

# S.C.R. Lifecoach Chatbot

---



## Membri del gruppo

- Amati Cristian
- Campana Antony
- Cervino Alessio
- Marino Giuseppe
- Mauro Alessandro

Questo software è stato realizzato da un gruppo composto da sette studenti della facoltà di Informatica all'Università degli studi di Bari, durante il corso di Sistemi ed Agenti tenuto dalla docente B. De Carolis.

L'obiettivo prefissatoci era la realizzazione di un bot lifecoach che assistesse gli utenti durante alcune comuni attività sportive sfruttando l'immediatezza di una piattaforma come Telegram. Il processo di lavoro, della durata di circa 3 settimane, è stato diviso tra apprendimento di piattaforme e software vario, sviluppo del codice, test e documentazione.

## Cosa fa?

---

Il chatbot **S.C.R LifeCoach** è un bot disponibile per la piattaforma di messaggistica Telegram. Il bot permette ad un utente che pratica sport quali nuoto, corsa e ciclismo in modo amatoriale di pianificare le attività durante le giornate, tenere traccia delle sue prestazioni a fine attività e ricevere consigli sulla base di queste.

## Telegram

---



Telegram è un servizio di messaggistica istantanea e broadcasting basato su cloud ed erogato senza fini di lucro fondata dall'imprenditore russo Pavel Durov. I client ufficiali di Telegram sono distribuiti come software libero per Android, Linux, iOS, MacOS, Windows 10.

Caratteristiche di Telegram sono la possibilità di scambiare messaggi di testo tra due utenti o tra gruppi fino a 200.000 partecipanti, effettuare chiamate vocali e videochiamate cifrate punto-punto, scambiare messaggi vocali, videomessaggi, fotografie, video, sticker e file di qualsiasi tipo fino a 2 GB. Attraverso i canali è anche possibile la trasmissione in diretta di audio/video e testo verso i membri che si uniscono.

Da giugno 2015 Telegram ha introdotto una piattaforma per permettere, a sviluppatori terzi, di creare i bot. I Bot sono degli account Telegram, gestiti da un programma, che offrono molteplici funzionalità con risposte immediate e completamente automatiche. In altre parole, i bot sono un comodo sistema per automatizzare alcune interazioni con gli utenti, fornendo risposte prestabilite e "precompilate" alle loro domande e richieste. Ce ne sono di diversi tipi, tra quelli dedicati ad azioni lato singolo utente o quelli dedicati alla gestione di canali con migliaia di iscritti.

## Creazione di un bot su Telegram

La creazione di un bot su Telegram è un'operazione abbastanza semplice e viene svolta interagendo con il bot BotFather già esistente e visibile a tutti. È sufficiente cercare @BotFather nel campo di ricerca dell'app e avviare l'applicazione ufficiale BotFather che, tra le varie operazioni, consente di creare un nuovo bot tramite il comando `/newbot`.

Telegram chiede di dare un nome e uno username al bot. Al termine della procedura di creazione pubblica una chiave HTTP API (detto token) da usare per controllare il bot in remoto tramite un'applicazione esterna.

Infine, occorre sviluppare l'applicazione esterna per controllare il bot in remoto e dichiararla sulla piattaforma dell'instant messaging.

# Wit.ai

---



Per aiutare gli sviluppatori a costruire interfacce conversazionali basate su testo, l'azienda ha rilasciato una soluzione bot engine, promuovendo la realizzazione di bot tramite gli strumenti di Natural Language Processing. Il motore di bot è stato pensato in modo da fornire supporto allo sviluppo di interfacce anche se senza un'apposita interfaccia grafica. La piattaforma permette quindi di creare interfacce conversazionali, in grado di tradurre il linguaggio naturale in dati tramite apposite API e di apprendere nel corso delle interazioni, migliorando nel tempo grazie a tecniche di apprendimento automatico.

Le API si occupano di convertire parole e frasi in dati strutturati, interpretabili da una macchina. Una volta ottenuti questi dati, è compito poi del programmatore decidere come utilizzarli. Ad esempio, con l'utilizzo di queste API possono essere realizzate applicazioni di domotica e interfacce di controllo vocale.

## Intents

Intents			<a href="#">+ Intent</a>
<input type="text"/>			
Name ↕	Entities		
add_cycling	Ciclismo:Ciclismo, Corsa:Corsa		...
add_running	Corsa:Corsa		...
add_swimming	Nuoto:Nuoto		...

Un intento è l'intenzione dell'utente. Ad esempio, se un utente digita "mostrami le notizie finanziarie di ieri", l'intento dell'utente è recuperare un elenco di titoli finanziari. Agli intenti viene assegnato un nome, spesso un verbo e un sostantivo, come "add\_activity".

Per il nostro progetto sono stati definiti tre diversi intets rlativi alle tre diverse attività sportive che il chatbot gestisce.

## Entities

Entities <span>+ Entity</span>			
<input type="text"/>			
Entity ↕	Roles	Intents	
Ciclismo	Ciclismo	add_cycling	...
Corsa	Corsa	add_cycling, add_running	...
Nuoto	Nuoto	add_swimming	...
wit/datetime	datetime, day, time		...
wit/distance	distance		...
wit/duration	duration		...
wit/location	location		...

Un'entità modifica un intento. Esse rappresentano concetti propri del dominio considerato. Ad esempio, se un utente digita "mostrami le notizie finanziarie di ieri", le entità sono "ieri" e "finanziaria". Alle entità viene assegnato un nome e un collegamento ai relativi intenti.

## NLP

L'elaborazione del linguaggio naturale (NLP) si riferisce ad un ramo dell'informatica, e più specificamente, dell'intelligenza artificiale o AI, che si occupa di dare ai computer la capacità di comprendere il testo e le parole allo stesso modo degli esseri umani.

La NLP combina la linguistica computazionale, la modellazione basata su regole del linguaggio umano, con modelli statistici, di machine learning e di deep learning. Insieme, queste tecnologie consentono ai computer di elaborare il linguaggio umano sotto forma di testo o dati vocali e di "comprenderne" il significato completo, completo dell'intento e del sentimento di chi parla o scrive trasformandoli in dati strutturati comprensibili.

## Ambiente di sviluppo



È stato utilizzato il linguaggio Python grazie al suo interfacciamento moderno ed efficace con gli strumenti utilizzati per il funzionamento del bot. Python è uno dei linguaggi di programmazione più utilizzati al mondo

per la sua estrema semplicità e supporto da parte degli sviluppatori.

L'ambiente di sviluppo scelto è stato PyCharm. PyCharm è un ambiente di sviluppo integrato (IDE) utilizzato in particolare per il linguaggio Python. È sviluppato dalla società ceca JetBrains. Fornisce analisi del codice, debugger grafico, unit tester integrato ed è cross platform.

Sono state utilizzate le seguenti librerie compatibili con il linguaggio Python.

Libreria	Funzionalità
botogram	botogram è un framework Python, che consente di creare bot Telegram
pywit	PyWit fornisce collegamenti all'API HTTP di Wit.ai
itranslate	iTranslate consente la traduzione degli input degli utenti appoggiandosi alle API Google
sqlite3	sqlite3 fornisce una interfaccia compatibile DB-API 2.0 al database relazionale SQLite.

Come strumento di interfacciamento tra il linguaggio Python e telegram è stato utilizzato il framework Botogram. Dai suoi creatori:

*"Botogram è un framework Python, che ti consente di concentrarti solo sulla creazione dei tuoi bot di Telegram, senza preoccuparti dell'API Bot sottostante. La maggior parte dei dettagli di implementazione di Telegram sono gestiti da botogram, quindi puoi concentrarti solo sul tuo bot."*

## Funzionamento

Quando un utente invia un messaggio al bot il framework botogram analizza il contenuto del messaggio che può essere di:

- callback
- comando
- qualunque

e in base ad esso richiama la funzione di gestione corretta per quella specifica tipologia.

Ad eccezione dei comandi di start ed help, la funzione principale è la *Process\_Message* la quale prende il messaggio per controllare se l'utente sta già eseguendo un'azione, in caso contrario il messaggio viene inviato a Wit.ai per essere processato dal motore di Natural Language Processing per identificare una possibile azione. Quando l'utente esegue un'azione il bot lo identifica salvando in un database sql lo stato dell'utente, chiamato contesto utente, composto da l'azione eseguita più i suoi dati.

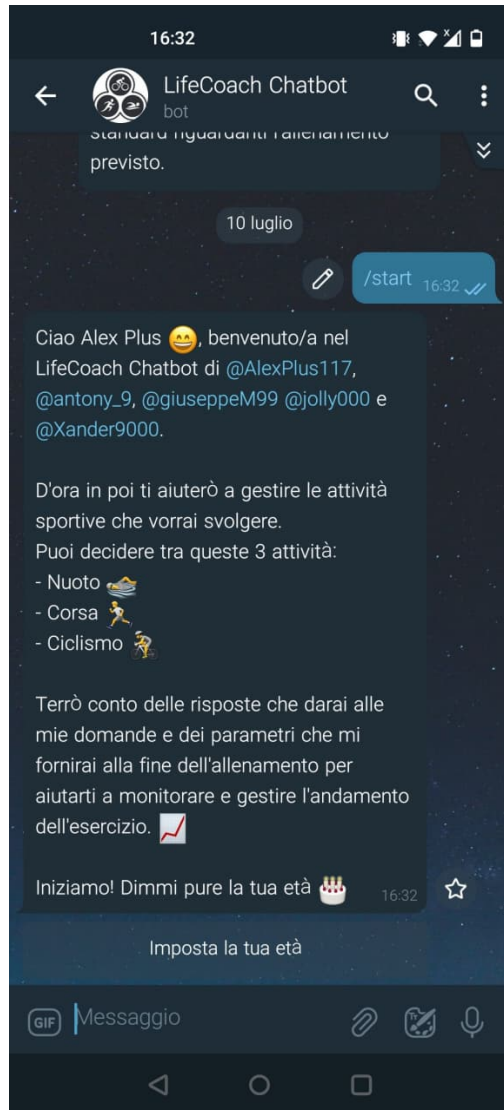
Ogni volta che l'utente modifica l'attività il contesto viene analizzato tramite *Process\_Message*, resettato nel caso in cui l'azione venga terminata o se viene espressa l'intenzione di effettuare una nuova azione. In base al tipo di azione il messaggio viene processato da diverse funzioni che a seconda della necessità acquisiscono i dati dell'utente attraverso le entities di Wit.ai e le salvano in un db specifico per le attività.

Terminato l'inserimento dei dati, all'utente viene notificata l'avvenuto inserimento e il bot attende il tempo prestabilito dell'utente per l'azione e notifica con 5 minuti di anticipo attraverso un metodo timer proprio della libreria di Botogram.

Alcune azioni basilari sono confermate attraverso pulsanti contestuali gestiti direttamente da Telegram e le azioni vengono eseguite tramite apposite funzioni notificate a Botogram analogamente alla definizione dei comandi testuali.

## Esempio d'uso

Dopo aver avviato l'applicazione la schermata dovrebbe presentarsi simile a questa:



Da qui sarà possibile digitare i comandi desiderati scrivendoli ogni volta nella barra della tastiera nella parte bassa della schermata. Come prima cosa consigliamo di digitare il comando **/help** per mostrare i comandi disponibili. Di seguito una breve descrizione delle funzionalità di ogni comando.

**help**→Visualizza la lista dei comandi disponibili.

**start**→Avvia il bot

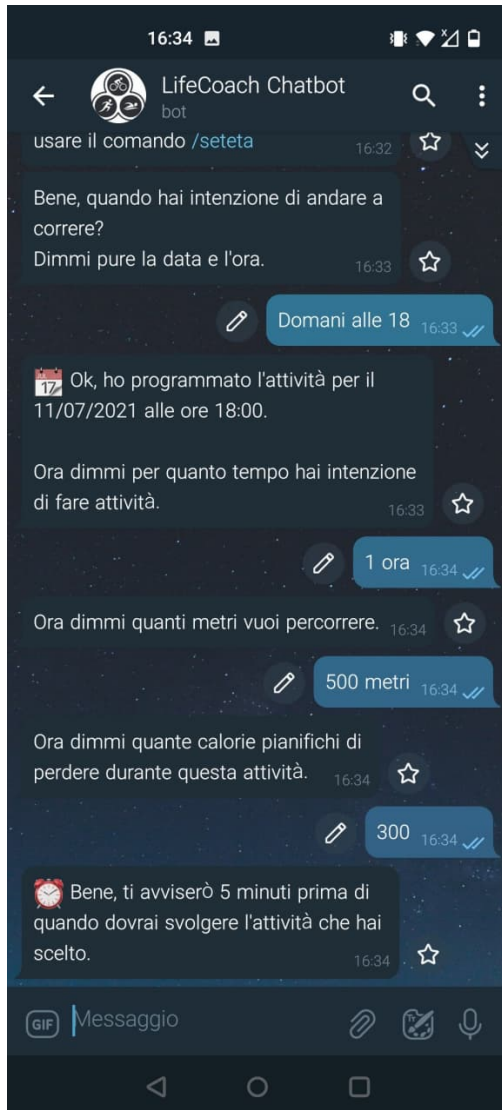
**lista**→Mostra la lista delle attività ancora da svolgere numerandole progressivamente

**elimina**→Elimina una attività. Il comando va usato associandoci il numero dell'attività da eliminare ottenuto visualizzando la lista delle attività.

**seteta**→Imposta o modifica l'età dell'utente

## Fase di impostazione dell'attività

Dopo aver avviato il bot ed aver settato l'età corretta il bot chiederà di selezionare un'attività sportiva tra le tre disponibili (nuoto, corsa e ciclismo). la selezione avviene tramite semplice pressione di bottoni contestuali sotto la richiesta del bot. Selezionata l'attività partirà la fase di settaggio durante la quale verranno richieste diverse informazioni legate all'attività da svolgere e che serviranno al bot per memorizzarla ed offrire supporto al termine della stessa.



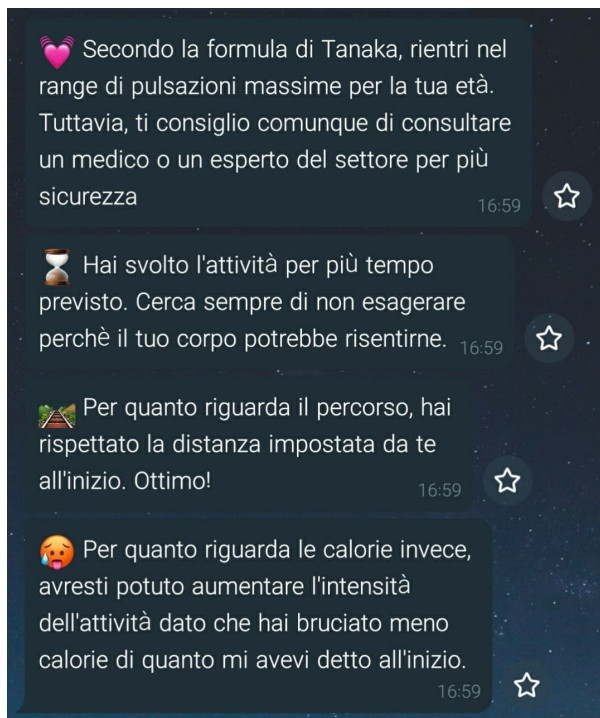
## Calendarizzazione e promemoria attività

Dopo la conferma dell'avvenuto salvataggio dell'attività il bot provvederà a inviare un promemoria poco prima dell'orario di inizio attività stabilito per ricordare l'utente.



## Riepilogo finale

Al termine dell'attività il bot provvederà a chiedere all'utente una serie di dati riguardanti le performance avute durante l'attività sportiva. Elaborandole, provvederà dunque a fornire un risultato approssimato della qualità dell'attività svolta in relazione ai dati precedentemente immessi.



## Processo di sviluppo e organizzazione del lavoro

---

Per le operazioni di organizzazione del lavoro di questa applicazione il nostro gruppo ha utilizzato i seguenti strumenti:

- Telegram
- Discord

Abbiamo optato per l'uso di Discord (una applicazione per la comunicazione VoIP, inizialmente ideata per la comunità videoludica), la quale ci permetteva di comunicare vocalmente tra noi e di condividere lo schermo del nostro computer, per poter seguire il lavoro degli altri o per fornire/chiedere assistenza.

D'altro canto Telegram è stata la scelta più ovvia come ambiente di esecuzione del nostro bot. Infatti, nonostante si possano creare bot per diverse piattaforme tra le quali anche Facebook Messenger, Telegram fa dei bot un suo grosso punto di forza rispetto ad altre app, grazie alla varietà di strumenti di supporto e sviluppo presenti.

Gli incontri erano solitamente organizzati su Discord nel primo pomeriggio, nedurante i quali si proseguiva con lo sviluppo del bot. Telegram è stato invece usato sia per comunicazioni rapide ed informali tramite un gruppo dedicato sia per il test del bot.

## Informazioni

Tutti gli screenshots presenti in questa guida sono tratti dall'esecuzione dell'applicazione su Telegram versione desktop o mobile:



A seconda dell'ambiente di esecuzione e della versione software in uso la rappresentazione grafica potrà apparire leggermente diversa da quella mostrata.