UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat d'Informàtica de Barcelona

FIB

Barcelona School of Informatics
Polytechnic University of Catalonia

Bachelor's Degree in Informatics Engineering - Computation

Migration of EventExplorer to Web application and upgrades in SkyMap plotter

Defended on January 27, 2023

Victor Gregori Barrera
Javier Castañeda Pons - Institut de Ciències del Cosmos ICCUB
Marta Fairen González - Departament de Ciències de la Computació

Abstract

Scientists deal with lots of data daily, and sometimes it can be difficult for them to manage all. In order to help them, their team members make programs, so they can see the data and their statistics easily.

In the Gaia mission, they developed a Java desktop application called Event Explorer. The Event Explorer is used by the members of the Gaia team, so they can look at all the events and some information about them in an easy and comfortable way. In addition, this application can produce some plots based on the events loaded and filtered by the user. However, despite all these functionalities, the most relevant one, is the Sky Map generator. This functionality consists in the capacity of producing a density plot of the galaxy, where it can be seen which parts of the sky are affected by the events.

The objective of this project is to upgrade the application as a web application. Firstly, as said, we will migrate the desktop application to a web one, so the users will not have to download the application when they want to use it or when an update is released. Secondly, we will implement some interactivity upgrades to the Sky Map such as knowing the events that affect each pixel by clicking them, zooming or rotation. Finally, a database will be implemented, so the user will no longer need to load the events file.

Resum

Els científics disposen de moltes dades i informació al mateix temps, que a vegades, és complicat tenir-la tota controlada i utilitzar-la de forma eficaç. Per ajudar-los, altres membres dels equips s'encarreguen de fer programes de forma que puguin visualitzar les dades i, a més a més, estadístiques extretes a partir d'aquesta informació.

En la missió Gaia es va desenvolupar una aplicació d'escriptori utilitzant Java anomenada Event Explorer. Aquesta aplicació és feta servir pels membres de l'equip de Gaia per així poder veure el llistat d'events i informació sobre aquests, d'una forma pràctica i còmoda. A part del llistat, l'Event Explorer és capaç de produir gràfics a partir dels events carregats i filtrats per l'usuari. A més a més, a part de les funcionalitats esmentades, la més rellevant, és el generador de Sky Maps. Aquesta funcionalitat consisteix en poder crear gràfics de densitat amb una projecció de la Galàxia per poder veure quines parts de l'espai són afectades per cada event.

L'objectiu d'aquest projecte és actualitzar l'aplicació en una aplicació web. En primer lloc, com s'ha dit anteriorment, cal migrar l'aplicació d'escriptori a una tipus web de forma que els usuaris no hagin de descarregar-se l'aplicació quan la volen usar o cada cop que hi hagi una actualització disponible. En segon lloc, s'implementaran millores en la interactivitat dels Sky Maps de forma que es pugui rotar, fer zoom o bé veure quins events afecten un pixel en concret al clicar-lo. Finalment, se li afegirà una base de dades a la aplicació de forma que l'usuari no hagi de penjar els seu fitxer d'events cada cop que vulgui usar l'aplicació.

Resumen

Los científicos disponen de muchos datos e información a la vez, y a veces, es complicado tenerla toda controlada y utilizarla de forma eficaz. Para ayudarles, otros miembros de sus equipos se encargan de realizar programas de forma que puedan visualizar los datos y, además, estadísticas extraídas a partir de esta información.

En la misión de Gaia desarrollaron una aplicación de escritorio utilizando Java llamada Event Explorer. Esta aplicación es utilizada por los miembros del equipo de Gaia para así poder ver el listado de eventos e información sobre estos, de una forma práctica y cómoda. Aparte del listado, Event Explorer es capaz de producir gráficos a partir de los eventos cargados y filtrados por el usuario. Además, a parte de las funcionalidades citadas, la más relevante es el generador de Sky Maps. Esta funcionalidad consiste en crear gráficos de densidad con una proyección de la galaxia para poder ver qué partes del espacio son afectadas por cada evento.

El objetivo de este proyecto es actualizar la aplicación a una aplicación web. En primer lugar, como he dicho anteriormente, es migrar la aplicación de escritorio a una web de forma que los usuarios no tengan que descargarse la aplicación cuando quieren usarla o cada vez que haya una actualización disponible. En segundo lugar, se implementarán mejoras en la interactividad de los Sky Maps de forma que se pueda rotar, hacer zoom o ver qué eventos afectan a un píxel en concreto al clicar. Por último, se le añadirá una base de datos a la aplicación de forma que el usuario no tenga que colgar su archivo de eventos cada vez que desee usar la aplicación.

1 Introduction

Since the existence of the first human beings, we have been interested in what lies in the sky. At first, no one knew what were the lights people saw in the sky, later some women and men started to use them for orientation purposes, while others saw them as gods. As history passed, mankind studied what stars were and what else could we find above us. In recent years, thanks to all the research done in this topic, we have learned a lot of properties of all the celestial bodies and applications which help astronomical research grow faster.

With the desire to understand even more all these topics, the European Space Agency, or commonly known as ESA [1], was formed. The European Space Agency, with the headquarters in Paris, is an international organization founded in 1967 aiming to know how the space we see every night is structured, and to solve all the mysteries we haven't solved yet about our galaxy as well as the celestial bodies outside the Milky Way.

In order to reach its goal, ESA has a lot of missions that gather professionals from a number of different topics, such as physicists, computer scientists, or electronic engineers, just to say some of them. These mentioned missions cover topics like life outside Earth, with ExoMars; visualize the space itself, thanks to the James Webb Space Telescope in collaboration of the National Aeronautics and Space Administration (NASA); or the gathering of the properties of spatial bodies with Gaia, which we will talk later as it is the mission this project is about.

1.1 Context

Gaia's ambitious mission is to chart a three-dimensional map of the Milky Way. So it will reveal the formation, composition and the evolution of the galaxy we live in. Gaia provides positional and radial velocity measurements with accuracies enough to produce a kinematic and stereoscopic census of more than one billion stars in our galaxy. This amount of stars is about 1 percent of the Galactic stellar population.

The topic of this Degree Final project, or TFG as its initials stand for Treball de Fi de Grau, of modality (B), is set on the Event Explorer application made by the Gaia team members. This project belongs to the computing specialization of the Bachelor Degree in Informatics Engineering. However, it covers some topics of the Software specialization such as web applications software, and it is needed to evaluate the client needs for the application.

1.2 Concept

In the following section, we will be explaining the main concepts that need to be known in order to understand and to be able to follow the project without difficulties.

1.2.1 Gaia mission

As we said before, the European Space Agency holds a huge amount of projects and missions, one of them is the Gaia mission[2]. Gaia is an extremely ambitious project which was adopted within the programme of ESA in October 2000. In December 2013 the Gaia spacecraft was launched to space. Gaia will be acquiring all types of data over a nominal period of five years, plus five additional years of extended mission operation, until the end of the mission scheduled for 2025.

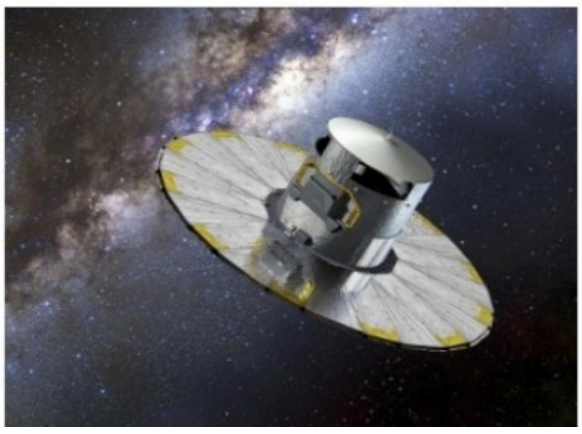


Figure 1: Gaia Spacecraft

Gaia's mission purposes are astrometric, as it aims to make a three-dimensional visualization of the Milky Way, making use of the unprecedented positional and radial velocities, the motions and parallaxes of approximately 1 billion stars measurements. These visualizations, obtained by Gaia, reveal the structure, the composition and the evolution of our galaxy.

Launched in 1989 and operated until 1993, the first satellite devoted to precision astronomy was Hipparcos, Gaia's predecessor. Thanks to its proven principles, Gaia is able to scan the galaxy. Its catalogue contains, for near 120.000 stars, 200 times accurate distances, movements and positions. However, Gaia significantly improved either the number of objects mapped as the precision of the measurements obtained.

1.2.2 Event Explorer

In the Gaia mission, sometimes some situations occur that affect the spacecraft directly or the mission itself, these situations will be called events from now on. These events can occur for lots of causes like some of them being failures on the downloading of the data, re-calibrations of the telescopes of the spacecraft or switching to safe mode, to say just some of them.

To evaluate the impact of these events caused in the data, some members of the Gaia team made the Event Explorer. The Event Explorer is a desktop application used to show information of the events and, what is more, to plot charts based on their statistics such as the duration of them or frequency. In addition, one of its most important features, not to mention the most one, is to plot density sky maps of our galaxy to know in which regions of the Milky Way more events have impact.

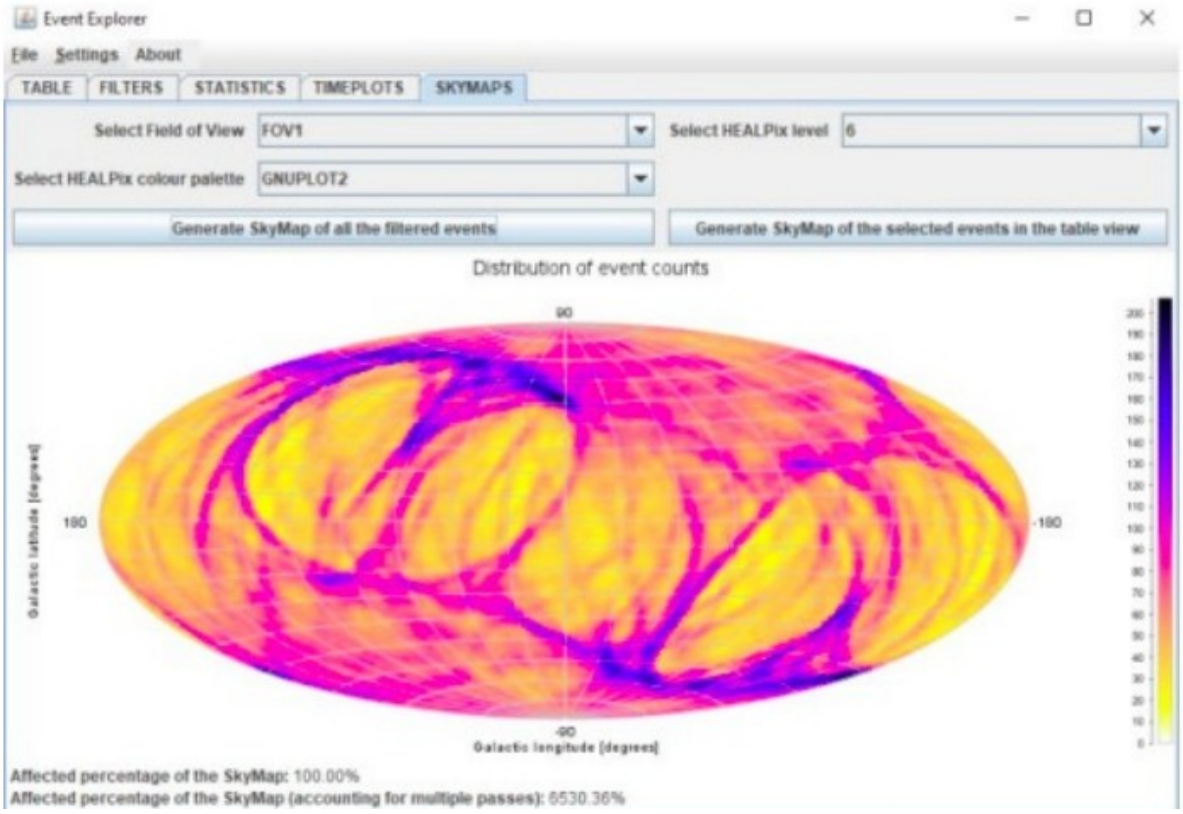


Figure 2: Screenshot of the SkyMaps view of the EventExplorer Java application

1.3 The Problem

As we said before, Event Explorer is a desktop application made in Java that receives the data, all the existing events until the date or just a subset of them, with a CSV file that has to be uploaded to the application by the user. As a desktop application, it is not practical for the members of Gaia to use it, as they have to download and install the application, and what's more, each time the application receives an update, they have to do it again. In addition, the fact that people have to upload the data by themselves, make it impractical as they must get the new data-set each time a new event occurs, to later on upload it again, so people can be up-to-date.

In addition, the current application has lots of limitations on its functionalities and features. Starting with these limitations, one of the most relevant ones, is the Sky Maps features ones. The problem is that the Event Explorer, when it's time to show the Sky Map, is only capable of showing an image of it, which causes the loss of interactivity with it. As it doesn't have interactivity, it is not possible to zoom in some regions of the chart nor to know which events are having impact in certain regions. Other limitations occur on the filtering of the events, the current application only allows filtering the events if they contain or not some certain text. More problems appear in the statistics section, as it is not comfortable to read, and it shows all the statistics all the time, while sometimes you only want to see a specific one, so you want to minimize the others.

Because of all these limitations we said before, despite the Event Explorer application being useful and accomplishing its goal, some members of Gaia don't make use of it. In conclusion, the problem resides in the need of some updates on its features, functionalities, and even more to how the application is made.

1.4 Reasons

As we have explained, the problem with the Event Explorer is that as it is a desktop application, it has to be downloaded and installed to make use of it, and it has to be done each time the application is updated. In addition, it has some limitations or lack of features. To solve this, I'll migrate the application to a Web based one, with two servers, one for the application and the UI itself, and another one for the database of the events.

1.5 Study of the options

In this section, I'll analyze different options for each part of the project so we use the most suitable option in each case.

1.5.1 Client side

For this project we have analyzed a bunch of different web frameworks in order to know which software will be the most suitable to migrate the Event Explorer as a web application. To do it, we have analyzed the three most popular frameworks which are Angular, VueJs and React.

All of them are really popular between the front-end developers as they give a lot of resources and ways to implement the client-side part of a web application, and all of them have really similar performances. We will try to see the differences between them and the main uses of each while we are analyzing their pros and cons, so the decision will be made easily.

Angular[3]:

Angular is a framework built in Typescript, so its learning curve it's a bit higher than usual, as for a starter web developer has to learn Typescript as well as the Angular procedures. Then, Angular is developed by Google, so it assures that it will be patches regularly to fix errors, and it will keep improving, despite lately it has been growing slower than when it came out. However, this is not always good, as the web applications have to be updated regularly to stay tuned to new versions and not get deprecated. Usually Angular is used in medium or high level of complexity web applications, thanks to the fact that it contains a lot of features that, at the same time, makes Angular bigger and slower, for example, when it has a lot of interactive components.

About its structure, Angular projects are structured into modules, components and services. Each component contains a template, written in HTML; a class that defines the logic and a Metadata (decorators), which it says where to find the building blocks that it needs to be created. Another thing to consider about Angulars structure is that it is a two-way data binding framework, in which every change on the class will automatically be reflected on the template, and the other way. Moreover, Angular is a single page application which means that all the web is maintained in one page, for example only in index.html instead of having index.html, login.html ...

Taking in account all the points, the learning curve of Angular is considerably larger than other frameworks, because of its great content of features, and as it's a framework without much flexibility, so the developer has to adjust as its way of function, which this at the same time gives more reliability and security at the time of making the software work. All of this, and if at the same time we add the fact that it works with TypeScript, it makes Angular a bad framework for small groups.

To sum up all, Angular is a framework designed with static Typescript to make high complex level pages as it contains a lot of features, and it contains a strong opinion on how the app should be designed, which causes a large learning curve.

React[4]:

Following with React, despite we group it with Vue and Angular, React is not a framework, it is a library instead, however it is considered as a framework in a lot of places. React is developed by another big software company as angular. This company is Facebook, and as it happened with Angular, React has always Facebook on its back making it better, as its errors are fixed easily and have improved features regularly.

React is a straightforward library only concerned for the UI components, so while Angular is used to make a whole application only taking in account Angular, react it's only used for the UI part. Also, the Model-view-controller design requires Flux to implement, but it provides more flexibility on how to organize your code.

As we said, React doesn't force you to have a specific project structure. React has the React elements, which are the smallest building blocks of apps, while components are larger building blocks that define independent and reusable pieces to be used throughout the app. React is based on JavaScript, but is mostly combined with JSX(JavaScript XML), a syntax extension that allows you to create elements with HTML and JavaScript at the same time.

As React is a library and not a framework, to start working on it, it is as easy as to import the library and start coding. However, a lot of users of React complain because the main language used is JSX, which is a tether as it doesn't help you to make the learning as easy as it should. You can use pure JavaScript as well, but as all the users do it with JSX, it is quite difficult to find information.

In conclusion, React is a straightforward library made to declare the UI components of a web application, with a not really beginner-friendly learning curve as it doesn't have clean code, plus the fact of learning JSX. With a good company on its back.

VueJS[5]:

Once we have talked about Angular and React, it’s time to talk about Vue.js. The first thing to consider from Vue is that unlike the others, Vue doesn't have a big company on its back, and it gets updated thanks to its big community. It is important to say, that among the three of them, Vue is the one with more stars on GitHub, which at least says a lot about its popularity and community.

VueJS is a progressive framework, which means that you can start having a tiny project, and whenever you decide you can scale it to a bigger project organically. Another thing to take in account of Vue, is that despite being a framework it gives you a lot of flexibility which makes it easy for people to start working on it, unlike Angular which is strongly rigid.

The Vue pattern is inspired by the Model-view-ViewModel pattern, which means that you will be working mostly on the ViewModel layer. Vue’s template syntax lets you create View components, and it combines familiar HTML with special directives and features. Vue uses Single File Components with the .vue extension. These files contain at the same time the HTML, CSS and JavaScript information, so it is a clean methodology to work with.

Vue’s learning curve, unlike the others, is shorter, as it doesn't make you learn new things, besides how the framework works, as it doesn’t use JSX and Typescript, you can use them, but it is not the best option. Another thing that makes it easy for beginners, is the fact that it’s mostly used to create small web applications without too many obstacles. However, one minor problem that it has, is that a big part of the community is Chinese, and sometimes it’s a problem to understand some.

In brief, Vue is a framework actually easy for new web developers to get introduced to, and with a clean structure of the project. However, as it is mainly used for small projects, it doesn’t have as many features as other frameworks like Angular.

Selection:

First, we want to say that we haven't talked at all about performance in terms of speed, because all of them have a really similar performance. It is true that for example, Angular can get a bit slow when it has a lot of components, but it's difficult to spot a large difference between them.

Then, If we talk about the learning curve, clearly Vue.js is the best option as you only have to learn how the framework works and the basis(Js,CSS,HTML), while React makes you learn JSX and Angular Typescript, which is useful to know them but adds a layer of difficult at the time of doing our project.

The structure of the projects are different, and while React, as it's a library and not a framework, doesn't have a clean structure of projects, Angular and Vue are better options. If we have to choose between them, we would say that the answer depends on the hours you want to spend on it, as both of them have a good and clean structure, but the Angular one is a bit more complex than the Vue, as we have said before.

In terms of features, Angular is the one that stands out more, as its big size allows it to have a lot of different and useful features and components. Vue despite being a tiny framework it would be a good option as it is still having a lot of features to use, but not on the same level as Angular.

To end with this part, all of them have an excellent support on their back. React and Angular have the biggest Tech companies, while as it can be seen in GitHub, Vue has a strong community of supporters that help Vue get improved regularly.

If we only tconsider the framework, Vue would be the best option, as it is useful for small projects, like the EventExplorer, and at the same time it has the opportunity to grow. In addition, we want to remark that Vue is the most beginner-friendly, so it will help to start the project. However, the framework is not the only thing to take in account, as Angular, for example, has it easier to work with different programs and has more features that will help searching for some way to plot the interactive SkyMaps. So once this has been said, before choosing one I'll see the compatibility with other programs that will be necessary for the EventExplorer like NodeJs.

ATTRIBUTES	ANGULAR	REACT	VUE.JS
Type	JavaScript framework	Open Source JS Library	Progressive JavaScript Framework
Npm weekly downloads (2018)	444,794	5,036,078	996,293
Size	167 KB production 1.2 MB development	109.7 KB production 774.7 KB development	30.67 KB production 279 KB development
Easy to learn	Steep (Learn TypeScript)	Moderate	Easy
Coding speed	Slow	Normal	Fast
Documentation	✓	✓	✓
Performance	✓	✓	✓
Startup time	Longer due to its large codebase	Quick	Quick
Complete web apps	Can be used on standalone basis	Needs to be integrated with many other tools	Requires third party tools
Data binding	Bi-directional	Uni-directional	Bi-directional
Rendering	Client side	Server side	Server side
Model	MVC	Virtual	Virtual
Code reusability	Yes	No, only CSS	Yes, CSS & HTML
When to use	Production, esp. enterprise apps with Material UI	Production, custom UI apps	Startups, production

Figure 3: Table summarizing the properties of the different frameworks

1.5.2 Server Side

In the previous section, we talked about the three main client-side frameworks that are used in lots of projects, so seeing all the differences between them, it would be easier to choose which one is more appropriate for the migration of the EventExplorer to web application. In the end, we considered that Vue would be the better option followed by Angular, the only reason is that Vue is easier to start with, however if doing it with Angular, the time spent on learning the framework would be rewarded by more features available on the framework. As we said before, this only consider the framework itself, without taking on account the rest of programs that will be used to migrate EventExplorer, that's way in this section we will try to explain the server side programs that will be used, and how they are compatible with the frameworks seen before.

Firstly, instead of start explaining all the server-side services that exist and can be used we want to remark that we filtered them first by the most used ones, as we did with the framework before, and we got four server-side services that are Flask, Django, NodeJs and Deno. The first two used Python, while Node and Deno used Javascript and Typescript(in the case of Deno). As Django and Flask use python, we decided to filter them out, because we consider that it is better to use as much as possible services and programs that make use of the same programming language like JavaScript, so while working with Angular/Vue and at the same time with Node or Deno, the only language that will be used will be JavaScript, and maybe Typescript, depends on the final decision.

Node.js[6]

Node.js is a JavaScript runtime that can execute JavaScript code outside the navigator, in other words, Node is used to execute JS code in the server-side, so it's only necessary to know JavaScript in order to make callbacks to the computer, like: accessing to specific ports, read and write files, etc. All of this thanks to the V8 engine, written in C++.

Node declares itself in its website as an “asynchronous event-driven JavaScript runtime, Node.js is designed to build scalable network applications”. Let’s analyze part by part what this sentence refers to. Firstly, it says that node is an “asynchronous event-driven”, this means that NodeJs can handle the events of the back-end, like accessing to some file in an asynchronous way, which refers to the fact that when an event occurs, NodeJs is able to handle it without making the CPU get blocked. That way, Node in order to handle a lot of events, it never gets blocked as the thread is executing some program, when this needs some file from the database, instead of waiting for the file to be delivered to the program, it starts with another program, so it doesn’t stop the execution for the I/O, and later on, when the file is ready it comes back, and follow the execution of the first program. Following with the sentence, it says that it is a “JavaScript runtime”, this refers to the thing we talked about in the first paragraph, so we won’t explain it again. In the last segment, it says that “is built for scalable network applications” which means that it is build so all the applications made with it are able to scale horizontally in the case that they grow and have to handle more events than usual.

To sum it all up, Node is used for all the back-end parts, so thanks to the V8 engine and NodeJs, it is possible to execute JavaScript code to do some server and CPU callbacks easily. In addition, all of this is made asynchronously, so it’s faster as it can handle multiple events at the same time without getting the CPU blocked. Finally, it is important to remark that Node.js is widely used for lots of famous applications like Microsoft, LinkedIn, Netflix, and some others.

Deno[7]:

Ryan Dahl, creator of Node.js, in May 2018 released Deno as a new JavaScript runtime designed to fix a long list of inherent problems of Node. Despite, NodeJs is still a really good and popular JavaScript runtime, in the words of its creator, there are still lot of thing that he didn’t consider too much at the time of developing Node.js, some of these things, but not all, are: security, modules and dependencies.

Deno is an event-driven Typescript runtime, despite you can use it as a JavaScript runtime too, it is not recommended, as use it with Typescript gives you more functionalities and it how all users do it, so it will be much more documentation with Ts than with JS. As a difference with node, which is written in C++, Deno is written in Rust as its eventl oop. Following the differences between Node and Deno, as Deno tries to solve the problems of node, it focuses more on its security. As opposite as Node.js, when you try to access to: the file system, network, the execution of other scripts and the environment variables, it prompts some requests as it doesn't permit all this accesses, so you have to give it permissions when executing, while Node execute it without asking. If we talk about modules, it is actually simple to import them as you can import it only with the URL.

However, using Deno it's not only good things, despite it improved in a lot of aspects the problems that Node had, it's still really new, only 4 years since it was released, so it lacks of documentation about it, and despite it will grow with the time and will gain importance and features, nowadays, it doesn't arrive at the same point of relevancy as NodeJs. Besides, the fact that it is mostly used with Typescript, it adds a layer of difficulty as you have to learn Typescript besides the Deno commands.

Conclusion and Compatibilities:

Now that both Deno and Node have been seen, and we remarked its differences and its pros and cons, we would opt for Node.js as, despite it still has some problems that can be solved by using Deno, the difference on the lack of documentation and the fact of having to learn Typescript to use Deno properly it makes it much more difficult than just picking Node.

As we considered Node.js as a better option than Deno, it's time to see if it is compatible with the frameworks that we have talked about before, which were Vue.js and Angular. At first, we opted for Vue.js as its learning curve is shorter and as it is more suitable for small projects like the Event explorer. However, Angular it's still a viable option as a framework if there is some problem with the compatibility between services.

About their compatibility with Node.js, gladly both of them are compatible with nodejs, and it is not difficult to implement it. In both cases it is as simple as having the framework and node installed, creating the project, and modifying some files to connect them, and it should work, later on we will see it more in detail. Despite both of them working well with node.js, there are still some services that are needed to see before choosing the framework. The section below will talk about the plotter that will be used for the migration of the EventExplorer, so it will be necessary, too, to see its compatibility.

1.5.3 Plots and Graphics

For this project it is necessary to find software which is able to plot the statistics of all the events, and at the same time it is necessary another software, or not, that can be able to generate Sky Maps that can handle some kind of interactivity, as it is one of the main reasons of this project. When researching about it, we found the Chart Js library that can be used for the plot side.

Chart.js[8]:

The EventExplorer is a desktop application that needs to be migrated to a web application, for this, it needs some kind of JavaScript plotter, to be able to plot all the bar time plots either for events per day as events per week and at the same time the scatter plot. In this case, I'll use Chart.js. Chart.js is an open source JavaScript charting that allows designers and developers to produce different types of charts according to their needs.

Chart.js allows you to plot 8 kinds of different charts, and the mix of them. These kinds of charts are: Area chart, Bar chart, Bubble Chart, Doughnut and Pie Chart, Line Chart, Polar Area Chart, Radar Chart and the last one, Scatter Chart. For the EventExplorer, the only charts needed are the bar chart and the scatter one, as are the ones used in the web application.

Another good thing to take in account about Chart.js is the fact that in its website, you can find different types of integrators to use Chart.js with different frameworks as Vue.js and Angular. These integrators are vue-chartjs and ng2-charts. To use them, in both cases, you only need to install the libraries, and later on in the project, import them. Once imported, in order to use it, you only need to create a HTML canvas, and in a new script create a new Chart using the canvas of the HTML document. Also, you can choose the data set that will use the type, some options… In the following figure, there is an example of how to create a bar chart.

```
<canvas id="myChart" width="400" height="400"></canvas>
<script>
const ctx = document.getElementById('myChart').getContext('2d');
const myChart = new Chart(ctx, {
    type: 'bar',
    data: {
        labels: ['Red', 'Blue', 'Yellow', 'Green', 'Purple', 'Orange'],
        datasets: [{
            label: '# of Votes',
            data: [12, 19, 3, 5, 2, 3],
            backgroundColor: [
                'rgba(255, 99, 132, 0.2)',
                'rgba(54, 162, 235, 0.2)',
                'rgba(255, 206, 86, 0.2)',
                'rgba(75, 192, 192, 0.2)',
                'rgba(153, 102, 255, 0.2)',
                'rgba(255, 159, 64, 0.2)'
            ],
            borderColor: [
                'rgba(255, 99, 132, 1)',
                'rgba(54, 162, 235, 1)',
                'rgba(255, 206, 86, 1)',
                'rgba(75, 192, 192, 1)',
                'rgba(153, 102, 255, 1)',
                'rgba(255, 159, 64, 1)'
            ],
            borderWidth: 1
        }]
    },
    options: {
        scales: {
            y: {
                beginAtZero: true
            }
        }
    }
});
</script>
```

Figure 4: Code showing the base syntax of the ChartJs

D3Js[9]

However, in the initial GEP report, we didn’t talk about any software used to generate the Sky Maps. Initially, we thought that we could use WebAssembly to use some Java code that could plot the desired density map, inside the Javascript code. But we didn't test it out, so we didn’t know which software would be used for this section. As the project progressed, we realized that there is not enough documentation to make it possible to work with WebAssembly, Java and Vuejs at the same time. During the project, thanks to colleagues of other missions inside the astronomic sector and asking senior programmers, we came up with using the D3Js library, which allows us to make some plotting and interactive maps.

D3js or Data-Driven Documents library is a JavaScript library that allows you to manipulate documents based on data to life using HTML, SVG or CSS. Thanks to it, you can use the full capabilities of modern browsers without having to use a proprietary framework. D3 allows you to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document.

In order to modify documents using the W3C DOM API is tedious, as the method names are verbose, and the imperative approach requires manual iteration. So that is why the D3 uses a declarative approach, operating on arbitrary sets of nodes called selections. These selectors are supported actively by modern browsers like Chrome or Firefox. In addition, in D3 you can specify as functions of data styles, attributes or other properties, so they are not just simple constants, which has a powerful performance despite seeming simple.

As the D3js gives you such powerful and useful features, we will use it to generate the Sky Maps and to implement all the interactive features around them. However, as the plotting features are already made, it would be necessary to change them, and try to implement them with D3js too, but it will take an extra time than the expected in the initial GEP report.

1.5.4 Database

Although we know we haven’t talked yet about how to reproduce the SkyMap on the Web, first, we will go into detail on the database. For the database, the software that will be used will be mysql as it is already being used in other projects of the Gaia missions in the ICC-UB. So it won’t be necessary to learn it from scratch. Also, the compatibility with nodejs is made thanks to the npm mysql module. That allows you to connect and make queries to the database easily.

2. Scope and Methodology

2.1 Objective

The main objective of this project is to take the Event Explorer application made by the Gaia team members, and analyze its features and functionalities, to adapt them to a web based application. In other words, to migrate the Event Explorer java application to Web and at the same time, improve or add functionalities to different parts of the application, but mainly the Sky Maps generator as it is the most relevant

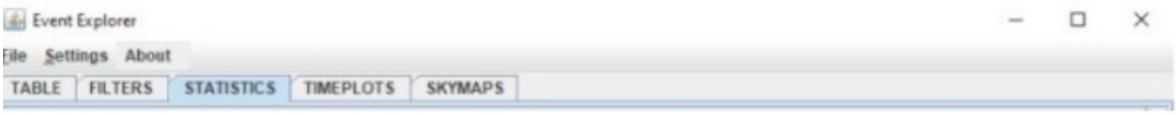


Figure 5: Screenshot of the tabs menu of the EventExplorer Application

In the figure above it can be seen all the main features that the Event Explorer contains and have to be implemented.

2.2 Methodology

These are the main steps that will be done to accomplish the goal of the project:

1. **Search and analysis of the software used**
    a. Search and analyze the main frameworks used in web applications
    b. Search and analyze different server-side tools
    c. Search and analyze data bases software and servers
    d. Search and analyze other softwares that will be used during the project such as plotters.
2. **Implementation of the structure of the application**
    a. Implementation of the main menu
    b. Implementation of the menu functionalities
3. **Implementation of the different tabs**
    a. Implementation of the Table tab
    b. Implementation of the Filters tab
    c. Implementation of the Statistics tab
    d. Implementation of the Timeplots tab
4. **Implementation the SkyMaps Tab**

e. Search a way to show a SkyMap of the density of the events
f. Implement the SkyMaps functionalities
g. Implement a tooltip feature to the sky map generated
h. Implement different options for the sky map such as colors or regions to generate
i. Implement the zoom features.

5. **Implementation of the Server-side application**
    j. Implementation of a basic back-end of the application
    k. Addition of a database to the application
    l. Add hosting server functionalities

The main methodology to do this project will consist of following the main steps mentioned before, however the order of the steps inside the main ones will probably differ.

2.3 Obstacles and Risks

One of the main obstacles that can be found during the development of this project, is the fact that the software chosen, VueJs, is relatively new, and it's still emerging, so it is possible to have a lack of information on certain topics that can be related to the Event Explorer. Regarding this, it can be possible that because of the lack of information or compatibility with other softwares, it is necessary to change the softwares half the way in order to make the web work.

Despite the lack of information of VueJs, another kind of obstacles that will be found will occur due the fact that this project is not only a TFG, besides it is to an application for the Gaia team, so it can cause that some members will ask for functionalities that are outside the scope of the project, so it will have to be implemented and perhaps replace some of the feature that are thought to be done.

3. Time planification

The final degree project usually starts during the month of September, at the same time the new college course starts, and ends in December/January when the project has to be submitted. However, in my case the project started in July, so it will last at least for roughly 6 months, as the project needs to be delivered as late as 19th of December. Considering the project started on 11th of July, this TFG will be done during a period of 162 days, if we exclude weekends and holidays, it will remain a total amount of 108 days. During the first 3 months, the hours dedicated to the project will be around 7.5 hours a day, however in the last 3, it will be 5 h a day, so this makes it nearly 700 hours. Despite having that huge total of hours, probably most of them like one hundred or more will not be used, because as the project is reaching its final, I'll be assigned other jobs that will take off hours from the Event Explorer migration.

3.1 Resources needed

For this project, like all the projects, it will require some kind of resources for the development of it. Here, I'll be describing which and in what they consist of the following point.

· **Human Resources (R1):** As it is logical, this project will depend on the people of the Gaia team. Like the director and co-director, who will help with the organization, give some tips, and in the case of the director it will look over the project and give ideas on how to improve it or what to do. Later, there is the programmer who will be responsible for all the design and development of the webpage.

· **Hardware (R2):** For the project, two computers will be used, the first one will be a desktop computer located in the office of the Institut de Ciències del Cosmos - UB, and later a personal laptop to use when working from home.

· **Software (R3):** All the software used will be open software. Such as, Trello, for the management of the tasks, Visual Studio for the project development, Subversion as a version controller, etc.

3.2 Tasks descriptions

In this section all the tasks that we mentioned superficially before, will be described and explain why we will assign a certain amount of time. In addition, we want to say, that the total sum of the task times, it will be lower than the one said before, as that time is the maximum possible time to do the project, however probably either some hours will be dedicated to some other tasks as we assigned before, or the project will be finished before the total amount is reached.

3.2.1. Project management(T1)

This part of the project is the most important one, as thanks to it, the project will be structured correctly, and it will be much easier to work. This section goes from the GEP assignments to the writing of the report of the project, passing by the reunions and reports to the directors of this TFG.

Scope and Contextualization(T1.1)

This section it's one of the most important ones in the whole project, as we said before it is the one that establishes the basis of the project. Here are the objectives, the work that will be done, the context and the reason on all the topics mentioned are explained in detail. The time used in this section was 30 h.

Time Planification(T1.2)

If the tasks were explained and shown in the scope and contextualization, in this section they will be assigned an estimation of the time needed to complete each task. For this task, the time employed it's around 15 h.

Budget and Sustainability(T1.3)

In this task, the budget needed to do the project will be calculated and analyzed. In some ways, the task is the same as the last one, but only taking into account the economical points. This task doesn't affect much in this thesis as, despite its part of an organization, all the services or software that will be used are free. So the task one takes too long, probably around 8 h.

Report(T1.4)

The report of the project is the organization and management task that will take the longest time. The task consists of describing and explaining all the things done during the thesis. In addition, all the GEP have to be added to the report and be modified to adapt the changes made during the project. We estimate that the task will take around 100 h.

Meetings(T1.5)

Here will take into account all the reunions that will be made, either the reports that will be done twice a week with the director and some other staff members, which will be used to propose changes and functionalities, and the reports that will be done with the co-director of the project. As the regular reunions will be around 15 minutes each, so it will be 11,5 h plus the four or five 1h long reunions to report to the co-director it will be 16 h.

3.2.2 Previous Work and Research(T2)

Despite all the knowledge learned during the degree, here we will take in account some more time to do an extensive research on the most suitable software for this project. As there are lots of different softwares to migrate the application, and there are lots of possible options, we would consider that the research time will be around 40 h (T2.1). Aside from that, it is needed to consider the time necessary to gain all the necessary knowledge to use the software picked before. This time it will be an amount of 20 h (T2.2).

3.2.3 Design and Implementation(T3)

In this last section, the parts that will be taken into consideration will be the tasks which are related to the development of the web application.

Design(T3.1)

The design part is extremely important, as a bad decision in the design of the application will lead to fewer people who would use the application. This section will need lots of time as it considers the time needed to design either the GUI and the formats and functionalities that will be used. Taking into account the points mentioned before, the total time will be around 60 h.

Development(T3.2)

The development part, as its name suggests, is about all the development of the application. This will take into account all the tabs, the GUI of the application, the functionalities and the SkyMap generators. As it's the most important section of the application, and at the same time the most extensive one, the time dedicated will be around 230 h.

These 230 hours will be distributed in different sub-activities that are part of the development of the project. Firstly, we will have the development of the menu interface and its functionalities, this will take around 40 h (T3.2.1). After that, another sub-task would be the development of each tab, but the Sky-Map, which will have its own task. All of these tasks will be around 35 hours each, three tasks, 105 h. (T3.2.2 x 3). Finally, the remaining hours will be dedicated to the Sky-Map, which makes a total of 85 hours for the last task(T3.2.3).

Testing(T.3.3)

At the same time the development section will be done, the testing part will be done, as it is necessary to test all the functionalities during the development. This section will need lots of time as there are quite a lot of functionalities, we estimate it will be around 80 more hours.

3.3 Time Planification Update

To end this topic, we want to remark if some changes need to be made in order to have coherence between the first planification and the status of the tasks made. Most of the tasks are being done in a time approximate to the one estimated at the start of the project. However, as we said before, as the Statistics tab will have to be partially remade,if there is sufficient time, it will need around 15 hours to finish it. But as a part of this, the project is being on schedule.

Code	Task	Time	Dependencies	Resources
T1	**Project Management**	**169h**		**R1,R2**
T1.1	Scope and Contextualization	30h		R1,R2
T1.2	Time Planification	15h	T1.1	R1,R2
T1.3	Budget and Sustainability	8h	T1.1,T1.2	R1,R2
T1.4	Report	100h	T2,T3	R1,R2
T1.5	Meetings	16h		R1,R2
T2	**Previous Work**	**60h**		
T2.1	Research	40h		R1,R2
T2.2	Acknowledgment	20h		R1,R2,R3
T3	**Design and Implementation**	**370h**		
T3.1	Design	60h		R1,R2,R3
T3.2	Development	230h	T3.1,T3.2	R1,R2,R3
T3.2.1	Menu	40h		R1,R2,R3
T3.2.2 x3	Tabs	35h x 3	T3.2.1	R1,R2,R3
T3.2.3	Sky-Map	85h	T3.2.1, T3.2.2	R1,R2,R3
T3.3	Testing	80h	T3.2	R1,R2,R3

Table 1: Table of times, dependencies and resources dedicated to each task

4. Budget

In this section, we will do a study of the budget plan destined to this project in order to investigate and estimate the budget of this project and if it's viable the options offered. We will divide the budget into three different types: staff, generic and unforeseen circumstances.

4.1.- Staff budget

To make an estimation of this budget, firstly, it is necessary to know which kind of staff will participate in the project. Foremost, there is the project manager which is shared between the director, the co-director and the author of this TFG, which at the same time has a second role which is the developer of the thesis. In the following table, we can see which are the average salaries of those roles.

Role	Gross Salary	SS	Retribution
Project Manager	15,65 €/h	5,48 €/h	21,13 €/h
Software Developer	10 €/h	3,5 €/h	13,5 €/h

Table 2: Gross Salary per hour for the roles of the project

Once we have the salaries of both the project manager and the software developer, it's time to take a look at the Gantt chart, so we can calculate the budget we need to dedicate to the staff in this project for each activity done. In table 2, we can see the hours dedicated by each role to the corresponding task.

Code	Task	Project Manager	Software Developer	Time Cost
T1	**Project Management**	**169h**	**0h**	**3.570,55€**
T1.1	Scope and Contextualization	30h	0h	633,83€
T1.2	Time Planification	15h	0h	316,91€
T1.3	Budget and Sustainability	8h	0h	169,02€
T1.4	Report	100h	0h	2.112,75€
T1.5	Meetings	16h	0h	338,04€
T2	**Previous Work**	**0h**	**60h**	**810€**
T2.1	Research	0h	40h	540€
T2.2	Acknowledgement	0h	20h	270€
T3	**Design and Implementation**	**0h**	**370h**	**4.995€**
T3.1	Design	0h	60h	810€
T3.2	Development	0h	230h	3.105€
T3.2.1	Menu	0h	40h	540€
T3.2.2 x3	Tabs	0h	105h (35h x3)	1.417,50€
T3.2.3	Sky-Map	0h	85h	1.147,50€
T3.3	Testing	0h	80h	1.080€
	Total	**169h**	**430h**	**9.375,55€**

Table 3: Table of the cost of each task and the hours corresponding to each role.

4.2 General Budget

In this section will be all the costs that don't depend on the tasks done, besides there will be the general costs such as the energetic costs, the amortization and more.

4.2.1 Amortization

For this project, in terms of hardware, we will need a desktop computer, a screen, and a laptop to work from home. The cost of the computer is around 500 €, while the cost of the laptop is a bit less, around 400€. The screen cost is 100€. If we assume a lifetime of 5 years(60 months) for the three products, we have a total cost of 5/60 x (500+400+100) = **83.3** €. Regarding the software, all the software used is open source, so it’s free of charge.

4.2.2 Workspace

The project will be done in the offices of the ICCUB located at the Parc Científic de Barcelona(PCB). The cost of the office goes around 300€/month. If we consider that the project last 6 months, and there are 3 people working on the office, its cost is (300*6)/3 = **600€**.

4.2.3 Energetic Cost

Doing a quick research, we can find the average cost of the price of the electricity in Spain at the working hours goes around 0,33782 €/kWh. Considering that the power hours used by the desktop computer and the screen are the same as the laptop, half of the total working hours for each we have: (30 W(screen)+200 W (desktop PC)* 265 h + 50 W(laptop) * 265 h =74.2kWh; 74.2kWh * 0,33782 €/kWh = **25€**.

4.2.4 Total Generic Costs

Concept	Cost
Amortization	83.3€
Workspace	600€
Electric Consum	25€
Total Generic Costs	**708.3 €**

Table 4: Cost of each generic cost and its total

In table 3, we can see a summary of the total generic costs of the project justified previously.

4.3 Contingencies

As with every project, we have to consider the possibility that some obstacles will appear which can make the project to last longer than it is supposed. For this, we will make contingency plans that will help to confront those situations. Usually in software, projects these contingency levels are around 15%. Thus, the contingency cost is (6944.85 € + 708.3 €) *0.15 = **1147.97 €.**

4.4 Total Cost of the Project

In the following table we can see the sum of all the costs calculated in the previous sections.

Concept	Cost
Staff Costs	9.375,55€
General Costs	708.3€
Contingencies	1147.97€
Total	**11.231,85€**

Table 5: Total costs of the project

4.5 Management Control

During the whole project some revisions will be done in order to calculate the deviation of the estimated costs so we can try to minimize them. we will use the following metrics:

Deviation of the Cost = (EC - RC) * RHS
Deviation of the consum = (EHS - RHS) * EC

with:

EC: Estimated Cost
RC: Real Cost
EHS: Estimated Hours Spent
RHC: Real Hours Spent

Thanks to this control, we can detect if there is some deviation from the estimated planning, and in the case there is, we can use the contingency plan in order to cover up all the extra costs. If the costs are much higher than expected it will be necessary to make up the planning again in order to have faithful planning.

5. Sustainability

Thanks to the survey held by “mon universitari per EDINSOST”, once we answered it, we have a better understanding of my knowledge on sustainability in the IT sector. Now we consider that my level is not too high as expected for a college student. However, now we can analyze the flaws we have on these topics, such as in a theoretical way or a practical one. As a reflection of why it is that way, we think it is due to the absence of these kinds of topics during the projects done during the degree, as they weren’t meant to have a repercussion in the society or the environment. Besides the knowledge, we think my degree on the awareness of this topic is not as low as the knowledge.

In addition, we want to remark that the computing specialization focuses on the research, so it doesn’t have that much impact on the environment as, usually it only problem is the electricity consumed during the computation of the task, which is less, than the impact on the environment caused by the manufacturing of the computer and its components which is more relevant to the Hardware specialization.

5.1 Environment

During the making of the project, we have estimated in some way the environmental impact that the project is causing. As we said before, the only impact of the project is the energy consumed while doing the project, and during its lifetime. This only can be reduced using the minimum amount of hours to finish the project, so the cost will be lowered. As in the State Of Art, the cost of its lifetime is pretty much similar to the project because the only impact as we said multiple times is the energy needed to host the application. So in both cases it will be the amount of energy consumed by the server that has the application running.

5.2 Economical

As it can be seen in the previous section, the total amount of budget needed for the project is described there, so it is not necessary to extend more in this topic. However, if we compare this project with the state of art we can say that we could reduce the cost of it, as the state of art uses more powerful software or closed source software.

5.3.- Social

This project had a great impact on me as its the first big project I've done in my life so it gave me a lot of experience. However, it will have an impact on the gaia mission members as they will be the ones to use it in order to help them in their spatial research. So despite it not having a direct impact, it helps too to the scientific community, more specifically the astronomic one.

6. Background

In this section, now that all the preparation for the project has been explained, let's start with the project itself. First of all, before starting the implementation itself, some background information is needed to fully understand how the project works or how it is implemented. First we will explain what the HEALPix algorithm is, to later describe how we use it in order to make the Skymap thanks to the Spacecraft attitude and the time the events occur. Finally, to have some idea on how the project is made, besides the main purpose of this project is the Skymap, and not the web migration itself, we will explain the basis of how web application works and Vue too.

6.1 HEALPix

HEALPix[10] is an acronym for Hierarchical Equal Area isoLatitude Pixelisation of a 2-sphere. Healpix is an algorithm for pixelisation a 2-Sphere, and as it names suggests, each partition covers the same surface area as every other one.

The main purpose in the ideation of the Healpix algorithm consisted in the necessity to measure cosmic microwave background (CMB) anisotropy in satellite missions, like NASA's Wilkinson Microwave Anisotropy Probe (WMAP) or ESA's Planck mission. Some principal requirements were to create a mathematical structure which supports a suitable discretization of functions on a sphere at sufficiently high resolution, and to facilitate fast and accurate statistical and astrophysical analysis of massive full-sky data sets. Thanks to the following essential properties, HEALPix can satisfy all of these requirements:

1. The sphere is hierarchically divided into curvilinear quadrilaterals, where in the lowest resolution the sphere is composed by 12 base pixels. We will call it the first level of pixels. To obtain greater resolution, each of the 12 regions is divided into 4 more, and recursively, each time the level is increased each of these new subdivisions is partitioned in 4 more, giving a total of 12 * 4^n regions, where n is the level of resolution. In the figure below, it can be seen an example of HEALPix for the first 4 levels.
2. At a given resolution, all pixel areas are identical.

3. Pixels are distributed on lines of constant latitude. This property is essential for all harmonic analysis applications involving spherical harmonics. Due to the iso-latitude distribution of sampling points, the speed of computation of integrals over individual spherical harmonics scales as ~N1/2 with the total number of pixels, as opposed to the ~N scaling for the non-iso-latitude sampling distribution.

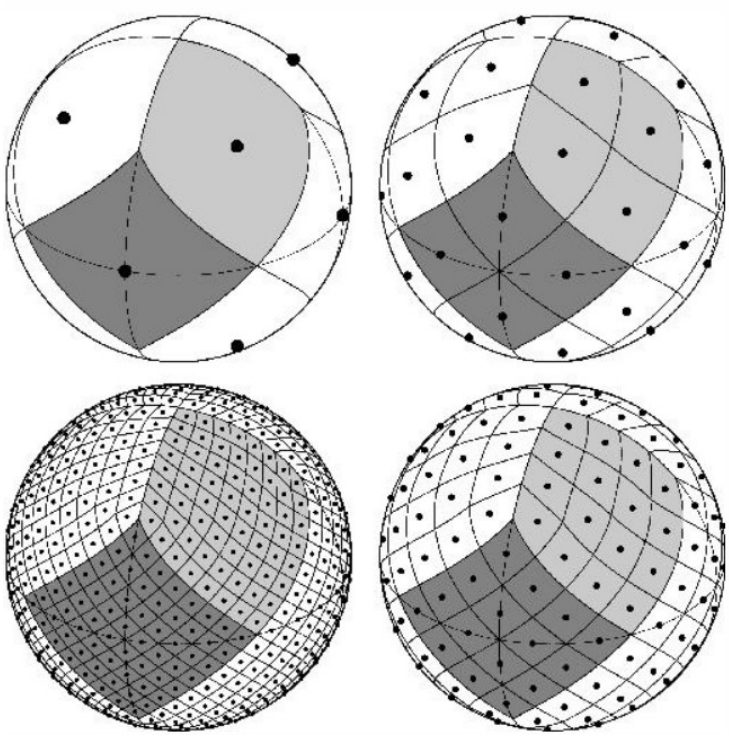


Figure 6: Different levels of resolution of the HealPix algorithm. In clockwise order, level 0 (12 pixels), level 1(48 pixels), levels 2(192 pixels) and level 3 (768 pixels)

As we can see in the top left image of the figure above, when the resolution level is set to 1, the sphere is divided into 12 base pixels. These twelve pixels are divided in three parts, four of them are on the top of the sphere (North pixels), then four more are situated on the bottom part(South pixels), and finally the last four of them are in the middle of the sphere(Equatorial pixels). Once the resolution increases, each pixel is divided into 4 more pixels located on the north, east, south and west parts of the original pixel.

HealPix allows two different pixel numeration schemes. The RING and the NESTED. The first one, consists in enumerating the pixels from north to south along each iso-latitude ring. While the NESTED consists in creating a tree structure in each of the 12 base pixels, so when the resolution increases it is as easy as to, in binary code, add the corresponding suffix to the base pixel for each new pixel. For the help it gives at the time to define the new pixels, we will use the NESTED numeration scheme, as there can be needed to change the resolution level in the EventExplorer application.

6.2 Space Attitude and AttToHealPix

In the last section, we explain what HEALPix is and how it works. Thanks to it, now we can understand the type of outputs we will use in the application, and some base knowledge on one of the main tools that will allow us to generate SkyMaps of the Milky Way.

However, to fully understand this project, we need more background on how these HEALPix Ids are obtained. Hence, in this part we will explain the theory on how we obtain the HEALPixes with the AttToHealPix algorithm starting with the attitude of the Gaia spacecraft. To start on this, we'll explain what is the attitude of the Gaia spacecraft. The attitude of Gaia represents how the orientation of the spacecraft is in a determined time. When using the attitude, a precise determination of it is required to be sure the region of the sky seen in a time t is the correct one. To obtain it, initially, will be from the Attitude and Orbit Control System (AOCS) represented by quaternions. The quaternions provide a convenient and elegant mathematical notation for representing orientations and rotations. From these quaternions, a B-spline (a Bèzier curve generalization) will be adjusted and used to interpolate the attitude for any time during the mission.

In order to understand the algorithm used to obtain the Id of the Healpixes corresponding to the events that need to be plotted, it is necessary to give some basic information about all the reference systems used on it. As it can be seen in Figure 7, the first reference system is the Barycentric International Celestial Reference System (BCRS/ICRS), which is a reference This reference system only moves with the Gaia spacecraft and is defined kinematically non-rotating with respect of the ICRS/BC from the CoMRS, we will introduce the Scanning Reference System (SRS), which is a new reference system co-moving and co-rotating with the spacecraft, which is mainly used to define its attitude. Finally, there are two Field-of-View Reference Systems (FoVRS), one for each sky field seen by Gaia, with their origins at the center of mass of the spacecraft and with the x axis pointing to the optical center of both fields of view. The angular celestial coordinates in each FoVRS are called field angles (η and ζ). In figure 8, we can see how are the coordinates of the Focal Plane Arrays of both Field of View, that thanks to them, Gaia spacecraft Focal plane, which consists of 106 CCDs that sum up to 1 gigapixel, can gather all the necessary data for the mission.

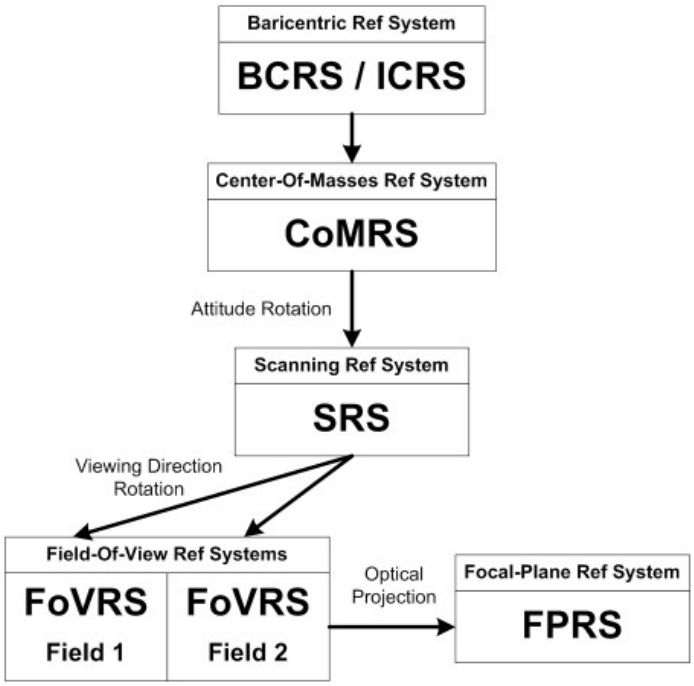


Figure 7: Diagrams of the coordinates systems of the Att2Healpix

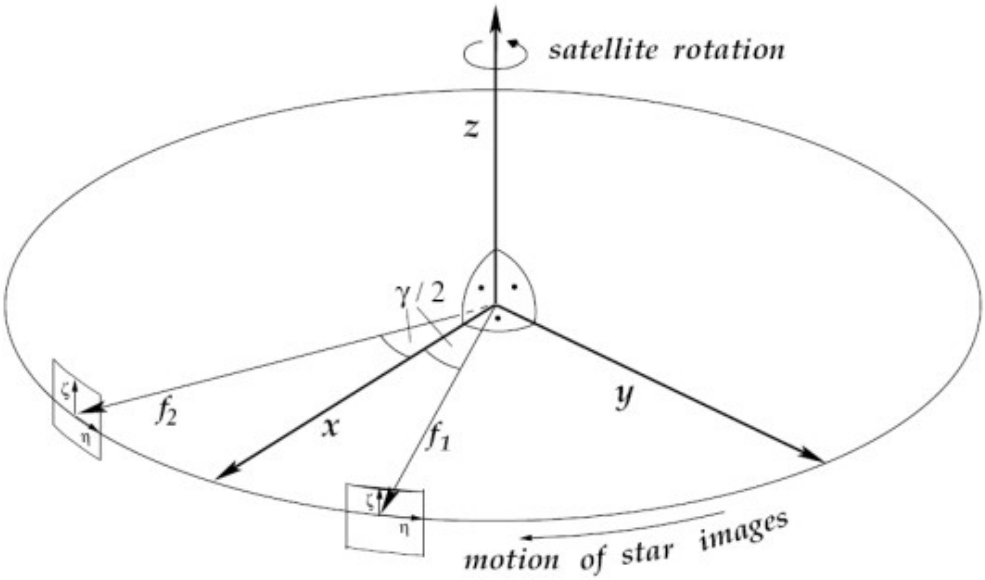


Figure 8: Representation of the coordinates of the spacecraft and its axis

Once the reference systems are all explained, we can proceed on how we obtain the pixelId set for each field of view. Firstly, as the inputs we will need will be the time when the event occurs, this will be defined by a starting time t0 and a finish time t1, which each of the events has it as a field of information about them. In addition, another input is the HealPix Level, this will be generally set to 6, as it's a level that doesn't require a high computation requirements, and the resolution of the map we obtain is good enough. Finally, it is necessary to give the algorithm the coordinate system. Ee will use the Galactic coordinate system, as it's used more commonly. There won't be any problem to see the Sky Map in different coordinate systems as the application itself will be the one in charge of changing the by applying some rotations, more detailed explanations later.

The AttToHealPix algorithm will take the inputs explained in the paragraph above, in addition, it will obtain the attitude of the spacecraft in the desired span of time, which as we explained is one of the inputs. Thanks to the attitude, the algorithm is able to obtain the HealPixelsId relatives of the events. To obtain them, the algorithm takes the attitude of the spacecraft and knowing how it was oriented on a certain time, the algorithm traces a ray from each field of view that will point to the respective HealPix pixel with the level of resolution passed as an input. The output of the algorithm will be 2 sets of Healpixels, one for each field of view, in the coordinate system set as an input. These pixels will be in the nested scheme as it's the one commonly used in the Gaia mission and will allow us to implement more easily some functionalities or upgrades in the EventExplorer.

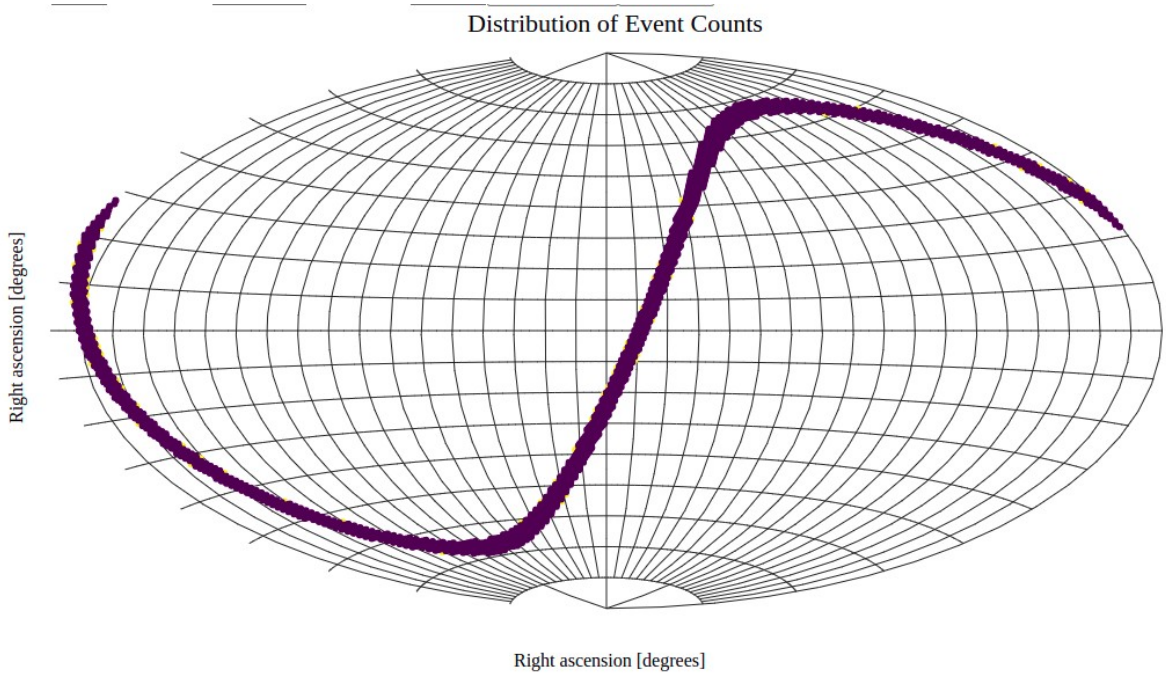


Figure 9: Strip of healpixes generated and drawn by the SkyMap plotter

6.3 Web development and Vue js

To put an end to the background needed to understand the project, it is necessary to have any insight on how the web applications works. Web application development consists mainly in two parts: the backend and the frontend. We call the backend of an application to all the parts relative to the server and its logic, while we say frontend to what the client sees, in other words the interface and the web itself. Then we will proceed to explain them in more detail.

6.3.1 Backend

As we said before, the backend of an application or the server-side consists of what's in the server and the logic in it, in other words as the name says, it is what lays in the back part of the application, the part that people can’t see. This part encompasses multiple things, such as all the highly cost algorithms, the management of the database and the data uses, and the connection with the frontend.

Each time the client performs an action such as filling a contact form, making a purchase or typing a mail address,1 the information is sent to the server, so it can be processed by the backend. This is done because the browser is not able to perform such complex algorithms in an efficient way as the browser doesn’t have as powerful computation power as the server. The backend too is in charge of the management of all the data and the database. So whenever it is necessary for the application to obtain, modify or delete some data, the server-side will create a connection with the database and make the operation.

Another thing the backend part manages is the connection with the frontend. This is mainly done using two methods. Those methods are the GET and POST methods. Below, we will try to explain the differences between them, and when it’s better to use one or the other.

On one hand, there is the GET method. This method is used to request data from a specified resource. Using the GET method, the query string is sent to the URL of the webpage. This allows a limit on the length of the GET request as the URL is limited, maximum URL length is 2048 characters. Other things that need to be considered is that as the requests are sent via URL is less secure as the information can be easily seen. So it is not recommended to send passwords or sensitive information using this method.

On the other hand, the POST method is used to send data to a server to create/update a resource, such as the database. As a difference from the GET method, the data sent to the body is stored in the request body of the HTTP request, instead of the GET method that sends it to the server in the URL. This allows the POST methods not to have restrictions on the data length. Another thing to take into account is that this method is a little safer than the first one explained, as the parameters are not stored in the web server logs or in the browser history.

To sum up, it is recommended to use GET when the data that needs to be sent is neither too big, as it has a limit on the size of it, nor is in a format different from a string, and the most important, it is not some type of data that needs to be sent in a secure way such passwords or sensitive data. In addition, as the data is sent in the URL, it’s more efficient. While we will use the POST method, when the data it’s not a string, or contains either lots of information, too big for a GET query string, or sensitive data that cannot be easily leaked like passwords. In the following table are the main differences between both methods in a more visible way.

GET	POST
In GET method, values are visible in the URL.	In POST method, values are not visible in the URL.
GET has a limitation on the length of the values, generally 255 characters.	POST has no limitation on the length of the values since they are submitted via the body of HTTP.
GET performs are better compared to POST because of the simple nature of appending the values in the URL.	It has lower performance as compared to GET method because of time spent in including POST values in the HTTP body.
This method supports only string data types.	This method supports different data types, such as string, numeric, binary, etc.
GET results can be bookmarked.	POST results cannot be bookmarked.
GET request is often cacheable.	The POST request is hardly cacheable.
GET Parameters remain in web browser history.	Parameters are not saved in web browser history.

Figure 10: Table summarizing the main differences between GET and POST methods.

6.3.2 Frontend

The frontend or the client-side is the part in charge of all the interfaces and the visible part, in other words, all that is relative to the client that uses the web application. As we said, the frontend includes everything the user experiences directly, that goes from text, images and colors to buttons, scroll bars and navigation menus. It is important to make a difference between the designer of the application and the frontend developer. The first one, is only in charge of creating the design and choosing the colors of the website, while the second one is who makes it possible using different languages which are HTML, CSS and JavaScript.

HTML or HyperText Markup Language is the fundamental coding markup language that creates and organizes web content, so it can be displayed by a browser. HTML is the language in charge of the elements that appear in the web page. The elements are defined by tag, compressed by <> characters.

While HTML is the language in charge of the elements of the website, CSS or the Cascading Style Sheets language is responsible for the style of the web page. It is what people use to change the characteristics of the markup like color or size of the text or to create simple animations.

Finally, JavaScript is the main programming language used in website development. It is mainly used for the interactive elements like drop down menus, modal windows, and contact forms.

To put an end to this section, we will talk about how Vue works and some of its functionalities.Vue is a JavaScript framework for building user interfaces. It builds on top of standard HTML, CSS, and JavaScript and provides a declarative and component-based programming model that helps you efficiently develop user interfaces, be they simple or complex. One of the most used features in Vuejs are the Single File Components (SFC), also known as *.vue files. These files are HTML-like files that, as the name suggests, encapsulates the component's logic (JavaScript), template (HTML), and styles (CSS) in a single file. In the following figure it can be seen as an example of these types of files.

```vue
<script>
export default {
  data() {
    return {
      count: 0
    }
  }
}
</script>

<template>
  <button @click="count++">Count is: {{ count }}</button>
</template>

<style scoped>
button {
  font-weight: bold;
}
</style>
```

Figure 31: Template of a .vue single file component

Then, vue gives a lots of functionalities. First, there is the v-on directe to allow listen the DOM events to call functions. Due to its frequent use, v-on alsho has a shorthand syntax @. Later, thusing the directives v-bind and v-on, we can create two-way bindings on input elements. In addition, there is the conditional rendering that allows us to render and element depending on the value of some boolean condition. To finish with the .vue functionalities, I'll talk about the list rendering. With the v-for directive, we can render a list of elements based on a source array.

7. EventExplorer: Design and Implementation

Next, we will start with the practical part of this project. We will talk about different topics or parts of the application from two different points of view, one more focused on the design and the other in the implementation. Some topics that will be explained will be the structure of the web application, like the menu, the tabs and the filters section. Another topic that will be discussed is the backend of the application, how it works, and which algorithms are used to make the website work correctly. Finally, which is highly related to the backend topic, we will put emphasis on the SkyMap generator.

7.1 The Structure

First, we want to remark that, as the main objective of this project is to migrate the EventExplorer desktop application to a web application, we tried to preserve most of the structure and functionalities, however it will be seen during the whole project that despite trying to maintain the basics, things such as the filters, or the style have been completely changed, as it was necessary a better approach to them.

Figure 12: Screenshot of the tabs menu of the EventExplorer Application

As it can be seen in the figure above and Figure 21, the two menus of the desktop application, the tab menu and the menu with the main options(File and about), have been adapted to a unique menu with all the tabs, unless the filters one which we'll go later in more detail, and a dropdown menu, that appears once the cursor hovers over the three bars button. The tabs menu only changes the tab seen in the screen to the respective one. However, the dropdown menu has different options. It is divided into two different sections. The first, with the name of File, is used either for loading the table of events or to clearing it. It contains three different buttons. One is used to upload a .csv file containing a table of events, the second one, as the name says, only clears the table of events, and lastly in the Load from DB button, you can load the table of events from the database that contains all of the events. Finally the about section only pops up a new window with some author information.

As for the implementation of the structure, the application is composed of a main .vue component called App. This component is the one in charge of managing the data needed by more than one tab, such as the data of the table that is needed in all of them. This main component contains the rest of tabs as .vue subcomponents that only need to be shown when the button is clicked. A part of all the tabs, there are two other childs components: the Filter and the Menu. For the filter, it's only shown once the user has loaded a table. Instead of being another tab, as in the original application, it has been implemented that can be accessed from all the tabs to modify the table from everywhere. Similarly, the menu has been implemented so it can be accessed in every moment.

In the following figure, it can be seen the components tree, and which components are dependent on other components. As we have seen, the main component is the App.vue, which contains the 5 different tabs components, the menu and the filters. Then, we can see that the other component with children is the statistics, as it has one for each statistic which are: dataSegmentTable, DurationStats, GenerealStats, OffsetsStats, TimeStats, TypeStats.

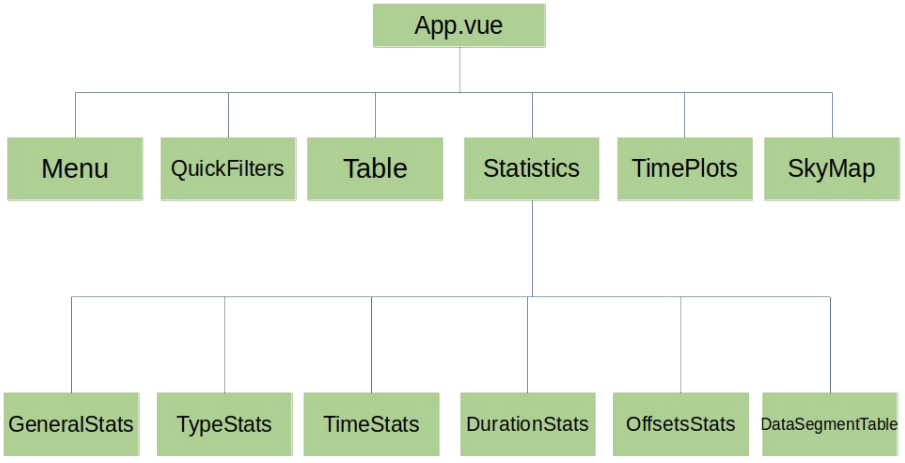


Figure 13: Diagram of the components used for the Event Explorer Application

7.2 Table

The table tab that has been the tab that has received fewer changes in comparison to the original application. When you open it, it consists only of an empty table without any other element. However, once you load either from a file or from the database, the table filled with all the elements appears. All the colors to differentiate the type of events, the columns and its order have been maintained. The only changes that have been applied to the table are the addition of the new column Index, which now will be a property of each event to identify it later for the SkyMap, explanation on the SkyMap tab section, and the suppression of the event selection feature. Each time that some information would be computed, it would be from the filtered table. Then, the same way the last application would do it, if some row is double-clicked, it will appear a modal with all the information of the event clicked. All the columns can be resized, and you can sort the table by column just by clicking on the header. Finally, one addition respecting the original work, is that if the mouse is hovering over some cell of the table, a tooltip appears with all the information of that cell, in order to be able to read all content of the cell without having to resize the respective column.

As for the implementation of this tab, there is nothing to remark. The table was made thanks to the Vue features that allow us to list all the headers and the rows as elements of the table with a simple syntax. For the rest of the tab, the vast majority of it was made thanks to the css, even the colors, the resizeable elements and the tooltip.

To put an end, if you look at the following figure and figure 21, as we said in the structure section, all the filters, including the quick filters of the table tab and the ones in the filters tab have been moved to one menu which appear when the purple square button with the + symbol is clicked.

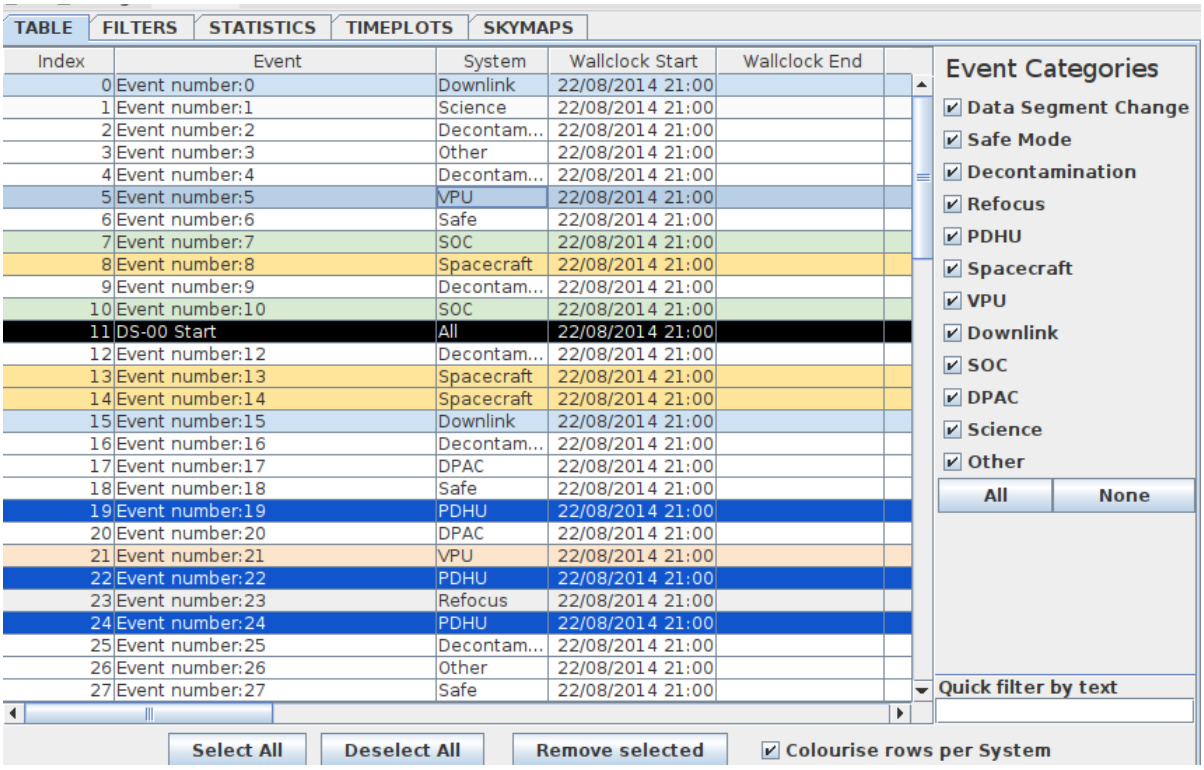


Figure 44: Table tab of the EventExplorer Java application

7.3 Filters

Following the topic, the filters are one with more changes on it. First, as we already mentioned, they all have been wrapped in a single menu that can be accessed from all different tabs. This menu, which starts minimized in the form of a purple square button with the + symbol is clicked, once it is clicked it shows all the different filters that it has.

To understand the changes made on the filters, first we will take a look at Figure 15, which shows the filters of the original application. In the application, you could add a bunch of different conditions, all related if the event contains or not different texts. Then you could choose if those events needed to be hidden/shown and if the logic to combine them was the AND or the OR. Then in Figure 14, the quick filters, only were a simple filter by text and for the category of the event.

In this project, we went for a more basic approach on what filters referred to, and made a menu with a bunch of basic filters, that are just necessary to make an advanced search for a certain group of events. To start, if we look at figure 22, there are two text fields, where the first one is used for search events that include a certain text and the other is used to exclude the ones with that text. Each of these text fields is accompanied by a button, with a text that says columns, and once is clicked, a modal appears with a checkbox for each column. These checkboxes decide in which columns the filter applies. Later, there is a checkbox to decide if the table has to be colourised according to the categories. Then there is the category filter, which consists of a button that makes a modal appear with checkboxes to choose which event categories need to be shown. Similarly, later it can be seen a button that has the same effect in the columns instead of the categories and a button that allows the user to choose if some data segments need to be hidden or shown. The data segments are periods of time between some key events that are already marked as the data segments category. The application, once it loads the table, takes into account the events loaded and computes the data segments for this filter. Finally, the last two filters work similarly. Both are range sliders, one is used to decide the duration of the events shown, and the other is thought for the time they occurred. For the time slider, the user is able to choose if the values are in the form of dates or revolutions[1]. As for the duration slider, it can be chosen too, if the value is in revolutions, hours.

For the implementation of the filters, all of them, once they are changed, they send the information to the App object, and there the app object checks all the rows of the table if they pass the filters or not. For the set-up of the filters, once the user loads the data, first, the application makes a search for the elements inside the data segment category and computes the periods between each of them to have the data segments. This will be used later on the statistics tabs, as there it shows a table with the data segments and their respective times. Then, the application searches for the boundaries of the sliders. For this, when it is doing a walk through all the events, it looks for the maximum duration and for the last and first event. At the same time, it makes the conversion to the other possible value types, such as the revolutions or the dates.

1Revolutions: A revolution happens when the Gaia Spacecraft describes a full circle, it lasts 6h

Figure 15: Filters tab of the EventExplorer Java application.

7.4 Statistics

For the statistics tab, the main functionality has been preserved. All the statistics that are shown in the original application are still being in the new web application, figure 23. In addition, a new statistic has been added. This statistic is a table that contains the start time and the end date of all the data segments in period of time. Although in terms of functionality it is still mostly the same, the main changes are in the design. The problem of the original application is that it is not practical for the user to look at them as all the statistics are located in the same column, so in the case that the table contains lots of different event categories, the user has to scroll down in order to see some part of the statistics. To avoid this problem, we designed the application such that each statistic can be placed in the desired order and in different places of the window. Three regions have been implemented to drop these draggable statistics. In addition, those draggable items can be minimized, so the user is able to decide which statistics are shown at the same time. Below, we'll list the statistics items and what they mean:

- General Statistics:
    - Total number of events loaded: A count of all events loaded.
    - Total number of events shown: A count of all filtered events.
    - Total range: The range between the smallest OBMT Start and the Largest OBMT Start/End
    - Total range length: Difference between the largest OBMT Start/End and the smallest OBMT Start

- Number of Events per Type: Here it displays the total number of events for each event category. The categories are: Data Segment Change, Safe Mode, Decontamination, Focus, PDHU, Spacecraft, VPU, Downlink, SOC, DPAC, Science and Others.

- Total Time Covered by Events: As the name says, it lists the categories and their respective sum of the total time covered by them. The units are in revolutions.

- Statistics of Event Durations: It lists some statistics about the duration of the events. Those statistics are: The total sum, the minimum duration, the maximum, the mean, median and the percentiles at values 10 and 90.

- Offsets: Here it displays some statistics about the offsets between events, the offsets are calculated with the difference between the end time of an event and the starting time of the following event. The statistics are: Minimum, Maximum, Mean, Median, Percentile 10 and Percentile 90

- Data Segment Table: Table with the starting time and ending time of each data segment

About the implementation of the statistics tab, there are a couple of things to be remarked. First, all the statistics are computed once the user loads the table or applies some filter to the table. For the computation of the statistics, the application does a walk through for the table and analyzes each event, and it makes the calculus needed to compute all the Statistics tab. Then, to make the code more usable and easy to understand, as it can be seen in figure 14, we separated the computation of the statistics and the html part in different .vue items, so one single file is not overloaded with all the code.

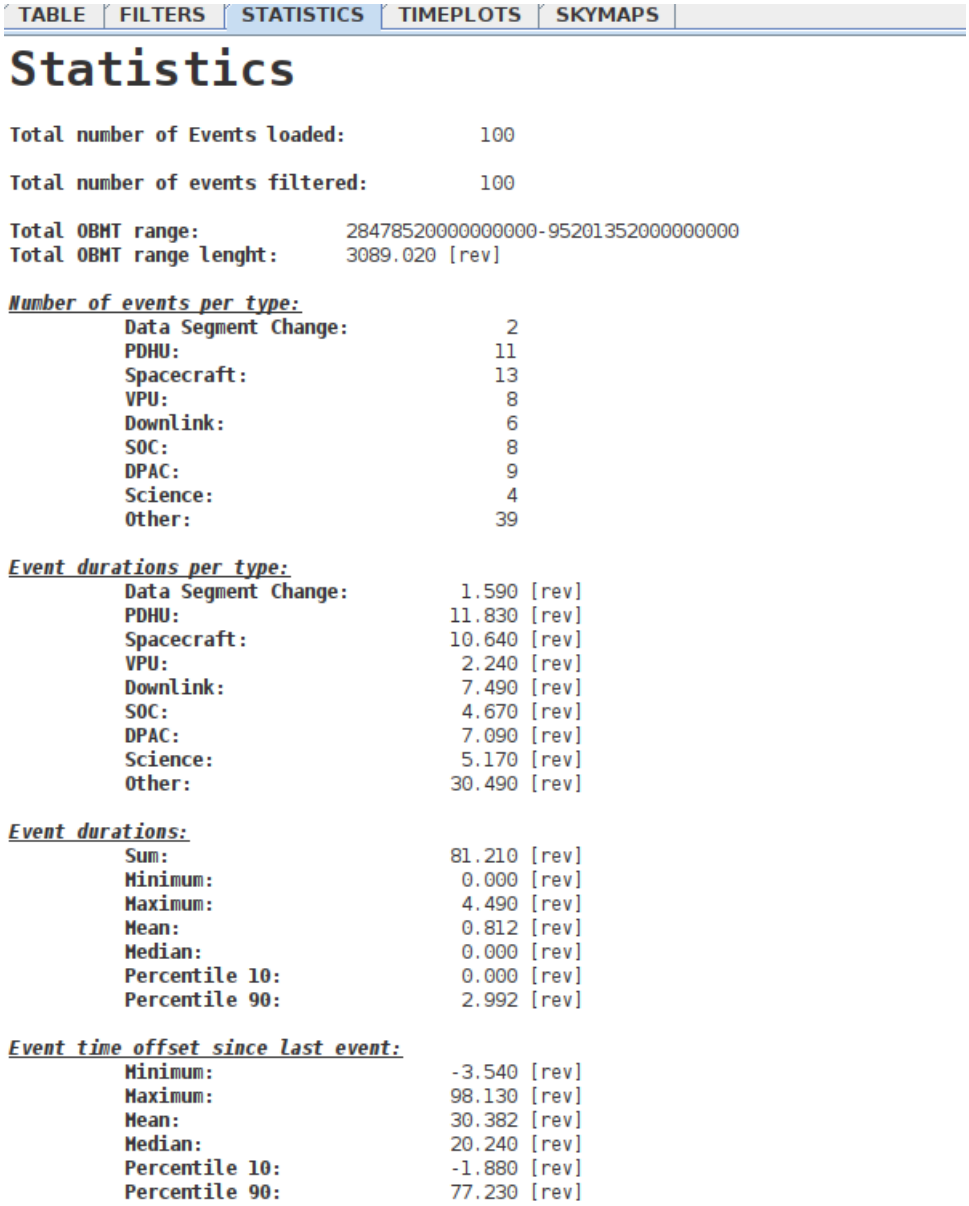


Figure 16: Statistics tab of the EventExplorer Java application

7.5 TimePlots

First, the timePlots tabs was migrated using the ChartJs module exlplained in the early sections. That implementation presented some issues, for example the change of plot was not automatical once the user used the select menu, and made it not practical to change the plot. Then, to preserve a coherency inside this project, we decided that it was better to change it, and made it again using D3Js this time.

In figures 17 and 18, we can see the two different types of plots of the Java desktop application. The first one is a scatter plot where it has the duration of the events as the y-axis and in the x-axis we can see the OBMT start of the events. While the other plot consists in a histogram of the events depending on their OBMT start too, in bins representing a week. The y-axis represents the total count of events and the x-axis the OBMT start. In addition, there is another histogram plot with bin size equal to one day (1 day = 4 revs.). In those plots, the user has the possibilty to zoom-in/out.

Then, if we pay atention to figures 24 and 25, we can see the same plots in the new web based application. The plots are mostly the same, however, the bin size in the histogram plot has been changed, now it is equal to 4 revs(a day). Then zoom has been upgraded, as in the original application, it made zoom in the two axis, making it not practical as it is not necessary to zoom-ou the y-axis, it only makes it more difficult to look at some data. The zoom in the new application takes the movement done by the wheel of the mouse to compute a new plot, in an efficient way, with a new range of values, so the y-axis is updated depending on the zoom made. In addition, we have implemented a tooltip feature that allows the user to see different information depending on the plot. In the scatter plot, we can see either the type and the name of the event. While on the histograms, the tooltip will show the range of the bin the mouse is hovering over. Lastly, as said, the bin size of the histograms has been changed and new ones added. In this application the bin sizes that can be found are: 1 rev(6h), 4 revs(1 day), 256 revs(64 days, a total scan of the galaxy), and 1460(1 year). It is planned for a future version to implement new sizes for the plots.

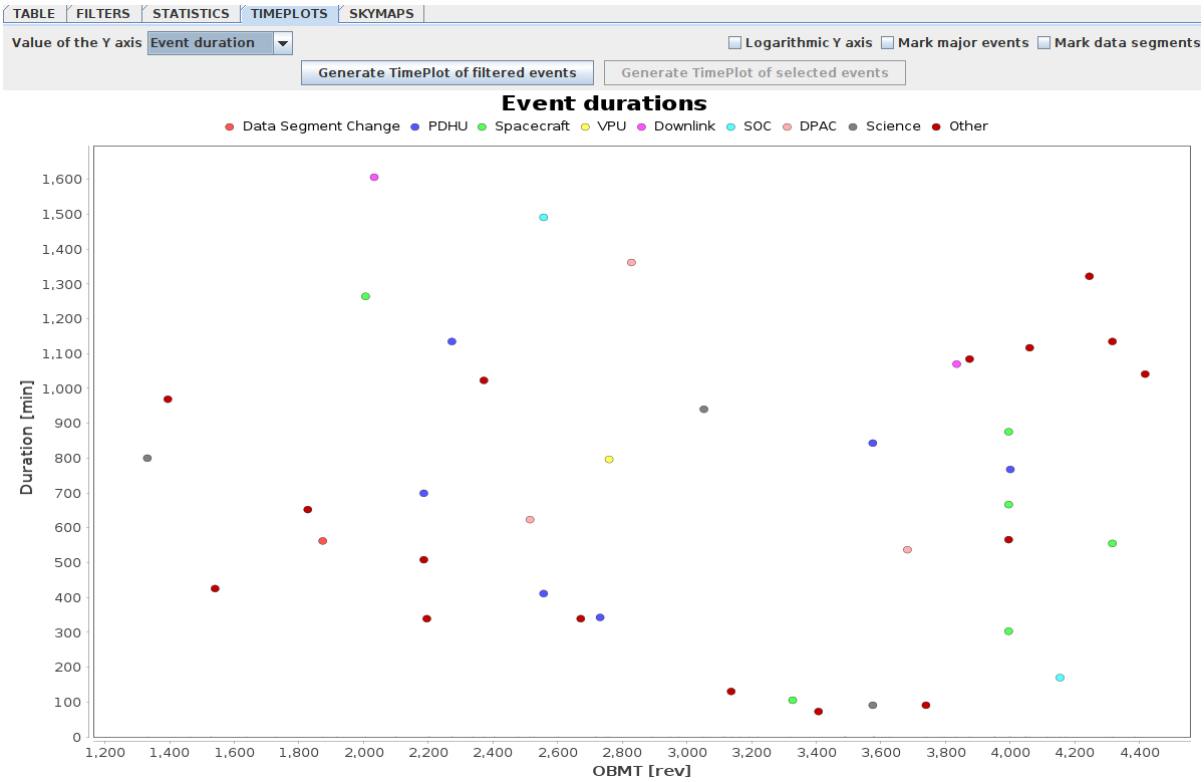


Figure 17: Scatter Plot of the EventExplorer Java application.

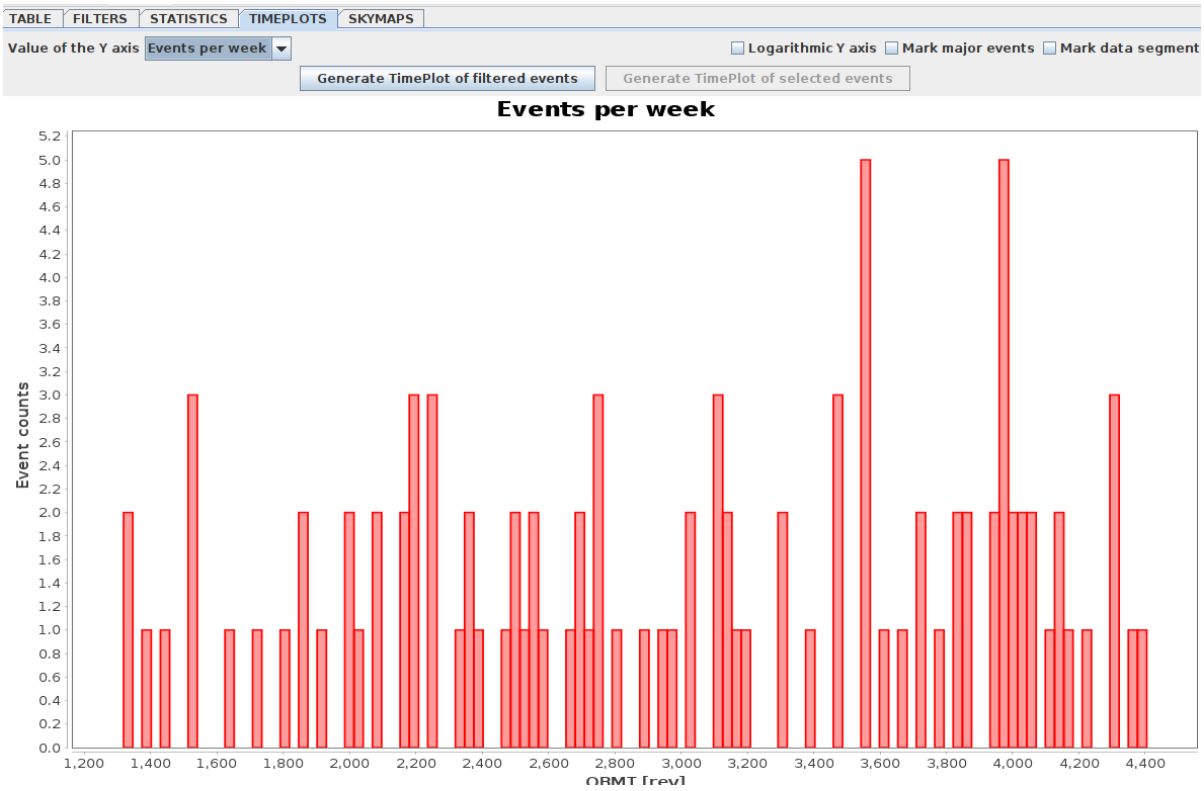


Figure 18: Histogram Plot (bin Size per week) of the EventExplorer Java application.

7.6 SkyMaps

The SkyMaps tab is one of the main reasons for the migration of the EventExplorer, and at the same time, is the tab that took more time. That is why, we will elaborate more on it, as well as the backend section which is pretty related to the SkyMaps.

In terms of design, it can be seen in figure 19 and 26, that some options have been conserved while others haven't. About these options, we can see that the only ones that have been preserved are the selection of field of view, the coordinates in which the SkyMap is represented and the color palette of the plot. All these selections work differently than the original ones, because in the desktop application, they are selected before the generation of the plot, and once the plot is generated, you have to regenerate the plot to see the changes on these settings. However, in this project, the change of an option changes the plot dynamically so the user can see the updated plot at the time he makes a change. Then, we removed the selection of the level of resolution of the HealPix, as we considered that using a low level of Healpix it is not enough to appreciate the events, and using a high level, it doesn't make such a difference for the cost it implies. So for these reasons, we will use a 6 level resolution for the plotting of the SkyMaps, as it gives an enough resolution to appreciate the details and it doesn't imply an extraordinary cost. Then we added a new checkbox, to decide if the graticule needs to be rotated as well as the points of the plot. To finish the settings menu, the web page contains different buttons for the generation of the SkyMap. There exist two different options. The first one consists of uploading a .csv file, later we will talk more deeply about it, and plot these events. Later, there is the main plotting option, which consists of plotting all the events of the table with the corresponding information.

After the settings of the SkyMap, there is the plot itself. We can observe that if the plot hasn't been drawn, there still is the graticule. Next to it, there is a color palette that synchronizes with the option mentioned before, so when the user changes the color scale, can see at the moment the new one, even if the plot hasn't been drawn. Then, this color scale, once the user generates a SkyMap, the color scale it is used as a legend for the conversion from number of events to the respective color. Later, in Figure 26, on the right side of the application we can see some information about the events that were clicked. If no event has been clicked or if the SkyMap hasn't been generated it doesn't show anything. There it shows the number of events, the coordinates and the list of events of that HealPix coordinate. In case the user uploads a file to draw, if no information of the event is available, instead of listing the events, it will be undefined.

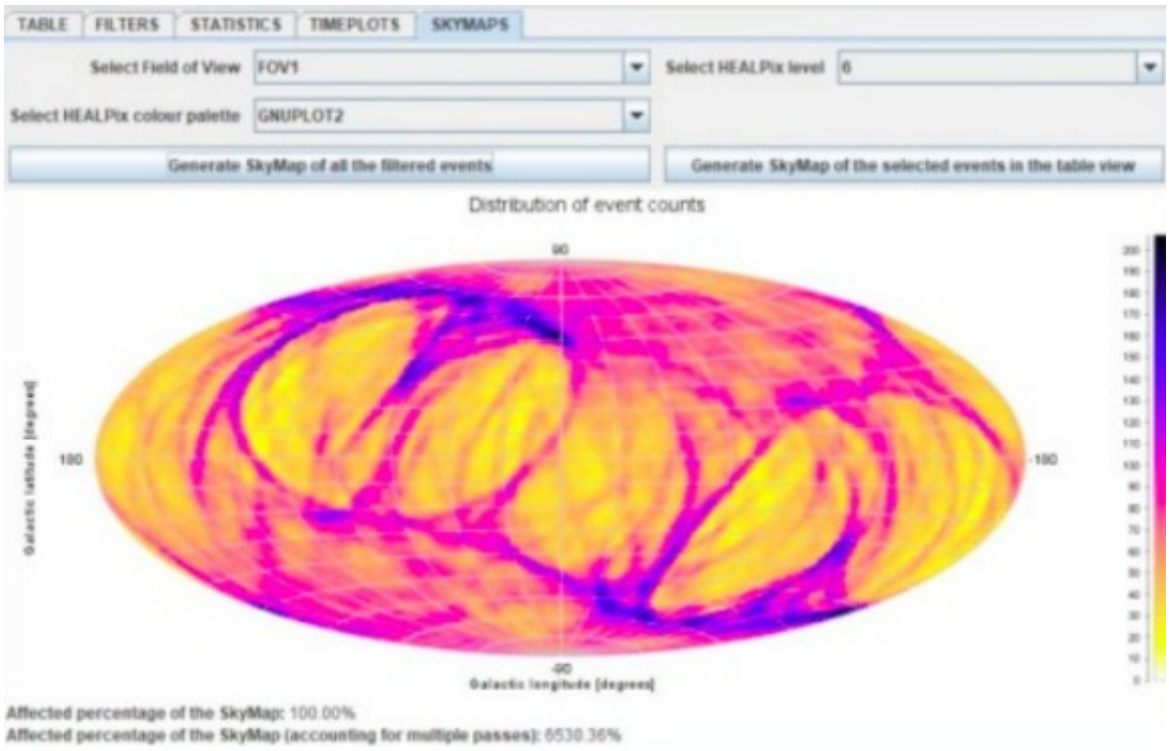


Figure 19: SkyMap tab of the EventExplorer Java application.

As for the implementation of the SkyMaps there are lots of things to say. First we'll talk about the implementation of the tab itself, so later explain how the SkyMaps are computed, either if they are generated from the table or from an uploaded file. With the D3Js library, once the tab is mounted, the application creates a SVG (Scalable

Vector Graphics) element, later it declares an Aitoff projection and the paths of the graticule, both of them obtained from the library. And later, knowing the paths, the application takes the coordinates of those paths and draws the corresponding lines inside the svg element applying the Aitoff projection. Also, it draws a fixed contour for the graticule, as the graticule can rotate, and sometimes the borders cannot be seen well for the angle of rotation applied. Later, it creates one hundred rectangles whose colors, all together form a color palette, this is made to make the color scale.

Once the base page is done, the user has two options to plot the SkyMap. The first one, consists in the user uploading a file, and once it is done, the user clicks the button to add this data to the plot. To do it, the application creates a map object with the total count of events in each HealPix to later send it to the backend, so it can generate the respective GeoJSON[2] file for the plotting. Once the application receives the file, which contains all the information about the points that need to be drawn, we draw all the points with each respective information thanks to D3Js. As for the second case, that happens when the user decides to draw the events from the table, once the button is clicked, the application sends to the backend, the list of the index of the events and a map of the names for each event that needs to be plotted, so, similarly to the last case, the backend computes and sends the GeoJSON file needed for the D3Js.

Later, in both cases it is necessary to create a click effect for each point, so when a point is clicked it shows the coordinates, total count of events and the list of events in the right space of the plot. After that, the application takes into account the coordinate system applied, and updates the plot to the coordinates setted. Finally, for the rotation of the SkyMap, once the space bar is clicked, the application takes the mouse movement and applies a rotation that depends on the distance done by the mouse. If the rotation to graticule checkbox is true the graticule will rotate too, however if it is set to false it will come back to its original position.

2GeoJSON: File type that is derived from a basic JSON file that defines points, lines and polygons. It can be added some properties too.

7.7 Backend

Last but not least, there is the Backend part of the application. This part is made using Javascript, with the express js module to allow the communication between the frontend and the backend, in which the frontend axios is used. Later, as for the database, MySQL is used.

About the database, as it can be seen in figure 20, it is formed by 4 main tables. The first one, is a table that contains all the events and its information. That table is used for the application to load all the events directly, without the user having to upload a file. Then, the 2 tables represent the healpixes corresponding to each event per each field of view. They contain 2 columns, one for each event and the other for the list of healpixes. Finally the last table contains the coordinates of each healpix, in alpha and delta, so we can use them at the time the plot has to be computed.

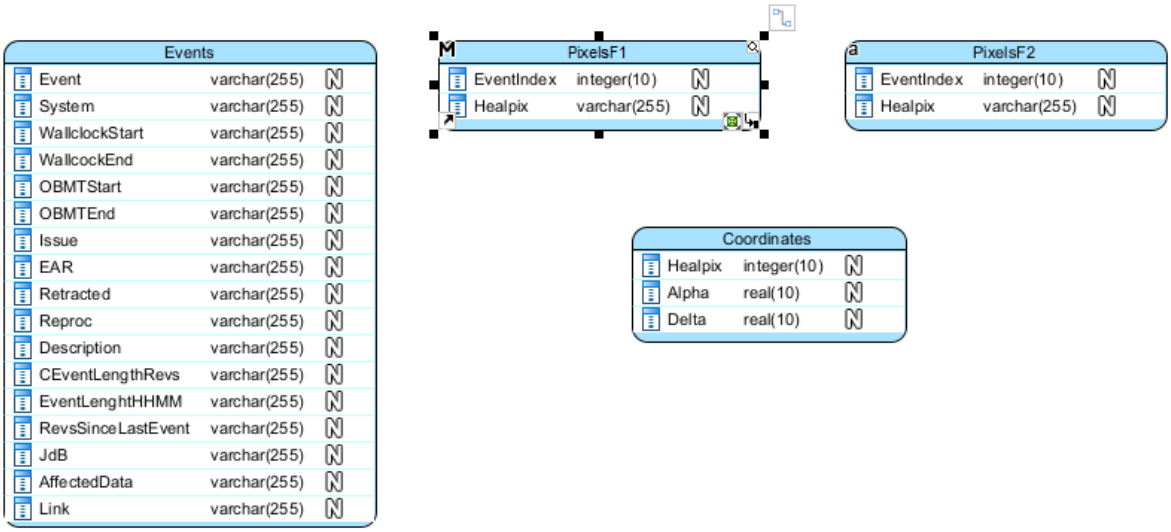


Figure 20: Database tables

Once the database is explained, let's proceed on the functioning of the backend. First of all, once it is executed, the main program will access the database to load the four tables to the program, so when the user needs to use some of them, the access will be immediate. All these tables will be stored in different objects such as maps or matrixes.

After the backend finishes the access to the database, the main program will stay awake so it can process any request the frontend sends. Depending on the request it receives it will do different things. Both of them consist of the creation of a JSON file to send it back to the front end, so the D3Js will use it to make the SkyMap plot. In the case the request is to draw the events from the table, the application will update a map that contains the count of events in each healpix, so the new draw can be accumulated to the last one, if there was some already plotted. To do it, it will go for all the indexes of events the frontend sent, and update the map for each healpix the event occurred in. Later, it will sort the map of healpix, so when the plot is made, it will paint it in ascendent order of count of events. Finally, it will generate a JSON file and send it to the frontend so it can generate the plot. As for when the user wants to add some events from a file, the program will receive a map, which will be used to update the map object that contains the current count of events, and later, as in the last case, it will generate the JSON file.

8. Assesment

The main objective of the project was to migrate the EventExplorer Java desktop application to a web type based one. So the Gaia members, and in the future all the Gaia Consortium, could have and upgraded application that has more features than the original one, and it is more accessible to all the users.

During all the project, as it can be seen in Apendix A, the application has been developed correctly, adding most of the functionalities that were planned at the start of this project, and later, tested them. Mostly all the features implemented work correctly and eficiently, however during the testing phase, with the private events of the Gaia mission, that are already not published, when the list of them is too high, some functionalities can go slower than it was planned. These features that doesn't work as well as expected are the rotation of the SkyMap, as it contains thounsands of points that implies lots of computational resources, and the listing of all the events, for a similar reason.

Finally, this project only has been tested in a local environment, either in the offices of the ICC-UB or in personal computers, and we have got a good perfomance on the result of it. However, it is planned to deploy this application to all the Gaia members so they can make use of it during their researches. And later on, to deploy the Web based EventExplorer to all the international science community, as it was made with the Gaia data releases.

9. Conclusions

During all this project we have seen all the steps we have done in order to design and implement the EventExplorer web application. This project consisted of the migration of the EventExplorer desktop application to a web application implementing some upgrades to certain features, and at the same time, maintaining the concept and uses of the application.

All the features of the desktop application have been migrated correctly without losing any functionality. Instead, all of them have gotten improvements. The filtering of the events have been totally changed, allowing lots of more options at the time to filter the table, such as the duration filter and the date filter. With all these filters, the table tab feels renewed as mostly the filters affect this tab. For the statistics part, as the design has been completely changed, and thanks to the data segments table it helps the user to have a better understanding of the events and its information. Then, there is the addition of new time plots and the new features they have received, as now it is possible to zoom in and out, and the panning of them.

Finally, the upgrade and migration of the SkyMap tab, which was one of the main purposes at the time of doing this project. The main objective of migrating it was successfully done, as the implementation of the SkyMap was done and the basic functionalities were conserved. Later, it was possible to implement most of the new desired features such as interactivity with the plot, like the rotation of it, and the information shown when the events are clicked. Also, in this new web application, the changes made to the plot, such as the desired fields of view or the color palette are applied dynamically, so it is not necessary to wait for a new plot to be generated.

9.1 Future Work

Despite most of the planned work has been done, it still needs some time to be deployed to the Gaia research group adn to the whole Gaia Cosnortium. In this section we will talk about the features that still need to be implemented, or some changes that will be done in the near future to the EventExplorer web application.

First, as in one of the reunions before the submission of this project was talked about, it is planned to change the basic style of this application to a one agreed by the members of Gaia and IEEC teams, that will be used in more applications apart from the EventExplorer.

Another change planned regards the database. As the MySQL database needs a license in order to use it and deploy the application to the Gaia team, it has been decided that SQLite will be used instead, so this change implies changing part of the current backend part.

Then, talking about the SkyMaps, more upgrades and new features are designed to be done in a future version. First one of them, and the most important one, will be the addition of a zoom feature, as it was planned for this project. It couldn't be implemented, as other features took more time than expected. After the zoom implementation, another feature that would be interesting to implement would be the possibility to add a base image in the plot, like one of the milky way galaxy to know more precisely in which areas of the galaxy each event is occurring.

Apart from all the changes and improvements said in this project, once the application is deployed and used by the scientific community, more upgrades and requests will be requested.

List of Acronims

DPAC: Data Processing and Analysis Consortium

ESA: European Space Agency

HealPix: Hierarchical Equal Area isoLatitude Pixelisation

ICC-UB: Institut de Ciencies del Cosmos – Universitat de Barcelona

PDHU: Payload Data Handling Unit

SOC: Science Operation Centre

SVG: Scalable Vector GRaphics

VPU: Video Processing Unit

References

[1] European Space Agency [Online; Retrieved on October 5, 2022]. URL: https://www.esa.int/Space_in_Member_States/Spain

[2] Gaia mission, ESA. [Online; Retrieved on October 5, 2022]. URL: https://sci.esa.int/web/gaia

[3] Angular. [Online; Retrieved on October 7, 2022]. URL: https://angular.io/

[4] React. [Online; Retrieved on October 7, 2022]. URL: https://reactjs.org/

[5] Vue.Js. [Online; Retrieved on October 7, 2022]. URL: https://vuejs.org/

[6] NodeJs. [Online; Retrieved on Oictober 7, 2022]. URL: https://nodejs.org/en/

[7] Deno. [Online; Retrieved on October 7, 2022]. URL:https://deno.land/

[8] ChartJs. [Online; Retrieved on October 15, 2022]. URL: https://www.chartjs.org/

[9] D3Js. [Online; Retrieved on November 20, 2022]. URL: https://d3js.org/

[10] HealPix. [Online; Retrieved on November 23, 2022]. URL: https://healpix.sourceforge.io/

Apendix A

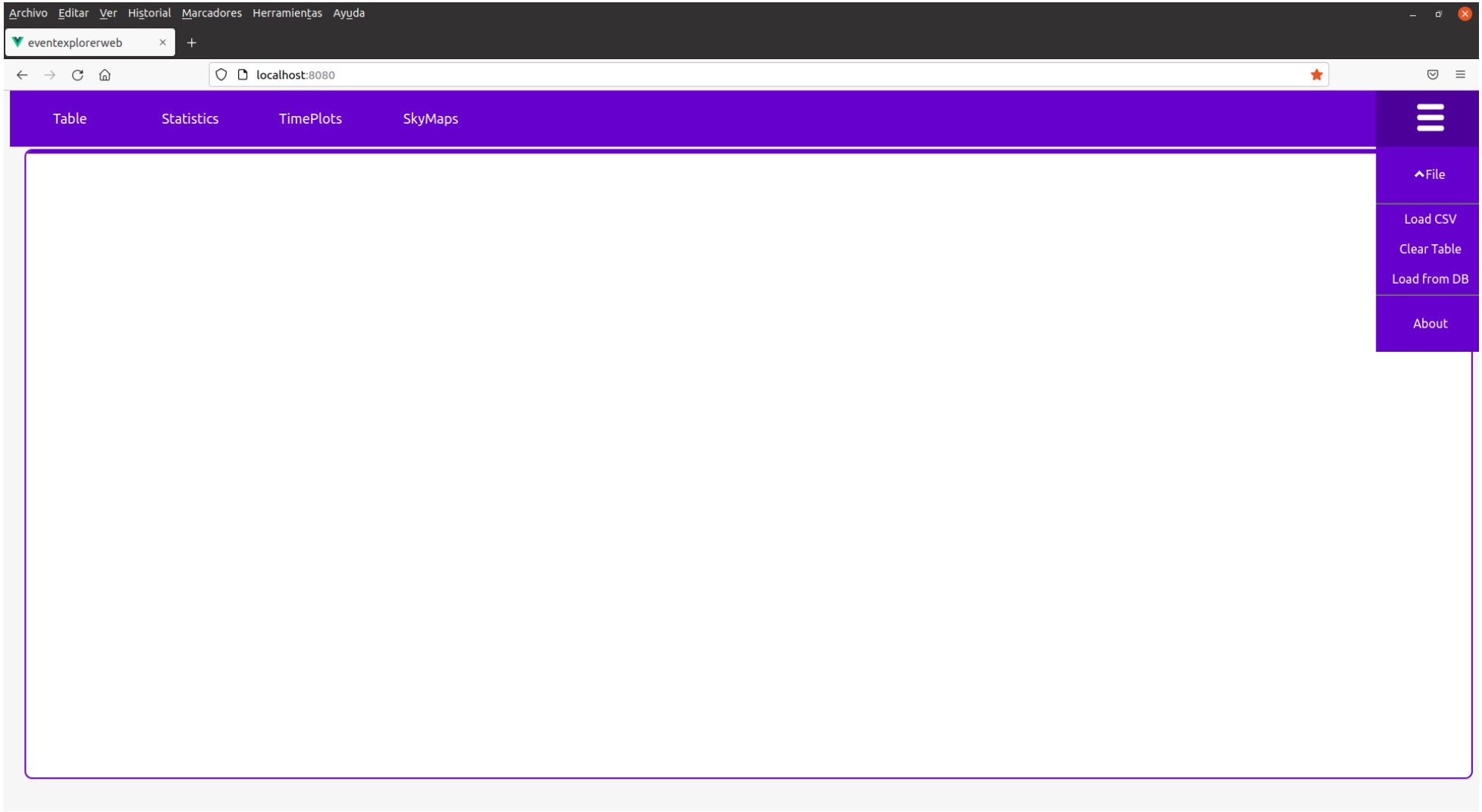
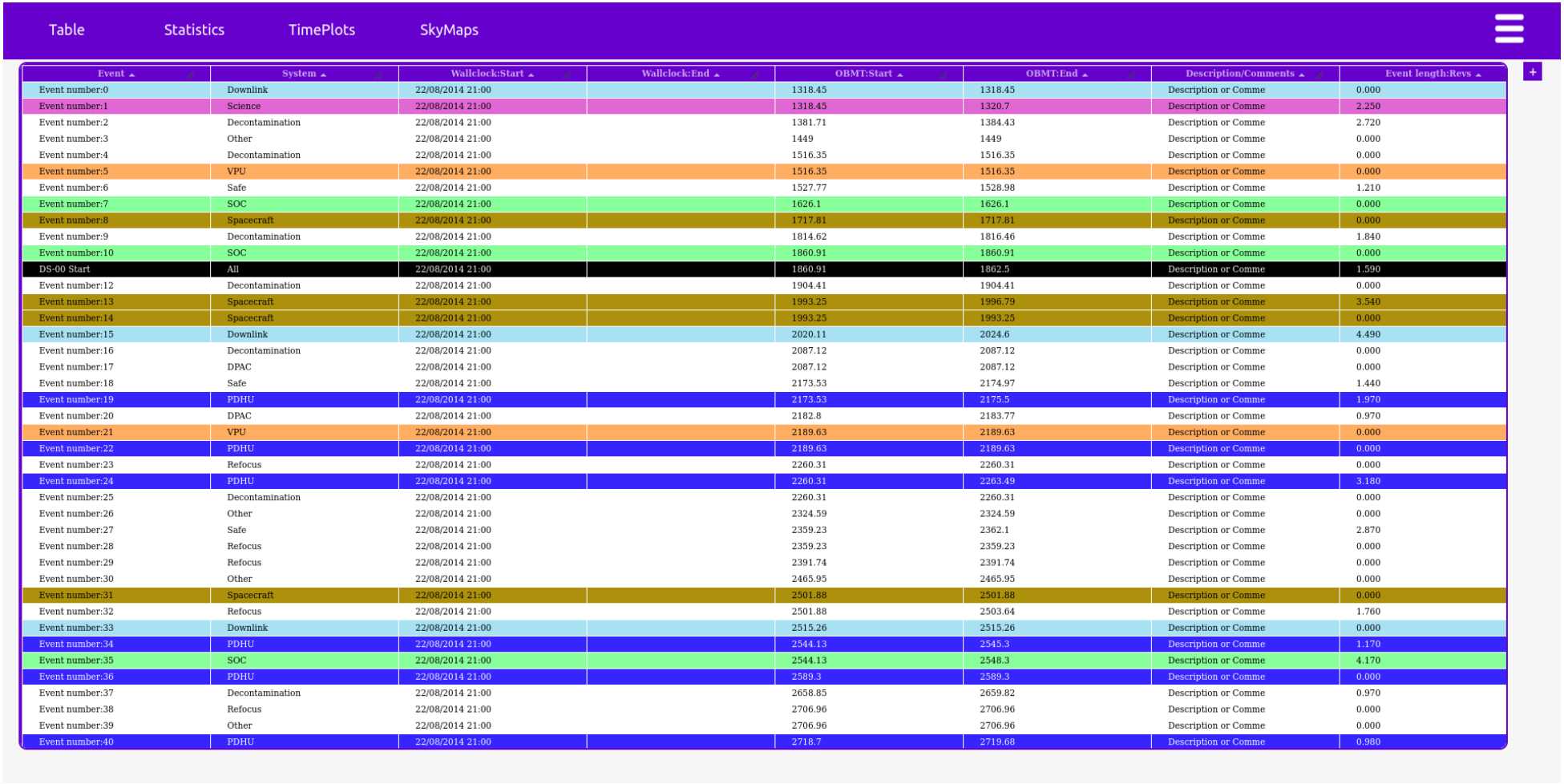


Figure 20: Menu of the EventExplorer Web application



Event	System	Wallclock:Start	Wallclock:End	OBMT:Start	OBMT:End	Description/Comments	Event length:Revs
Event number:0	Downlink	22/08/2014 21:00		1318.45	1318.45	Description or Comme	0.000
Event number:1	Science	22/08/2014 21:00		1318.45	1320.7	Description or Comme	2.250
Event number:2	Decontamination	22/08/2014 21:00		1381.71	1384.43	Description or Comme	2.720
Event number:3	Other	22/08/2014 21:00		1449	1449	Description or Comme	0.000
Event number:4	Decontamination	22/08/2014 21:00		1516.35	1516.35	Description or Comme	0.000
Event number:5	VPU	22/08/2014 21:00		1516.35	1516.35	Description or Comme	0.000
Event number:6	Safe	22/08/2014 21:00		1527.77	1528.98	Description or Comme	1.210
Event number:7	SOC	22/08/2014 21:00		1626.1	1626.1	Description or Comme	0.000
Event number:8	Spacecraft	22/08/2014 21:00		1717.81	1717.81	Description or Comme	0.000
Event number:9	Decontamination	22/08/2014 21:00		1814.62	1816.46	Description or Comme	1.840
Event number:10	SOC	22/08/2014 21:00		1860.91	1860.91	Description or Comme	0.000
DS-00 Start	All	22/08/2014 21:00		1860.91	1862.5	Description or Comme	1.590
Event number:12	Decontamination	22/08/2014 21:00		1904.41	1904.41	Description or Comme	0.000
Event number:13	Spacecraft	22/08/2014 21:00		1993.25	1996.79	Description or Comme	3.540
Event number:14	Spacecraft	22/08/2014 21:00		1993.25	1993.25	Description or Comme	0.000
Event number:15	Downlink	22/08/2014 21:00		2020.11	2024.6	Description or Comme	4.490
Event number:16	Decontamination	22/08/2014 21:00		2087.12	2087.12	Description or Comme	0.000
Event number:17	DPAC	22/08/2014 21:00		2087.12	2087.12	Description or Comme	0.000
Event number:18	Safe	22/08/2014 21:00		2173.53	2174.97	Description or Comme	1.440
Event number:19	PDHU	22/08/2014 21:00		2173.53	2175.5	Description or Comme	1.970
Event number:20	DPAC	22/08/2014 21:00		2182.8	2183.77	Description or Comme	0.970
Event number:21	VPU	22/08/2014 21:00		2189.63	2189.63	Description or Comme	0.000
Event number:22	PDHU	22/08/2014 21:00		2189.63	2189.63	Description or Comme	0.000
Event number:23	Refocus	22/08/2014 21:00		2260.31	2260.31	Description or Comme	0.000
Event number:24	PDHU	22/08/2014 21:00		2260.31	2263.49	Description or Comme	3.180
Event number:25	Decontamination	22/08/2014 21:00		2260.31	2260.31	Description or Comme	0.000
Event number:26	Other	22/08/2014 21:00		2324.59	2324.59	Description or Comme	0.000
Event number:27	Safe	22/08/2014 21:00		2359.23	2362.1	Description or Comme	2.870
Event number:28	Refocus	22/08/2014 21:00		2359.23	2359.23	Description or Comme	0.000
Event number:29	Refocus	22/08/2014 21:00		2391.74	2391.74	Description or Comme	0.000
Event number:30	Other	22/08/2014 21:00		2465.95	2465.95	Description or Comme	0.000
Event number:31	Spacecraft	22/08/2014 21:00		2501.88	2501.88	Description or Comme	0.000
Event number:32	Refocus	22/08/2014 21:00		2501.88	2503.64	Description or Comme	1.760
Event number:33	Downlink	22/08/2014 21:00		2515.26	2515.26	Description or Comme	0.000
Event number:34	PDHU	22/08/2014 21:00		2544.13	2545.3	Description or Comme	1.170
Event number:35	SOC	22/08/2014 21:00		2544.13	2548.3	Description or Comme	4.170
Event number:36	PDHU	22/08/2014 21:00		2589.3	2589.3	Description or Comme	0.000
Event number:37	Decontamination	22/08/2014 21:00		2658.85	2659.82	Description or Comme	0.970
Event number:38	Refocus	22/08/2014 21:00		2706.96	2706.96	Description or Comme	0.000
Event number:39	Other	22/08/2014 21:00		2706.96	2706.96	Description or Comme	0.000
Event number:40	PDHU	22/08/2014 21:00		2718.7	2719.68	Description or Comme	0.980

Figure 21: Table tab of the EventExplorer Web application

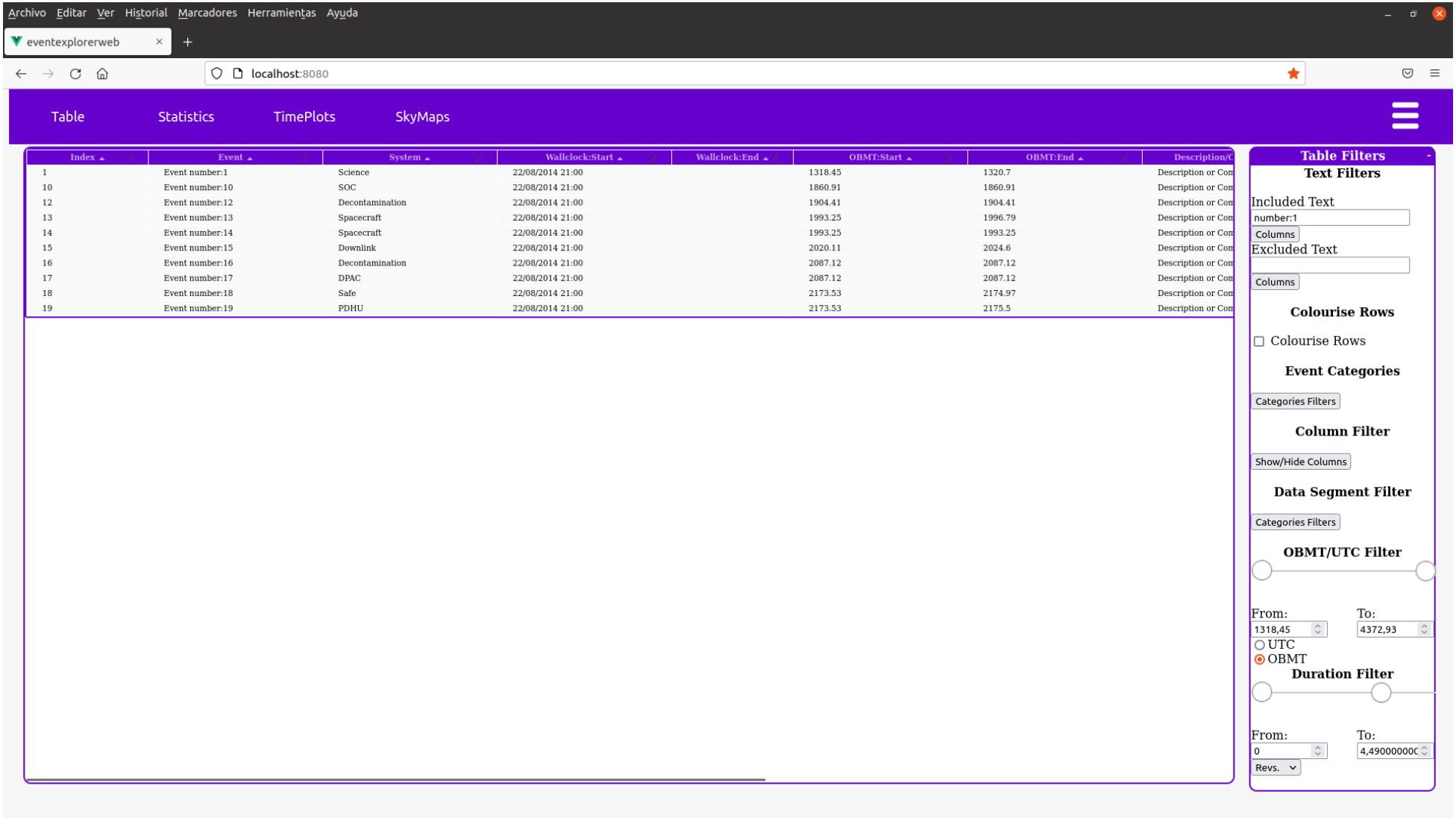


Figure 22: Filters Menu of the EventExplorer Web application

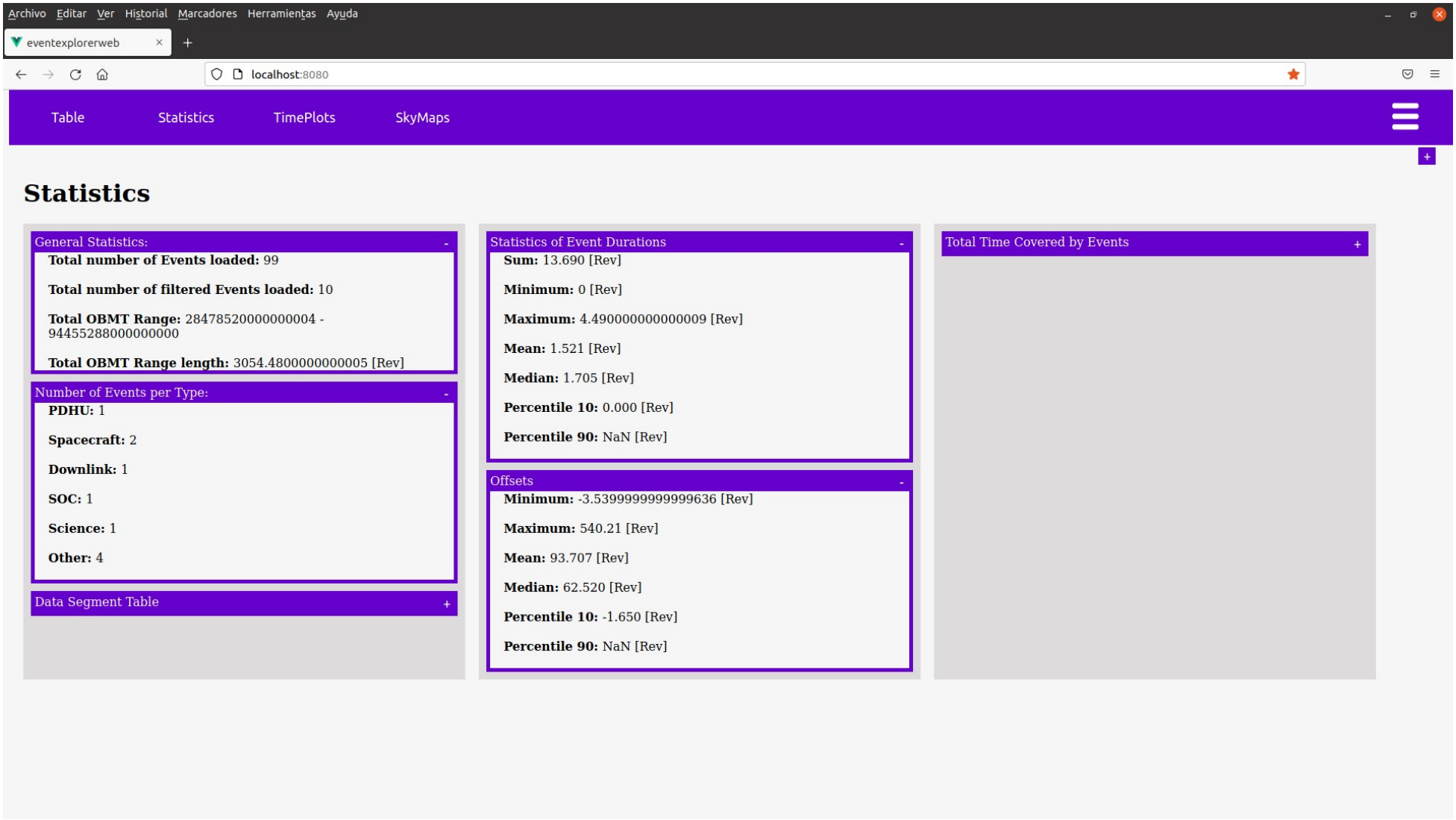


Figure 23: Statistics tab of the EventExplorer Web application

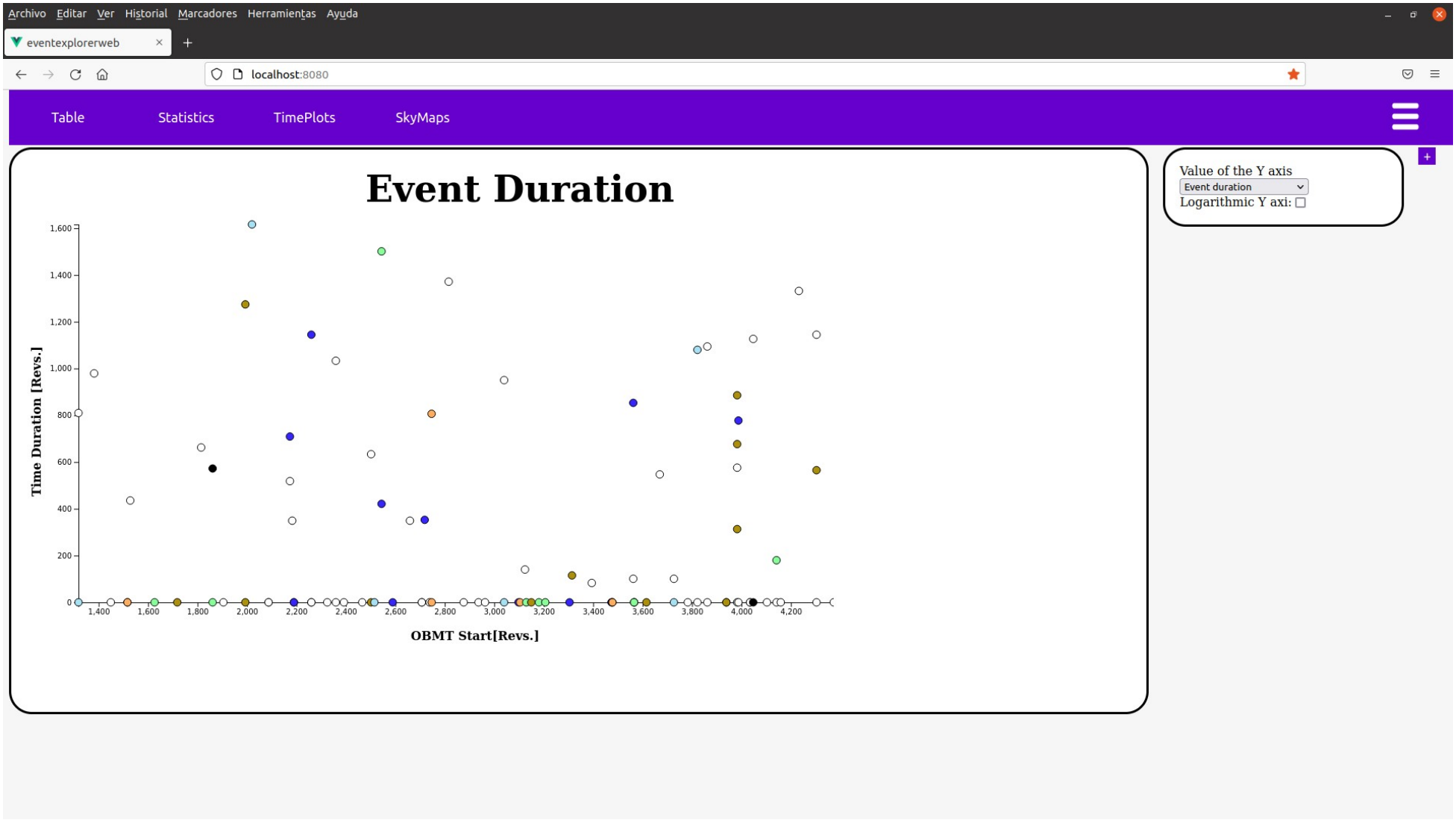


Figure 24: Scatter plot of the EventExplorer Web application

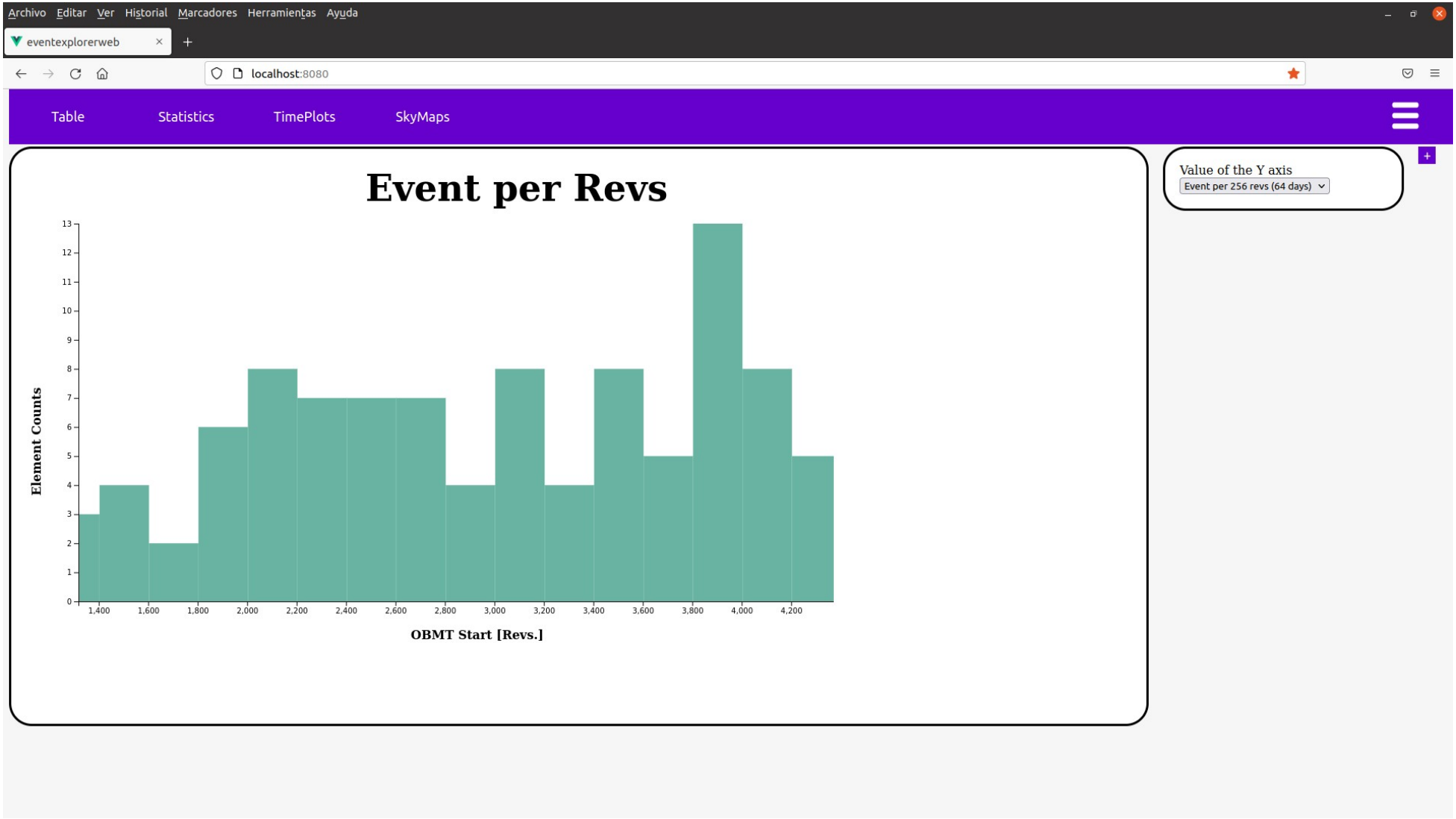


Figure 25: Histogram plot of the EventExplorer Web application

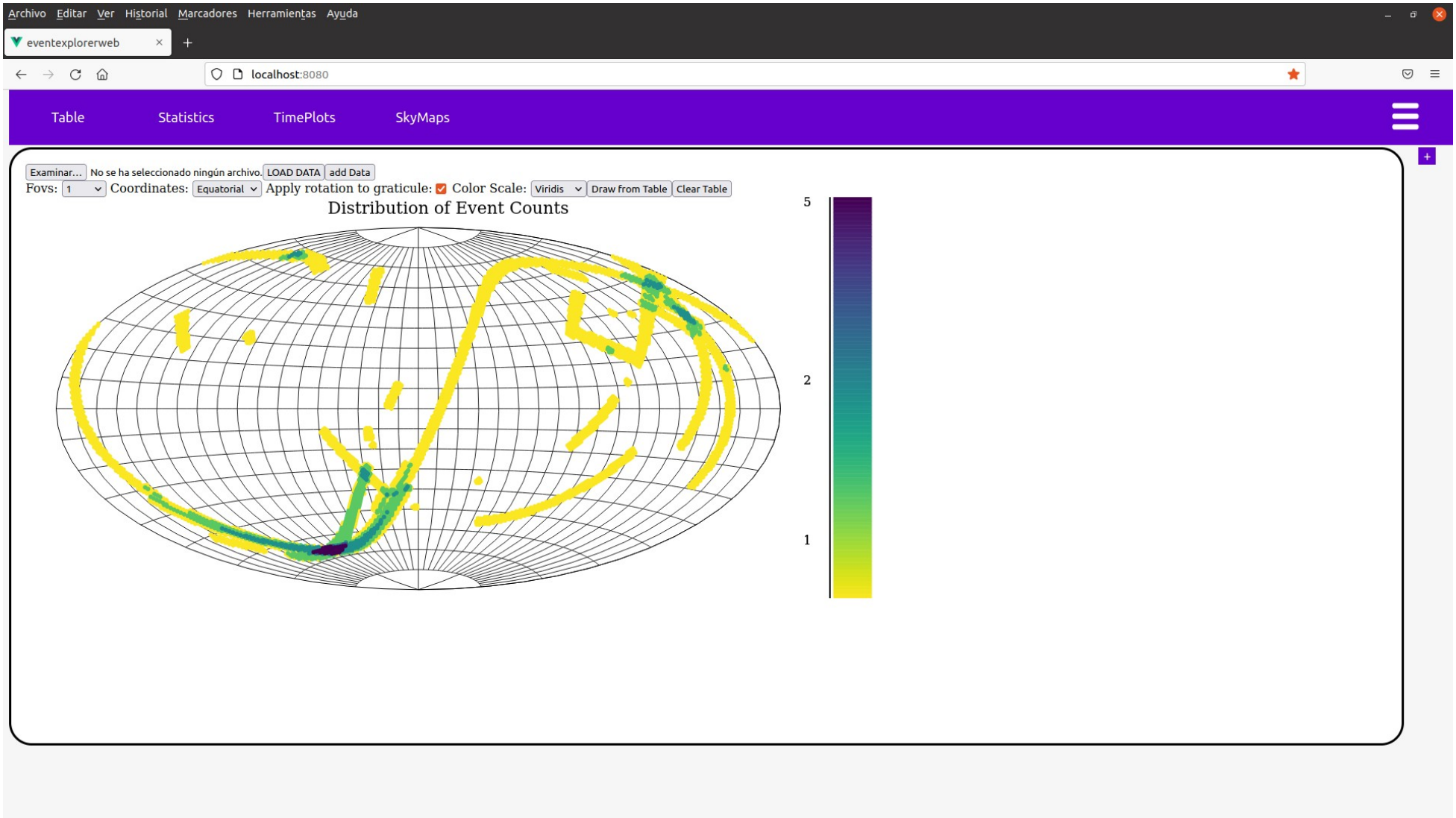


Figure 26: SkyMaps tab of the EventExplorer Web application