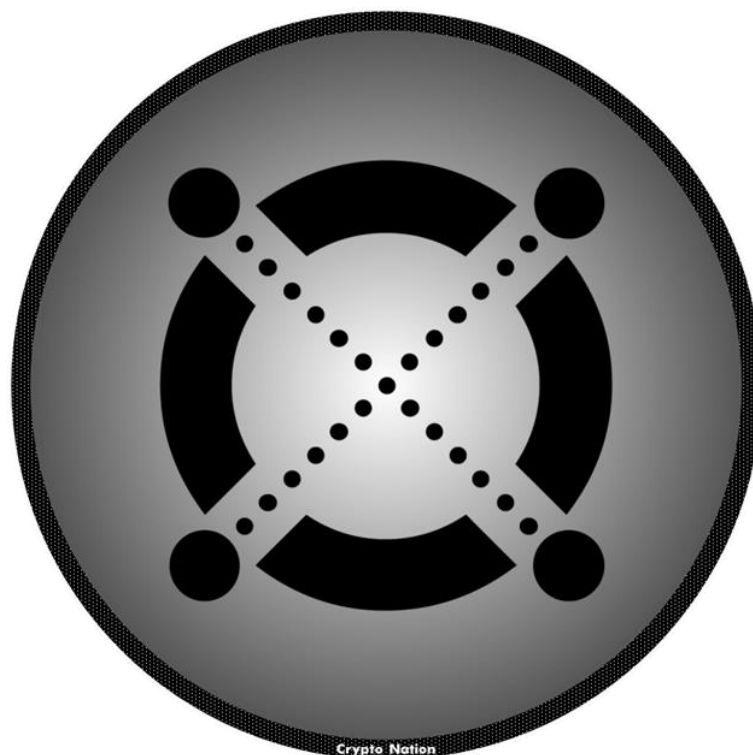


Zpectrum



Francisco González Herrera

Cristian Fernández Mirón

Tutor: Raquel Cerdá Losa

FP: DAM

Instituto: EDIX

Index

Introduction	Page 3
Keywords	Page 4-5
Moduls Used	Page 6
Tools/Languages	Page 7
Contribution	Page 8
Project phases	Pages 9-23
Data model	Page 9
Classes diagram	Page 9
E/R model	Page 10
Use case diagram	Page 10
Gantt diagram	Page 11
Interfaces design	Pages 12-15
Project functionality	Page 16-25
Conclusion and future improvements.....	Page 26
Bibliography	Page 27
Appendix	Page 27

Introduction

Zpectrum consists of a desktop app which pick up data from MetaverseX blockchain and produces all the necessary info to be useful and make possible to manage your cryptos inside the blockchain, as well as the main blockchain token's market price, the EGLD, their tokens, projects inside of it, including their respective nft's.

With Zpectrum you'll be up to date about the price quote of the exchange's tokens, you'll know the wallet's tokens if there is any, their balances, nft's and some more functionalities.

The target of this app is to make possible to manage your cryptos in a simple way and make it easier.

The purpose of why the user has chosen this blockchain over others more famous like Ethereum or Bitcoin are:

- Low fees.
- PoS (Proof of Stake) -> the way it secures the net and makes the users earn some more EGLD token by holding and delegating them to the nodes with daily compensation.
- More environmentally friendly (consumes fewer resources). Faster transactions.
- Backed by international financial funds.
- Very active community.

Keywords

- **Token or crypto**: digital asset born from an existing blockchain.
- **Blockchain**: It works as an accounting register, with sheets of accounts (transactions) that are executed as a block, and cannot be modified in terms of order or content once a transaction is registered in the blockchain nodes. Users can say that it follows a data structure with cryptographic headers. It contains the following features:
 - **Traceable**: wallet transactions are tracked without knowing the identity of the person managing the wallet.
 - **Distributed**: it is supported by nodes in such a way that if one node goes down, the network operation is not disabled since it is routed in alternative paths.
 - **Immutable**: Once a transaction has been made, it cannot be modified.
 - **Secure**: the more computers that support the network, the more secure the network will be.
 - **Decentralized**: all nodes are at the same level.
- **Nodes**: computers that work P2P and includes the global and decentralized network, store the blocks and contribute their computational power.
- **Consensus**: Acceptance by the members (nodes) of the blockchain network that validated information is the same for all.
- **Smart contracts**: It is an informatic program where you can execute a contract between two or more parts decentralizing the trust of making it. There are rules and there are executed. It is important to highlight that is not controlled by any part, it is decentralize by a preprogrammed condition. Depending on the blockchain type any person could see the smart contract content.
- **Blockchain MetaverseX**: It is a project in which the first objective is to achieve a fast, scalable blockchain that can be used for finance, business use, gaming, predictive models, etc. It differs from other blockchains because of its reliability, low fees and the diversity of the projects working on it.
- **EGLD**: is the name of the main token of the MetaverseX blockchain.
- **EGLD Address (ERD)**: unique address of a wallet in the MetaverseX blockchain, a fundamental element to be able to manage tokens, nfts and to be able to participate in the different events that occur in the blockchain.

- **Non Fungible Token (NFT)**: is a non-fungible unit of value that is assigned to a business model, which may be associated with a token project.

The user can think of NFTs as a store of value, they can be bought and sold, and their market value fluctuates.

The user can think of an NFT as a work of art, not only on an artistic level, but also on a representative level of ownership, i.e., by owning an NFT you may also be the owner of a tangible asset.

Moduls

1. First year:
 - Databases
 - Markup Language
 - Computer systems
 - Development environments
 - Programming

2. Second year:
 - Service and process programming
 - Data access
 - English
 - Interface Development

Tools/Languages

- Maven.
- SpringBoot.
- JSON.
- Postman.
- Rest.
- Eclipse.
- Github.
- Microsoft Teams

Programming language

- Java.

Framework

- Spring

Data base

- MySql.

Most important libraries

- JavaSwing.
- Jackson
- Hibernate
- Jcodec
- MigLayout
- Tomcat
- Spring
- Jakarta

Api's

- JPA
- Elrond API

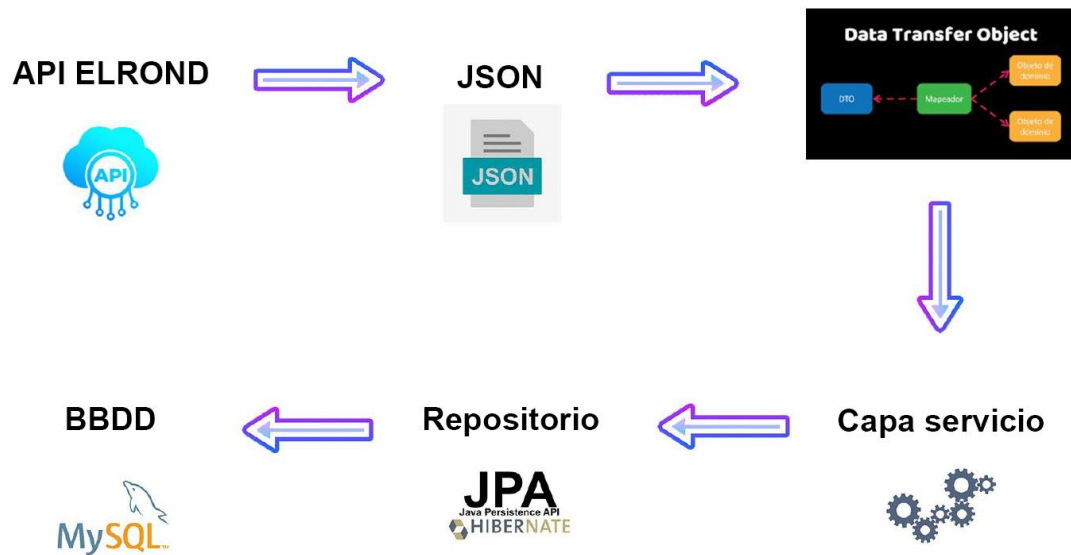
Contribution

As we set out in the preliminary project, each part of the project has been associated with each of the participants, although we have been meeting to solve problems in each of the developments:

- Development of the desktop application with Javaswing (UI), Francisco was in charge of the development. Once done, we were implementing improvements to the app, such as event handlers, new buttons, etc, in which both parts participated.
- Spring service development and HTTP calls, Francisco was in charge of development.
- Connection to MySQL 8 database, Cristian was in charge of this development.
- Review, improvements and testing, both parties were in charge, so that we could test the app and find bugs.
- Project report by Cristian.

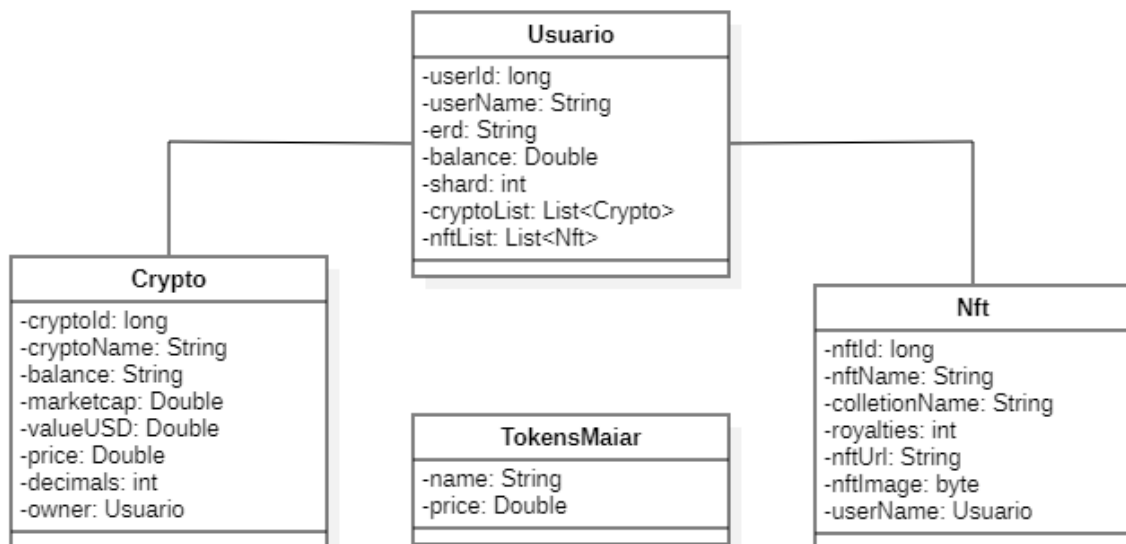
Project phases

Data model



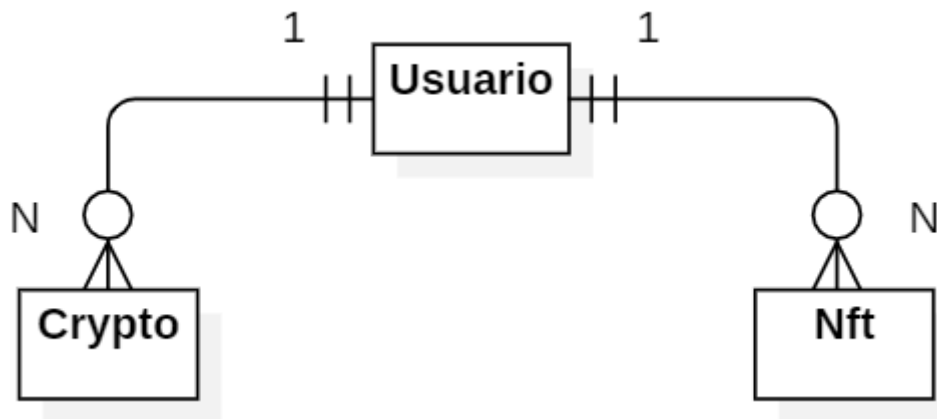
Data model

Classes diagram



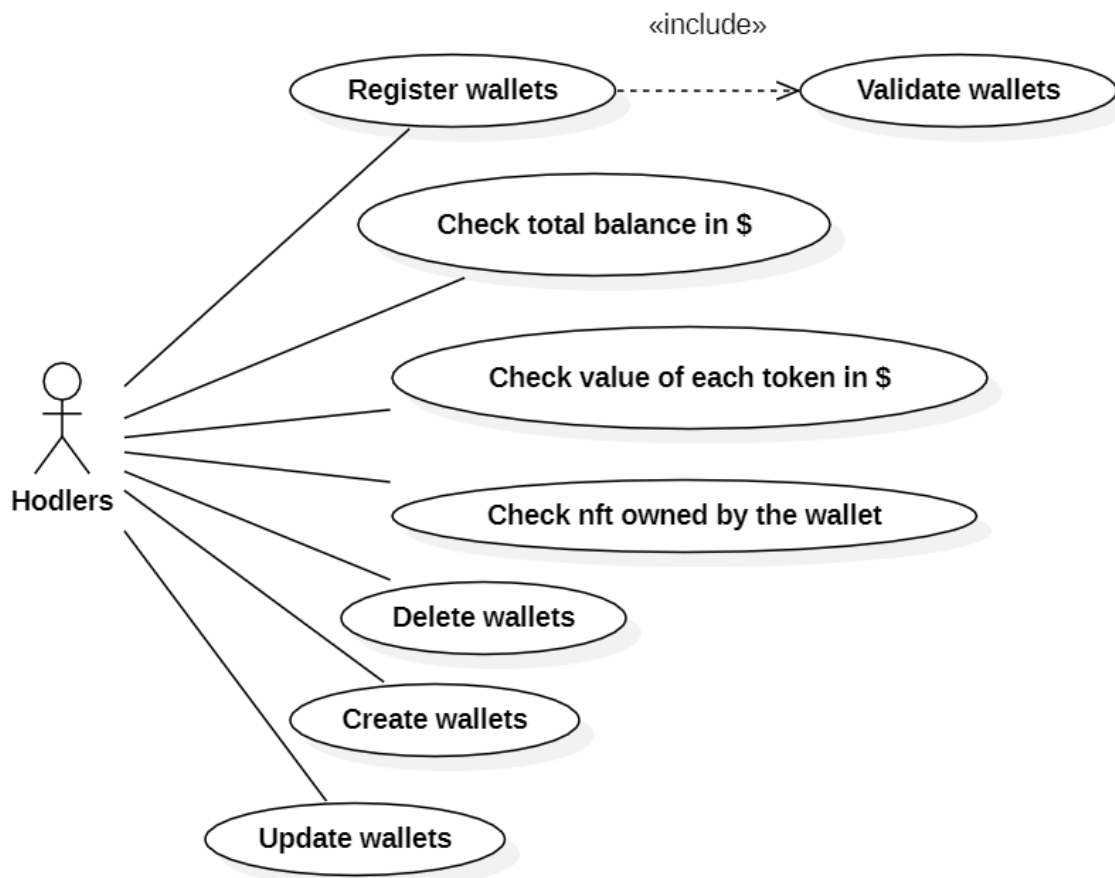
Classes diagram

E-R model



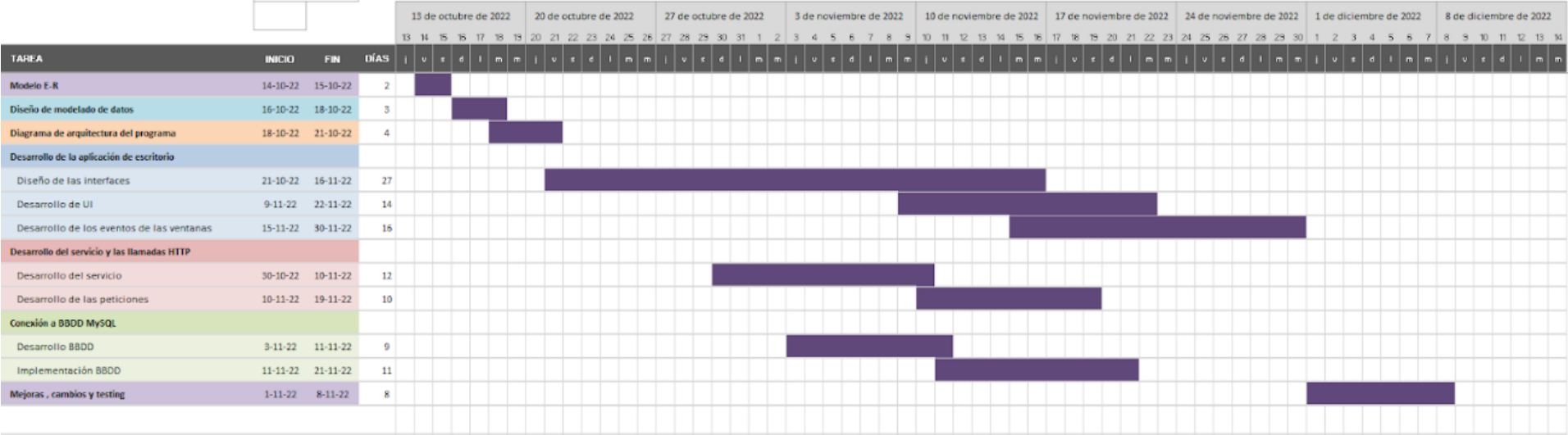
E-R model

Uses cases model



Uses cases model

Gantt diagram

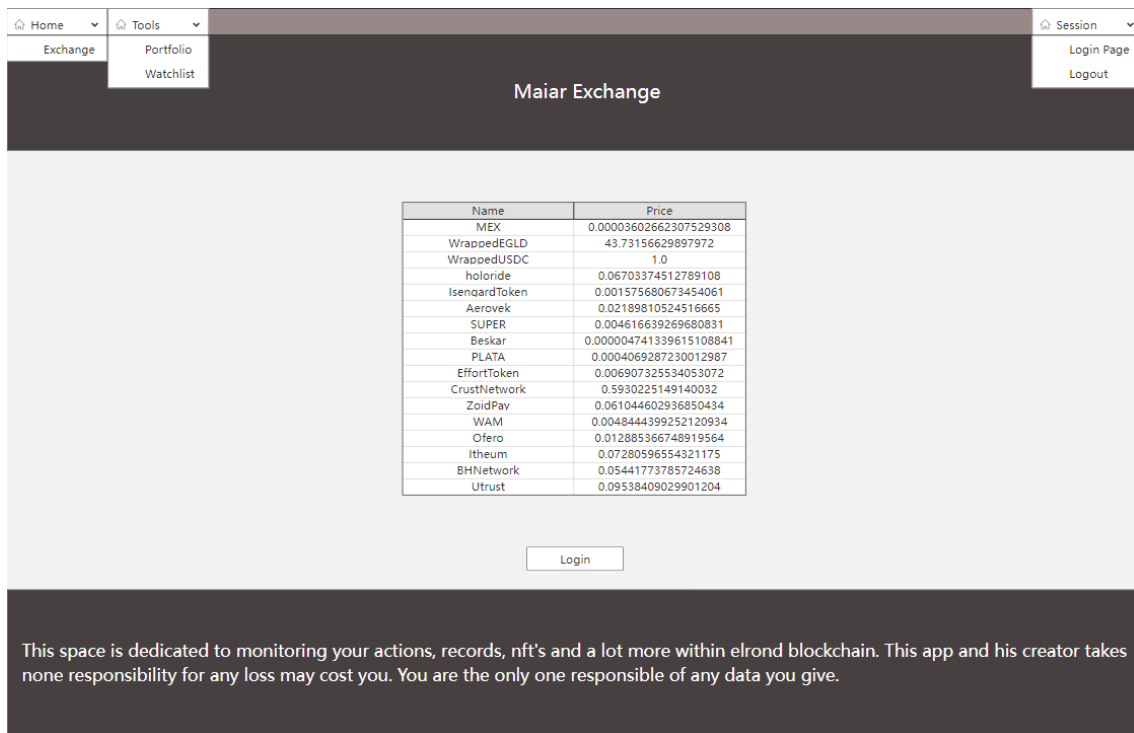


Gantt diagram

Interfaces design

When running the application, we would find the initial screen with the price of the updated tokens (internally performs an HTTP request to the MetaverseX API when loading the initial page), and with a "Login" button to access the user page.

The application is developed in English for a global purpose and can be used internationally:

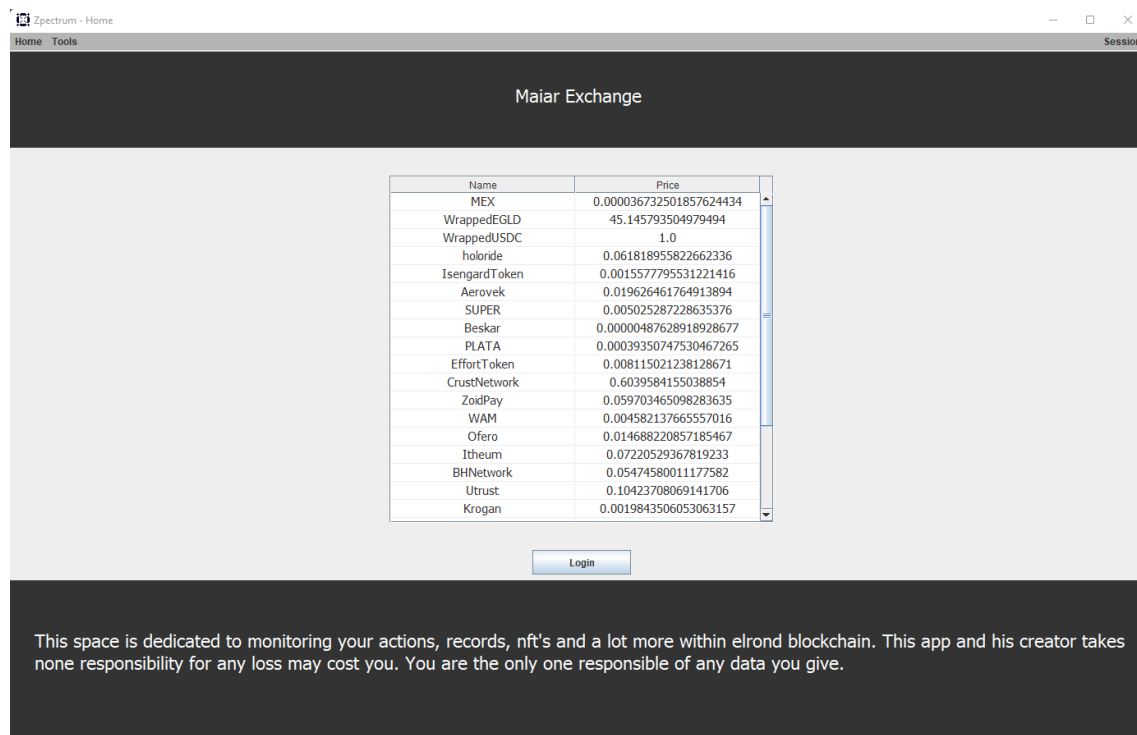


Design Maiar Exchange Page

It is important to keep in mind that Zpectrum app is using a CardLayout model, so as the user moves through the menus of the program, only the central space will vary (the program will display the "cards" or panels selected by the user).

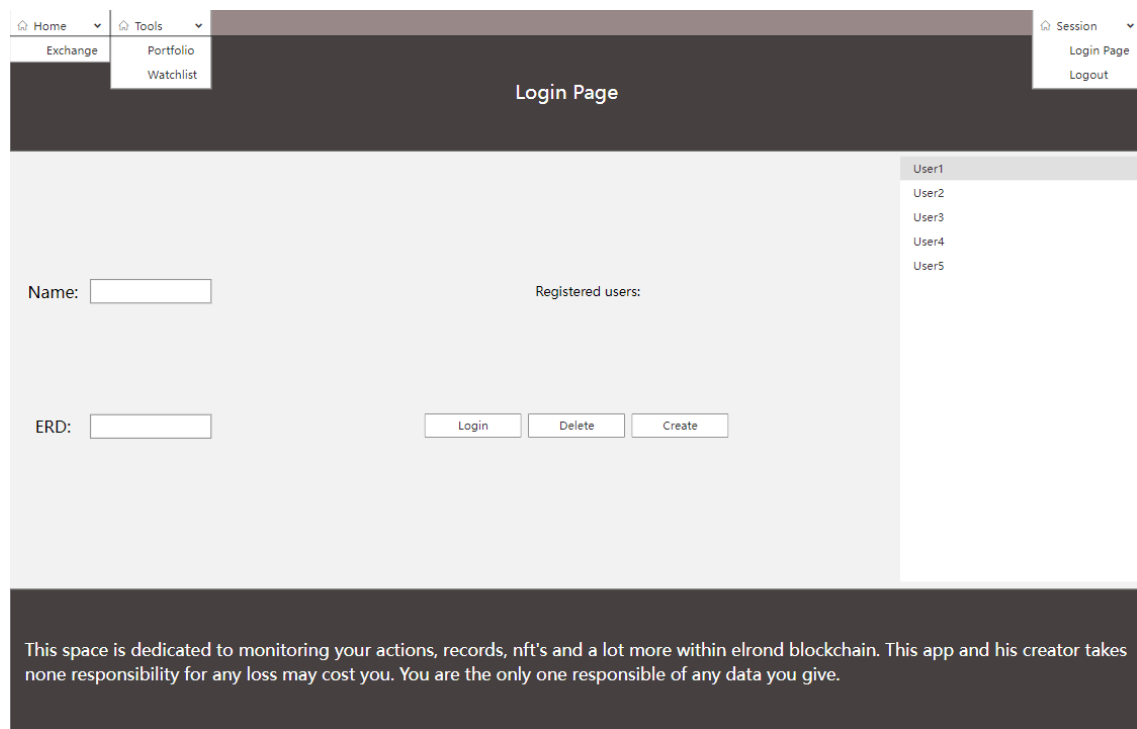
To disclaim any liability for misuse or loss of value either in dollars or euros the program will display a warning permanently on the responsible party for actions performed within the blockchain with ZpectrumApp.

The result of the application would be as follows:



Maia Exchange Page

In the user panel (or Login Page) the user's name and ERD to be registered in the application are entered:



Design Login Page

The result should be:

zpectrum - Home

Home Tools Session

Login Page

Name:

Registered users:

ERD: ⚠

Login Delete Create

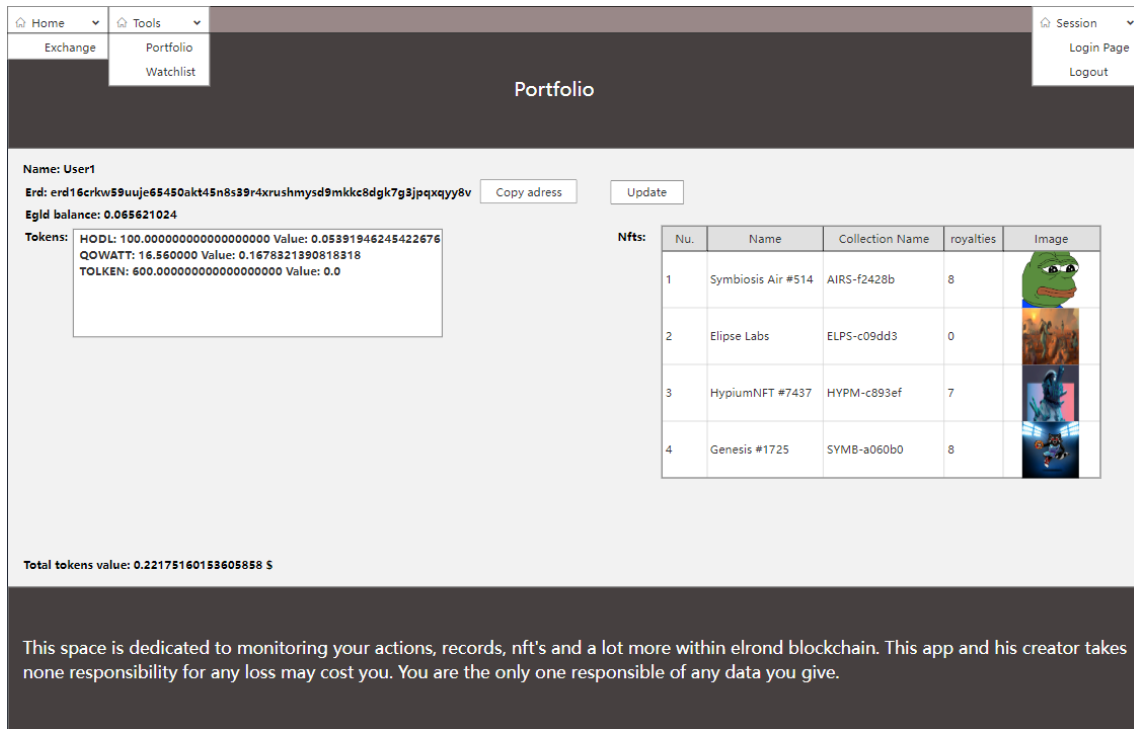
No hay usuarios registrados

This space is dedicated to monitoring your actions, records, nft's and a lot more within elrond blockchain. This app and his creator takes none responsibility for any loss may cost you. You are the only one responsible of any data you give.

Login Page

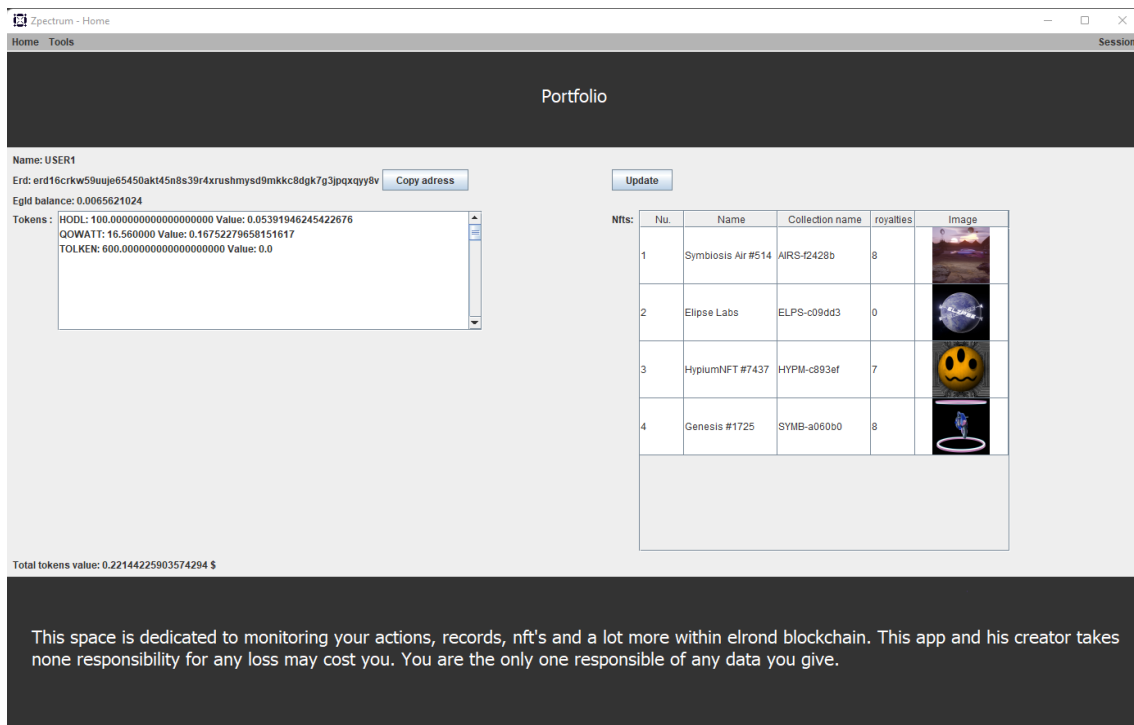
After entering a user, logging in and passing all the program filters, the user will click on Portfolio which will show us the account balance, the tokens in the wallet with their balances, dollar value in a list and the NFTs associated to the wallet in a table showing the number, the name of the NFT, the name of the collection to which it belongs, the royalties or commission for the creator of the NFT and the image of the NFT.

Below the list of tokens, we have the data of the dollar value of all the tokens accumulated in the wallet. These items will be detailed later:



Design Portfolio Page

The result should be:



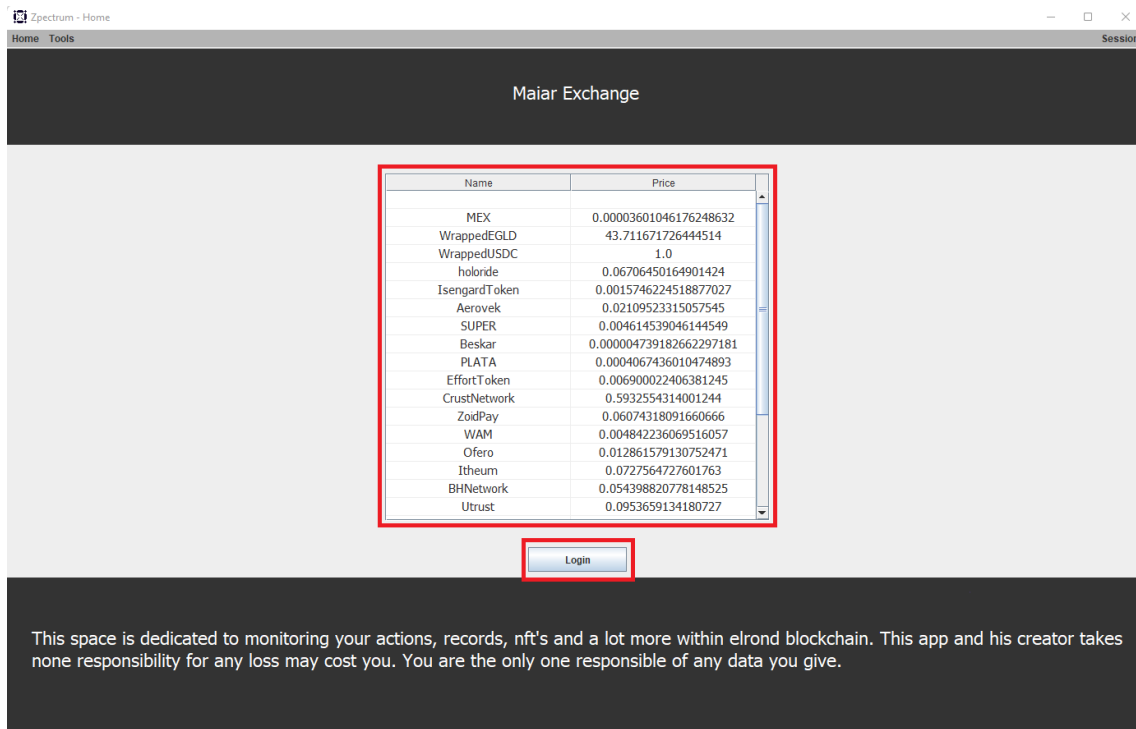
Portfolio Page

Project functionality

As we have commented before, the functionality of this project in general lines is to give the users of this blockchain the possibility to manage better their assets in a simple way. Because of this below, we will explain in a more technical way the full functionality:

Maia Exchange

When the app starts, it displays the MainWindow page, which contains all the information about the program elements. At the start of the app, it will show a card layout "MainWindowContainer" containing a table with the name and current price of the tokens in this cryptocurrency and a "Login" button.



Maia Exchange Page

The table is painted by a call to the Elrond API which returns a Json with a lot of information about the Exchange. Zpectrum subtracts and processes each token with its name and market price to add it to the table and display it in the main program window.

The "Login" button has the functionality to takes us to the Login page. It does this by displaying the "UserPanel" layer.

Login Page

On the next page the user can see, two text fields, to enter the user's name and the ERD of the Elrond wallet. Next to the ERD text field there is a yellow circle with a question mark. The user also has 3 buttons, one to login, one to delete the user from the list and one to create a new wallet. Finally, on the right the list of registered users.

Home Tools Session

Login Page

Name:

ERD: ⓘ

Registered users:

User1

Login Delete Create

The ERD address (elrond abbreviated) address data it's required by the elrond blockchain to locate the adress and all info related

This space is dedicated to monitoring your actions, records, nft's and a lot more within elrond blockchain. This app and his creator takes none responsibility for any loss may cost you. You are the only one responsible of any data you give.

Login Page

The rules for a user to be able to register correctly are:

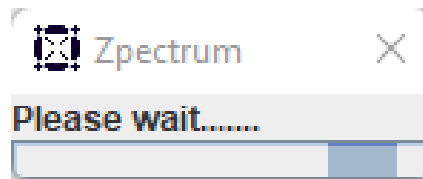
- The user's name must not be empty or repeated in the user list.
- The ERD must be exactly 62 characters long (an ERD with a different length cannot exist) and must not be repeated in the list of users.
- That the wallet with the ERD entered exists. This is checked by making a call to the Elrond API.

If the user positions on the yellow circle with mouse cursor over it, he will see the message below the buttons, which gives us an explanation of what the ERD is:

The ERD adress (elrond abbreviated) adress data it's required by the elrond blockchain to locate the adress and all info related

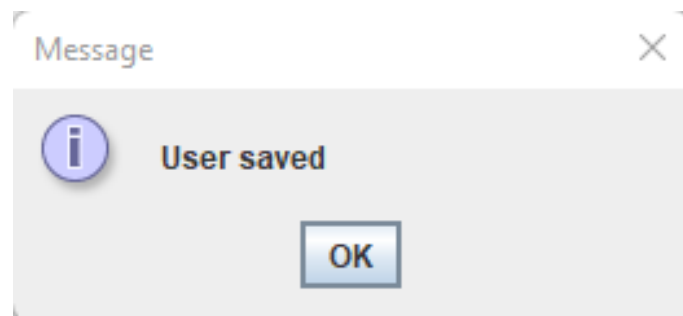
The "Login" button registers the user entered in the DB and displays it in the list on the right if it meets all prerequisites. Internally, it first checks that the username is not empty and the ERD is 62 characters long. Subsequently, an API call is made to check that the ERD exists as a wallet and that the user entered is not already created with the same name in the DB.

While the user is being saved, a window will pop up telling us to wait:



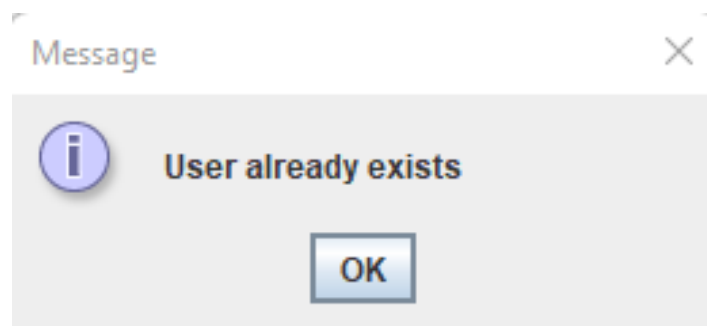
Please wait...

At the end of saving process, it will update the Portfolio page with the user's wallet data and display the following message:



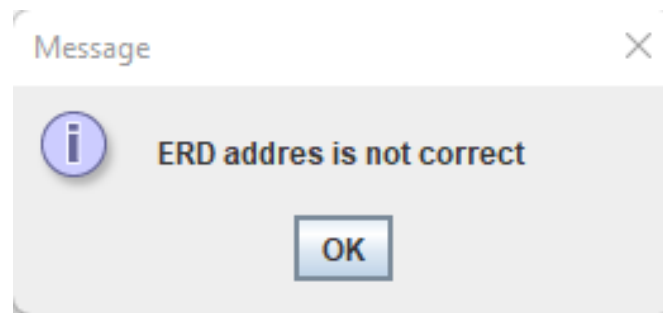
User saved

In case the user's name or ERD exists in the list it will display the following message:



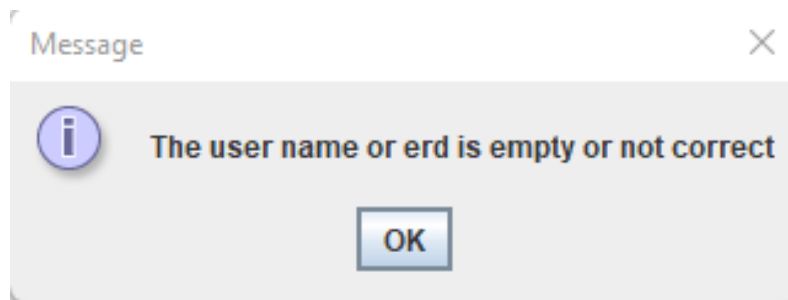
User already exists

If the wallet associated to the ERD does not exist, it will be displayed:



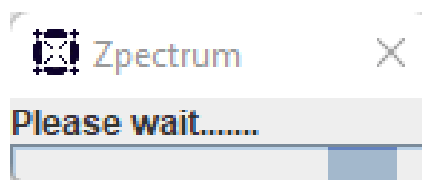
ERD address is not correct

If the username is empty or the length of the ERD is different from 62 it will show us:



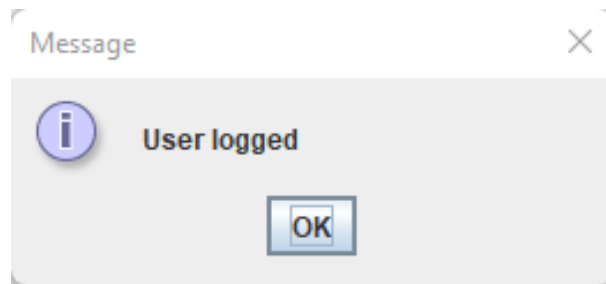
The user's name or ERD is empty or not correct

Another functionality of the "Login" button is that, if the user has selected one of the users in the list and he clicks on it, the app will log in with that user, displaying the following message while it loads:



Please wait...

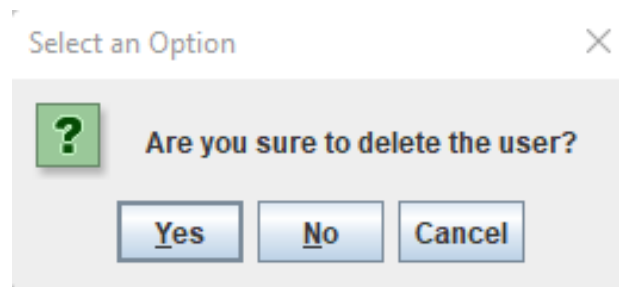
At the end of the load, the app will update the portfolio panel with the user's data and the following message will be shown:



User logged

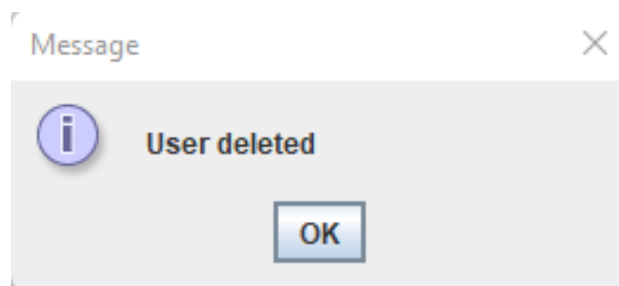
The "Delete" button is used to delete a user from the list, to do this you must have selected the user that you want to delete from the list. Internally it checks that we have a user selected and it will show us a message to ask us if the user is sure that he wants to delete the user, it will look for the name in the DB and it will delete it. Then, it will update the list of users. In case you are logged in with the user you want to delete, it will logout first.

This will be the message to confirm to erase the user:



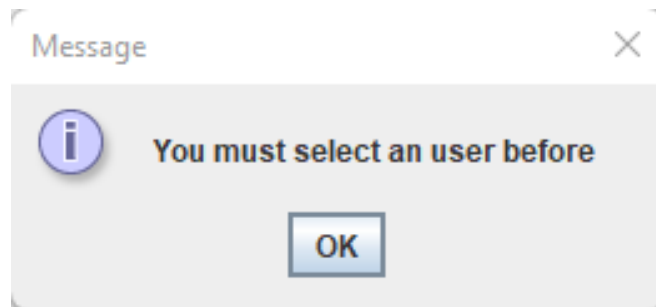
Are you sure to delete the user?

If the user clicks "Yes" it will remove the user from the list and it will show the following message:



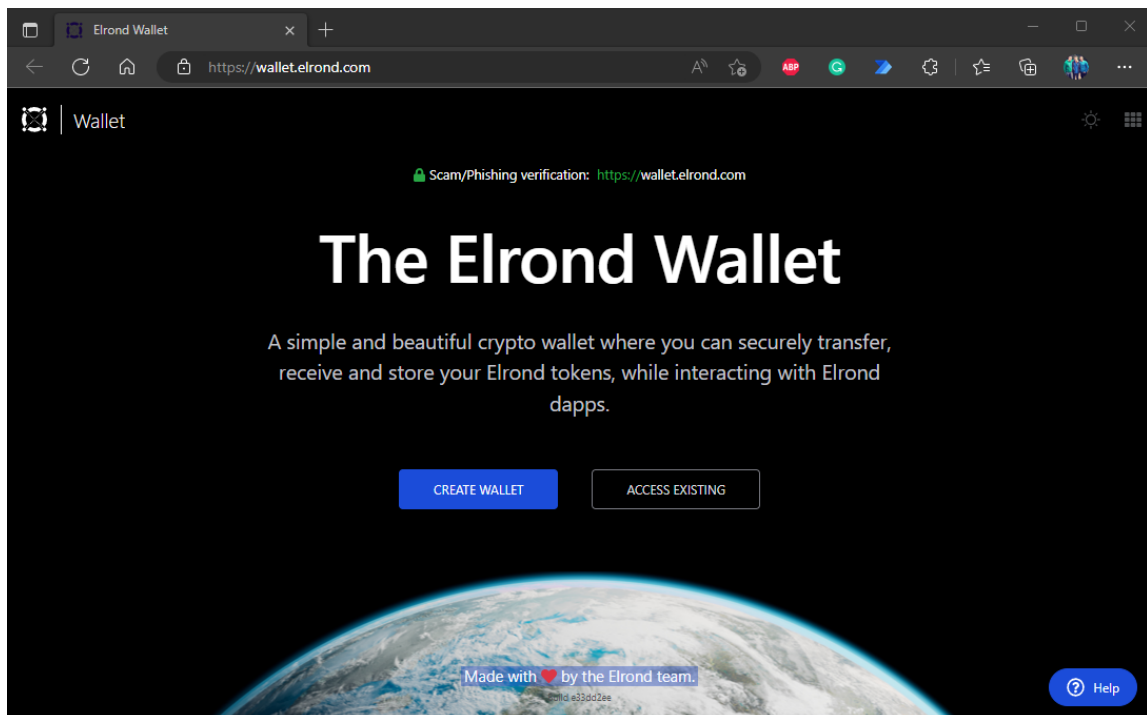
User deleted

If the user clicks on "Delete" and he does not have any user selected or he has selected "There is no users" it will show us the error warning:



You must select an user before

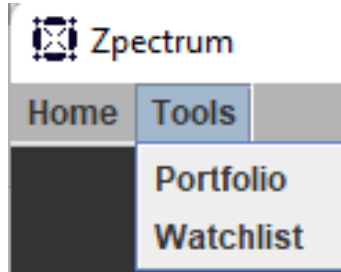
The "Create" button will redirect us to the Elrond page where a new wallet can be created (saving the secret phrase with its respective Json to access it) which can then be registered in the app to be able to manage it:



Elrond wallet web

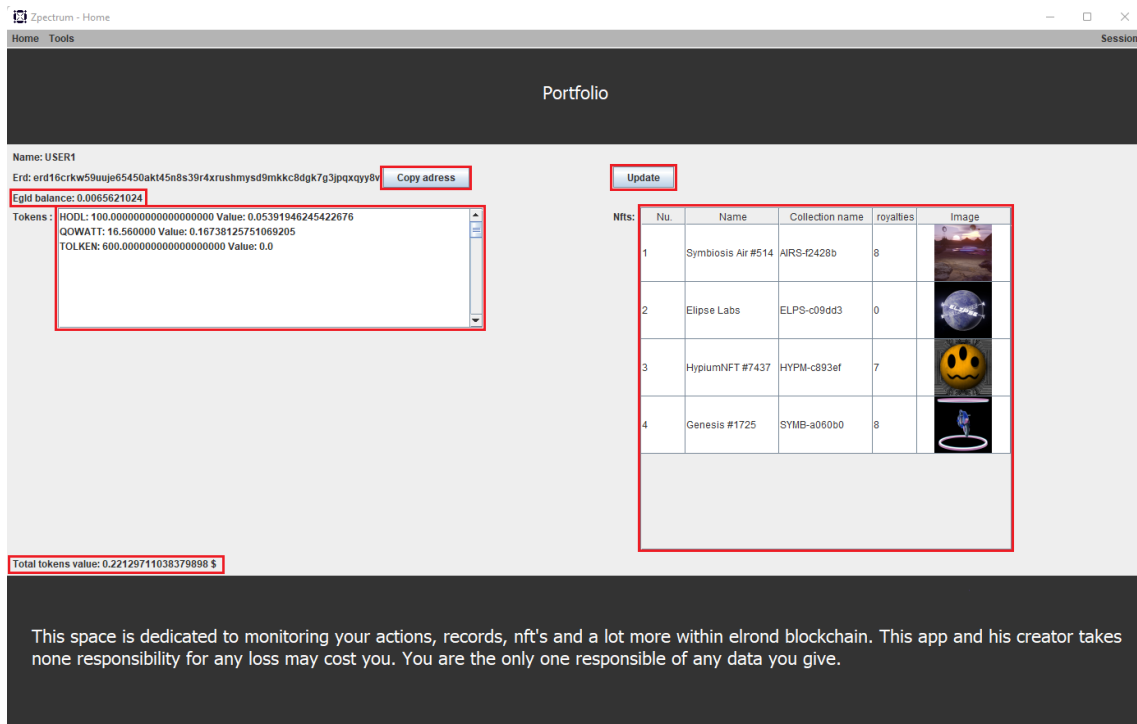
Portfolio

To access the Portfolio page, click on the Tools tab, which opens a drop-down menu, then click on Portfolio:



Tools Menu

It will show the following window which consists of a Cardlayout (PortfolioPanel which is the panel contained within the "layers" panel in MainWindow) containing a MigLayout panel of two columns and six rows in which we find texts showing the user's name, the ERD, the account balance and the total value of the tokens. It also has a list of the tokens in the wallet and a table of the NFTs with their data. And finally two buttons, the "Copy Address" button to copy the ERD to the clipboard and the "Update" button to update the tokens and NFTs.



Portfolio Page

The list of tokens is obtained through a call to the Elrond API which returns a Json with a series of data, from which it will be obtained using Jackson library, processed to show the token name, the token balance and its dollar value.

Tokens :	HODL:	100.000000000000000000	Value: 0.05366550620484729
	QOWATT:	16.560000	Value: 0.16659059417197336
	TOLKEN:	600.000000000000000000	Value: 0.0

Wallet token list





The NFTs table works in the same way, it makes a call to the Elrond API which returns a Json that Zpectrum processes for each NFT, gets a url (it can contain an image or a video) and processes them to display them as an image and save it in DB, finally when the images are processed it adds them to the table with the associated NFTs in which it shows the row number, the name of the NFT, the collection to which the NFT belongs, the royalties which is the economic compensation received by the creator of the NFT and the rendered image.

There is the possibility that the wallet contains an NFT that has no associated image, in this case Zpectrum assigns a default image indicating that there is no image:



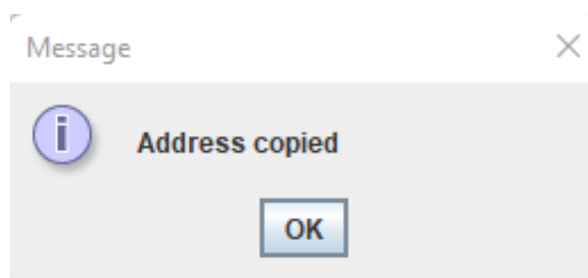
Nft default image

Below is an example of a table of a purse:

Nfts:	Nu	Name	Collection name	Royalties	Image
	1	Symbiosis Air #514	AIRS-f2428b	8	
	2	Elipse Labs	ELPS-c09dd3	0	
	3	HypiumNFT #7437	HYPM-c893ef	7	
	4	Genesis #1725	SYMB-a060b0	8	

Wallet Nft list

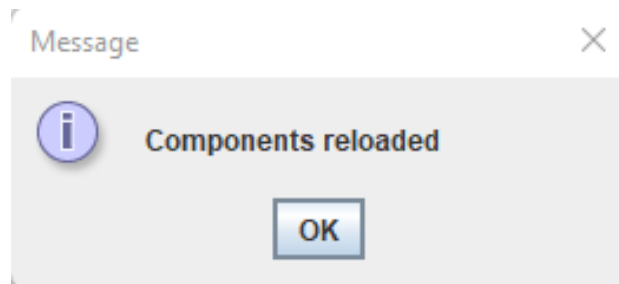
The "Copy Address" button is used to copy the ERD address to the clipboard, an element that is widely used by users who need to use the wallet in different actions. It uses the function "Toolkit.getDefaultToolkit().getSystemClipboard();" to copy the ERD to the clipboard. Finally, it will show us the following message:



Address copied

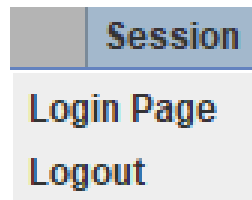
The "Update" button has the function of updating the wallet, it is responsible for repainting both tokens and associated NFT's to check if any tokens or NFT's have been added or sent from the wallet. Internally, it clears both the list of tokens and NFT's and

then repaints the lists with the updated data. When the wallet has been updated it will show the following message:



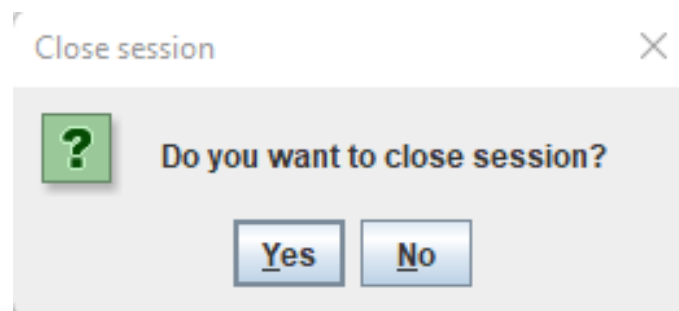
Components reloaded

If the user wants to log out of the wallet we are in, he will need click on the Session menu and then on Logout:



Menu Session

The user will be asked to confirm this operation:



Do you want to close session?

If the user clicks "Yes" it will close the Portfolio page and take us to the Login Page.

Conclusion and future improvements

At this point Zpectrum is a desktop application in which the user can manage a lot of information related to all the wallets that the user wants, visualize the existing tokens within the blockchain to make the decisions that are deemed appropriate making transactions with a more solid base. Manage wallets also with information about the NFT's contained in that wallet in real time, the values of the token balances. This makes Zpectrum a useful tool for today's "Traders" who need all the information possible to optimize their operations, increasing their profits and minimizing their losses.

Due to time constraints, we have not been able to implement some of the things we had planned:

- The Watchlist tab, which will show us statistics and graphs of the wallet we have logged in at that moment.
- Optimize the login of accounts that have many NFTs since it takes too long to download images and videos for each one.

Bibliography

Webs:

- <https://es.stackoverflow.com/>
- www.github.com
- www.wikipedia.com
- <https://www.baeldung.com/>
- <https://javacodehouse.com/index.html>
- <https://spring.io/>
- <https://www.oracle.com/java/technologies/>
- <https://api.elrond.com/>
- <https://docs.elrond.com/sdk-and-tools/rest-api/rest-api/>

Contents taught during the 1st and 2nd year of training in the following areas

<https://institutotecnologico.edix.com/>

Anexo

Hemos subido el código de la aplicación a Github: (Enlace a Github)