



# HexaTech

*Gruppo HexaTech - Progetto "NaturalAPI"*  
hexatech016@gmail.com

## Manuale Utente

<b>Versione</b>	1.0.0
<b>Approvazione</b>	Gerardo Kokoshari
<b>Redazione</b>	Jacopo Battilana
	Gerardo Kokoshari
	Denis Salviato
	Alessio Barbiero
<b>Verifica</b>	Matteo Brosolo
	Luca Marcon
	Eduard George Serban
	Esterno
<b>Uso</b>	
<b>Destinato a</b>	<i>teal.blue</i>
	Prof. Tullio Vardanega
	Prof. Riccardo Cardin
	Gruppo <i>HexaTech</i>

### Descrizione

Manuale utente relativo al progetto *NaturalAPI* sviluppato dal gruppo *HexaTech*, rivolto agli utenti che vogliono usare il prodotto.

## Registro delle modifiche

Versione	Ruolo	Nominativo	Descrizione	Data
1.0.0	Responsabile di progetto	Eduard George Serban	Approvazione per il rilascio	2020-07-16
0.1.1	Programmatore, Verificatore	Gerardo Kokoshari, Alessio Barbiero	Stesura §3.5 e relativa verifica	2020-07-15
0.1.0	Responsabile di progetto	Gerardo Kokoshari	Rilascio per Revisione di Qualifica	2020-06-06
	Programmatore, Verificatore	Denis Salviato, Matteo Brosolo	Stesura appendice A e relativa verifica	2020-06-04
	Programmatore, Verificatore	Jacopo Battilana, Alessio Barbiero	Stesura §3 e relativa verifica	2020-06-02
	Programmatore, Verificatore	Gerardo Kokoshari, Luca Marcon	Stesura §1 e §2 e relativa verifica	2020-05-31
	Redattore, Verificatore	Jacopo Battilana, Eduard George Serban	Creazione template documento L <sup>A</sup> T <sub>E</sub> X e impostato lo scheletro del documento e relativa verifica	2020-05-24

## Indice

<b>1</b>	<b>Introduzione</b>	<b>6</b>
1.1	Scopo del documento . . . . .	6
1.2	Scopo del prodotto . . . . .	6
1.3	Glossario . . . . .	6
1.4	Comunicazione e malfunzionamenti . . . . .	6
<b>2</b>	<b>Configurazione</b>	<b>7</b>
2.1	Requisiti di sistema . . . . .	7
2.2	Requisiti NaturalAPI Discover . . . . .	7
2.3	Requisiti NaturalAPI Design . . . . .	7
2.4	Requisiti NaturalAPI Develop . . . . .	8
<b>3</b>	<b>Funzionalità ed uso di NaturalAPI</b>	<b>11</b>
3.1	Premessa . . . . .	11
3.2	Informazioni utili . . . . .	11
3.3	File di prova . . . . .	11
3.4	Command Line Interface . . . . .	11
3.4.1	Schermata iniziale . . . . .	11
3.4.2	NaturalAPI Discover . . . . .	12
3.4.3	Controllo di documenti già salvati . . . . .	12
3.4.4	Aggiunta di un documento ".txt" . . . . .	13
3.4.5	Rimozione di un documento . . . . .	16
3.4.6	Confronto tra BDL e scenari Gherkin . . . . .	16
3.4.7	NaturalAPI Design . . . . .	18
3.4.7.1	Controllo di documenti già salvati . . . . .	18
3.4.7.2	Aggiunta di un file Gherkin (.scenario) . . . . .	18
3.4.8	NaturalAPI Develop . . . . .	21
3.4.8.1	Controllo di documenti già salvati . . . . .	21
3.4.8.2	Aggiunta di un file BAL (.json) . . . . .	21
3.5	Graphic User Interface . . . . .	25
3.5.1	Schermata iniziale . . . . .	25
3.5.2	NaturalAPI Discover . . . . .	25
3.5.2.1	Creazione di un BDL partendo da dei documenti ".txt" . . . . .	26
3.5.2.2	Confronto tra BDL e scenari Gherkin . . . . .	31
3.5.3	NaturalAPI Design . . . . .	33
3.5.3.1	Modifica del BAL . . . . .	35
3.5.4	NaturalAPI Develop . . . . .	37
3.5.4.1	Generazione dell'API . . . . .	39
<b>A</b>	<b>Glossario</b>	<b>43</b>
A.1	A . . . . .	43
A.1.1	API . . . . .	43
A.2	B . . . . .	43
A.2.1	BAL . . . . .	43
A.2.2	BDL . . . . .	43
A.3	G . . . . .	43

	A.3.1	Gherkin	43
A.4	J		43
	A.4.1	Java	43
	A.4.2	JavaScript	43
	A.4.3	Json	43
A.5	N		44
	A.5.1	NaturalAPI Design	44
	A.5.2	NaturalAPI Develop	44
	A.5.3	NaturalAPI Discover	44
A.6	P		44
	A.6.1	PLA	44
A.7	S		44
	A.7.1	Scenario	44

## Elenco delle figure

1	Scenario Gherkin . . . . .	8
2	PLA Java . . . . .	10
3	PLA Javascript . . . . .	10
4	Casi d'uso - Schermata iniziale . . . . .	12
5	Casi d'uso - NaturalAPI Discover . . . . .	12
6	Casi d'uso - Estrazione BDL . . . . .	13
7	Output Discover - CSV Sostantivi . . . . .	14
8	Output Discover - CSV Verbi . . . . .	15
9	Output Discover - CSV Predicati . . . . .	15
10	Discover - Rimozione documento . . . . .	16
11	Casi d'uso - NaturalAPI Design . . . . .	18
12	Casi d'uso - Estrazione BAL . . . . .	19
13	Esempio output - BAL . . . . .	20
14	Casi d'uso - NaturalAPI Develop . . . . .	21
15	Casi d'uso - Creazione API . . . . .	22
16	Esempio output - API . . . . .	23
17	Esempio output - Stub dei Test . . . . .	24
18	GUI - Schermata principale . . . . .	25
19	GUI - Navigazione Discover . . . . .	26
20	GUI - Creazione BDL . . . . .	27
21	Output Discover - CSV Sostantivi . . . . .	29
22	Output Discover - CSV Verbi . . . . .	30
23	Output Discover - CSV Predicati . . . . .	30
24	GUI - Check between BDL and Scenarios . . . . .	31
25	GUI - NaturalAPI Design . . . . .	34
26	GUI - Modifica del BAL . . . . .	35
27	GUI - Aggiunta tipo complesso . . . . .	36
28	Esempio output - BAL . . . . .	37
29	GUI - NaturalAPI Develop . . . . .	38
30	GUI - Generazione API . . . . .	39
31	Esempio output - API . . . . .	41
32	Esempio output - Stub dei Test . . . . .	42

# 1 Introduzione

## 1.1 Scopo del documento

Il presente documento ha la finalità di fornire una guida dettagliata all'utente per l'utilizzo del prodotto software *NaturalAPI*. In particolar modo vengono definite tutte le funzionalità che si possono avere nei tre tool che compongono il programma.

## 1.2 Scopo del prodotto

Il prodotto finale mira a creare un toolkit chiamato *NaturalAPI* in modo da ridurre il divario tra i requisiti di un progetto software e la creazione di APIs. Grazie al giusto mix di elaborazione del linguaggio naturale e generazione del codice, *NaturalAPI* consentirà agli sviluppatori software di scrivere APIs che siano più coerenti, più prevedibili e più mantenibili. Scrivendo le funzionalità di un prodotto software in un linguaggio controllato ma vicino al linguaggio naturale, gli stakeholders saranno in grado di trasformare i casi d'uso e i requisiti del progetto in punti di ingresso per il codice effettivo (APIs).

## 1.3 Glossario

Per evitare ogni ambiguità relativa al linguaggio impiegato nel seguente manuale viene fornito, in appendice A, un glossario che raccoglie termini tecnici o riguardanti particolari funzionalità di *NaturalAPI*.

## 1.4 Comunicazione e malfunzionamenti

Nel caso di malfunzionamenti o comportamenti indesiderati durante l'esecuzione di *NaturalAPI*, si invita l'utente a contattare il fornitore al seguente indirizzo e-mail: *hexatech016@gmail.com*. Inoltre si prega d'inserire come oggetto dell'email "MALFUNZIONAMENTO" e di specificare le seguenti informazioni:

- la versione del sistema operativo in uso;
- una breve descrizione del problema riscontrato;
- riportare il contenuto dei messaggi d'errore visualizzati;
- se possibile allegare uno screenshot.

Il fornitore prenderà in carico ogni richiesta o segnalazione effettuata e risponderà nel minor tempo possibile per soddisfare le problematiche riscontrate.

## 2 Configurazione

### 2.1 Requisiti di sistema

Per eseguire NaturalAPI bisogna disporre della JVM (Java Virtual Machine) inclusa in JRE(Java Runtime Environment) che è disponibile gratuitamente nel seguente link: <https://www.java.com/it/download/>. Inoltre si ricorda che il prodotto software è composto da tre tool, ovvero NaturalAPI Discover, NaturalAPI Design e NaturalAPI Develop, ognuno dei quali è indipendente dagli altri.

### 2.2 Requisiti NaturalAPI Discover

NaturalAPI Discover necessita di uno o più file di input con formato “.txt” contenenti i testi dei documenti che si vogliono utilizzare per generare un BDL.

### 2.3 Requisiti NaturalAPI Design

NaturalAPI Design necessita in input di un file con estensione “.scenario ” che rispetti il formato Gherkin e di un BDL. Il primo file deve essere creato manualmente e deve rispettare la struttura e sintassi che si usa nel linguaggio Gherkin mentre il BDL viene generato dal tool Discover. Le seguenti keyword Gherkin sono supportate:

- Scenario;
- Given;
- When;
- Then;
- And.

N.B. la keyword “And” viene usata per poter aggiungere più frasi in Given o Then a seconda delle necessità. Per lo standard Gherkin va utilizzato un singolo When nei propri scenari.

Alcune linee guida per ottenere degli scenari usufruibili al meglio sono:

- usare solo frasi al presente;
- creare frasi con poche subordinate;
- creare frasi in prima persona;
- porre condizioni facilmente verificabili;
- evitare riferimenti alle interfacce.

Un esempio di file .scenario nel formato Gherkin è mostrato qui sotto.

```
Scenario: All done
  Given I am out shopping
  And I have eggs
  And I have milk
  And I have butter
  When I check my list
  Then I don't need anything
```

Figura 1: Scenario Gherkin

## 2.4 Requisiti NaturalAPI Develop

NaturalAPI Develop necessita in input di un BAL e un PLA valido: il primo file viene generato dal tool Design ed è in formato ".json", mentre il secondo è un file ".pla" che specifica la struttura di un linguaggio di programmazione. Dei PLA validi sono precaricati di default in *NaturalAPI* e specificano il linguaggio Java o JavaScript. Se invece si vuole utilizzare un linguaggio differente bisogna creare un file di testo che contiene le specifiche del linguaggio desiderato. Per strutturare correttamente il file è necessaria la presenza di alcune parole chiave, suddivise in sezioni. L'intestazione del PLA richiede la specifica di alcune nomenclature necessarie per la corretta generazione dell'API e prevede i seguenti tag:

- **<--classComment-->**: indica dove inserire un eventuale commento descrittivo della classe;
- **<--classExtension: xyz**: indica l'estensione dei file generati dal PLA. Il parametro "xyz" va sostituito con l'estensione corretta da utilizzare;
- **<--integer: xyz**: descrive il tipo integer nel codice. Il parametro "xyz" va sostituito con la nomenclatura corretta da utilizzare. Nel caso di linguaggio non tipizzato va usato "/" al posto del nome;
- **<--float: xyz**: descrive il tipo float nel codice. Il parametro "xyz" va sostituito con la nomenclatura corretta da utilizzare. Nel caso di linguaggio non tipizzato va usato "/" al posto del nome;
- **<--string: xyz**: descrive il tipo string nel codice. Il parametro "xyz" va sostituito con la nomenclatura corretta da utilizzare. Nel caso di linguaggio non tipizzato va usato "/" al posto del nome;
- **<--boolean: xyz**: descrive il tipo boolean nel codice. Il parametro "xyz" va sostituito con la nomenclatura corretta da utilizzare. Nel caso di linguaggio non tipizzato va usato "/" al posto del nome.

Il corpo del PLA contiene invece le informazioni necessarie a strutturare correttamente il codice nel linguaggio di programmazione desiderato. Esso prevede l'utilizzo dei tag di seguito descritti:



- `<--struct.start-->`: indica dove cominciano le istruzioni per la costruzione delle strutture;
- `<--structName-->`: indica dove posizionare il nome della struttura;
- `<--structParamType-->`: indica dove posizionare il tipo dei parametri della struttura;
- `<--structParamName-->`: indica dove posizionare il nome dei parametri della struttura;
- `<--struct.end-->`: indica dove finiscono le istruzioni per la costruzione delle strutture;
- `<--method.start-->`: indica dove cominciano le istruzioni per la costruzione dei metodi;
- `<--methodReturn-->`: indica dove posizionare il tipo di ritorno del metodo;
- `<--methodName-->`: indica dove posizionare il nome del metodo;
- `<--methodParamType-->`: indica dove posizionare il tipo dei parametri del metodo;
- `<--methodParamName-->`: indica dove posizionare il nome dei parametri del metodo;
- `<--methodComment-->`: indica dove inserire un eventuale commento descrittivo del metodo;
- `<--method.end-->`: indica dove finiscono le istruzioni per la costruzione dei metodi;
- `<--className-->`: indica dove posizionare il nome della classe;
- `<--struct.here-->`: indica dove posizionare le strutture nel codice, formate secondo lo schema definito;
- `<--method.here-->`: indica dove posizionare i metodi nel codice, formati secondo lo schema definito.

Due esempio di PLA, per il linguaggio di programmazione Java e JavaScript, già presenti in NaturalAPI Develop, sono mostrati nelle due figure seguenti.

```

/*
  <--classComment-->
  <--classExtension: java
  <--integer: int
  <--float: float
  <--string: String
  <--boolean: boolean
*/

<--struct.start-->
private class <--structName-->{
  <--structParamType--> <--structParamName-->
}
<--struct.end-->

<--method.start-->
public <--methodReturn--> <--methodName-->(<--methodParamType--><--methodParamName-->){
  <--methodComment-->
}
<--method.end-->

public class <--className-->{

  <--struct.here-->

  <--method.here-->
}

```

Figura 2: PLA Java

```

/*
  <--classComment-->
  <--classExtension: js
  <--integer: /
  <--float: /
  <--string: /
  <--boolean: /
*/

<--struct.start-->
function <--structName-->(<--structParamName-->){
}
<--struct.end-->

<--method.start-->
<--methodName-->(<--methodParamName-->){
  <--methodComment-->
  <--methodReturn-->
}
<--method.end-->

<--struct.here-->

class <--className-->{

  constructor(){
  }

  <--method.here-->
}

```

Figura 3: PLA Javascript

## 3 Funzionalità ed uso di NaturalAPI

### 3.1 Premessa

NaturalAPI è un'applicazione con tre funzionalità principali. Ogni macro funzionalità viene rappresentata da un tool ed ognuno di essi svolge uno specifico scopo:

- **NaturalAPI Discover:** cerca i termini più utilizzati all'interno di un documento di testo prelevando i verbi, sostantivi e i predicati. I termini individuati con un numero di ricorrenza accettabile vengono utilizzati per creare una Business Domain Language (BDL).
- **NaturalAPI Design:** permette di importare e convertire un documento nel formato Gherkin con estensione scenario e un BDL di supporto in una Business Application Language (BAL).
- **NaturalAPI Develop:** consente tramite l'importazione di un BAL di generare una API in un linguaggio di programmazione specifico. Fornisce anche la possibilità di aggiungere una Business Ontology che descriva strutture dati complesse da utilizzare nel codice prodotto. Offre anche la possibilità di utilizzare un linguaggio di output esterno scelto dall'utente, importando un corretto file con estensione pla.

### 3.2 Informazioni utili

Il programma integra una guida che può essere richiamata tramite un comando se si usa la cli o premendo un pulsante se si usa la gui, queste guide permettono all'utente di avere dei suggerimenti o una breve descrizione delle funzionalità del tool che si sta usando.

### 3.3 File di prova

Per poter eseguire tutti i tool in sequenza vengono usati come esempio due file, il primo con estensione ".txt" che contiene un testo appartenente al dominio "streaming" e il secondo con estensione ".scenario" che contiene delle features appartenente allo stesso dominio del precedente file.

### 3.4 Command Line Interface

#### 3.4.1 Schermata iniziale

Per avviare il programma bisogna posizionarsi da console nella cartella contenente il file "NaturalAPICLI.jar" ed eseguire il comando `java -jar NaturalAPICLI.jar`.

Una volta avviato correttamente NaturalAPI comparirà la schermata iniziale dove sarà possibile scegliere il tool che si desidera utilizzare:

- **1:** Discover;
- **2:** Design;
- **3:** Develop;
- **4:** guida;

- 5: chiusura del software.

```
                                --NATURAL API--  
Use case:  
1: Discover  
2: Design  
3: Develop  
4: Guide  
5: Exit
```

Figura 4: Casi d'uso - Schermata iniziale

### 3.4.2 NaturalAPI Discover

Per selezionare il tool Discover è sufficiente digitare "1". Viene quindi mostrato a schermo il relativo menù contenente i casi d'uso del tool Discover con le seguenti opzioni:

- 1: controllo di documenti già salvati;
- 2: aggiunta di un documento ".txt";
- 3: confronto tra BDL e scenari Gherkin;
- 4: rimozione di un documento;
- 5: guida;
- 6: ritorno alla schermata iniziale.

```
                                --DISCOVER--  
Use case:  
1: Check for saved documents  
2: Add a document (.txt)  
3: Check between BDL and Gherkin  
4: Remove a document  
5: Guide  
6: Back
```

Figura 5: Casi d'uso - NaturalAPI Discover

### 3.4.3 Controllo di documenti già salvati

Digitando il tasto "1" viene avviata una procedura automatica che controlla la presenza di un file di backup. In caso la procedura non trovi documenti salvati si viene avvisati dell'assenza del documento, ripresentando poi il menù iniziale del tool Discover.

Nel caso esistono documenti già salvati in precedenza viene mostrato un messaggio, con affiancate le risposte ammissibili, che chiede se si vuole caricare tale documento.

- Per confermare il caricamento digitare "Y". Terminata l'operazione di caricamento viene mostrato il menù di creazione del BDL;
- per annullare il caricamento digitare "N". Viene quindi eliminato il file di backup e viene ritornato il menù iniziale del tool Discover.

Nel seguente esempio questa funzionalità non viene usata in quanto il file di prova non è stato ancora memorizzato nel programma.

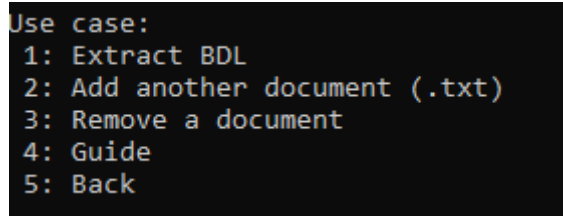
#### 3.4.4 Aggiunta di un documento ".txt"

Digitando il tasto "2" viene richiesto in input il percorso di sistema in cui è presente il file ".txt", prendiamo come esempio il documento ".txt" descritto nel paragrafo 3.3, inseriamo quindi un percorso generico dove è presente questo ipotetico file (ex. C:\Users\User\Desktop\esempio.txt).

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".txt" viene restituito un messaggio di errore e si è riportati al menù iniziale del tool Discover;
- Nel caso in cui il percorso inserito sia corretto viene visualizzato il menù di estrazione del BDL.

Nel seguente esempio il file di prova viene accettato e quindi viene riportato il seguente menù:

- 1: estrarre il BDL dai documenti caricati;
- 2: aggiungere un ulteriore documento per l'estrazione del BDL;
- 3: rimozione di un documento;
- 4: guida;
- 5: ritornare al menù principale del tool.



```
Use case:
1: Extract BDL
2: Add another document (.txt)
3: Remove a document
4: Guide
5: Back
```

Figura 6: Casi d'uso - Estrazione BDL

- Per estrarre il BDL dal documento di testo inserito è sufficiente digitare il tasto "1" e inserire il nome del file che si vuole generare nel output in questo caso scriviamo "esempioBdl". Dopo una breve attesa vengono creati nella directory del progetto, una cartella Discover, contenente un file ".BDL", un file ".BO" e tre file ".csv", contenenti delle liste leggibili dei sostantivi, dei verbi e dei predicati presenti nel BDL. I tre file ".csv" vengono aperti con il relativo programma impostato di default. Si viene quindi riportati al menù iniziale del tool Discover;
- Se si vuole aggiungere ulteriori documenti, basterà digitare "2" e la procedura è analoga a quella descritta precedentemente dal primo file inserito, per semplicità in questo esempio abbiamo usato un solo file di prova;
- Nel caso in cui si sia caricato un file scorretto si può rimuovere premendo "3" e la procedura è descritta nella sezione successiva.

Il file esempioBdl.BDL generato precedentemente tramite l'importazione del file esempio.txt ha la seguente struttura:

```
Streaming,99
Media,54
Music,51
Video,50
User,34
Stream,32
Protocol,31
Content,28
Napster,27
Network,27
Service,22
Internet,22
File,19
Format,18
Server,18
Computer,17
Time,13
Revenue,12
Industry,11
Bandwidth,11
Demand,11
Audio,11
System,11
Group,10
Player,10
Platform,10
Delivery,10
Television,10
Site,9
Percent,8
Application,8
Recording,8
Year,8
Access,7
Record,7
TV,7
Mbit,7
Number,7
Technology,7
Transport,7
P2p,7
Business,7
Datum,7
Example,7
```

Figura 7: Output Discover - CSV Sostantivi

```

Stream,35
Become,10
Base,10
Require,9
Send,8
Include,7
Enable,6
Develop,6
Increase,5
Introduce,5
Live,5
Lead,5
Share,5
Play,5
Deliver,5
Save,4
Create,4
Grow,4
Offer,4
Code,4
Buffer,4
Provide,4
Allow,4
Copyright,4
Download,4
Reduce,4

```

Figura 8: Output Discover - CSV Verbi

```

Use protocol,4
Send stream,4
Stream content,3
Display media,2
Require bandwidth,2
Grow percent,2
Save file,2
Use format,2
Enable streaming,2
Use which,2
Detect loss,2
Make copy,2
Use player,2
Play video,2
Search multimedia,1
Deliver media,1
File downloading,1
Obtain file,1
Have length,1
Require form,1
Digitize content,1
Distribute content,1
Lack bandwidth,1
Experience stop,1
Slow buffering,1
Lack system,1
Include Netflix,1
Live which,1
Live site,1
Push syndication,1
Grant patent,1
Have bandwidth,1
Support rate,1
Create path,1
Prevent underrun,1
Download file,1
Save it,1
Store it,1

```

Figura 9: Output Discover - CSV Predicati

### 3.4.5 Rimozione di un documento

Digitando il tasto "4" (oppure il tasto "3" nel caso del menù di estrazione del BDL) viene esposta una lista di documenti caricati e viene chiesto all'utente di scegliere il numero del documento da rimuovere. In caso non si voglia eliminare un documento è sufficiente digitare "q".

```
The following document(s) are already stored:
1. C:\Users\batti\OneDrive\Desktop\NaturalAPI\Streaming Video\streaming_video_wiki.txt
2. C:\Users\batti\OneDrive\Desktop\NaturalAPI\Streaming Video\streaming_video_wiki2.txt

Which document you want to delete?
Insert the number of the document you want to remove or
type 'q' to go back.
```

Figura 10: Discover - Rimozione documento

### 3.4.6 Confronto tra BDL e scenari Gherkin

Digitando il tasto "3" viene effettuato un confronto tra BDL e scenari Gherkin. Per effettuare questo confronto bisogna prima importare un BDL:

- **BDL esterno:** viene richiesto in input il percorso di sistema in cui è presente il file ".BDL", fornendone un possibile esempio (ex. C:\Users\User\Desktop\example.BDL).
  - Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".BDL" viene restituito un messaggio di errore e si è riportati al menù di caricamento del file ".BDL";
  - nel caso in cui il caricamento vada a buon fine viene visualizzato il menù di importazione dello scenario Gherkin;
- **BDL già creato:** viene controllato se è presente un BDL nella cartella di output di Discover e, in caso positivo, viene importato. In caso negativo l'utente viene riportato alla scelta di importazione del BDL.

Si ricorda che nel seguente esempio è già stato creato un file BDL nella cartella Discover, quindi in questo caso basterà selezionare l'opzione "2" e il programma troverà in automatico il file "esempioBDL.BDL". Dopo aver importato correttamente il BDL viene richiesto di importare uno scenario Gherkin in maniera analoga all'importazione dei documenti precedenti. Viene quindi richiesto in input il percorso di sistema in cui è presente il file ".scenario", prendiamo come esempio il documento ".scenario" descritto nel paragrafo 3.3, inseriamo quindi un percorso generico dove è presente questo ipotetico file (ex. C:\Users\User\Desktop\esempio.scenario).

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".scenario" viene restituito un messaggio di errore e si è riportati al menù di caricamento del file ".scenario";
- nel caso in cui il caricamento vada a buon fine viene restituita una percentuale rappresentante l'affinità fra gli scenari Gherkin scritti e il BDL utilizzato e viene realizzato un file di log contenente i dettagli del confronto.

Nel seguente esempio il file di prova con estensione ".scenario" viene accettato, ciò comporta che verrà realizzato il file di log, che rappresenta il confronto tra "esempioBDL.BDL" ed "esempio.scenario". Di seguito viene mostrato il file log generato.



## NOUNS:

The following nouns you use in your Gherkin correspond to the Business Domain Language:

video; stream;

You may need to use the following nouns in your Gherkin:  
streaming; media; music; user; protocol; content; Napster;  
network; service; internet; file; format; server; computer;

You are using more than one time the following nouns but they don't correspond to the Business Domain Language:  
volume; app; account; command;

## VERBS:

The following verbs you use in your Gherkin correspond to the Business Domain Language:  
stream; play; access;

You may need to use the following verbs in your Gherkin:  
become; base; require; send; include; enable; develop;  
increase; introduce; live; lead; share; deliver; save;  
create; grow; offer; code; buffer; provide; allow; copyright;  
download; reduce; pioneer; transform; launch; acquire; own  
; fall; encode; follow; compress; emerge; interact; mean;  
cause; view; call; apply; refer; distribute;

You are using more than one time the following verbs but they don't correspond to the Business Domain Language:  
select; start; input; press;

## PREDICATES:

The following predicates you use in your Gherkin correspond to the Business Domain Language:  
play video;

You may need to use the following predicates in your Gherkin:  
use protocol; send stream; stream content; display media;  
require bandwidth; grow percent; save file; use format;  
enable streaming; use which; detect loss; make copy; use player;

You are using more than one time the following predicates but they don't correspond to the Business Domain Language:  
select video; start app; stream video; change volume;

### 3.4.7 NaturalAPI Design

Per selezionare il tool Design è sufficiente digitare "2". Viene quindi mostrato a schermo il relativo menù contenente i casi d'uso del tool Design con le seguenti opzioni:

- **1:** controllo di documenti già salvati;
- **2:** aggiunta di un file Gherkin (.scenario);
- **3:** guida;
- **4:** ritorno alla schermata iniziale,

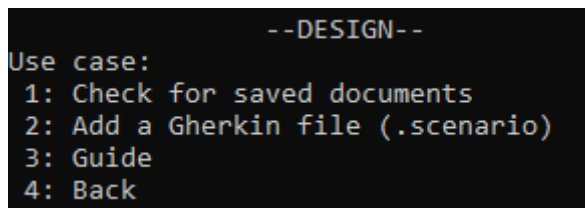


Figura 11: Casi d'uso - NaturalAPI Design

#### 3.4.7.1 Controllo di documenti già salvati

Digitando "1" viene avviata una procedura automatica che controlla la presenza di un file di backup. In caso la procedura non trovi documenti salvati allerta l'utente dell'assenza del documento, ripresentando poi il menù iniziale del tool Design.

Nel caso esistono documenti già salvati in precedenza viene mostrato un messaggio, con affiancate le risposte ammissibili, che chiede se si vuole caricare tale documento.

- Per confermare il caricamento digitare "Y". Terminata l'operazione di caricamento viene mostrato il menù di input ;
- per annullare il caricamento digitare "N". Vengono quindi eliminati i file di backup e viene ritornato il menù iniziale del tool Design.

Nel seguente esempio questa funzionalità non viene usata in quanto il file di prova non è stato ancora memorizzato nel programma.

#### 3.4.7.2 Aggiunta di un file Gherkin (.scenario)

Digitando "2" Viene richiesto in input il percorso di sistema in cui è presente il documento ".scenario", prendiamo come esempio il documento ".scenario" descritto nel paragrafo 3.3, inseriamo quindi un percorso generico dove è presente questo ipotetico file (ex. C:\Users\User\Desktop\esempio.scenario).

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".scenario" viene restituito un messaggio di errore e si è riportati al menù iniziale del tool Design;
- Nel caso in cui il caricamento vada a buon fine viene visualizzato un menù per il caricamento di un BO[opzionale] e di un BAL.

Nel seguente esempio il file di prova viene accettato e quindi viene riportato il seguente menù:

- **1**: aggiungere una Business Ontology [opzionale];
- **2**: estrarre il BAL;
- **3**: guida;
- **4**: ritornare al menù principale del tool.

```
Use case:  
1: Add a Business Ontology (.json) [optional]  
2: Extract BAL  
3: Guide  
4: Back
```

Figura 12: Casi d'uso - Estrazione BAL

Digitando il tasto "1" si può aggiungere una Business Ontology (BO) che permette al programma di proporre degli oggetti complessi che potrebbero essere attinenti al BAL che si sta creando. Per aggiungere una BO viene richiesto in input il percorso di sistema in cui è presente il documento ".json", fornendone un possibile esempio (ex. C:\Users\User\Desktop\example.json).

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".json" viene restituito un messaggio di errore e si è riportati al menù di estrazione del BAL.
- Nel caso in cui il caricamento vada a buon fine viene avviata l'estrazione del BAL.

Nel seguente esempio per semplicità non inseriamo nessun tipo di BO e procediamo direttamente all'estrazione del file output. Per poterlo fare è sufficiente digitare "2" e inserire il nome del file che si vuole generare, in questo caso scriviamo "esempioBAL". Dopo una breve attesa viene creato un file ".json" nella directory del progetto, nella cartella Design. Durante la creazione viene data la possibilità di modificare il tipo di ritorno dei metodi e il tipo dei parametri:

- **Y**: permette di cambiare il tipo scegliendo fra:
  - **v**: void (non disponibile per i parametri);
  - **s**: string;
  - **i**: integer;
  - **f**: float;
  - **b**: boolean;
  - **c**: complex object:
    - \* **0**: permette di creare un nuovo oggetto complesso, scegliendone nomi e tipi dei parametri contenuti;
    - \* **1+**: sono presenti solo se è stata aggiunta una BO, o se sono già stati creati dei tipi complessi.

Dopo ogni cambiamento viene sempre chiesto se il tipo modificato è un array o meno:

- **Y**: il tipo è un array;
- **N**: il tipo non è un array;
- **N**: il tipo rimane quello mostrato in linea e viene proposto il prossimo cambiamento finché non vengono esauriti.

Una volta terminata la procedura di modifica dei tipi viene terminata l'estrazione del BAL e viene aperto usando il programma impostato di default, subito dopo si viene riportati al menù principale del tool Design. Di seguito viene mostrata una sezione del file esempioBAL.BAL generato tramite l'importazione del file di prova ".scenario".

```
{
  "openapi" : "3.0.0",
  "info" : {
    "version" : "1.0.0",
    "title" : "streaming",
    "license" : {
      "name" : "MIT"
    }
  },
  "paths" : {
    "/chooseVideo" : {
      "get" : {
        "operationId" : "chooseVideo",
        "description" : "Scenario: select video---Given I started the app correctly---A",
        "parameters" : [ {
          "in" : "query",
          "name" : "video",
          "description" : "Default",
          "required" : true,
          "schema" : {
            "type" : "string"
          }
        } ],
        "responses" : {
          "200" : {
            "description" : "page ",
            "content" : {
              "/*/*" : {
                "schema" : {
                  "type" : "string"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Figura 13: Esempio output - BAL

### 3.4.8 NaturalAPI Develop

Per selezionare il tool Develop è sufficiente digitare "3". Viene quindi mostrato a schermo il relativo menù contenente i casi d'uso del tool Develop con le seguenti opzioni:

- **1:** controllo di documenti già salvati;
- **2:** aggiunta di un file BAL (.json);
- **3:** guida;
- **4:** ritorno alla schermata iniziale

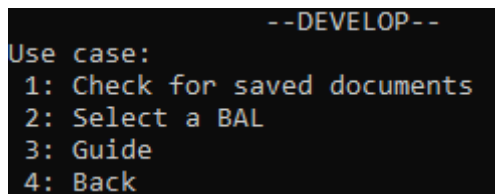


Figura 14: Casi d'uso - NaturalAPI Develop

#### 3.4.8.1 Controllo di documenti già salvati

Digitando il tasto "1" viene avviata una procedura automatica che controlla la presenza di un file di backup. In caso la procedura non trovi documenti salvati allerta l'utente dell'assenza del documento, ripresentando poi il menù iniziale del tool Develop.

Nel caso esistono documenti già salvati in precedenza viene mostrato un messaggio, con affiancate le risposte ammissibili, che chiede se si vuole caricare tale documento.

- Per confermare il caricamento digitare "Y". Terminata l'operazione di caricamento viene mostrato il menù di creazione dell'API;
- per annullare il caricamento digitare "N". Vengono quindi eliminati i file di backup e viene ritornato il menù iniziale del tool Develop.

Nel seguente esempio questa funzionalità non viene usata in quanto il file di prova non è stato ancora memorizzato nel programma.

#### 3.4.8.2 Aggiunta di un file BAL (.json)

Digitando il tasto "2" Viene richiesto in input il percorso di sistema in cui è presente il documento ".json", prendiamo come esempio il file generato precedentemente nel tool di Design, inseriamo quindi il percorso dove è stato generato (ex. C:\Users\User\Desktop\esempioBAL.json).

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".json" viene restituito un messaggio di errore e si è riportati al menù iniziale del tool Develop;
- Nel caso in cui il caricamento vada a buon fine viene visualizzato un menù per la creazione dell'API.

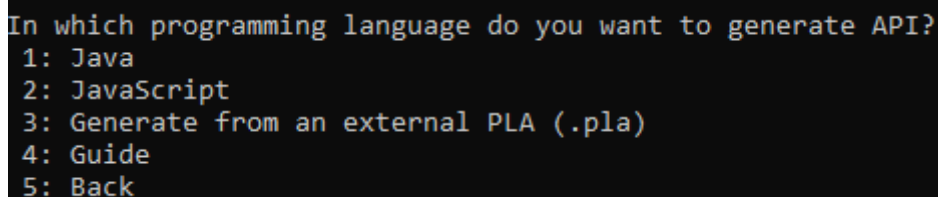
Nel seguente esempio il file viene accettato e quindi viene riportato il seguente menù :

- **1:** creare l'API, scegliendo un PLA per specificare il linguaggio;

- **2:** annullare la creazione.

Digitando "1" si va a creare un API tramite l'uso di un PLA che va a definire in quale linguaggio viene prodotta, di default i linguaggi disponibili sono java e javascript. Comparirà quindi un menù con le seguenti scelte:

- **1:** Java;
- **2:** Javascript;
- **3:** PLA esterno;
- **4:** guida;
- **5:** ritorno al menù principale del tool.



```
In which programming language do you want to generate API?  
1: Java  
2: JavaScript  
3: Generate from an external PLA (.pla)  
4: Guide  
5: Back
```

Figura 15: Casi d'uso - Creazione API

Digitando "1" o "2" si andrà a creare il corpo dell'API ed i relativi stub dei test, mentre digitando "3" si andrà a inserire un nuovo PLA che genererà un API e gli stub dei relativi test, in base al tipo di linguaggio di programmazione specificato nel file inserito. Una volta creata l'API in un linguaggio è possibile selezionarne un altro o importare un altro PLA. Una volta soddisfatti si può digitare "5" per uscire. Nel seguente esempio scegliamo di creare un API tramite il linguaggio di default Java che genera quindi un file di tipo Java denominato esempioBAL.java. Di seguito vengono mostrati i file esempioBAL.java ed esempioBALTest.java creati tramite l'importazione del file "esempioBAL.json".

```
/*
  <--classExtension: java
  <--integer: int
  <--float: float
  <--string: String
  <--boolean: boolean
*/

public class Streaming{

  private class Music{
    String length;
    String format;
  }

  private class Storage{
    String volume;
  }

  public String chooseVideo(String video){
    //page - chooseVideo
  }

  public String giveCommand(String video, String command){
    //video - giveCommand
  }

  public String inputPassword(String password, String username){
    //access - inputPassword
  }

  public String pressVideo(String video){
    //video - pressVideo
  }

  public String haveBandwith(String bandwith, String video){
    //error - haveBandwith
  }

  public String pressCommand(String command){
    //volume - pressCommand
  }

}
```

Figura 16: Esempio output - API

```
public class StreamingTest{

    @Given I started the app correctly
    public void givenIStartedTheAppCorrectly(){
        //Insert a test implementation
    }

    @And I want to watch a video
    public void andIWantToWatchAVideo(){
        //Insert a test implementation
    }

    @When I choose the video
    public void whenIChooseTheVideo(){
        //Insert a test implementation
        Streaming.chooseVideo();
    }

    @Then I will get the page of the video )
    public void thenIWillGetThePageOfTheVideo(){
        //Insert a test implementation
    }

    @Given I selected a video )
    public void givenISelectedAVideo(){
        //Insert a test implementation
    }

    @And I want to stream the video
    public void andIWantToStreamTheVideo(){
        //Insert a test implementation
    }

    @When I selected a video
    public void whenISelectedAVideo(){
        //Insert a test implementation
        Streaming.giveCommand();
    }

    @And I give the play command
    public void andIGiveThePlayCommand(){
        //Insert a test implementation
    }

    @Then The app will start the video
    public void thenTheAppWillStartTheVideo(){
        //Insert a test implementation
    }
}
```

Figura 17: Esempio output - Stub dei Test



## 3.5 Graphic User Interface

### 3.5.1 Schermata iniziale

Per avviare il programma bisogna posizionarsi da console nella cartella contenente il file "NaturalAPIGUI.jar" ed eseguire il comando `java -jar NaturalAPIGUI.jar`. Alternativamente è sufficiente fare doppio click sul file "NaturalAPIGUI.jar".

Una volta avviato correttamente NaturalAPI comparirà la finestra iniziale dove sarà possibile scegliere tramite dei pulsanti il tool che si desidera utilizzare:

- 1: Discover;
- 2: Design;
- 3: Develop;
- 4: guida.

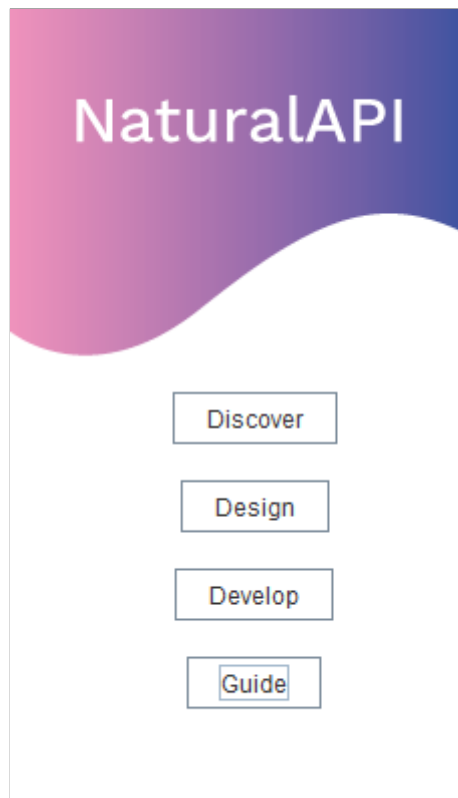


Figura 18: GUI - Schermata principale

Inoltre sarà disponibile su tutti i tool e anche in questa finestra un pulsante dedicato alla guida che se cliccato fornirà suggerimenti e aiuti in caso di bisogno.

### 3.5.2 NaturalAPI Discover

Per selezionare il tool Discover è sufficiente premere il pulsante "Discover". Viene quindi mostrata una nuova schermata che permette di scegliere il caso d'uso del tool Discover desiderato:

- 1: creazione di un BDL partendo da dei documenti ".txt";
- 2: confronto tra BDL e scenari Gherkin;
- 3: guida;
- 4: ritorno alla schermata iniziale.

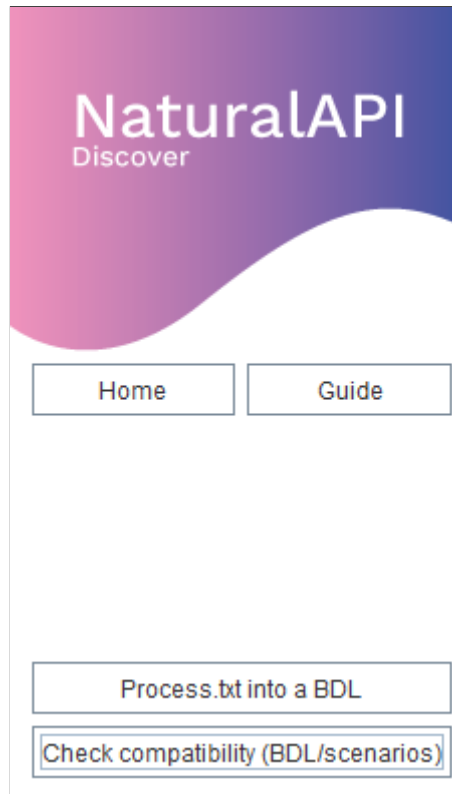


Figura 19: GUI - Navigazione Discover

#### 3.5.2.1 Creazione di un BDL partendo da dei documenti ".txt"

Premendo il pulsante "Process .txt into a BDL" viene mostrata una schermata dedicata alla creazione di un BDL da dei file ".txt" in input.

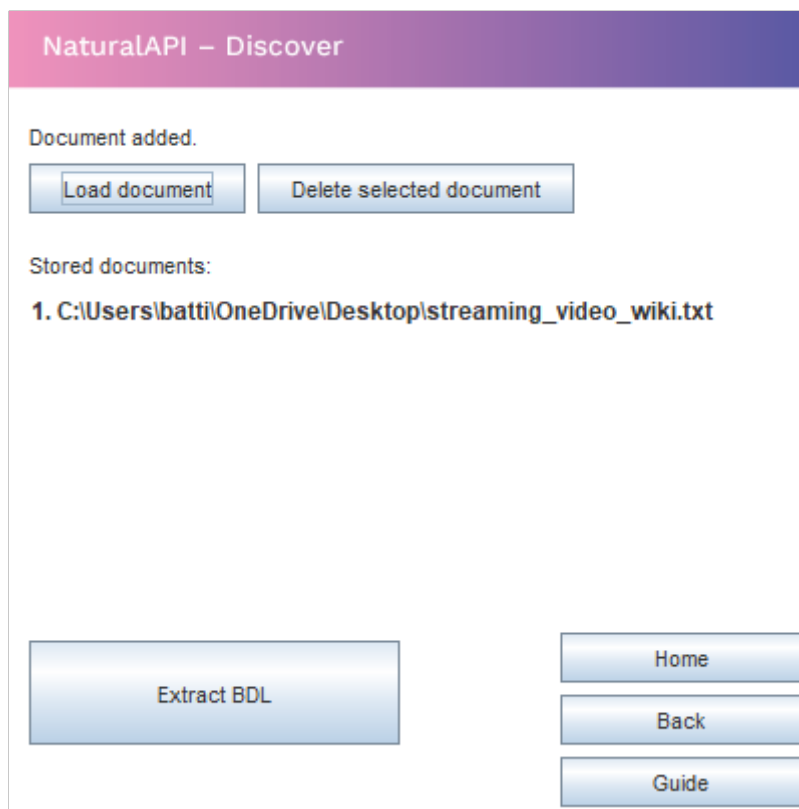


Figura 20: GUI - Creazione BDL

Il programma controllerà in automatico se sono presenti dei file di backup da utilizzare come input. In caso il programma trovi dei file, prima della comparsa della finestra dedicata a Discover, eseguirà un messaggio pop-up che segnalerà la presenza di file già presenti in memoria.

- Nel caso in cui si voglia confermare l'utilizzo di questi file basterà selezionare il tasto "yes".
- Nel caso in cui si voglia rifiutare l'utilizzo di questi file basterà selezionare il tasto "no".

Nella schermata dedicata all'estrazione del BDL sono presenti i seguenti elementi:

- bottone per caricare un documento ".txt";
- bottone per eliminare un documento ".txt";
- lista raffigurante i documenti caricati;
- bottone per estrarre il BDL;
- bottone per ritornare alla schermata iniziale.
- bottone per ritornare alla schermata precedente.
- bottone per aprire un suggerimento.

Cliccando il bottone "Load document" si aprirà una finestra dove sarà richiesto in input di selezionare il percorso di sistema in cui è presente il file ".txt", prendiamo come esempio il documento ".txt" descritto nel paragrafo 3.3, una volta selezionato e caricato nel programma premiamo il pulsante "Confirm" della finestra.

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista, l'estensione indicata non sia ".txt" o il file sia già stato caricato viene restituito un messaggio pop-up di errore;
- Nel caso in cui il documento sia corretto viene visualizzato il percorso e il nome del file nella sezione della finestra dedicata e viene attivato il pulsante dedicato all'estrazione del BDL;

In caso si voglia rimuovere un documento caricato bisogna selezionarlo nella lista dei documenti e in seguito premere il bottone "Delete selected document". Se nessun documento è stato selezionato viene aperto un pop-up di errore.

Nel seguente esempio il file di prova viene accettato e quindi basterà selezionare il pulsante "ExtractBDL" che procederà con l'estrazione del BDL. Di seguito sono descritti brevemente gli step che il programma esegue per estrarre il BDL.

- **1:** Viene richiesto di inserire il nome del nuovo BDL tramite un messaggio pop-up;
- **2:** Il programma genererà il BDL ed i relativi file ".csv" nella cartella "Discover";
- **3:** Il programma genererà una Business Ontology nella cartella "Discover";
- **4:** Il programma farà avviare il programma di default e mostrerà il BDL creato.

Di seguito vengono mostrati i file esempioBdl.BDL generati tramite l'importazione del file esempio.txt.

```

Streaming,99
Media,54
Music,51
Video,50
User,34
Stream,32
Protocol,31
Content,28
Napster,27
Network,27
Service,22
Internet,22
File,19
Format,18
Server,18
Computer,17
Time,13
Revenue,12
Industry,11
Bandwidth,11
Demand,11
Audio,11
System,11
Group,10
Player,10
Platform,10
Delivery,10
Television,10
Site,9
Percent,8
Application,8
Recording,8
Year,8
Access,7
Record,7
TV,7
Mbit,7
Number,7
Technology,7
Transport,7
P2p,7
Business,7
Datum,7
Example,7

```

Figura 21: Output Discover - CSV Sostantivi

```

Stream,35
Become,10
Base,10
Require,9
Send,8
Include,7
Enable,6
Develop,6
Increase,5
Introduce,5
Live,5
Lead,5
Share,5
Play,5
Deliver,5
Save,4
Create,4
Grow,4
Offer,4
Code,4
Buffer,4
Provide,4
Allow,4
Copyright,4
Download,4
Reduce,4

```

Figura 22: Output Discover - CSV Verbi

```

Use protocol,4
Send stream,4
Stream content,3
Display media,2
Require bandwidth,2
Grow percent,2
Save file,2
Use format,2
Enable streaming,2
Use which,2
Detect loss,2
Make copy,2
Use player,2
Play video,2
Search multimedia,1
Deliver media,1
File downloading,1
Obtain file,1
Have length,1
Require form,1
Digitize content,1
Distribute content,1
Lack bandwidth,1
Experience stop,1
Slow buffering,1
Lack system,1
Include Netflix,1
Live which,1
Live site,1
Push syndication,1
Grant patent,1
Have bandwidth,1
Support rate,1
Create path,1
Prevent underrun,1
Download file,1
Save it,1
Store it,1

```

Figura 23: Output Discover - CSV Predicati

### 3.5.2.2 Confronto tra BDL e scenari Gherkin

Premendo il pulsante "Check compatibility (BDL/scenarios)" viene mostrata una schermata dedicata al confronto tra BDL e scenari Gherkin.

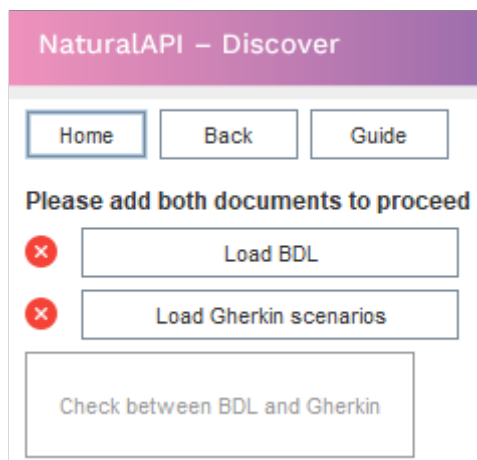


Figura 24: GUI - Check between BDL and Scenarios

Il contenuto della schermata è il seguente

- bottone per il ritorno alla schermata iniziale;
- bottone per il ritorno alla schermata precedente;
- bottone per aprire un suggerimento;
- bottone per l'aggiunta di un BDL;
- bottone per l'aggiunta di uno scenario Gherkin;
- bottone per confrontare BDL e scenari Gherkin;

Selezionando il pulsante "Load BDL" si aprirà una finestra dove sarà richiesto in input di selezionare il percorso di sistema in cui è presente il file ".BDL", nel seguente esempio prendiamo il file "esempioBDL.BDL" generato precedentemente nella cartella "Discover" del programma e quindi immaginiamo di averlo selezionato e caricato nel programma premendo il tasto conferma della finestra.

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".BDL" viene restituito un messaggio pop-up di errore;
- Nel caso in cui il documento sia corretto viene visualizzato un messaggio di successo e viene cambiata l'icona adiacente al pulsante;

Nel seguente esempio il file viene accettato e quindi possiamo procedere a importare uno scenario Gherkin. Clicchiamo il pulsante "Load Gherkin scenarios" e come per la prima importazione, si aprirà una finestra dove sarà richiesto in input di selezionare il percorso di sistema in cui è presente il file ".scenario", prendiamo come esempio il documento "esempio.scenario" descritto nel paragrafo 3.3 e immaginiamo di averlo selezionato e caricato nel programma premendo il tasto conferma della finestra.

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".scenario" viene restituito un messaggio pop-up di errore;
- Nel caso in cui il documento sia corretto viene visualizzato il messaggio "BDL added" in una sezione della finestra, viene cambiata l'icona adiacente al pulsante e viene attivato il pulsante "Check between BDL and Gherkin";

Una volta importati tutti i file, possiamo procedere al confronto, selezionando il pulsante "Check between BDL and Gherkin" (ora disponibile poiché sono presenti tutti i file di input), viene creato un file di log contenente i dettagli del confronto. Questo file di log quindi rappresenta il confronto tra "esempioBDL.BDL" ed "esempio.scenario". Di seguito viene mostrato il file log generato.

#### NOUNS:

The following nouns you use in your Gherkin correspond to the Business Domain Language:  
video; stream;

You may need to use the following nouns in your Gherkin:  
streaming; media; music; user; protocol; content; Napster;  
network; service; internet; file; format; server; computer;

You are using more than one time the following nouns but they don't correspond to the Business Domain Language:  
volume; app; account; command;

#### VERBS:

The following verbs you use in your Gherkin correspond to the Business Domain Language:  
stream; play; access;

You may need to use the following verbs in your Gherkin:  
become; base; require; send; include; enable; develop;  
increase; introduce; live; lead; share; deliver; save;  
create; grow; offer; code; buffer; provide; allow; copyright;  
download; reduce; pioneer; transform; launch; acquire; own  
; fall; encode; follow; compress; emerge; interact; mean;  
cause; view; call; apply; refer; distribute;

You are using more than one time the following verbs but they don't correspond to the Business Domain Language:  
select; start; input; press;

#### PREDICATES:



The following predicates you use in your Gherkin correspond to the Business Domain Language:

play video;

You may need to use the following predicates in your Gherkin:

use protocol; send stream; stream content; display media;  
require bandwidth; grow percent; save file; use format;  
enable streaming; use which; detect loss; make copy; use player;

You are using more than one time the following predicates but they don't correspond to the Business Domain Language:  
select video; start app; stream video; change volume;

### 3.5.3 NaturalAPI Design

Per selezionare il tool Design è sufficiente premere il pulsante "Design". Viene quindi mostrata una nuova schermata contenente diversi pulsanti che rappresentano i casi d'uso del tool Design:

- bottone di ritorno alla schermata iniziale;
- bottone per aprire un suggerimento;
- bottone per l'aggiunta di un file ".scenario";
- bottone per l'aggiunta di un file ".BO" [Opzionale];
- bottone per generare il file ".BAL".

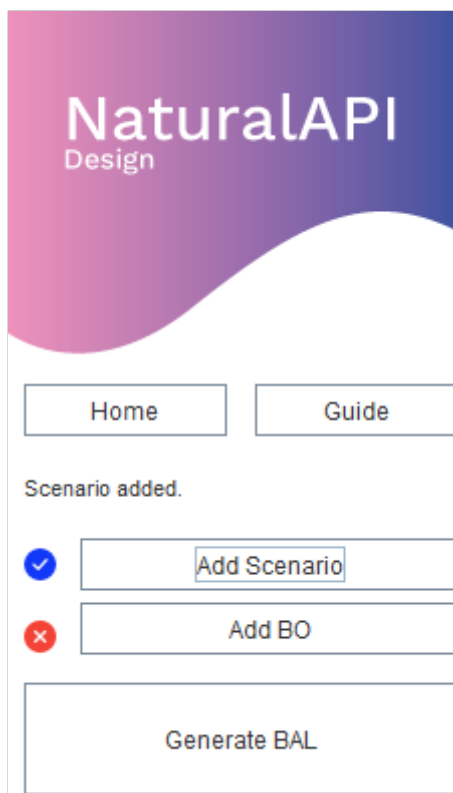


Figura 25: GUI - NaturalAPI Design

Il programma controllerà in automatico se sono presenti dei file di backup da utilizzare come input. Nel caso in cui il programma trovasse dei file, prima della comparsa della finestra dedicata a Discover, eseguirà un messaggio pop-up che segnalerà la presenza di file già presenti in memoria. Nel seguente esempio non sono presenti file di backup in quanto non stati ancora caricati i file di prova.

- Nel caso in cui si voglia confermare l'utilizzo di questi file basterà selezionare il tasto "yes".
- Nel caso in cui si voglia rifiutare l'utilizzo di questi file basterà selezionare il tasto "no".

Selezionando il pulsante "Add Scenario" si aprirà una finestra dove sarà richiesto in input di selezionare il percorso di sistema in cui è presente il file ".scenario", prendiamo come esempio il documento "esempio.scenario" descritto nel paragrafo 3.3, una volta selezionato e caricato nel programma premiamo il pulsante conferma della finestra.

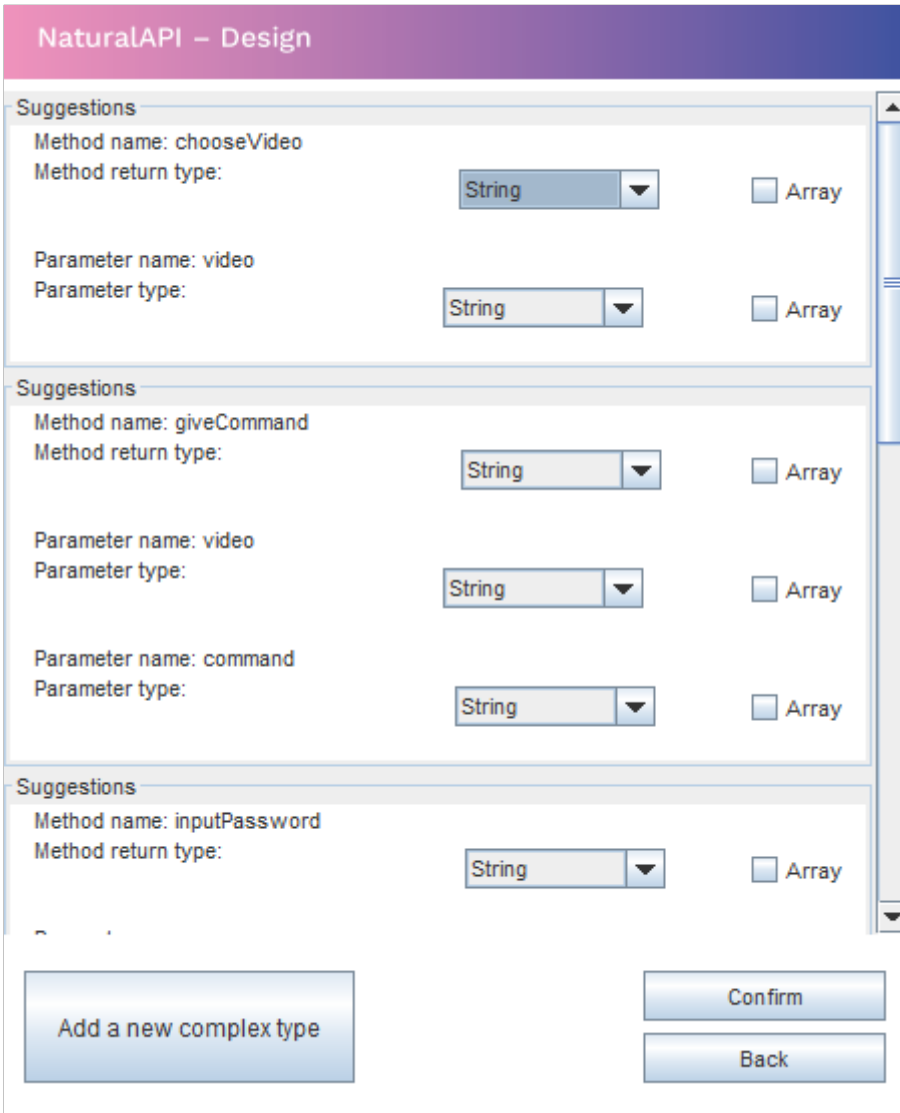
- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".txt" viene restituito un messaggio pop-up di errore;
- Nel caso in cui il documento sia corretto viene visualizzato il messaggio "Scenario added", cambiata l'immagine adiacente al bottone e attivato il pulsante "Generate BAL";

Nel seguente esempio il file viene accettato e si può procedere ad aggiungere un file BO con estensione ".json" oppure direttamente generare il BAL. Si ricorda inoltre che il file

BO viene generato in automatico nel tool "Discover" insieme al BDL, per completezza inseriamo la BO nel tool, quindi premiamo il tasto "Add BO". Si aprirà una finestra di scelta del percorso di sistema, andiamo nella directory del programma e poi nella cartella "Discover" selezioniamo il file "esempioBO.json" e premiamo il pulsante conferma della finestra. Una volta caricato correttamente il BO procediamo con l'estrazione, quindi selezioniamo il pulsante "Generate BAL". Viene richiesto tramite una finestra pop-up il nome del nuovo file, nel seguente esempio lo chiameremo "esempioBAL.json". Subito dopo comparirà una nuova finestra dove viene data la possibilità di modificare il tipo di ritorno dei metodi e il tipo dei parametri.

### 3.5.3.1 Modifica del BAL

In questa finestra aggiuntiva è possibile modificare il tipo di ritorno dei metodi e il tipo dei parametri.



The screenshot shows a window titled "NaturalAPI - Design". Inside, there's a "Suggestions" section with three entries:

- Method name:** chooseVideo  
**Method return type:** String (dropdown), ☐ Array
- Parameter name:** video  
**Parameter type:** String (dropdown), ☐ Array
- Method name:** giveCommand  
**Method return type:** String (dropdown), ☐ Array
- Parameter name:** video  
**Parameter type:** String (dropdown), ☐ Array
- Parameter name:** command  
**Parameter type:** String (dropdown), ☐ Array
- Method name:** inputPassword  
**Method return type:** String (dropdown), ☐ Array

At the bottom, there are three buttons: "Add a new complex type", "Confirm", and "Back".

Figura 26: GUI - Modifica del BAL

Viene creata una "suggestion" modificabile per ogni metodo individuato a partire dagli scenari caricati. Se in qualsiasi momento si volesse annullare l'operazione è sufficiente premere il bottone "Back" per venire riportati al menù precedente. È disponibile un menù a tendina che se permette di cambiare il tipo scegliendo fra:

- **v**: void;
- **s**: string;
- **i**: integer;
- **f**: float;
- **b**: boolean;
- tipi complessi aggiunti manualmente o rilevati al seguito dell'aggiunta della BO.

Adiacente al menù a tendina è disponibile un campo che permette di rendere un array o meno il campo scelto.

È disponibile un pulsante "Add a new complex type" che se selezionato aprirà una finestra dedicata all'aggiunta di un tipo complesso.

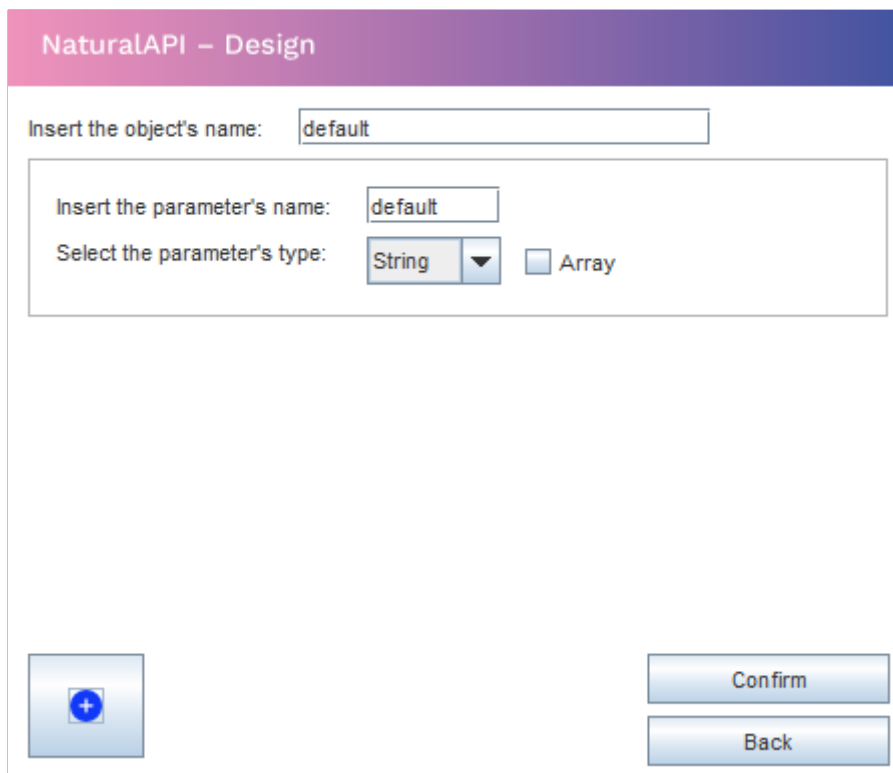


Figura 27: GUI - Aggiunta tipo complesso

In questa finestra si può aggiungere tramite il bottone "+" dei parametri al tipo complesso e analogamente alle "suggestions" della finestra precedente se ne può modificare il tipo e renderli degli array. Se si vuole annullare l'operazione è sufficiente premere il bottone "Back" e si verrà riportati alla finestra precedente. Una volta terminata la creazione pre-

mendo il bottone "Confirm" viene creato il tipo complesso utilizzabile nelle "suggestions" e si viene riportati alla finestra precedente.

Una volta soddisfatti con le modifiche desiderate è sufficiente premere il pulsante "Confirm" per salvare il BAL e terminare le operazioni di modifica. Di seguito viene mostrata una sezione del file esempioBAL.json generato tramite l'importazione del file "prova.scenario".

```
{
  "openapi" : "3.0.0",
  "info" : {
    "version" : "1.0.0",
    "title" : "streaming",
    "license" : {
      "name" : "MIT"
    }
  },
  "paths" : {
    "/chooseVideo" : {
      "get" : {
        "operationId" : "chooseVideo",
        "description" : "Scenario: select video---Given I started the app correctly---",
        "parameters" : [ {
          "in" : "query",
          "name" : "video",
          "description" : "Default",
          "required" : true,
          "schema" : {
            "type" : "string"
          }
        } ],
        "responses" : {
          "200" : {
            "description" : "page ",
            "content" : {
              "*/*" : {
                "schema" : {
                  "type" : "string"
                }
              }
            }
          }
        }
      }
    }
  }
}
```

Figura 28: Esempio output - BAL

### 3.5.4 NaturalAPI Develop

Per selezionare il tool Develop è sufficiente premere il pulsante "Develop". Viene quindi mostrata una nuova schermata contenente diversi pulsanti che rappresentano i casi d'uso del tool Develop

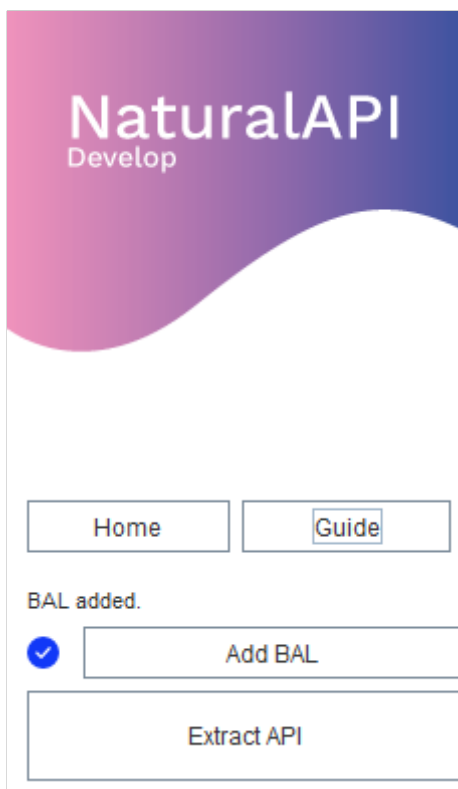


Figura 29: GUI - NaturalAPI Develop

Nella schermata sono presenti:

- bottone per il ritorno alla schermata iniziale
- bottone per aprire un suggerimento;
- bottone per l'aggiunta di un file BAL;
- bottone per avviare l'estrazione dell'API.

Il programma controllerà in automatico se sono presenti dei file di backup da utilizzare come input. Nel caso in cui il programma trovasse dei file, prima della comparsa della finestra dedicata a Discover, eseguirà un messaggio pop-up che segnalerà la presenza di file già presenti in memoria. Nel seguente esempio non sono presenti file di backup in quanto non stati ancora caricati i file di prova.

- Nel caso in cui si vuole confermare l'utilizzo di questi file basterà selezionare il tasto "yes".
- Nel caso in cui si vuole rifiutare l'utilizzo di questi file basterà selezionare il tasto "no".

Nel seguente esempio questa finestra pop-up non comparirà in quanto il file di prova non è stato ancora memorizzato nel programma.

### 3.5.4.1 Generazione dell'API

Selezionando il pulsante "Add BAL" si aprirà una finestra dove sarà richiesto in input di selezionare il percorso di sistema in cui è presente il file ".json", nel seguente esempio prendiamo il file "esempioBAL.json" generato nella directory "Design" dal precedente tool, quindi immaginiamo di averlo selezionato e caricato nel programma premendo il tasto conferma della finestra.

- Nel caso in cui il percorso inserito sia scorretto, il file indicato non esista o l'estensione indicata non sia ".txt" viene restituito un messaggio pop-up di errore;
- Nel caso in cui il documento sia corretto viene visualizzato il messaggio "BAL added" in una sezione della finestra, viene cambiata l'icona adiacente al bottone e viene attivato il pulsante dedicato all'estrazione del API;

Nel seguente esempio il file viene accettato e si può quindi procedere alla generazione del API. Selezioniamo quindi il pulsante "ExtractAPI" si aprirà una nuova schermata contenente diversi pulsanti per scegliere in che linguaggio si vuole generare l'API ed i relativi stub dei test.

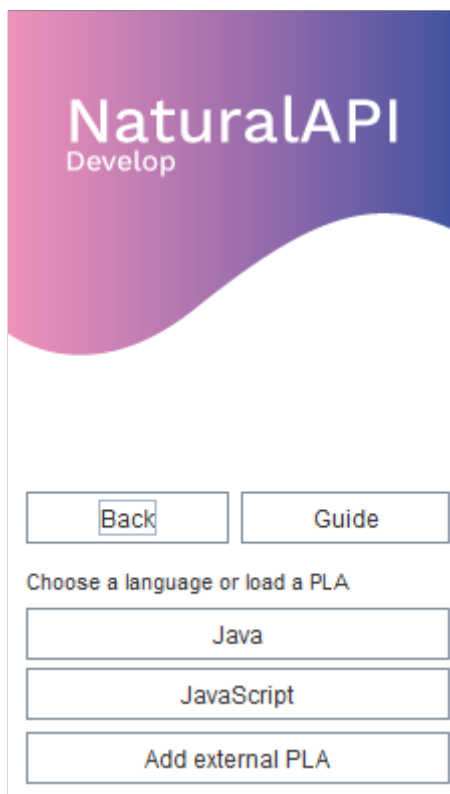


Figura 30: GUI - Generazione API

Le scelte possibili sono:

- 1: Java;
- 2: Javascript;
- 3: Aggiunta di un PLA esterno;

- 4: bottone per aprire un suggerimento;
- 5: bottone per ritornare alla schermata precedente,

In questa finestra è possibile scegliere due linguaggi di default o di aggiungere un file PLA esterno se si vuole usare una sintassi di un linguaggio differente. Nel seguente esempio scegliamo di creare un API tramite il linguaggio di default Java selezioniamo quindi il pulsante "Java" che genera un file di tipo Java denominato esempioBAL.java.

Di seguito viene mostrato Il file esempioBAL.java generato tramite l'importazione del file "esempioBAL.json".



```
/*
  <--classExtension: java
  <--integer: int
  <--float: float
  <--string: String
  <--boolean: boolean
*/

public class Streaming{

  private class Music{
    String length;
    String format;
  }

  private class Storage{
    String volume;
  }

  public String chooseVideo(String video){
    //page - chooseVideo
  }

  public String giveCommand(String video, String command){
    //video - giveCommand
  }

  public String inputPassword(String password, String username){
    //access - inputPassword
  }

  public String pressVideo(String video){
    //video - pressVideo
  }

  public String haveBandwith(String bandwith, String video){
    //error - haveBandwith
  }

  public String pressCommand(String command){
    //volume - pressCommand
  }

}
```

Figura 31: Esempio output - API

```
public class StreamingTest{

    @Given I started the app correctly
    public void givenIStartedTheAppCorrectly(){
        //Insert a test implementation
    }

    @And I want to watch a video
    public void andIWantToWatchAVideo(){
        //Insert a test implementation
    }

    @When I choose the video
    public void whenIChooseTheVideo(){
        //Insert a test implementation
        Streaming.chooseVideo();
    }

    @Then I will get the page of the video )
    public void thenIWillGetThePageOfTheVideo(){
        //Insert a test implementation
    }

    @Given I selected a video )
    public void givenISelectedAVideo(){
        //Insert a test implementation
    }

    @And I want to stream the video
    public void andIWantToStreamTheVideo(){
        //Insert a test implementation
    }

    @When I selected a video
    public void whenISelectedAVideo(){
        //Insert a test implementation
        Streaming.giveCommand();
    }

    @And I give the play command
    public void andIGiveThePlayCommand(){
        //Insert a test implementation
    }

    @Then The app will start the video
    public void thenTheAppWillStartTheVideo(){
        //Insert a test implementation
    }
}
```

Figura 32: Esempio output - Stub dei Test

## A Glossario

### A.1 A

#### A.1.1 API

Una Application Programming Interface (API) è un insieme di procedure e funzioni offerte ai programmatori per facilitare lo sviluppo e per permetterne il riuso.

### A.2 B

#### A.2.1 BAL

Un Business Application Language (BAL) è un termine usato per indicare dei tipi di file che rappresentano una determinata struttura che viene creata dall'utente o generata dal programma e rappresenta il metalinguaggio intermedio tra Scenari ed API.

#### A.2.2 BDL

Un Business Domain Language (BDL) è un termine usato per indicare dei tipi di file che rappresentano un testo con una struttura specifica.

### A.3 G

#### A.3.1 Gherkin

Linguaggio naturale strutturato, facilmente comprensibile in prima lettura, usato per strutturare scenari dei casi d'uso.

### A.4 J

#### A.4.1 Java

Linguaggio di programmazione ad alto livello, orientato agli oggetti e a tipizzazione statica, appositamente progettato per essere il più possibile indipendente dalla piattaforma di esecuzione.

#### A.4.2 JavaScript

Linguaggio di programmazione orientato agli oggetti e agli eventi, comunemente utilizzato nella programmazione Web per la creazione di effetti dinamici interattivi in siti web e applicazioni web.

#### A.4.3 Json

JavaScript Object Notation (JSON) è un formato di interscambio di dati, che utilizza testo leggibile per archiviare e trasmettere oggetti costituiti da coppie attributo-valore e tipi di dati serializzabili.

## **A.5 N**

### **A.5.1 NaturalAPI Design**

Tool di NaturalAPI che genera un BAL a partire da un BDL e scenari Gherkin.

### **A.5.2 NaturalAPI Develop**

Tool di NaturalAPI che genera un API a partire da un BAL e un PLA.

### **A.5.3 NaturalAPI Discover**

Tool di NaturalAPI che genera un BDL a partire da dei file di testo.

## **A.6 P**

### **A.6.1 PLA**

Programming Language Adapter (PLA) è un termine usato per indicare un tipo di file che rappresenta le specifiche di un determinato linguaggio di programmazione.

## **A.7 S**

### **A.7.1 Scenario**

Sequenza di azioni che definisce il comportamento di un software utilizzando un linguaggio formale.