



HexaTech

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

Manuale Utente

Versione	0.1.0
Approvazione	Gerardo Kokoshari
Redazione	Jacopo Battilana
	Gerardo Kokoshari
	Denis Salviato
	Alessio Barbiero
Verifica	Matteo Brosolo
	Luca Marcon
	Eduard George Serban
	Esterno
Uso	
Destinato a	<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
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 ^A T _E X e impostato lo scheletro del documento e relativa verifica	2020-05-24

Indice

1	Introduzione	6
1.1	Scopo del documento	6
1.2	Scopo del prodotto	6
1.3	Glossario	6
1.4	Comunicazione e malfunzionamenti	6
2	Configurazione	7
2.1	Requisiti di sistema	7
2.2	Requisiti NaturalAPI Discover	7
2.3	Requisiti NaturalAPI Design	7
2.4	Requisiti NaturalAPI Develop	8
3	Funzionalità ed uso di NaturalAPI	11
3.1	Premessa	11
3.2	Command Line Interface	11
3.2.1	Schermata iniziale	11
3.2.2	NaturalAPI Discover	11
3.2.2.1	Controllo di documenti già salvati	12
3.2.2.2	Aggiunta di un documento (.txt)	12
3.2.2.3	Confronto tra BDL e scenari Gherkin	14
3.2.3	NaturalAPI Design	17
3.2.3.1	Controllo di documenti già salvati	17
3.2.3.2	Aggiunta di un file Gherkin (.scenario)	17
3.2.3.3	Esempio di input di Design	19
3.2.3.4	Esempio di output di Design	20
3.2.4	NaturalAPI Develop	21
3.2.4.1	Controllo di documenti già salvati	21
3.2.4.2	Aggiunta di un file BAL (.json)	21
A	Glossario	23
A.1	A	23
A.1.1	API	23
A.2	B	23
A.2.1	BAL	23
A.2.2	BDL	23
A.3	G	23
A.3.1	Gherkin	23
A.4	J	23
A.4.1	Java	23
A.4.2	JavaScript	23
A.4.3	Json	23
A.5	N	24
A.5.1	NaturalAPI Design	24
A.5.2	NaturalAPI Develop	24
A.5.3	NaturalAPI Discover	24
A.6	P	24

A.6.1	PLA	24
A.7	S	24
A.7.1	Scenario	24

Elenco delle figure

1	Scenario Gherkin	7
2	PLA Java	9
3	PLA Javascript	10
4	Casi d'uso - Schermata iniziale	11
5	Casi d'uso - NaturalAPI Discover	12
6	Casi d'uso - Estrazione BDL	13
7	Output Discover - CSV Sostantivi	13
8	Output Discover - CSV Verbi	14
9	Output Discover - CSV Nomi	14
10	Casi d'uso - NaturalAPI Design	17
11	Casi d'uso - Estrazione BAL	18
12	Esempio output - BAL	20
13	Casi d'uso - NaturalAPI Develop	21
14	Casi d'uso - Creazione API	22
15	Esempio output - API	22

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, When o Then a seconda delle necessità.

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 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;
- **<--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 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.

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-->
    //RETURN: <--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.

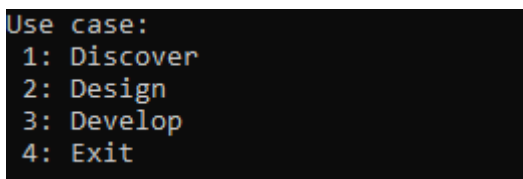
3.2 Command Line Interface

3.2.1 Schermata iniziale

Per avviare il programma bisogna posizionarsi da console nella cartella contenente il file "NaturalAPI.jar" ed eseguire il comando `java -jar NaturalAPI.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: chiusura del software;



```
Use case:
1: Discover
2: Design
3: Develop
4: Exit
```

Figura 4: Casi d'uso - Schermata iniziale

3.2.2 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:** ritorno alla schermata iniziale.

```
Use Case:
1: Check if there are saved documents
2: Add a document (.txt)
3: Check between BDL and Gherkin
4: Back
```

Figura 5: Casi d'uso - NaturalAPI Discover

3.2.2.1 Controllo di documenti già salvati

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 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.

3.2.2.2 Aggiunta di un documento (.txt)

Viene richiesto all'utente in input il percorso di sistema in cui è presente il documento ".txt", fornendone un possibile esempio (ex. C:\Users\User\Desktop\example.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;

3.2.2.2.1 Estrazione BDL

Il menù di estrazione del BDL permette di:

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

```

Use Case:
1: Extract BDL
2: Add another document (.txt)
3: Back
  
```

Figura 6: Casi d'uso - Estrazione BDL

Per estrarre il BDL dai documenti di testo inseriti è sufficiente digitare il nome del file. Dopo una breve attesa vengono creati nella directory del progetto, nella cartella Discover, un file ".BDL", un file ".BO" e tre file ".csv", contenenti delle liste leggibili dei sostantivi, dei verbi e dei predicati contenuti nel BDL. I tre file ".csv" vengono aperti con il relativo programma impostato di default. L'utente viene quindi riportato al menù iniziale del tool Discover.

Per aggiungere ulteriori documenti invece, la procedura è analoga a quella descritta precedentemente. Viene richiesto all'utente in input il percorso di sistema in cui è presente il documento ".txt", fornendone un possibile esempio (ex. C:\Users\User\Desktop\example.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 di estrazione del BDL;
- Nel caso in cui il percorso inserito sia corretto viene visualizzato il menù di estrazione del BDL;

3.2.2.2.2 Esempio di estrazione BDL

```

Card,8
Debit,3
ATM,3
Credit,2
Football,2
Banking,1
Variety,1
Amount,1
Bank,1
Account,1
Case,1
Fee,1
Interest,1
Rate,1
Slot,1
Chip,1
Side,1
  
```

Figura 7: Output Discover - CSV Sostantivi

```
Insert,2  
Play,2  
Link,1  
Ensure,1  
Face,1  
Like,1
```

Figura 8: Output Discover - CSV Verbi

```
Insert card,2  
Like football,1  
Play football,1
```

Figura 9: Output Discover - CSV Nomi

3.2.2.3 Confronto tra BDL e scenari Gherkin

Per effettuare un confronto tra BDL e uno scenario Gherkin bisogna prima importare un BDL:

- **BDL esterno:** viene richiesto all'utente 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.

Dopo aver importato correttamente il BDL viene richiesto di importare uno scenario Gherkin in maniera analoga all'importazione dei documenti precedenti. Viene quindi richiesto all'utente in input il percorso di sistema in cui è presente il file ".scenario", fornendone un possibile esempio (ex. C:\Users\User\Desktop\example.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.

3.2.2.3.1 Esempio di log

NOUNS:

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

ATM; card; machine; cash; money; account; fund;

You may need to use the following nouns in your Gherkin:
deposit; bank; envelope; check; time; institution; pin; slip;
customer; amount; name; debit; transaction; slot; number;

You are using more than one time the following nouns but they don't correspond to the Business Domain Language:
holder; request;

VERBS:

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

You may need to use the following verbs in your Gherkin:
deposit; insert; enter; include; indicate; transfer; associate;
check; add; account; provide; need; automate; accept; allow;
prompt; access; require; sign; prepare; place; label; fill;
issue; own; match;

You could use "insert" instead of "introduce";
You could use "enter" instead of "introduce";
You could use "check" instead of "contain";

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

PREDICATES:

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

You may need to use the following predicates in your Gherkin:
use ATM; insert card; allow you; enter amount; have option;
withdraw money; check balance; transfer money; access account;
use slip; enable customer; perform transaction; account inquiry;
include money; have sign; indicate name; own ATM; make variety;
make withdrawal; accept deposit; account number; prepare slip;
insert envelope; make deposit; deposit money; deposit check;
enter number; insert check;

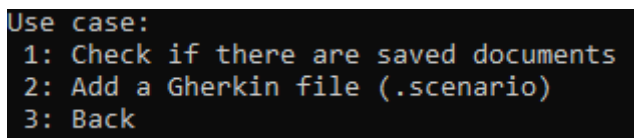
You are using more than one time the following predicates but they don't correspond to the Business Domain Language:

introduce card; have fund; contain money;

3.2.3 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:** ritorno alla schermata iniziale,



```
Use case:
1: Check if there are saved documents
2: Add a Gherkin file (.scenario)
3: Back
```

Figura 10: Casi d'uso - NaturalAPI Design

3.2.3.1 Controllo di documenti già salvati

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.

3.2.3.2 Aggiunta di un file Gherkin (.scenario)

Viene richiesto all'utente in input il percorso di sistema in cui è presente il documento ".scenario", fornendone un possibile esempio (ex. C:\Users\User\Desktop\example.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 BDL.

Il menù a scelta multipla permette all'utente di annullare l'operazione o di procedere eseguendo la procedura di caricamento del BDL. Viene quindi richiesto all'utente in input il percorso di sistema in cui è presente il documento ".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 per la estrazione del BAL.

3.2.3.2.1 Estrazione BAL

Il menù di estrazione del BAL permette di:

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

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

Figura 11: Casi d'uso - Estrazione BAL

Aggiungere una Business Ontology (BO) permette al programma di proporre degli oggetti complessi che potrebbero essere attinenti al BAL che si sta creando. Per aggiungere una BO viene richiesto all'utente 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.

Quando viene avviata l'estrazione del BAL viene creato un file ".json" nella directory del progetto, nella cartella Design. Durante la creazione viene data la possibilità all'utente 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 e sono i tipi complessi rilevati all'interno.

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 e si viene riportati al menù principale del tool Design.

3.2.3.3 Esempio di input di Design

Scenario: Buy last coffee

Given there are 1 coffees left in the machine

And I have deposited 1 dollar

When I press the coffee button

Then I should be served a coffee

3.2.3.4 Esempio di output di Design

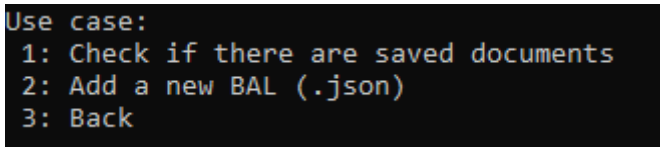
```
{
  "openapi": "3.0.0",
  "info": {
    "version": "1.0.0",
    "title": "Test",
    "license": {
      "name": "MIT"
    }
  },
  "paths": {
    "/buyLastCoffee": {
      "get": {
        "operationId": "buyLastCoffee",
        "description": "null",
        "tags": [
          "{Nouns:[card,debit,credit,banking]]",
          "{Verbs:[insert,link]]",
          "{Predicates:[insert card,like football]]"
        ],
        "parameters": [
          {
            "in": "query",
            "name": "button ",
            "description": "Default",
            "required": true,
            "schema": {
              "type": "string"
            }
          }
        ],
        "responses": {
          "200": {
            "description": "coffee ",
            "content": {
              "*/*": {
                "schema": {
                  "type": "string"
                }
              }
            }
          }
        }
      }
    }
  },
  "components": {
    "schemas": {
    }
  }
}
```

Figura 12: Esempio output - BAL

3.2.4 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:** ritorno alla schermata iniziale



```
Use case:
1: Check if there are saved documents
2: Add a new BAL (.json)
3: Back
```

Figura 13: Casi d'uso - NaturalAPI Develop

3.2.4.1 Controllo di documenti già salvati

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.

3.2.4.2 Aggiunta di un file BAL (.json)

Viene richiesto all'utente in input il percorso di sistema in cui è presente il documento ".scenario", 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 iniziale del tool Develop;
- Nel caso in cui il caricamento vada a buon fine viene visualizzato un menù per la creazione dell'API.

3.2.4.2.1 Creazione API

Il menù di creazione dell'API permette di:

- **1:** creare l'API, scegliendo un PLA per specificare il linguaggio;
- **2:** annullare la creazione.

Il PLA va a definire in quale linguaggio viene prodotta l'API. Nel menù di scelta sono presenti:

- **1:** Java;

- 2: Javascript;
- 3: PLA esterno;
- 4: 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: Back
```

Figura 14: Casi d'uso - Creazione API

Una volta selezionato il linguaggio viene creato il corpo dell'API ed i relativi stub dei test.

3.2.4.2.2 Esempio di input di Develop

In input vengono forniti l'output di Design e il PLA, illustrato in precedenza.

3.2.4.2.3 Esempio di output di Develop

```
public class Test{  
  
    private class Persona{  
        String cognome;  
        String nome;  
        int telefono;  
    }  
  
    private class CartaDiCredito{  
        int pin;  
        String titolare;  
    }  
  
    public Persona[] buyLastCoffee(boolean button ){  
        //coffee - buyLastCoffee  
    }  
  
    public String buyLastCoffee2(String button ){  
        //coffee - buyLastCoffee2  
    }  
  
    public String buyLastCoffee3(String button ){  
        //coffee - buyLastCoffee3  
    }  
}
```

Figura 15: Esempio output - API

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.