****

Daniel Machado

Daniel Lima

Inês Clavel

João Figueiredo

Thales Lemos

Duck Squad

Duck Soft Works & Co MEMBERS:

**SPRINT 1**

User Story:

Make a Language Detection Prototype

Project Report and Analysis

UpSkill Java Capgemini

Index

[**Task Distribution** 2](#_Toc104805093)

[**Project Analyses** 3](#_Toc104805094)

[**POC** 3](#_Toc104805095)

[**Performance tests results:** 5](#_Toc104805096)

[**Simple Analyzer** 5](#_Toc104805097)

[**Standard Analyser** 5](#_Toc104805098)

[**Tests with N-Grams from 4 to 5** 5](#_Toc104805099)

[**Tests with N-Grams from 1 to 5** 6](#_Toc104805100)

[**Tests with N-Grams from 4 to 10** 6](#_Toc104805101)

[**Results Analysis** 6](#_Toc104805102)

[**Prototype** 7](#_Toc104805103)

[**Introduction and Design** 7](#_Toc104805104)

[**Technical Specifications** 7](#_Toc104805105)

[**Considerations for future improvements** 8](#_Toc104805106)

[**Domain Model** 9](#_Toc104805107)

[**Class Diagram** 10](#_Toc104805108)

[**Prototype Sequence Diagram** 10](#_Toc104805109)

# **Task Distribution**

|  |  |
| --- | --- |
| **Task Distribution** | **Members** |
| **POC - Make a language recognition small application through console reading text** | |
| POC – Language recognition in console reading text | Daniel Machado and Thales Lemos |
| Reading and understanding Apache Lucene | All the Team Members |
| POC - JavaDoc | Daniel Machado |
| Performance Tests | Daniel Machado, Inês Clavel, Thales Lemos |
| Report (POC Analysis) | Inês Clavel |
| Report (Performance Tests) | Inês Clavel, Thales Lemos |
| **Prototype - Make a language recognition small application through console reading text** | |
| Architectural Organization | All the Team Members |
| SpringBoot - Initial Project Skeleton | Inês Clavel |
| Controller | Inês Clavel |
| Value Objects | Daniel Lima, Thales Lemos |
| Task (Aggregate Root) | João Figueiredo |
| TaskDomainDTOAssembler | Daniel Machado |
| TaskService | Daniel Machado, Daniel Lima, Thales Lemos |
| AnalyzerService | Daniel Machado, Daniel Lima, Thales Lemos |
| Dictionary Services | Daniel Machado, Daniel Lima, Thales Lemos |
| TaskDomainDTOAssembler | Daniel Machado |
| TaskDTO | Daniel Machado |
| NewTaskInfoDTO | Thales Lemos |
| Controller Mock Test | Thales Lemos |
| AnalizerService Unit Test | Inês Clavel |
| TaskService Tests | João Figueiredo |
| DictionaryService Tests | João Figueiredo |
| Controller javaDoc | Thales Lemos |
| AnalizerService javaDoc | Inês Clavel |
| TaskService javaDoc | João Figueiredo |
| DictionaryService javaDoc | João Figueiredo |
| Domain Diagram | Daniel Machado |
| Sequence Diagram | João Figueiredo |
| Sequence Diagram | Inês Clavel, Thales Lemos |
| Glossary | João Figueiredo |
| Report (Structure) | Inês Clavel |
| Report (General Analysis) | Inês Clavel |
| Report (Design and Technical Analysis) | Daniel Lima |

# **Project Analyses**

# **POC**

For this first Sprint we decided to start by creating a *Proof Of Concept* (*POC*) to better understand how Lucene works and how it could better serve our purpose.

In this first step we have focused on the language detection without using the any type of Architecture Design. For that we used the Aspell dictionaries as base of comparison.

Our approach was to test either text files or input reading through the console and, with the Lucene base, analyzing and detecting the language used in its majority. The process begins by running through the dictionaries and by analyzing the inputted text to compare and attribute the detected main language.

We have also integrated a score output that shows the percentage of probability for main language being used in that text.

For now, we are using 3 languages (English, Portuguese and Spanish) with the possibility of future expansion.

In this POC, we made a few Performance tests to analyze the efficiency VS efficacy through the console. For that, both the scores (in percentage) and the duration of the process (by milliseconds) were used.

The tests were made using 3 different sizes of texts in all 3 languages. Using the same story book (The Little Prince), we made performance tests within the different search analyzers – Simple Analyzer and Standard Analyzer that use methods from Lucene Library which, consequently, uses tokenization to compare the words to make the language definition. We also used the Custom Analyzer where we made use of N-Grams, with several dimensions, to make the distinction between fragments of the words.

The following phrases were the ones used to make the performance tests:

In English

Phrase 1 - *Yes of course.*

Phrase 2 *- My drawing was not a picture of a hat. It was a picture of a boa constrictor digesting an elephant. But since the grown-ups were not able to understand it, I made another drawing: I drew the inside of the boa constrictor, so that the grown-ups could see it clearly.*

Phrase 3 - *The grown-ups' response, this time, was to advise me to lay aside my drawings of boa constrictors, whether from the inside or the outside, and devote myself instead to geography, history, arithmetic and grammar. That is why, at the age of six, I gave up what might have been a magnificent career as a painter. I had been disheartened by the failure of my Drawing Number One and my Drawing Number Two. Grown-ups never understand anything by themselves, and it is tiresome for children to be always and forever explaining things to them. So then I chose another profession, and learned to pilot airplanes. I have flown a little over all parts of the world; and it is true that geography has been very useful to me. At a glance I can distinguish China from Arizona. If one gets lost in the night, such knowledge is valuable. In the course of this life I have had a great many encounters with a great many people who have been concerned with matters of consequence. I have lived a great deal among grown-ups. I have seen them intimately, close at hand. And that hasn't much improved my opinion of them. Whenever I met one of them who seemed to me at all clear-sighted, I tried the experiment of showing him my Drawing Number One, which I have always kept. I would try to find out, so, if this was a person of true understanding. But, whoever it was, he, or she, would always say: "That is a hat."*

Portuguese

Phrase 1 -*Claro que sim.*

Phrase 2 - *Meu desenho não representava um chapéu. Representava uma jibóia digerindo um elefante. Desenhei então o interior da jibóia, a fim de que as pessoas grandes pudessem compreender. Elas têm sempre necessidade de explicações.*

Phrase 3 - *As pessoas grandes aconselharam-me deixar de lado os desenhos de jibóias abertas ou fechadas, e dedicar-me de preferência à geografia, à história, ao cálculo, à gramática. Foi assim que abandonei, aos seis anos, uma esplêndida carreira de pintor. Eu fora desencorajado pelo insucesso do meu desenho número 1 e do meu desenho número 2. As pessoas grandes não compreendem nada sozinhas, e é cansativo, para as crianças, estar toda hora explicando. 3 Tive pois de escolher uma outra profissão e aprendi a pilotar aviões. Voei, por assim dizer, por todo o mundo. E a geografia, é claro, me serviu muito. Sabia distinguir, num relance, a China e o Arizona. É muito útil, quando se está perdido na noite. Tive assim, no decorrer da vida, muitos contatos com muita gente séria. Vivi muito no meio das pessoas grandes. Vi-as muito de perto. Isso não melhorou, de modo algum, a minha antiga opinião. Quando encontrava uma que me parecia um pouco lúcida, fazia com ela a experiência do meu desenho número 1, que sempre conservei comigo. Eu queria saber se ela era verdadeiramente compreensiva. Mas respondia sempre: "É um chapéu". Então eu não lhe falava nem de jibóias, nem de florestas virgens, nem de estrelas. Punha-me ao seu alcance. Falava-lhe de bridge, de golfe, de política, de gravatas. E a pessoa grande ficava encantada de conhecer um homem tão razoável.*

Spanish

Phrase 1 - *Claro que si.*

Phrase 2 - *Mi dibujo no representaba un sombrero. Representaba una serpiente boa que digiere un elefante. Dibujé entonces el interior de la serpiente boa a fin de que las personas mayores pudieran comprender. Siempre estas personas tienen necesidad de explicaciones.*

Phrase 3 - *Las personas mayores me aconsejaron abandonar el dibujo de serpientes boas, ya fueran abiertas o cerradas, y poner más interés en la geografía, la historia, el cálculo y la gramática. De esta manera a la edad de seis años abandoné una magnífica carrera de pintor. Había quedado desilusionado por el fracaso de mis dibujos número 1 y número 2. Las personas mayores nunca pueden comprender algo por sí solas y es muy aburrido para los niños tener que darles una y otra vez explicaciones. Tuve, pues, que elegir otro oficio y aprendía pilotear aviones. He volado un poco por todo el mundo y la geografía, en efecto, me ha servido de mucho; al primer vistazo podía distinguir perfectamente la China de Arizona. Esto es muy útil, sobre todo si se pierde uno durante la noche. 2 A lo largo de mi vida he tenido multitud de contactos con multitud de gente seria. Viví mucho con personas mayores y las he conocido muy de cerca; pero esto no ha mejorado demasiado mi opinión sobre ellas. Cuando me he encontrado con alguien que me parecía un poco lúcido, lo he sometido a la experiencia de mi dibujo número 1 que he conservado siempre. Quería saber si verdaderamente era un ser comprensivo. E invariablemente me contestaban siempre: "Es un sombrero". Me abstenía de hablarles de la serpiente boa, de la selva virgen y de las estrellas. Poniéndome a su altura, les hablaba del bridge, del golf, de política y de corbatas. Y mi interlocutor se quedaba muy contento de conocer a un hombre tan razonable.*

## **Performance tests results:**

### **Simple Analyzer**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phrase** | **Language** | **Matches found** | **Score (%)** | **TimeStamp result (ms)** |
| 1 | English | 1 (EN) | 100 | 12 |
| 2 | English | 3 (EN, SP, PT) | 89.39 (6.39, 4.22) | 27 |
| 3 | English | 3 (EN, PT, SP) | 90.56 (6.28, 3.16) | 37 |
| 1 | Portuguese | 3 (PT, EN, SP) | 52.63 (25.43, 21.93) | 20 |
| 2 | Portuguese | 3 (PT, EN, SP) | 77.88 (16.06, 6.06) | 22 |
| 3 | Portuguese | 3 (PT, EN, SP) | 78.20 (12.18, 9.62) | 32 |
| 1 | Spanish | 3 (PT, EN, SP) | (49.41) 36.62 (13.97) | 18 |
| 2 | Spanish | 3 (SP, EN, PT) | 51.58 (34.23, 14.20) | 19 |
| 3 | Spanish | 3 (SP, EN, PT) | 44.66 (33.57 21.77) | 24 |

This Analyzer uses the Tokenizer without StopWords.

### **Standard Analyser**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phrase** | **Language** | **Matches found** | **Score (%)** | **TimeStamp result (ms)** |
| 1 | English | 1(EN) | 100 | 12 |
| 2 | English | 3 (EN, SP, PT) | 88.66 (6.82, 4.53) | 25 |
| 3 | English | 3 (EN, PT, SP) | 89.63 (6.91, 3.45) | 33 |
| 1 | Portuguese | 3 (PT, SP, EN) | 55.55 (22.60, 21.85) | 20 |
| 2 | Portuguese | 3 (PT, EN, SP) | 79.47 (14.45, 6.08) | 20 |
| 3 | Portuguese | 3 (PT, EN, SP) | 79.48 (10.93, 9.59) | 23 |
| 1 | Spanish | 3 (PT, EN, SP) | (50.58) 36.60 (12.82) | 16 |
| 2 | Spanish | 3 (SP, EN, PT) | 52.62 (32.56, 14.82) | 17 |
| 3 | Spanish | 3 (SP, EN, PT) | 45.96 (31.10, 22.94) | 20 |

This Analyzer usually uses StopWords, but we removed this function for this test.

### **Tests with N-Grams from 4 to 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phrase** | **Language** | **Matches found** | **Score (%)** | **TimeStamp result (ms)** |
| 1 | English | 1 (EN, PT, SP) | 83.73 (9.59, 6.67) | 22 |
| 2 | English | 3 (EN, SP, PT) | 72.81 (14.20, 13.00) | 25 |
| 3 | English | 3 (EN, PT, SP) | 76.67 (11.99, 11.34) | 46 |
| 1 | Portuguese | 3 (SP, PT, EN) | (46.47) 37.52 (16.01) | 17 |
| 2 | Portuguese | 3 (PT, SP, EN) | 52.48 (23.85, 23.67) | 22 |
| 3 | Portuguese | 3 (PT, SP, EN) | 56.35 (25.20, 18.46) | 34 |
| 1 | Spanish | 3 (SP, PT, EN) | 46.47 (37.52, 16.01 | 17 |
| 2 | Spanish | 3 (SP, EN, PT) | 45.32 (28.95, 25.73) | 20 |
| 3 | Spanish | 3 (SP, PT, EN) | 47.62 (29.77, 22.61) | 33 |

### **Tests with N-Grams from 1 to 5**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phrase** | **Language** | **Matches found** | **Score (%)** | **TimeStamp result (ms)** |
| 1 | English | 1 (EN, PT, SP) | 50.75 (26.84, 22.41) | 26 |
| 2 | English | 3 (EN, SP, PT) | 47.92 (26.71, 25.37) | 40 |
| 3 | English | 3 (EN, PT, SP) | 49.64 (25.44, 24.92) | 62 |
| 1 | Portuguese | 3 (SP, PT, EN) | 34.95 (33.87, 31.18) | 22 |
| 2 | Portuguese | 3 (PT, SP, EN) | 44.82 (28.49, 26.68) | 32 |
| 3 | Portuguese | 3 (PT, SP, EN) | 43.62 (29.85, 26.52) | 72 |
| 1 | Spanish | 3 (SP, PT, EN) | 35.15 (33.93, 30.93) | 23 |
| 2 | Spanish | 3 (SP, EN, PT) | 37.21 (31.72, 31.07) | 37 |
| 3 | Spanish | 3 (SP, PT, EN) | 39.07 (32.88, 28.05) | 45 |

### **Tests with N-Grams from 4 to 10**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Phrase** | **Language** | **Matches found** | **Score (%)** | **TimeStamp result (ms)** |
| 1 | English | 1 (EN, PT, SP) | 86.45 (7.96, 5.60) | 25 |
| 2 | English | 3 (EN, SP, PT) | 75.58 ( 16.16, 8.26) | 30 |
| 3 | English | 3 (EN, SP, PT) | 81.75 (9.34, 8.91) | 89 |
| 1 | Portuguese | 3 (SP, PT, EN) | (46.51) 37.57 (15.91) | 19 |
| 2 | Portuguese | 3 (PT, SP, EN) | 60.68 (21.21, 18.11) | 29 |
| 3 | Portuguese | 3 (PT, SP, EN) | 64.23 (22.58, 13.18 ) | 109 |
| 1 | Spanish | 3 (SP, PT, EN) | 46.51 (37.57, 15.91) | 11 |
| 2 | Spanish | 3 (SP, EN, PT) | 54.94 (23.55, 21.51) | 30 |
| 3 | Spanish | 3 (SP, PT, EN) | 52.90 (27.86, 19.25) | 100 |

## **Results Analysis**

With these simple tests made, we can verify that both the SimpleAnalyzer, StandardAnalyzer are the most efficient and effective to define the correct language. Although the StandardAnalyzer is faster, the SimpleAnalyser can be slightly more accurate.

As for the N-Grams tests, we can see that, without the proper N-Grams dictionaries, the results were worse than expected as we have analyzed a few other applications (e.g. Lingua) that also make use of the N-Grams and with greater results. We did not try it with those dictionaries in the POC but we are considering to try it out in the main application in a future version.

Another repair that we can observe generally is that the smallest the phrase is, the least accurate the language detection can be.

Therefore, we can conclude that, with the methods that were used, SimpleAnalyzer is the most efficient and StandardAnalyzer the most effective.

# **Prototype**

## **Introduction and Design**

The evolution of the *POC* was to transform this simple application in a prototype application. In this approach we continue to keep things in a very simple way as, throughout the process, we did not know for sure what the main application was going to consist of.

Accordingly, we made some decision mainly about design, organization, and possibility of development of the app. For that, we used *SpringBoot* in our application that serves the same purpose as the *POC* but with the **Onion Layered Architecture** emphasizing overall good practices.

We are also applying the following design patterns:

**RDD**(Responsibility Driven Design) was applied in conjunction with **GRASP**, **MVC** and **DDD** patterns designating a TaskController as the task delegator, several services (namely TaskService, AnalyzerService and DictionaryService) as the creators. The domain layer was centered on representing key concepts for the domain, representing them as entities, aggregate roots and value objects when needed be. Encapsulation of the business core was attained via usage of *DTO* across layers.

**SOLID** design was also taken into consideration by attempting to uphold the “S” - single-responsibility principle and the “O” – open-closed principle.

**HCLC** was another one of the guidelines for the development of this prototype seen as scalability and maintainability are future concerns for the improvement of the concept. As such, an attempt for the lowest possible coupling between layers was thought at all times.

## **Technical Specifications**

The concept application is essentially a *RESTfuI API* that receives stateless data via *HTTP* request. This request is then mapped into a *DTO* by the Rest Controller (TaskControlller). Once it has been mapped, it will start the process to create a new task.

The task creation is handled by several services that act together to do so:

-The *Analyzer Service* .It is responsible for creating Lucene’s analyzer, reader and searcher. Simply put, all of the necessary “cogs” to analyze the input text, strip it of any unnecessary characters and compare the terms or phrases by querying the “clean text” with available dictionaries of terms. By doing so, it will grant a weighted score to the query based on *Lucene’s* similarity algorithm. This score is then used to establish the prevalent language.

-The *Dictionary Service*. This is where all the rules for mapping, configuring and adding documents to an index that will serve as dictionaries are contained. Without doing so, there would be no way to compare text with an input at all, which in turn would prove Lucene to be impractical.

-The *Task Service*. The part responsible for creating and using the other services to realize the user scenario. This is where the object containing the data that will be assembled to be sent as a client-side response. In other words, where the information with the determined language will be stored and sent upstream towards the user.

## **Considerations for future improvements**

At this embryonic stage we overlook several things, as we should, hence there are a lot of matters to be considered, added or improved further in the project.

We have decided not to have anything persisted at the time but it is something to be applied later. This is one of the elements that we did not consider to be important this early, nor needed for this Prototype.

Likewise, there are many validations that will have to be included and exceptions yet not treated (e.g., input of an empty text).

In the same way, the unit tests (and overall tests) are not at all concluded. There are only a few that serve the simple process of confirming that the application is working.

We would also like to report that we made a trial with an immensely big book (the Bible). As a result, the application went over 5 minutes when trying to process the information. The test was interrupted at that stage, as it was clear that there are things to be improved and taken care of, that are still not in accordance with our standards.

To conclude, we are aware of the project weaknesses at the moment when considering the future and final application. At the same time, we are happy when thinking at this product only as a prototype.

We are now prepared to move forward with the development to make the best we can to transform it in the a, hopefully, successfully final product.

# **Domain Model**

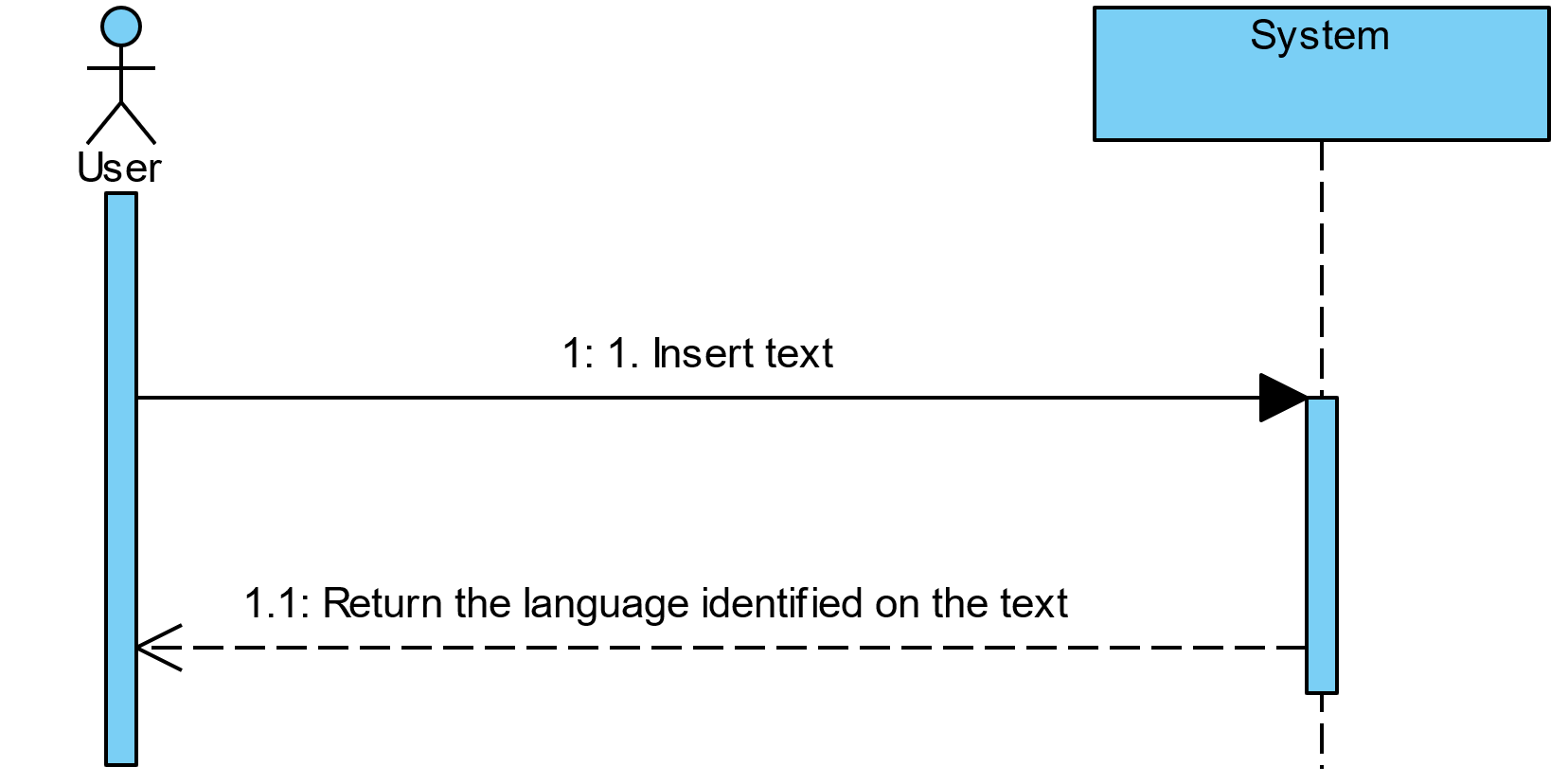
**Diagram, box and whisker chart

Description automatically generated**

# **Class Diagram**

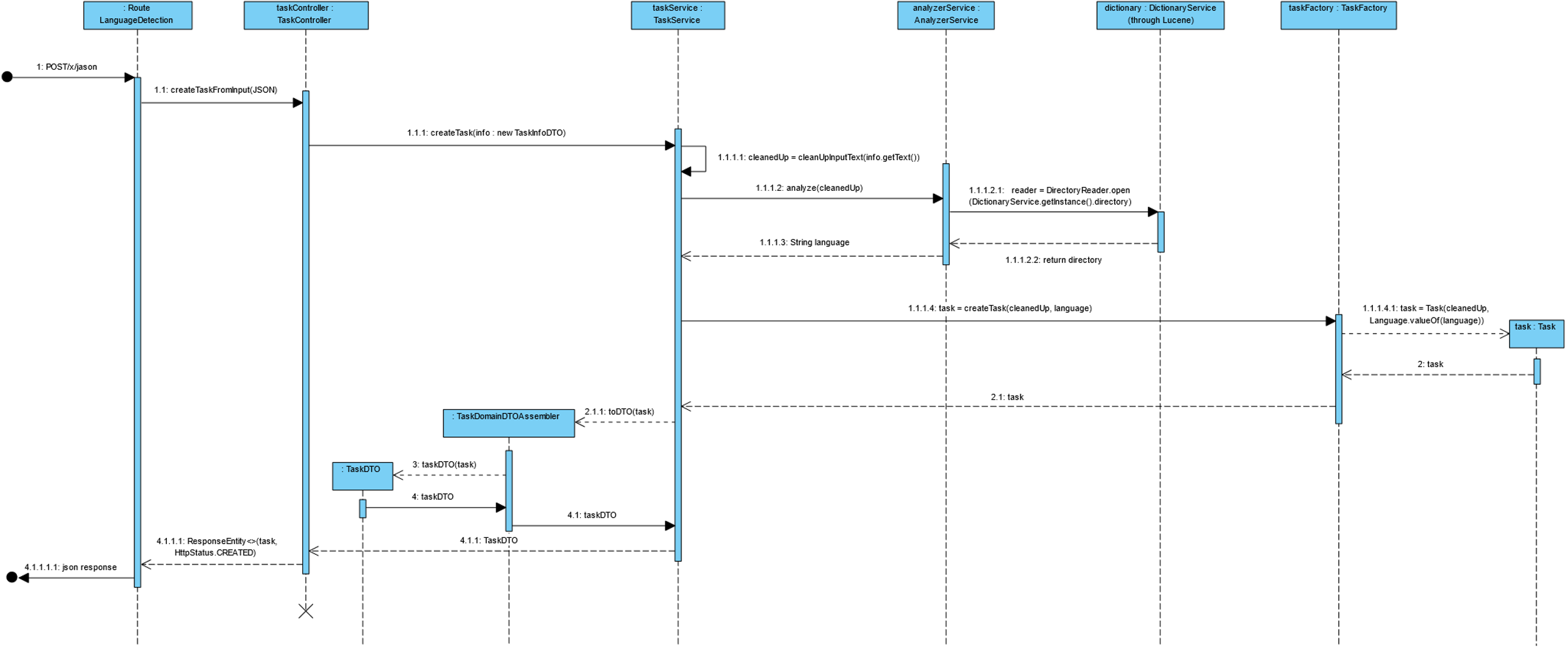
# 

# **System Sequence Diagram**

****

# 

# **Sequence Diagram**



**Note:**

In this diagram we are not representing the interior of the Lucene Classes that are the true connection to the dictionaries because it enters into the Library and, at this phase, it did not seem relevant to represent it.