

# Checkpoint 5 – Final Delivery

**Rodrigo Pinto**

Instituto Superior Técnico  
Lisbon, Portugal

rodrigo.melo.pinto@tecnico.ulis  
boa.pt

**Eduardo Miranda**

Instituto Superior Técnico  
Lisbon, Portugal

eduardomiranda@tecnico.ulis  
boa.pt

**Gilles de Waha**

Instituto Superior Técnico  
Lisbon, Portugal

gilles.dewaha@gmail.com

## ABSTRACT

UPDATED—8 November 2022. This paper describes the making of a visualization of the Pokemon TCG (trading card game). To understand this complex game and its specificities, we wanted to implement the ultimate visual interface.

## Authors Keywords

Pokemon TCG; visualization; boxplot; parallel coordinates; parallel sets.

## INTRODUCTION

Pokemon TCG has been through the ages and is still one of the most popular card games. A couple of years ago, while we were still in school, everyone used to play it. The competition was tough, we were all trying to have the invincible deck. Even if it was done in a naïve way, we were always trying to understand how each type measures against each other with the different stats, how rare cards measures against the common ones and what were the strongest cards to counter each type.

Some time has passed, and we are finally able to answer the questions we were asking ourselves as children, with a clear, easy to use and interactive visualization. To achieve that, we displayed how types affect the hp, how certain cards correlated according to level, hp, damage and energy cost and how is the distribution of cards between the different categories the following five attributes, evolution, rarity, type, resistances, weaknesses.

The questions we tried to tackle were:

- Is there a pattern between type and hp?
- Do the uncommon cards with a fire weakness have higher hp in relation to the common ones?
- Which cards with damage above 50 are stage 2 evolution?
- Does a high energy cost correlate with high damage if the card is a stage 1 evolution?
- Do the cards in the smallest flow-line have strong correlation between damage and energy cost?

## RELATED WORK

There exist many other visualizations about Pokemon cards, from which we of course got inspiration. However, we think ours is unique and answer best the questions we want to tackle. We found four related works and we will discuss them:

- [Data Visualization - Pokemon Data](#): This visualization about the Pokemon dataset contains many different idioms: there are two distplots displaying the hp distribution and the attack/defense distribution among cards; a radar chart showing for the selected Pokemon card(s) its (their) quantitative values and comparing them; two scatterplots to visualize: the correlation between hp and defense based on speed for the first one and the correlation between attack and defense of electric Pokemons over generations, based on hp; a simple bar chart displaying the number of Pokemon of each type; and finally three line charts showing: the trend of quantitative attributes by generation, type 1 and by every Pokemon of normal type.

This visualization is complete and answers our first question however as the dataset used is different from ours, (ours is from the trading card game and theirs from the video games) they do not have the weakness, energy cost and resistance, so the visualization does not answer our other questions. Despite this, this vis answers a lot of questions about the Pokemon video games dataset containing many different idioms. However, three major problems make it unclear and not very interactive. First, we are not able to see all the idioms on one page, making it harder to compare them. Moreover, this problem is emphasized by the lack of interactivity between the idioms. Selecting one Pokemon or one attribute on one chart will not highlight it on the other ones. Finally, it is impossible to change the items directly on the viz. For example, in the radar chart, the only way to choose another Pokemon is to change the code. It would have been much more user friendly to create a menu enabling to choose which Pokemon the user wants to display, like we did.

- [Visualizing the World of Pokemon](#): This visualization does not feature because as the previous one is using data from the Pokemon video games so the attributes previously mentioned are also missing. It contains four idioms: a bar chart showing the mean value of the quantitative attributes of the selected type of Pokemon; a line chart displaying the average of the quantitative

values of each generation; a bar chart to visualize the effectiveness of the selected Pokemon against all types; and finally, a scatterplot showing the correlation between defense and attack according to the type, sized by hp and with the possibility to select only the (non-)legendary Pokemons.

The most interesting idiom in this visualization, that maybe shows the data clearer than ours, is the bar chart displaying the selected Pokemon effectiveness against all types. It enables to have a quick overview about the strength of the Pokemon. Apart from that, our visualization answers better our questions and is easier to use thanks to the fact that everything is on one page and that the idioms interact with each other.

- [Gotta sort 'em all!](#) (An interactive visualization of the entire collection of Pokemon through type classification): This visualization shows all the Pokemons and classifies them according to their type. Moreover, it is possible to choose which generation of Pokemons you want to display. Finally, on the top of the visualization you can see the proportion of each generation in the total number of cards.

Even if it is a beautiful and interactive viz, it was not suitable at all for the questions we were asking ourselves. In fact, the visualization does not show any trend or interesting insights about the dataset, and it contains the generation attribute that we did not use at all.

- [Marvel vs DC: Comic Book Characters](#), one of our inspirations is a project from the last year course where the visualization had two search bars to compare two specific characters, a parallel coordinates comparing the stats between the characters, a unit chart encoding the characters by the year of their first appearance and a cleveland dot plot encoding the number of characters that have a certain power.

Obviously, our datasets are completely different so this visualization can't answer our questions. However, we got inspiration from the search bar and the parallel coordinates to answer our questions.

## THE DATA

### Data source

The data that we used in this project was from the Pokémon TCG official site that has a database with all the cards since 1999.

### Data cleaning

In CPIX to clean the dataset using pandas we started by removing the attributes we didn't need those attributes were removed because they will not be used to answer our questions. Then we discarded items that were not Pokémon, because in our domain the only cards that have all the attributes to answer the questions are the Pokémon cards. For the missing values we discarded every item that was missing a value for the rarity or attacks attributes, we discarded these items because the domain we are working on had many more items so discarding a few (12) was no big deal. For the outliers there weren't any, as in the dataset we are using, all the items are cards that exist in the game. We also simplified some columns as explained in CPIX, adding the damage and energyCost columns which are simplifications from the attacks attribute and a evolution column which is a simplification from the subtypes attribute.

### Challenges and compromises

On later checkpoints we had to reduce the amount of cards in our dataset because one of our idioms is the parallel coordinates and with the initial number of cards, this idiom was not readable. We also had to add the level attribute again to the dataset since the parallel coordinates was only using 3 quantitative attributes, so to make a little bit more interesting we added the 4<sup>th</sup> quantitative attribute (level).

## VISUALIZATION

### Overall Description

Our solution to the Pokémon TCG problem mentioned in the introduction consists of 3 idioms and a search bar.

The search is on the left of the screen to the right is the first idiom (boxplot), under this idiom is the second one (parallel coordinates) and on the right most side of the screen is the third idiom (parallel sets). With the search bar we can find a specific card to then locate it on the 3 idioms.

Our first idiom is the boxplot, the two axes of the idiom are the hp and type. In this idiom can find the outliers of each type, we can also see the median hp of each type and we can select each boxplot to then sort the 2 other visualizations by that type to only show card of that specific type. This chart has a customization that is a line across all the boxplots showing the average hp considering all the types.

The second idiom is the parallel coordinates where we have 4 attributes (hp, damage, energyCost, level) in this idiom we can select a specific interval in each attribute using a brush that will select all the cards in that interval and then update the other two idioms. We also can see which Pokémon is represented by a specific line if the number of lines in the idiom is 200 or less.

Our third idiom is the parallel sets where we have five attributes represented (types, weakness, resistance, evolution, rarity), we can switch the order of the attributes and the categories. We also can select two different flow lines that will show with different colors in the parallel coordinates for us to be able to compare them and update the boxplot with the set of cards that were selected.

### Rationale

In the boxplot we thought that by representing the outliers we could see which ones were the best cards since that was the one of objectives of the visualization, we also wanted to spot a pattern and compare how each type hp differs, so for this reasons the boxplot seemed like the best choice.

We chose the parallel coordinates in our visualization because we had a lot of quantitative attributes and this way we can compare different Pokemon stats side by side and filter them using the brush, so it seemed like the best option.

We also had five nominal attributes, so a parallel sets would be a good option to see the distribution of the cards between these attributes and it would allow us to filter categories, this combined with interactivity in the other two idioms would allow us to answer all of our questions.

In relation to scraped ideas we had in the first prototype a 4<sup>th</sup> idiom, a scatter plot, where we would correlate level and damage, but ultimately, we decided against it because the parallel coordinates could already correlate the two attributes.

### Custom Visualization

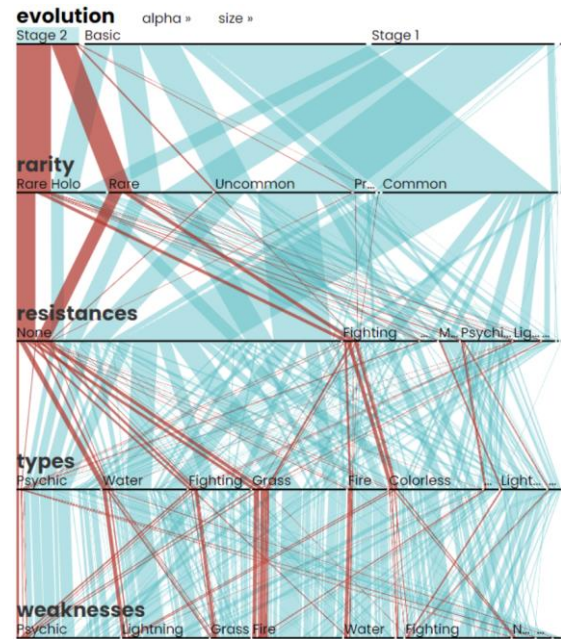
We chose the boxplot to be able to spot patterns between hp and type of the cards, to see if there was a type that had higher hp than the others and at the same time spot the outliers.

To extend it we thought about a bar chart that would show the number of cards of each type but since the parallel sets already shows the number of cards per type the extension would not be very useful.

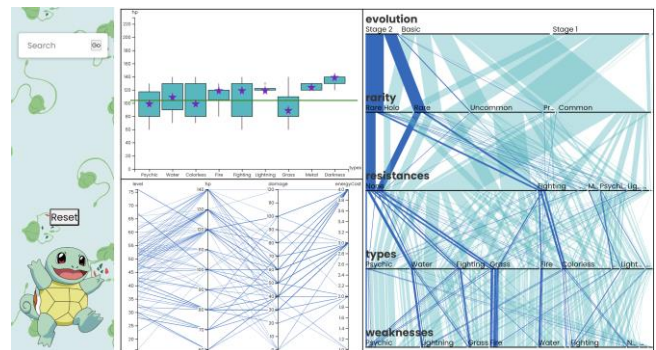
Our second idea was a simple line across the boxplot showing the average hp of the dataset, this extension is more useful because we can compare the average with the hp of the different types. It helps determine if a specific type or card is above average in the hp attribute allowing us to see if a card is good which is one of the objectives of our visualization.

### Demonstrate the Potential

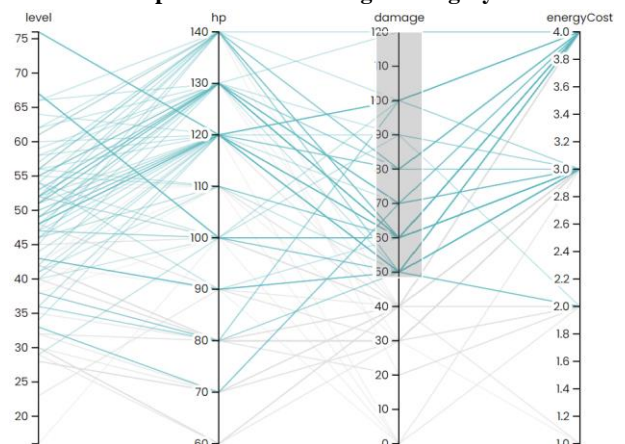
**Which cards with damage above 50 are stage 2 evolution?**



**Step 2 - Hover the stage 2 category in the evolution attribute**

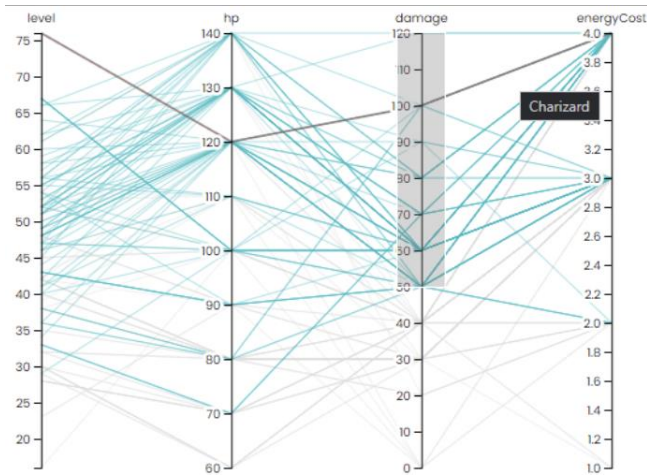


**Step 1 - Click in the stage 2 category**



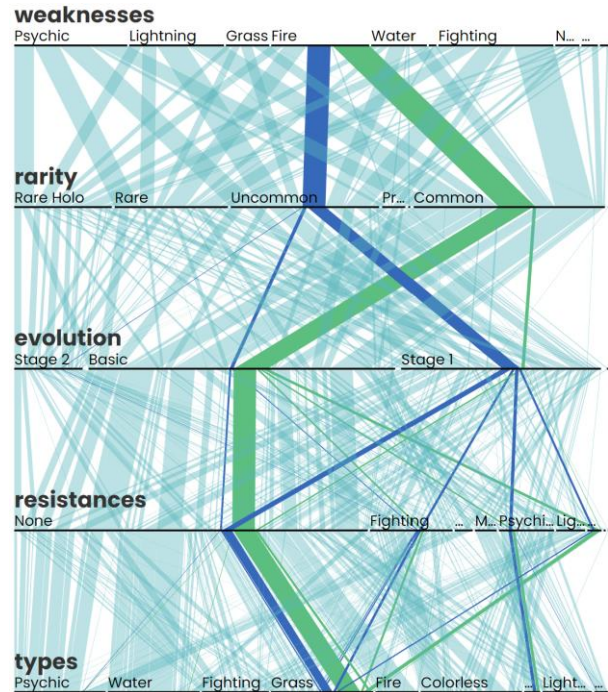
**Step 3 - In the parallel coordinates go to the damage column and create a brush from 50 to 120**





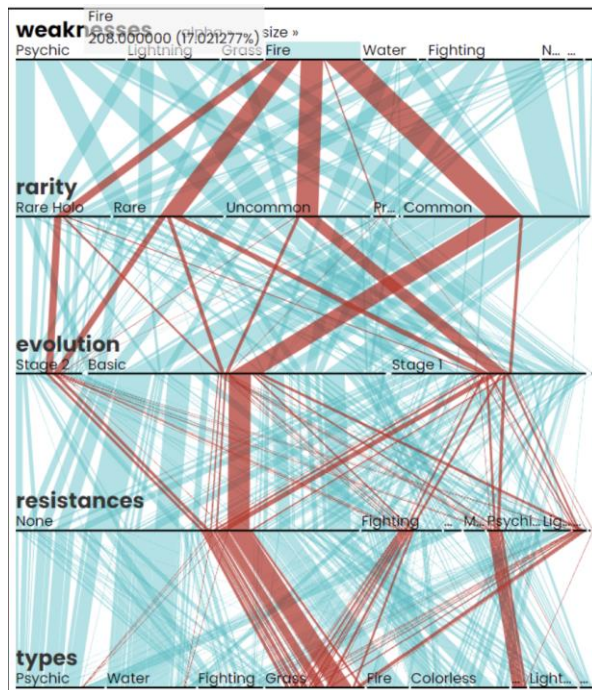
**Step 4 - Each one of the blue lines is a card that enters in the conditions of the question**

**Answer:** There are some pokemons that fulfill the filters of the question, Charizard being one of them.

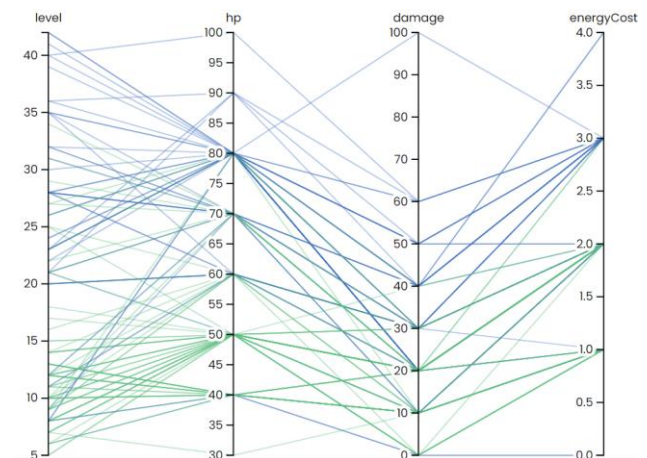


**Do the uncommon cards with a fire weakness have higher hp in relation to the common ones?**

**Step 2 – Select the flow lines corresponding to fire weakness with uncommon rarity and fire weakness with common rarity**



**Step 1 - Order the attributes with weakness first and rarity second and then hover over the fire weakness to see which flow lines go to uncommon and common rarities**

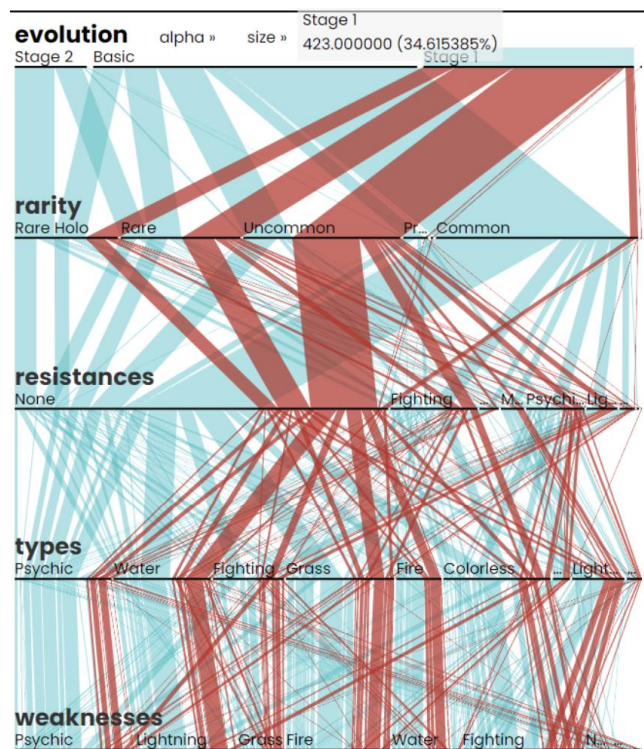


**Step 3 – Compare the hp of the two groups of cards**

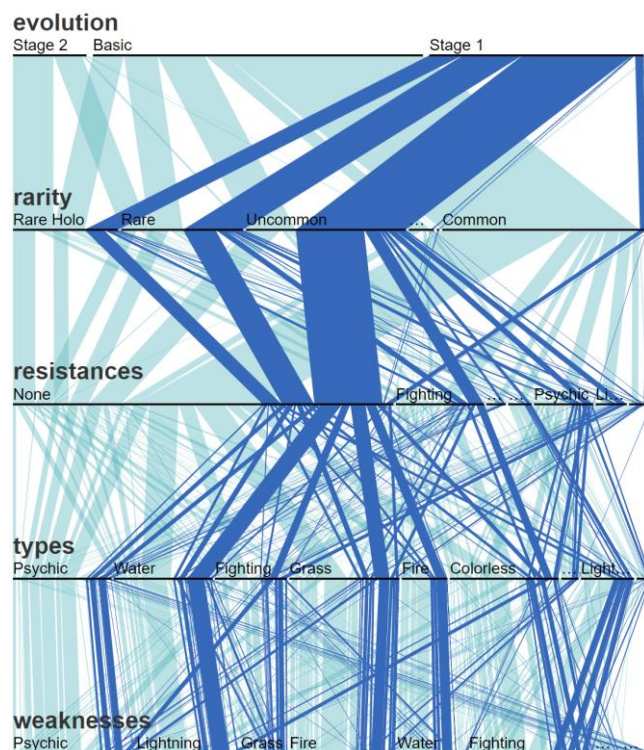
**Answer:** As we can see uncommon cards with fire weakness tend to have not only higher hp but also higher stats in general than the common cards with fire weakness.



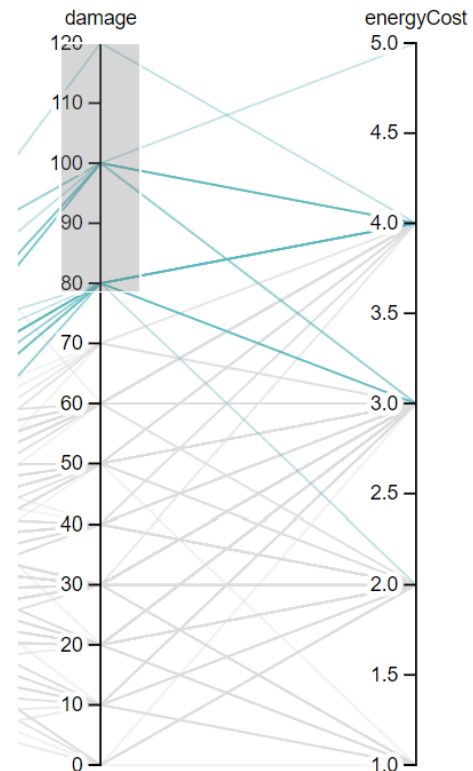
Does a high energy cost correlate with high damage if the card is a stage 1 evolution?



Step 1 – Hover the stage 1 category in the evolution attribute.



Step 5 - Click in the stage 1 category.



Step 3 – Brush the high damage or high energy cost and see if there is correlation.

**Answer:** We can conclude that there is a significant correlation between high damage and high energy cost if the card is a stage 1 evolution because by looking at the parallel coordinates, we can see that almost all cards with high damage have an energy cost of at least 3.0.

## IMPLEMENTATION DETAILS

The first challenge we faced was the learning curve of d3.js because it is a lot different from the usual JavaScript libraries. When we started the implementation of each idiom, we looked for some examples on the internet, but the vast majority was coded in older versions of d3.js, this was also a problem because there were significant changes between versions.

In the boxplot we started by looking at the example provided in the reference. Here our biggest difficulty was implementing the outliers position because we wanted them to be separated to be able to hover them, to solve this we created the placeOutliers function that puts them side by side if they have the same hp so that each outlier can be hovered.

In the parallel coordinates we followed the example provided in the reference and changed it to the way we wanted, here our biggest difficulty was implementing the brush because it was not working properly initially but we corrected while debugging.

In the parallel sets we followed the example in the references here we had two major problems, first the version problem discussed earlier and then implementing double selection. To solve the double selection, we adapted the highlight function provided in the example and added another event for clicking.

In the implementation of the links we followed the examples given by the course. In general, we create a new update function for each idiom that filtered the data according to global variables altered by other idioms, then this function would recreate the idiom according to the new data. This function is called every time a relevant filter happens on another idiom.

To implement the search bar, we created a function that reads the input in the search bar and then according to the input suggests pokemon names to the user. Then by searching for a specific card when the user clicks the button “Go” the specific card is highlighted in the idiom. This functionality was implemented by adding a new event to the button.

## CONCLUSION & FUTURE WORK

### WHAT DID WE LEARN?

By doing this project we can say that we learned a lot, but more about the Pokémon TCG theme, some examples of that are: the fact that the cards that are stage 2 have more damage in relation to other stages, the type that is most abundant is grass followed by colorless, the rarity influences the stats of the card because uncommon cards tend to have higher stats than common ones, the metal type pokemons have a higher distribution of hp than the other types but wailord is the one with the highest hp by far, etc...

With the implementation of all 3 idioms we were able to address not only almost all questions in Checkpoint I but also other ones, therefore we were able to take a lot of interesting conclusions like the ones referred above.

As mentioned above there is one question from CPI in particular that we were not able to address which is “*Do the cards in the smallest flow-line have strong correlation between damage and energy cost?*”, this happened because the question is badly formulated as there are some flow-lines that are the “smallest” because they have only 1 card in them and we did not specified in which order of attributes we were talking about.

Another thing that we learned by doing this project is how to clean a dataset using pandas and we now have a significant knowledge about d3.js library which could help us in the future.

### WHAT WOULD WE DO DIFFERENTLY?

Now that the project is finished, the first thing we would change is the question described above, we would formulate it differently so that it made more sense.

Another thing is probably the boxplot idiom because we could not really find a pattern between types, just compare

them with each other, this could be corrected by instead allow ordering the types and provide option to change the y-axis to other quantitative attributes or maybe change the idiom altogether. The other two idioms connect very well with each other and in some ways are letting the boxplot a little bit apart.

### WHAT WOULD WE DO TO ENRICH THE SOLUTION?

If we had one month more and 3000€ to spend on this we could create a search system that shows the card itself, expand the boxplot to also allow to combine other nominal attributes with other quantitative attributes, allow for the user to save favorite cards and compare them individually on the idioms and give cards suggestions based on the filters applied.

Finally, we could create an app to go alongside the web visualization to increase the interactivity and the user experience. In fact, he would be able to choose which idiom he wants to see, zoom in, see the image and the stats of every card... The money would be used to host the app and to release it on the App Store/Play Store.

## REFERENCES

1. Parallel Sets – A visualization technique for multidimensional categorical data  
<https://www.jasondavies.com/parallel-sets/>
2. Brushable Parallel Coordinates  
<https://observablehq.com/@d3/brushable-parallel-coordinates>
3. Boxplot  
<https://observablehq.com/@d3/box-plot?collection=@d3/charts>
4. Moodle VI  
<https://pcm.rnl.tecnico.ulisboa.pt/moodle/course/view.php?id=25>