubblica il suo primo romanzo *Gomorra*, edito da Mondadori; tradotto in oltre 50 paesi e con 10 milioni di copie vendute, diviene nel 2017 un best-seller internazionale.

In *Gomorra, viaggio nell’impero economico e nel segno di dominio della camorra,* Saviano si mostra fortemente accusatorio nei confronti di una organizzazione affaristica con ramificazioni su tutto il pianeta, la cui forza negli anni, è sempre stata quella di godere del silenzio. Per l’autore inizia però il periodo delle lettere minatorie, di telefonate mute, ma anche della scorta con isolamento ambientale. È il 2008 quando un ispettore di Polizia della DIA di Milano viene a conoscenza dal pentito Carmine Schiavone di un piano per uccidere lo scrittore; Saviano decide così di lasciare l’Italia scrivendo: “[…]E voglio ancora scrivere, scrivere, scrivere perché è quella la mia passione e la mia resistenza e io, per scrivere, ho bisogno di affondare le mani nella realtà, strofinarmela addosso […]”.

Il 20 ottobre 2008, sei premi Nobel – Dario Fo, Michail Gorbačëv, Günter Grass, Rita Levi-Montalcini, Orhan Pamuk e Desmond Tutu, si mobilitano chiedendo che lo stato italiano faccia qualsiasi sforzo per proteggere Saviano e sconfiggere la camorra e a novembre dello stesso anno lo scrittore viene invitato in Svezia a tenere un discorso sulla libertà di espressione all’Accademia dei Nobel.

Oltre a *Gomorra*, i romanzi *ZeroZeroZero* e *La paranza dei bambini* di Saviano si affermano come due grandi successi. Da citare inoltre due racconti, *Il contrario* e *Super Santos* e i saggi *La bellezza e l’inferno* e *La parola contro la camorra*.

Attualmente l’autore mantiene numerose collaborazioni: in Italia con L’Espresso e La Repubblica, negli Stati Uniti con il Washinton Post, il New York Times, Newsweek e il Time, in Spagna con El Pais, in Germania con Die Zeit e Der Spiegel, in Svezia con Expressen e in Gran Bretagna con il Guardian e il Times.

Interrogato su come i nuovi media lo aiutano a vincere l'isolamento causato dal programma di protezione a cui è sottoposto, e in che modo sfrutta la tecnologia per declinare la sue attività editoriali, lo scrittore napoletano ha spiegato che Internet e i social network gli hanno permesso di vivere un'esperienza senza precedenti: “Anche fuori dall'Italia, è impossibile trovare due milioni di fan che seguono un autore che non si occupa di fiction. Ma c'è molto di più. Il contatto costante con loro, anche con i detrattori, mi nutre, genera un confronto che mi aiuta costantemente a capire come si stanno evolvendo le varie forme di comunicazione”.

Per Saviano non esiste più alcuna struttura isolata quindi, tutto è connesso. La forza della parola e la potenza di Internet, dunque, diventano gli ingredienti del suo modo di comunicare: “E all'origine di tutto c'è la parola, dalla quale poi possono svilupparsi molti altri formati” e ancora *“Internet significa scardinare le pareti di questa stessa stanza, significa togliere i limiti al fisico, alla carne, togliere spessore al corpo per arrivare agli occhi e alle orecchie di qualunque lettore in un modo che altrimenti mi sarebbe impossibile”.*

Saviano è un esempio per ciò che riguarda l’assumersi le proprie responsabilità. Nonostante gli sia stata negata un bene prezioso come la libertà, l’autore non si è mai sottratto alle conseguenze di ciò che ha scritto e che ha dichiarato. Convinto del suo fine e della sua missione di scrittore, Saviano in questi anni si è fatto portavoce del valore inestimabile della parola e con un pubblico di giovani e giovanissimi, ha calibrato i suoi discorsi per muovere le coscienze.

La parola deve avere un senso per lo scrittore, che dice “Se scrivo di Primo Levi o di Anna Politkovskaya, se parlo di Šalamov o di “No, i giorni dell’arcobaleno” in televisione, lo faccio perché questi argomenti sono me. Hanno contribuito e contribuiscono ad alimentare la mia vita e mi aiutano a comprendere ciò che vivo, che vedo, ciò che mi piace e ciò che mi disgusta”.

Scritta o pronunciata, la parola assume obbligatoriamente un valore e diventa quindi impossibile cancellarla. Se è vero da un lato che “Oggi l’informazione è continuamente aggiornata e tutto perde il carattere dell’essenzialità, tutto può essere sostituito, contraddetto, smentito dopo poco”, d’altro canto le reazioni e le suggestioni create, non possono svanire nel nulla.

Saviano negli ultimi periodi interviene spesso in trasmissioni televisive che sono perlopiù seguite da un pubblico giovane. Il suo intento principale risulta proprio quello di parlare loro dell’importanza della comunicazione e del male che essa può provocare se utilizzata istintivamente e senza riflessioni, "Il messaggio arriva subito, la viralità non prevede ragionamento, per il ragionamento ci vuole tempo”.

Nell’ euforia di una comunicazione che non costa nulla (vedi Facebook, Twitter e tutti gli altri social networks), scrivere è troppo spesso solo, o poco più, un atto di presenza. L’autore però desidera ardentemente rendere consapevoli che quella presenza può decidere che persona essere e dice in un’intervista: “Bisogna pretendere da se stessi, e da chi partecipa alle discussioni, un apporto costruttivo”.

In un’intervista su “La Stampa”, il giornalista formula a Saviano la domanda “La cronaca ci ha purtroppo regalato parole nuove come cyber bullismo e le violenze vengono sempre più spesso filmate e messe in rete. Come si educa una generazione che trova tutto questo quasi naturale?” e l’autore risponde dicendo: “Insegnando l’empatia. A gioire o soffrire se il prossimo è felice o infelice. A non fargli ciò che non vorresti fosse fatto a te. A guardarlo in faccia mentre soffre, solo così si diviene empatici, cosa impossibile se tutto è limitato alla relazione virtuale”.

L’intervistatore continua domandandogli se per lui i social hanno reso il mondo migliore, peggiore o se banalmente abbia amplificato l’esistente. L’autore napoletano risponde: “Banalmente hanno amplificato l’esistente. C’è una cosa che però deve essere chiara. Sui social il simpatico diventa simpaticissimo, l’antipatico diventa insopportabilmente sarcastico, il colto diventa pesante e spocchioso, l’ignorante diventa illeggibile. Nessuno è davvero se stesso”.

Saviano si presenta in prima serata su reti televisive con un grido “Ragazzi, basta bullismo: in rete le parole sono pericolose”.

Online tutti i giorni, Saviano è ben consapevole del livello di scontro spaventoso sui social, di una violenza che non si userebbe mai di persona: “Siccome si può cancellare tutto, anche la tua morale diventa on line, non va mai off line, non senti di essere responsabile. Pensi: 'Se ho esagerato lo cancello’, ma non si cancellano l'offesa ricevuta e il dolore inflitto. C'è un pensiero della tradizione ebraica che dice: 'Quando la parola non l'hai pronunciata sei tu a comandarla, ma una volta che l'hai pronunciata è lei che comanda te'".

E Roberto Saviano, una tra le persone più influenti del nostro tempo, scrittore diretto e portatore di verità, intende insegnarci come si fa ad imparare il valore delle parole:

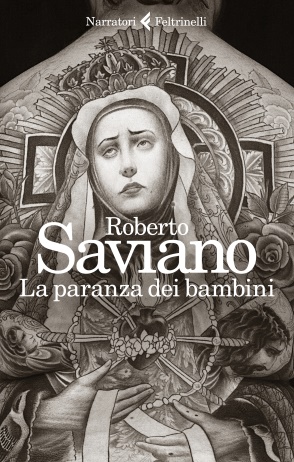
"Guardando chi ha pagato per ciò che ha scritto. Quando capisci quale sia stato il prezzo per pronunciare parole libere, capisci il valore da dare alle parole. Si può partire da lontano”.

Questo autore ha pagato per ciò che ha scritto ma, nonostante tutto, continua a crederci.

## La paranza dei bambini

Ho incontrato Roberto Saviano nel corrente anno scolastico per la presentazione del suo ultimo romanzo “La paranza dei bambini”,pubblicato nel [2016](https://it.wikipedia.org/wiki/2016) da [Feltrinelli](https://it.wikipedia.org/wiki/Giangiacomo_Feltrinelli_Editore), pesantemente ispirato alla realtà della camorra napoletana degli [anni 2010](https://it.wikipedia.org/wiki/Anni_2010).

«E ti pare che io mi metto paura di un bambino come te? Io per diventare bambino c’ho messo dieci anni, per spararti in faccia ci metto un secondo.»

Nella descrizione sul sito della casa editrice troviamo “*La paranza dei bambini* narra la controversa ascesa di una paranza – un gruppo di fuoco legato alla Camorra – e del suo capo, il giovane Nicolas Fiorillo. Appollaiati sui tetti della città, imparano a sparare con pistole semiautomatiche e AK-47 mirando alle parabole e alle antenne, poi scendono per le strade a seminare il terrore in sella ai loro scooter. A poco a poco ottengono il controllo dei quartieri, sottraendoli alle paranze avversarie, stringendo alleanze con vecchi boss in declino.  
Paranza è un nome che viene dal mare, nome di barche che vanno a caccia di pesci da ingannare con la luce. E come nella pesca a strascico la paranza va a pescare persone da ammazzare, qui si racconta di ragazzini guizzanti di vita come pesci, di adolescenze “ingannate dalla luce”, e di morti che producono morti.”.

È il primo romanzo interamente di finzione, ma ispirato a fatti veri. Viene descritta la parte della gioventù attratta dal denaro facile e dal potere, che pur di arrivare ai propri obiettivi opta per il crimine, la violenza e la sopraffazione come scelta di vita, pur sapendo che morirà.

In questo romanzo emerge in modo particolare la comunicazione virtuale. Essa diviene il tramite per mantenere la segretezza tra i personaggi, attraverso un linguaggio in codice. Pagina dopo pagina, l’uso dei social e della messaggistica istantanea diventano, tramite l’uso compulsivo dello smartphone, il fulcro del sistema di comunicazione della paranza.

WhatsApp è uno strumento strategico: il gruppo fisico è coincidente con il gruppo virtuale.

Twitter è poco frequentato, ma diviene utile per conoscere le notizie del mondo di fuori. Su Instagram, invece, persino i narcos, i mafiosi e i camorristi postano le immagini di un benessere esagerato e senza rispetto della povertà altrui.

Il vero allenamento virtuale per affrontare la realtà è su Youtube, un archivio senza fine di modelli da imitare. È lì che i bambini imparano la violenza; la conoscenza non si apprende in questo caso dai libri o dall’esperienza pratica. Tutto è a portata di mano, basta un click per ottenere una risposta.

Sembra quasi un paradosso.

Lo stesso Internet che Roberto Saviano usa quotidianamente da anni per parlare della mafia e per veicolare la sua intenzione di combatterla, è lo stesso che i bambini del romanzo utilizzano per imparare ad essere mafiosi. Loro si nutrono di Internet per imparare ad essere violenti.

# 2.0 NETIQUETTE AND CYBERBULLYING

The Internet is a worldwide collection of computer networks, cooperating with each other to exchange data, developed to help scientists share ideas.

Since the invention of the World Wide Web (www) in 1989 by Tim Berners-Lee, the Internet use has grown rapidly and nowadays almost everyone can connect to that.

Moreover people can post and update information, and access all kinds of multimedia information by their own device.

## 2.1 Social Network



To communicate online, many people use social networking, an online platform that is used by people to build [social networks](https://en.wikipedia.org/wiki/Social_network) or [social relations](https://en.wikipedia.org/wiki/Social_relation) with friends and people with same interests.

It was first proposed in early development of World Wide Web and through years it has become a great way to keep in touch since it is easy for users to get in touch with each other.

## 2.2 Netiquette

When using social networking sites, people should always try to be polite and respectful by following some rules of good behaviours that are called ‘netiquette’.

Netiquette comes from the two words network and etiquette.

It governs what conduct is socially acceptable in an online or digital situation, a kind of a social code of network communication, like avoid writing in capital letters, because it is a form of screaming, and don’t use abusive, racist and offensive words.

It is often combined with the concept of a ‘netizen’ witch itself is a contraction of the words Internet and citizen and refers to a person who uses the Internet to participate in society online and an individual who has accepted the responsibility of using the internet in productive and socially responsible ways. This term has been used as early as 1998 and has gone through several changes in description as newer technological advances have changed the method and frequency of how people interact with each other.

Unfortunately, not all Internet users follow netiquette.

## 2.3 Cyberbullying

There are often no regulations or check on the material that is placed on the Internet and many people find ways to irritate, embarrass or take advantage of other online users. Bullies have been around forever, but technology now gives them a whole new platform for their actions.

Cyberbullying is the use of technology to harass, threaten, embarrass, or target another person. It is a serious problem, it is ‘real’ as any other type of bullying; sometimes even worse. If someone is bullying you at school, when you leave for the day it’s over. But cyberbullyng can follow you home, and continue all day. Being bullied at home can take away the place teeanagers feel most safe.



The methods used are limited only by the child’s imagination and access to technology. Examples of cyberbullying include mean text messages or emails, rumors sent by email or posted on social networking sites, and embarrassing pictures, videos, websites, or fake profiles. Social media can be used for positive activities, like connecting kids with friends and family, helping students with school, and for entertainment but, at the same time, can also be used to hurt other people.

Communicating with others via the Internet without misunderstanding in the heat of the moment can be challenging, the impersonal nature of text messages and emails make it very hard to detect the sender’ tone so it also can happen accidentally.

Whether done in person or through technology, the effects of bullying are similar. Sever, long-term, or frequent cyberbullying can leave both victims and bullies at greater risk of anxiety, depression, and other stress-related disorders. Kids who are cyberbullied are more likely to use alcohol and drugs, have lower self-esteem and have more health problems.

Uploading and downloading videos, pictures or music of other people without having the authorization to do so, is a violation of human rights and privacy laws, and you can be prosecuted and is very important to keep this concept in mind. Because the use of online communication has grown so rapidly and the crime is new, many governments and jurisdictions are deliberating over cyberbullying laws.

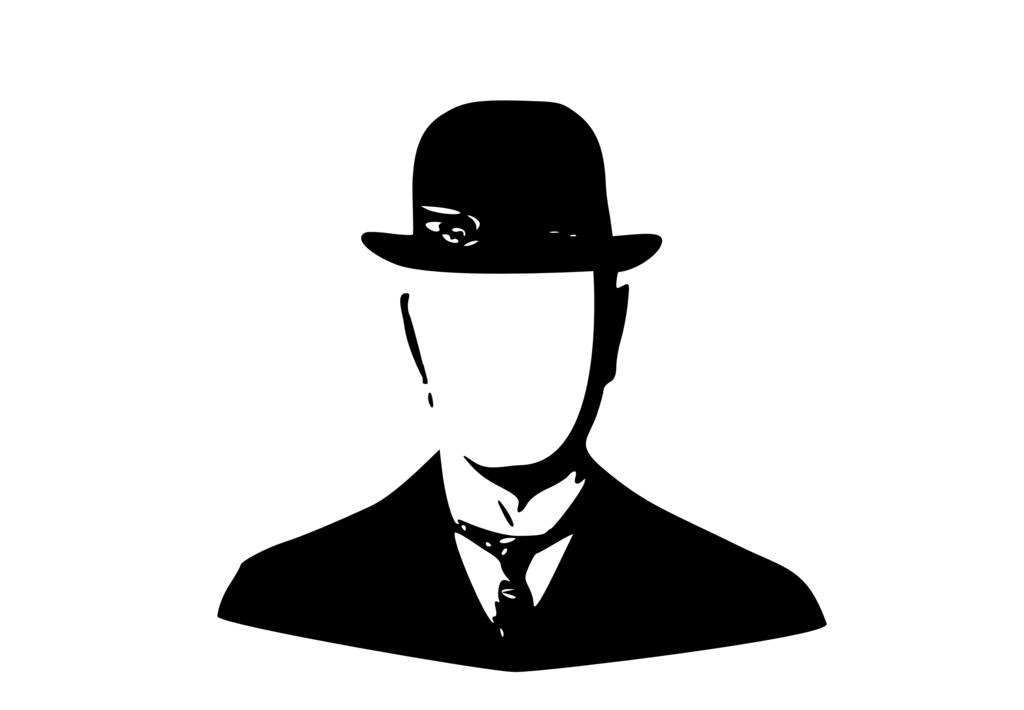
The Italian parliament has recently approved long-awaited legislation to address cyberbullying, to prevent users of the Internet and social platforms from abusing a minor.

The law, definitively voted on May 17 by the Chamber of Deputies with only one abstention, came after nearly three years of parliamentary debate in response to several cases of suicide of victims of online bullying.  
The Italian law follows similar legislative initiatives in other countries.

In the United States, only 23 countries have included cyber abuse in their bullying laws and 18 states punish it with criminal sanctions, according to cyberbullying.org.  School policy is required by all but one US state.

In Italy, schools will have to appoint a teacher to lead initiatives against cyberbullying.

Moreover, minors above 14 years of age who have been victims of onlineabuse, as well as their parents, can ask website hosts and social platforms to remove and block abusive content within 48 hours**.**



Especially young people don’t always realize how dangerous it is to put other personal information online and how much it can hurt people’s feelings and damage their self-esteem. The problem is compounded by the fact that a bully can hide behind a pseudonymous user name, disguising his or her true identity.

There is a sense of disinhibition in which people might feel more empowered to behave in any way with no consequences, an invisible veil of anonymity allows and excuses a bad behaviour. For this reason I think it is very important to create first among young people a culture of empathy to fight this phenomenon.

The Internet has profoundly changed the way we live, communicate and access information. It is so powerful that it can either build or destroy human relationships. We should create a respectful atmosphere and always remember that on the other end of the line there is someone else just like us.

# 

# 3.0 SICUREZZA DELLE COMUNICAZIONI: CRITTOGRAFIA

Con lo sviluppo del commercio elettronico, l’aumento dello scambio di dati in formato telematico e la necessità di effettuare pagamenti in formato elettronico, il problema della sicurezza nella circolazione delle informazioni in rete è diventato di notevole importanza. Questo problema ha trovato soluzione nella crittografia, una scienza molto antica e strettamente legata all’esigenza di inviare messaggi riservati in tempo di guerra.

Il termine “crittografia” deriva dal greco “krypto-s” che significa “nascosto” e “graphia” che significa “scrittura”. Essa non mira a nascondere il messaggio in se, ma il suo significato.

La crittografia consiste in un insieme di tecniche che consentono di trasformare un messaggio in un testo che risulti incomprensibile per chiunque non ne conosca la chiave.

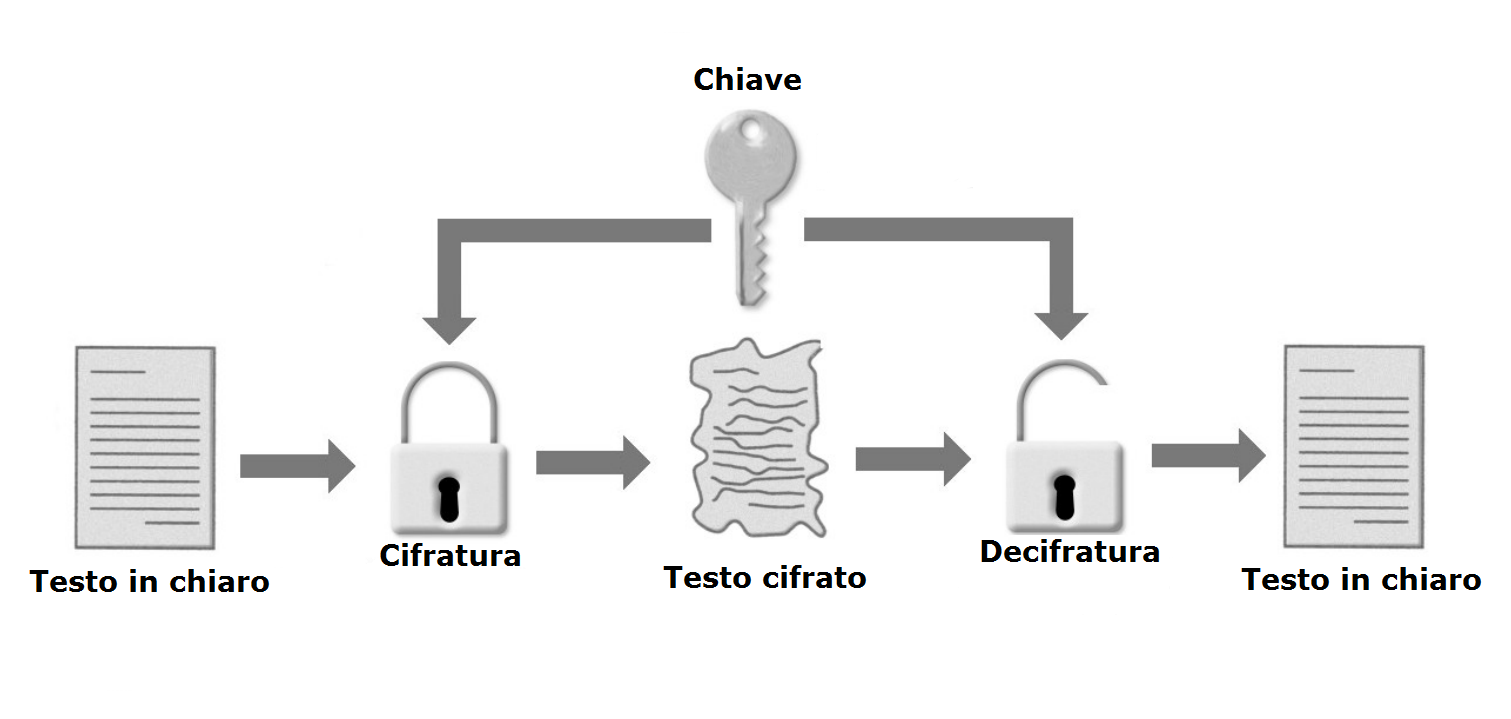
Il messaggio da proteggere viene detto *in chiaro* mentre quello ‘trasformato’ viene detto *cifrato*. L’operazione di trasformazione da testo in chiaro a testo cifrato si chiama *cifratura*, mentre *decifratura* è detta la trasformazione inversa. Il meccanismo della criptografia si compone di due elementi: l’algoritmo, l’insieme dei passi da seguire per modificare il messaggio secondo delle regole definite e la chiave, una sequenza di cifre e caratteri applicati all’algoritmo per ottenere un testo cifrato. La sicurezza non dipende tanto dall’algoritmo, il quale potrebbe anche essere reso pubblico, ma quanto dalla segretezza della chiave. La dimensione della chiave, generalmente misurata in bit, dipende dal particolare algoritmo usato. Alcuni algoritmi utilizzano chiavi di lunghezze diverse e in questi casi, più lunga è la chiave tanto più difficile sarà forzare il messaggio cifrato.

Le tecniche crittografiche possono essere classificate in due principali sistemi:

* Criptografia a chiave simmetrica;
* Criptografia a chiave asimmetrica.

## 3.1 Crittografia simmetrica

La crittografia tradizionale prevede l’uso della medesima chiave tanto nella fase di codifica quanto in quella di decodifica, per questo motivo si parla di crittografia simmetrica.



Il messaggio da inviare M, sottoposto ad un algoritmo di crittografia simmetrica C con chiave k, restituisce il messaggio cifrato m.

C(M,k) = m

Se il messaggio ricevuto m viene sottoposto ad un algoritmo di decrittazione D con chiave k, si ottiene il messaggio iniziale M.

D(m,k) = M

Due interlocutori, al fine di comunicare tra loro, devono seguire il seguente paradigma:

1. Scegliere un algoritmo e concordare un chiave comune;
2. Individuare un canale di comunicazione sicuro per lo scambio della chiave, che verrà poi effettuato.

In seguito allo scambio è poi necessario prestare la massima attenzione e cautela affinché nessun altro venga a conoscenza della chiave.

Il principio di funzionamento dello scambio dei messaggi prevede tre semplici passi:

1. Il mittente cifra il messaggio con la chiave segreta;
2. Si trasmette il messaggio cifrato attraverso un canale di comunicazione;
3. Il destinatario riceve il messaggio cifrato e lo decifra con la chiave segreta.

Questo modello di criptografia presenta evidentemente alcuni notevoli inconvenienti legati allo scambio e alla gestione delle chiavi:

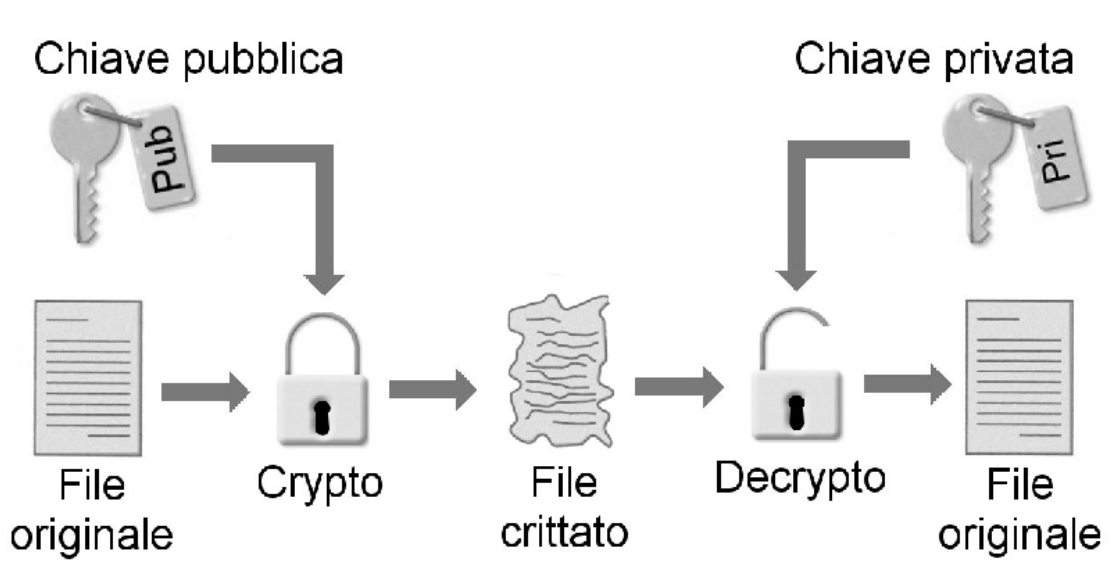
* Mittente e destinatario devono conoscere la stessa chiave che diventa fissa e facilmente individuabile a scapito della sicurezza;
* E’ necessario un numero molto elevato di chiavi: dato un sistema di n persone che devono comunicare tra loro, sono necessarie n(n-1)/2 chiavi.
* La distribuzione delle chiavi.

Il problema principale di questo metodo è infatti la distribuzione delle chiavi: se due interlocutori vogliono utilizzare questo tipo di approccio è necessario che la trasmissione della chiave criptografica avvenga preliminarmente all’invio del messaggio e soprattutto attraverso un canale sicuro. Dato che il canale utilizzato per la trasmissione non è sicuro (altrimenti potrebbe essere utilizzato per inviare il messaggio evitando di crittografarlo) non può essere utilizzato per lo scambio; questo problema è stato risolto con l’invenzione della crittografia a chiave pubblica. Tale metodo tuttavia presenta il vantaggio della velocità per quel che concerne sia la generazione della chiave che le operazioni di cifratura e decifratura.

Tra i vari algoritmi a chiave privata, quello che ha ottenuto maggiore sviluppo e successo è stato il DES (Data Encryption Standard). Poiché le tecnologie si evolvono, sono stati presi in esame vari candidati a diventare il nuovo standard di crittografia simmetrica; nel 2000 AES (Advanced Encryption Standard) è divenuto il nuovo standard. L’attuale standard presenta chiavi da 128 bit, contro i 56 del DES e la possibilità di usarne da 192 e 256 bit per situazioni particolarmente importanti.

## 3.2 Crittografia asimmetrica

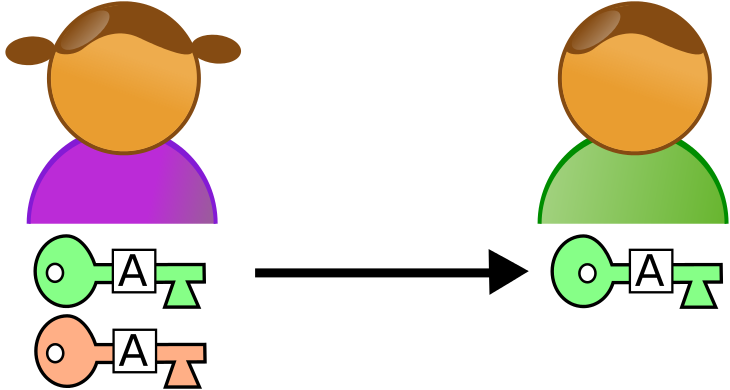
Con questo tipo di crittografia ogni utente ha a disposizione due chiavi, diverse ma complementari: una pubblica da distribuire a tutti quelli con cui vuole comunicare, ed una privata da tenere segreta. Ogni chiave mette in chiaro il messaggio che l’altra ha codificato, ma il processo non è reversibile: ciò che viene cifrato con la chiave pubblica può essere decifrato solo con la chiave privata corrispondente. Per comunicare in modo sicuro con un utente basta cifrare il messaggio con la sua chiave pubblica; in questo modo viene superato il problema di comunicare la chiave segretamente in quanto questa è nota a tutti.



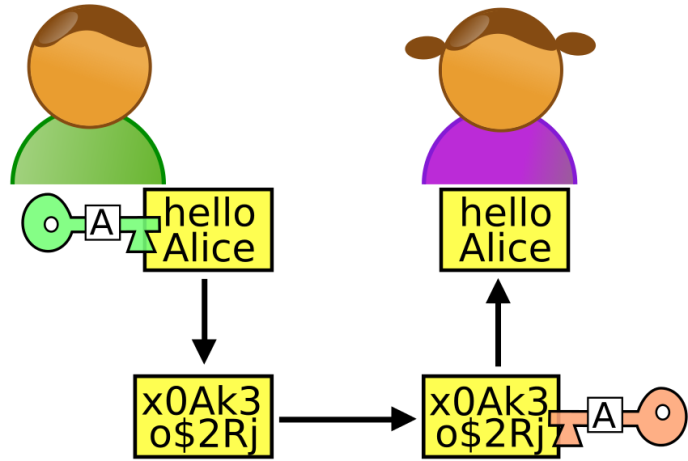
Il modello su cui si basa un generico algoritmo a chiave pubblica può descritto come di seguito:

1. Condivisione dello stesso algoritmo;
2. Scambio della chiave pubblica;
3. Cifratura del messaggio in chiaro utilizzando la chiave pubblica;
4. Invio del messaggio;
5. Decifratura del messaggio utilizzando la propria chiave privata.

Di seguito viene proposto un esempio: Se Alice vuole ricevere un messaggio segreto da Bob, si preoccupa di fornirgli la sua chiave pubblica.



Bob utilizzerà la chiave pubblica per cifrare il messaggio che desidera inviare ad Alice. Bob invierà ad Alice il messaggio che chiunque potrà vedere ma che nessuno potrà leggere in quanto, essendo Alice l’unica ad avere la chiave privata, è l’unica in grado di decifrarlo.



Uno dei principali limiti nell’utilizzo degli algoritmi a crittografia asimmetrica consiste nei numerosi e complessi calcoli che rendono la loro implementazione poco efficiente.

Attualmente l’algoritmo più diffuso e utilizzato è RSA. Le tipiche implementazioni prevedono chiavi che vanno da 512 a 2048 bit, ma per ottenere una discreta sicurezza, è necessario utilizzare chiavi di almeno 2048 bit.

## 3.3 Crittografia ibrida

Si potrebbe pensare che dopo l’invenzione della crittografia a chiave pubblica quella a chiave segreta abbia perso interesse, ma non è così; la soluzione più indicata nasce da una combinazione dei due sistemi al fine di trarne i vantaggi.

|  |  |  |
| --- | --- | --- |
|  | Crittografia simmetrica | Crittografia asimmetrica |
| Pro | Molto veloce | Nessun problema nello scambio delle chiavi. |
| Contro | Problema nello scambio e nella gestione delle chiavi. | Molto lenta dal punto di vista computazionale. |

Essendo gli algoritmi asimmetrici computazionalmente molto più onerosi di quelli simmetrici, la crittografia ibrida prevede l’uso della crittografia asimmetrica per concordare una chiave segreta da utilizzare poi nello scambio dei messaggi tramite un algoritmo a chiave simmetrica.

Questo approccio combina la flessibilità della crittografia a chiave pubblica con l’efficienza della crittografia a chiave segreta.

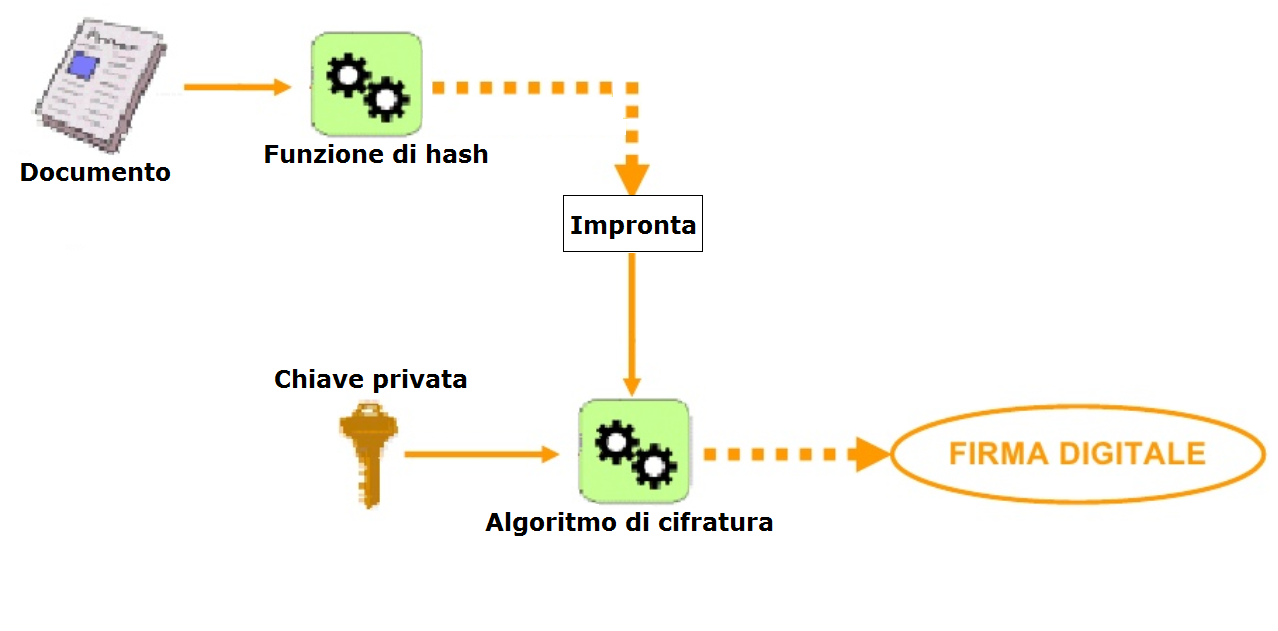
## 

## 3.4 Firma digitale

Una delle principali innovazioni permesse dalla crittografia asimmetrica è la firma digitale. Questo meccanismo prevede l’suo di funzioni hash, funzioni matematiche che trasformano dei dati di lunghezza variabile in una sequenza binaria di dimensione fissa, chiamata *impronta*. Se, ad esempio, stiamo parlando della funzione hash MD5, qualsiasi sia la lunghezza dell’input, l’output sarà sempre di 128bit, ossia 32 caratteri esadecimali.

Oltre ad essere unidirezionali, quindi difficili da invertire, le funzioni hash risultano non essere soggette a collisioni. Tale condizione risulta essere molto rara e avviene quando due diversi input producono lo stesso output; ciò ne ha permesso un ampio utilizzo negli ambiti di sicurezza informatica.

Il sistema prevede per sottoscrivere la firma digitale che il mittente cifri con la propria chiave privata l’impronta’ (detta *digest*) ottenuta attraverso la funzione hash applicata al messaggio. L’impronta cifrata viene poi allegata al messaggio ed inviata.



Il destinatario, una volta ricevuto il messaggio firmato, calcola l’impronta del testo in chiaro, decifra la firma (assicurandosi dell’appartenenza della chiave pubblica al legittimo firmatario attraverso il certificato digitale) e confronta le due impronte: se coincidono la firma è autentica e il documento è integro. In questo modo non si garantisce maggiore segretezza al messaggio, in quanto chiunque può decifrarlo usando la relativa chiave pubblica, ma garantisce:

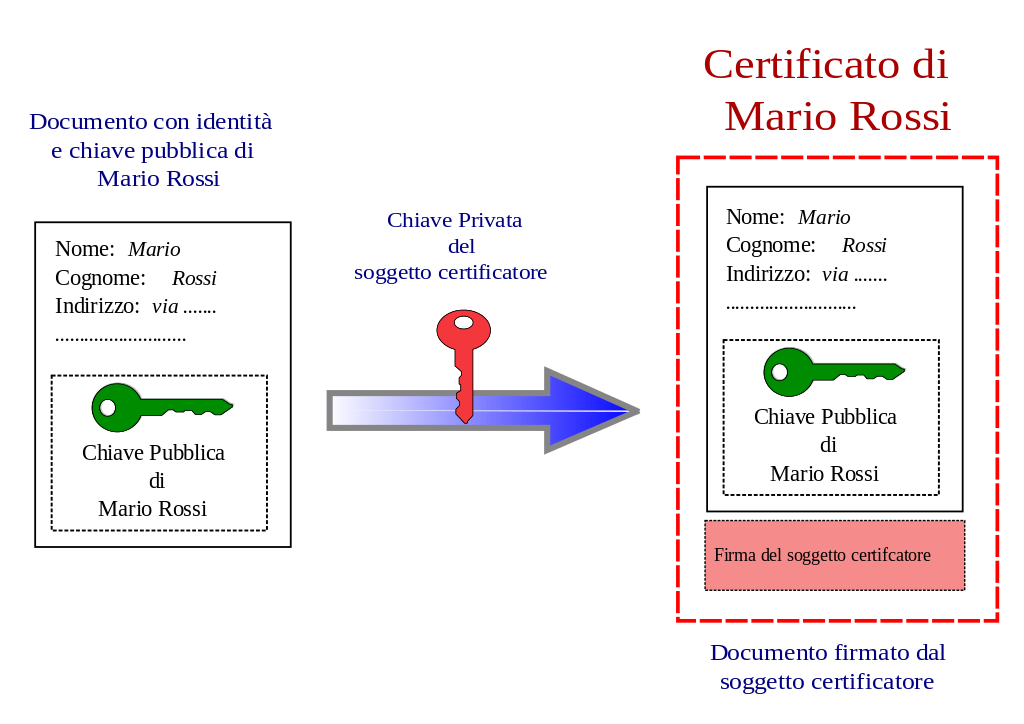
* Autenticazione: il destinatario è sicuro che il mittente sia davvero chi dice di essere, in quanto solo lui possiede la chiave privata complementare alla pubblica utilizzata per decifrare il messaggio;
* Integrità: il messaggio non è stato alterato da terzi in quanto una sua modifica richiederebbe la conoscenza della chiave privata;
* Non ripudiabilità: il mittente non potrà negare di essere l’autore del messaggio in quanto unico proprietario della chiave privata.

Sfortunatamente questo sistema fornisce anche una condizione di termine per i programmi che tentano di forzare la cifratura. Tali programmi tentano di ricostruire la chiave privata del destinatario prendendo come riferimento la firma digitale del messaggio . Propone n chiavi private, per ognuna decifra il messaggio, ne calcola l’hash e lo confronta con l’hash di riferimento decifrato con la chiave pubblica: se coincidono, è stata trovata la chiave privata giusta.

## 3.5 Certificati digitali

Il meccanismo della crittografia asimmetrica prevedere di trasmettere la chiave pubblica su canali non sicuri senza mettere a rischio il messaggio crittografato. Tuttavia sorge il problema di garantire la paternalità delle chiavi.

La soluzione generale è quella di utilizzare una infrastruttura a chiave pubblica o PKI (Public Key Infrastructure), alla base della quale vi è un ente chiamato autorità di certificazione, o CA (Certification Authority). Ogni utente presenta la propria chiave pubblica alla CA e si indentifica con essa. La CA, accertata la corrispondenza, firma la chiave pubblica dell’utente utilizzando una firma digitale e rilasciando un certificato digitale che ne attesti l’associazione univoca.



Attraverso i certificati è molto più semplice trovare le rispettive chiavi che potranno essere inviate direttamente dal destinatario oppure recuperate da un database pubblico; il mittente potrà verificare il certificato in riferimento alla chiave utilizzando quella della CA.

L’uso più comune dei certificati digitali è per l’accesso ai siti web via HTTPS. Il protocollo prevede che, alla connessione, il server fornisca il proprio certificato digitale; se esso è firmato da un’autorità di certificazione riconosciuta e la decifratura della firma ha buon fine, allora si può utilizzare la chiave pubblica presente nello stesso per avviare una comunicazione sicura.

# 4.0 PROTOCOLLI PER LA TRASMISSIONE DI INFORMAZIONI: HTTP

Nell’ambito informatico un protocollo di comunicazione è un insieme di procedure standard definite al fine di favorire la comunicazione tra una o più entità; se le due entità sono remote si parla di protocollo di rete.

Taluni protocolli sono gestiti da organismi quali il World Wide Web Consortium (abbreviato W3C), oltre che da organismi internazionali per gli standard, quali ISO/OSI. Fra i protocolli più utilizzati troviamo HTTP, il principale sistema per la trasmissione di informazioni sul web. L’HyperText Transfer Protocol (HTTP) è un protocollo di livello applicativo che permette una lettura ipertestuale saltando da un server Web ad un altro mediante l’utilizzo di hyperlink. Usato in una tipica architettura client-server, il server resta in ascolto delle richieste dei client sulla porta 80.

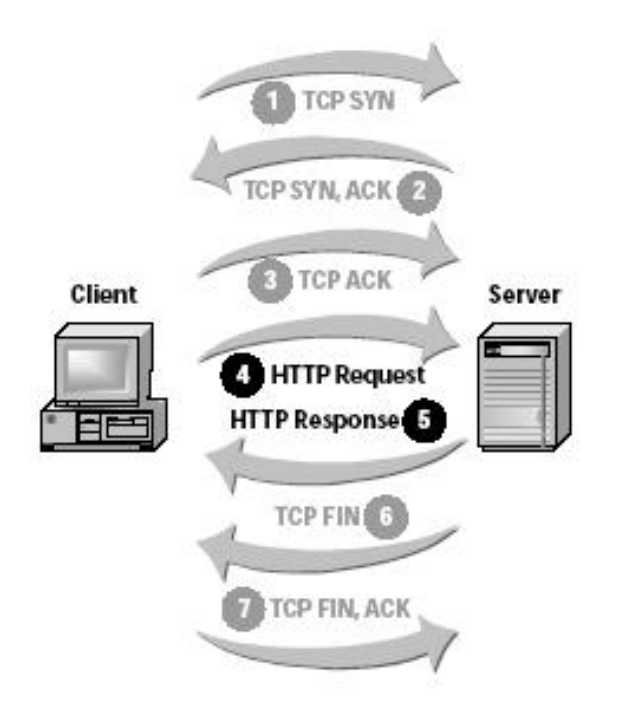
## 4.1 Storia

La prima versione dell’HTTP risale alla fine degli anni 1980 e costituiva, insieme con il linguaggio HTML e gli URL, il nucleo base del World Wide Web (WWW) sviluppata da Tim Berners-Lee al CERN di Ginevra. La prima versione effettivamente disponibile del protocollo, la HTTP/1.0, venne implementata dallo stesso Berners-Lee nel 1991 e proposta come RFC 1945 all’ente normatore IETF nel 1996. Nel corso degli anni il WWW conobbe un successo crescente e divennero evidenti alcuni limiti della versione 1.0 del protocollo che, nel 1997, venne esteso nella versione HTTP/1.1 e successivamente aggiornato nel 1999 come descritto dal RFC 2616.

## 

## 4.2 Connessione

HTTP fa uso del protocollo TCP a livello di trasporto. Poiché un client http è responsabile dell’avvio della comunicazione, esso è anche l’iniziatore del processo che stabilisce la connessione TCP. Nella figura che segue si può notare come avviene il processo:



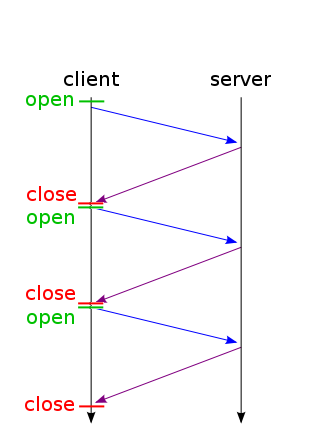
L’acquisizione di una risorsa da parte del client può essere schematizzata in quattro fasi:

* Connessione: il client crea una connessione TCP/IP con il server;
* Richiesta: il client invia la richiesta di una risorsa;
* Risposta: la risposta inviata al server è un messaggio in linguaggio HTML nel quale è contenuta la risorsa richiesta(o un messaggio d’errore);
* Disconnessione: il server subito dopo aver spedito la risorsa richiesta si disconnette.

Il client può domandare al server, nel messaggio di richiesta, quale tipo di comunicazione utilizzare tra:

* Connessione non persistente 🡪 per ogni richiesta e relativa risposta viene stabilita una connessione TCP;
* Connessione persistente 🡪 più richieste e relative risposte utilizzano la stessa connessione TCP.

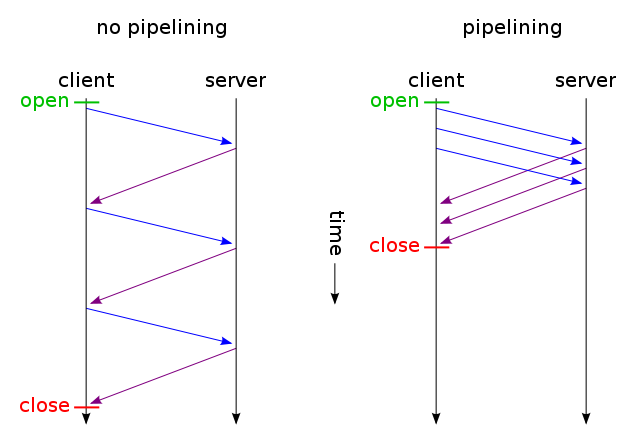
La connessione non persistente prevede che per ogni richiesta e relativa risposta venga stabilita una connessione TCP. Il client inizia una connessione TCP col server, invia un messaggio di richiesta HTTP includendo il nome del percorso e attende di ricevere il messaggio di risposta. Una volta completato lo scambio dei messaggi, la connessione viene chiusa.

Se l’oggetto incapsulato è un file html, il client estrae il file dal messaggio di risposta e lo analizza. Se trova dei riferimenti ad oggetti, per esempio JPEG, deve ristabilire una connessione con il server per farne richiesta.

Ciascuna connessione si chiude dopo che il server invia l’oggetto richiesto, non rimanendo dunque disponibile per altri oggetti. Verranno dunque generate tante connessioni quanti sono gli oggetti da richiedere.

Le connessioni non persistenti introducono una latenza aggiuntiva a quelle persistenti in quanto, al termine di ogni risposta, è necessario effettuare la chiusura della connessione ed, eventualmente, aprirne una nuova procedendo con i passaggi del three-way handshake.

Con una connessione persistente il server lascia aperta la connessione TCP dopo aver spedito la risposta. Le successive richieste e risposte fra gli stessi client e server possono essere inviate sull’identica connessione. In particolare, un’intera pagina web può essere spedita su una singola connessione. Tipicamente, il server http chiude una connessione quando non è usata da un certo tempo (intervallo di *timeout*).

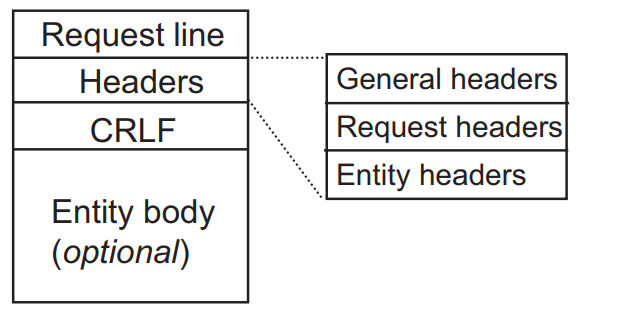
Esistono due versioni della connessione persistente: con o senza pipelining. Mentre senza pipelining il client invia una nuova richiesta solo quando la risposta alla precedente è stata ricevuta, per la versione con il pipelining possono essere inviate più richieste senza aspettare le risposte corrispondenti; le risposte devono però essere fornite dal server nello stesso ordine in cui sono state fatte le richieste. Con questo metodo si ottiene un significativo miglioramento nella velocità di caricamento delle pagine HTML.

## 4.3 Messaggi

Il protocollo prevede due tipi di messaggi: messaggi di richiesta (request) e messaggi di risposta (response).



**Messaggio di richiesta**

****

Il messaggio di richiesta è composto di 3 parti:

* Riga di richiesta (request line);
* Sezione header (informazioni aggiuntive);
* Body (contenuto della richiesta).

La riga di richiesta è composta da:

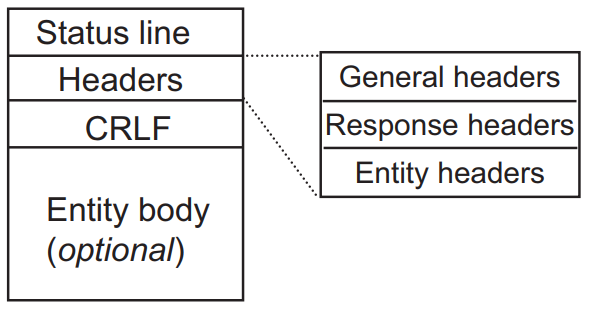
* Metodo;
* URI (Uniform Resource Identifier 🡪 identifica univocamente una risorsa generica);
* Versione del protocollo.

Una tipica riga di richiesta è:

Http 3

In questo caso il metodo richiesto è GET, l’URI è / e il browser implementa la versione HTTP/1.1.

**Messaggio di risposta**

****

Il messaggio di risposta è composta di 3 parti:

* Riga di stato (status line);
* Sezione header (informazioni aggiuntive):
* Body (contenuto della risposta).

La riga di stato è composta da tre campi:

* Versione http;
* Codice di stato;
* Messaggio di stato.

Una tipica riga di stato è:

Http 4

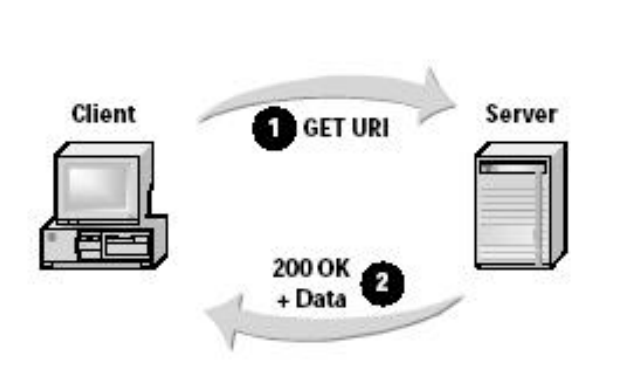
In questo caso il server implementa la versione HTTP/1.1, il codice di stato è 200 e *OK* è il messaggio di stato corrispondente al codice 200.

## 4.4 Metodi della richiesta

Il protocollo HTTP mette a disposizione del client una serie di metodi che, considerabili come comandi del protocollo, vengono inviati come richiesta al server. Ne esistono vari in quanto, con l’avanzare delle versioni del protocollo, sono state apportate delle modifiche. I metodi http più comuni sono *GET, POST* e *HEAD*:

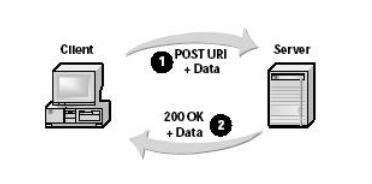
* GET 🡪 richiede una risorsa al server.

Rappresenta la più semplice operazione HTTP e serve ad un client per recuperare un oggetto da un server. E’ il tipo di richiesta che viene inviata quando un utente fa clic su un hyperlink o immette un URI nell’apposito campo del browser. Rappresenta un semplice scambio di due messaggi: il client invia un messaggio contenente il metodo GET e l’URI al server che, dopo aver verificato e identificato la risorsa, la invia indicando il successo dell’operazione con il codice di stato 200. In caso contrario inserirà il codice più opportuno.



* POST 🡪 Invia informazioni all’URI specificato.

Questa operazione permette ai client di trasmettere informazioni al server. La maggior parte dei browser web usa comunemente tale metodo per sottomettere i dati di una form ad un’applicazione sul server, identificata dall’URI specificata nella richiesta.

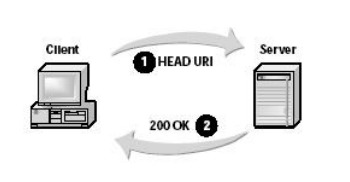


GET e POST sono entrambi capaci, attraverso tecniche differenti, di mandare sottomissioni di form ai server.

Il metodo GET inserisce i dati dell’utente all’interno della URL includendoli tutti dopo il punto interrogativo; tale metodo risulta però essere limitato dalla dimensione dell’URL

Descrizione: http 10

POST invece inserisce i dati alla fine del messaggio di richiesta, nel corpo entità, non ponendo così limiti alla dimensione dei dati. In definitiva, POST risulta essere più affidabile di GET in quanto le informazioni di input vengono inviate come parte dei dati del pacchetto HTTP piuttosto che nella URL.

* HEAD 🡪 richiede solo l’header, senza risorsa: di fatto viene utilizzato soprattutto per la diagnostica. Si differenzia dalla GET solo per il fatto che il server non invia la risorsa richiesta.

In HTTP 1.1 inoltre possiamo trovare:

* PUT 🡪 richiede l’upload di un file sul server, creandolo o riscrivendolo;
* DELETE 🡪 richiede la cancellazione di un file sul server;
* OPTIONS 🡪 richiede l’elenco dei metodi permessi dal server;
* TRACE 🡪 traccia una richiesta, visualizzando come viene trattata dal server;
* CONNECT 🡪 richiede una connessione mediante proxy.

## 4.5 Codici di stato

Il codice di stato è un codice a tre cifre che ha la funzione di fornire al client delle informazioni di stato riguardo l’esito delle ricezione della richiesta. La prima cifra specifica il tipo di stato:

|  |  |  |
| --- | --- | --- |
| Codice di stato | Classe messaggio | Significato |
| 1xx | Informational | Messaggi informativi |
| 2xx | Success | La richiesta è stata soddisfatta |
| 3xx | Redirection | Non c’è risposta diretta, ma la richiesta è ritenuta corretta e viene reinstradata ad un differente server |
| 4xx | Client error | La richiesta non può essere soddisfatta perché sbagliata |
| 5xx | Server error | La richiesta non può essere soddisfatta per un problema interno al server |

I codici di risposta più comuni sono:

* 200 OK: il server ha fornito correttamente la risorsa;
* 302 Found: redirezione a un nuovo URL in quanto l’originale è stato spostato;
* 400 Bad request: la richiesta non è comprensibile al server;
* 403 Forbidden: il client non è autorizzato a ricevere i dati richiesti;
* 403 Not found: la risorsa richiesta non è stata trovata e non se ne conosce l’ubicazione;
* 500 Internal server errore: il server non è in grado di rispondere alla richiesta per un suo problema interno.

## 4.6 Intestazioni

Le richieste e le risposte http possono includere nessuna o più intestazioni che consentono di fornire delle informazioni supplementari.

Possono essere di quattro tipi:

* generali;
* di richiesta;
* di risposta;
* di entità.

Le intestazioni generali si applicano alle comunicazioni http in generale, quelle di richiesta e di risposta ai messaggi specifici di richiesta\risposta e quelle di entità all’eventuale corpo del messaggio incluso in essi. Ognuna di queste righe è composta da un nome che qualifica il tipo di intestazione, seguito da due punti e dal valore dell’intestazione.

Di seguito vengono riportati alcuni esempi:

* User-Agent (di richiesta):

un client può indicare nome del browser e versione, nome del sistema operativo e versione e lingua predefinita.

http 13

In questo modo, per esempio, i siti web sono in grado di rilevare se il navigatore sta utilizzando un browser per dispositivi mobili e gestirne il reindirizzamento a una versione mobile del sito.

* Accept-Language (di richiesta):

questa intestazione visualizza l’impostazione della lingua predefinita dell’utente.

http 18

* Content-Type (di entità):

l’intestazione Content-Type indica il tipo di contenuto restituito attraverso i cosiddetti “mime-type”. Questo consente al browser di capire come interpretare il contenuto. Ad esempio, una pagina html può restituire questo:

Descrizione: C:\Users\user\AppData\Local\Microsoft\Windows\INetCache\Content.Word\http 16.png

‘Text’ è il tipo e ‘html’ è il sottotipo del documento. Per una immagine gif potrebbe essere inviato questo:

http 14

* Content-Lenght (di entità):

tramite l’intestazione Contente-Lenght, il server può indicare al browser la dimensione del contenuto trasmesso (in byte).

Http 17

* Date (generale):

l’intestazione *Date* indica l’ora e la data del momento in cui il messaggio HTTP è stato creato e spedito.

wergergergbe

* Server (di risposta):

tramite l’intestazione *Server*, un server identifica il software che usa per implementare l’HTTP.

wefwefwefwef

## 4.7 Cookies

L’HTTP è un protocollo stateless (senza memoria), cioè un protocollo che non conserva memoria della connessione fatta. Opera come se ogni richiesta fosse indipendente da tutte le altre e ciò è necessario affinché sia possibile saltare velocemente da un server Web ad un altro attraverso link ipertestuali. Il server risponde ad ogni richiesta senza memorizzare alcuna informazione di stato relativa ai client ma, in alcune applicazioni, ha interesse a ricordarsi degli utenti. Il meccanismo definito dall’HTTP per il mantenimento dello stato è costituito dai cookies. I cookies HTTP sono file di testo di piccola dimensione che consentono ai server web di riconoscere i client durante le comunicazioni. Il server mantiene all’interno di un database le informazioni riguardanti il client e gli assegna un identificatore che viene fornito al client come *cookie*. Un server crea i cookies quando vuole tenere traccia dello stato di un client e li manda al client nella sua risposta. Una volta che il client riceve un cookie, può includerlo in richieste successive allo stesso server per farsi riconoscere.

****

## 4.8 HTTPS

Dal momento che tutto il traffico http è in chiaro diviene necessario sviluppare alternative per garantire sicurezza nel trasferimento di dati sensibili.

L’HyperText Transfer Protocol over Secure Socket Layer (HTTPS) è una variante del protocollo HTTP per la comunicazione sicura attraverso una rete di computer largamente utilizzato.

HTTPS consiste nella comunicazione tramite il protocollo HTTP all’interno di un canale di comunicazione criptato tra il client e il server. In sostanza, tra il protocollo TCP e il protocollo HTTP si interpone un livello di criptografia/autenticazione come il Secure Sockets Layer (SSL) o il Transport Layer Security.

In tale maniera si ottengono:

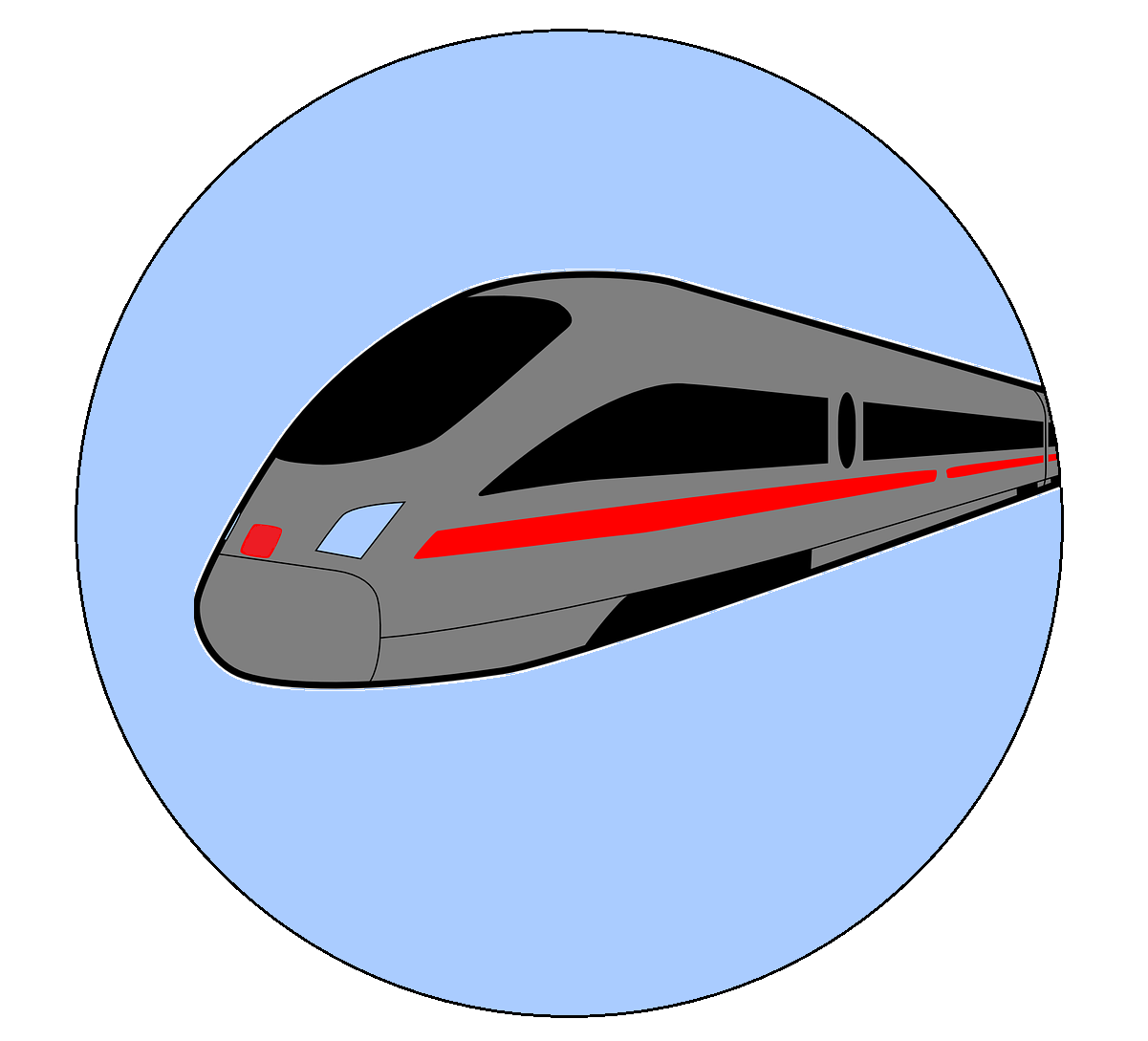
1. Un’autenticazione del sito visitato;
2. La protezione della privacy;
3. L’ integrità dei dati scambiati tra le parti comunicanti.

L’HTTPs prevede l’uso della porta 443 al posto della 80.

La comunicazione HTTPs garantisce che solamente il Client e il Server siano in grado di conoscere il contenuto della comunicazione e impedisce intercettazioni dei contenuti attuate con la tecnica del MiM – Man in the Middle. Impedisce a terzi di leggere, inserire o modificare i messaggi scambiati tra due interlocutori senza che nessuno dei due sia in grado di sapere se il collegamento è stato compromesso.

# QuickTreno

**Documentazione del Progetto**



## Presentazione

*QuickTreno* è il prodotto di tutte le competenze acquisite durante il mio percorso di studi nella scuola secondaria di secondo grado.

Tale progetto è nato dalla mia esperienza in quanto pendolare, che quotidianamente si è spostato dalla località di residenza, Santarcangelo di Romagna, per raggiungere quella della propria scuola, Cesena.

Prese in considerazione le necessità di colui che viaggia in treno, ho pensato di creare un applicativo capace di permettere un accesso alle informazioni inerenti ai treni della nota compagnia Trenitalia.

Le essenziali funzionalità per un pendolare, in maniera semplice ed intuibile.

## 5.1 Tecnologie in uso



Il progetto si basa su Telegram, un servizio di messaggistica istantanea erogato senza fini di lucro dalla società Tegram LLC.

Telegram è stato fondato nel 2013 dai fratelli Nikolai e Pavel Durov. I messaggi inviati sono salvati su cloud, così da garantire la sincronizzazione istantanea; il risultato consente all’utente di poter accedere ai messaggi da diversi dispositivi contemporaneamente.

Le API di Telegram sono disponibili gratuitamente a tutti gli sviluppatori e sono di due tipi: il primo consente di sviluppare il proprio client Telegram, mentre il secondo è dedicato allo sviluppo di programmi che utilizzano Telegram per interfacciarsi all’utente.

Dal giugno 2015 Telegram ha introdotto una piattaforma per permettere a sviluppatori terzi di creare i Bot, account utili per interfacciarsi ad un programma in esecuzione sulla rete, che offre molteplici funzionalità con risposte immediate e completamente automatizzate.

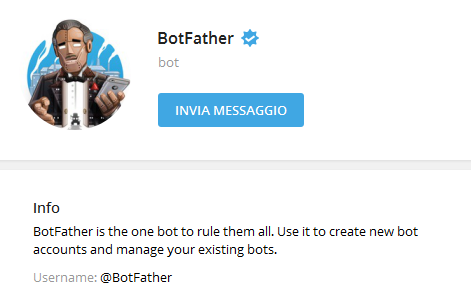
Visual_Studio_2012_logo_and_wordmark

Per realizzare il mio programma di gestione del bot ho utilizzato Visual Studio, un ambiente di sviluppo integrato multipiattaforma sviluppato da Microsoft che supporta il linguaggio C# e che permette la realizzazione di applicazioni, siti web, applicazioni web e servizi web.

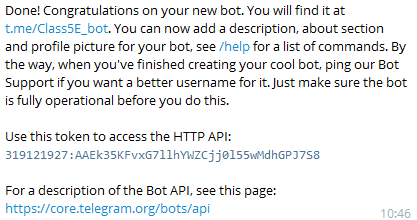
## 5.2 Bot

**Creazione**

Per creare e gestire i propri bot è stato creato *Botfather* (@*BotFather*)



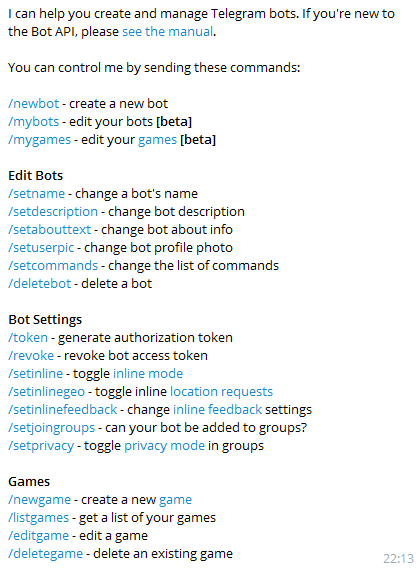
Per creare un bot è sufficiente parlare con Botfather e seguirne le semplici indicazioni. Il bot, al termine del dialogo, fornisce un token, un codice segreto ed identificativo per il proprio bot.



Terminata questa fase, si prosegue con la programmazione del bot.

**Configurazione**

BotFather offre una vasta gamma di funzionalità per il bot a partire da aspetti di design fino ad una vera e propria configurazione. Troviamo:



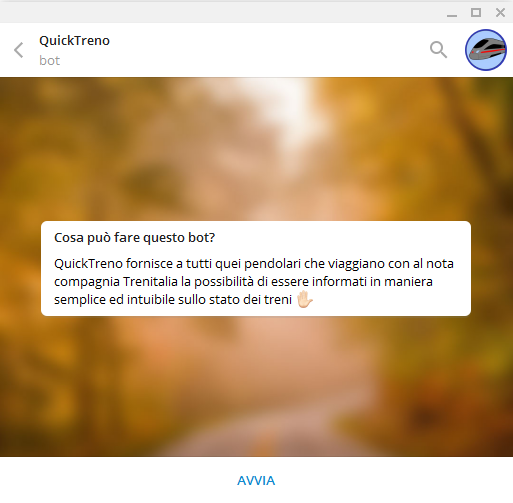
E’ dunque possibile modificare a proprio piacimento il nome, la descrizione e tante altre informazioni che caratterizzano il proprio bot.

## 5.3 Interfaccia utente

Telegram offre un’ interfaccia utente semplice ed intuitiva. Per contattare un bot è sufficiente inserire l’username nella barra di ricerca.

Cerca

Aperta la chat viene fornita una piccola presentazione del bot ed è possibile iniziare a dialogare con esso.



Il tasto *AVVIA* ha la funzionalità di inviare a nome dell’utente il seguente messaggio:



Esso è previsto dal protocollo, come regola generale, per l’avvio di un dialogo.:



Viene dunque avviata una comunicazione a tempo indeterminato con l’applicativo e viene fornita la possibilità di usufruire dei servizi offerti mediante i comandi supportati. I comandi non sono altro che stringhe che, inviate dall’utente attraverso dei messaggi, vengono percepite dal bot come veri e proprio comandi.

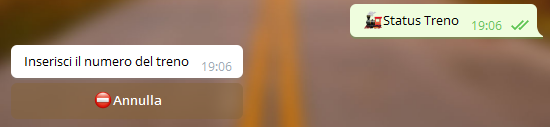
Per questo progetto è stata adottata un’interfaccia orientata all’utente, applicata attraverso molteplici strategie. Prima di tutto si è optato per un uso di comandi differenti da quelli classici, caratterizzati dal tipico carattere ‘/’; si è preferito l’uso di strumenti che l’ambiente di messaggistica ha recentemente sviluppato. Tra questi rientra l’uso delle *custom keyboards,* speciali tastiere intercambiabili, differenti da quelle standard, che presentano i comandi possibili da utilizzare.



Alla pressione dei bottoni consegue l’invio immediato del rispettivo comando e questo semplifica l’interazione dell’utente con il bot:

Telegram 3

Un altro strumento è rappresentato dalle *Inline keyboards*, tastiere integrate all’interno dei messaggi che consentono di eseguire funzioni senza dover scrivere nella chat:



**Utenti**

Il bot può essere richiamato, mediante tecnologie Internet, da qualsiasi dispositivo che supporti Telegram.

Telegram è multipiattaforma per cui il bot potrà essere contattato da differenti tipologie di dispositivi. Esistono applicazioni dedicate ad IOS, Android e Windows Phone. E’ inoltre utilizzabile il client web di Telegram oppure una delle applicazioni desktop per Windows, OSX e Linux.

**5.4 Analisi**

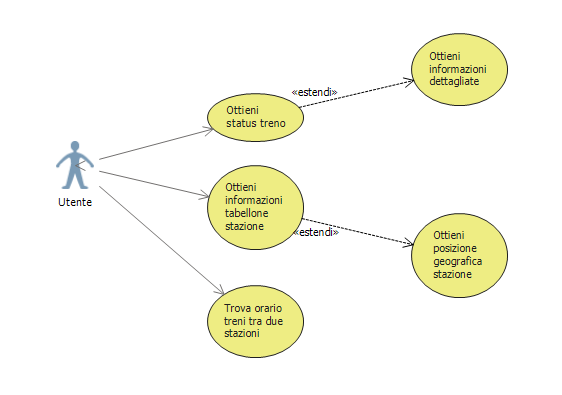
L’obiettivo del progetto è quello di fornire in modo semplice e veloce informazioni sui treni della nota compagnia Trenitalia.

In primis è necessario definire quali funzioni si vogliono realizzare ed offrire. Principalmente l’offerta si compone di tre opzioni:

1. Offrire le informazioni basilari sullo stato di un treno nel momento della richiesta: stazione di partenza, di arrivo, ultimo rilevamento, ritardo, ecc;
2. Offrire la lista dei treni in arrivo e in partenza in una determinata stazione, le informazioni dunque presenti sui tabelloni elettronici installati sul luogo;
3. Offrire gli orari dei treni che viaggiano da una stazione ad un’altra;

Oltre a queste funzionalità potrebbe essere utile fornire la possibilità di ottenere qualche dettaglio in più su un determinato treno: stazioni effettuate, stazioni da effettuare, orari specifici, ….

Oltretutto risulterebbe molto efficace poter memorizzare i propri treni e le stazioni preferite al fine di accedere alle prime due funzionalità in maniera molto più facile ed immediata.



## 5.5 Progettazione

Stabilite le funzionalità in fase di analisi, si ipotizza di procedere secondo il principio della macchina a stati, affinché si possa gestire con precisione e in maniera formale il sistema. I vari passaggi sono identificati da comandi che possono essere pubblici, conosciuti anche all’utente, oppure privati, conosciuti unicamente dal sistema. Questo metodo conferisce ampia flessibilità al progetto per consentire di apportare modifiche in modo rapido e mirato.

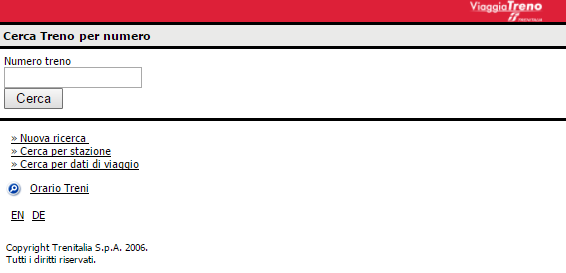
**Risorse**

Le informazioni dovranno essere scaricate da fonti sicure ed offerte tramite strumenti istantanei e di facile utilizzo, che possano essere fruibili dal maggior numero di persone. Per tale motivo potrebbe essere consigliato l’uso di applicativi di messaggistica istantanea oggi soggetti ad un uso su scala mondiale sempre più massiccio. Telegram offre la più vasta gamma di possibilità in questo ambito, incoronandosi come migliore possibilità open source sul mercato; permette di interfacciarsi ad un programma in esecuzione sulla rete.

QuickTreno si occupa di fornire informazioni relative ai treni della società Trenitalia, la quale non offre però servizi ufficiali per accedervi. La soluzione che verrà adottata consisterà nell’utilizzare la sezione dedicata al mobile del sito ufficiale della compagnia:

<http://viaggiatreno.it/vt_pax_internet/mobile>

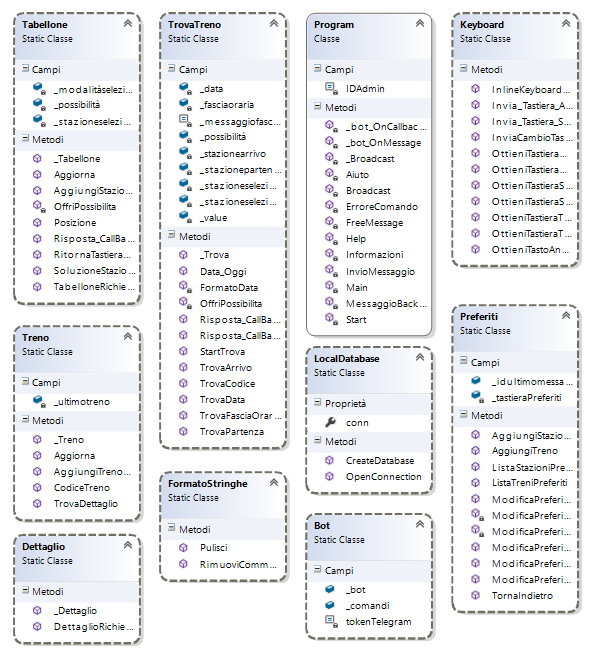
Il sito web fornisce pagine dedicate a funzionalità diverse. Di seguito viene mostrata l’interfaccia per ottenere informazioni sullo stato di un treno:



**Classi**

All’interno del progetto dovranno essere presenti le seguenti classi:

* Bot: contiene il riferimento al bot, il token e il dizionario dei comandi;
* Treno: permette di ottenere informazioni su un treno (status, ecc.);
* Dettaglio: permette di ottenere la lista delle stazioni percorse da un treno, con orari;
* Tabellone: permette di ottenere la lista dei treni in arrivo e in partenza in una stazione;
* TrovaTreno: permette di trovare gli orari dei treni che viaggiano tra due stazioni;
* LocalDatabase: permette di interagire con il DB;
* Keyboard: offre delle *custom keyboards* predefinite;
* FormatoStringhe: offre dei metodi per la pulizia delle stringhe.



**5.6 Studio dell’origine dell’informazione**

Trovata una fonte di dati autentici, è stato necessario studiare il codice sorgente delle diverse pagine html fornite come risposta dal server web. Di seguito un esempio di pagina:

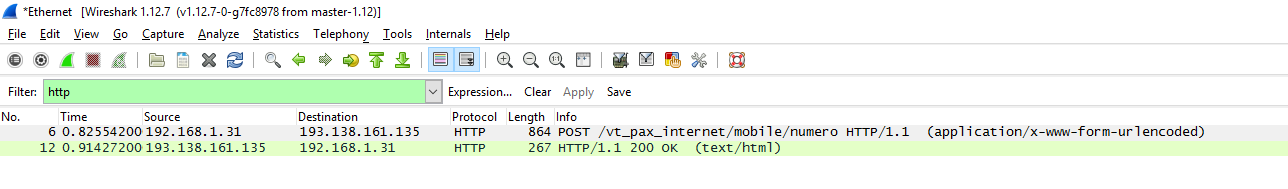


Lo studio è stato essenziale per comprendere il formato e l’impaginazione delle pagine html al fine di estrarne e manipolarne i dati.

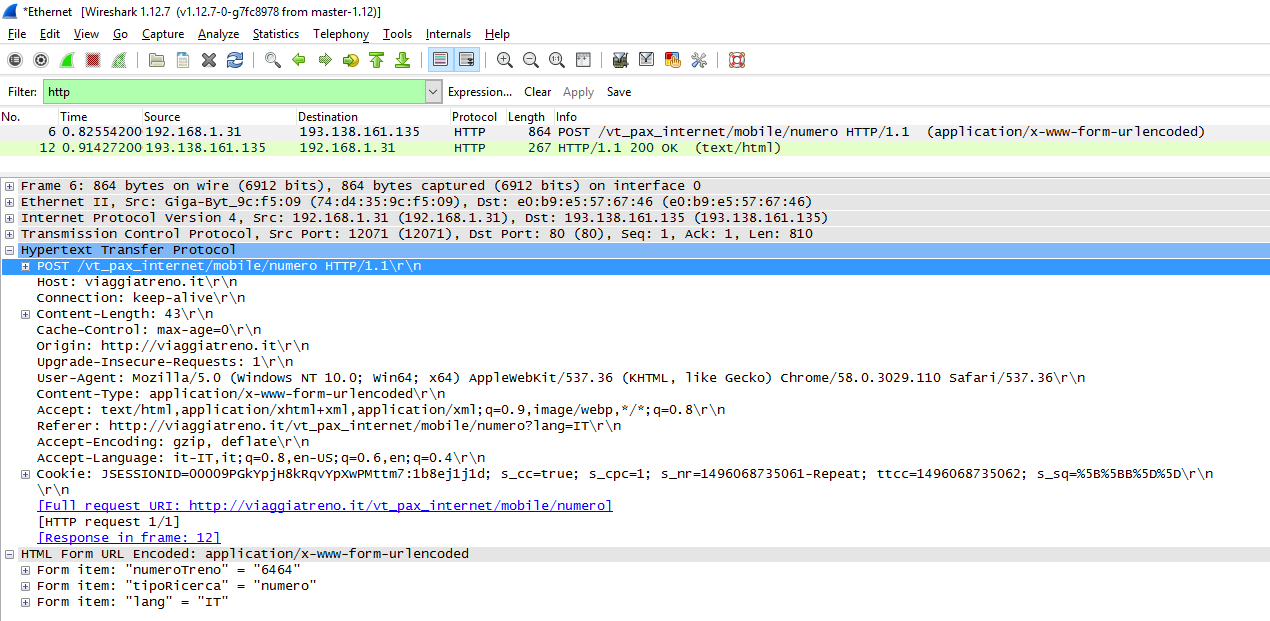
Richieste diverse comportano risposte diverse, per cui è stato necessario, attraverso l’uso del software Wireshark, studiare il traffico dati a livello applicativo per ottenere maggiori informazioni riguardo il dialogo client-server. Attraverso questo software è possibile analizzare il traffico presente sulla rete. È dunque bastato effettuare le richieste d’interesse al server per poter analizzare le caratteristiche del traffico generato.

Di seguito viene mostrato il traffico della richiesta http alla pagina:

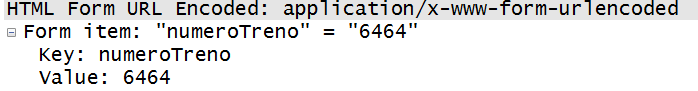
<http://viaggiatreno.it/vt_pax_internet/mobile/numero>



Più in dettaglio, osserviamo:



E’ possibile notare le informazioni di POST:

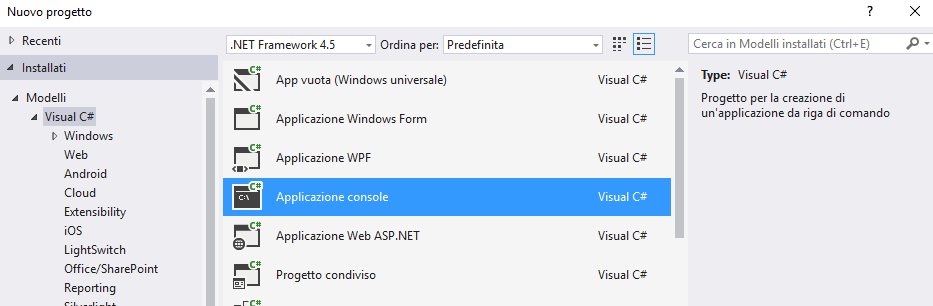


Il POST rappresenta uno dei metodi messi a disposizione del protocollo http, utile all’ invio di informazioni all’URI specificato. Si è dunque scoperto, per ogni tipologia di richiesta, il metodo http utilizzato, i dati inviati e la conseguenza incidenza sull’impaginazione della risposta del web server.

Attraverso le informazioni ricavate è ora possibile replicare lo scambio dei messaggi.

## 5.7 Sviluppo

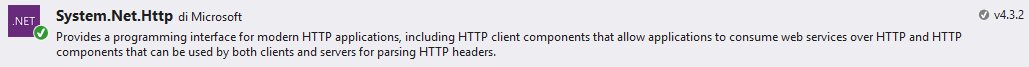
QuickTreno è un bot del noto servizio Telegram gestito, in questo caso, da un programma scritto in linguaggio C#.

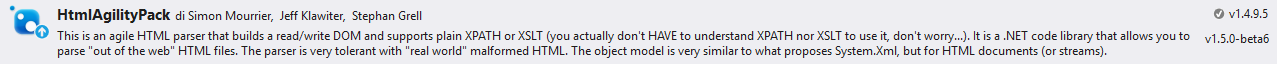


Il progetto base non comprende tutte i pacchetti necessari allo sviluppo per cui è stato necessario ricorrere alla gestione dei pacchetti NuGet per installarne di nuovi.

Di seguito i pacchetti utilizzati:

1

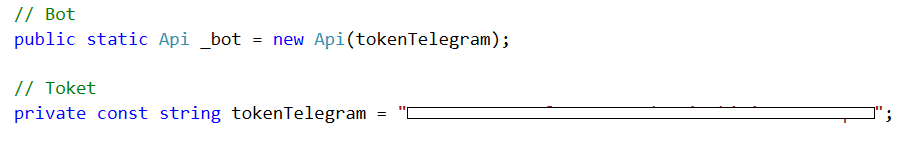




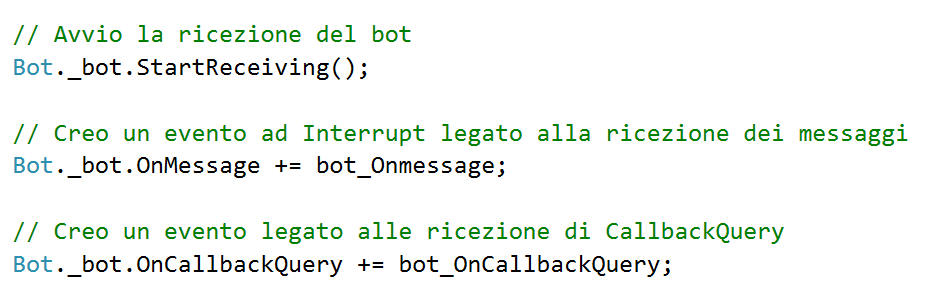
4

Le classi sono state create tutte statiche, in quanto sono risultate necessarie variabili condivise e un accesso ai metodi valido per tutti.

In primis è stato istanziato il bot fornendo, come parametro, il token fornito da BotFather:



Successivamente si è stabilito che ad ogni ricezione di messaggi o CallBackQuery vengano richiamati metodi ad interrupt, al fine di gestire le informazioni ricevute:



Al fine di gestire due componenti diverse, vengono utilizzati due metodi, con firme distinte:

* OnMessage si occupa di gestire i messaggi tradizionali;

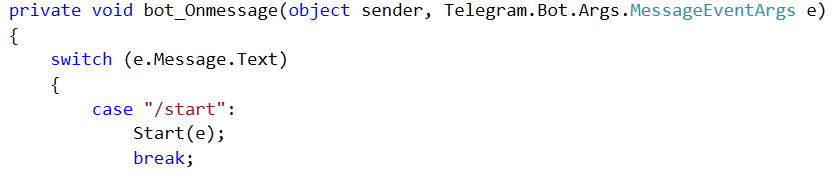
telegram 7

* OnCallBackQuery si occupa di gestire le CallBackQuery, query generate dalla pressione dei bottoni in una inline keyboard.

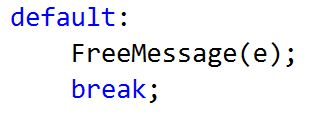
telegram 8

All’interno di questi metodi il traffico viene smistato attraverso un particolare strumento: lo switch.

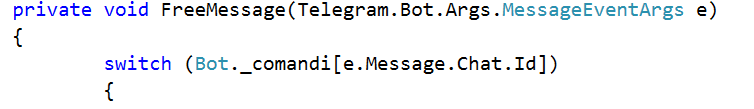
Il metodo Onmessage, che si occupa di gestire i comandi pubblici, esamina il contenuto del messaggio ricevuto ed esegue un confronto con la lista dei comandi supportati. Se un candidato fa match viene istantaneamente richiamato il metodo previsto per soddisfare la funzionalità richiesta.



In caso non venga riconosciuto alcun comando, l’ispezione passa ad un secondo metodo, *FreeMessage*:



Questo metodo, esattamente come il precedente, cerca di richiamare, per corrispondenza di stringhe, il metodo opportuno.



La ricerca non viene più effettuata all’interno del contenuto del messaggio, bensì all’interno di un dizionario, il cui compito è memorizzare, per ciascun utente, un comando:

telegram 12

Ciascun utente è identificato da un codice univoco che risulta ideale per un ottimale uso del dizionario. Ad ogni codice viene dunque associato un comando, assegnato e modificato nel corso dell’adempimento della funzionalità richiesta. Infatti, a differenza di molti bot, QuickTreno non prevede il classico protocollo comando-risposta, bensì ad ogni comando possono essere previste più richieste di informazioni, al fine di soddisfare la richiesta. Si specifica che tale operazione nasce dall’idea di rendere l’interazione più orientata all’utente.

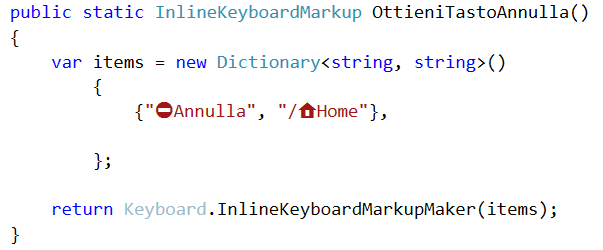
Il bot riconosce un dato ricevuto come risposta ad una determinata domanda, memorizzando l’ultimo comando eseguito. E’ necessario dunque che, ogni qualvolta un metodo invia un messaggio di richiesta, venga memorizzato il suo comando identificativo. In questo modo è possibile associare il dato ad una richiesta e, in caso di errore da parte dell’utente, è possibile ripercorrere il passaggio.

Questi comandi “privati”, gestiti dal metodo FreeMessage, risultano fondamentali per fornire all’utente una migliore esperienza d’uso, un applicativo più agile e meno meccanico.

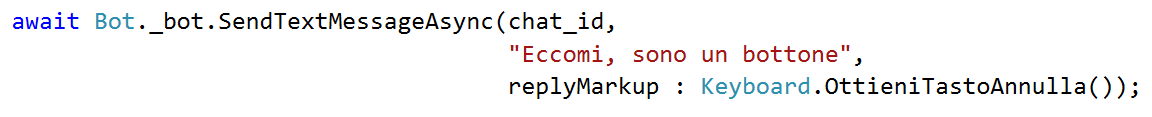
Questo obiettivo è stato raggiunto anche grazie al metodo OnCallBackQuery. Esso si occupa di gestire comandi che non possono essere definiti né privati né pubblici, in quanto vengono forniti all’utente senza che esso se ne accorga.

Tale metodo permette di gestire le *inline keyboard*.

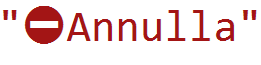
I bottoni delle *inline keyboard*  vengono creati tramite il seguente codice:



e inviati attraverso un messaggio:



La particolarità è la capacità di creare dei bottoni con un testo visibile all’utente, in questo caso:



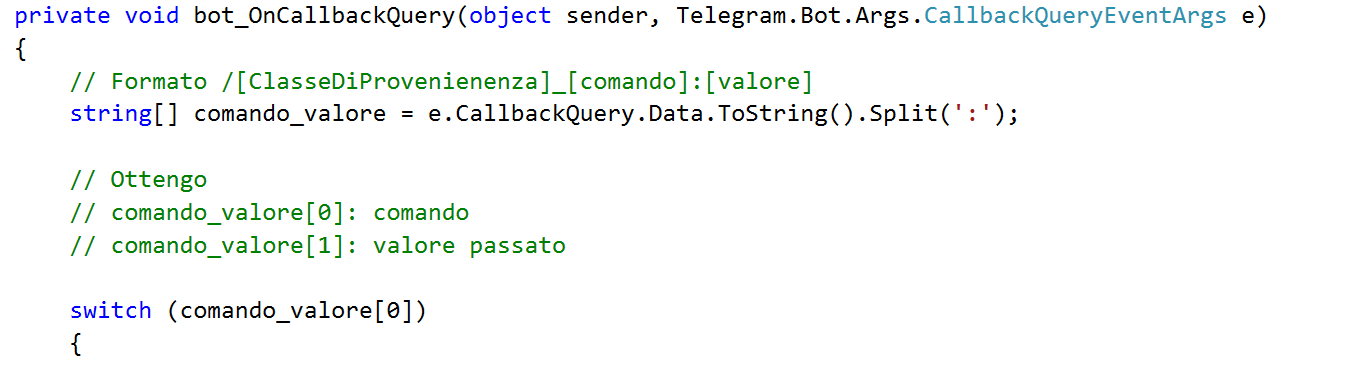
ed uno nascosto e associato ad esso, quale:

Telegram 16

Attraverso questo strumento è quindi possibile offrire all’utente un bottone adatto alle sue esigenze e facile da comprendere, utilizzando, allo stesso tempo, lo stesso sistema di comandi del resto del progetto.

L’unica differenza è che la pressione di uno di questi bottoni non comporta come risultato l’invio di un messaggio all’interno della chat; le *inline keyboards* lavorano nel “retroscena”.

Il metodo OnCallBackQuery, richiamato ad interrupt ogni qualvolta un bottone delle inline keyboards viene premuto, agisce esattamente come i metodi sopra citati.



Esso presenta infatti uno switch che esamina il comando ricevuto e lo confronta con quelli supportati.

Al fine di ottenere una gestione propensa al futuro è stato necessario definire un formato per i messaggi non visibili all’utente, secondo la seguente struttura:

*/ClasseDiProvenienza\_Comando:Valore*

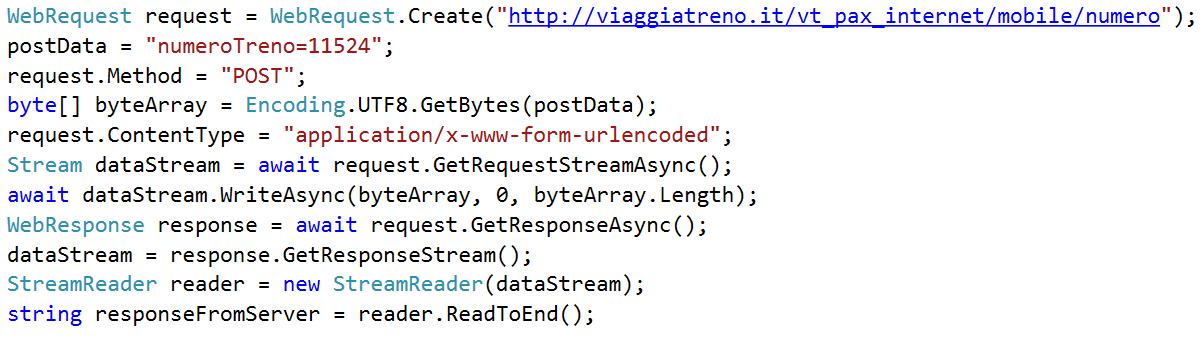
Il formato prevede:

* /: Barra classica di identificazione dei comandi;
* ClasseDiProvenienza: nome della classe che ha creato il bottone;
* Comando: nome del comando che viene richiesto dal quello specifico bottone;
* Valore: valore passato al metodo identificato dal comando.

Vengono dunque presi i due campi fondamentali, comando e valore, separati dai due punti e viene confrontato il comando con quelli disponibili. Trovata una corrispondenza viene richiamato il metodo corretto e, come parametro, viene passato il valore fornito. I metodi possono essere categorizzati differentemente: quelli che domandano informazioni all’utente e, contemporaneamente, registrano i dati che questo ha fornito ad una richiesta precedente e i metodi “conclusivi”, che fanno uso di tutte le informazioni raccolte al fine di concludere la richiesta effettuata. Questi ultimi presentano tutti lo stesso paradigma:

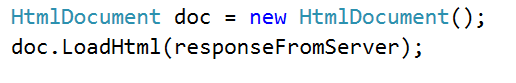
* Effettuano la richiesta al web server;
* Analizzano la risposta del web server;
* Inviano le informazioni ottenute all’utente.

Una volta inviate le informazioni, la richiesta può considerarsi soddisfatta. La richiesta al web server viene eseguita fornendo, tramite metodo POST, le informazioni ottenute precedentemente. Di seguito l’impostazione di una richiesta:



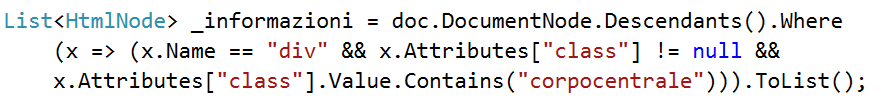
La struttura con cui esse devono essere formattate è stato argomento di studio nella fase di analisi.

Il metodo restituisce un flusso di dati che rappresenta una pagina html la quale, ora in formato stringa, verrà caricata all’interno di una istanza della classe HtmlDocument:



La pagina html potrà così essere gestita come struttura a nodi. Ecco due esempi:

Codice 3

**

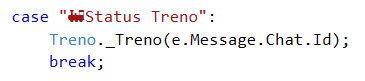
## 5.8 Esempio

Viene ora mostrata una delle funzionalità più semplici: Status Treno.

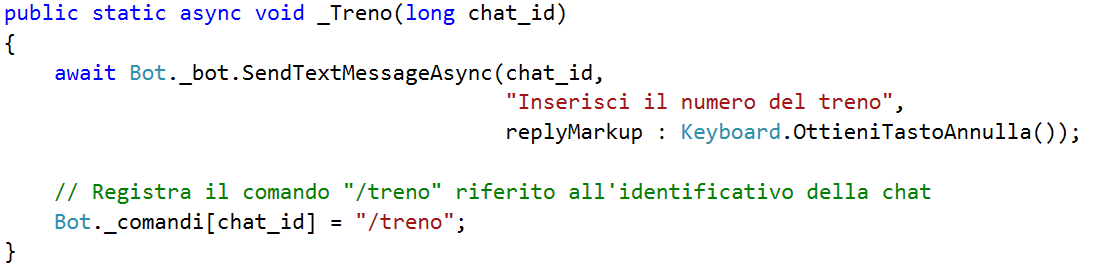
L’utente seleziona la funzionalità dalla tastiera ottenendo così l’invio del messaggio:

Telegram 18

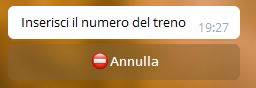
La ricezione del messaggio da parte del bot implica il richiamo del metodo ad interrupt bot\_Onmessage che, analizzandone il contenuto, trova match con:



Viene dunque richiamato un metodo della classe Treno, per l’esattezza \_Treno:



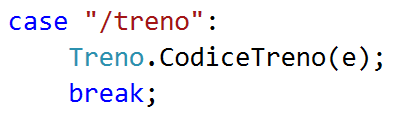
Dopo aver preso in carico l’invio del messaggio di richiesta all’utente, il cui codice identificativo è contenuto all’interno della variabile chat\_id, il metodo si occupa di memorizzare all’interno del dizionario dei comandi la stringa “/treno” a nome del medesimo utente:



L’utente deve poi rispondere con il codice del treno desiderato:

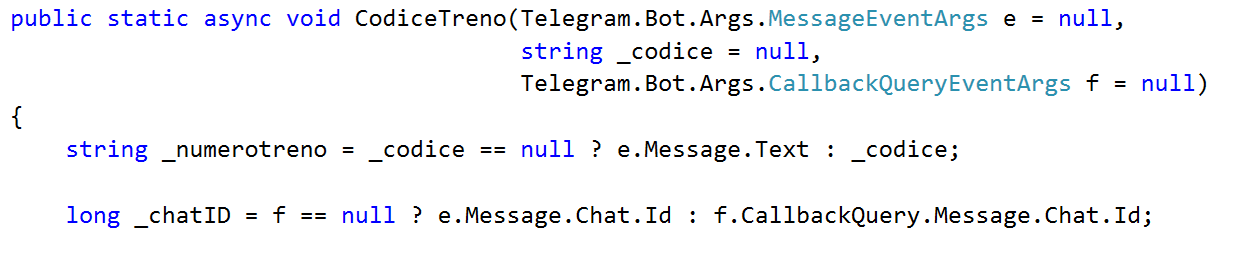
Telegram 22

Questa volta, non contenendo il messaggio un comando, il compito di trovare una corrispondenza viene passato al metodo FreeMessage. Analizzando il comando contenuto nel dizionario, il match avverrà con:

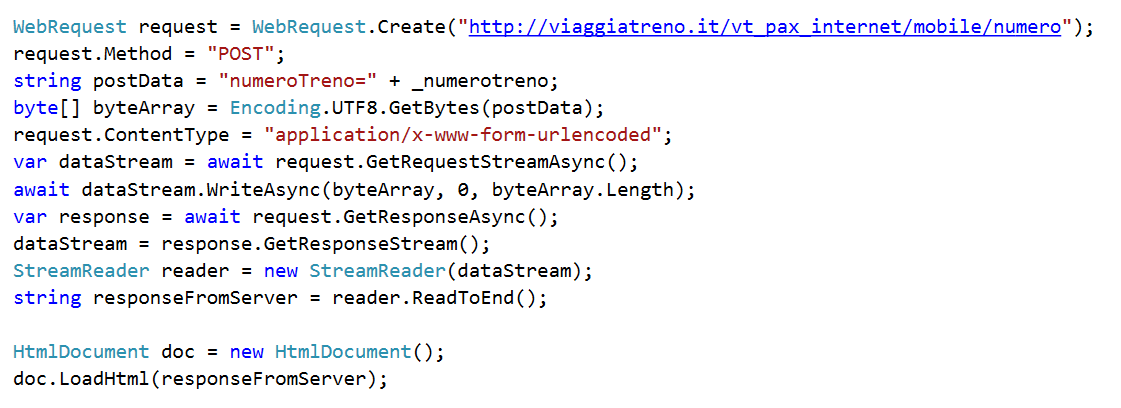


Viene dunque richiamato nuovamente un metodo della classe Treno, per l’esattezza CodiceTreno.

Inizialmente esso stabilisce in quale modalità il metodo è stato richiamato e le variabili dipendenti da questo fattore vengono assegnate correttamente:

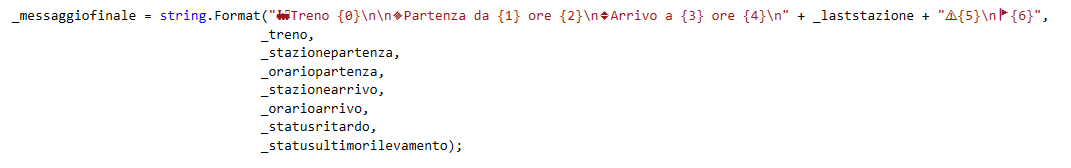


Viene eseguita l’operazione di POST:

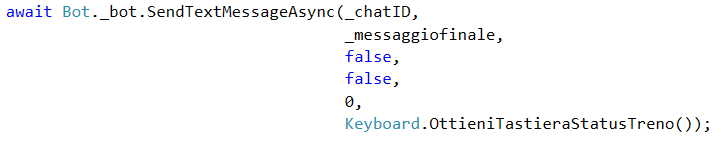


Ottenuta la risorsa richiesta, si inizia ad analizzare l’HtmlDocument, estraendo i dati necessari come, ad esempio, la sigla del treno:

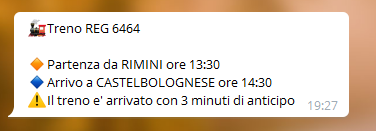
Telegram 27

Si crea dunque il messaggio di risposta:

Si procede con l’invio:



Si ottiene, lato utente, questo risultato:



**Bibliografia**

Roberto Saviano, La paranza dei bambini, Feltrinelli, 2016

*Social networking and netiquette*, Oxford University Press

E. Baldino, R. Rondano, A. Spano, C. Iacobelli, *Internetworking Sistemi E Reti*, Juvenilia Scuola

C. Iacobelli, E. Baldino, F. Beltramo, R. Rondano – *Progettazione Tecnologie In Movimento*, Juvenilia Scuola

E. Baldino, R. Rondano, A. Spano, C. Iacobelli, *Internetworking Sistemi E Reti,* Juvenilia Sc

**Sitografia:**

<http://www.robertosaviano.com/biografia/>

<https://it.wikipedia.org/wiki/Roberto_Saviano>

<http://espresso.repubblica.it/opinioni/l-antitaliano/2013/11/14/news/se-scompaiono-i-fatti-e-le-notizie-1.141111>

<http://www.lastampa.it/2017/04/07/societa/saviano-i-social-network-eiq2ZdqcoQ7CIs878GXPaP/pagina.html>

http://www.repubblica.it/spettacoli

<http://www.corrierecomunicazioni.it/it-world/31208_roberto-saviano-il-digitale-e-liberta.htm>

<https://www.milanofree.it/milano/cultura/saviano_al_forum_della_comunicazione_digitale_il_web_e_liberta.html>

<https://firstmaster.it/saviano-scrivere-e-responsabilita/>

<https://www.wired.it/>

<http://www.feltrinellieditore.it/>

<http://www.stopcyberbullying.org/what_is_cyberbullying_exactly.html>

<https://www.stopbullying.gov/cyberbullying/what-is-it/>

<https://www.pacerteensagainstbullying.org/experiencing-bullying/cyber-bullying/>

<http://www.ncpc.org/topics/cyberbullying/what-is-cyberbullying>

<http://onlinelearningtips.com/2015/11/04/creating-a-culture-of-empathy-to-combat-cyberbullying/>

<https://www.webroot.com/us/en/home/resources/tips/ethics-and-legal/ethics-netiquette-and-online-ethics-what-are-they>

<https://digitalmobileadv.com>

<http://www.fidadisturbialimentari.com>

<http://ww2.unime.it/sei/pdf/sei7.pdf>

<https://avires.dimi.uniud.it/claudio/teach/sicurezza2013/lezione-04.pdf>

<http://bottaroluca5f.altervista.org/crittografia-simmetrica-asimmetrica/>

<https://it.wikipedia.org/wiki/Crittografia_simmetrica>

<http://slideplayer.it/slide/3635902/>

<https://www.w3.org/Protocols/rfc2616/rfc2616-sec4.html>

<https://it.wikipedia.org/wiki/Hypertext_Transfer_Protocol>

<http://openskill.info/infobox.php?ID=357>

<https://www.francoangeli.it/Ricerca/Scheda_libro.aspx?CodiceISBN=9788846416377>

<http://www.fisgeo.unipg.it/luca.gammaitoni/fisinfo/documenti-fisici/Crittografia3.pdf>

<https://www.gianofamily.org/esperto-risponde/linee-guida-di-base/differenza-tra-http-e-https.html>

<http://it.ccm.net/contents/44-il-protocollo-http>

<http://reti.dsi.uniroma1.it/eng/maselli/Lezione_4-aa1415.pdf>

<http://www.di-srv.unisa.it/~ads/corso-security/www/CORSO-9900/http/>

<http://tweb.ing.unipi.it/tia/http.pdf>

<https://code.tutsplus.com/it/tutorials/http-headers-for-dummies--net-8039>

<http://www.ce.uniroma2.it/~lopresti/Didattica/RetiWeb/RetiWeb1213/HTTP_completo4.pdf>